Joint Flow Routing and Relay Node Assignment in Cooperative Multi-Hop Networks

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

***Abstract*—It has been shown that cooperative communications (CC) has the potential to significantly increase the capacity of wireless networks. However, most of the existing results are limited to single-hop wireless networks. To explore the behavior of CC in multi-hop wireless networks, we study a joint optimization problem of relay node assignment and flow routing for a group of sessions. We develop a mathematical model and propose a solution procedure based on the branch-and-bound framework augmented with cutting planes (BB-CP). We design several novel components to speed-up the computational time of BB-CP. Via numerical results, we show the potential rate gain that can be achieved by incorporating CC in multi-hop networks.**

***Index Terms*—Cooperative communications, flow routing, relay assignment, multi-hop, wireless network.**

I. INTRODUCTION

COOPERATIVE communications (CC) is a novel physical layer mechanism where each node is equipped with only a single antenna and spatial diversity is achieved by exploiting the antennas on other nodes in the network. Although there has been active research on CC at the physical layer or for single-hop communications, results on CC in *multi-hop* wireless networks remain very limited. In this paper, we explore CC in multi-hop wireless networks by investigating a joint problem of relay node assignment and multi-hop flow routing. The objective of this problem is to maximize the minimum rate among a group of sessions, where each session may need to traverse multiple hops from its source to destination. The key problem we will address includes (1) the assignment of relay nodes (either for the purpose of CC or as a multi-hop relay) to each user session, and (2) the coupling problem of multi-hop flow routing and relay node assignment.

To solve the problem, we develop a mathematical characterization for cooperative relay node assignment and multi-hop flow routing. For the nonlinear constraints in the problem formulation, we show how to convert them into linear constraints by exploiting some problem-specific properties. The final problem formulation is in the form of a mixed-integer linear program (MILP). We propose a solution procedure based on a *branch-and-bound* framework augmented with cutting planes (BB-CP). Our proposed solution includes three novel components that make it highly efficient. First, we develop an efficient polynomial-time local search algorithm to generate feasible flow routes that exploit CC along individual hops. Second, by exploiting our problem structure, we devise a clever strategy for generating cutting planes that significantly decreases the number of branches in our branch-and-bound tree. Third, we present an innovative approach to perform branching operations that exploits problem-specific properties to choose superior branches and reduce the overall computational time. Our solution procedure provides $(1-\epsilon)$-optimal solutions, with ϵ being the desired approximation error bound.

The remainder of this paper is organized as follows. Section II presents related work. Section III describes our reference model for CC. In Section IV, we develop a mathematical model and problem formulation for joint cooperative relay node assignment and multi-hop routing. In Section V, we present our solution to the optimization problem. Section VI presents numerical results and Section VII concludes this paper.

II. RELATED WORK

Research of CC at the physical layer has been very active in recent years (see e.g., [4]–[6], [14], [16], [17]). These findings at the physical layer have found their applications in ad hoc networks, either for single-hop networks [2], [19], [22], [24] or for multi-hop networks [7], [8], [12], [15], [23]. In single-hop networks, the focus has been mainly on relay node assignment.

For multi-hop networks, Khandani *et al.* [8] studied a minimum energy routing problem (for a single message) by exploiting both wireless broadcast advantage and CC (called wireless cooperative advantage in the paper). They developed a dynamic programming based solution along with two heuristic algorithms. In [23], Yeh and Berry aimed to generalize the well known maximum differential backlog policy [20] in the context of CC. They formulated a challenging nonlinear program that characterized the network stability region, but only provided solutions for a few simple cases. In [15], Scaglione *et al.* proposed two architectures for multi-hop cooperative wireless networks. Under these architectures, nodes in the network can form multiple cooperative clusters. They showed that the network connectivity could be improved by using such cooperative clusters. In [7], [12], the authors proposed heuristics that decoupled the routing problem from relay node

Manuscript received 29 January 2011; revised 19 July 2011. An early version of this paper appeared in the Proceedings of IEEE INFOCOM 2010.
S. Sharma is with Brookhaven National Laboratory, Upton, NY 11705, USA (e-mail: sushant@bnl.gov). This work was completed while he was with Virginia Tech, Blacksburg, VA, USA.

Y. Shi, Y.T. Hou, H.D. Sherali, and S.F. Midkiff are with Virginia Tech, Blacksburg, VA 24061, USA (e-mail: yshi@vt.edu, thou@vt.edu, hanifs@vt.edu, midkiff@vt.edu).

S. Kompella is with the US Naval Research Laboratory, Washington, DC 20375, USA (e-mail: sastry.kompella@nrl.navy.mil).
Digital Object Identifier 10.1109/JSAC.2012.120203.

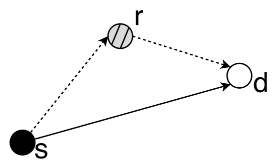


Fig. 1. A three-node reference model for CC.

assignment (so that only one problem was addressed at a time). In contrast, we consider joint flow routing and relay node assignment in this paper, which is necessary to explore optimality, but also is much harder to solve.

III. Reference Models

The essence of CC is to exploit (1) the wireless broadcast advantage and (2) the relaying capability of neighboring nodes so as to achieve higher data rate, lower transmission error, or other objectives in transmission [8], [10]. Figure 1 shows a three-node reference model for CC, where node s is a source node, node d is a destination node, and node r is a relay node.

In this paper, we employ orthogonal channels to resolve contention in multi-hop wireless network. Under this model, each node uses separate channels for transmission and reception and thus can transmit and receive data on different channels at the same time without self-interfering. Such an operation can be achieved by using a single antenna that has enough antenna bandwidth to accommodate separate channels for transmission and reception. In the following, we present the achievable rate between s and d under CC. We consider both the amplify-and-forward (AF) and decode-and-forward (DF) coding schemes [9], as well as direct transmission.

CC with Amplify-and-Forward (AF) Under this mode, relay node r receives, amplifies, and forwards the signal from source node s (all in analog form) to destination node d [9], and the destination node d combines the different signals received from s and r. Let h_{sd}, h_{sr}, h_{rd} capture the effects of path-loss, shadowing, and fading within the respective channels between nodes s and d, s and r, and r and d, respectively. Also, denote by z_d and z_r the zero-mean background noise at nodes d and r, with variance σ_d^2 and σ_r^2, respectively. For simplicity, we assume the background noise at a node has the same stochastic property on different channels. Denote by P_s and P_r the transmission powers at nodes s and r, respectively.

