

Joint Optimization of Session Grouping and Relay Node Selection for Network-Coded Cooperative Communications

Sushant Sharma, *Member, IEEE*, Yi Shi, *Member, IEEE*, Y. Thomas Hou, *Senior Member, IEEE*, Hanif D. Sherali, and Sastry Kompella, *Member, IEEE*

Abstract—Network-coded cooperative communications (NC-CC) is a new paradigm for communications in wireless networks that employs network coding (NC) to improve the performance of CC. A key problem to harness the potential of NC-CC is how to put sessions into different groups, and assign a relay node for each group. In this paper, we study this joint grouping and relay node selection problem for NC-CC. We provide a formal proof of NP-hardness for this problem. Due to NP-hardness, we propose a distributed and online algorithm and show that it offers near-optimal solution to this problem. The key idea in this algorithm is to have each neighboring relay node of a new session calculate the best local group that it can offer and advertise this information; and then to have the source node of the new session select the best local group to join among all offers. We show that our distributed algorithm has polynomial time complexity. Using extensive numerical results, we show that our distributed algorithm adapts well to online network dynamics.

Index Terms—Optimization, Session grouping, Relay node selection, Cooperative communications, Network coding

1 INTRODUCTION

NETWORK-CODED cooperative communications (NC-CC) [3], [21], [22], [25], [26] is a new paradigm for communications in wireless networks. NC-CC exploits network coding (NC) to improve the performance of CC. To see why NC-CC is needed, let's first understand how CC works in a multi-session network and then see how NC could help.

- Under CC, a source node exploits the overhearing and relaying capabilities of its neighboring nodes when it sends data to a destination node. Such path diversity, when exploited appropriately, could lead to a potential increase in the achievable rate [15]. In the simple three-node model [19], [23] shown in Fig. 1, the source node s_0 can exploit a neighboring relay node r when it sends data to its destination node d_0 . Here, a time-frame is divided into two slots. In the first time slot, source node s_0 transmits data to destination node d_0 , which is also overheard by r . In the second time slot, r re-transmits its overheard signal to d_0 (with amplify-and-forward (AF), decode-and-forward (DF), compress-and-forward (CF), or other schemes [15]). The destination node can now

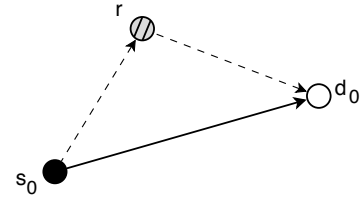


Fig. 1. A reference model for three-node CC.

combine these two signals from different paths. It was shown in [19] that such a CC scheme could increase the achievable rate between nodes s_0 and d_0 over direct transmission.

- When the same relay node is being used by multiple sessions, say N_s sessions, one would divide the time-frame into $2N_s$ time slots, as shown in Fig 2(a). Note that among the $2N_s$ time slots, N_s time slots are used for relaying data for each of the N_s sessions. This is clearly wasteful, and is precisely the place where NC can be leveraged to improve efficiency. It was recognized in [21], [22] that by exploiting NC's capability to combine/aggregate signals inside the network, one can consolidate the N_s time slots used for relaying into just a single time slot as shown in Fig. 2(b). Here, a frame is divided into only $N_s + 1$ time slots. The first N_s time slots are used for transmission by each of the N_s source nodes. Then, the relay node combines all the signals it overhears in the previous N_s time slots and transmits the combined signal in the $(N_s + 1)$ -th time slot. The signal combination process at the relay node is a linear addition of the analog signals received from the source nodes. This

-
- An abridged version of this paper appeared in the Proc. of IEEE INFOCOM 2011, Shanghai, China.
 - S. Sharma is with the Computational Science Center, Brookhaven National Laboratory, Upton, NY, 11973 USA e-mail: (sushant@bnl.gov). This work was completed while he was a Ph.D. student at Virginia Tech.
 - Y. Shi and Y. Thomas Hou are with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, 24061 USA e-mail: (yshi@vt.edu, thou@vt.edu).
 - H.D. Sherali is with the Grado Department of Industrial Engineering, Virginia Tech, Blacksburg, VA, 24061 USA e-mail: (hanifs@vt.edu).
 - S. Kompella is with the Information Technology Division, US Naval Research Laboratory, Washington, DC, 20375 USA (e-mail: sasstry.kompella@nrl.navy.mil).

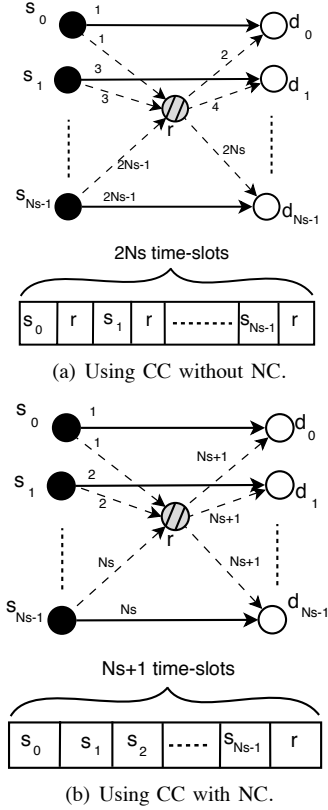


Fig. 2. An example illustrating how NC-CC may improve over CC.

combined signal is then received by all the destination nodes, which can subtract the unwanted signals that were overheard in the first N_s time-slots, thereby extracting their desired signal [22]. We can see that the number of time-slots used by the relay node is reduced to one (from N_s), which significantly increases time-slot efficiency. As a result of the reduction in the number of time-slots, the bandwidth of each time slot for transmission is increased.

Based on our discussion so far, it may appear that for a single relay node, we can group as many sessions as we want. But, in a recent study [22], Sharma *et al.* showed that there exists a so-called “NC noise” at a destination node when extracting the desired signal from the network-coded signal. Further, it was shown that as the group size (i.e., the number of sessions in a group) increases, the NC noise also increases, thereby decreasing the achievable rates. That is, there exists a trade-off between the time slot efficiency and the NC noise. As a result, instead of grouping all the sessions in a single group, it may be necessary to put sessions into different groups in order to keep NC noise under control. However, how to perform session grouping is not a trivial problem.

In this paper, we are interested in a more general setting where there are multiple relay nodes in the network. In this setting, a session has the option to select a relay node from different available relay nodes. So, in addition to session grouping, we also have a relay node selection problem. We study a joint session grouping and relay node selection problem in NC-CC. The goal is to maximize the sum of weighted

session rates in the network. Our main contributions are the following:

- 1) We prove that a generalized weight version of the joint session grouping and relay node selection problem is NP-hard.
- 2) We propose a distributed and online algorithm called D-GRS (Distributed Grouping and Relay node Selection). We show that D-GRS has polynomial-time complexity.
- 3) Using extensive numerical results, we show that D-GRS can offer near-optimal solutions when compared to a centralized solver (CPLEX). Further, it adapts well to online network dynamics.

The remainder of this paper is organized as follows. In Section 2, we describe the joint session grouping and relay selection problem. The proof of NP-hardness is given in Section 3. In Section 4, we propose our D-GRS algorithm to solve the joint session grouping and relay selection problem. Section 5 presents simulation results to demonstrate the performance and time complexity of D-GRS. In Section 6, we discuss related work, and Section 7 concludes this paper.

2 PROBLEM STATEMENT

In this section, we first give some mathematical background to compute the achievable rate in NC-CC (Section 2.1). Then in Section 2.2, we describe the joint session grouping and relay node selection problem.

2.1 Preliminaries

We start with the simple case where all sessions are in the same group and share the same relay node to perform AF CC with the help of analog NC. By identifying the NC noise, we introduce the grouping mechanism and discuss the case where sessions sharing the same relay node can be put into different groups. Finally, we consider the general case where there are multiple relay nodes in the network.

The Single-Group Single-Relay Case. Consider the simple case in Fig. 2(b) where all sessions share the same relay node. Denote \mathcal{S}_r as the set of source nodes $\{s_0, s_1, \dots, s_{N_s-1}\}$ for all sessions in the network. Denote W (in Hz) as the total bandwidth. Let h_{uv} capture the effect of path-loss, shadowing, and fading within the channel between two nodes u and v . Denote P_u as the transmission power at node u . Assume that the background noise at node v has zero mean and a variance of σ_v^2 . Denote SNR_{uv} as the signal-to-noise ratio at receiving node v (for the signal from node u).

Under this setting, the achievable rate for a session (s_i, d_i) can be written as [22]

$$R_{\text{NC-CC}}(s_i, r, \mathcal{S}_r) = \frac{W}{N_s + 1} \cdot I_{\text{NC-CC}}(s_i, r, \mathcal{S}_r), \quad (1)$$

where $I_{\text{NC-CC}}(s_i, r, \mathcal{S}_r)$ is the mutual information between nodes s_i and d_i . The channel that combines both the direct path (s_i to d_i) and the relay path (s_i to r to d_i) can be modeled as a one-input two-output complex Gaussian vector

channel [19]. As a result, under analog NC with AF CC [22], $I_{\text{NC-CC}}(s_i, r, \mathcal{S}_r)$ can be written as

$$I_{\text{NC-CC}}(s_i, r, \mathcal{S}_r) = \log_2 \left(1 + \text{SNR}_{s_i d_i} + \frac{\text{SNR}_{s_i r} \text{SNR}_{r d_i}}{|\mathcal{S}_r| \frac{\sigma_{d_i}^2 \text{NC}}{\sigma_{d_i}^2} + \text{SNR}_{r d_i} + \frac{\sigma_{z_{d_i}^2 \text{NC}}}{\sigma_{d_i}^2} \sum_{s_k \in \mathcal{S}_r} \text{SNR}_{s_k r}} \right), \quad (2)$$

where $\text{SNR}_{s_i d_i} = \frac{P_{s_i}}{\sigma_{d_i}^2} |h_{s_i d_i}|^2$, $\text{SNR}_{s_i r} = \frac{P_{s_i}}{\sigma_r^2} |h_{s_i r}|^2$, $\text{SNR}_{r d_i} = \frac{P_r}{\sigma_{d_i}^2} |h_{r d_i}|^2$, and $\sigma_{z_{d_i}^2 \text{NC}}$ denotes the variance of the NC noise at node d_i under NC-CC. P_{s_i} and P_{r_j} are the transmission powers of source node s_i and relay node r_j , respectively. The value of $\sigma_{z_{d_i}^2 \text{NC}}$ can be written as [22]

$$\sigma_{z_{d_i}^2 \text{NC}}^2 = \sigma_{d_i}^2 + (|\mathcal{S}_r| - 1) (\alpha_r h_{r d_i})^2 \sigma_r^2 + \sigma_{d_i}^2 \sum_{\substack{s_k \neq s_i \\ s_k \in \mathcal{S}_r}} \left(\frac{\alpha_r h_{s_k r} h_{r d_i}}{h_{s_k d_i}} \right)^2, \quad (3)$$

where α_r is the amplification factor at the relay node r and is given by

$$\alpha_r^2 = \frac{P_r}{|\mathcal{S}_r| \sigma_r^2 + \sum_{s_i \in \mathcal{S}_r} P_{s_i} |h_{s_i r}|^2}. \quad (4)$$

