

Distributed Stochastic Cross-layer Optimization for Multi-hop Wireless Networks with Cooperative Communications

Shusen Yang, Zhengguo Sheng, Julie A. McCann, and Kin K. Leung
Imperial College London

Abstract—Cooperative communication has been shown to have great potential in improving wireless link quality. Incorporating cooperative communications in multi-hop wireless networks has been attracting a growing interest. However, most current research focuses on either centralized solutions or schemes limited to specific network problems. In this paper, we propose a distributed framework that uses Network Utility Maximization (NUM) to optimize the following joint objectives: flow control, routing, scheduling, and relay assignment; for multi-hop wireless cooperative networks with general flow and cooperative relay patterns. We define two special graphs, Hyper Forwarding Graphs (HFG) and Hyper Conflict Graphs (HCG), to represent all possible cooperative routing policies and interference relations among the cooperative relays respectively. Based on HFG and HCG, a stochastic mixed-integer non-linear programming problem is formulated. We then propose lightweight algorithms to solve these in a fully distributed manner, and derive the theoretical performance bounds of these proposed algorithms. Simulation results verify our theoretical analysis and reveal the significant performance gains of our framework, in terms of throughput, flexibility, and scalability. To our knowledge, this is the first distributed cross-layer optimization framework for multi-hop wireless cooperative networks with general flow and cooperative relay patterns.

I. INTRODUCTION

Cooperative communication schemes [1]–[7] are well-recognized as an effective way of exploiting spatial diversity to significantly improve the quality (e.g. capacity and reliability) of wireless links at the physical layer. The key idea is that multiple wireless single-antenna devices cooperatively share their antenna resources and aid each other’s wireless transmission by forming virtual and distributed antenna arrays. During the last decade, such schemes have been extensively studied at the physical layer in two-hop wireless networks.

Recently, there has been an increasing interest in incorporating cooperative communication schemes in multi-hop wireless networks [8]–[12] to improve the end-to-end network performance such as throughput [9], energy consumption [13], and reliability [12]. However, these proposed schemes are often centralized or restrictive. Specifically, most of them focus on specific cases such as network topologies (e.g. two-hop networks [9]), traffic flow patterns (e.g. single source-destination pairs [8]), network problems (e.g. routing [13]), or cooperative relaying patterns (e.g. single relay [10]).

For cross-layer optimization in multi-hop networks with general network topologies and traffic flow patterns, Network Utility Maximization (NUM) approaches [14]–[16] have

shown to be powerful tools, but most of them focus on wireless networks with pure SISO links. The use of NUM techniques for wireless cooperative networks is however very few [9], [17]. Further, such work is still restricted to specific network topologies (e.g. two-hop networks [9]) or cooperative relay patterns (single relay [17]).

In this paper, we propose a distributed cross-layer optimization framework for joint flow control, routing, scheduling, and relay assignment in multi-hop wireless cooperative networks with *general* network topologies, cooperative relaying, and traffic flow patterns, by combining the NUM technique with novel graph-theoretic approaches. Our overarching goal is to design lightweight and efficient distributed algorithms for maximizing network utility (e.g. throughput and fairness) by exploiting the broadcast advantages of wireless transmissions, the potential capacity gains of cooperative transmissions, and useful interactions among different layers.

The proposed framework considers time-varying Rayleigh-fading channels, and the following three one-hop wireless transmission schemes [13]: direct transmission over single-input single-output (SISO) links, broadcast over virtual single-input multiple-output (SIMO) links, and cooperative beamforming (e.g. [18]) over virtual multiple-input single-output (MISO) links. We assume that time division multiple access (TDMA) is adopted at the link layer and use the commonly-used node exclusive model (e.g. [19], [20]) to model interference among hybrid direct, broadcast, and beamforming links. The following four key issues are addressed in our framework.

- **Cooperative Scheduling and Relaying.** At the link layer, how to schedule the hybrid direct, broadcast and beamforming links for concurrent contention-free transmissions. This scheduling problem also determines cooperative relay assignment: the scheduling of a virtual SIMO or MISO link implies that a set of nodes incident to the scheduled link are assigned as cooperative relays.
- **Cooperative Routing.** At the network layer, how to compute optimal end-to-end cooperative routing policies (single path or multipath), which consist of sequences of hybrid one-hop direct, broadcast, and beamforming links.
- **Flow Control.** At the transport layer, how to allocate the rate of each flow to achieve network utility optimization and system stability (i.e. the allocated flow rates can be afforded by the underlying routing, scheduling and time-varying channel capacities).
- **Complexity Reduction.** The optimal cooperative routing

problem [13], [21] and the optimal scheduling problem (even for wireless networks with pure direct links) [22] have been proven to be NP-hard in general. Consequently, the tradeoff between complexity and optimality is also a key issue considered in our framework.

A. Our Contributions

The major contributions of this paper are summarized as follows:

- We propose two specific concepts of graphs, the Hyper Forwarding Graph (HFG) and the Hyper Conflict Graph (HCG), to respectively represent general end-to-end cooperative routing policies and interference relations among hybrid direct, broadcast, and beamforming links under the node-exclusive interference model. Based on HFG and HCG, we formulate a stochastic mixed-integer non-linear programming problem for joint flow control, routing, scheduling, and cooperative relay assignment in multi-hop wireless cooperative networks, characterized by the Rayleigh-fading channel model.
- A distributed global optimal algorithm is developed to solve the formulated optimization problem by using Lagrangian duality theory and novel graph-theoretic approaches. The proposed algorithms automatically adjust flow rates, as well as select forwarding links and cooperative relays based on the time-varying channel state.
- As the optimal solution to the cooperative scheduling problem has high computational complexity, we propose a lightweight greedy algorithm to solve the cooperative scheduling problem in a fully distributed way. In addition, the number of all possible cooperative relays could be of the exponential order of the total number of nodes in the network, but most of them are not useful for forwarding data. To further reduce the system complexity, we develop an effective scheme to delete such useless relays.
- We provide formal proofs for the optimality and convergence of the global distributed system, and derive the worst-case performance and overhead bounds of the greedy scheduling algorithm. Simulation results demonstrate that 80.2% network throughput improvement can be achieved by incorporating cooperative communications, and the performance of the distributed greedy scheme with much less convergence time is close to the optimal (more than 88%).
- We also provide three useful extensions: global outage probability minimization, all possible cooperative routing policies, and Lyapunov queueing systems, which demonstrate the flexibility of our framework.

B. Related Work

1) *Multi-hop Wireless Cooperative Networks*: Recently, there have been increasing research efforts in applying cooperative communication in multi-hop wireless networks [8]–[12], [23], [24]. For distributed approaches, the cross-layer schemes proposed in [23], [25] are heuristic and cannot provide performance guarantees. Based on dynamic programming, energy-aware cooperative routing algorithms are developed in [13],

[26] for general cooperative relaying patterns and network topologies, as for our paper. However, their approaches consider the delivery of individual messages rather than end-to-end flows. [12] presents a distributed routing scheme to minimize end-to-end outage probability based on the Bellman-Ford algorithm. However, it is limited to the linear network topologies and single source-destination pair. [8] presents a centralized approach to maximize the rate of a single end-to-end flow. [10] aims to maximize the minimal rate of multiple competing flows (i.e. max-min optimization). However, this work is not only centralized and deterministic, but also only considers the specific three-node model for cooperative relaying.

2) *Network Utility Maximization*: NUM techniques [14], [27], [28] such as Lagrange duality and Lyapunov optimization have been studied extensively for various problems in wireless networks with pure direct links, including routing [29], flow control [30], random access control [28], power control [31], and scheduling [32]. In addition, NUM-based approaches have also been used in network coding and opportunistic routing to exploit broadcasting advantages (i.e. receiver-side diversity) of wireless communications [33]–[35], which are similar to our work. However, none of above focus on wireless networks with cooperative communications that exploit the wireless diversities on both the receiver and transmitter sides. Recent comprehensive surveys of NUM research can be found in [14]–[16].

3) *Using NUM in Wireless Cooperative Networks*: Little NUM-based research exists for wireless cooperative networks [9], [17]. In [9], a *centralized* approach is proposed that extends backpressure algorithms to *two-hop* wireless networks with general cooperative communication patterns. In contrast to [9], we focus on a *fully distributed* approach for *multi-hop* wireless cooperative networks with *arbitrary* network topologies. [17] formalizes a deterministic convex optimization problem for joint congestion control and power control. However, it assumes restrictive cooperative relaying (i.e. single relay) and traffic patterns (single commodity). More importantly, wireless interference is not addressed. By taking general cooperative relaying patterns, time-varying channel states, and interference among hybrid direct, broadcast, and beamforming into account, the stochastic optimization problem, considered in our framework, is much more realistic and challenging than that in [17].

Compared with the above related work, our cross-layer framework considers not only the distributed and stochastic nature of multi-hop wireless networks, but also general network topologies, traffic flow patterns, and cooperative relaying patterns. To our knowledge, this is the first work that takes all these issues into account for multi-hop wireless cooperative networks.

C. Outline

The remainder of this paper is organized as follows. Section 2 presents the system models and problem formalization. Section 3 proposes the distributed algorithms of our framework and corresponding theoretical analysis. Three useful extensions are provided in Section 4. Section 5 presents extensive

simulations. Finally, we conclude the paper in Section 6. All proofs are placed in Appendices A–D, which can be found in the supplemental material.

II. SYSTEM MODEL

A. Channel Model

We consider a set of statically-deployed nodes that inter-communicate through wireless links. The system operates at discrete time slots $t = \{0, 1, \dots\}$. Each node is equipped with a single omnidirectional antenna. For a wireless transmission, the signal-to-noise ratio (SNR) at the destination y from source x can be modeled as

$$SNR_{x,y} = P_x H_{x,y} / BN_0 \quad (1)$$

where P_x is the transmission power of x , B is the channel bandwidth, N_0 is the noise spectral density, and $H_{x,y}$ is the power gain. We assume that the transmission power is fixed but not necessarily identical for different nodes. Considering both path loss and Rayleigh fading, $H_{x,y}$ is exponentially distributed with mean $\bar{H}_{x,y} = dis_{x,y}^{-\alpha}$, where $dis_{x,y}$ is the Euclidean distance between nodes x and y , and α is the path loss exponent. $H_{x,y}$ is assumed to be independently and identically distributed (i.i.d.) over the time slot. We further assume that power gain is independent across the links.

The wireless network can be described as a directed graph $G(N, L)$, where N is the set of all nodes and L is the set of all direct transmission links (i.e. SISO links). A link $(x, y) \in L$ is considered to exist if the long-term mean $SNR_{x,y}$ is larger than a predefined small threshold; all other weak links are ignored. Every link $(x, y) \in L$ can communicate using direct transmissions. The link capacity of (x, y) is

$$c_{x,y} = B \log_2(1 + SNR_{x,y}) \quad (2)$$

It is worth noting that the actual transmission data rate should be smaller than the link capacity. To focus on the systematic cross-layer approach, we do not consider specific physical-layer details (e.g. modulation and coding), and we assume that (2) is the maximum transmission data rate that can be achieved without a decoding error¹. However, the essence of our cross-layer approach is nonetheless preserved.

Let $N_x = \{y | (x, y) \in L\}$ be the set of all one-hop neighbors of a node x . Besides direct transmission, our model also considers the following two kinds of one-hop transmission schemes:

Broadcast. A node x can broadcast a message to a set of nodes $RS \subseteq N_x$, $|RS| \geq 2$. In order to ensure that all nodes in RS can correctly receive the message, the link quality of a broadcast transmission is constrained by the minimum $SNR_{x,z}$, $z \in RS$. Therefore, the capacity of the virtual SIMO link (x, RS) can be defined as:

$$c_{x,RS} = B \log_2(1 + \min_{z \in RS} SNR_{x,z}) \quad (3)$$

Cooperative Beamforming. If a set of nodes $RS \subseteq N_y$, $|RS| \geq 2$, have duplicate copies of a message to be forwarded to y , they can phase-align and scale their transmission

signals so that the message can be received coherently by y . The destination node receives multiple copies of the same information transmitted through different wireless channels and the equivalent capacity for virtual MISO link (RS, y) is derived as:

$$c_{RS,y} = B \log_2(1 + \sum_{z \in RS} SNR_{z,y}) \quad (4)$$

For further details and real-world implementation of this physical layer scheme, we refer the reader to [6], [7], [18] and references therein. We assume that the channel capacities (2)-(4) can be estimated at the beginning of every slot. Since power gain $H_{x,y}$ is i.i.d. over time, the capacities (2)-(4) of direct, broadcast, and beamforming links are also i.i.d. over time.

B. Topology Model and Hyper Forwarding Graph

We denote a *relay set* as a set of nodes receiving the same broadcasting messages or sending messages through beamforming. To represent end-to-end cooperative routing policies and all possible relay sets, we define a Hyper Forwarding Graph (HFG) as follows.

Definition 1. A given wireless network $G(N, L)$ has a corresponding Hyper Forwarding Graph $G_f(N \cup \mathcal{R}, L \cup L(\mathcal{R}))$, where \mathcal{R} is the set of relay sets and $L(\mathcal{R})$ is the set of virtual SIMO and MISO links:

$$\mathcal{R} = \{RS | x, y \in N, RS \subseteq N_x \cap N_y, |RS| \geq 2\}$$

$$L(\mathcal{R}) = \bigcup_{x \in N, RS \subseteq N_x \cap \mathcal{R}} ((x, RS) \cup (RS, x))$$

Figure 1(a) shows an example of $G(N, L)$. Since node pairs (1, 5) and (2, 4) shares three neighbors $\{2, 3, 4\}$ and $\{1, 3, 5\}$ respectively, the corresponding HFG shown in Figure 1(b) has 8 relay sets, as well as 16 virtual SIMO and 16 MISO links, where SISO links are represented as black solid lines and virtual SIMO/MISO links are represented as red dashed lines. Note that although Figure 1 (a) and (b) are illustrated as undirected graph for readability, all the links are bidirectional.