Following the same approach as that for deriving the rate under AF mode in [9], it can be shown that the achievable rate between s and d (with r as a relay) is

$$C_{\text{AF}}(s, r, d) = W \cdot I_{\text{AF}}(s, r, d) ,$$

where

$$I_{\text{AF}}(s, r, d) = \log_2 \left(1 + \text{SNR}_{sd} + \frac{\text{SNR}_{sr} \cdot \text{SNR}_{rd}}{\text{SNR}_{sr} + \text{SNR}_{rd} + 1} \right) ,$$

$\text{SNR}_{sd} = \frac{P_s}{\sigma_d^2}|h_{sd}|^2$, $\text{SNR}_{sr} = \frac{P_s}{\sigma_r^2}|h_{sr}|^2$, $\text{SNR}_{rd} = \frac{P_r}{\sigma_d^2}|h_{rd}|^2$, and W is the channel bandwidth.

CC with Decode-and-Forward (DF) Under this mode, relay node r first decodes and estimates the received signal from source node s, and then transmits the estimated data to destination node d [9]; the destination node d combines the different signals received from s and r. The achievable rate under DF mode can be developed by following the same approach as that in [9], which is

$$C_{\text{DF}}(s, r, d) = W \cdot I_{\text{DF}}(s, r, d) ,$$

where

$$I_{\text{DF}}(s, r, d) = \min\{\log_2(1 + \text{SNR}_{sr}), \log_2(1 + \text{SNR}_{sd} + \text{SNR}_{rd})\}.$$

Direct Transmission (without CC) When CC is not used, the achievable rate from source node s to destination node d is simply

$$C_{\text{D}}(s, d) = W \log_2(1 + \text{SNR}_{sd}) .$$

A couple of comments are in order. First, note that $I_{\text{AF}}(\cdot)$ and $I_{\text{DF}}(\cdot)$ are increasing functions of P_s and P_r, respectively. This suggests that, in order to achieve the maximum rate under either AF or DF, both the source node and the relay node should transmit at their maximum power P. Thus, we set $P_s = P_r = P$. Second, based on the rate expressions, one can see that although AF and DF are different physical layer mechanisms, the achievable rates for both of them have the same mathematical form, i.e., both of them are functions of SNR_{sd}, SNR_{sr}, and SNR_{rd}. Therefore, any solution procedure designed for AF can be readily extended for DF. As a result, it is sufficient to focus on developing a solution procedure for one of them, for which we choose AF in this paper.

IV. CC in Multi-hop Networks

A. Network Setting

We consider a group of sessions in a multi-hop wireless network. The data flow for each session may traverse multiple hops from its source to destination. As discussed in Section III, we employ orthogonal channels in the network, which allow different nodes to transmit simultaneously without interfering each other.

We distinguish relay nodes in the network into two types, based on their functionalities. We call a relay node used for CC purpose (i.e., node r in Fig. 1) as a *Cooperative Relay* (CR) and a relay node used for multi-hop relaying in the traditional sense as a *Multi-hop Relay* (MR). Note that a CR operates at the physical layer while an MR operates at the network layer.

Physical limitations of a wireless node may prohibit it from transmitting (or receiving) different data on multiple channels at the same time. As a result, we assume that a relay node may serve either as a CR or an MR, but not both at the same time. This also limits an MR to receive data from only one node, and to transmit data to one other node at any given time. Similarly, a CR node can serve at most one one transmitter and receiver pair. For the same reason, a source node (or destination node) cannot serve as a CR.

In [24], Zhao *et al.* showed that for a single hop, the diversity gain obtained by exploiting multiple relay nodes is only marginally higher than the diversity gain that can be obtained by selecting the *best* relay. As a result, we only consider at most one relay node for CC between each sender and receiver in this paper.

B. Mathematical Modeling

In this section, we present a mathematical model for our joint flow routing and relay node assignment problem. Denote $\mathcal{N}$ as the set of nodes in the network, with $|\mathcal{N}| = N$. In set $\mathcal{N}$, there are three subsets of nodes, namely, (i) the set of source nodes, $\mathcal{N}_s = \{s_1, s_2, \cdots, s_{N_s}\}$, with $N_s = |\mathcal{N}_s|$, (ii) the set of destination nodes, $\mathcal{N}_d = \{d_1, d_2, \cdots, d_{N_d}\}$, with $N_d = |\mathcal{N}_d| = N_s$, and (iii) the set of remaining nodes that are available for serving as CR or MR nodes, $\mathcal{N}_r = \{r_1, r_2, \cdots, r_{N_r}\}$, with $N_r = |\mathcal{N}_r|$. For clarity, we assume that all the source and destination nodes are distinct. Then we have $N = N_s + N_d + N_r = 2N_s + N_r$.

Role of Relay Nodes. Due to the existence of CRs, it is necessary to introduce integer variables to characterize whether or not an available relay node will be used as CR. A binary variable A_{uv}^w is defined for this purpose. Specifically,

$$A_{uv}^w = \begin{cases} 1 & \text{if node } w \text{ is used as a CR on hop } (u,v), \\ 0 & \text{otherwise.} \end{cases}$$

We also introduce another binary variable B_{uv} to specify whether or not the link from u to v is active in the routing solution. That is,

$$B_{uv} = \begin{cases} 1 & \text{if } v \text{ is the next hop node of node } u, \\ 0 & \text{otherwise.} \end{cases}$$