A closer look at the NC noise in (3) shows that, as more sessions share the same relay node (i.e., $|\mathcal{S}_r|$ increases), the NC noise also increases monotonically. Further, as NC noise increases, the value of mutual information for each session (s_i, d_i) in (2) decreases. From Eq. (3), we can also see that the NC noise at a destination node d_i depends on three types of channel gains: (i) the channel gains between all the source nodes (that are sharing the relay node r) and d_i , (ii) the channel gains between all the source nodes (that are sharing the relay node r) and the relay node r itself, and (iii) the channel gain between the relay node r and d_i . All the nodes in the network always transmit at a constant power.

The Multi-Group Single-Relay Case. Recognizing the above NC noise problem associated with a single group, one can introduce multiple groups to control the NC noise. This is illustrated in an example in Fig. 3. Here, instead of putting all six sessions in the same group, one can put them into three separate groups. From (3), we find that NC noise is directly tied to the number of sessions in group \mathcal{S}_r . When the number of sessions in a group is reduced, the NC noise for the sessions in the group is also reduced. As a result, the value of mutual information in (2) will increase.

To support multiple groups sharing the same relay node, we need to re-organize time slot structure in a frame. Figure 3(d) shows the proposed time slot structure for multiple groups sharing the same relay. The time slot structure in Fig. 3(d) is based on that in Fig. 3(b), where each session is allocated equal time for direct transmission (i.e., NC-CC is not used). Since s_0, s_1 , and s_2 are now in one group, the total time

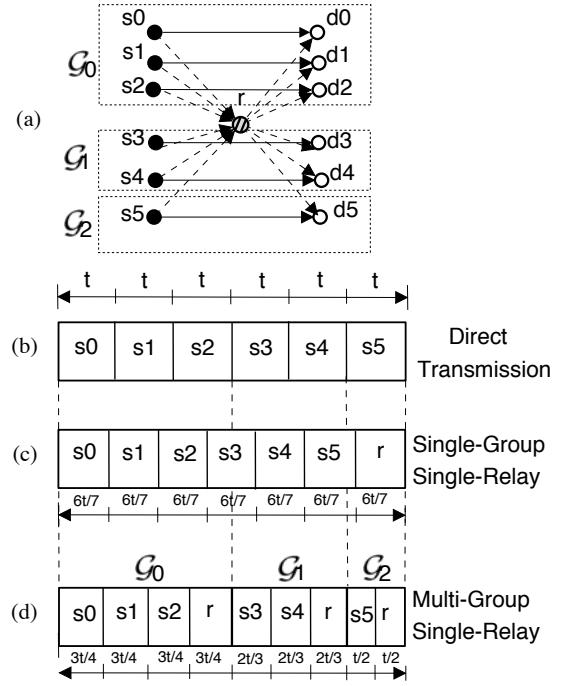


Fig. 3. An example illustrating the time slot structures for single-group single-relay case and multi-group single-relay case in NC-CC.

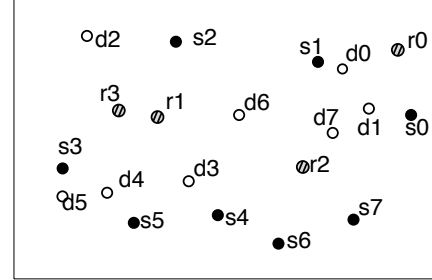


Fig. 4. A 20-node network.

available to them is $3t$. Under NC-CC, as one additional time slot is needed for the relay node, we divide $3t$ into 4 equal-sized time slots, and thus the length of each time slot is $3t/4$. Following the same token, each time slot for $\mathcal{G}_1 = \{s_3, s_4\}$ is $2t/3$. Finally, each time slot for $\mathcal{G}_2 = \{s_5\}$ is $t/2$.

A naive approach to structure time slots in Fig. 3(d) is to set equal time slot sizes across all groups, i.e., $6t/9$ (or $2t/3$) for each time slot. We argue that this is not a fair way to allocate time, as the total time allocated to $\mathcal{G}_0 = \{s_0, s_1, s_2\}$ will be $4 \cdot 2t/3 = 8t/3$, which is less than its fair share of $3t$ in Fig. 3(b). Likewise, s_5 will have a total time of $2 \cdot 2t/3 = 4t/3$, which is greater than its fair share of t .

We now show that this multi-grouping mechanism affects the achievable rate for a session (s_i, d_i) in a group. Denote $\mathcal{G}_r^{s_i}$ the group that contains s_i and uses relay node r for NC-CC. Under our approach, there are a total of $|\mathcal{G}_r^{s_i}|$ time slots (each of size t) for this group. Then the size of each time slot for this group under NC-CC will be $\frac{|\mathcal{G}_r^{s_i}| \cdot t}{|\mathcal{G}_r^{s_i}| + 1}$. The achievable

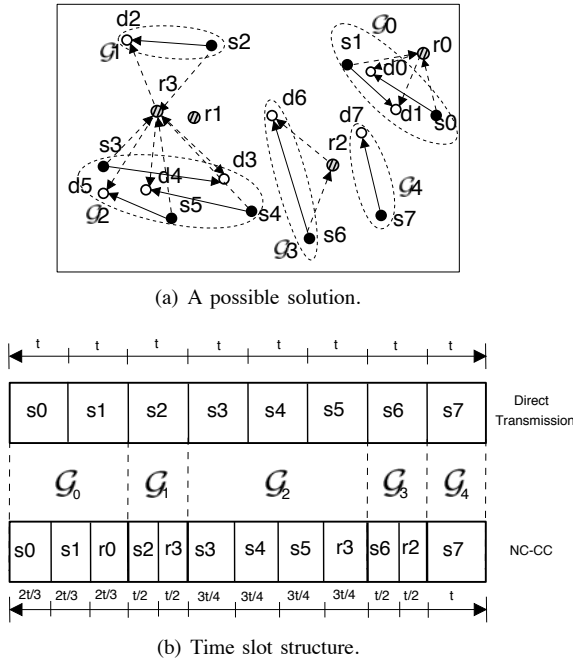


Fig. 5. An example showing grouping and relay node selection.

rate for session (s_i, d_i) in this group is

$$R_{\text{NC-CC}}(s_i, r, \mathcal{G}_r^{s_i}) = \left(\frac{|\mathcal{G}_r^{s_i}| \cdot t}{|\mathcal{G}_r^{s_i}| + 1} \right) \cdot \frac{W}{N_s} \cdot I_{\text{NC-CC}}(s_i, r, \mathcal{G}_r^{s_i})$$

$$= \frac{|\mathcal{G}_r^{s_i}|}{|\mathcal{G}_r^{s_i}| + 1} \cdot \frac{W}{N_s} \cdot I_{\text{NC-CC}}(s_i, r, \mathcal{G}_r^{s_i}). \quad (5)$$

Comparing (5) to (1), we find that when we use multiple groups, the effective session bandwidth $\left(\frac{|\mathcal{G}_r^{s_i}|}{|\mathcal{G}_r^{s_i}| + 1} \cdot \frac{W}{N_s} \right)$ will always be less than the effective session bandwidth in the single-group case $\left(\frac{W}{N_s + 1} \right)$, as $|\mathcal{G}_r^{s_i}| < N_s$. This reduction in effective session bandwidth can also be observed by comparing the time slot size in Figs. 3(c) and 3(d). On one hand, multiple groups can increase a session's mutual information. But on the other hand, multiple groups also reduce the effective bandwidth for a session. Therefore, we have a trade-off between effective bandwidth and mutual information of a session. Such a trade-off should be incorporated within our problem formulation.

The Multi-Group Multi-Relay Case. Our previous discussion of the multi-group single-relay case shows the significance of putting sessions into different groups, even when there is only a single relay node. In general, there may be multiple relay nodes available in the network. In this case, in addition to the grouping problem, we also need to address the relay node selection problem. That is, we have a joint problem of session grouping and relay node selection. This is the focus of this paper.

We use the network in Fig. 4 to illustrate our problem. In this network, we have eight sessions $\{(s_0, d_0), (s_1, d_1), \dots, (s_7, d_7)\}$ and four relay nodes $\{r_0, r_1, r_2, r_3\}$. All nodes are within the interference range of each other and therefore simultaneous transmissions by two or more sessions are not allowed. Figure 5(a) shows a possible grouping and relay

node selection. In this solution, there are five groups: $\mathcal{G}_0 = \{s_0, s_1\}$, $\mathcal{G}_1 = \{s_2\}$, $\mathcal{G}_2 = \{s_3, s_4, s_5\}$, $\mathcal{G}_3 = \{s_6\}$, and $\mathcal{G}_4 = \{s_7\}$. Group \mathcal{G}_0 is assigned to relay node r_0 . Groups \mathcal{G}_1 and \mathcal{G}_2 are both assigned to the same relay node r_3 . Group \mathcal{G}_3 , which contains only source s_6 , uses the relay node r_2 , and group \mathcal{G}_4 with source node s_7 does not use any relay node (i.e., it uses direct transmission). We can also see that relay node r_1 is not being used by any group. For the session grouping and relay node selection in Fig. 5(a), the time-slot structure in a frame is shown in the lower portion of Fig. 5(b). In Fig. 5(b), we also show the time-slot structure when only direct transmission is employed in the network. It is not hard to see that there are many other possible ways to do the grouping and relay node selection for this network.