HFG can represent a very large class of end-to-end cooperative routing policies. In Figure 1, for instance, routing policy $1 \rightarrow \{2, 3\} \rightarrow 5$ means that node 1 first broadcasts data to nodes 2 and 3, then they send data to node 5 by beamforming. The generalized version of our framework, which considers all possible cooperative routing policies is provided in Subsection 4.2.

For readability, we use the terms a *hyper node* and a *hyper link* to refer to a vertex and an edge in a HFG respectively. In a given HFG, a hyper node $i \in N \cup \mathcal{R}$ can either be a node or relay set. For a hyper node $i \in N \cup \mathcal{R}$, we define a hyper neighbor table $\mathcal{N}_i = \{j | (i, j), (i, j) \in L \cup L(\mathcal{R})\}$. It is obvious that if a hyper node $i \in \mathcal{R}$, then $j \in \mathcal{N}_i$ must be a node; otherwise, j can be either a node or a relay set. In the rest of this paper, unless specifically mentioned, we will use the notations (i, j) and (x, y) to refer to a hyper link and a direct link respectively.

¹We can also define the ϵ - outage capacity such as in [36] to model the capacity-outage tradeoff.

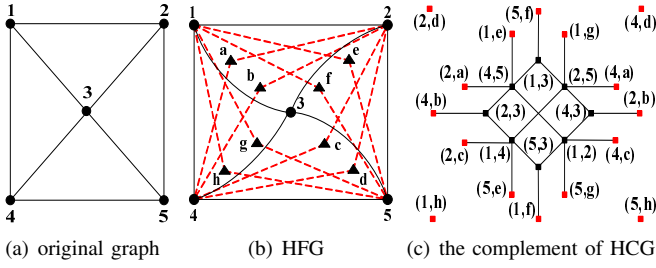


Fig. 1. An example of HFG and HCG. The 8 relay sets are: $a=\{1, 3\}$, $b=\{1, 5\}$, $c=\{5, 3\}$, $d=\{1, 3, 5\}$, $e=\{2, 3\}$, $f=\{2, 4\}$, $g=\{3, 4\}$, $h=\{2, 3, 4\}$.

C. Hyper Conflict Graph and Link Rate Region

We assume that time division multiple access (TDMA) is adopted at the link layer. We use the commonly-used node exclusive model (e.g. [19], [20]) to model interference, under which no node can transmit or receive simultaneously (in the same slot). This captures the half-duplex nature of the majority of current wireless transceiver hardware. To reflect contention relations among hybrid SISO, virtual SIMO, and virtual MISO links, we define the hyper conflict graph (HCG) as follows:

Definition 2: A given HFG has a corresponding HCG $G_c(V, E)$, where every vertex in V represents a hyper link in the HFG (hence $V = L \cup L(\mathcal{R})$). An edge in E means that the two corresponding transmission links in the corresponding HFG can not be both active at the same time. If two hyper links $(i, j), (i', j') \in L \cap L(\mathcal{R})$ are in the following three cases, then the edge $((i, j), (i', j')) \in E$.

Case 1. Both of them are SISO links: the two sets of nodes $\{i, j\}$ and $\{i', j'\}$ satisfy $\{i, j\} \cap \{i', j'\} \neq \emptyset$.

Case 2. One hyper link is a virtual SIMO or MISO link and the other is a SISO link: assume i is the relay set, then $(i \cup \{j\}) \cap \{i', j'\} \neq \emptyset$.

Case 3. Both of them are virtual SIMO or MISO links: assume i and i' are the two relay sets, then $(i \cup \{j\}) \cap (i' \cup \{j'\}) \neq \emptyset$.

For instance, Figure 1 (c) shows the complement of the HCG with regards to the HFG in Figure 1 (b). From HFG and HCG, we can clearly see that both the routing topology and interference relation for multi-hop wireless cooperative networks are much more complex than that of traditional wireless networks with pure SISO links.

A given set of hyper links in a HFG can transmit simultaneously only if it is an independent set² of the corresponding HCG. Define channel capacity vector $\mathbf{c} \in \mathbb{R}_+^{|L \cup L(\mathcal{R})|}$, where each entry c_l is the random capacity of a hyper link l defined by (2)-(4). For a given \mathbf{c} , we define the $|L \cup L(\mathcal{R})|$ -dimensional contention-free rate vector $s^k(\mathbf{c})$ for HCG's k th independent set I^k , where the l 's entry in $s^k(\mathbf{c})$ is

$$s_l^k(\mathbf{c}) = \begin{cases} c_l, & \text{if } l \in I^k \\ 0, & \text{otherwise} \end{cases}$$

Then the link rate region for a given network state \mathbf{c} is

defined as a convex hull of all possible $s^k(\mathbf{c})$:

$$\Pi(\mathbf{c}) = \{s(\mathbf{c}) | s(\mathbf{c}) = \sum_k a^k s^k(\mathbf{c}), a^k \geq 0, \sum_k a^k = 1\}$$

Hence, for a given \mathbf{c} , no $|L \cup L(\mathcal{R})|$ -dimensional vector outside $\Pi(\mathbf{c})$ is feasible for any scheduling policy.

D. Multi-commodity Traffic Flow Model

Let D be the set of all commodities (destinations)³ and $S \subseteq N$ be the set of all source nodes. Let $r_s^d(t)$, $s \neq d$ be the data generation rate of source node s for commodity $d \in D$ at slot t . Define the source rate vector $\mathbf{r}(t) \in \mathbb{R}_+^{|S| \times |D|}$, where each entry is $r_s^d(t)$.

Let $f_{i,j}^d(t)$ be the data forwarding rate over hyper link $(i, j) \in L \cup L(\mathcal{R})$ for commodity d at slot t . We define $f_{i,j}(t) = \sum_{d \in D} f_{i,j}^d(t)$. Due to the link capacity constraint, $f_{i,j}(t) \leq c_{i,j}(t)$. We define the $|L \cup L(\mathcal{R})|$ -dimensional forwarding rate vector $\mathbf{f}(t)$, where each entry is $f_{i,j}(t)$. According to the link rate region constraint, $\mathbf{f}(t) \in \Pi(\mathbf{c}(t))$. Finally, we define the mean rate vectors $\bar{\mathbf{r}} = \lim_{t \rightarrow \infty} 1/t \sum_t \mathbf{r}(t)$ and $\bar{\mathbf{f}} = \lim_{t \rightarrow \infty} 1/t \sum_t \mathbf{f}(t)$.

E. Problem Formalization

Based on the HFG structure and the link rate region constraint, the joint flow control, cooperative routing and scheduling problem is formalized as follows $\forall i \in N \cup \mathcal{R}$:

$$\max_{\mathbf{r}, \mathbf{f}} \sum_{s \in S, d \in D} \overline{U_s^d(r_s^d(t))} \quad (5)$$

$$\text{subject to} \quad \bar{r}_i^d + \sum_{j \in N_i} \bar{f}_{j,i}^d \leq \sum_{j \in N_i} \bar{f}_{i,j}^d, \quad i \neq d \quad (6)$$

$$\mathbf{f}(t) \in \Pi(\mathbf{c}(t)), \forall t \geq 0 \quad (7)$$

where the transport-layer utility function $U_s^d(r_s^d(t))$ of source node s is a differentiable, increasing, and strictly concave function such as the logarithmic function for the proportional fairness [37]. The objective (5) is to maximize the long-term average of the aggregate utilities, which results in a high throughput and fair flow rate allocation.

Constraint (6) states the flow conservation law at the network layer, i.e. for commodity d in a hyper node i , the sum of long-term average forwarding rates allocated to all i 's outgoing links must be not less than that of its all incoming links plus its source rate (note that $r_i^d = 0$, $\forall i \notin S$). In addition, this constraint also implies that flow splitting and multipath routing may be used to achieve the optimality. Since neither specific relay sets nor end-to-end routes are assigned in advance, $\bar{f}_{i,j}^d, (i, j) \in L \cup L(\mathcal{R}), d \in D$ represents the long-term optimal average cooperative relay set assignments and routing policies.

Constraint (7) considers both physical-layer capacity and link-layer interference constraints, which ensures that $\mathbf{f}(t)$ should be in the link rate region for every slot t . Due to the stochastic and discrete nature of $\Pi(\mathbf{c}(t))$, problem (5) is in

²An independent set is a set of vertices in a graph, no two of which are adjacent.

³For notational brevity, we assume that each commodity only has one destination. Our framework can be straightforward to extend to the context of multi-destination commodities.

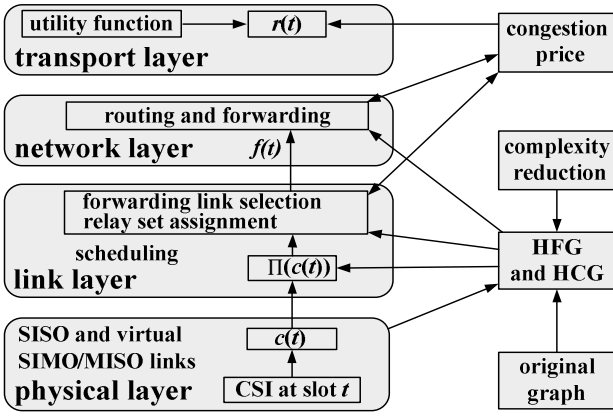


Fig. 2. Architecture of the cross-layer framework.

the form of stochastic mixed-integer programming, which is generally difficult to solve (e.g. #P-hard [38]).

It worth noting that the form of problem (5) looks similar as existing NUM-based flow control formalizations in wireless networks with pure SISO links e.g. [16], [31]. However, the constraints (6) and (7) are established based on the proposed HFG and HCG respectively rather than the original graph. Therefore, problem (5) has a much more complex structure in terms of cooperative routing and scheduling (e.g. the cooperative routing policies and interference relations in Figure 1) than existing NUM-based flow control problems in wireless networks with pure SISO links.

For notational brevity, this paper defines the objective (5) as the time-average utility $\sum_{s \in S, d \in D} \bar{U}_s^d(r_s^d(t))$ rather than the utility of time-average source rates $\sum_{s \in S, d \in D} U_s^d(r_s^d(t))$. Due to Jensen's inequality, $\sum_{s \in S, d \in D} \bar{U}_s^d(r_s^d(t)) \leq \sum_{s \in S, d \in D} U_s^d(r_s^d(t))$. However, the approaches introduced in this paper can be easily extended to maximize the $\sum_{s \in S, d \in D} U_s^d(r_s^d(t))$, by introducing auxiliary variables as in [15].

III. JOINT ALGORITHMS FOR FLOW CONTROL, COOPERATIVE ROUTING AND SCHEDULING

This section presents distributed solutions to problem (5). The architecture of the proposed distributed cross-layer framework is shown in Figure 2. In the initialization phase, nodes establish the HFG and corresponding data structures based on the original $G(N, L)$ in a fully distributed manner (Subsection 3.1). For every time slot t , each node obtains the Channel State Information (CSI) and computes the capacities of its corresponding links according to (2)–(4). Then each node operates a distributed global algorithm (Subsection 3.2) to allocate source rates $r(t)$ and hyper link forwarding rates $f(t)$ based on a congestion price and cooperative scheduling (Subsection 3.3). Here the congestion price is the Lagrangian multiplier associated with constraint (6), and this is proportional to the data queue backlog for a commodity in a node (e.g. [31]). In addition, a complexity reduction scheme is developed in Subsection 3.4 to delete useless cooperative relays and virtual SIMO/MISO links. We assume that all control messages used in our distributed algorithms are error-free.

A. Initialization–Distributed HFG Establishment

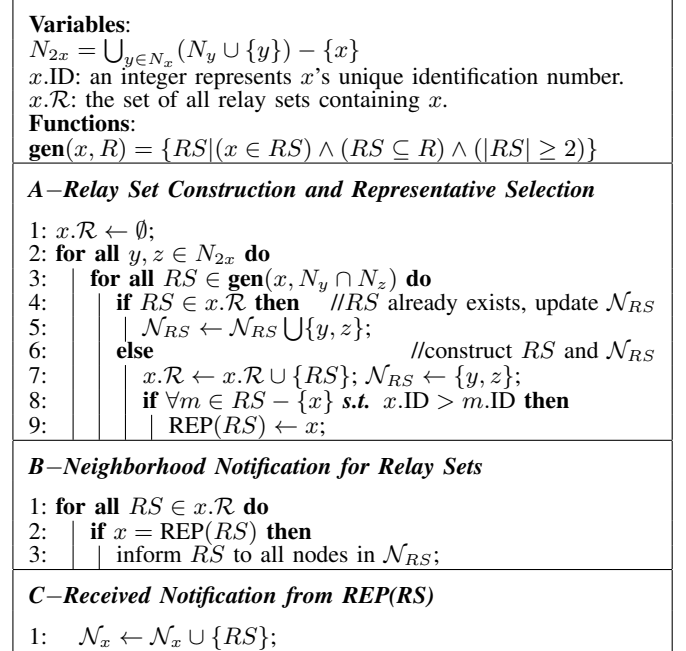


Fig. 3. The pseudocode of Algorithm 1: Distributed HFG establishment for every node $x \in N$.

After the deployment of the wireless network $G(N, L)$, every node $x \in N$ can obtain its neighbor table N_x . Then, node x can establish its two-hop neighbor table which contains the set of nodes $N_{2x} = \bigcup_{y \in N_x} (N_y \cup \{y\}) - \{x\}$, by broadcasting a one-hop beacon that contains N_x . To construct HFG in a fully distributed manner, we develop Algorithm 1 shown in Figure 3.

Relay sets are generated by a local function $\mathbf{gen}(x, N_y \cap N_z)$ (line 3, part A). For a set of nodes R and a node $x \in R$, $\mathbf{gen}(x, R)$ returns the set of R 's all subsets containing x . For instance, if R is a set of three nodes $\{1, 2, 3\}$, then $\mathbf{gen}(1, R)$ returns the set of 3 relay sets: $\{1, 2\}$, $\{1, 3\}$, and $\{1, 2, 3\}$.