Each MR $w \in \mathcal{N}_r$ can receive data from only one previous hop, i.e., $\sum_{t\in\mathcal{N}}^{t\neq w} B_{tw} \leq 1$, and it can transmit data to only one next hop, i.e., $\sum_{t\in\mathcal{N}}^{t\neq w} B_{wt} \leq 1$. Furthermore, as a CR $w \in \mathcal{N}_r$ can serve only one hop, $\sum_{u\in\mathcal{N}}^{u\neq w, u\neq v} \sum_{v\in\mathcal{N}}^{v\neq w} A_{uv}^w \leq 1$. Also, we know that a relay can serve as only a CR or an MR, which can be enforced as follows:

$$\sum_{u\in\mathcal{N}}^{u\neq w, u\neq v} \sum_{v\in\mathcal{N}}^{v\neq w} A_{uv}^w + \sum_{t\in\mathcal{N}}^{t\neq w} B_{tw} \leq 1 \quad (w \in \mathcal{N}_r), (1)$$

$$\sum_{u\in\mathcal{N}}^{u\neq w, u\neq v} \sum_{v\in\mathcal{N}}^{v\neq w} A_{uv}^w + \sum_{t\in\mathcal{N}}^{t\neq w} B_{wt} \leq 1 \quad (w \in \mathcal{N}_r). (2)$$

In both (1) and (2), if the first term is 1 (i.e., node w is used as a CR), then the second term must be 0 (i.e., w cannot be used as an MR). Similarly, if the second term is 1 (i.e., node w is used as an MR), then the first term must be 0 (i.e., w cannot be used as a CR).

For a relay node $w \in \mathcal{N}_r$ that is being used as an MR, since w is not the destination node of any communication session, the traffic entering node w must also exit. This can be written as follows:

$$\sum_{u\in\mathcal{N}}^{u\neq w} B_{uw} = \sum_{v\in\mathcal{N}}^{v\neq w} B_{wv} \quad (w \in \mathcal{N}_r) . \quad (3)$$

Note that (3) also holds when w is not an MR. In this case, all B variables in (3) are zero.

It can be shown that once we have (3), it is sufficient to include either (1) or (2), but not both. As a result, we only include (1) in the problem formulation.

Furthermore, we may assign a relay node as a CR to hop (u,v) only if it is active (i.e., if $B_{uv} = 1$). Otherwise, no relay node should be assigned as a CR to hop (u,v). This constraint can be characterized as follows:

$$B_{uv} - \sum_{w\in\mathcal{N}_r}^{w\neq u, w\neq v} A_{uv}^w \geq 0 \quad (u \in \mathcal{N}, v \in \mathcal{N}, v \neq u) . \quad (4)$$

From the above constraint, we can see that when the value of some B_{uv} is 1, $\sum_{w\in\mathcal{N}_r}^{w\neq u, w\neq v} A_{uv}^w$ can be 1 or 0. This means that hop (u,v) is free to use either direct transmission ($\sum_{w\in\mathcal{N}_r}^{w\neq u, w\neq v} A_{uv}^w = 0$) or CC ($\sum_{w\in\mathcal{N}_r}^{w\neq u, w\neq v} A_{uv}^w = 1$).

Flow Routing. As explained earlier, due to transceiver limitations, a node can only transmit on one channel at any given time. As a result, we limit the transmission and reception of data at the network layer to only one transmitter and one receiver. This can be mathematically characterized by the following constraints:

$$\sum_{v\in\mathcal{N}}^{v\neq s_i} B_{s_i v} = 1 \quad (s_i \in \mathcal{N}_s) , \quad (5)$$

$$\sum_{v\in\mathcal{N}}^{v\neq u} B_{uv} \leq 1 \quad (u \notin \mathcal{N}_s) , \quad (6)$$

$$\sum_{u\in\mathcal{N}}^{u\neq v} B_{uv} \leq 1 \quad (v \notin \mathcal{N}_d) , \quad (7)$$

$$\sum_{v\in\mathcal{N}}^{v\neq d_i} B_{v d_i} = 1 \quad (d_i \in \mathcal{N}_d) , \quad (8)$$

where (5) says that a source node must transmit data to some other node and (8) says that a destination node must receive data from some node.

We note that there are some redundant constraints in (6) and (7). Constraint (6) can be partitioned into the following two sets of constraints:

$$\sum_{v\in\mathcal{N}}^{v\neq w} B_{wv} \leq 1 \quad (w \in \mathcal{N}_r), \quad (9)$$

$$\sum_{v\in\mathcal{N}}^{v\neq d_i, s_i} B_{d_i v} \leq 1 \quad (d_i \in \mathcal{N}_d). \quad (10)$$

Similarly, constraint (7) can be partitioned into the following two sets of constraints:

$$\sum_{u\in\mathcal{N}}^{u\neq w} B_{uw} \leq 1 \quad (w \in \mathcal{N}_r), \quad (11)$$

$$\sum_{u\in\mathcal{N}}^{u\neq s_i, d_i} B_{u s_i} \leq 1 \quad (s_i \in \mathcal{N}_s). \quad (12)$$

Note that due to (3), (9) is equivalent to (11). Thus, instead of using (6), we will use (10) in the final problem formulation.

Denote $f_{uv}(s_i)$ as the flow rate on link (u,v) that is attributed to session (s_i, d_i). The flow balance at an intermediate node w along the path between s_i and d_i can be formulated as follows:

$$\sum_{u\in\mathcal{N}}^{u\neq w, u\neq d_i} f_{uw}(s_i) = \sum_{v\in\mathcal{N}}^{v\neq w, v\neq s_i} f_{wv}(s_i)$$
$$(s_i \in \mathcal{N}_s, w \in \mathcal{N}, w \neq d_i, w \neq s_i) . \quad (13)$$

Using (13), it is easy to show that $\sum_{w\in\mathcal{N}}^{w\neq s_i} f_{s_i w}(s_i) = \sum_{w\in\mathcal{N}}^{w\neq d_i} f_{w d_i}(s_i)$, which states that all data generated by a source node s_i must be sent to its destination node d_i.

Rate Constraints. To ensure the feasibility of the routing solution, we must consider the capacity constraint on each hop in the network. That is, the aggregate flow rates traversing link (u,v) must not exceed the capacity on this link, i.e.,

$$\sum_{s_i\in\mathcal{N}_s}^{s_i\neq v} f_{uv}(s_i) \leq \left(1 - \sum_{w\in\mathcal{N}_r}^{w\neq u, w\neq v} A_{uv}^w\right) C_{\text{D}}(u,v) B_{uv} +$$
$$\sum_{w\in\mathcal{N}_r}^{w\neq u, w\neq v} A_{uv}^w C_{\text{AF}}(u,w,v) B_{uv} \quad (u \in \mathcal{N}, v \in \mathcal{N}, v \neq u) . (14)$$

Note that on the right-hand-side (RHS) of (14), there can be at most one non-zero term, depending on whether direct transmission or CC is employed. If direct transmission is employed, then the first term on the RHS of (14) is non-zero and the second term is 0; the converse is true when CC is employed.