In general, the achievable rate for session (s_i, d_i) in group $\mathcal{G}_{r_j}^{s_i}$ is

$$R_{\text{NC-CC}}(s_i, r_j, \mathcal{G}_{r_j}^{s_i}) = \frac{|\mathcal{G}_{r_j}^{s_i}|}{|\mathcal{G}_{r_j}^{s_i}| + 1} \cdot \frac{W}{N_s} I_{\text{NC-CC}}(s_i, r_j, \mathcal{G}_{r_j}^{s_i}), \quad (6)$$

which is similar to (5), with the only difference being that r (single relay) is now replaced by r_j (one of the relays).

2.2 Problem Statement

Denote $\mathcal{N}_s = \{s_0, s_1, \dots, s_{N_s-1}\}$ as the set of source nodes, $\mathcal{N}_d = \{d_0, d_1, \dots, d_{N_d-1}\}$ as the set of destination nodes, and $\mathcal{N}_r = \{r_0, r_1, \dots, r_{N_r-1}\}$ as the set of relay nodes, respectively. We assume $N_s = N_d$ and that all source and destination nodes are distinct.¹ Each source node is expected to transmit data to its destination node, either with or without the assistance of a relay node. Furthermore, a session (or a group of sessions) may use at most one relay node for NC-CC.

We now define our objective function. A number of objective functions can be used for this problem. In this paper, we choose the objective of maximizing the sum of weighted data rates of all sessions, where the weight w_i for session (s_i, d_i) is a pre-defined constant. We can write the weighted rates for session (s_i, d_i) under NC-CC and direct transmission as

$$R_{\text{NC-CC}}^w(s_i, r_j, \mathcal{G}_{r_j}^{s_i}) = w_i \cdot R_{\text{NC-CC}}(s_i, r_j, \mathcal{G}_{r_j}^{s_i}), \quad (7)$$

$$R_{\text{D}}^w(s_i, d_i) = w_i \cdot \frac{W}{N_s} \cdot \log_2(1 + \text{SNR}_{s_i d_i}). \quad (8)$$

Our session grouping and relay node selection problem can now be formally defined as follows: Determine how to put all sessions into different groups and assign a relay node to each group so that the sum of the weighted data rates for all sessions is maximized.

Note that a solution to the above optimization problem does not exclude a session from employing direct transmission (e.g., \mathcal{G}_4 in Fig. 5(b)). Further, in the special case when a group contains only one session, then only CC is employed for that session (e.g., \mathcal{G}_3 in Fig. 5(b)). In other words, both direct transmission and CC without NC are allowed in our solution. As a result, an optimal solution to this problem should be at

1. In the case where a node serves in multiple roles, we can partition this node logically into multiple nodes and visualize each as a separate node.

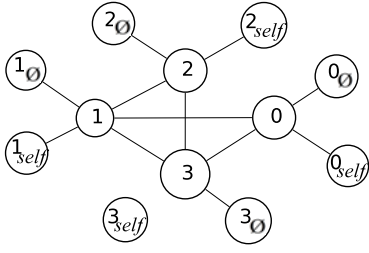


Fig. 6. A two-symmetric hypergraph for the network with four sessions and one relay node. The maximum number of sessions per group is limited to two.

least as good as a solution that only employs CC (without NC) or direct transmission.

3 PROOF OF NP-HARDNESS

We now prove that the joint grouping and relay node selection (GRS) problem in general is an NP-hard problem. The proof consists of a two step approach. First, a well known NP-hard problem called 3-dimensional matching is transformed into a maximum weight matching problem in a 3-symmetric hypergraph. Second, we show that an instance of the matching problem in a 3-symmetric hypergraph can be reduced to a special case of the GRS problem in polynomial time.

We first consider a scenario where the network has a set of n sessions $((s_i, d_i)$ pairs) indexed by $0, 1, \dots, n-1$. In addition, the network has only one relay r , and the maximum number of sessions in a group is limited to two, i.e., a single session (s_i, d_i) can: (a) use direct transmission, (b) use the relay node without sharing it with any other session, or (c) share the relay node with one other session. We can represent this scenario using a graph on the node set $\{0, \dots, n-1\} \cup \{0_{self}, \dots, (n-1)_{self}\} \cup \{0_0, \dots, (n-1)_0\}$, with edges

- (i, i_0) : indicating direct transmission, with weight of edge = $R_D^w(s_i, d_i)$, for all $i = 0, 1, \dots, n-1$.
- (i, i_{self}) : exists if and only if $R_{NC-CC}^w(s_i, r, \{s_i\}) > R_D^w(s_i, d_i)$, with the weight of the edge being $R_{NC-CC}^w(s_i, r, \{s_i\})$, for all $i = 0, 1, \dots, n-1$.
- (i, j) : exists if and only if $R_{NC-CC}^w(s_i, r, \{s_i, s_j\}) + R_{NC-CC}^w(s_j, r, \{s_i, s_j\}) > R_D^w(s_i, d_i) + R_D^w(s_j, d_j)$, with the weight of the edge being $R_{NC-CC}^w(s_i, r, \{s_i, s_j\}) + R_{NC-CC}^w(s_j, r, \{s_i, s_j\})$, for all $i, j = 0, 1, \dots, n-1, i \neq j$.

As an example, Fig. 6 shows a graph for a network with four sessions and one relay node. Note that some edges (e.g., $(0, 2)$) are missing from the graph. This is because they do not satisfy the existence condition described above (i.e., the sum of session rates on this edge is less than the sum of session rates under direct transmission). Then the GRS problem is to find a maximum weight matching in this graph, i.e., find a set of edges such that the sum of their weights is maximum while having each node incident to at most one selected edge. This

problem can be solved in polynomial time [20]. In general, we denote such a graph by G^2 . Also note that this graph is a two-symmetric hypergraph. The m -symmetric property of a hypergraph states that every edge in the graph will have cardinality of m , i.e., it will have exactly m incident nodes.

Next, we extend this problem to allow the maximum number of sessions in a group to be three. Now, the problem can be described as the maximum weight matching problem in a hypergraph with largest edge cardinality of three, where there exist hyperedges involving three distinct nodes from $\{0, 1, \dots, n-1\}$ at a time. We denote the hypergraph for this new restricted problem by G^3 . In G^3 , every new hyperedge that connects distinct nodes i, j , and k will have an associated weight given by the sum of $R_{NC-CC}^w(s_i, r, \{s_i, s_j, s_k\})$, $R_{NC-CC}^w(s_j, r, \{s_i, s_j, s_k\})$, and $R_{NC-CC}^w(s_k, r, \{s_i, s_j, s_k\})$; and such an edge will be constructed if and only if it is corresponding to a “candidate” session group, i.e., we have $R_{NC-CC}^w(s_i, r, \{s_i, s_j, s_k\}) + R_{NC-CC}^w(s_j, r, \{s_i, s_j, s_k\}) + R_{NC-CC}^w(s_k, r, \{s_i, s_j, s_k\}) > R_D^w(s_i, d_i) + R_D^w(s_j, d_j) + R_D^w(s_k, d_k)$. We define a “candidate” sessions group for a relay node as a session group that has the sum of weighted data rates of individual sessions under NC-CC as larger than the sum of the weighted data rates of individual sessions under direct transmission. This means that the session group is a “candidate” to be included in the final solution. Note that in addition to these new hyperedges of cardinality three, the previously defined edges of G^2 with cardinality of two will remain in this new hypergraph, i.e., $G^2 \subseteq G^3$. Thus, to solve this new restricted subproblem, we need to find a maximum weight matching in the hypergraph G^3 . We have the following theorem.

Theorem 1: The GRS problem with a single relay node, in which the groups of sessions are restricted to include at most three sessions, and where the session rates can take on arbitrary nonnegative values, is NP-hard.

Proof We first consider a well-known NP-hard problem called 3-dimensional matching (3DM) [8]. Given a graph on n nodes (with $n/3$ integral), where the nodes are partitioned into three disjoint subsets each having $n/3$ nodes, and where each defined hyperedge has cardinality of three and involves a single node from each of the three-node subsets, the 3DM seeks to find if there exists a perfect matching (i.e., one involving all the n nodes). The 3DM problem can be directly transformed into a maximum weight matching problem on a 3-symmetric hypergraph. This can be done by letting the weight assigned to each defined hyperedge be equal to one, and then seeking if the maximum weight matching equals $n/3$. Thus, the maximum weight matching problem in a 3-symmetric hypergraph is also an NP-hard problem. Furthermore, an instance of the latter problem can be polynomially reduced to the stated GRS problem by including the missing edges of cardinality other than three, and assigning zero weights to these edges. If we assume that the weighted session rates in the network can have arbitrary nonnegative values, this newly constructed hypergraph can be transformed back into a network instance of the GRS problem. This can be achieved by assigning weighted rate values of $H_i/3$ to each session involved in the hyperedge for that particular group, where H_i

is the weight of the i -th hyperedge. As a result, the restricted GRS problem with a single relay node, in which the groups of sessions are restricted to a maximum size of three, is NP-hard. ■

We can further extend the problem and allow the groups to include at most four sessions. The hypergraph for this problem can be constructed by extending G^3 , and adding hyperedges of cardinality four corresponding to the candidate session groups with four sessions. Continuing in this fashion, we can construct a general problem in which the maximum number of sessions in the group are allowed to be at most n , i.e., there is no limit on the size of a group. This general problem can be described as the maximum weight matching problem in a hypergraph with maximum edge cardinality of n . We denote the hypergraph for this problem as G^n , where $G^n \supseteq G^{n-1} \supseteq \dots \supseteq G^3 \supseteq G^2$, and where G^n contains candidate hyperedges of cardinality $n, n-1, \dots, 2$. We now have the following corollary for this general case.