Every node x stores $x.\mathcal{R}$, the set of all relay sets containing x . For every relay set RS , the node with the maximal ID in RS is selected as the unique representative of RS , denoted as $\text{REP}(RS)$. The set of all other nodes in RS is denoted as $\text{REST}(RS)$, i.e. $\text{REST}(RS) = RS - \{\text{REP}(RS)\}$. In addition, hyper neighbor tables \mathcal{N}_i are also established for all hyper node $i \in N \cup \mathcal{R}$. Above local topology information established by Algorithm 1 is acquired to facilitate the distributed solutions for the problem (5). In the later sections, we will see that only $\text{REP}(RS)$ is on behalf of RS to participate the distributed operations and all other nodes in $\text{REST}(RS)$ may be requested by $\text{REP}(RS)$ for some information about RS .

B. The Global Distributed Cross-layer Algorithm

Before solving the stochastic problem (5), we first consider a deterministic reference system in which the link capacity is constant over time. We relax constraint (6) by introducing a Lagrangian multiplier $\lambda \in \mathbb{R}^{|N \cup \mathcal{R}| \times |D|}$ for each hyper node and commodity. The resulting partial Lagrangian is obtained as

$$L(\mathbf{r}, \mathbf{f}, \boldsymbol{\lambda}) = \sum_{s \in S, d \in D} (U_s^d(r_s^d) - \lambda_s^d r_s^d) + \sum_{i \in N \cup \mathcal{R}} \sum_{j \in \mathcal{N}_i} \sum_{d \in D} f_{i,j}^d (\lambda_i^d - \lambda_j^d) \quad (8)$$

where λ_i^d is the congestion price for commodity d in hyper node i . Then the corresponding dual problem is

$$\min_{\boldsymbol{\lambda} \succeq 0} D(\boldsymbol{\lambda}) = \max_{\mathbf{r}, \mathbf{f}} \{L(\mathbf{r}, \mathbf{f}, \boldsymbol{\lambda})\} \quad (9)$$

where \succeq represents the entry-wise inequality. The dual problem can be hierarchically decomposed into the following two sub-problems:

$$\text{sub-1: } \min_{\boldsymbol{\lambda} \succeq 0} D_1(\boldsymbol{\lambda}) = \max_{\mathbf{r} \succeq 0} \sum_{s \in S, d \in D} (U_s^d(r_s^d) - \lambda_s^d r_s^d) \quad (10)$$

$$\text{sub-2: } \min_{\boldsymbol{\lambda} \succeq 0} D_2(\boldsymbol{\lambda}) = \max_{\mathbf{f} \in \Pi} \sum_{i \in N \cup \mathcal{R}} \sum_{j \in \mathcal{N}_i} \sum_{d \in D} w_{i,j}^d f_{i,j}^d \quad (11)$$

where $w_{i,j}^d = \lambda_i^d - \lambda_j^d$ and Π is the corresponding static link rate region. The first sub-problem (10) is the flow control problem at the transport layer. For the second sub-problem (11), since (i, j) can be either a SISO, virtual SIMO, or virtual MISO link, the determination of $f_{i,j}^d$ can be interpreted as a hybrid cooperative routing, relay assignment, and scheduling problem, ranging from physical layer to network layer. As can be seen, the global problem decomposes into a number of local optimization problems for every source node (the first sub-problem) and for every hyper link (the second sub-problem), and these sub-problems interact through congestion prices.

Since the objective function of first sub-problem is strictly concave, it admits a unique maximizer as for a given λ_s^d :

$$(r_s^d)^* = U_s^{d'-1}(\lambda_s^d) \quad (12)$$

where $U_s^{d'-1}()$ represents the inverse function of the utility function $U_s^d()$'s first derivative. For every hyper link $(i, j) \in L \cup L(\mathcal{R})$, we define the optimal commodity $d_{i,j}^* = \arg \max_{d \in D} w_{i,j}^d$ and corresponding congestion price differential $w_{i,j}^* = \max_{d \in D} w_{i,j}^d$. We assign $f_{i,j}^d = f_{i,j}$, if $d = d_{i,j}^*$; $f_{i,j}^d = 0$, otherwise. Then, for a given $\boldsymbol{\lambda}$, we have the set of optimal scheduled links F as

$$F = \{f_{i,j} | f_{i,j} \in \arg \max_{\mathbf{f} \in \Pi} \sum_{(i,j) \in L \cup L(\mathcal{R})} w_{i,j}^* f_{i,j}\} \quad (13)$$

which is the maximizer of the second sub-problem (11). We will discuss the scheduling problem in detail in Subsection III-C. In this subsection, we suppose that F can be obtained.

Due to the convexity and non-differentiability of dual problem (9), it can be solved by using the sub-gradient algorithm. The algorithm starts from the 0th step with initialized $\boldsymbol{\lambda}(0)$. At the k th step, the vector $\mathbf{g}(\boldsymbol{\lambda}(k)) \in \mathbb{R}^{|N \cup \mathcal{R}| \times |D|}$ is defined as a subgradient of dual function $D(\boldsymbol{\lambda})$ at point $\boldsymbol{\lambda}(k)$, where each entry is

$$g_i^d(\lambda_i^d(k)) = \sum_{j \in \mathcal{N}_i} f_{i,j}^d(k) - \sum_{j \in \mathcal{N}_i} f_{j,i}^d - r_i^d(k), \quad i \neq d \quad (14)$$

Therefore, the the congestion price is updated as

$$\lambda_i^d(k+1) = |\lambda_i^d(k) - \gamma g_i^d(\lambda_i^d(k))|_+, \quad i \neq d \quad (15)$$

where $\gamma > 0$ is a constant step size, and $|a|_+ = a$, if $a > 0$; $|a|_+ = 0$, otherwise. If $i = d$, then $\lambda_i^d(k) = 0$ for all $k \geq 0$.

Based on by the above deterministic reference system, we develop Algorithm 2 shown in Figure 4; the global cross-layer algorithm for the original stochastic problem (5), in which channel capacity $c_{i,j}(t)$ varies over slot t due to fading. In Algorithm 2, all variables in slot t correspond to the k th step of the subgradient algorithm in the deterministic reference system. We assign an upper bound r_{max} of the flow rate maximizer (line 1, part A) to ensure the convergence of Algorithm 2 (Theorems 1 and 2 below). r_{max} can be set to be sufficiently large to remain the optimality of Algorithm 2. $F(t)$ is the output of the cooperative scheduler at slot t , which will be discussed in next subsection in detail. It is obvious that Algorithm 2 shows the flow control, scheduling, and routing (i.e. next hop selection and data forwarding) processes for every slot $t > 0$. It also implies the relay set assignment: if a virtual SIMO link (x, RS) or MISO link (RS, x) is scheduled, then relay set RS is assigned in a slot t .

Input: Constant step size γ , and $\lambda_i^d(0), \forall i \in N \cup \mathcal{R}, d \in D$.	
Output: Optimal $r_s^d(t)$, $f_{x,y}^d(t)$, and $f_{z,RS}^d(t)$, $f_{RS,z}^d(t), \forall t > 0, x \in N, y \in N_x, RS \in \mathcal{R}, z \in \mathcal{N}_{RS}$.	
A—A node x at time slot t.	
01: if $x \in S$ then	//flow control
02: for all $d \in D$ do	
03: $r_x^d(t) \leftarrow \min(r_{max}, U_s^{d'-1}(\lambda_x^d(t));$	
04: $F(t) \leftarrow \text{transmission_scheduler}(\boldsymbol{\lambda}(t));$	//scheduling
05: for all $i \in \mathcal{N}_x, d \in D$ do	//routing for SISO/SIMO links
06: if $(f_{x,i}(t) \in F(t)) \wedge (d = d_{i,j}^*)$ then	
07: $f_{x,i}^d(t) \leftarrow w_{x,i}^d(t) c_{x,i}(t);$	//forwarding data for d
08: else	
09: $f_{x,i}^d(t) \leftarrow 0;$	
10: for all $d \in D, d \neq x$ do	//update λ_x^d
11: $\lambda_x^d(t+1) \leftarrow \lambda_x^d(t) - \gamma g_x^d(t) _+$	
B—A relay set RS at time slot t.	
//REP(RS) is on behalf of RS to participate the operations.	
1: $F(t) \leftarrow \text{transmission_scheduler}(\boldsymbol{\lambda}(t));$	//scheduling
2: for all $z \in \mathcal{N}_{RS}, d \in D$ do	//routing for virtual MISO links
3: if $(f_{RS,z}(t) \in F(t)) \wedge (d = d_{i,j}^*)$ then	
4: $f_{RS,z}^d(t) \leftarrow w_{RS,z}^d(t) c_{RS,z}(t);$	//forwarding data for d
5: else	
6: $f_{RS,z}^d(t) \leftarrow 0;$	
7: for all $d \in D$ do	// update λ_{RS}^d
8: $\lambda_{RS}^d(t+1) = \lambda_{RS}^d(t) - \gamma g_{RS}^d(t) _+$	

Fig. 4. The pseudocode of Algorithm 2: a distributed algorithm for joint flow control, cooperative routing and scheduling. The operations of a node $x \in N$ and a relay set $RS \in \mathcal{R}$.

1) *Performance Analysis:* Suppose $F(t)$ can be obtained in every slot $t > 0$, then we can show the convergence of Algorithm 2 by Theorems 1 and 2 as follows. The proofs of the two theorems are presented in Appendix A, available in supplemental material.

Theorem 1. *The expected average congestion price $\mathbb{E}[\overline{\boldsymbol{\lambda}(t)}]$ converges statistically to the optimal congestion price $\boldsymbol{\lambda}^*$ as $t \rightarrow \infty$ and $\gamma \rightarrow 0$, where $\overline{\boldsymbol{\lambda}(t)} = 1/t \sum_t \boldsymbol{\lambda}(t)$.*

Theorem 2. *The dual function $\overline{D}(\mathbb{E}[\overline{\boldsymbol{\lambda}(t)}])$ statistically converges to the optimal $\overline{D}(\boldsymbol{\lambda}^*)$, as $t \rightarrow \infty$ and $\gamma \rightarrow 0$,*

where $\overline{D}(\lambda) = 1/t \sum_t D(\lambda)$.

Since the utility function $U_s^d(r_x^d)$ is strictly concave, strong duality holds. Hence, the duality gap is zero. Therefore, Theorem 2 implies that the primary variables $r_s^d(t)$ and $f_{i,j}^d(t)$ converge statistically to the optimal.

C. Scheduling for Hybrid direct, broadcast and beamforming Links

This subsection focuses on the scheduling problem (11). At time slot t , define an undirected link (i, j) with weight $w_{i,j}^s = \max\{w_{i,j}^*(t)c_{i,j}(t), w_{j,i}^*(t)c_{j,i}(t)\}$, for each corresponding pair of directed hyper links (i, j) and (j, i) on HFG. Then, the network state can be represented as an undirected weighted hyper scheduling graph (WHSG) $G_w(N \cup \mathcal{R}, L^u \cup L^u(\mathcal{R}), W(t))$, where $L^u \cup L^u(\mathcal{R})$ is the set of all weighted undirected links. The optimal scheduling problem can be formalized as the following integer programming problem,

$$\max \sum_{x,y \in N} \beta_{x,y} w_{x,y}^s + \sum_{x \in N, RS \in \mathcal{R}} \beta_{x,RS} w_{x,RS}^s \quad (16)$$

subject to

$$\beta_{x,y} = \beta_{y,x}, \beta_{x,y} \in \{0, 1\} \quad (17)$$

$$\beta_{x,RS} = \beta_{RS,x}, \beta_{x,RS} \in \{0, 1\} \quad (18)$$

$$\sum_{y \in N_x} \beta_{x,y} + \sum_{RS \in N_x} \beta_{x,RS} \leq 1 \quad \forall x \in N \quad (19)$$

$$\sum_{x \in N_{RS}} \beta_{x,RS} + \sum_{x \in N_y, y \in RS} \beta_{x,y} \leq 1 \quad \forall RS \in \mathcal{R} \quad (20)$$

The objective (16) is to compute the maximum aggregate weights of all scheduled links. Constraints (17) and (18) state that the binary variable $\beta_{i,j}$ represents the activity of an undirected link $(i, j) \in L^u \cup L^u(\mathcal{R})$. Constraints (19) and (20) ensure that the active links must be contention-free from the aspects of nodes and relay sets respectively, under the node-exclusive model. Denote the optimal solution of problem (16), the set of scheduled links with maximum aggregate total weight, as $S_{opt}(G_w)$.

```

1:  $S_{gre}(G_w) \leftarrow \emptyset$ ;  $L_{rem} \leftarrow L^u \cup L^u(\mathcal{R})$ ;
2: while  $L_{rem} \neq \emptyset$  do
3:   pick locally heaviest link  $(i, j) \in L_{rem}$ ;
4:    $S_{gre}(G_w) \leftarrow S_{gre}(G_w) \cup \{(i, j)\}$ ;
5:   if  $(i, j) \in L$  then // both  $i$  and  $j$  are nodes
6:     | remove  $(i, j)$  and links incident to  $(i, j)$  from  $L_{rem}$ ;
7:   if  $(i, j) \in L(\mathcal{R})$  //  $i$  or  $j$  is a relay set
8:     | remove  $(i, j)$ , all links incident to  $(i, j)$ , and all links
      | incident to all elements of the relay set from  $L_{rem}$ ;

```

Fig. 5. Pseudocode of Algorithm 3, a centralized cooperative greedy scheduling scheme.

Since integer programming is NP-hard in general, it is prohibitive to compute the optimal solution of (16) at every slot t . To reduce complexity, we propose Algorithm 3 shown in Figure 5, a greedy solution to problem (17), inspired by the Longest-Queue-First (LQF) greedy scheduling schemes used in wireless networks with pure SISO links (e.g. [39]). The output of Algorithm 3 is denoted as $S_{gre}(G_w)$. To compute

$S_{gre}(G_w)$ in a fully distributed way, we develop Algorithm 4 summarized in Figures 6 and 7.

Variables:

$x.\mathcal{FR}$: the set of x 's all free relay sets, $x.\mathcal{FR} \subseteq x.\mathcal{R}$.