C. Problem Formulation

We consider a set of N_s sessions in the network, denoted by $\mathcal{N}_s$. The goal is to maximize the minimum flow rate among all active sessions via an optimal multi-hop flow routing and cooperative relay assignment. More formally, for a given session (s_i, d_i), denote the end-to-end flow rate (or throughput) as R_{s_i}, where $R_{s_i} = \sum_{v \in \mathcal{N}}^{v \neq s_i} f_{s_i v}(s_i)$. Denote $R_{\min}$ as the minimum flow rate among all sessions, i.e.,

$$R_{\min} \leq \sum_{v \in \mathcal{N}}^{v \neq s_i} f_{s_i v}(s_i) \quad (s_i \in \mathcal{N}_s) \ . \qquad (15)$$

Then our objective is to maximize $R_{\min}$.

As part of our reformulation effort, we would like to convert the nonlinear constraint (14) into a linear constraint. The constraint in (14) contains the product of two variables A_{uv}^w and B_{uv} and is thus in nonlinear form. We can reformulate it into a linear constraint by exploiting the following property for A_{uv}^w and B_{uv}:

Property 1. *For any $u \in \mathcal{N}, v \in \mathcal{N}, v \neq u, w \in \mathcal{N}_r, w \neq u, w \neq v$,*

$$B_{uv} \cdot A_{uv}^w = A_{uv}^w .$$

Proof: This property is proved by considering both cases of B_{uv}.

(i) When $B_{uv} = 1$, the equality holds trivially.

(ii) When $B_{uv} = 0$, link (u, v) is not active. As a result, no CR should be assigned to (u, v), i.e., $A_{uv}^w = 0$ for $w \in \mathcal{N}_r, w \neq u, w \neq v$ by (4). Hence, the equality again holds. ■

By using Property 1, we can rewrite (14) as follows:

$$\sum_{s_i \in \mathcal{N}_s}^{s_i \neq v} f_{uv}(s_i) \leq \left(B_{uv} - \sum_{w \in \mathcal{N}}^{w \neq u, w \neq v} A_{uv}^w \right) C_{\mathrm{D}}(u,v) + \sum_{w \in \mathcal{N}}^{w \neq u, w \neq v} A_{uv}^w C_{\mathrm{AF}}(u,w,v) \qquad (u \in \mathcal{N}, v \in \mathcal{N}, v \neq u) \ , (16)$$

which is now a linear constraint.

All of our constraints are now linear. We have the following problem formulation:

$$\begin{array}{ll} \text{Max} & R_{\min} \\ \text{s.t.} & (1), (3), (4), (5), (7), (8), (10), (13), (15), (16) \\ & R_{\min}, f_{uv}(s_i) \geq 0 \\ & \quad (s_i \in \mathcal{N}_s, u \in \mathcal{N}, v \in \mathcal{N}, u \neq v, d_i, v \neq s_i) \\ & A_{uv}^w, B_{uv} \in \{0, 1\} \\ & \quad (w \in \mathcal{N}_r, u \in \mathcal{N}, v \in \mathcal{N}, u \neq v \neq w) \end{array}$$

where $R_{\min}$, $f_{uv}(s_i)$, A_{uv}^w, and B_{uv} are optimization variables.

It is not hard to see that this formulation is in the form of a *mixed-integer linear program* (MILP), which is NP-hard in general [3], [21].

V. Proposed Solution Procedure

For the MILP problem formulation, we propose a solution procedure based on the so-called *branch-and-bound* framework augmented with cutting planes [11] (BB-CP). BB-CP is an enhancement of BB by using a CP method to efficiently handle integer variables [1], [11]. Under this framework, we propose several novel problem-specific components. We show that our solution procedure yields a $(1 - \epsilon)$-optimal solution to the MILP problem, where the value of $0 \leq \epsilon \ll 1$ reflects the desired accuracy.

In Section V-A, we give a brief overview of the BB-CP framework [11]. Then in Sections V-B to V-D, we describe several novel components in the solution procedure.

A. Overview of the Algorithm

The BB-CP solution procedure consists of a set of iterative steps. During the first iterative step, an upper bound on the objective value is obtained by solving a "relaxed version" of the MILP problem. This relaxed problem is in the form of an LP and thus can be solved in polynomial time. However, due to relaxation, the values of A_{uv}^w and B_{uv} in the solution may become fractional, and the relaxed solution will thus not be feasible to the original MILP problem. Therefore, a local search algorithm, which we call *Feasible Solution Construction* (FSC), is proposed to obtain a feasible solution from the relaxed solution. The feasible solution obtained from FSC provides a lower bound on the objective value. If the gap between the upper and lower bounds is greater than ϵ (the desired gap), cutting planes are added to the problem. A cutting plane is a linear constraint that reduces the feasible region of the relaxed problem (but not the original MILP), thereby improving the values of the upper and lower bounds. After adding each new cutting plane, the relaxed LP is solved again. This relaxed solution will yield an improved upper bound (possibly with fractional A_{uv}^w- and B_{uv}-values). Each new upper bounding solution can then be used to find a new feasible (possibly improving) lower bounding solution via our local search FSC algorithm. The process of adding cutting planes to the relaxed problem continues until the improvement in upper and lower bounds becomes marginal, i.e., within a certain percentage threshold.

After cutting planes can no longer improve the bounds, the problem is partitioned into two subproblems. The relaxed versions of the two subproblems are then solved and FSC is used to obtain the upper and lower bounds for each subproblem. This step finishes the iteration.

After each iteration, if the gap between the largest upper bound (among all the subproblems), and the largest lower bound (among all the subproblems) is more than ϵ, another iterative step (similar to the first step) is performed on the subproblem having the largest upper bound. Note that after every iteration, the chosen subproblem is partitioned into two subproblems, increasing the total number of subproblems that we have.