Corollary 1.1: The generalized-weight GRS problem with a single relay node is NP-hard.

Proof The proof follows that of Theorem 1. An instance of a maximum weight matching problem in a 3-symmetric hypergraph can be reduced to the stated GRS problem following a similar approach as in Theorem 1, i.e., by including the missing edges of cardinality other than three, and assigning zero weights to these edges and then computing the maximum total weighted rates. As a result, we have that the restricted GRS problem with a single relay node, in which the groups of sessions are restricted to a maximum size of n , is NP-hard. ■

Since the GRS problem with one relay node ($N_r = 1$) is a special instance of the general GRS problem (with N_r relay nodes), we have the following corollary.

Corollary 1.2: The generalized-weight GRS problem with multiple relay nodes is NP-hard.

Note that our NP-hardness proof pertains to the generalized-weight GRS problem where the weighted rates of groups of sessions can take on arbitrary nonnegative values. The question whether the GRS problem with hyperedge weights given by the specific formulae (7) and (8) remains NP-hard is left open for future research. However, we conjecture this to be the case since, as in the proof of Theorem 1 and Corollary 1.1, even a simple instance of GRS with a single relay and with non-negative weights on sessions of cardinality three is shown to be NP-hard.

4 A DISTRIBUTED ALGORITHM

In this section, we present D-GRS, a distributed and online algorithm for the GRS problem that produces near-optimal results. By “online”, we mean that network dynamics are unknown *a priori*. That is, sessions can join and leave the network as time progresses. Further, we allow a relay node to be active (“on”) and inactive (“off”) over time. The goal of D-GRS is to accomplish session grouping and relay node selection via local computation and distributed message exchange among the nodes so as to maximize the sum of the weighted rates of all sessions. In our distributed algorithm, we separate

the control plane used for executing the D-GRS algorithm from the data plane used for data transport by the sessions. That is, we assume the execution of the D-GRS algorithm is done on a separate control channel, which is independent from the data frame carrying sessions’ data.

In Section 4.1, we describe the information that needs to be maintained at each source and relay node. In Section 4.2, we give a description of the three core subroutines of D-GRS. Section 4.3 presents how D-GRS handles session arrivals and departures. In Section 4.4, we show how D-GRS works in a setting where the relay nodes are also allowed to switch between active and inactive status over time. In Section 4.5, we discuss the stability of D-GRS, and in Section 4.6, we analyze D-GRS’ complexity.

4.1 Information Maintained at Nodes

We first describe the information that needs to be maintained at each node.

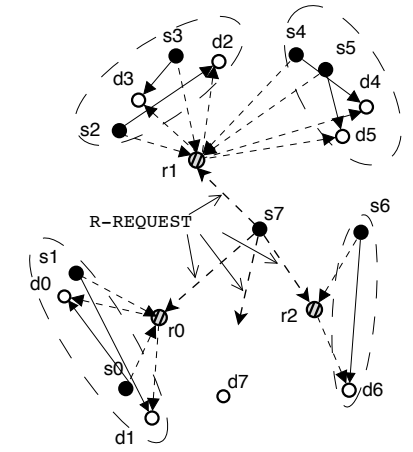
Source node. Each source node s_i in the network maintains the following information:

- Channel state information (CSI) (i) between s_i and its destination node d_i^2 , and (ii) between the source nodes of other sessions and d_i . Information in (ii) can be obtained by having d_i hear the other source nodes’ transmissions over a time frame. Then d_i can inform its source node about this information. A source node needs this information so that it can forward this information to the relay node. The relay node in turn will use this information to calculate the data rates that it will include in its offers.
- The number of active sessions in the network, and the number of sessions in its current session group. This information is needed by the source to determine time slot structure. To acquire this information, the source node of a new session sends a broadcast request (REQ-`ACT-SESSIONS`). Upon hearing this request, one of the relay nodes will reply with this information (RAS-`REPLY`). Note that only one relay needs to reply. This can be achieved by setting a random timer at every relay, and have those relay nodes refrain their response once they hear that some other relay node has already responded. For the active source nodes in the network, they also update the information regarding the number of active sessions in the network when they hear the REQ-`ACT-SESSIONS` message.

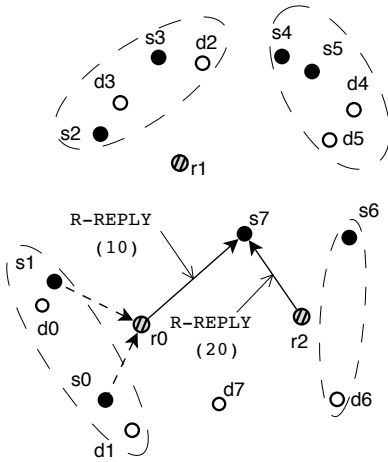
Relay node. Every relay node in the network maintains the following information:

- (i) The CSI between the source nodes using this relay and this relay node, (ii) the CSI between the relay node and the destination nodes of the above source nodes, and (iii) the CSI maintained at the above source nodes. The information in (i) can be obtained when a new session initiates and starts direct transmission. The information in (ii) can be obtained from the destination nodes. Similarly,

2. A number of mechanisms can be employed to obtain this information. Discussion of these mechanisms is beyond the scope of this paper.



(a) Source node s_7 broadcasting the R-REQUEST message.



(b) Relay nodes r_2 and r_{10} sending back R-REPLY messages with OFFERS.

Fig. 7. An example illustrating core SOS operations.

the relay node can obtain the information in (iii) from source nodes.

- The number of active sessions in the network, and the number of sessions in its local groups. This information is updated whenever a new session is initiated or an existing session terminates, and after a session selects a group.

4.2 Core Subroutines

The core subroutines in D-GRS can be summarized as “SEEK-OFFER-SELECT” (or SOS), which we describe as follows:

SEEK. Initiated by a source node, which sends a broadcast request in the hope of finding a new/better relay node. The source node seeks offers from the relay nodes by broadcasting a message for relay node selection (R-REQUEST). The transmitted request includes the CSI information maintained at the source node. This information is required by the neighboring relay nodes to determine beneficial assignments. The message also includes the weight associated with this session, and its current data transmission rate. The source node will then wait

to hear any offer from the relay nodes. This is the “SEEK” step of SOS.

OFFER. This is performed by the relay nodes. Each relay node makes an offer to the requesting source node regarding grouping based on its local computation.

When a relay node r_j receives the R-REQUEST from some source s_i , it will compose an offer for s_i . Initially, the relay node r_j uses the CSI information that it maintains to determine the data rate it can offer to session (s_i, d_i) . The relay node r_j is aware of the groups of sessions it is currently supporting, and can determine local “candidate” groups for the new session (s_i, d_i) . A candidate group for a relay node is defined as the one in which the weighted sum of achievable rates of all sessions is not less than that under direct transmission. Note that one candidate group could be an empty group (i.e., if the new session joins this group, only CC will be used). Now, the relay node determines the local group for (s_i, d_i) that has the potential to maximize the objective function. That is, the relay now considers the new group $\hat{\mathcal{G}}_k = \mathcal{G}_k + s_i$. Denote \hat{U}_k as the sum of weighted session rates in $\hat{\mathcal{G}}_k$. Denote $w_i R_i$ as the current weighted rate of s_i , which is available in the R-REQUEST message. Denote U_k as the sum of weighted session rates in a local group \mathcal{G}_k that is supported by r_j . Then by comparing $\hat{U}_k - U_k - w_i R_i$ among all local groups \mathcal{G}_k at this relay node, the relay can identify the group that offers the largest gain, which we call LOCAL_GAIN.

This calculated LOCAL_GAIN is then included in the relay node’s offer (R-REPLY) to the source s_i . In the case that the relay node cannot find any candidate group for (s_i, d_i) , or if the LOCAL_GAIN is negative, then the relay node does not reply to the requesting source node. This completes the “OFFER” step of SOS.

SELECT. The source node selects the best offer among the relay nodes. To accomplish this, the source node waits for a prescribed time after transmitting the R-REQUEST message. Among all the offers that it receives, the source node of the session selects the relay node that offers the largest LOCAL_GAIN. The source node will then transmit a message (CONFIRMATION) informing the relay node of its selection. Upon receiving this confirmation message, the time slot structure will be updated accordingly. This completes the “SELECT” operation of SOS.

As an example, Fig. 7 shows the core SOS operations. Source node s_7 broadcasts an R-REQUEST message (SEEK). In reply (see Fig 7(b)), the relay nodes r_0 and r_2 offer LOCAL_GAINs of 10 and 20, respectively (OFFER); relay node r_1 does not reply because it does not find a candidate group for session (s_7, d_7) . Finally, s_7 selects relay r_2 with the largest LOCAL_GAIN (SELECT).

In the rest of this section, we will show how D-GRS uses these core subroutines during different events.

