Optimal Relay Assignment for Cooperative Communications

Yi Shi* Sushant Sharma† Y. Thomas Hou*
*Department of Electrical and Computer Engineering
†Department of Computer Science
Virginia Tech, Blacksburg, VA, USA
{yshi,sushant,thou}@vt.edu

Sastry Kompella
Information Technology Division
Naval Research Laboratory
Washington DC, USA
kompella@itd.nrl.navy.mil

ABSTRACT

Recently, cooperative communications, in the form of keeping each node with a single antenna and having a node exploit a relay node's antenna, is shown to be a promising approach to achieve spatial diversity. Under this communication paradigm, the choice of relay node plays a significant role in the overall system performance. In this paper, we study the relay node assignment problem in a network environment, where multiple source-destination pairs compete for the same pool of relay nodes in the network. The main contribution of this paper is the development of a polynomial time algorithm to solve this problem. A key idea in this algorithm is a "linear marking" mechanism, which is able to offer a linear complexity for each iteration. We give a formal proof of optimality for this algorithm. We also show several attractive properties associated with this algorithm.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

General Terms

Algorithm

Keywords

Cooperative Communications, Wireless Networks, Network Capacity

1. INTRODUCTION

Spatial diversity, in the form of employing multiple transceiver antennas, is shown to be very effective in coping fading in wireless channel. However, equipping a wireless node with multiple antennas may not be practical, as the footprint of multiple antennas may not fit on a wireless node (particularly handheld wireless device). To achieve spatial diversity without requiring multiple transceiver antennas on the same node, the so-called *cooperative communications* has been introduced [6, 8]. Under cooperative communications, each node is equipped with only a single transceiver and spatial diversity is achieved by exploiting the antenna on another (cooperative) node in the network.

There are two categories of cooperative communications, namely, *amplify-and-forward* (AF) and *decode-and-forward* (DF) [6]. Under AF, the cooperative relay node performs a linear operation on the signal received from the information source before forwarding it to the destination node. Under DF, the cooperative relay node decodes the received signal, and re-encodes it before forwarding it to the destination node. Regardless of AF or DF, the choice of a relay node plays a significant role in the final performance of cooperative communications [2, 3, 10]. As we shall see in Section 2, an improperly chosen relay node may offer a smaller capacity for a source-destination pair than that under direct transmissions.

In this paper, we study relay node assignment problem in a network environment. Specifically, we consider an ad hoc network environment where there are multiple active source-destination pairs and the remaining nodes can be exploited as relay nodes. We want to find out how to optimally assign relay nodes to the source and destination pairs so as to maximize the minimum capacity among all pairs. Although solution to this problem can be found via exhaustive search (among all possible relay node assignments), the complexity of this approach is exponential. Our goal in this paper is to find a polynomial-time complexity algorithm to this problem.

1.1 Related Work

The cooperative communication paradigm can trace back to the pioneering works done by van der Meulen [9] and Cover and El Gamal [4]. The readers are also referred to [1, 5] for some recent work on this subject. In this section, we focus our attention on related work for the relay node assignment problem.

Although it is possible for a source-destination pair to employ *multiple* relay nodes for cooperative communications, the benefit of such approach appears limited, as shown in a recent work by Zhao et al. [10]. In this work, Zhao et al. showed that for a source-destination pair, in the presence of multiple relay nodes, it is sufficient to choose the "best" relay to achieve full diversity order than to have multiple relay nodes participate. This result is interesting, as it paves the way for research on assigning no more than one relay node to a source-destination pair, which is the setting that we will adopt in this paper.

There has been much effort on selecting an optimal relay node (among a set of relay nodes) for a single source-destination pair (see, e.g., [2]). However, these schemes are limited to a single source-destination pair and cannot be easily extended to a network environment where there are multiple source-destination pairs competing for the same pool of relay nodes, which is the focus of this paper.

In [7], Ng et al. studied a utility maximization problem for the joint optimization of relay node selection, cooperative communica-

MobiHoc'08, May 26–30, 2008, Hong Kong SAR, China.

tions, and resource allocation in a cellular network. A key assumption in the solution to the optimization problem is infinite number of channels in the network (so that the duality gap of the optimization problem is zero). But this assumption may not hold in practice. Also, the complexity of the proposed solution is not polynomial.

In [3], Cai et al. studied relay node selection and power allocation for AF based wireless relay network. A simple network consisting of single source-destination pair was first studied. Then, the authors considered multiple source-destination pairs and proposed a semi-distributed algorithm on relay node selection. This algorithm is heuristic in nature and there is no performance guarantee (in terms of optimality).

1.2 Main Contribution of This Paper

In this paper, we study the optimal relay node assignment problem in a network setting. Specifically, we consider how to assign a set of relay nodes to a set of source-destination pairs so as to maximize the minimum capacity among the pairs. The main contributions of this paper are the following.

- We develop an algorithm, called Optimal Relay Assignment (ORA) algorithm, to solve the relay node assignment problem. A key idea in ORA is a "linear marking" mechanism, which is able to offer a linear complexity at each iteration. Due to this mechanism, ORA is able to achieve polynomial time complexity.
- We offer a formal proof of optimality for the ORA algorithm. The proof is based on contradiction and hinges on a clever recursive trace-back of source nodes and relay nodes in the solution by ORA and another hypothesized better solution.
- We show a number of nice properties associated with ORA. These include: (i) the algorithm works regardless whether the number of relay nodes in the network is more than or less than the number of source-destination pairs; (ii) the final capacity for each source-destination pair is guaranteed to be no less than that under direct transmissions; (iii) the algorithm is able to find the optimal objective regardless of initial relay node assignment.

1.3 Paper Organization

In Section 2, we give a brief overview of cooperative communications, which includes capacity calculation for both AF and DF. In Section 3, we present mathematical model for the relay node assignment problem in a network environment. Section 4 presents our Optimal Relay Assignment (ORA) algorithm. In Section 5, we give a proof of ORA's optimality. Section 6 presents some numerical results to demonstrate the capabilities of the ORA algorithm. Section 7 concludes this paper.

2. COOPERATIVE COMMUNICATIONS: A PRIMER

The essence of cooperative communications is best explained by a three-node example in Fig. 1. In this figure, node s is the source node, node d is the destination node, and node r is a relay node. Transmission from s to d is done on a frame-by-frame basis. Within a frame, there are two time slots. In the first time slot, source node s makes a transmission to destination node d. This transmission is also overheard by relay node r, due to the broadcast nature of wireless communications. In the second time slot, node r forwards the data received in the first time slot to d. Note that such a two-slot structure is necessary for cooperative communications due to the half-duplex nature of most wireless transceivers.

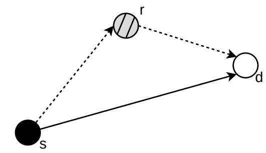

Figure 1: A three-node schematic for cooperative communications.

In this section, we give expressions for capacity under cooperative communications and direct transmissions (i.e., no cooperation). For cooperative communications, we consider both the so-called amplify-and-forward (AF) and decoded-and-forward (DF) modes [6].