For some subproblem, if the upper and lower bounds coincide, then this subproblem is completely solved, and this subproblem is not chosen for branching in future iterations. Also, since our goal is to obtain a $(1 - \epsilon)$-optimal solution, if $(1 - \epsilon)$ times the upper bound of a certain subproblem is less than or equal to the largest lower bound among all the subproblems, then this subproblem can be removed from the problem list, as it will not affect the $(1 - \epsilon)$-optimality of the final solution. This can be explained by considering the following two cases:

- **Case 1:** The global optimal solution is not in the subproblem that was removed: In this case, the removal of the subproblem will not cause the removal of the optimal solution.
- **Case 2:** The global optimal solution is in the subproblem that was removed: In this case, the current lower bound is already $(1-\epsilon)$-optimal. Thus, removal of the subproblem will not prevent us from finding the $(1-\epsilon)$-optimal solution (as we already have one at hand, i.e., the lower bounding solution).

The iterations of BB-CP continue until the largest upper bound (among all the current subproblems) and the largest lower bound among all the subproblems (i.e., the best feasible solution value) are within ϵ of each other. At this point, the best feasible solution is $(1-\epsilon)$-optimal. As one can see, the key challenge in implementing a BB-CP framework is in the details, i.e., how each component is designed. We propose the following novel components:

1) An efficient polynomial-time local search algorithm, called *Feasible Solution Construction* (FSC) algorithm. FSC generates feasible flow routes that exploit CC along individual hops.
2) By exploiting the problem structure, we establish a judicious strategy to generate cutting planes that significantly decrease the number of branches in our BB tree.
3) An effective approach to perform branching operations, which exploits problem-specific properties to select superior branches and hence reduce the overall computational time.

Although the worst case complexity of our solution remains exponential (due to MILP), the actual run-time is in fact reasonable. This reasonable run-time is mainly attributed to our proposed new components in the branch-and-bound framework.

B. FSC Algorithm

After solving the relaxed MILP, the solution may have fractional values for some of the interger variables A^w_{uv} or B_{uv}, which is clearly infeasible. The proposed FSC is a *local search* algorithm that constructs a feasible solution based on a given infeasible solution by determining feasible routings, CR assignments, and flow rates for all sessions in the network.

Our proposed FSC algorithm is a polynomial-time algorithm that consists of three phases: *Path Determination, CR Assignment,* and *Flow Re-calculation*. In the following, we give details on these three phases.

Phase 1: Path Determination. The goal of this first phase is to find a feasible and potentially high capacity path for each session. In this phase, FSC starts by assuming no prior paths exist for any session in the network. Among the sessions whose paths are yet to be determined, the algorithm performs path determination for a session (chosen at random) iteratively.

When determining the next-hop node, FSC takes the following approach. Suppose that we are searching the next hop node for a node r_i. In the relaxed solution, it is possible that r_i may have multiple next-hop nodes. Here, among these candidate next-hop nodes, we select the node r_j to which r_i is transmitting the largest amount of data (in the relaxed solution) for source node s_i. This "widest pipe" approach, although heuristic, has the potential of finding a high capacity path. Note that once a node is included in a path, it will not be considered for inclusion in the other paths during subsequent iterations.

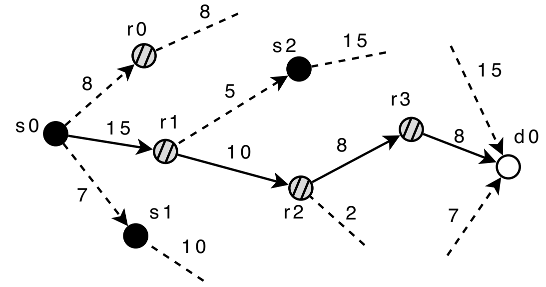


Fig. 2. A simple path between a source and its destination where intermediate nodes are all in $\mathcal{N}_r$.

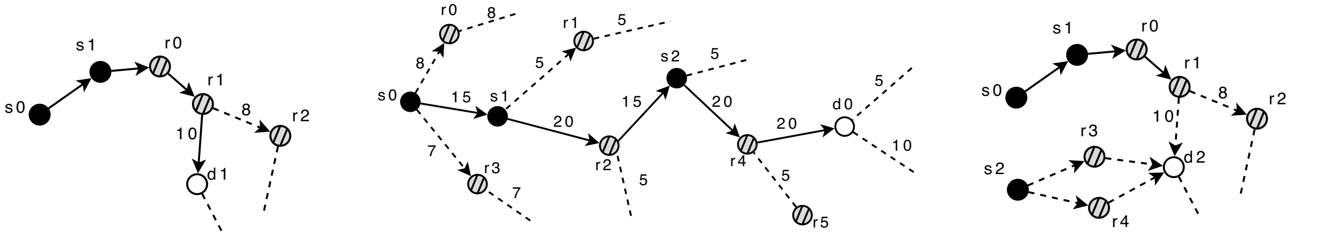
Case 1 – Simple Path We consider the simple case first, where after the widest-pipe approach, the intermediate nodes between the source and destination nodes of a session are all from the set $\mathcal{N}_r$. An example is shown in Fig. 2, where the final path between s_0 and d_0 will go through r_1, r_2 and r_3, with r_1, r_2 and r_3 all being in set $\mathcal{N}_r$. In this case, Phase 1 for the selected session is considered complete, and the algorithm will move on to Phase 2 for the selected session.

Case 2 – Overlapping Path In this case, based on the widest-pipe approach, we have encountered an intermediate node from the set $\mathcal{N}_s$ or $\mathcal{N}_d$, i.e., a source or destination node. So the path under consideration may overlap with the path of another session. This is the most complex situation that we need to deal with in the path determination phase. Depending on the type of this intermediate node (source or destination), different mechanisms need to be devised.

Sub-case 2.1: The encountered intermediate node is the source node of another session. In this case, the encountered intermediate node (say s_j) is included in the path as the next-hop node. At the same time, s_j is recorded in a special list (denoted as $\mathcal{L}$) to keep track of such source nodes of other sessions that have not yet found their own paths to their corresponding destination nodes but are included in the path during the current iteration. Note that the source node s_i for the path under construction is not listed in $\mathcal{L}$.

Sub-case 2.2: The encountered intermediate node is a destination node. This includes a number of scenarios.