$x.\mathcal{FN}$: the set of x 's all free neighbors, $x.\mathcal{FN} \subseteq N_x$.

$RS.\mathcal{FN}$: the set of all free neighbors of $RS \in x.\mathcal{FR}$.

$x.optimal_state$: one end of the x 's locally heaviest hyper link which could be either x itself or a relay set $RS \in x.\mathcal{FR}$.

$x.optimal_neighbor$: the other end of x 's local heaviest link.

$x.schedule_ready$: a boolean value represents whether x is ready to schedule.

$x.schedule_state$: a boolean value that represents whether x has been scheduled or not.

Control Messages:

scheduling-apply (SA), scheduling-reply(SR), and drop

Functions:

send(T , source, destination): the source multicasts/unicasts a one-hop control message with type $T \in \{SA, SR, drop\}$ to the destination.

C_query(RS): Discussion between $REP(RS)$ and $REST(RS)$ to check whether a link incident to RS is locally optimal of all nodes in RS .

C_confirm(RS): when a link incident to relay set RS is scheduled, $REP(RS)$ informs this information to $REST(RS)$.

Fig. 6. Definitions of Algorithm 4 (distributed greedy scheduling) for a node $x \in N$.

In Algorithm 4, a node or a relay set that is not involved in any scheduled hyper link is termed as **free**. After initialization in part A, every node $x \in N$ executes and repeats the *while loop* of part B and processes the triggered events (parts C–E), until the text condition of the *while loop* occurs, i.e. x itself or a relay set containing x is scheduled ($x.schedule_state=true$), or x has no free neighbor ($x.\mathcal{FN} = \emptyset$).

Algorithm 4 has two levels of operations. The upper level is the greedy scheduling for all links in $L^u \cup L^u(\mathcal{R})$, based on selecting the locally optimal link and exchanging three one-hop control messages: schedule apply (SA), schedule reply (SR) and drop. For a relay set RS , only $REP(RS)$ is on the behalf of RS for the upper level scheduling process. The lower level is to ensure that information symmetry between $REP(RS)$ and $REST(RS)$ of a relay set $RS \in \mathcal{R}$, which is implemented by two functions $C_query(RS)$ and $cooperative_confirm(RS)$. The interface between the upper and lower level operations is the variable $x.schedule_ready$. $REP(RS)$ is qualified to send a SA or reply a SR only if $REP(RS).schedule_ready=true$ (line 14 in B and line 10 in C).

At the upper level, every node x selects a local heaviest-weighted free hyper link $(i_x^*, j_x^*) = (x.optimal_state, x.optimal_neighbor)$ from its own point of view. If $x = i_x^*$ or $x = REP(i_x^*)$, then x sends a SA with source i_x^* to j_x^* (i.e. $x.optimal_neighbor$) in order to request the scheduling of link (i_x^*, j_x^*) . If j_x^* is a node, x directly sends the SA message to it; otherwise x sends the SA to $REP(j_x^*)$ (part B). Either the SA request can be granted (part C), if the link (i_x^*, j_x^*) is also the locally heaviest link for j_x^* ; or (i_x^*, j_x^*) is eventually dropped (parts C and D). If $(i_x^*, j_x^*)^*$ is dropped, then x selects a new locally heaviest link.

At the lower level, for a relay set RS , the two functions


```

A—initialization
1:  $x.\mathcal{FN} \leftarrow \mathcal{N}_x$ ;  $x.\text{scheduled\_state} \leftarrow \text{false}$ ;  $x.\mathcal{FR} \leftarrow x.\mathcal{R}$ ;
2: for all  $RS \in x.\mathcal{FR}$  do
3:    $RS.FN \leftarrow \mathcal{N}_{RS}$ ;

B—locally optimal link selection and scheduling apply
01: while ( $x.\text{scheduled\_state} = \text{false}$ )  $\wedge$  ( $x.\mathcal{FN} \neq \emptyset$ ) do
02:    $w_1^* \leftarrow \max_{j \in x.\mathcal{FN}} w_{x,j}^s$ ;  $j^* \leftarrow \arg \max_{j \in x.\mathcal{FN}} w_{x,j}^s$ ;
03:    $x.\text{optimal\_state} \leftarrow x$ ;  $x.\text{optimal\_neighbor} \leftarrow j^*$ ;
04:    $x.\text{scheduling\_ready} \leftarrow \text{false}$ ;
05:   if  $\exists (RS, y)$  s.t.  $RS \in x.\mathcal{FR}, y \in RS.FN$  then
06:      $w_2^* \leftarrow \max_{(RS, y) \in x.\mathcal{FR} \times RS.FN} (w_{RS,y}^s)$ ;
07:      $(RS^*, y^*) \leftarrow \arg \max_{(RS, y) \in x.\mathcal{FR} \times RS.FN} (w_{RS,y}^s)$ ;
08:     if  $w_2^* > w_1^*$  then  $\parallel x.\text{optimal\_state} \neq x$ 
09:        $x.\text{optimal\_state} \leftarrow RS^*$ ;  $x.\text{optimal\_neighbor} \leftarrow y^*$ ;
10:       if  $x = \text{REP}(x.\text{optimal\_state})$  then
11:          $x.\text{scheduling\_ready} \leftarrow \text{C\_query}(x.\text{optimal\_state})$ ;
12:       else  $\parallel x.\text{optimal\_state} = x$ 
13:          $x.\text{scheduling\_ready} \leftarrow \text{true}$ ;
14:       if  $x.\text{scheduling\_ready} = \text{true}$  then
15:         send(SA,  $x.\text{optimal\_state}$ ,  $x.\text{optimal\_neighbor}$ );

C—received a (SA,  $i, j$ ) message
01: if  $i = x.\text{optimal\_neighbor}$  then
02:   if  $x.\text{optimal\_state} = x$  then
03:      $x.\text{scheduled\_state} \leftarrow \text{true}$ ;
04:     send(SR,  $x, i$ ); send(drop,  $x, x.\mathcal{FN}$ );
05:     for all  $RS \in x.\mathcal{FR}$  s.t.  $\text{REP}(RS) = x$  do
06:       send(drop,  $RS, RS.FN$ );
07:   else  $\parallel x = \text{REP}(x.\text{optimal\_state})$ 
08:     if  $x.\text{scheduling\_ready} = \text{false}$  then
09:        $x.\text{scheduling\_ready} \leftarrow \text{C\_query}(x.\text{optimal\_state})$ ;
10:     if  $x.\text{scheduling\_ready} = \text{true}$  then
11:        $x.\text{scheduled\_state} \leftarrow \text{true}$ ;
12:       send(SR,  $x.\text{optimal\_state}, i$ );
13:       C_confirm( $x.\text{optimal\_state}$ ); send(drop,  $x, x.\mathcal{FN}$ );
14:       send(drop,  $x.\text{optimal\_state}, x.\text{optimal\_state}.FN$ );
15:   else if  $x.\text{optimal\_state} = x$  then  $\parallel j$  is a node,  $x$  deletes  $i$ 
16:      $x.\mathcal{FN} \leftarrow x.\mathcal{FN} - \{i\}$ ;
17:   else  $\parallel j$  is a relay set,  $x$  deletes  $i$ 
18:      $x.\text{optimal\_state}.FN \leftarrow x.\text{optimal\_state}.FN - \{i\}$ ;

D—received a (SR,  $i, j$ ) message
1:  $x.\text{scheduled\_state} \leftarrow \text{true}$ ;
2: if  $x.\text{optimal\_state} = x$  then
3:   send(drop,  $x, x.\mathcal{FN}$ );
4:   for all  $RS \in x.\mathcal{FR}$  s.t.  $\text{REP}(RS) = x$  do
5:     send(drop,  $RS, RS.FN$ );
6: else
7:   send(drop,  $x.\text{optimal\_state}, x.\text{optimal\_state}.FN$ );
8:   C_confirm( $x.\text{optimal\_state}$ );

E—received a (drop,  $i, j$ ) message
1:  $x.\mathcal{FN} \leftarrow x.\mathcal{FN} - \{i\}$ ;
2: for all  $RS \in x.\mathcal{FR}$  do
3:   if  $i \in RS$  then
4:      $x.\mathcal{FR} \leftarrow x.\mathcal{FR} - \{RS\}$ ;
5:   if  $i \in RS.FN$  then
6:      $RS.FN \leftarrow RS.FN - \{i\}$ ;

```

Fig. 7. Operations of Algorithm 4 (distributed greedy scheduling) for a node $x \in N$.

$\text{C_query}(RS)$ and $\text{C_confirm}(RS)$ need to communicate between the $\text{REP}(RS)$ and the node(s) in $\text{REST}(RS)$. We discuss their operations and logical flows based on Figure 8. The action of the $\text{C_query}(RS)$ function is the two-way handshake between $\text{REP}(RS)$ and $\text{REST}(RS)$ during $[t_1, t_2]$ in Figure 8 (a)-(d). If $\text{REP}(RS)$ finds that a free link incident to the relay set RS , say l^* , is its locally heaviest link, it sends a Cooperative Apply (CA) message to $\text{REST}(RS)$ to check whether l^* is also locally optimal for all nodes in $\text{REST}(RS)$. Every node in $\text{REST}(RS)$ responses to $\text{REP}(RS)$

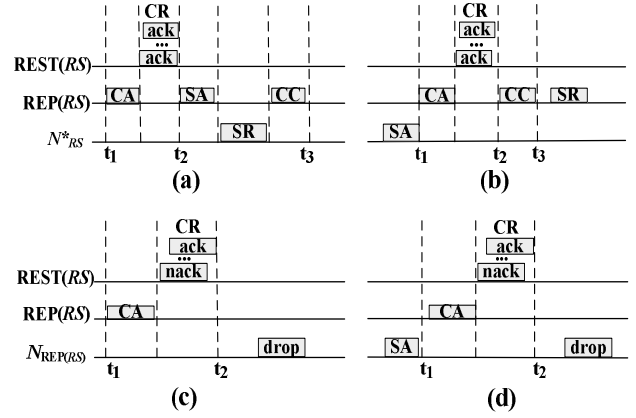


Fig. 8. The time table of distributed scheduling of a hyper link incident to RS , where N_{RS}^* denotes the optimal neighbor of $\text{REP}(RS)$. (a) and (b) shows successful handshakes between $\text{REP}(RS)$ and $\text{REST}(RS)$, while (c) and (d) show two unsuccessful cooperative handshakes.

by sending a Cooperative Reply (CR, ack or nack) message carrying the result. If every node in $\text{REST}(RS)$ replies an ack, $\text{REP}(RS).\text{scheduling_ready}$ is set as true (line 11 in B and line 09 in C), then $\text{REP}(RS)$ is qualified to attend the upper level scheduling (i.e. sends a SA or replies a SR); otherwise, $\text{REP}(RS)$ deletes RS from $\text{REP}(RS).\mathcal{FR}$, and selects its new locally heaviest link.

During $[t_2, t_3]$ shown in Figure 8 (a) and (b), when a link $l^* \in L^u(\mathcal{R})$ incident to a relay set RS is successfully scheduled, $\text{REP}(RS)$ calls the $\text{C_confirm}(RS)$ function (line 13 in C and line 8 in D), i.e. $\text{REP}(RS)$ multicasts a Cooperative Confirm (CC) message to all nodes in $\text{REST}(RS)$ to inform them that l^* is successfully scheduled (l^* is the locally heaviest link for all nodes in RS). Upon receiving the CC message, every node $y \in \text{REST}(RS)$ sets $y.\text{scheduled_state}$ as true, then sends a drop message to all its free neighbors and relay sets RS , $y = \text{REP}(RS)$.

It is worth noting that the upper-level control messages SA, SR and drop are sent over one-hop links and the lower-level messages CA, CR and CC may cover two hops (at most two hops), which is feasible because the two-hop neighbor table $N_{2x}, x \in N$ was established in the initialization phase (i.e. Algorithm 1).

1) *Performance Analysis*: Theorems 3, 4, and 5 below provide analytical results for the communication overhead, convergence and optimality of Algorithm 4 respectively. The proofs of the three theorems and supporting lemmas are presented in Appendix B, available in supplemental material.

Theorem 3. *The total number of control messages sent by Algorithm 4 is not more than $4 + 2 \max_{RS \in \mathcal{R}} |RS|$ per hyper link.*

According to Theorem 3, the per hyper link communication overhead of Algorithm 4 is $O(1)$ with respect to $|L \cup L(\mathcal{R})|$, which demonstrates the scalability of Algorithm 4.

Theorem 4. *Algorithm 4 terminates for every node $x \in N$.*

Theorem 5. *At least $1/\max_{RS \in \mathcal{R}} |RS| + 1$ performance of the optimal solution of problem (16) can be achieved by Algorithm 4, i.e.*

$$W(S_{gre}(G_w)) \geq \frac{1}{\max_{RS \in \mathcal{R}} |RS| + 1} W(S_{opt}(G_w))$$

where $W(S_{gre}(G_w))$ and $W(S_{opt}(G_w))$ are the aggregated weights of greedy scheduling $S_{gre}(G_w)$ and optimal scheduling $W(S_{opt}(G_w))$ respectively.

It is worth noting that Theorem 5 provides a very loose lower bound for the optimality of Algorithm 4. Through simulation in Section 5, we will show that Algorithm 4 can achieve more than 88% performance of the optimal scheduling in practice.

D. Complexity Reduction

It can be seen that $|\mathcal{R}|$ and $|L(\mathcal{R})|$ is of the order of $|N|2^\tau$, where τ is the average network degree of original graph $G(N, L)$. Therefore, even a small-scale (especially dense) network can produce large-scale HFG and HCG (see the example shown in Figure 1). This results in high system complexity such as the low convergence speed of Algorithm 2. To reduce the complexity, we propose a simple scheme operating in the initialization phase (followed by Algorithm 1), which deletes virtual SIMO and MISO links that would almost never be used to forward data.