Amplify-and-Forward (AF) Under this mode, let h_{sd}, h_{sr}, h_{rd} capture the effect of path-loss, shadowing, and fading between nodes s and d, s and r, and r and d, respectively. Also denote 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.

Denote x_s the signal transmitted by source node s in the first time slot. Then the received signal at destination node d, y_{sd}, can be expressed as

$$y_{sd} = h_{sd} x_s + z_d \ , \tag{1}$$

and the received signal at the relay node r, y_{sr}, is

$$y_{sr} = h_{sr} x_s + z_r \ . \tag{2}$$

In the second time slot, relay node r transmits to destination node d. The received signal at d, y_{rd}, can be expressed as

$$y_{rd} = h_{rd} \cdot \alpha_r \cdot y_{sr} + z_d \ ,$$

where α_r is the amplifying factor at relay node r and y_{sr} is given in (2). Thus, we have

$$y_{rd} = h_{rd}\alpha_r \cdot (h_{sr} x_s + z_r) + z_d \ . \tag{3}$$

The amplifying factor α_r at relay node r should satisfy power constraint $\alpha_r^2(|h_{sr}|^2 P_s + \sigma_r^2) = P_r$, where P_s and P_r are the transmission powers at nodes s and r, respectively. So, α_r is given by

$$\alpha_r^2 = \frac{P_r}{|h_{sr}|^2 P_s + \sigma_r^2} \ .$$

We can re-write (1), (2) and (3) into the following compact matrix form

$$\mathbf{Y} = \mathbf{H} x_s + \mathbf{B}\mathbf{Z} \ ,$$

where

$$\mathbf{Y} = \begin{bmatrix} y_{sd} \\ y_{rd} \end{bmatrix}, \ \mathbf{H} = \begin{bmatrix} h_{sd} \\ \alpha_r h_{rd} h_{sr} \end{bmatrix},$$
$$\mathbf{B} = \begin{bmatrix} 0 & 1 & 0 \\ \alpha_r h_{rd} & 0 & 1 \end{bmatrix}, \ \text{and } \mathbf{Z} = \begin{bmatrix} z_r \\ z_d \\ z_d \end{bmatrix}. \tag{4}$$

It has been shown in [6] that, the above channel, which combines both the direct path (s to d) and relay path (s to r to d), can be modeled as a one-input, two-output complex Gaussian noise channel. The capacity $C_{\text{AF}}(s, r, d)$ from s to d can be given by

$$C_{\text{AF}}(s,r,d) = \frac{W}{2}\log_2[\det(\mathbf{I} + (P_s\mathbf{H}\mathbf{H}^\dagger)(\mathbf{B}E[\mathbf{Z}\mathbf{Z}^\dagger]\mathbf{B}^\dagger)^{-1})], \tag{5}$$

where W is the bandwidth, $\det(\cdot)$ is the determinant function, $\mathbf{I}$ is the identity matrix, the superscript "$\dagger$" represents the complex conjugate transposition, and $E[\cdot]$ is the expectation function.

After putting (4) into (5) and performing algebraic manipulations, we have $C_{\text{AF}}(s,r,d) = \frac{W}{2}\log_2\left(1 + \frac{P_s}{\sigma_d^2}|h_{sd}|^2 + \frac{P_s|h_{sr}|^2 P_r|h_{rd}|^2}{P_s\sigma_d^2|h_{sr}|^2 + P_r\sigma_r^2|h_{rd}|^2 + \sigma_r^2\sigma_d^2}\right)$. Denote $\text{SNR}_{sd} = \frac{P_s}{\sigma_d^2}|h_{sd}|^2$, $\text{SNR}_{sr} = \frac{P_s}{\sigma_r^2}|h_{sr}|^2$, and $\text{SNR}_{rd} = \frac{P_r}{\sigma_d^2}|h_{rd}|^2$. We have

$$C_{\text{AF}}(s,r,d) = W \cdot I_{\text{AF}}(\text{SNR}_{sd}, \text{SNR}_{sr}, \text{SNR}_{rd}) \ ,$$

where $I_{\text{AF}}(\text{SNR}_{sd}, \text{SNR}_{sr}, \text{SNR}_{rd}) = \frac{1}{2}\log_2\left(1 + \text{SNR}_{sd} + \frac{\text{SNR}_{sr}\cdot\text{SNR}_{rd}}{\text{SNR}_{sr}+\text{SNR}_{rd}+1}\right)$.

Decode-and-Forward (DF) Under this mode, relay node r decodes and estimates the received signal from source node s in the first time slot, then transmits the estimated data to destination node d in the second time slot. The capacity for DF under the two time-slot structure is given by [6]

$$C_{\text{DF}}(s,r,d) = W \cdot I_{\text{DF}}(\text{SNR}_{sd}, \text{SNR}_{sr}, \text{SNR}_{rd}) \ ,$$

where $I_{\text{DF}}(\text{SNR}_{sd}, \text{SNR}_{sr}, \text{SNR}_{rd}) = \frac{1}{2}\min\{\log_2(1+\text{SNR}_{sr}), \log_2(1+\text{SNR}_{sd}+\text{SNR}_{rd})\}$.

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 capacity under either mode, both source node and relay node should transmit at the maximum power. In this paper, we let $P_s = P_r = P$.

Direct Transmissions When cooperative communications (i.e., relay node) are not used, source node s transmits to destination node d in both time slots. The capacity from node s to node d is

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

Based on the above results, we have two observations. First, comparing C_{AF} (or C_{DF}) to C_{D}, it is hard to say cooperative communications is always better than direct transmissions. In fact, a poor choice of relay node could make the capacity under cooperative communications be smaller than direct transmissions. This fact underlines the significance of relay node selection in cooperative communications. Second, although AF and DF are different mechanisms, the capacities for both of them have the same form, i.e., a function of SNR_{sd}, SNR_{sr}, and SNR_{rd}. Therefore, a relay node assignment algorithm designed for AF can be easily extended for DF. In this paper, we develop a relay node assignment algorithm for AF, which can also be used should DF is employed.

3. THE RELAY NODE ASSIGNMENT PROBLEM

Based on the background in the last section, we consider relay node assignment problem in a network setting. There are N nodes in an ad hoc network, with each node being either a source node, a destination node, or a potential relay node (see Fig. 2). In order to avoid interference, we assume that orthogonal channels are available in the network (e.g., using OFDMA), which is used for cooperative communications [6]. The path loss between nodes u and v is captured in h_{uv} and is given a priori. The discussion of channel measurement techniques is beyond the scope of this paper.