A. **This node is the destination node of a source node in $\mathcal{L}$:** In this case, this node will be included in the current path under construction. Its corresponding source node will be removed from $\mathcal{L}$, since the path for that source node is complete. An an example, in Fig. 3(a), the path under construction is for source node s_0. Currently, node d_1 is considered as the next hop node along the path and the corresponding source node s_1 is in list $\mathcal{L}$. In this case, d_1 is added to the path and s_1 is removed from list $\mathcal{L}$.
B. **This node is the destination node of the current path under construction.** In this case, this node is included in the path and the path construction for the intended



(a) Case 2.2(A): The source of the encountered destination node is in $\mathcal{L}$.

(b) Case 2.2(B): The encountered node is the destination node of the current path under construction.

(c) Case 2.2(C): Source node for d_2 is not on the path under construction.

Fig. 3. Examples illustrating different scenarios in Sub-case 2.2 during Phase 1 of FSC.

source node is complete. If list $\mathcal{L}$ is empty, the current iteration finishes and the algorithm moves on to the next iteration for the remaining nodes.

But list $\mathcal{L}$ may not be empty at this point, meaning that some source nodes of other overlapping paths still do not have a complete path to their corresponding destination nodes. As an example, in Fig. 3(b), the source node of the path under construction is s_0. Source nodes s_1 and s_2 are included in the path and thus are in $\mathcal{L}$. When the destination node d_0 (for s_0) is included in the path, the path construction for s_0 is complete. But the paths for s_1 and s_2 remain incomplete.

If $\mathcal{L}$ is not empty, the iteration continues by taking a source node from $\mathcal{L}$ as the current intended source node and continues path construction for this node. In our algorithm, if there are multiple source nodes in $\mathcal{L}$, we pick the source node that has the largest share of out-going flow at the current encountered node. Once chosen, this source node is removed from $\mathcal{L}$ and the path construction continues.

In the case that the current encountered node is not carrying any flow for any of the source nodes in $\mathcal{L}$, then all source nodes in $\mathcal{L}$ will be removed from the current path as well as from list $\mathcal{L}$. This is done by removing a source node from the current path (which was in list $\mathcal{L}$) and directly connecting its preceding and succeeding nodes in the path. For example, in Fig. 3(b), the resulting path by removing s_1 and s_2 will be s_0-r_2-r_4-d_0. At this point, list $\mathcal{L}$ is empty and the current iteration finishes. The algorithm will move on to the next iteration to examine the remaining nodes.

C. **This node is the destination node whose source node is not on the current path under construction.** In this case, this node must not be included in the path and the node receiving the next largest flow will be considered. This scenario is illustrated in Fig. 3(c).

In Fig. 3(c), when d_2 is considered for the next node of r_1, then d_2 will not be included in the path because s_2 is not on the current path. As a result, another node r_2 will be considered and included as the next node for r_1. This will ensure that a different path can be explored for s_2 in a future iteration.

Upon completion of an iteration of path determination, list $\mathcal{L}$ must be empty. The algorithm will then move on to the next iteration of path determination for the remaining source nodes whose paths are not yet determined. Note that all the source nodes, destination nodes, and relay nodes that are already included in a path are removed from further consideration, due to the physical layer constraint we discussed earlier. The iteration continues until all the paths for all the source nodes are determined. The pseudo-code for this phase is given in [18].

Phase 2: CR Assignment. After Phase 1, there may still be some nodes in $\mathcal{N}_r$ that are not yet used and are thus available to serve as CR nodes. The goal of Phase 2 is to consider how to assign these remaining relay nodes as CR nodes to increase capacity.

In our algorithm, we use the values of the A_{uv}^w-variables in the relaxed solution to assign these remaining relay nodes. First, we introduce a term called *capacity-flow-ratio* (CFR) for a hop as the ratio of the hop's capacity (assuming direct transmission without CC) to the number of overlapping sessions (as determined by Phase 1) on that hop. Then, we order the hops among all paths in Phase 1 in non-decreasing order of CFR. The assignment of CRs starts with the hop having the minimum CFR and continues in increasing order. For a particular hop, say (u, v), under consideration, we choose the CR node with the largest A_{uv}^w-value among all CR nodes from the relaxed solution. In the case that the largest A_{uv}^w is 0 for this hop in the relaxed solution, no CR node will be assigned to this hop. The iteration continues until all the hops are considered for CR node assignment or the remaining available CR nodes are all assigned.

Phase 3: Flow Re-calculation. Upon the completion of Phases 1 and 2, all the integer variables $A_{u,v}^w$ and B_{uv} are now fixed (either 0 or 1). Consequently, the original MILP (in Section IV-C) is now reduced to an LP as follows:

$$
\begin{array}{lll}
\text{Max} & R_{\min} \\
\text{s.t.} & (13), (15), (16) \\
& R_{\min}, f_{uv}(s_i) \geq 0 \\
& (s_i \in \mathcal{N}_s, u \in \mathcal{N}, v \in \mathcal{N}, u \neq v, d_i, v \neq s_i).
\end{array}
$$

In Phase 3, we solve the above LP and obtain feasible values for the $f_{uv}(s_i)$-variables. The value of the objective function obtained via this LP can be used as a lower bound in the branching process.

C. Generating a Cutting Plane

The process of adding a cutting plane, which is a linear constraint, begins by examining the values of the A_{uv}^w- and

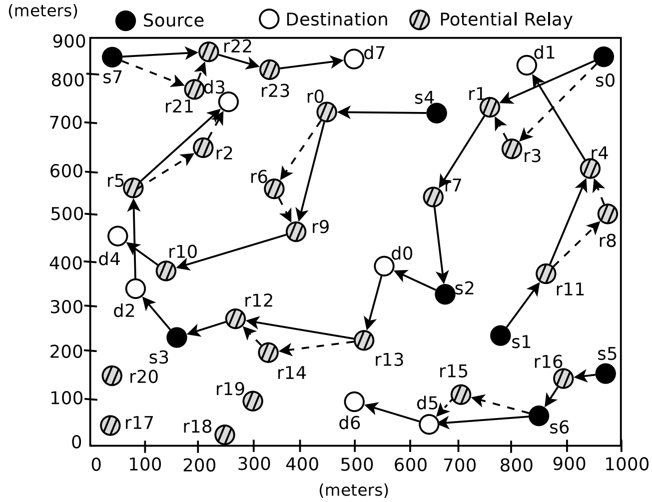


Fig. 4. Jointly optimal CR node assignment and flow routing for the 40-node network.