4.3 Session Initiation or Termination

We first consider a network scenario where a new session initiates or an existing session terminates in the network. Here, the set of relay nodes are assumed to remain active in the

<p>Search-Beneficial-Relay() //used by source s_i seeking relay node</p> <ol style="list-style-type: none"> 1. If the session is newly initiated, then //collect information 2. Broadcast REQUEST-ACTIVE-SESSIONS. 3. Receive the number of active sessions in the RAS-REPLY. 4. Receive the CSI information from the destination node d_i. 5. Start direct transmission. 6. Compose and broadcast the R-REQUEST message. 7. Set a reply-timer. 8. Store all the R-REPLY messages received before the reply-timer expires. <p>// Every R-REPLY(j) contains the relay node that transmitted it, and // the session to which it was transmitted.</p> <ol style="list-style-type: none"> 9. For any two received R-REPLY(j) and R-REPLY(k), $j \neq k$ 10. If R-REPLY(j).relay == R-REPLY(k).relay, then 11. If R-REPLY(j).source \neq R-REPLY(k).source 12. $r = \text{R-REPLY}(j).\text{relay}$ 13. Remove all the R-REPLYS transmitted by r from storage. <p>// Every received R-REPLY(j) also contains a LOCAL_GAIN and the // group information</p> <ol style="list-style-type: none"> 14. If the largest received LOCAL_GAIN > 0, then 15. Send a CONFIRMATION message to the relay r_{\max} that offered largest LOCAL_GAIN. 16. Adjust the time slot structure, and start using r_{\max} as the cooperative relay. <ol style="list-style-type: none"> 17. Return ;
<p>Permission-Message-Received()</p> <p>// Used by s_i, if it hears a permission message.</p> <ol style="list-style-type: none"> 1. If the permission is from $\mathcal{R}(s_i)$, 2. If the permission is for s_i 3. Search-Beneficial-Relay() 4. else If the permission is for another source in the same local group as s_i 5. Cannot leave the current relay (because someone else now has the permission).
<p>Collect-CSI(d_i) // Function used by destination node d_i.</p> <ol style="list-style-type: none"> 1. Upon hearing an REQUEST-ACTIVE-SESSIONS transmitted by s_i, do the following 2. Collect CSI information between all the source nodes and itself during the next time frame. 3. At the end of the frame, transmit the collected CSI back to s_i.

Fig. 8. Pseudo code for a session (s_i, d_i) .

network. The case of a relay node's on/off behavior will be discussed in Section 4.4.

A new session initiates. When a session (s_i, d_i) initiates, the source node s_i broadcasts a message requesting the number of active sessions in the network (REQ-ACT-SESSIONS). This request serves two purposes: (i) the relay nodes and other source nodes in the network will know about this new session, and can adjust their time slots appropriately to accommodate the new session, and (ii) the source s_i will get a response (RAS-REPLY) from one of the relay nodes, and will start direct transmission in its time slot based on the new frame structure. Meanwhile, upon hearing the REQ-ACT-SESSIONS, the corresponding destination node d_i starts to collect the CSI between the other active source nodes and itself. This CSI information is necessary so that the source node can begin the SEEK operation. After one time-frame, the destination node of this new session will transfer the collected CSI information back to the new source node. The source node of this new session will now broadcast a request message for relay nodes (R-REQUEST) (i.e., SEEK). Upon receiving this message, each relay node will find a best local group for this session to maximize the objective. Then the relay node will reply to the new source node with this information (i.e., OFFER). Upon receiving the replies from

<p>Reply-to-R-REQUEST(s_i, C_{s_i})</p> <p>// Function called when relay node r_j receives the request message // from s_i. C_{s_i} is the current weighted rate of session s_i.</p> <ol style="list-style-type: none"> 1. For all the session-groups \mathcal{G}_m currently being assisted by r_j 2. Initialize LOCAL_GAIN$_m = 0$. 3. $\mathcal{G} = \mathcal{G}_m$. 4. Add s_i to group \mathcal{G}_m, i.e., $\mathcal{G}_m = \mathcal{G}_m + s_i$. 5. Set feasible = Check-Feasibility(\mathcal{G}_m). 6. If (feasible == 1) 7. $\text{sum}_m = \sum_{s_k \in \mathcal{G}_m} R_{A-NCC}^w(s_k, r_j, \mathcal{G}_m)$. 8. Remove s_i from group \mathcal{G}_m, i.e., $\mathcal{G}_m = \mathcal{G}_m - s_i$. 9. $\overline{\text{sum}}_m = C_{s_i} + \sum_{s_k \in \mathcal{G}_m} R_{A-NCC}^w(s_k, r_j, \mathcal{G}_m)$. 10. LOCAL_GAIN$_m = \text{sum}_m - \overline{\text{sum}}_m$. 11. $\mathcal{G}_m = \mathcal{G}$. 12. Set value as the largest of all LOCAL_GAIN$_m$'s. 13. If (value > 0) 14. Compose an R-REPLY message. 15. Include value and the id for the selected \mathcal{G}_m in the composed message. 16. Transmit the composed R-REPLY to source s_i. 17. Start a timer, and wait until timer expires. <p>// This timer is to give some time to s_i to send back CONFIRMATION. // If timer expires, and confirmation is not received, then r_j will // assume that s_i is not interested in joining at this time. // Relay r_j is now free to respond to other R-REQUESTS.</p>
<p>Receive-ACTIVE-SESSIONS-REQUEST(s_i)</p> <p>// Called by r_j, when it hears a request for number of active // sessions from some new source/relay.</p> <ol style="list-style-type: none"> 1. Start a timer with random time period. 2. Upon timer expiration, 3. If no other relay has transmitted the number of active sessions, then transmit the number of active sessions. 4. If the received message was from a new source node 5. Return; 6. For all source nodes s_j in local groups, set flag(s_j) = 1. 7. Set count = 0. 8. For all local groups \mathcal{G}_k, 9. For some ($s_j \in \mathcal{G}_k : \text{flag}(s_j) == 0$) 10. Set flag($s_j$) = 1 11. count++ 12. If (Check-Feasibility($\mathcal{G}_k - s_j$) == 1) 13. Transmit PERMISSION-to-LEAVE to s_j. 14. Start a timer. 15. Before the timer expiration, if some permitted sessions have left, then 16. Adjust the time slots for the affected groups. 17. If timer expires and count > 0 18. Goto step 6.
<p>Receive-Leaving(s_i)</p> <p>// Called by r_j, when source s_i leaves the network and transmits // LEAVING. All the sessions will adjust their time-slots upon // hearing this message.</p> <ol style="list-style-type: none"> 1. Identify the session-group $\mathcal{G} \ni s_i$ assisted by relay r_j. 2. Remove s_i from group \mathcal{G}, i.e., $\mathcal{G} = \mathcal{G} - s_i$. 3. Set flag = Check-Feasibility(\mathcal{G}). 4. If (flag == 1) then Return ; 5. For all $s_k \in \mathcal{G}$ 6. $\text{diff}_k = R_{A-NCC}^w(s_k, r_j, \mathcal{G}) - R_D^w(s_k, d_k)$. 7. Send a REMOVE-SRC message to the s_k with smallest diff_k. 8. $\mathcal{G} = \mathcal{G} - s_k$. 9. If ($\mathcal{G} \neq \emptyset$) 10. Goto step 3. 11. Return ;
<p>Check-Feasibility(\mathcal{G})</p> <ol style="list-style-type: none"> 1. For all $s_k \in \mathcal{G}$ 2. If $C_{A-NCC}(s_k, r_j, \mathcal{G}) < C_D(s_k, d_k)$ 3. Return (0); 4. Return (1);

Fig. 9. Pseudo code for a relay node r_j .

all the relay nodes (or after the timer expiration), the source node selects the relay node with the best offer and sends a confirmation message (CONFIRMATION) to the selected relay

node (i.e., SELECT). Subsequently, the new session joins the group in the chosen relay.

A session departs. When a session (s_i, d_i) departs, the source node of the session broadcasts a message (LEAVING) indicating its new status. Other sessions will update their knowledge about the number of sessions in the network, and can adjust their time slot structure accordingly.

Due to the session's departure, additional adjustments in the group of the departing session may be necessary to ensure that the remaining group remains a candidate group. The corresponding relay node r_j again calculates the new data rates for the remaining sessions in the group. If the weighted sum of new data rates of all the remaining sessions is above their weighted direct transmission rates, then no other operation is performed. Otherwise, the remaining group is considered inferior (i.e., it cannot be a candidate group). The relay node will now offload some sessions from the group (starting from the session with the largest rate drop below its direct transmission rate). To offload a session (s_k, d_k) from this group, the relay node will send a message (REMOVE-SRC) announcing this removal to the source node s_k . Upon receiving this message, source node s_k falls back to direct transmission. The relay node will repeat this process for the remaining sessions in this group until the group becomes either a candidate for the final solution or empty.

The sessions that are being offloaded in the above process need to wait for a random amount of time before performing the core SOS operations to seek other relays. Note that this time, the SEEK operation (i.e., the R-REQUEST message) should contain a flag indicating that this request is from an ongoing session, instead of a new one. This flag is required to indicate that no change is needed in the time slot structure, which is different from the case of a new session.

Figure 8 shows the pseudo code of the procedures used by the source and destination nodes in the network.

4.4 Relay Activation and Deactivation

We now consider the scenario where the relay nodes can also become active and inactive as time progresses.

A relay node becomes active. Upon activation, the relay node broadcasts a message (REQ-ACT-SESSIONS). The purpose of this message is twofold: (i) to inform other sessions in the network regarding its activation, and (ii) to request information regarding the number of active sessions in the network. The information in (ii) is required to construct an OFFER in response to some SEEK request. Upon receiving this request, one of the active relay nodes will reply with the latter information (RAS-REPLY). Note that only one reply is needed, as other relay nodes can hear the first reply and then refrain from transmitting the same information again.

The activation message transmitted by a newly active relay node is heard by existing source nodes. Upon hearing such a message, each source will try to check if the newly active relay can improve its transmission rate. This check is performed by each source after waiting for a random amount of time. However, before a source tries to seek a new relay and change its session group, the source needs to make sure that after it

leaves its current session group, the remaining session group remains a candidate. That is, the weighted sum of the session rates in the remaining session group does not fall below the weighted sum of session rates under direct transmission. Such candidacy can be determined by the current relay node that the source is using. As a result, the source node will ask for permission from the current relay node to seek a new relay node. This relay node may or may not grant a permission. Note that permissions will be granted in sequence by a relay node to one source node at a time. If a permission is granted, then the source node can start the SEEK operation. After a source node selects some relay node, it will transmit a confirmation message (CONFIRMATION) to its new relay node. This confirmation message will be used to re-adjust the existing time slot structure.

A relay node becomes inactive. If a relay node r_j decides to become inactive, it broadcasts a message (R-LEAVING). The source nodes that are using r_j will adjust their time slots and fall back to direct transmission. Subsequently, these source nodes will wait for a random amount of time and then perform the three core SOS steps.