Denote $\mathcal{N}_s = \{s_1, s_2, \cdots, s_{N_s}\}$ the set of source nodes, $\mathcal{N}_d = \{d_1, d_2, \cdots, d_{N_d}\}$ the set of destination nodes, and $\mathcal{N}_r = \{r_1, r_2, \cdots, r_{N_r}\}$ the set of relays (see Fig. 2). We consider unicast where every source node s_i is paired with a destination node d_i, i.e., $N_d = N_s$. Each node is equipped with a single transceiver and can transmit/receive within one channel at a time. Further, we assume

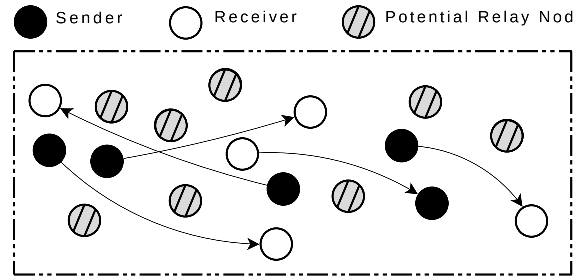

Figure 2: An ad hoc network consisting of source nodes, destination nodes, and relay nodes.

that each node can only serve a unique role of source, destination, or relay. That is, $N_r + 2N_s = N$.

Note that a source node may not always get a relay node. This is because there may not be sufficient number of relay nodes in the network (e.g., $N_r < N_s$). Even if there are enough relay nodes, a sender may still not use a relay node if it leads to a smaller capacity than direct transmissions (see discussion at the end of Section 2).

We now discuss the objective function of our problem. Although different objectives can be used, a widely-used objective for cooperative communications is capacity. For the multi-pair network environment considered in this paper (see Fig. 2), each source-destination pair will have a different capacity after we apply a relay node assignment algorithm. So a plausible objective is to maximize the minimum capacity among all the source-destination pairs.

More formally, denote $\mathcal{R}(s_i)$ the relay node assigned to s_i. For both AF and DF, its capacity can be written as (see Section 2)

$$W I_{\text{R}}(\text{SNR}_{s_i,d_i}, \text{SNR}_{s_i,\mathcal{R}(s_i)}, \text{SNR}_{\mathcal{R}(s_i),d_i}) \ ,$$

with $I_{\text{R}}(\cdot) = I_{\text{AF}}(\cdot)$ for AF and $I_{\text{R}}(\cdot) = I_{\text{DF}}(\cdot)$ for DF. In the case that s_i does not use a relay, we denote $\mathcal{R}(s_i) = \emptyset$ and the capacity is the direct transmission capacity, i.e.,

$$C_{\text{R}}(s_i, \emptyset) = C_{\text{D}}(s_i, d_i) \ .$$

Combining both cases, we have

$$C_{\text{R}}(s_i,\mathcal{R}(s_i)) = \begin{cases} W I_{\text{R}}(\text{SNR}_{s_i,d_i}, \text{SNR}_{s_i,\mathcal{R}(s_i)}, \text{SNR}_{\mathcal{R}(s_i),d_i}) & \\ & \text{if } \mathcal{R}(s_i) \neq \emptyset, \\ W\log_2(1+\text{SNR}_{s_i,d_i}) & \text{if } \mathcal{R}(s_i) = \emptyset. \end{cases} \quad (6)$$

Note that we do not list d_i in function $C_{\text{R}}(s_i, \mathcal{R}(s_i))$ since for each source node s_i, the corresponding destination node d_i is deterministic.

Denote C_{min} the minimum capacity among all source nodes. That is,

$$C_{\text{min}} = \min\{C_{\text{R}}(s_i, \mathcal{R}(s_i)) : s_i \in \mathcal{N}_s\}.$$

Our objective is to find an optimal relay node assignment for all the source-destination pairs such that C_{min} is maximized.

4. AN OPTIMAL RELAY ASSIGNMENT ALGORITHM

We can formulate the relay node assignment problem as an integer linear program. It is important to note here that an integer linear programming problem is NP-hard *in general*, i.e., there does not exist a general polynomial-time solution procedure to solve *every* integer linear program. However, as we show in this paper, we can exploit problem specific properties, and design a polynomial-time solution for our specific problem. The main contribution of this paper is a polynomial-time algorithm to the relay node assignment problem, which we will present in this section.

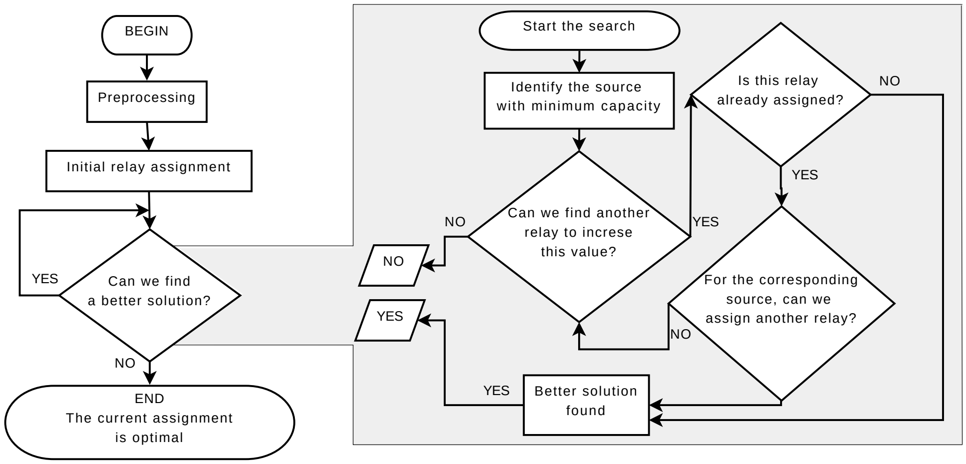

Figure 3: The flow chart of ORA algorithm.

4.1 Basic Idea

The algorithm we will present is called *Optimal Relay Assignment* (ORA) algorithm. Figure 3 shows the flow chart of ORA algorithm.

Initially, ORA algorithm starts with a random feasible relay node assignment. By feasible, we mean that each source-destination pair can be assigned at most one relay node and that a relay node can be assigned only once. Such initial feasible assignment is easy to construct, e.g., direct transmission between each source-destination pair (without the use of a relay) is a special case of feasible assignment.

Starting with this initial assignment, ORA adjusts the assignment during each iteration, with the goal of increasing the objective function C_{min}. Specifically, during each iteration, ORA identifies the source node that corresponds to C_{min}. Then, ORA helps this source node to search a better relay such that this "bottleneck" capacity can be increased. In the case that the selected relay is already assigned to another source node, further relay node adjustment on that source node is necessary (so that its current relay can be released). Such adjustment may have a chain effect on a number of source nodes in the network. It is important that for any adjustment on a relay node, the affected source node should always maintain a capacity larger than C_{min}. There are only two outcomes from such search in an iteration: (i) a better assignment is found, in which case, ORA moves on to the next iteration; or (ii) a better assignment cannot be found, in which case, ORA terminates.

There are two key technical challenges we aim to address in the design. First, for any non-optimal solution, the algorithm should be able to find a better solution. As a result, upon termination, the final assignment is optimal. Second, its running time must be polynomial. We will show that ORA addresses both problems successfully. Specifically, we show the complexity of ORA algorithm is polynomial in Section 4.4. We will also give a correctness proof of its optimality upon termination in Section 5.