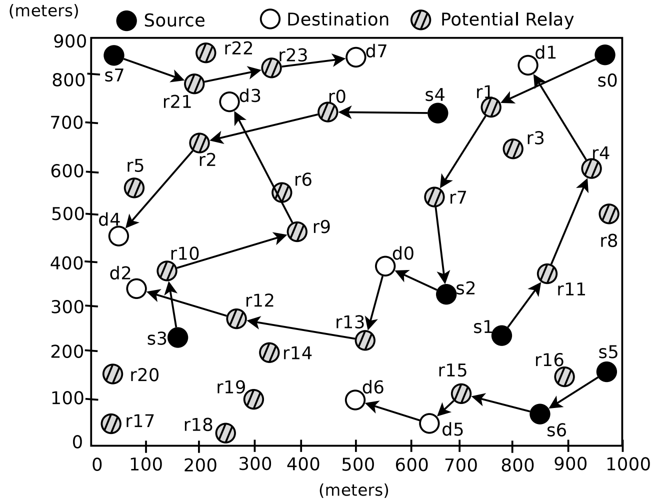


Fig. 5. Optimal flow routing solution for the 40-node network (CC is not employed).

B_{uv}-variables in the solution to the relaxed MILP problem for any selected subproblem. If there are multiple A_{uv}^w- or B_{uv}-variables having fractional values, then a decision must be made on which one of these fractional variables will be chosen to generate a cutting plane. Such a decision should exploit some problem-specific property and our algorithm is based on the following observation. If any A_{uv}^w- or B_{uv}-variable is assigned to 1, then some other relevant variables can immediately be assigned to 0. For example, if A_{uv}^w is assigned to 1, then node w cannot be used as an MR on any path. So, if we can improve the chances of a variable getting assigned to 1 (by selecting it for generating a cutting plane), then we can quickly fix a number of other relevant variables and reduce the problem size in the subsequent branching process. Thus in our algorithm, we propose to choose a variable that is fractional, but closest to the value of 1 in the relaxed solution when generating a cutting plane.

After selecting the variable for the cutting plane, the next step is to generate a linear constraint as the cutting plane, for which we choose the Gomory cutting plane method [11], [13], which we find the most efficient in implementation. The details can be found in [18].

D. Selection of Branching Variables

When the addition of cutting planes is no longer able to offer much improvement in upper and lower bounds for a relaxed problem, we move on to the branching process in the algorithm. The candidate variables for branching are those A_{uv}^w- and B_{uv}-variables having fractional values in the relaxed solution. Although the choice among these variables will not affect the convergence of the BB-CP algorithm, a wise choice of branching variable can significantly speed up the convergence process.

In our solution procedure, we choose an A_{uv}^w- or B_{uv}-variable that is fractional, but nearest to either 0 or 1 for branching. We note here that although this is contrary to conventional wisdom (one typically selects a fractional variable that is closest to 0.5 for branching), the structure of our problem and the nature of the LP relaxation solution made this choice favorable. Once chosen, the current problem is then partitioned into two subproblems, with the value of the branching variable fixed as 0 in one subproblem and 1 in the other subproblem. Our choice of the branching variable is based on the following reasoning. If the variable that is closest to 0 is chosen, then in the two subproblems, it will have the value 1 in one subproblem and 0 in the other. For the subproblem where its value is 1, the new upper bound may be reduced significantly. Consequently, this subproblem (with the branching variable at value of 1) has the potential of having an upper bound lower than the current lower bound, making it eligible for removal from the problem list for further consideration. A similar argument also holds for the case when the variable chosen is closest to 1. Furthermore, note that fixing a particular variable to 1 also enables us to fix some other relevant variables to 0 and reduce the problem size (see the constraints in Section IV-B).

TABLE I
THROUGHPUT COMPARISON FOR THE 40-NODE NETWORK

Session	With CC (Mb/s)	Without CC (Mb/s)
$s_0 - d_0$	44.5	31.9
$s_1 - d_1$	47.4	34.8
$s_2 - d_2$	50.4	42.1
$s_3 - d_3$	49.0	31.1
$s_4 - d_4$	46.6	34.8
$s_5 - d_5$	47.2	42.9
$s_6 - d_6$	47.2	42.9
$s_7 - d_7$	91.7	79.1

VI. Numerical Results

In this section, we present some numerical results for our proposed solution. In our simulations, we set $W = 22$ MHz bandwidth for each channel. The maximum transmission power at each node is set to 1 W. For simplicity, we assume that h_{sd} only includes the path loss between nodes s and d and is given by $|h_{sd}|^2 = ||s-d||^{-4}$, where $||s-d||$ is the distance (in meters) between nodes s and d, and path loss index is 4. For the AWGN channel, we assume the noise variance is

10^{-10} W at all nodes. We set $\epsilon = 0.1$ in all cases to obtain $(1-\epsilon)$-optimal solutions.

We present results for a network with 40 wireless nodes as shown in Fig. 4. There are eight sessions in the network, with eight source nodes ($N_s = 8$), eight destination nodes ($N_d = 8$), and 24 available relay nodes ($N_r = 24$). For this 40-node network, Fig. 4 displays our solution for joint CR node assignment and flow routing. The rate for each session is shown in Table I (second column), with the minimum rate among all sessions being 44.5 Mb/s. As a comparison, Fig. 5 also exhibits the optimal flow routing solution when CC is not used. The rate for each session when CC is not used is shown in Table I (third column) with the minimum rate among all sessions being 31.1 Mb/s, which is less than 44.5 Mb/s.

VII. Conclusions

In this paper, we explored CC in multi-hop wireless networks by studying a joint relay node assignment and multi-hop flow routing problem. This optimization problem is inherently difficult due to its mixed-integer nature and very large solution space. We developed an efficient solution procedure based on a branch-and-bound framework augmented with a cutting plane algorithm that has several novel components to speed up computations. Our results demonstrated the significant rate gains that can be achieved by incorporating CC in a multi-hop wireless network.