Figure 9 presents the pseudo code of the procedures used by the relay nodes in the network.

4.5 Stability

We will now discuss the stability of our D-GRS algorithm. We show that the D-GRS algorithm is inherently stable under various situations.

A session departs. When a session departs, the relay node may decide to offload some of the remaining sessions one by one from the group to which the departing session belongs. While the offloading is in progress, some other session in the network may broadcast an R-REQUEST message (the SEEK operation). Now, the question is what will the relay node (that is currently offloading the sessions) do? We propose that in this scenario, the relay node will not construct a new OFFER for this session, and will continue finishing its offloading process.

Multiple sessions become active. When multiple sessions become active at the same time, there will be multiple SEEK operations initiated in the network. Here, we exploit the fact that the R-REQUEST messages will be transmitted sequentially (instead of simultaneously). All the relay nodes will construct the OFFERs sequentially. That is, the R-REPLY message for the first session will be constructed first. Only after the first session is done (e.g., it has accepted an offer), the relay nodes will construct an OFFER in response to the second R-REQUEST message. Further, the R-REPLY message will contain the identity of the session for which this message is constructed.

4.6 Overhead Analysis

Since the D-GRS algorithm is activated by various events in the network, we will analyze the number of messages exchanged (i.e., the overhead) associated with each event in the network.

A session (s_i, d_i) initiates The source node of the session broadcasts an R-REQUEST message, and can get at most N_r R-REPLY messages in reply. After receiving the R-REQUEST message, each relay node may further request CSI values from the source nodes it is currently supporting (see the information maintained at a relay node in Section 4.1). As a session uses only one relay node, the total message exchanges in the network in this event cannot be more than $O(N_r + N_s)$.

A session (s_i, d_i) terminates. The source node of the departing session broadcasts a LEAVING message. This LEAVING message may result in at most N_s REMOVE-SRC messages (to offload the other sessions in the group). When a session is offloaded, it may want to seek other relay nodes. In this case, one R-REQUEST message from every offloaded source node will be transmitted. We know that a single R-REQUEST message can result in at most $O(N_r + N_s)$ messages. Thus, the total messages exchanged in the network due to a session termination cannot exceed $O(N_s \cdot (N_r + N_s))$.

A relay node becomes active. The relay node will broadcast a message requesting the number of active sessions in the network. This will result in a single reply from one of the existing relay nodes. Next, active relay nodes may transmit permission messages to their source nodes. This can result in at most N_s permission messages in the network. Every permission message will allow a source node to search for another relay node. The search for a relay node requires at most $O(N_r + N_s)$ messages as explained earlier. Thus, the total message exchanges in this case cannot exceed $O(N_s \cdot (N_r + N_s))$.

A relay node becomes inactive. The relay node broadcasts a single message indicating its deactivation. This will result in every source node using this relay node to seek other relay nodes. There are at most N_s source nodes in the network, and the search for another relay node requires at most $O(N_r + N_s)$ messages as explained earlier. This will result in at most $O(N_s \cdot (N_r + N_s))$ messages in the network.

5 SIMULATION RESULTS

In this section, we present simulation results to demonstrate the performance and complexity of the proposed D-GRS algorithm. As a benchmark, we also formulate the joint session grouping and relay node selection as an integer linear program, and use a centralized optimization solver, CPLEX [7], to solve it. We will compare results from D-GRS with the optimal results from CPLEX. As expected, the run-time of CPLEX is exponential in the worst case due to NP-hardness and the discrete nature of the formulated problem.

5.1 Mathematical Formulation

The joint session grouping and relay node selection problem as defined in Section 2.2 can be formulated in different ways. We show one formulation here.

Denote \mathcal{I}_{r_j} as the set of all candidate groups of source nodes that may use relay node r_j for NC-CC. Recall that a *candidate* group (with respect to a relay node) is defined as a group of source nodes sharing this relay node that have the sum of

their weighted rates under NC-CC not less than the sum of their weighted rates under direct transmission. The size of the set \mathcal{I}_{r_j} can be exponential in the worst case. We further denote $\mathcal{J}_{r_j}^{s_i} \subset \mathcal{I}_{r_j}$ as the set of feasible groups that contain the source node s_i . Thus, we have $\bigcup_{s_i \in \mathcal{N}_s} \mathcal{J}_{r_j}^{s_i} = \mathcal{I}_{r_j}$.

Denote \mathcal{G} as a group of source nodes. We define a binary variable $X_{r_j}^{\mathcal{G}}$ as follows:

$$X_{r_j}^{\mathcal{G}} = \begin{cases} 1 & \text{if group } \mathcal{G} \text{ uses relay } r_j \text{ for NC-CC, } \mathcal{G} \in \mathcal{I}_{r_j}, \\ 0 & \text{otherwise.} \end{cases}$$

Since a source node s_j can be in at most one group (which itself can use at most one relay), we have

$$\sum_{r_j \in \mathcal{N}_r} \sum_{\mathcal{G} \in \mathcal{J}_{r_j}^{s_i}} X_{r_j}^{\mathcal{G}} \leq 1 \quad (s_i \in \mathcal{N}_s). \quad (9)$$

Note that by (9), a source node can belong to only one group, and there are N_s source nodes. Thus, the maximum number of groups that can be formed is N_s .

For the objective of maximizing the weighted sum of achievable rates among all source nodes in the network, we can formulate the joint grouping and relay node selection problem as follows:

$$\begin{aligned} \text{Max} \quad & \sum_{s_i \in \mathcal{N}_s} \left[\left(\sum_{r_j \in \mathcal{N}_r} \sum_{\mathcal{G} \in \mathcal{J}_{r_j}^{s_i}} X_{r_j}^{\mathcal{G}} R_{r_j}^w \text{NC-CC}(s_i, r_j, \mathcal{G}) \right) + \right. \\ & \left. \left(1 - \sum_{r_j \in \mathcal{N}_r} \sum_{\mathcal{G} \in \mathcal{J}_{r_j}^{s_i}} X_{r_j}^{\mathcal{G}} \right) R_D^w(s_i, d_i) \right] \\ \text{subject to} \quad & \sum_{r_j \in \mathcal{N}_r} \sum_{\mathcal{G} \in \mathcal{J}_{r_j}^{s_i}} X_{r_j}^{\mathcal{G}} \leq 1 \quad (s_i \in \mathcal{N}_s), \\ & X_{r_i}^{\mathcal{G}} \in \{0, 1\}, \quad (r_i \in \mathcal{N}_r, \mathcal{G} \in \mathcal{I}_{r_i}). \end{aligned}$$

Note that the objective function contains the sum of two different terms for every source node s_i in the network. The first term is the achievable rate under NC-CC while the second term contains the achievable rate under direct transmission; only one of these two terms will be non-zero and taken into the summation over $s_i \in \mathcal{N}_s$.

The above optimization problem is a 0-1 integer linear programming (ILP) problem, with an exponential number of variables ($X_{r_j}^{\mathcal{G}}$) in the worst case. Due to the combinatorial nature of the problem, any alternative formulation will also involve integer (or binary) variables in it. This kind of formulation tends to have a relatively tight linear programming relaxation, but needs to be solved using a suitable column generation algorithm (e.g. see [4], [9]). Any such solution procedure, or a direct solution of a 0-1 ILP using a commercial solver (e.g. CPLEX) will have the following disadvantages:

- 1) It will require an exponentially long run-time (due to the NP-hard nature of the problem in general), which will become impractical in solving moderate to large sized problem instances.
- 2) The solution procedure will be centralized (e.g., using a branch-and-bound or column-generation procedure), which will require that each node has global network

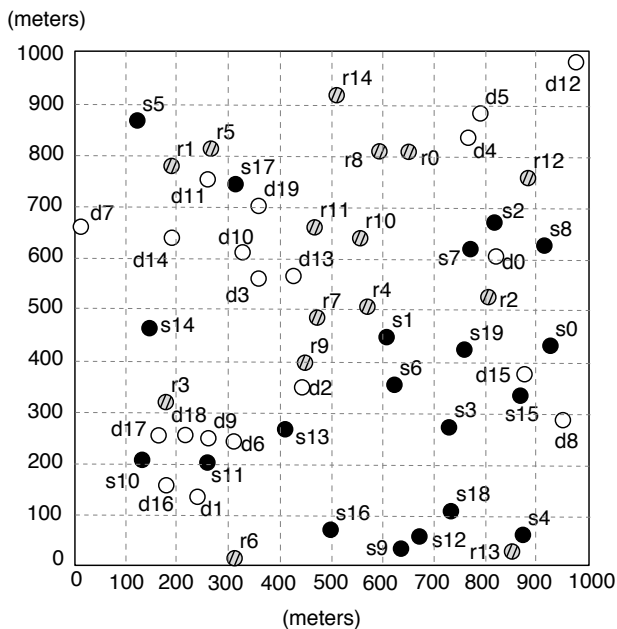


Fig. 10. A 55-node network.

knowledge. As a result, if the network topology changes often, the optimization scheme will quickly become impractical.

5.2 Simulation Settings

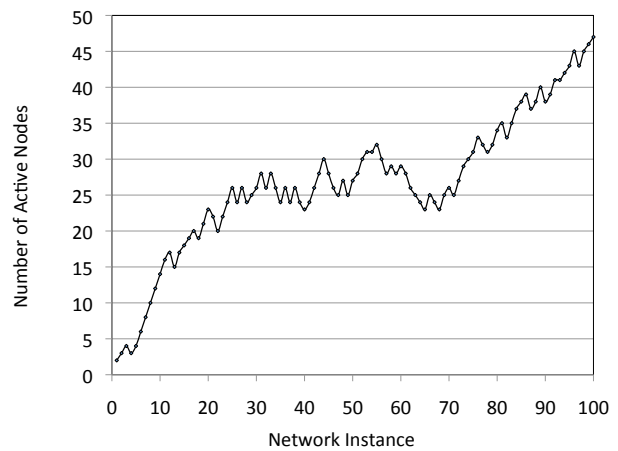
For all network instances in the simulation, we assume the transmission power at each node is 1 W and that the channel bandwidth is $W = 22$ MHz [11]. We assume the channel gain between two nodes s and d is $\|s - d\|^{-4}$, where $\|s - d\|$ is the distance (in meters) between s and d and 4 is the path loss index. We assume that the white Gaussian noise at all nodes has a variance of 10^{-10} W. We consider two scenarios. One, in which the weight for each session is assumed to be 1. The second, in which the weight for each session is assigned a random value between 0 and 1.