4.2 Algorithm Details

In the beginning, ORA algorithm performs a "preprocessing" step. In this step, for each source-destination pair, the source node s_i considers each relay node r_j in the network and computes the corresponding capacity $C_{\text{R}}(s_i, r_j)$ by (6). Each source node s_i also computes the capacity $C_{\text{R}}(s_i, \emptyset)$ by (6) under direct transmissions (i.e., without the use of a relay node). After these computations, each source node s_i can identify those relay nodes that can offer an increase of its capacity compared to direct transmissions,

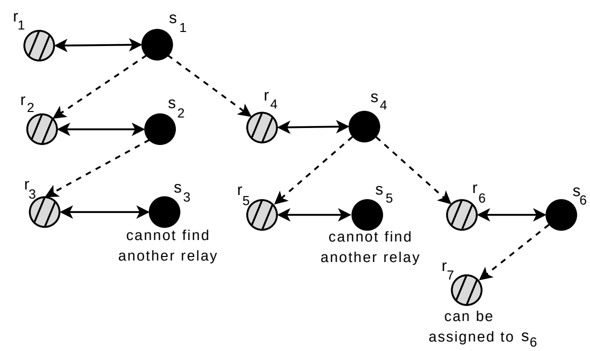

Figure 4: A sample tree topology in ORA algorithm for finding a better solution.

i.e., those relays with $C_{\text{R}}(s_i, r_j) > C_{\text{R}}(s_i, \emptyset)$. Obviously, it only makes sense to consider these relays for cooperative communications. In the case that no relay can offer any increase of capacity compared to direct transmissions, we will just employ direct transmissions for these source nodes.

After the preprocessing step, we enter the initial assignment step. The objective of this step is to obtain an initial feasible solution for ORA algorithm so that it can start its iteration. In the preprocessing step, we have already identified, for each source node, the list of relay nodes that can increase capacity compared to direct transmissions. We can randomly assign a relay node from this list to a source node. Note that once a relay node is assigned to a source node, it cannot be assigned again to another source node. Thus, if the selected relay node is already assigned to other source node, then this source node will simply employ direct transmissions, i.e., without the use of a relay. Upon the completion of this assignment, each source node will have a capacity no less than that under direct transmissions.

The next step in ORA algorithm is finding a better assignment, which represents an iteration process. This is the key step in ORA algorithm. The detail of this step is shown in the right hand side of Fig. 3. As a starting point of this step, ORA algorithm identifies the smallest capacity C_{min} among all sources. ORA algorithm aims to increase this minimum capacity for the corresponding source node, while keeping all other source nodes to have their capacities stay above C_{min}. Without loss of generality, we use Fig. 4 to illustrate a search process.

- Suppose ORA identifies that s_1 has the smallest capacity C_{min} under the current assignment (with relay node r_1). Then s_1 examines other relays with a capacity larger than C_{min}. If it cannot find such a relay, then no better solution is found and the ORA algorithm is completed.

- Otherwise, i.e., there are better relays, we consider these relays in the *non-increasing* order in terms of achieved capacity (should it be assigned to s_1). That is, we try the relay that can offer the maximum possible increase in capacity first.

- Suppose that source node s_1 considers relay node r_2.[1] If this relay node is not yet assigned to any other source node, then r_2 can be immediately assigned to s_1. In this simple case, we find a better solution and the current iteration is completed.

- Otherwise, i.e., r_2 is already assigned to a source node, say s_2, we mark r_2 to indicate that r_2 is "under consideration" and check whether r_2 can be released by s_2.

- To release r_2, source node s_2 needs to find another relay (or use direct transmissions) while making sure that such new assignment still makes its capacity larger than C_{min}. This process is the same as what we have done at s_1, with the only (but important) difference that s_2 will not consider a relay that is already "marked", as that relay node has already been considered by a source node encountered earlier in the search process of this iteration.

- Suppose that source node s_2 now considers relay r_3. If this relay node is not yet assigned to any source node, then r_3 can be assigned to s_2; r_2 can be assigned to s_1; and the current iteration is completed. If the relay being considered is r_1 (or $\emptyset$), then a better solution, where r_1 (or $\emptyset$) is assigned to s_2 and r_2 is assigned to s_1, is found and the current iteration is completed. Otherwise, we mark r_3 and check further to see whether r_3 can be released by its corresponding source node, say s_3.

- Suppose that s_3 cannot find any "unmarked" relay that offers a capacity larger than C_{min} and its capacity under direct transmissions is not larger than C_{min}. Then s_2 cannot use r_3 as its relay.

- If any "unmarked" relay that has a capacity larger than C_{min} cannot be assigned to s_2, then s_1 cannot use r_2 and will move on to consider the next relay on its non-increasing capacity list, say r_4.

- The search continues, with relay nodes being marked along the way, until a better solution is found or no better solution can be found. For example, in Fig. 4, s_6 finds a new relay r_7. As a result, we have a new assignment, where r_7 is assigned to s_6; r_6 is assigned to s_4; and r_4 is assigned to s_1.

Note that the "mark" on a relay node will not be cleared throughout the search process in the same iteration. We call this "linear marking" mechanism. These marks will only be cleared when the current iteration terminates and before the start of the next iteration. The pseudo-code of ORA algorithm is shown in Fig. 5.

It should be clear that ORA works regardless of whether $N_r \geq N_s$ or $N_r < N_s$. For the latter case, i.e., the number of relay nodes in the network is less than the number of source nodes, it is only necessary to consider relay node assignment for a reduced subset of N_r source nodes, where the capacity of each source in this subset under direct transmissions is less than the capacity of those $(N_s - N_r)$ source nodes not in this subset. As a result, in the case of $N_s > N_r$, ORA will run even more efficiently due to a smaller problem size.

[1]We note that r_2 cannot be $\emptyset$. Due to the preprocessing step, we have $C_{\text{R}}(s_1, r_1) \geq C_{\text{R}}(s_1, \emptyset)$. As r_2 has $C_{\text{R}}(s_1, r_2) > C_{\text{min}} = C_{\text{R}}(s_1, r_1)$, we have $C_{\text{R}}(s_1, r_2) > C_{\text{R}}(s_1, \emptyset)$. Thus, $r_2 \neq \emptyset$.

Main algorithm

1. Perform preprocessing and an initial relay node assignment.
2. Set all the relay nodes in the network as "unmarked".
3. Determine the objective value C_{min}, the smallest capacity among all source nodes.
4. Suppose that source node s_i with its relay $\mathcal{R}(s_i)$ has the smallest capacity.
5. For every node r_j with $C_{\text{R}}(s_i, r_j) > C_{\text{min}}$, do the following in the non-increasing order of $C_{\text{R}}(s_i, r_j)$.
6.   Run Check_Relay_Availability(r_j, C_{min}).
7.   If r_j is available, then do the following.
8.   Remove $\mathcal{R}(s_i)$'s assignment to s_i.
9.   Assign relay r_j to s_i and go to line 2.
10.   Otherwise, continue on to next r_j and go to line 6.
11. If no relay is available, then the algorithm terminates.