Acknowledgements

The work of Y.T. Hou, H.D. Sherali, and S.F. Midkiff was supported in part by NSF under grant CNS-1064953. The work of S. Kompella was supported in part by the ONR.

References

[1] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty, *Nonlinear programming: Theory and algorithms,* Wiley, New York, 2006.

[2] J. Cai, S. Shen, J. W. Mark, and A. S. Alfa, "Semi-distributed user relaying algorithm for amplify-and-forward wireless relay networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 4, pp. 1348–1357, April 2008.

[3] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness,* W.H. Freeman and Company, New York, 1979.

[4] D. Gunduz and E. Erkip, "Opportunistic cooperation by dynamic resource allocation," *IEEE Trans. Wireless Commun.*, vol. 6, no. 4, pp. 1446–1454, April 2007.

[5] O. Gurewitz, A. de Baynast, and E.W. Knightly, "Cooperative strategies and achievable rate for tree networks with optimal spatial reuse," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3596–3614, October 2007.

[6] T.E. Hunter and A. Nosratinia, "Diversity through coded cooperation," *IEEE Trans. Wireless Commun.,* vol. 5, no. 2, pp. 283–289, February 2006.

[7] 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. Mobile Comput.,* vol. 6, no. 5, pp. 579–594, June 2007.

[8] A.E. Khandani, J. Abounadi, E. Modiano, and L. Zheng, "Cooperative routing in static wireless networks," *IEEE Trans. Commun.,* vo. 55, no. 11, pp. 2185–2192, November 2007.

[9] J.N. Laneman, D.N.C. 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, December 2004.

[10] F. Li, K. Wu, and A. Lippman, "Energy-efficient cooperative routing in multi-hop wireless ad hoc networks," In *Proc. 25th IEEE International Performance, Computing, and Communications Conference,* pp. 215–222, Phoenix, AZ, April 10–12, 2006.

[11] Y. Pochet and L.A. Wolsey, *Production Planning by Mixed Integer Programming,* Springer, New York, 2006.

[12] S. Lakshmanan and R. Sivakumar, "Diversity routing for multi-hop wireless networks with cooperative transmissions," In *Proc. IEEE SECON,* pp. 610–618, Rome, Italy, June 22–26, 2009.

[13] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization,* Wiley, New York, 1999.

[14] S. Savazzi and U. Spagnolini, "Energy aware power allocation strategies for multihop-cooperative transmission schemes," *IEEE J. Sel. Areas Commun.,* vol. 25, no. 2, pp. 318–327, February 2007.

[15] A. Scaglione, D.L. Goeckel, and J.N. Laneman, "Cooperative communications in mobile ad hoc networks," *IEEE Signal Processing Mag.*, vol. 23, no. 5, pp. 18–29, September 2006.

[16] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity – part I: system description," *IEEE Trans. Commun.,* vol. 51, no. 11, pp. 1927–1938, November 2003.

[17] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity – part II: implementation aspects and performance analysis," *IEEE Trans. Commun.,* vol. 51, no. 11, pp. 1939–1948, November 2003.

[18] S. Sharma, Y. Shi, Y.T. Hou, H.D. Sherali, S. Kompella, and S.F. Midkiff, "Joint flow routing and relay node assignment in cooperative multi-hop networks," *Technical Report,* Bradley Dept. of Electrical and Computer Engineering, Virginia Tech, September 2011. Available online at http://sushantsharma.com/.

[19] Y. Shi, S. Sharma, Y.T. Hou, and S. Kompella, "Optimal relay assignment for cooperative communications," In *Proc. ACM MobiHoc*, pp. 3–12, Hongkong, China, May 27–30, 2008.

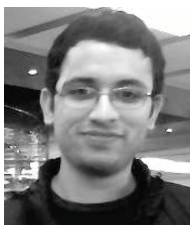
[20] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Contr.*, vol. 37, no. 12, pp. 1936–1948, December 1992.

[21] V.V. Vazirani, *Approximation Algorithms,* Springer Verlag, Berlin, Germany, 2001.

[22] B. Wang, Z. Han, and K.J.R. Liu, "Distributed relay selection and power control for multiuser cooperative communication networks using buyer/seller game," In *Proc. of IEEE INFOCOM,* pp. 544–552, Anchorage, Alaska, May 6–12, 2007.

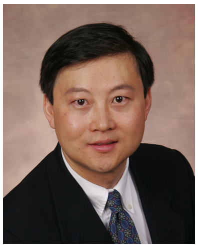
[23] E.M. Yeh and R.A. Berry, "Throughput optimal control of cooperative relay networks," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3827–3833, October 2007.

[24] Y. Zhao, R. Adve, and T.J. Lim, "Improving amplify-and-forward relay networks: optimal power allocation versus selection," In *Proc. IEEE International Symposium on Information Theory,* pp. 1234–1238, Seattle, USA, July 9–14 2006.



Sushant Sharma (S'06–M'11) received his Ph.D. degree in computer science from Virginia Tech, Blacksburg, VA, in 2010. He is currently a Research Associate at Brookhaven National Laboratory, Upton, NY. His research interests include developing novel algorithms to solve optimization problems in wired and wireless networks.

Yi Shi (S'02–M'08) received his Ph.D. degree in computer engineering from Virginia Tech, Blacksburg, VA, in 2007. He is currently a Research Scientist in the Department of Electrical and Computer Engineering at Virginia Tech. Dr. Shis research focuses on algorithms and optimization for wireless networks.



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. He is currently an Associate Professor at Virginia Tech, Blacksburg, VA. His research interests are cross-layer design and optimization for wireless networks.

Hanif D. Sherali is a University Distinguished Professor and the W. Thomas Rice Chaired Professor of Engineering in the Industrial and Systems Engineering Department at Virginia Polytechnic Institute and State University. He is an elected member of the U.S. National Academy of Engineering.

Sastry Kompella (S'04–M'07) received his Ph.D. degree in computer engineering from Virginia Tech, Blacksburg, VA, in 2006. He is Head of Wireless Network Research Section of US Naval Research Laboratory, Washington, DC. His research focuses on complex problems in wireless networks.

Scott F. Midkiff (S'82–M'85–SM'92) is Professor and Department Head in the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA. His research interests include wireless networks, network services for pervasive computing, and cyber-physical systems.