5.3 Results for Online Dynamics

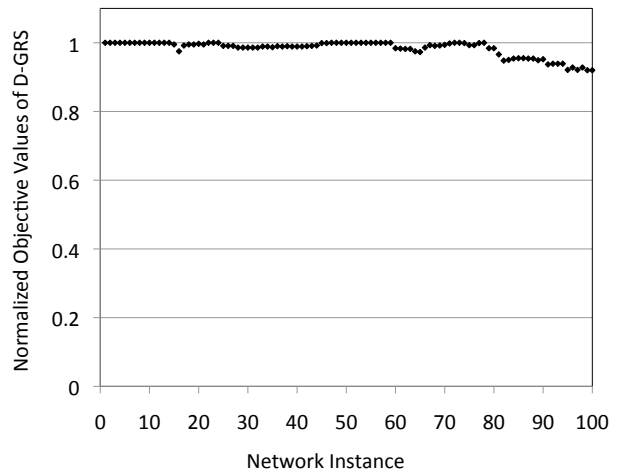
In this set of results, we consider a 55-node network consisting of 20 sessions and 15 relay nodes. The location of each node in the network is shown in Fig. 10. We show results from this 55-node network when nodes join and leave the network at random.

5.3.1 Constant Weights

All the sessions were assigned a constant weight value of 1. Initially, all sessions and relay nodes are assumed to be inactive. Then we allow new sessions to initiate and ongoing sessions to terminate, as well as permit the activation/deactivation of relay nodes. The sequence of these online dynamics is chosen to be random. Figure 11(a) shows the number of active nodes (including both source/destination nodes and relay nodes) in the network for each of the 100 events.



(a) Number of active nodes in the network.



(b) Ratio of objective values from D-GRS and CPLEX.

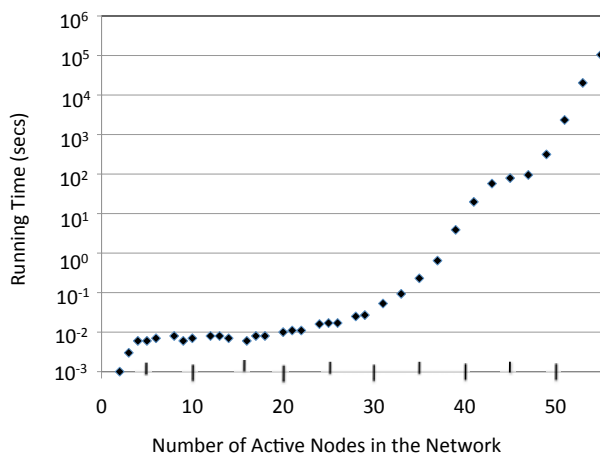
Fig. 11. Results showing near-optimality of D-GRS under online dynamics (constant weights).

The D-GRS runs continuously over the 100 events. Under each event, we compare the results from D-GRS and those from CPLEX. Figure 11(b) shows the normalized objective values of D-GRS (over those from CPLEX) under each event. We find that the performance of D-GRS is highly competitive (98.3% of optimal on average).

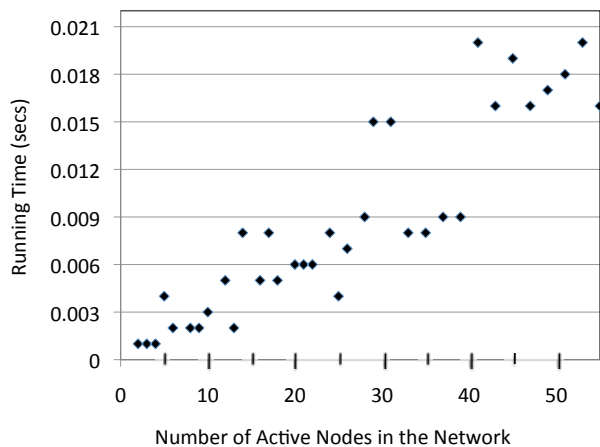
We now compare the complexity (in terms of running time) of D-GRS and CPLEX. Figure 12(a) shows the time required to obtain the optimal solutions as the number of nodes in the network increases. Note that the y-axis in Fig. 12(a) is in log-scale, indicating the exponential running time of CPLEX. On the other hand, Fig. 12(b) shows the running time (in linear scale) of the proposed D-GRS algorithm, which is orders of magnitude smaller than that under CPLEX.

5.3.2 Random Weights

Each session was assigned a random weight value between 0 and 1. Similar to the previous simulation, all sessions and relay nodes are initially assumed to be inactive. Then we allow new sessions to initiate and ongoing sessions to terminate, as well as permit the activation/deactivation of relay nodes. The sequence of these online dynamics is chosen to be random. Figure 13(a) shows the number of active nodes (including both



(a) Solutions from CPLEX (exponential growth).



(b) Solutions from D-GRS.

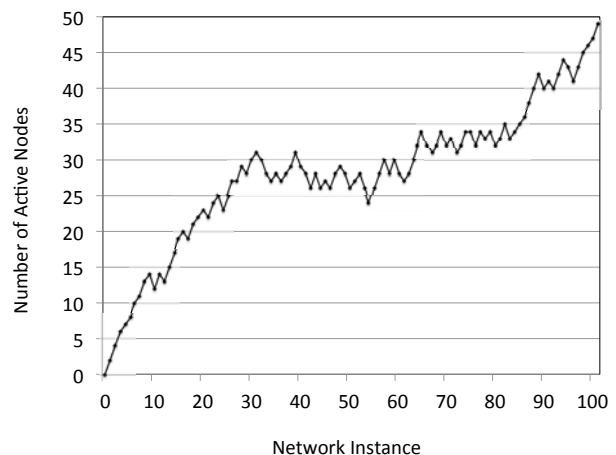
Fig. 12. Time to obtain solutions.

source/destination nodes and relay nodes) in the network for each of the 100 events. The D-GRS runs continuously over the 100 events. Figure 13(b) shows the normalized objective values of D-GRS (over those from CPLEX) under each event. We find that the performance of D-GRS is highly competitive (91.9% of optimal on average).

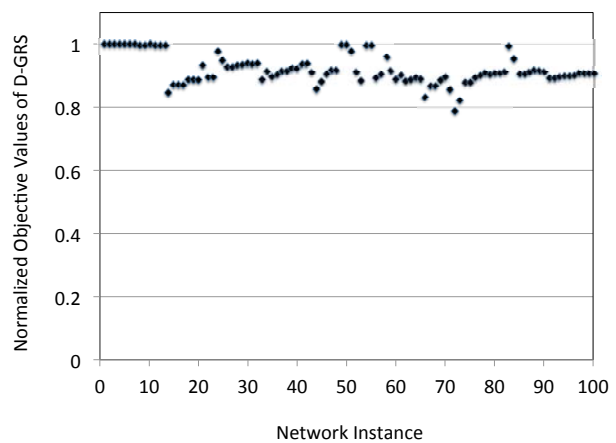
6 RELATED WORK

We review related work on CC and NC separately, followed by related work on NC-CC.

(a) CC. The concept of CC can be traced back to the introduction of a three-terminal communication channel (or a relay channel) by van der Meulen [24]. Subsequently, Cover and El Gamal [6] developed a lower bound on the capacity of a general relay channel. Recent research on CC at the physical layer was motivated by these early results, and led to a number of CC protocols at the physical layer (e.g., [1], [15], [16]). These physical layer protocols proposed different ways in which distributed antennas could cooperate with each other, and aimed at improving the mutual information between transmitters and receivers. As the choice of a relay node in CC directly affects its performance, several researchers studied the problem of relay node assignment in single-hop



(a) Number of active nodes in the network.



(b) Ratio of objective values from D-GRS and CPLEX.

Fig. 13. Results showing near-optimality of D-GRS under online dynamics (random weights).

networks (see e.g., [5], [23], and the references within). For multi-hop networks, the relay node problem was shown to be coupled tightly with flow-routing (see e.g., [12], [18], and the references within).

(b) NC. The concept of NC [10] was first introduced by Ahlswede *et al.* in [2], where they showed that NC can save bandwidth in a wired network with multicast flows. The core idea of NC is to reduce the number of time slots required to transmit packets by combining multiple packets at the physical layer. Due to this important property, NC has quickly found applications in wireless networks, and can be categorized into two types: digital network coding (DNC) [13] and analog network coding (ANC) [14]. The reduction in the required time slots due to NC makes it an ideal candidate to improve the performance of CC.

(c) NC-CC. The benefits of employing NC in CC were recognized in [3], [21], [22], [25], [26], [27]. Due to the usage of NC with CC at physical layer, relay node selection is tightly coupled with session grouping. Most of the existing studies on NC-CC are information theoretic and limited to illustrating only the mechanism to combine NC with CC and the benefits of the combined approach, i.e., the time slot advantage. They do not address the issue of NC noise and how this tradeoff

could be leveraged in a general network through appropriate session grouping and relay node selection.

In [3], Bao *et al.* showed how to use NC with CC to improve the outage probability in a network with multiple source nodes and a single destination node. In [21], Peng *et al.* performed an analysis of outage probability in a network where NC is used by a single relay to enable CC for multiple sessions. In [25], Xiao *et al.* showed how NC could be used with CC to reduce the packet error rate in a simple two-source single-destination wireless network. In [22], Sharma *et al.* considered NC-CC with only one relay node. Their analysis showed that NC is not always good for CC, and introduced an important concept called NC noise. They showed that data rates of individual sessions in NC-CC are directly tied to the NC noise, which depends on individual sessions and the relay node. This motivated the study of the joint grouping and relay node selection problem in this paper.