Subroutines

Check_Relay_Availability(r_j, C_{min}):

12. If r_j is not assigned to any source node, then r_j is available.
13. If $r_j = \mathcal{R}(s_i)$ or $r_j = \emptyset$, then r_j is available.
14. Otherwise,
15.   Set r_j as "marked".
16.   Run Find_Another_Relay ($\mathcal{S}(r_j), r_j, C_{\text{min}}$), where $\mathcal{S}(r_j)$ is the source node that uses r_j.
17.   If $\mathcal{S}(r_j)$ can find another relay, then r_j is available.
18.   Otherwise r_j is unavailable.

Find_Another_Relay($\mathcal{S}(r_j), r_j, C_{\text{min}}$):

19. For every "unmarked" relay r_k with $C_{\text{R}}(\mathcal{S}(r_j), r_k) > C_{\text{min}}$, do the following in the non-increasing order of $C_{\text{R}}(\mathcal{S}(r_j), r_k)$.
20.   Run Check_Relay_Availability(r_k, C_{min}).
21.   If r_k is available, then do the following.
22.   Remove relay node r_j's assignment to $\mathcal{S}(r_j)$.
23.   Assign relay node r_k to $\mathcal{S}(r_j)$, i.e., $\mathcal{S}(r_j)$ finds another relay.
24.   Otherwise, continue on to next r_k and go to line 20.
25. If all the relays are unavailable, then $\mathcal{S}(r_j)$ cannot find another relay.

Figure 5: Optimal Relay Assignment (ORA) algorithm.

4.3 Caveat on the Marking Mechanism

We now re-visit the marking mechanism in ORA algorithm. Although different marking mechanisms may be designed to achieve the optimal objective, the algorithm complexity under different marking mechanisms may differ significantly. In this section, we first present a marking mechanism, which appears to be a natural approach but leads to exponential complexity for each iteration. Then we re-examine our marking mechanism and show that it leads to a linear complexity for each iteration.

A natural approach is to perform both marking and unmarking within an iteration. This approach is best explained with an example. Again, let's look at Fig. 4. Source node s_1 first considers r_2. Since r_2 will be used by s_1 in the new solution, r_2 is marked. Source node s_2 considers r_3, which is already assigned to s_3. Since s_3 cannot release r_3 without reducing its capacity below the current C_{min}, this branch of search is futile and s_1 now considers a different relay node r_4. Since r_4 is currently assigned to s_4, we try to find a new relay for s_4. Now the question is: shall we remove those marks on r_2 and r_3 that we put earlier in the process within this iteration? Under this natural approach, r_2 and r_3 should be unmarked so that they can be considered as candidate relay nodes for s_4 in its search. Similarly, when we try to find a relay for s_6, relay nodes r_2, r_3, and r_5 should be unmarked so that they can be considered as candidate relay nodes for s_6, in addition to r_7. In summary, under this approach, each relay node that has been considered earlier in the search process by a source node should be unmarked when this source node considers the next relay node, so that this relay node can remain in the pool of candidate relay nodes to be considered in the search process. It is not hard to show that such marking/unmarking mechanism considers all possible assignments and can guarantee to find a better solution (if it exists). However, the complexity of such approach is exponential for each iteration.

In contrast, under ORA algorithm, there is no unmarking mechanism within an iteration. That is, relay nodes that are marked earlier in the search process by some source nodes will remain marked. For example, in Fig. 4, when s_4 tries to find another relay, it will no longer consider r_2 and r_3 that have been marked earlier. Similarly, when s_6 tries to find another relay, it will not consider r_2, r_3, r_4, and r_5. As a result, any relay node will be considered at most once in the search process, which leads to a linear complexity for each iteration of ORA algorithm. Unmarking for all nodes is performed only upon the termination of an iteration.

An immediate question on our marking mechanism is that how such a "linear marking" can lead to an optimal solution, as it appears that many possible assignments that may increase C_{min} are not considered. This is precisely the question that we will address in Section 5, where we will prove that ORA can guarantee that its final solution is optimal (Theorem 1).

4.4 Complexity Analysis

We now analyze the computational complexity of ORA algorithm. Most computations in ORA are for iteratively finding a better solution. During each iteration, due to the "linear marking" mechanism in our algorithm, a relay node is checked for its availability at most once. Thus, the complexity of each iteration is $O(N_r)$. Now we examine the number of iterations that ORA will execute. For each source node, the number of possible capacities is $(N_r + 1)$. Thus, the total number of possible capacities (i.e., objective values) among all the source nodes in the network is $O(N_s(N_r + 1))$. Since the objective value is increased at each iteration (except the last iteration), the number of iterations is $O(N_s(N_r + 1))$. So the overall complexity of all the iterations is $O(N_s(N_r + 1) \cdot N_r) = O(N_s N_r^2)$.

4.5 An Example

We now use an example to illustrate the operation of the ORA algorithm, in particular, its "linear marking" mechanism. Suppose that there are four source-destination pairs and six relay nodes in the network. Table 1(a) shows the capacity for each source node s_i when relay node r_j is assigned to it. The symbol $\emptyset$ indicates direct transmissions, i.e., without the use of a relay node. Also shown in Table 1(a) is an initial relay node assignment, which is indicated by an underscore on the intersecting row (s_i) and column (r_j). That is, the initial assignment is r_5 for s_1, r_4 for s_2, r_3 for s_4, r_2 for s_3. Note that the preprocessing step before the initial assignment ensures that the capacity for each source-destination pair by the initial assignment is no less than direct transmissions.

Under the initial relay node assignment in Table 1(a), source s_3 is identified as having the smallest capacity of 13, which is the current value of C_{min}. Since consideration of relay nodes is performed in the order of non-increasing capacity for the source node under consideration, r_4 is therefore considered for s_3 (as it offers the largest capacity among all relay nodes for s_3). But r_4 is already assigned to source node s_2, so r_4 is "marked". Now s_2 needs to find another relay. But, any other relay (or direct transmissions) will result in a capacity no greater than the current objective value $C_{\text{min}} = 13$. This means that r_4 cannot be taken away from s_2. Since r_4 does not work out for s_3, s_3 will then consider the relay node that offers the second largest capacity, i.e., relay node r_3. Since r_3 is already assigned to sender s_4, r_3 will be "marked". Now, ORA algorithm checks to see if s_4 can find another relay.

Table 1: An example illustrating the operation of ORA algorithm.

(a) Initial relay node assignment.

	$\emptyset$	r_1	r_2	r_3	r_4	r_5	r_6
s_1	14	7	24	5	15	$\underline{15}$	25
s_2	9	8	10	11	$\underline{20}$	10	12
$\rightarrow s_3$	11	10	$\underline{13}$	17	18	8	9
s_4	10	9	16	$\underline{19}$	24	9	13

(b) Assignment after the first iteration.

	$\emptyset$	r_1	r_2	r_3	r_4	r_5	r_6
$\rightarrow s_1$	14	7	24	5	15	$\underline{15}$	25
s_2	9	8	10	11	$\underline{20}$	10	12
s_3	11	10	13	$\underline{17}$	18	8	9
s_4	10	9	$\underline{16}$	19	24	9	13

(c) Final assignment upon algorithm termination.