Definition 3. Consider two nodes x and y such that $|N_x \cap N_y| \geq 2$, then a long-term 2-Hop Hyper Routing Policy (2-HHRP) is denoted as a triple (x, i, y) , $i \in N_x \cap N_y$. Consider the flow conservation law and node-exclusive model, the long-term mean capacity of (x, i, y) , $\bar{C}_{x,i,y}$, can be approximately estimated as $\min(\bar{c}_{x,i}, \bar{c}_{i,y})/2$.

To maximize the aggregate flow rates, Algorithm 2 tends to select hybrid direct and cooperative routes consisting of sequences of 2-HHRP triples (x, i, y) , $x, y \in N, i \in N_x \cap N_y$ with large capacity and small interference (i.e. small-size i). Hence, we propose a complexity reduction scheme as follows:

A. For all $x, y \in N, |N_x \cap N_y| \geq 2$, delete all broadcast links (x, RS) and beamforming links (RS, y) , if $\exists z \in RS$, s.t. $\bar{C}_{x,z,y} \geq \bar{C}_{x,RS,y}$. This is because that both (x, z, y) and (x, RS, y) have the same routing functionality, but $\bar{C}_{x,z,y} \geq \bar{C}_{x,RS,y}$ and (x, z, y) has a smaller interference.

B. For all $RS \in \mathcal{R}$, $x, y \in N$, delete broadcast link (x, RS) and beamforming link (RS, y) , if $RS = N_x \cap N_y$. The reason is that we can find two hyper nodes $i_1, i_2, i_1 \cap i_2 = \emptyset, i_1 \cup i_2 = N_x \cap N_y$ such that the maximum data rate transmitted by using two 2-HHRP triples (x, i_1, y) and (x, i_2, y) is larger than using one 2-HHRP triple $(x, N_x \cap N_y, y)$. This is formally proved in Theorem 6 below.

Theorem 6. For any $x, y \in N, |N_x \cap N_y| \geq 2$, there exist two hyper nodes $i_1, i_2, i_1 \cup i_2 = N_x \cap N_y, i_1 \cap i_2 = \emptyset$ such that $\bar{C}_{x, N_x \cap N_y, y} < (\bar{C}_{x, i_1, y} + \bar{C}_{x, i_2, y})/2$.

Proof. Please refer to Appendix C, which can be found in supplemental material.

The simulation results (Section 5) show that the proposed scheme can significantly reduce system complexity. However, formal analysis of this scheme remains part of our future work.

IV. EXTENSIONS

A. Outage Probability Minimization

Outage probability [1] is a key metric in cooperative communications. Our framework can be extended to balance the tradeoff between network utility and global (multipath end-to-end) outage probability, by introducing an aggregated penalty $-V \sum_{(i,j) \in L \cup L(\mathcal{R})} P_{i,j}^{out}$ in the original objective function (5). Here, $P_{i,j}^{out}$ is the outage probability for a hyper link (i, j) , which is a complex non-linear function of forwarding rate $f_{i,j}$. $V \in [0, +\infty]$ is control parameter that is chosen to affect a desired tradeoff between the network utility and outage probability.

We use a linear approximation for outage probability, $P_{i,j}^{out} \approx cost_{i,j} f_{i,j}$, $(i, j) \in L \cup L(\mathcal{R})$, where the expression of $cost_{i,j}$ is provided in Theorem 7 below.

Theorem 7. For Rayleigh-fading channels, the closed-form expression of $cost_{i,j}$ is

$$cost_{i,j} = \begin{cases} \ln(2) \overline{SNR}_{i,j} / B & i, j \in N \\ |j| (\ln(2))^{|j|} \prod_{y \in j} \overline{SNR}_{i,y} / B & i \in N, j \in \mathcal{R} \\ \ln(2) \sum_{x \in i} Q_x / B & i \in \mathcal{R}, j \in N \end{cases}$$

where

$$Q_x = \frac{\prod_{m \in i} (\overline{SNR}_{m,j})}{\prod_{m \in i, m \neq x} (\overline{SNR}_{m,j} - \overline{SNR}_{x,j})} \quad (21)$$

Proof. Please refer to Appendix D, which can be found in supplemental material.

A fully distributed cross-layer framework that jointly optimizes network utility and global outage probability can be obtained by redefine $w_{i,j}(t)$ as $\lambda_i(t) - \lambda_j(t) - V cost_{i,j}$ in (subproblem11). By redefining $w_{i,j}(t)$, all proposed algorithms can be directly used without modification.

B. All Possible Cooperative Routing Policies

The currently-defined HFG considers a large class of cooperative routing policies, but not all possible routing policies. For instance, assume that there is a flow with source node 4 and destination node 2 in Figure 1, the following routing policy can not be presented by currently-defined HFG: node 4 first broadcasts data to $\{3, 5\}$, then $\{3, 5\}$ send data to 1 by using beamforming, and finally $\{1, 3, 5\}$ send data to 2 using beamforming.

Definition 4. For a given HFG $G_f(N \cup \mathcal{R}, L \cup L(\mathcal{R}))$, the completed hyper forwarding graph (C-HFG), $G_f^c(\mathcal{R}', L'(\mathcal{R}'))$, represents all possible end-to-end cooperative routing policies, where

$$\begin{aligned} \mathcal{R}' &= \left(\bigcup_{x \in N} \{x\} \right) \bigcup \left(\bigcup_{RS \in \mathcal{R}, y \in N_{RS}} \{y\} \cup RS \right) \bigcup \mathcal{R} \\ L'(\mathcal{R}') &= \{(RS_1, RS_2) | RS_1, RS_2 \in \mathcal{R}', RS_1 \neq RS_2, \\ &\quad \exists x \in RS_1, y \in RS_2 \text{ s.t. } x \in N_y\} \end{aligned}$$

C-HFG is defined at the **network layer**. By using C-HFG, the example routing policy mentioned above can be expressed

as $\{4\} \rightarrow \{3, 5\} \rightarrow \{1, 3, 5\} \rightarrow \{2\}$. The well-known three-node cooperative relay pattern [1], [10], [17] can also be represented easily by using C-HFG. In Figure 1, for instance, $\{1\} \rightarrow \{1, 2\} \rightarrow \{3\}$ represents the routing policy that node 1 first sends data to node 2 via direct transmission, then nodes 1 and 2 send data to node 3 using beamforming. Compared with HFG, C-HFG has more vertexes and edges.

At the *physical layer*, we define the sets of all possible SISO and virtual SIMO/MISO links L^{phy} as

$$L^{phy} = \bigcup_{x \in N, i \subseteq N_x \cap R'} \{(\{x\}, i)\} \cup \{(i, \{x\})\}$$

The capacities of each link $(i, j) \in L^{phy}$ can be computed by (2)–(4).

At the *link layer*, we bridge the gap between network-layer routing policies and actual physical-layer data transmissions by defining a link mapping rule

$$M : L'(\mathcal{R}') \rightarrow L^{phy}$$

For an edge $(RS_1, RS_2) \in L'(\mathcal{R}')$, if $RS_2 \subset RS_1$, then $M(RS_1, RS_2) = \emptyset$, since RS_1 can send data to RS_2 without any physical-layer transmission; otherwise,

$$M(RS_1, RS_2) = \{(i, j) | i \subseteq RS_1, j \subseteq RS_2 - RS_1 \cap RS_2 \text{ } (i, j) \in L^{phy}\}$$

For instance, in Figure 1, edge $(\{3, 4\}, \{3, 5\})$ can be mapped into two SISO links (4, 5) and (3, 5), and one virtual MISO link $(\{3, 4\}, 5)$.

Now we discuss how to generalize our framework to consider all possible cooperative routing policies. We can first establish C-HGF in a distributed way by slightly modifying Algorithm 1. Then we can develop a global algorithm similar as Algorithm 2 by introducing a congestion price λ_{RS}^d for every relay set $RS \in \mathcal{R}'$ and commodity $d \in D$. For $(RS_1, RS_2) \in L'(\mathcal{R}')$ in slot t , an unique optimal physical-layer transmission link

$$(i^*, j^*) = \arg \max_{(i, j) \in M(RS_1, RS_2)} c_{i, j}(t)$$

can be obtained. Since $(i^*, j^*) \in L^{phy}$, Algorithm 4 can be directly used for scheduling without modification.

C. Stochastic Queueing Networks

Lyapunov queuing (e.g. [31]) is a popular research area in stochastic network optimization and shows great promise for practical implementations. Our current dual-decomposition based framework can be transferred to the Lyapunov backpressure system, with the following three simple modifications:

1. Transfer the unit of channel capacities (2)–(4) from (*bits per second*) to (*packets per second*).

2. Define the queue backlog q_i^d in every hyper node i for every commodity d , and replace the congestion price λ_i^d throughout this paper by corresponding q_i^d . Instead of using subgradient, the queue backlog updating process is

$$q_i^d(t+1) = |q_i^d(t) + r_i^d + \sum_{j \in \mathcal{N}_i} f_{j, i}^d - \sum_{j \in \mathcal{N}_i} f_{i, j}^d|_+, \quad i \neq d$$

3. Modify the flow controller as

$$r_s^d(t) = \min(r_{max}, U_s^{d' - 1}(q_s^d(t)/V)), \quad \forall s \in S$$

where $V \in [0, +\infty]$ is a control parameter for the trade-off between the network utility and average queue backlog (delay).

V. SIMULATIONS

In this section, we present numerical simulations to demonstrate the convergence, efficiency and performance gain of the proposed algorithms, as well as to provide quantitative understanding of the optimization at different layers. In particular, we compare the following three versions of the proposed cross-layer framework: (1) using pure SISO links and the perfect scheduler⁴ (which we term as direct-optimal); (2) using hybrid SISO and virtual SIMO/MISO links, and the perfect scheduler (cooperative-optimal); (3) using hybrid SISO and virtual SIMO/MISO links, and the distributed greedy scheduler (cooperative-greedy).

A. Simulation Setting

We consider 10 and 25-node networks shown in Figures 9 (a) and (b) respectively. We set channel bandwidth $B = 20$ MHz, transmission power $P_x = 20$ dbm, $\forall x \in N$, noise power $BN_0 = -80$ dbm, and path loss exponent $\alpha = 4$. We use the proportional-fair utility function $U_s^d(r_s^d) = \log(r_s^d)$, $s \in S$, $d \in D$ [37], and set step size $\gamma = 0.3$ and $r_{max} = 20$ Mbps. The channel capacity of every hyper link is computed in every time slot based on nodes' locations and a generator of exponential-distributed random variables (for the Rayleigh-fading power gain).

B. Results

We first consider the 10-node network. The location of each node can be inferred from Figure 9 (a). Two competing multi-commodity flows (4→8) and (5→0) are considered in this network.

Figure 10 (a) shows the logical topology of the original HFG established by Algorithm 1, which contains 28 relay sets, 65 broadcast links, and 65 beamforming links. Figure 10 (b) illustrates the HFG after complexity reduction (the scheme in Section 3.4), including six remaining relay sets ($\{9, 2\}$, $\{1, 2\}$, $\{1, 9\}$, $\{3, 7\}$, $\{5, 3\}$), and $\{7, 5\}$, six remaining broadcast links, and six remaining beamforming links. Note that Figure 10 illustrates the logical topology of the HFGs rather than the actual physical deployment of the 10-node network. It can be seen that the proposed complexity reduction scheme can significantly reduce the scale of the HFG.

Figure 11 (a)-(c) show the source rate evolution of direct-optimal, cooperative-optimal, and cooperative-greedy schemes respectively. It can be seen that the source rate of each flow converges within the neighborhood of a fixed value and oscillates around them, exhibiting limit-cycle behavior. The oscillations are due to the non-differentiability of the dual

⁴The perfect scheduler is a centralized solver for problem (16).

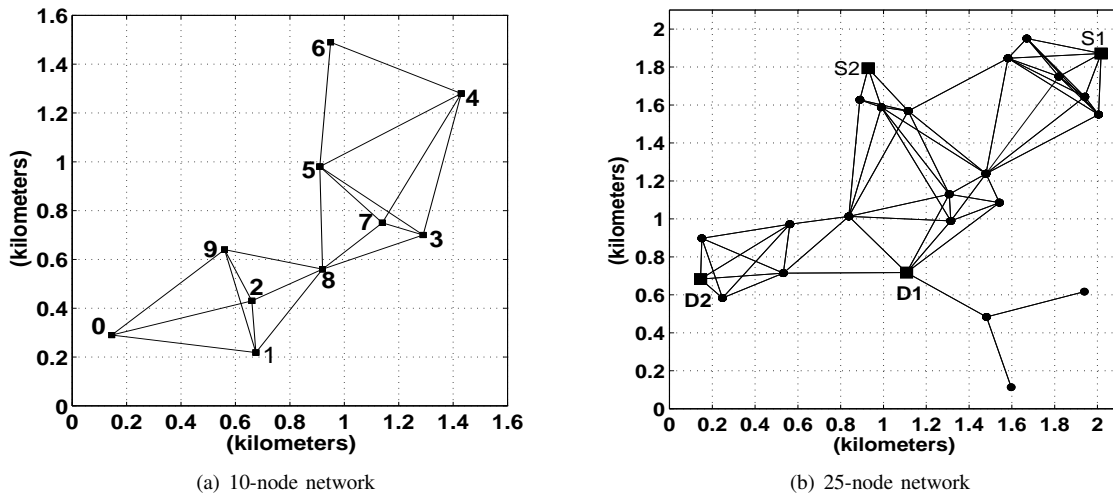


Fig. 9. Network deployment and original graphs used in simulations.