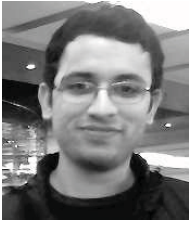
In [26], Xu and Li presented a CC framework for cellular networks that exploited NC opportunities but only worked in the presence of *bi-directional* traffic between two transmitters. It was not clear if/how their framework can be extended in a general network setting with unicast traffic and/or multiple destination nodes. In [27], the analysis of NC-CC was shown to improve throughput in a multi-hop wireless network. Again, the analysis in [27] was limited to bi-directional traffic, and the simple scenarios of two transmitters exploiting one relay node.

7 CONCLUSIONS

NC-CC is a powerful paradigm that uses NC to improve the performance of CC in a multi-session network. However, the benefits of NC-CC can only be fully exploited by appropriate session grouping and relay node selection. In this paper, we studied this problem with the goal of maximizing the sum of weighted rates among all the sessions in the network. We proved a generalized-weight version of the problem to be NP-hard, and then developed a distributed and online algorithm that offers near-optimal solution to this problem. We showed that our distributed algorithm has polynomial-time complexity, and demonstrated that D-GRS adapts well to online network dynamics.

REFERENCES

- [1] B. Aazhang, R.S. Blum, J.N. Laneman, K.J.R. Liu, W. Su, and A. Wittneben (Editors), *IEEE Journal on Selected Areas in Communications – Special Issue on Cooperative Communications and Networking*, vol. 25, no. 2, Feb. 2007.
- [2] R. Ahlswede, N. Cai, S.R. Li, and R.W. Yeung, “Network information flow,” *IEEE Trans. on Information Theory*, vol. 46, issue 4, pp. 1204–1216, July 2000.
- [3] X. Bao and J. Li, “Adaptive network coded cooperation (ANCC) for wireless relay networks: Matching code-on-graph with network-on-graph,” *IEEE Trans. on Wireless Communications*, vol. 7, no. 2, pp. 574–583, February 2008.
- [4] C. Barnhart, E. Johnson, G. Nemhauser, M. Savelsbergh, and P.H. Vance, “Branch-and-price: column generation for solving huge integer programs,” *Operations Research*, vol. 46, pp. 316–329, 1998.
- [5] A. Bletsas, A. Khisti, D. Reed, and A. Lippman, “A simple cooperative diversity method based on network path selection,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, pp. 659–672, March 2006.
- [6] T.M. Cover and A. EL Gamal, “Capacity theorems for the relay channel,” *IEEE Trans. on Information Theory*, vol. 25, issue 5, pp. 572–584, Sept. 1979.
- [7] IBM ILOG CPLEX: <http://www.ibm.com/software/integration/optimization/cplex/>
- [8] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman & Co., 1990.
- [9] A. Ghoniem and H.D. Sherali, “Complementary column generation and bounding approaches for set partitioning formulations,” *Optimization Letters*, vol. 3, no. 1, pp. 123–136, Jan. 2009.
- [10] The network coding home page: <http://www.ifp.uiuc.edu/~koetter/NWC/index.html>
- [11] IEEE 802.11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: <http://standards.ieee.org/getieee802/download/802.11-2007.pdf>
- [12] 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. on Mobile Computing*, vol. 6, no. 5, pp. 579–594, June 2007.
- [13] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, “XORs in the air: Practical wireless network coding,” in *Proc. ACM SIGCOMM*, pp. 497–510, Pisa, Italy, September 11–15, 2006.
- [14] S. Katti, S. Gollakotta, and D. Katabi, “Embracing wireless interference: Analog network coding,” in *Proc. ACM SIGCOMM*, pp. 397–408, Kyoto, Japan, August 27–31, 2007.
- [15] G. Kramer, I. Maric, and R.D. Yates, *Foundations and Trends in Networking: Cooperative Communications*, NOW publishers, vol. 1, nos. 3–4, 2006.
- [16] G. Kramer, R. Berry, A. El Gamal, H. El Gamal, M. Franceschetti, M. Gastpar, and J.N. Laneman (Editors), *IEEE Trans. on Information Theory – Special Issue on Models, Theory, and Codes for Relaying and Cooperation in Communication Networks*, vol. 53, no. 10, Oct. 2007.
- [17] M. Kurth, A. Zubow, and J-P. Redlich, “Cooperative opportunistic routing using transmit diversity in wireless mesh networks,” in *Proc. of IEEE INFOCOM*, pp. 1984–1992, Phoenix, AZ, April 15–17, 2008.
- [18] S. Lakshmanan and R. Sivakumar, “Diversity routing for multi-hop wireless networks with cooperative transmissions,” in *Proc. IEEE SECON*, pp. 1–9, Rome, Italy, June 22–26, 2009.
- [19] J.N. Laneman, D.N.C. Tse, and G.W. Wornell, “Cooperative diversity in wireless networks: Efficient protocols and outage behavior,” *IEEE Trans. on Information Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.
- [20] E.L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.
- [21] C. Peng, Q. Zhang, M. Zhao, and Y. Yao, “On the performance analysis of network-coded cooperation in wireless networks,” in *Proc. IEEE INFOCOM*, pp. 1460–1468, Anchorage, Alaska, May 6–12, 2007.
- [22] S. Sharma, Y. Shi, J. Liu, Y.T. Hou, S. Kompella, and S.F. Midkiff, “Network coding in cooperative communications: Friend or foe,” to appear *IEEE Trans. on Mobile Computing*, DOI: 10.1109/TMC.2011.130, 2012, available online at IEEE Xplore.
- [23] S. Sharma, Y. Shi, Y.T. Hou, and S. Kompella, “An optimal algorithm for relay node assignment in cooperative ad hoc networks,” *IEEE/ACM Trans. on Networking*, vol. 19, issue 3, pp. 879–892, June 2011.
- [24] E.C. van der Meulen, “Three terminal communication channels,” *Advances in Applied Probability*, vol. 3, pp. 120–154, 1971.
- [25] L. Xiao, T.E. Fuja, J. Kliewer, and D.J. Costello, “A network coding approach to cooperative diversity,” *IEEE Trans. on Information Theory*, vol. 53, no. 10, pp. 3714–3722, Oct. 2007.
- [26] H. Xu and B. Li, “XOR-assisted cooperative diversity in OFDMA wireless networks: Optimization framework and approximation algorithms,” in *Proc. IEEE INFOCOM*, pp. 2141–2149, Rio de Janeiro, Brazil, April 19–25, 2009.
- [27] J. Zhang and Q. Zhang, “Cooperative network coding-aware routing for multi-rate wireless networks,” in *Proc. IEEE INFOCOM*, pp. 181–189, Rio de Janeiro, April 19–25, 2009.



Sushant Sharma (S'06–M'11) received his Ph.D. degree in computer science from Virginia Tech, Blacksburg, VA, in 2010. He is currently working as a Research Associate in the Computational Science Center at Brookhaven National Laboratory, Upton, NY. His research interests include developing novel algorithms to solve network optimization problems, and building networked systems that utilize these algorithms.



Hanif D. Sherali is a University Distinguished Professor in the Industrial and Systems Engineering Department at Virginia Polytechnic Institute and State University. His areas of research interest are in analyzing problems and designing algorithms for specially structured linear, nonlinear, and integer programs arising in various applications, global optimization methods for non-convex programming problems, location and transportation theory and applications, and economic and energy mathematical modeling and analysis. He has published over 300 refereed articles in various Operations Research journals, has (co-) authored eight books in this area, and serves on the editorial board of eight journals. He is a Fellow of INFORMS and IIE, and an elected member of the *National Academy of Engineering*.



Yi Shi (S'02–M'08) received his Ph.D. degree in computer engineering from Virginia Tech, in 2007. He is an Adjunct Assistant Professor in the Department of Electrical and Computer Engineering at Virginia Tech. His research focuses on algorithms and optimization for interference management, energy management, cognitive radio networks, MIMO and cooperative communication networks, sensor networks, and ad hoc networks. He was a recipient of IEEE INFOCOM 2008 Best Paper Award and the only

Best Paper Award Runner-Up of IEEE INFOCOM 2011. He served as a TPC member for many major international conferences (including ACM MobiHoc and IEEE INFOCOM) and an Editor for IEEE Communications Surveys and Tutorials.



Y. Thomas Hou (S'91–M'98–SM'04) received his Ph.D. degree in Electrical Engineering from Polytechnic Institute of New York University in 1998. From 1997 to 2002, Dr. Hou was a Researcher at Fujitsu Laboratories of America, Sunnyvale, CA. Since 2002, he has been with Virginia Polytechnic Institute and State University (Virginia Tech), the Bradley Department of Electrical and Computer Engineering, Blacksburg, VA, where he is now a Professor.

Prof. Hous research interests are cross-layer optimization for wireless networks. Specifically, he is most interested in how to make significant improvement for network layer performance by exploiting new advances at the physical layer. He has published extensively in leading IEEE journals and top-tier IEEE/ACM conferences and received five best paper awards from IEEE (including IEEE INFOCOM 2008 Best Paper Award and IEEE ICNP 2002 Best Paper Award). Prof. Hou is currently serving as an Area Editor of IEEE Transactions on Wireless Communications, an Associate Editor of IEEE Transactions on Mobile Computing, an Editor of IEEE Journal on Selected Areas in Communications (Cognitive Radio Series), and an Editor of IEEE Wireless Communications. He currently serves as Chair of IEEE INFOCOM Steering Committee.

Prof. Hou co-edited a textbook titled *Cognitive Radio Communications and Networks: Principles and Practices*, which was published by Academic Press/Elsevier, 2010.



Sastry Kompella (S'04–M'07) received the Ph.D. degree in electrical and computer engineering from Virginia Polytechnic Institute and State University, Blacksburg, VA, in 2006. Currently, he is the Head of Wireless Network Research Section, Information Technology Division of US Naval Research Laboratory, Washington, DC. His research focuses on complex problems in cross-layer optimization and scheduling in wireless networks.