	$\emptyset$	r_1	r_2	r_3	r_4	r_5	r_6
s_1	14	7	24	5	15	15	$\underline{25}$
s_2	9	8	10	11	$\underline{20}$	10	12
s_3	11	10	13	$\underline{17}$	18	8	9
$\rightarrow s_4$	10	9	$\underline{16}$	19	24	9	13

Now s_4 checks relay nodes in non-increasing order of capacity. Since both r_4 (with the largest capacity) and r_3 (with the second largest capacity) are marked, they will not be considered. The relay with the third large capacity is r_2, which is unmarked. Relay r_2 offers a capacity of 16, which is greater than $C_{\text{min}} = 13$. So s_4 will choose r_2. The new assignment after the first iteration is shown in Table 1(b).

Now the objective value, C_{min}, is updated to 15, which corresponds to s_1. Before the second iteration, all markings done in the first iteration are cleared. In the second iteration, ORA algorithm will perform a new search of relay node for s_1 with the aim that after some relay node re-assignment on other source nodes, they all have a capacity larger than 15.

The iteration continues and the final assignment upon termination of ORA algorithm is shown in Table 1(c), with the optimal (maximum) value of C_{min} being 16.

5. PROOF OF OPTIMALITY

In this section, we give a correctness proof of ORA algorithm, that is, upon the termination of ORA algorithm, the solution (i.e., objective value and the corresponding relay node assignment) is optimal.

Our proof is based on contradiction. Denote ψ the final solution obtained by the ORA algorithm, with the objective value being C_{min}. For ψ, denote the relay node assigned to source node s_i as $\mathcal{R}(s_i)$. Conversely, for ψ, denote the source node that uses relay node r_j as $\mathcal{S}(r_j)$.

Assume there exists a better solution $\hat{\psi}$ than ψ. That is, the objective value by $\hat{\psi}$, denoted as $\hat{C}_{\text{min}}$, is greater than that by ψ, i.e., $\hat{C}_{\text{min}} > C_{\text{min}}$. For $\hat{\psi}$, denote the relay node assigned to source node s_i as $\hat{\mathcal{R}}(s_i)$. Conversely, for $\hat{\psi}$, denote the source node that uses relay node r_j as $\hat{\mathcal{S}}(r_j)$.

The key idea in the proof is to exploit the marking status at the end of the last iteration of ORA algorithm. Now let's take a close look at this last iteration. During this iteration, ORA attempts to find a better solution but concludes that it cannot find any, and thus the algorithm terminates. So this last iteration is the only "non-improving" iteration for the objective value. At the end of this last iteration, assume that s_i, with its assigned relay node $\mathcal{R}(s_i)$, is the "bottleneck" source node, i.e., $C_{\text{min}} = C_{\text{R}}(s_i, \mathcal{R}(s_i))$. Then we have the following fact for the marking status of $\mathcal{R}(s_i)$.

FACT 1. *For the bottleneck source node s_i under ψ, its relay node $\mathcal{R}(s_i)$ is not marked at the end of the last iteration of ORA algorithm.*

PROOF. In the last iteration of ORA algorithm, a relay node r_j is marked only if it has been checked for its availability and it is not $\mathcal{R}(s_i)$ (see Check_Relay_Availability() in Fig. 5). Thus, $\mathcal{R}(s_i)$ cannot be marked at the end of the last iteration of the ORA algorithm. □

Fact 1 will be a basis for contradiction in our proof for Theorem 1, the main result of this section. But first, we present the following three claims, which *recursively* examine relay node assignment under $\hat{\psi}$.

First, for the relay node assigned to s_i in $\hat{\psi}$, i.e., $\hat{\mathcal{R}}(s_i)$, we have the following claim.

CLAIM 1. *Relay node $\hat{\mathcal{R}}(s_i)$ cannot be $\emptyset$ and must be assigned to some source node under solution ψ. Further, it must be marked at the end of the last iteration of the ORA algorithm.*

PROOF. The proof for the first statement is based on contradiction. Suppose that $\hat{\mathcal{R}}(s_i) = \emptyset$ or relay node $\hat{\mathcal{R}}(s_i)$ is not assigned to any source node under solution ψ. Since $\hat{\psi}$ is a better solution than ψ, we have $C_{\text{R}}(s_i, \hat{\mathcal{R}}(s_i)) \geq \hat{C}_{\text{min}} > C_{\text{min}}$. Thus, in the last iteration of the ORA algorithm, we should check $\hat{\mathcal{R}}(s_i)$'s availability and a better solution should be found. However, we know that this last iteration is a non-improving iteration and the ORA algorithm cannot find a better solution. So, $\hat{\mathcal{R}}(s_i)$ cannot be $\emptyset$ and must be assigned to some source node under ψ.

We now prove the second statement. Since $C_{\text{R}}(s_i, \hat{\mathcal{R}}(s_i)) > C_{\text{min}}$, we should check $\hat{\mathcal{R}}(s_i)$'s availability in the last iteration of the ORA algorithm. Since ORA algorithm cannot find a better solution in this last iteration, $\hat{\mathcal{R}}(s_i)$ should be marked and then the result for checking $\hat{\mathcal{R}}(s_i)$'s availability must be unavailable. Thus, $\hat{\mathcal{R}}(s_i)$ must be marked at the end of ORA algorithm. □

Claim 1 states that in solution ψ, relay node $\hat{\mathcal{R}}(s_i)$ must be assigned to some source node. By the definition of $\mathcal{S}(\cdot)$, we have that $\hat{\mathcal{R}}(s_i)$ is assigned to source node $\mathcal{S}(\hat{\mathcal{R}}(s_i))$ in solution ψ. To simplify notation, define function $\mathcal{G}(\cdot)$ as

$$\mathcal{G}(\cdot) = \mathcal{S}(\hat{\mathcal{R}}(\cdot)) \ . \tag{7}$$

Thus, relay node $\hat{\mathcal{R}}(s_i)$ is assigned to source node $\mathcal{G}(s_i)$ in ψ (see the top portion of Fig. 6).

Now we recursively investigate the relay node assigned to $\mathcal{G}(s_i)$ under solution $\hat{\psi}$, i.e., $\hat{\mathcal{R}}(\mathcal{G}(s_i))$. We have the following claim (also see Fig. 6). Its proof follows the same token as that for Claim 1 and is omitted to conserve page length.

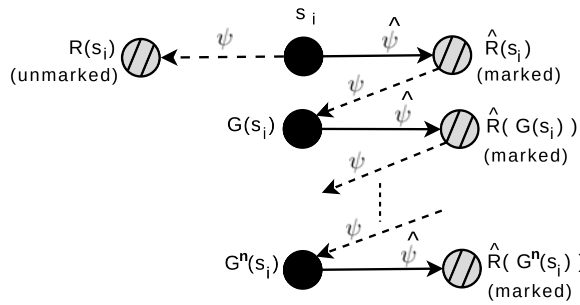

Figure 6: Marking status and the relationship among the nodes being examined in the proof of Theorem 1.