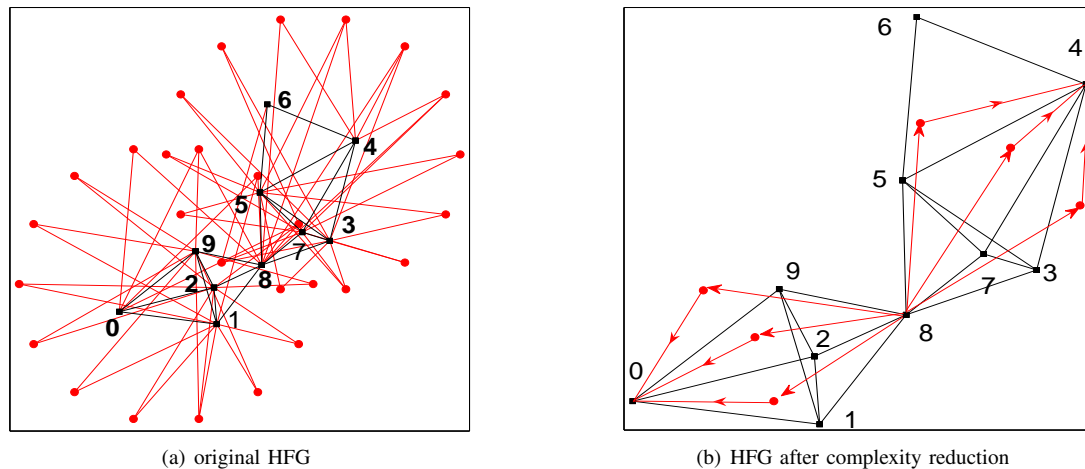


Fig. 10. Hyper forwarding graph and complexity reduction for the 10-node network.

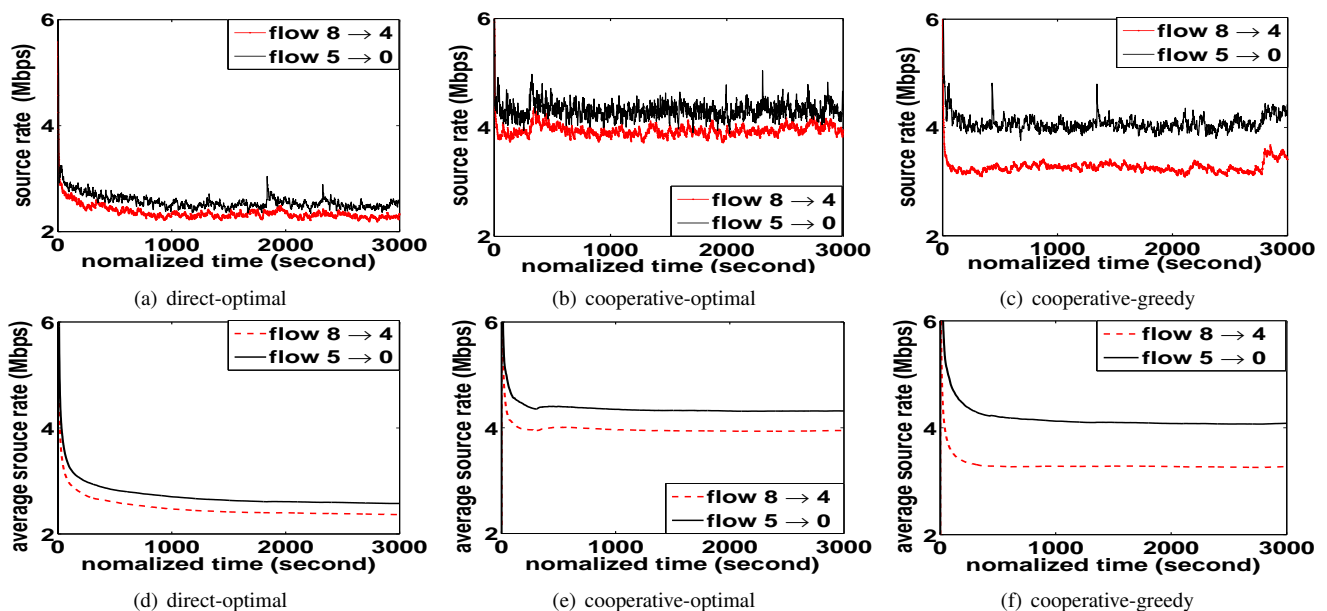


Fig. 11. Convergence results of the three schemes for the 10-node network: (a)-(c) show the evolution of source rates $r_s^d(t)$, (d)-(f) show the convergence of average source rates $r_s^a(t)$.

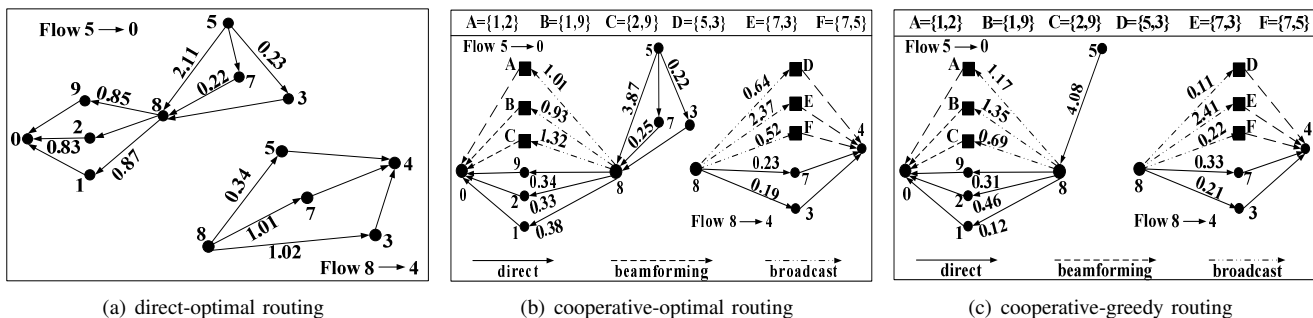


Fig. 12. Long-term optimal routes of the three schemes for the 10-node network.

function and the time-varying channel capacities. It can also be interpreted as the dynamic scheduling process in every slot.

From Figure 11 (d)-(f), we can see that the time-average source rates of all algorithms converge smoothly, which verifies the statistical convergence proof. We can also see that all the three simulations show similar convergence speed, which implies that increasing the network degree (i.e. adding additional relay sets as neighbors for nodes) does not lead to significant reduction in convergence speed. The two reasons for this are: (1) there exist only a small number of relay sets after the complexity reduction; (2) the convergence speed of dual-decomposition schemes is less sensitive to the network degree than to the network diameter⁵.

The average network throughput of cooperative-optimal is $r_8^4 + r_5^0 \approx 8.27$ Mbps, which is about 80.2% higher than that of direct-optimal (around 4.95 Mbps). The throughput can be further improved by extending HFG to C-HFG defined in Subsection 4.2. As expected, the network throughput of greedy-cooperative is less than that of optimal-cooperative (about 7.38 Mbps), but the degradation rate of throughput is only around 10.76%, which demonstrates that the actual performance of the greedy scheduling scheme (i.e. Algorithm 4) is much better than the lower bound provided by Theorem 5. This exciting result demonstrates that the proposed greedy scheduling scheme has great potential to perform well in practical wireless cooperative networks. Figure 12 shows the long-term routing results of the three schemes. It is clear that flow splitting and multipath routing are used to maximize the utilities (and flow rates).

The above results are for the 10-node network. The results of 25-node network are similar and we abbreviate our discussion. As shown in Figure 9 (b), there are two flows $S1 \rightarrow D1$ and $S2 \rightarrow D2$. Figure 13 shows the convergence of average source rates for the three cross-layer schemes. The convergence speeds of this 25-node network simulation are larger than that of 10-node network, but are similar to that of the three cross-layer schemes. The network throughput of cooperative-greedy is around 10.51 Mbps which is about 41.64% higher than that of direct-optimal (7.42 Mbps) and only 1.7% lower than that of cooperative-optimal (10.69 Mbps).

All results above show that our cross-layer framework can indeed improve the network throughput (and utility)

⁵Similar observations can also be found in wireless networks with pure SISO links such as [40].

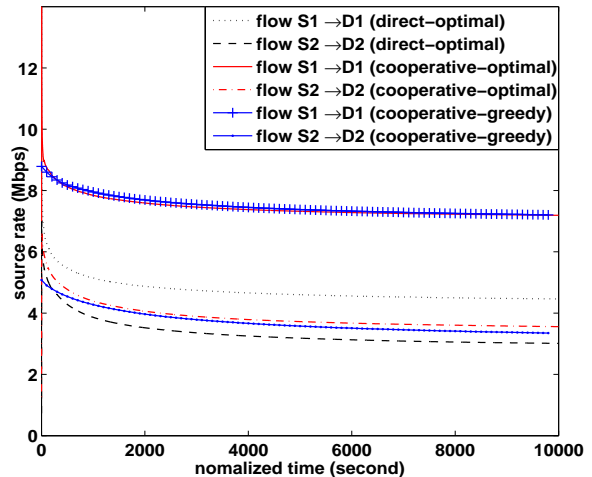


Fig. 13. Convergence results of the 25-node network.

significantly, and the proposed light-weight greedy scheduling algorithm is promising for practical implementations.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a fully distributed cross-layer optimization framework for joint flow control, routing, relay assignment, and scheduling in multi-hop wireless cooperative networks with time-varying fading channels. We first present two specific graphs, THE Hyper Forwarding Graph (HFG) and the Hyper Conflict Graph (HCG) to respectively represent general end-to-end cooperative routing policies and interference relations among hybrid direct, broadcast, and beamforming links. Then we formalize the cross-layer problem as a stochastic mixed-integer non-linear optimization problem, and propose distributed optimal and greedy solutions to the formalized problem, based Network Utility Maximization (NUM) techniques and novel graph-theoretic approaches. The convergence and optimality of the global system is formally proven, and the explicit performance and complexity bounds of the greedy scheduling algorithm are derived. Simulation results verify our theoretical analysis and show the advantages of our approach in terms of convergent speed, network throughput, and the performance of the greedy scheduling. In addition, three useful extensions are also provided to demonstrate the flexibility of the proposed framework.

There are several interesting research directions, including

effective complexity reduction schemes for C-HFG, extensions to general interference models, and NUM using physical SINR interference models. In addition, it could be fruitful to combine stochastic network optimization and cooperative relaying to address other network issues, such as hard QoS guarantees, incentive schemes, and application-layer traffic burstiness.

REFERENCES

- [1] J. N. Laneman, D. N. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.
- [2] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity, part i. system description," *IEEE Tans. commun.*, vol. 51, no. 11, pp. 1927 – 1938, nov. 2003.
- [3] A. Nosratinia, T. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *IEEE Commun. Mag.*, vol. 42, no. 10, pp. 74 – 80, oct. 2004.
- [4] D. Goeckel, B. Liu, D. Towsley, L. Wang, and C. Westphal, "Asymptotic connectivity properties of cooperative wireless ad hoc networks," in *IEEE J. Sel. Areas Commun.*, vol. 27, no. 7, 2009, pp. 1226 –1237.
- [5] E. Liu, Q. Zhang, and K. K. Leung, "Relay-assisted transmission with fairness constraint for cellular networks," *IEEE Trans. Moby. Comput.*, vol. 11, no. 2, pp. 230 –239, 2012.
- [6] H. Rahul, H. Hassanieh, and D. Katabi, "Sourcesync: A distributed wireless architecture for exploiting sender diversity," in *Proc. ACM SIGCOMM*, 2010.
- [7] M. M. Rahman, H. E. Baidoo-Williams, R. Mudumbai, and S. Dasgupta, "Fully wireless implementation of distributed beamforming on a software-defined radio platform," in *Proc. IPSN*, 2012, pp. 305–316.
- [8] D. Chatterjee, T. Wong, and R. A. Berry, "Cooperative transmission in a wireless relay network based on flow management," *IEEE Trans. Commun.*, vol. 58, no. 10, pp. 2816 – 2822, 2010.
- [9] E. M. Yeh and R. A. Berry, "Throughput optimal control of cooperative relay networks," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3827–3833, 2007.
- [10] S. Sharma, Y. Shi, Y. T. HouHanif, D.Sherali, and S. Kompella, "Cooperative communications in multi-hop wireless networks: Joint flow routing and relay node assignment," in *Proc. IEEE INFOCOM*, 2010, pp. 1 – 9.
- [11] L. Zhang, C.-C. Shen, L. Cimini, and B. Gui, "Networking with cooperative communications: Holistic design and realistic evaluation," *IEEE Commun. Mag.*, vol. 47, no. 8, pp. 113–119, 2009.
- [12] R. Babae and N. C. Beaulieu, "Cross-layer design for multihop wireless relaying networks," *IEEE Tans. Wreless Commun.*, vol. 9, no. 11, pp. 3522–3531, 2008.
- [13] R. Madan, N. Mehta, A. Molisch, and J. Zhang, "Energy-efficient decentralized cooperative routing in wireless networks," *IEEE Trans. Auto. cont.*, vol. 54, no. 3, pp. 512 –527, 2009.
- [14] M. Chiang, S. Low, A. Calderbank, and J. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," *Proc. IEEE*, vol. 95, no. 1, pp. 255–312, 2008.
- [15] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.
- [16] Y. Yi and M. Chiang, "Stochastic network utility maximization: A tribute to kelly's paper published in this journal a decade ago," *Eur. Trans. Telecommun.*, vol. 19, no. 4, pp. 421–442, 2008.
- [17] L. Leand and E. Hossain, "Cross-layer optimization frameworks for multihop wireless networks using cooperative diversity," *IEEE Trans. Wreless Commun.*, vol. 7, no. 7, pp. 2592–2602, 2008.
- [18] H. Ochiai, P. Mitran, H. Poor, and V. Tarokh, "Collaborative beamforming for distributed wireless ad hoc sensor networks," *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4110 – 4124, 2005.
- [19] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *Proc. IEEE INFOCOM*, 2006, pp. 1 –13.
- [20] L. Bui, A. Eryilmaz, R. Srikant, and X. Wu, "Asynchronous congestion control in multi-hop wireless networks with maximal matching-based scheduling," *IEEE/ACM Trans. Netw.*, vol. 16, no. 4, pp. 826 –839, 2008.
- [21] F. Li, "Cooperative multicast in wireless networks," *S.M. Thesis, MIT*, 2005.
- [22] Y. Yi, A. Proutiere, and M. Chiang, "Complexity in wireless scheduling: impact and tradeoffs," in *Proc. ACM MOBIHOC*, 2008, pp. 33–42.
- [23] G. Jakllari, S. V. Krishnamurthy, M. Faloutsos, P. V. Krishnamurthy, and O. Ercetin, "A cross-layer framework for exploiting virtual miso links in mobile ad hoc networks," *IEEE Trans. Moby. Comput.*, vol. 6, no. 6, pp. 579 –594, 2007.
- [24] H. Shan, W. Zhuang, and Z. Wang, "Distributed cooperative mac for multihop wireless networks," *IEEE Commun. Mag.*, vol. 47, no. 2, pp. 126 –133, 2009.
- [25] S. Lakshmanan and R. Sivakumar, "Diversity routing for multi-hop wireless networks with cooperative transmissions," in *Proc. IEEE SECON*, 2009, pp. 1–9.
- [26] A. E. Khandani, J. Abounadi, E. Modiano, and L. Zheng, "Cooperative routing in static wireless networks," *IEEE Trans. Commun.*, vol. 55, no. 11, pp. 2185–2192, 2007.
- [27] A. L. Stolyar, "Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm," *Queueing Systems*, vol. 50, no. 4, pp. 401–457, 2005.
- [28] J. Liu, A. L. Stolyar, M. Chiang, and H. Poor, "Distributed queue length based algorithms for optimal end-to-end throughput allocation and stability in multi-hop random access networks," *IEEE Trans. Inf. Theory*, vol. 55, no. 9, pp. 4087– 4098, 2009.
- [29] R. Madan and S. Lall, "Distributed algorithms for maximum lifetime routing in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 5, no. 8, pp. 2185–2193, 2006.
- [30] A. Eryilmaz and R. Srikant, "Joint congestion control, routing, and mac for stability and fairness in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1514–1524, 2006.
- [31] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89 – 103, 2005.
- [32] C.-p. Li and M. J. Neely, "Energy-optimal scheduling with dynamic channel acquisition in wireless downlinks," *IEEE Trans. Moby. Comput.*, vol. 9, no. 4, pp. 527–539, 2010.
- [33] M. J. Neely and R. Urgaonkar, "Optimal backpressure routing for wireless networks with multi-receiver diversity," *Ad Hoc Networks*, vol. 7, no. 5, pp. 862–881, 2009.
- [34] B. Radunović, C. Gkantsidis, P. Key, and P. Rodriguez, "Toward practical opportunistic routing with intra-session network coding for mesh networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 2, pp. 420–433, 2010.
- [35] J. Ho and H. Viswanathan, "On network coding and routing in dynamic wireless multicast networks," in *Proc. of Workshop on Information Theory and its Applications*, 2006.
- [36] Z. Sheng, Z. Ding, and K. K. Leung, "Cooperative wireless networks: From radio to network protocol designs," *IEEE Commun. Mag.*, vol. 49, no. 5, pp. 64 –69, 2011.
- [37] F. Kelly, "Charging and rate control for elastic traffic," *Eur. Trans. Telecommun.*, vol. 8, pp. 33–37, 1997.
- [38] M. Dyer and L. Stougie, "Computational complexity of stochastic programming problems," *mathematical programming*, vol. 106, no. 3, pp. 423–432, 2006.
- [39] C. Joo, X. Lin, and N. B. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks," in *Proc. IEEE INFOCOM*, 2008, pp. 1103 – 1111.
- [40] P. Wan and M. D. Lemmon, "Event-triggered distributed optimization in sensor networks," in *Proc. ACM/IEEE IPSN*, 2009.
- [41] M. Akkouchi, "On the convolution of exponential distributions," *Journal of the Chungcheong Mathematical Society*, vol. 21, no. 4, pp. 501–510, 2008.