CLAIM 2. *Relay node $\hat{\mathcal{R}}(\mathcal{G}(s_i))$ cannot be $\emptyset$ and must be assigned to some source node under solution ψ. Further, it must be marked at the end of the last iteration of the ORA algorithm.*

Claim 2 states that in solution ψ, relay node $\hat{\mathcal{R}}(\mathcal{G}(s_i))$ must be assigned to some source node. By the definition of $\mathcal{S}(\cdot)$, we have that $\hat{\mathcal{R}}(\mathcal{G}(s_i))$ is assigned to source node $\mathcal{S}(\hat{\mathcal{R}}(\mathcal{G}(s_i)))$ in solution ψ. By (7), we have $\mathcal{S}(\hat{\mathcal{R}}(\mathcal{G}(s_i))) = \mathcal{G}(\mathcal{G}(s_i))$. To simply the notation, denote function $\mathcal{G}^2(\cdot)$ as

$$\mathcal{G}^2(\cdot) = \mathcal{G}(\mathcal{G}(\cdot)) \ .$$

Thus, relay node $\hat{\mathcal{R}}(\mathcal{G}(s_i))$ is assigned to source node $\mathcal{G}^2(s_i)$ in ψ.

Following the same token for Claims 1 and 2, we can obtain a similar claim for the relay node assigned to $\mathcal{G}^2(s_i)$ under $\hat{\psi}$, i.e., $\hat{\mathcal{R}}(\mathcal{G}^2(s_i))$. This recursive investigation of source node of $\hat{\mathcal{R}}(\cdot)$ under ψ and relay node assigned to $\mathcal{G}(\cdot)$ under $\hat{\psi}$ continues and will terminate at some n-th step since the numbers of source and relay nodes are finite (see Fig. 6).

Denote

$$\mathcal{G}^0(s_i) = s_i \ ,$$
$$\mathcal{G}^k(s_i) = \mathcal{G}(\mathcal{G}^{k-1}(s_i)) \quad (1 \leq k \leq n) \ .$$

Thus, we have that in $\hat{\psi}$, the corresponding relay nodes for $s_i, \mathcal{G}(s_i)$, $\cdots, \mathcal{G}^n(s_i)$ are $\hat{\mathcal{R}}(s_i), \hat{\mathcal{R}}(\mathcal{G}(s_i)), \cdots, \hat{\mathcal{R}}(\mathcal{G}^n(s_i))$, respectively (see Fig. 6). We can prove one claim for each of these relay nodes. In summary, we have the following claim.

CLAIM 3. *Relay node $\hat{\mathcal{R}}(\mathcal{G}^k(s_i))$ cannot be $\emptyset$ and must be assigned to some source node under solution ψ, $k = 0, 1, 2, \cdots, n$. Further, it must be marked at the end of the last iteration of the ORA algorithm.*

Note that we already have Claims 1 and 2 for $k = 0$ and $k = 1$, respectively. The proof for the general case in this claim follows the same token and is omitted to conserve page length.

We are now ready to prove the following theorem, which is the main result of this section.

THEOREM 1. *Upon the termination of the ORA algorithm, the obtained solution ψ is optimal.*

PROOF. Referring to Fig. 6, we have Claim 3 for a set of relay nodes $\hat{\mathcal{R}}(s_i), \hat{\mathcal{R}}(\mathcal{G}(s_i)), \cdots, \hat{\mathcal{R}}(\mathcal{G}^n(s_i))$.

Under Claim 3, $\hat{\mathcal{R}}(\mathcal{G}^n(s_i))$ is assigned to a source node in solution ψ. We now investigate to which source node it is assigned in ψ. This source node must be a node among $\{s_i, \mathcal{G}(s_i), \cdots, \mathcal{G}^n(s_i)\}$, otherwise the recursive process will not terminate at $\hat{\mathcal{R}}(\mathcal{G}^n(s_i))$.

Table 2: Optimal assignments for Case 1 ($N_r \geq N_s$) under two different initial relay node assignments.

Sender	Direct Transmission Capacity (Mbps)	Relay Assignment I			Relay Assignment II		
		Initial	Final	Final Capacity (Mbps)	Initial	Final	Final Capacity (Mbps)
s_1	6.1	r_8	r_2	15.3	r_1	r_2	15.3
s_2	10.6	r_6	r_7	21.8	r_7	r_7	21.8
s_3	8.8	r_1	r_1	16.6	r_8	r_3	18.9
s_4	6.3	r_5	r_4	$\boxed{10.7}$	r_2	r_4	$\boxed{10.7}$
s_5	7.3	r_{10}	r_{14}	14.9	r_{13}	r_{12}	15.0
s_6	9.6	r_4	r_6	21.3	r_5	r_5	11.0
s_7	$\boxed{4.2}$	r_2	r_8	11.0	r_{10}	r_8	11.0
s_8	6.9	r_{16}	r_{12}	16.6	r_6	r_{10}	11.0
s_9	11.3	r_7	r_{10}	22.6	r_9	r_9	11.9
s_{10}	11.1	r_{19}	r_{17}	19.9	$\emptyset$	$\emptyset$	11.1
s_{11}	9.5	r_{17}	r_{20}	21.0	r_{16}	r_{20}	21.1
s_{12}	7.4	r_{18}	r_{18}	12.8	r_{19}	r_{19}	13.6
s_{13}	8.5	r_{20}	r_{16}	11.9	r_{18}	r_{18}	11.2
s_{14}	9.7	r_{15}	r_{15}	12.2	r_{17}	r_{16}	18.1
s_{15}	6.1	$\emptyset$	r_{19}	13.5	r_{20}	r_{17}	11.2

But under ψ, each of $\mathcal{G}(s_i)$, $\mathcal{G}^2(s_i)$, $\mathcal{G}^3(s_i)$, $\cdots$, $\mathcal{G}^n(s_i)$ has its own relay $\hat{\mathcal{R}}(s_i)$, $\hat{\mathcal{R}}(\mathcal{G}(s_i))$, $\hat{\mathcal{R}}(\mathcal{G}^2(s_i))$, $\cdots$, $\hat{\mathcal{R}}(\mathcal{G}^{n-1}(s_i))$, respectively. Thus, $\hat{\mathcal{R}}(\mathcal{G}^n(s_i))$ can only be assigned to s_i in solution ψ. On the other hand, relay $\mathcal{R}(s_i)$ is assigned to s_i in solution ψ. So we must have $\hat{\mathcal{R}}(\mathcal{G}^n(s_i)) = \mathcal{R}(s_i)$.

However, by Claim 3, we have $\hat{\mathcal{R}}(\mathcal{G}^n(s_i))$ is marked, while by Fact 1, we have $\mathcal{R}(s_i)$ is not marked. This is a contradiction. Thus the assumption that there exists a better solution $\hat{\psi}$ than ψ does not hold and the proof is complete. $\square$

Note that the proof of Theorem 1 does not depend on the initial assignment in the ORA algorithm. So we have the following important property for the ORA algorithm.

COROLLARY 1.1. *Under any initial relay node assignment, the ORA algorithm can find an optimal relay node assignment.*

6. NUMERICAL RESULTS

In this section, we present some numerical results to demonstrate the properties of the ORA algorithm.