APPENDIX A

PROOFS OF THEOREMS 1-2

Proof of Theorem 1. Consider the Lyapunov function $V(\lambda(t)) = \frac{1}{2} \sum_{i \in N \cup \mathcal{R}, d \in D} (|\lambda_i^d(t) - (\lambda_i^d)^*|)^2$, where $|\lambda_i^d(t) - (\lambda_i^d)^*|$ represents the absolute value of $\lambda_i^d(t) - (\lambda_i^d)^*$. Then its conditional expected drift for slot t is:

$$\begin{aligned}
 \mathbb{E}[\Delta V(\lambda(t)) | \lambda(t)] &= \mathbb{E}[V(\lambda(t+1)) - V(\lambda(t)) | \lambda(t)] \\
 &= \mathbb{E}[V(|\lambda(t) - \gamma \mathbf{g}(\lambda(t))|_+) - V(\lambda(t)) | \lambda(t)] \\
 &= \mathbb{E}[V(\lambda(t) - \gamma \mathbf{g}(\lambda(t))) - V(\lambda(t)) | \lambda(t)] \\
 &= \mathbb{E}[-\gamma \mathbf{g}(\lambda(t))^T |\lambda(t) - \lambda^*|_e + \frac{1}{4} \gamma^2 \mathbf{g}(\lambda(t))^T \mathbf{g}(\lambda(t)) | \lambda(t)]
 \end{aligned}$$

where $|(\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}^*)|_e$ represents the entry-wise absolute value of $(\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}^*)$. Using the law of iterated expectations, we have

$$\begin{aligned} & \mathbb{E}[\Delta V(\boldsymbol{\lambda}(t))] \\ &= \mathbb{E}[-\gamma \mathbf{g}(\boldsymbol{\lambda}(t))^T |\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}^*|_e + \frac{1}{4} \gamma^2 \mathbf{g}(\boldsymbol{\lambda}(t))^T \boldsymbol{\sigma} \mathbf{g}(\boldsymbol{\lambda}(t))] \end{aligned}$$

For every entry of subgradient $\mathbf{g}(\boldsymbol{\lambda}(t))$, we have

$$\begin{aligned} \mathbb{E}[g_i^d(\boldsymbol{\lambda}(t))] &= \mathbb{E}\left[\sum_{j \in \mathcal{N}_i} f_{i,j}^d(t) - \sum_{j \in \mathcal{N}_i} f_{j,i}^d(t) - r_i^d(t)\right] \\ &< \mathbb{E}\left[\sum_{j \in \mathcal{N}_i} c_{i,j}(t) + \sum_{j \in \mathcal{N}_i} c_{j,i}(t) + r_i(t)\right] \\ &= 2 \sum_{j \in \mathcal{N}_i} \bar{c}_{i,j} + r_{max} = \sigma_i \end{aligned}$$

where $\bar{c}_{i,j} = \mathbb{E}[c_{i,j}(t)] < \infty$. The first inequality is based on $f_{i,j}^d(t) \leq c_{i,j}(t), \forall t > 0, (i,j) \in N \cup \mathcal{R}, d \in D$. Since $|\mathcal{N}_i|$ is bounded by $|N \cup \mathcal{R}|$, $\sigma_i < \infty$. Hence, we can see that $\mathbb{E}[\mathbf{g}(\boldsymbol{\lambda}(t))]$ is bounded by $\boldsymbol{\sigma} \in \mathbb{R}^{|N \cup \mathcal{R}| \times |D|}$, where each entry is $\max_{i \in N \cup \mathcal{R}} \sigma_i$. Now we have:

$$\mathbb{E}[\Delta V(\boldsymbol{\lambda}(t))] \leq -\gamma \boldsymbol{\sigma}^T \mathbb{E}[|\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}^*|_e] + \frac{1}{4} \gamma^2 \boldsymbol{\sigma}^T \boldsymbol{\sigma} \quad (22)$$

Summing over $k \in \{0, 1, \dots, t-1\}$, we get

$$\begin{aligned} & \frac{1}{t} \boldsymbol{\sigma}^T \left| \sum_{k=0}^{t-1} \mathbb{E}[\boldsymbol{\lambda}(k)] - t \boldsymbol{\lambda}^* \right|_e \\ & \leq \frac{1}{\gamma t} (\mathbb{E}[V(\boldsymbol{\lambda}(0))] - \mathbb{E}[V(\boldsymbol{\lambda}(t))] + \frac{1}{4} \gamma^2 \boldsymbol{\sigma}^T \boldsymbol{\sigma} t) \\ & \leq \frac{1}{\gamma t} (\mathbb{E}[V(\boldsymbol{\lambda}(0))] + \frac{1}{4} \gamma^2 \boldsymbol{\sigma}^T \boldsymbol{\sigma} t) \end{aligned}$$

Taking a lim sup yields, we have

$$\limsup_{t \rightarrow \infty} \left(\frac{1}{t} \sum_{k=0}^{t-1} \mathbb{E}[\boldsymbol{\lambda}(k)] - \boldsymbol{\lambda}^* \right|_e \leq \frac{\gamma^2 \boldsymbol{\sigma}}{4}$$

Hence, we have:

$$\boldsymbol{\lambda}^* - \frac{\gamma^2 \boldsymbol{\sigma}}{4} \leq \limsup_{t \rightarrow \infty} (\mathbb{E}[\overline{\boldsymbol{\lambda}(t)}]) \leq \boldsymbol{\lambda}^* + \frac{\gamma^2 \boldsymbol{\sigma}}{4}$$

Hence $\overline{\boldsymbol{\lambda}(t)}$ converges to $\boldsymbol{\lambda}^*$ statistically as step size $\gamma \rightarrow 0$ and time $t \rightarrow +\infty$. \square

Proof of Theorem 2. Consider (22), we have

$$\begin{aligned} \mathbb{E}[\Delta V(\boldsymbol{\lambda}(t))] &\leq \mathbb{E}[\mathbf{g}(\boldsymbol{\lambda}(t))^T] |\boldsymbol{\lambda}^* - \gamma \mathbb{E}[\boldsymbol{\lambda}(t)]|_e + \frac{1}{4} \gamma^2 \boldsymbol{\sigma}^T \boldsymbol{\sigma} \\ &\leq \gamma |\overline{D}(\boldsymbol{\lambda}^*) - \overline{D}(\boldsymbol{\lambda}(t))|_e + \frac{1}{4} \gamma^2 \boldsymbol{\sigma}^T \boldsymbol{\sigma} \end{aligned}$$

where the second inequality is based on the definition of sub-gradient. Summing over $k \in \{0, 1, \dots, t-1\}$, we obtain

$$\frac{1}{t} \sum_{k=0}^{t-1} \mathbb{E}[\overline{D}(\boldsymbol{\lambda}(k))] \leq \frac{1}{\gamma t} \mathbb{E}[V(\boldsymbol{\lambda}(0))] + \overline{D}(\boldsymbol{\lambda}^*) + \frac{1}{4} \gamma^2 \boldsymbol{\sigma}^T \boldsymbol{\sigma}$$

Taking a lim sup yields, we have:

$$\limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{k=0}^{t-1} \mathbb{E}[\overline{D}(\boldsymbol{\lambda}(k))] \leq \overline{D}(\boldsymbol{\lambda}^*) + \frac{1}{4} \gamma^2 \boldsymbol{\sigma}^T \boldsymbol{\sigma}$$

Since $\overline{D}(\boldsymbol{\lambda})$ is convex, from Jensen's inequality, we get:

$$\begin{aligned} \limsup_{t \rightarrow \infty} \overline{D}(\mathbb{E}[\overline{\boldsymbol{\lambda}(t)}]) &= \overline{D}(\limsup_{t \rightarrow \infty} \mathbb{E}[\frac{1}{t} \sum_{k=0}^{t-1} \boldsymbol{\lambda}(k)]) \\ &\leq \limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{k=0}^{t-1} \mathbb{E}[\overline{D}(\boldsymbol{\lambda}(k))] \leq \overline{D}(\boldsymbol{\lambda}^*) + \frac{1}{4} \gamma^2 \boldsymbol{\sigma}^T \boldsymbol{\sigma} \end{aligned}$$

Since $\forall \boldsymbol{\lambda}, \overline{D}(\boldsymbol{\lambda}) \geq \overline{D}(\boldsymbol{\lambda}^*)$, we have

$$\overline{D}(\boldsymbol{\lambda}^*) \leq \limsup_{t \rightarrow \infty} \overline{D}(\mathbb{E}[\overline{\boldsymbol{\lambda}(t)}]) \leq \overline{D}(\boldsymbol{\lambda}^*) + \frac{1}{4} \gamma^2 \boldsymbol{\sigma}^T \boldsymbol{\sigma} \quad \square$$

APPENDIX B

PROOFS OF THEOREMS 3-5 AND SUPPORTING LEMMAS

Algorithm 4 is an asynchronous random process during slot t of Algorithm 2. We define δ_n as the event that the n^{th} link $(i_n, j_n) \in L^u \cup L^u(\mathcal{R})$ is scheduled (in order of occurrence). δ_0 is the wake up event and δ_1 is the event that the first link is scheduled. We set the initial remaining hyper link set as $L_{rem}^0 = L^u \cup L^u(\mathcal{R})$. When an event δ_n occurs, the L_{rem}^n updates as follows:

If $(i_n, j_n) \in L^u$, then

$$\begin{aligned} L_{rem}^n &= L_{rem}^{n-1} - \{(x, y)\} \cup \{(RS, z)\}, \\ &\quad \forall x \in \{i_n, j_n\}, \forall y \in i_n \cdot \mathcal{FN} \cup j_n \cdot \mathcal{FN}, \\ &\quad \forall RS \in i_n \cdot \mathcal{FR} \cup j_n \cdot \mathcal{FR}, \forall z \in RS \cdot FN \quad (23) \end{aligned}$$

If $(i_n, j_n) \in L^u(\mathcal{R})$, let i_n be the relay set, then

$$\begin{aligned} L_{rem}^n &= L_{rem}^{n-1} - \{(x, y)\} \cup \{(RS, z)\}, \\ &\quad \forall x \in \{k, j_n\}, \forall k \in i_n, \forall y \in k \cdot \mathcal{FN} \cup j_n \cdot \mathcal{FN}, \\ &\quad \forall RS \in k \cdot \mathcal{FR} \cup j_n \cdot \mathcal{FR}, \forall z \in RS \cdot FN \quad (24) \end{aligned}$$

Lemma 1. *There is at most 2 upper-level messages (SA, SR, and drop) sent over each link $(i, j) \in L^u \cup L^u \mathcal{R}$, during the scheduling process.*

Proof. (i, j) can be either scheduled or dropped. If (i, j) is scheduled at δ_n , then a SA and a SR is sent by i and j (or j and i); otherwise, a drop message is sent by i (or j) at δ_n (a SA may also be sent before δ_n). In both cases, link (i, j) will not be part of the scheduling process after δ_n . \square

Lemma 2. *There is at most $2+2|RS|$ lower-level messages (CA, CC, ack, and nack) sent for the 2-way handshake process of $C_query()$ and $C_confirm()$ for every relay set $RS \in \mathcal{R}$*

Proof. Recall that only $\text{REP}(RS)$ is on behalf of RS to attend the upper level scheduling process. If a relay set RS is scheduled, one CA and one CC messages are sent by $\text{REP}(RS)$, and $|RS| - 1$ ack(s) are sent by $\text{REST}(RS)$; otherwise, one CA is sent by the $\text{REP}(RS)$ and $|RS| - 1$ ack(s) or nack(s) are sent by the $\text{REST}(RS)$. Combine this with Lemma 1 and the fact that the maximal hop count between any pair of nodes in RS is 2, Lemma 2 obviously holds. \square

Proof of Theorem 3. For a link $l_1 \in L^u$: at most two upper-level messages are transmitted, no matter l_1 is scheduled or dropped, according to Lemma 1. For a link $l_2 \in L^u(\mathcal{R})$, two upper-level messages and $2+2|RS|$ lower-level messages are transmitted, regardless l_2 is scheduled or dropped, according to Lemmas 1 and 2. In summary, the total control message over

all hyper links is bounded by $(4+2 \max_{RS \in \mathcal{R}} |RS|)|L^u(\mathcal{R})| + 2|L| < (4+2 \max_{RS \in \mathcal{R}} |RS|)(|L(\mathcal{R}) \cup L|)$. \square

Lemma 4. The n^{th} scheduled link $(i, j) \in L^u \cup L^u(\mathcal{R})$ is locally heaviest weighted link for both i and j at δ_n .

Proof. Suppose i sends the SA and j replies SR between δ_{n-1} and δ_n . There are two cases:

Case 1. $(i, j) \in L^u$ (both i and j are nodes). i sends a SA, since $w_{i,j}^s \geq \max(w_{i,k}^s, w_{RS,m}), \forall k \in i.\mathcal{FN}, RS \in i.\mathcal{FR}, m \in RS.FN$; i replies a SR, since $w_{i,j}^s \geq \max(w_{j,k}^s, w_{RS,m}), \forall k \in j.\mathcal{FN}, RS \in j.\mathcal{FR}, m \in RS.FN$.

Case 2. $(i, j) \in L^u(\mathcal{R})$. Let i be the relay set. Then $\text{REP}(i)$ sends a SA with the source i or replies a SR, because (i, j) is locally heaviest for both $\text{REP}(i)$ and all nodes in $\text{REST}(i)$ (by using CA, CR and CC). j replies a SR according to the same reason of case 1.

In both cases, link (i, j) is the locally heaviest link for both i and j . \square

Proof of Theorem 4. For any node $x \in N$, the *while loop* in part *B* of Algorithm 4 has the following cases:

Case 1. Terminate when x or a relay set in $x.\mathcal{R}$ is scheduled.

Case 2. Terminate when its free neighbor table is empty. According to Lemma 1, x and each relay set RS , $\text{REP}(RS) = x$, can receive at most 1 drop or 1 SA from each of their hyper neighbors. From parts *D* and *E*, such non-optimal neighbors will be removed from their free neighbor tables, resulting in the L_{rem}^n updating processes (23) and (24). One condition of the *while loop* $x.\mathcal{FN} = \emptyset$ implies that $RS.FN = \emptyset, \forall RS$ s.t. $\text{REP}(RS) = x$.

Case 3. Wait for a SR message from $x.\text{optimal_neighbor}$. $x.\text{optimal_neighbor}$ will either reply a SR to $x.\text{optimal_state}$ or send a drop when it is scheduled for other link. Therefore, the *while loop* will either stop (when x is scheduled) or search for its next new locally heaviest link (when it receives a drop from $x.\text{optimal_neighbor}$ or a relay set contains $x.\text{optimal_neighbor}$), until case 1 or case 2 happens.

From the above three cases, the *while loop* of each node can terminate. Therefore, Algorithm 4 finishes when all nodes in N terminate. \square

Definition 5. Maximal and optimal schedule of G_w : A maximal schedule S_{maximal} is a schedule (a set of links in $L^u \cup L^u(\mathcal{R})$) such that $\forall l \notin S_{\text{maximal}}, S_{\text{maximal}} \cup \{l\}$ is no longer a schedule any more i.e. $S_{\text{maximal}} \cup l$ is not a feasible solution of problem (16). The optimal schedule S_{opt} is the schedule with the largest aggregated weights which is the optimal maximizer of problem (16).

Lemma 5. When Algorithm 4 terminates, the resulting $S_{\text{gre}}(G_w)$ is a maximal schedule.

Proof. Suppose there exists a link $(i, j) \in L^u \cup L^u(\mathcal{R})$ such that $S_{\text{gre}}(G_w) \cup \{(i, j)\}$ is also a feasible solution of problem (16). Then the scheduled states of both i and j must be false, hence neither i nor j has sent a drop to each other. In addition, neither i nor j has sent a SA and SR to each other, otherwise, the scheduled states of both i and j must be true. Hence, i has at least one free neighbor j , and j has a free neighbor i . Therefore, due to their false scheduled states and non-empty

free neighbor sets, Algorithm 4 does not terminate, which contradicts the termination of Algorithm 4 demonstrated in Theorem 4. \square

Proof of Theorem 5. We compare $S_{\text{gre}}(G_w)$ and $S_{\text{opt}}(G_w)$ by showing that the follow inequality

$$\sum_{k=0}^n w_{i_k, j_k}^s \geq \frac{\sum_{((a,b) \in S_{\text{opt}}(G_w)) \wedge (a \in N_{\text{gre}}^n \vee b \in N_{\text{gre}}^n)} w_{a,b}^s}{\max_{RS \in \mathcal{R}} |RS| + 1} \quad (25)$$

holds for every scheduling event δ_n , $n = 0, 1, 2, \dots$ until Algorithm 4 terminates, where, N_{gre}^n is the set of all scheduled hyper nodes (by using Algorithm 4) before δ_n (includes δ_n). The proof is based on mathematical induction.

Obviously, it holds for δ_0 , since both sides of (25) are equal to zero. Suppose (25) holds at δ_{n-1} for any given n . Then at δ_n , the left hand side increases w_{i_n, j_n}^s . If $(i_n, j_n) \in S_{\text{opt}}(G_w)$, the right hand side increases $w_{i_n, j_n}^s / (\max_{RS \in \mathcal{R}} |RS| + 1)$, then obviously (25) holds. If $(i_n, j_n) \notin S_{\text{opt}}(G_w)$, then there are two cases:

Case 1. $(i_n, j_n) \in L^u$. There are at most two links (a, i_n) , $(b, j_n) \in S_{\text{opt}}^n(G_w)$ incident to i_n, j_n , according to the node-exclusive model. Note that both (a, i_n) and (b, j_n) can be either direct, broadcast, or beamforming links. According to Lemma 4, (i_n, j_n) is the locally heaviest weighted link, and we have $2w_{i_n, j_n}^s \geq w_{a, i_n}^s + w_{b, j_n}^s$. Therefore, the increment of the right hand side is not larger than $2w_{i_n, j_n}^s$. Since $\max_{RS \in \mathcal{R}} |RS| + 1 > 2$, (25) holds for this case.

Case 2. $(i_n, j_n) \in L^u(\mathcal{R})$. Let i_n be the relay set. Under the node exclusive model, there is at most one hyper link $(b, j_n) \in S_{\text{opt}}^n(G_w)$ incident to j_n , and at most $|i_n|$ direct links incident to all elements (nodes) of i_n . Therefore, the increment of the right hand side is not larger than $w_{i_n, j_n}^s / (|i_n| + 1)$, according to Lemma 4. Since $|i_n|$ is upper bounded by $\max_{RS \in \mathcal{R}} |RS|$, (25) holds for this case.

Hence, (25) holds for any scheduling event $\delta_n, n \geq 0$. From Lemma 5, Algorithm 4 terminates with the maximal scheduling. Therefore, we have

$$W(S_{\text{gre}}(G_w)) \geq \frac{1}{\max_{RS \in \mathcal{R}} |RS| + 1} W(S_{\text{opt}}(G_w)) \quad \square$$

APPENDIX C

PROOFS OF THEOREM 6 AND A SUPPORTING LEMMA

We first study the capacity property of an arbitrary beamforming link (RS, y) , $RS \in \mathcal{R}$, $y \in N$.

Lemma 6. Denote a partition (i.e. the set of all non-overlapping non-empty sub-sets) of RS be \mathcal{P}_{RS} , we have:

$$c_{i_1, y}(t) < c_{RS, y}(t) < \sum_{i_2 \in \mathcal{P}_{RS}} c_{i_2, y}(t), \forall \mathcal{P}_{RS}, \forall i_1 \subset RS, \forall t > 0$$

Proof. The left hand side equality obviously holds according

to (4). For the right hand side inequality, we have:

$$\begin{aligned}
c_{RS,y}(t) &= B \log_2 \left(\sum_{z \in RS} 2^{\frac{c_{z,y}(t)}{B}} - |RS| + 1 \right) \\
&= B \log_2 \left(\sum_{i_2 \in \mathcal{P}_{RS}} \left(\sum_{z \in i_2} 2^{\frac{c_{z,y}(t)}{B}} \right) - |i_2| + 1 \right) - |\mathcal{P}_{RS}| + 1 \\
&< B \log_2 \left(\sum_{i_2 \in \mathcal{P}_{RS}} 2^{\frac{c_{i_2,y}(t)}{B}} \right) \\
&\leq B \sum_{i_2 \in \mathcal{P}_{RS}} \log_2^2 2^{\frac{c_{i_2,y}(t)}{B}} = \sum_{i_2 \in \mathcal{P}_{RS}} c_{i_2,y}(t)
\end{aligned}$$

where the first and second equality use the increasing and concave properties of the logarithmic function respectively. \square

Corollary 1.

$$\bar{c}_{i_1,y} < \bar{c}_{RS,y} < \sum_{i_2 \in \mathcal{P}_{RS}} \bar{c}_{i_2,y}, \forall \mathcal{P}_{RS}, \forall i_1 \subset RS \quad (26)$$

Proof of Theorem 6.

$$\begin{aligned}
\bar{C}_{x,N_x \cap N_y,y} &= \min(\bar{c}_{x,N_x \cap N_y}, \bar{c}_{N_x \cap N_y,y})/2 \\
&= \min(\min(\bar{c}_{x,i_1}, \bar{c}_{x,i_2}), \bar{c}_{N_x \cap N_y,y})/2 \\
&< \min(\min(\bar{c}_{x,i_1}, \bar{c}_{x,i_2}), \bar{c}_{i_1,y} + \bar{c}_{i_2,y})/2 \\
&\leq \min(\bar{c}_{x,i_1}, \bar{c}_{i_1,y}) + \min(\bar{c}_{x,i_2}, \bar{c}_{i_2,y})/2 \\
&= (\bar{C}_{x,i_1,y} + \bar{C}_{x,i_2,y})/2
\end{aligned}$$

where the first and second inequalities are due to (26) and the concavity of the $\min()$ function respectively. \square

APPENDIX D PROOF OF THEOREM 7

Let $\mathbb{E}[SNR_{i,j}] = \overline{SNR}_{i,j}$. If (i, j) is a direct link, we get

$$\begin{aligned}
P_{i,j}^{out} &= \Pr[c_{i,j}/B < f_{i,j}/B] \\
&= 1 - \exp(-(2^{f_{i,j}/B} - 1)\overline{SNR}_{i,j}) \quad (27)
\end{aligned}$$

$$\begin{aligned}
&\approx \overline{SNR}_{i,j}(2^{f_{i,j}/B} - 1) \\
&\approx (\ln 2 \overline{SNR}_{i,j}/B) f_{i,j} \quad (28)
\end{aligned}$$

where (27) is due to Rayleigh fading, and the two approximations use first two terms of Taylor expansion. If (i, j) is a broadcasting link, from $f_{i,j} = f_{i,y}, \forall y \in j$ and channel state is independent across links, we have

$$\begin{aligned}
P_{i,j}^{out} &= \Pr[c_{i,j}/B < f_{i,j}/B] \\
&= \Pr[\max_{y \in j} (c_{i,y}/B) < f_{i,j}/B] \\
&= \prod_{y \in j} (1 - \exp(-(2^{f_{i,y}/B} - 1)\overline{SNR}_{i,y})) \\
&\approx (\ln 2)^{|j|} \prod_{y \in j} \overline{SNR}_{i,y} (f_{i,j}/B)^{|j|} \\
&\approx (|j|(\ln 2)^{|j|} \prod_{y \in j} \overline{SNR}_{i,y}/B) f_{i,j}
\end{aligned}$$

If (i, j) is a beamforming link, by using equation (2.3) in [41] we have

$$\begin{aligned}
P_{i,j}^{out} &= \Pr[c_{i,j}/B < f_{i,j}/B] \\
&= \Pr[(\sum_{x \in i} \overline{SNR}_{x,j})/B < f_{i,j}/B] \\
&= \int_0^{f_{i,j}} \sum_{x \in i} Q_x \exp(-\overline{SNR}_{x,j}(2^{\frac{f_{i,j}}{B}} - 1)) d(2^{\frac{f_{i,j}}{B}} - 1) \\
&= \sum_{x \in i} (Q_x (1 - \exp(-\overline{SNR}_{x,j}(2^{\frac{f_{i,j}}{B}} - 1)))) / \overline{SNR}_{x,j} \\
&\approx \sum_{x \in i} (Q_x (\ln 2 \overline{SNR}_{x,j}/B) / \overline{SNR}_{x,j}) f_{i,j} \\
&= (\ln 2 \sum_{x \in i} (Q_x)/B) f_{i,j}
\end{aligned}$$

where Q_x is defined by (21). \square