6.1 Simulation Setting

We consider a 50-node ad hoc network. For this network, we consider both the cases of $N_r \geq N_s$ and $N_r < N_s$. In the first case, we have 15 source-destination pairs and 20 relay nodes. While in the second case, we have 20 source-destination pairs and only 10 relay nodes. The role of each node (either as a source, destination, or relay) for each case is shown in Figs. 7 and 9, respectively.

For the simulations, we assume $W = 22$ MHz bandwidth for each channel. The maximum transmission power at each node is set to 1 W. Each relay works on AF mode. For simplicity, we assume that h_{sd} only includes the path loss component between nodes s and d and is given by $|h_{sd}|^2 = ||s - d||^{-4}$, where $||s - d||$ is the distance (in meters) between these two nodes and 4 is the path loss index. For the AWGN channel, we assume the variance of noise is 10^{-10} W at all nodes.

6.2 Results

Case 1: $N_r \geq N_s$**.** In this case (see Fig. 7), we have 15 source-destination pairs and 20 relay nodes.

Under ORA, after preprocessing, we start with an initial relay node assignment in the first iteration. Such initial assignment is not

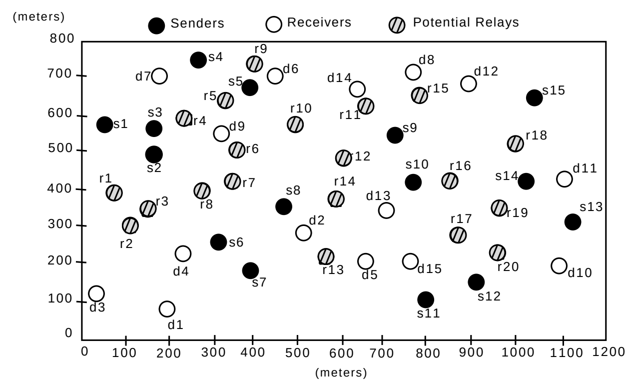

Figure 7: A 50-node network topology for Case 1 ($N_r \geq N_s$), with $N_s = 15$ and $N_r = 20$.

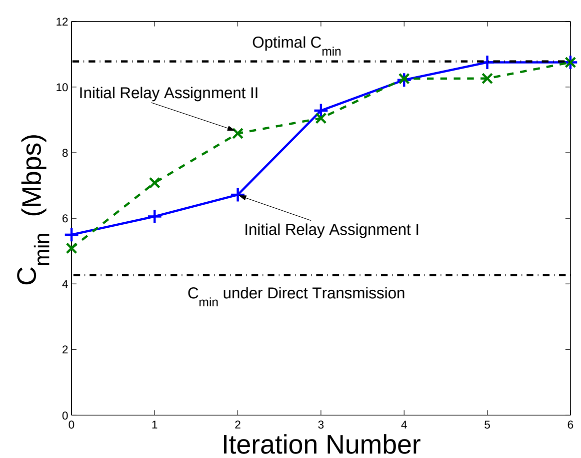

Figure 8: Case 1 ($N_r \geq N_s$): The objective value C_{min} at each iteration of the ORA algorithm under two different initial relay node assignments.

Table 3: Optimal assignments for Case 2 ($N_r < N_s$) under two different initial relay node assignments.

Sender	Direct Transmission Capacity (Mbps)	Relay Assignment I			Relay Assignment II		
		Initial	Final	Final Capacity (Mbps)	Initial	Final	Final Capacity (Mbps)
s_1	6.1	r_1	r_1	10.2	$\emptyset$	r_1	10.3
s_2	10.6	$\emptyset$	$\emptyset$	10.6	$\emptyset$	$\emptyset$	10.6
s_3	8.8	r_2	r_2	20.1	$\emptyset$	r_2	20.1
s_4	6.3	$\emptyset$	r_3	10.7	r_2	r_3	10.7
s_5	7.3	r_5	r_5	$\boxed{9.0}$	r_5	r_5	$\boxed{9.0}$
s_6	9.6	$\emptyset$	$\emptyset$	9.6	$\emptyset$	$\emptyset$	9.6
s_7	$\boxed{4.2}$	r_3	r_4	9.3	r_1	r_4	9.3
s_8	6.9	r_8	r_6	16.6	$\emptyset$	r_6	16.6
s_9	11.3	$\emptyset$	$\emptyset$	11.3	$\emptyset$	$\emptyset$	11.4
s_{10}	11.1	$\emptyset$	$\emptyset$	11.1	$\emptyset$	$\emptyset$	11.1
s_{11}	9.5	r_7	r_{10}	20.1	$\emptyset$	$\emptyset$	9.5
s_{12}	7.5	r_{10}	r_9	12.8	r_{10}	r_9	12.8
s_{13}	8.5	$\emptyset$	r_7	16.9	r_8	r_7	16.9
s_{14}	9.8	$\emptyset$	$\emptyset$	9.8	$\emptyset$	$\emptyset$	9.8
s_{15}	6.1	$\emptyset$	r_8	12.9	r_7	r_8	12.9
s_{16}	20.5	$\emptyset$	$\emptyset$	20.5	$\emptyset$	$\emptyset$	20.5
s_{17}	49.6	$\emptyset$	$\emptyset$	49.6	$\emptyset$	$\emptyset$	49.6
s_{18}	36.0	$\emptyset$	$\emptyset$	36.0	$\emptyset$	$\emptyset$	36.0
s_{19}	33.5	$\emptyset$	$\emptyset$	33.5	$\emptyset$	$\emptyset$	33.5
s_{20}	21.2	$\emptyset$	$\emptyset$	21.2	$\emptyset$	$\emptyset$	21.2

unique. But regardless of initial relay node assignment, the objective value can always converge to the optimum (by Corollary 1.1). To validate this point, in Table 2, we show the results of running the ORA algorithm under two different initial relay node assignments, denoted as I and II (see Table 2).

In Table 2, the second column shows the capacity for each source-destination pair under direct transmissions. Note that the minimum capacity among all pairs is 4.2 Mbps, which is associated with s_7. The third to fifth columns are results under initial relay node assignment I and sixth to eighth columns are results under initial relay node assignment II. The symbol $\emptyset$ denotes direct transmissions. Note than the initial relay node assignments I and II are different. As a result, the final assignment is different under I and II. However, the final objective value (i.e., $C_{\min}$) under I and II is identical (10.7 Mbps).

Figure 8 shows the objective value $C_{\min}$ at each iteration under initial relay node assignments I and II. Under either initial relay node assignments I or II, $C_{\min}$ is a non-decreasing function of iteration number. The increase of $C_{\min}$ by cooperative communications over direct transmissions is significant (from 4.2 Mbps to 10.7 Mbps).

Case 2: $N_r < N_s$. In this case (see Fig. 9), we have 20 source-destination pairs and 10 relay nodes.

Table 3 shows the results of this case under two different initial relay node assignments I and II. The second column in Table 3 lists the capacity under direct transmissions for each source-destination pair. As discussed at the end of Section 4.2, for the case of $N_r < N_s$, it is only necessary to consider relay node assignment for $N_r = 10$ source nodes corresponding to the 10 smallest capacities, i.e., nodes s_1, s_3, s_4, s_5, s_7, s_8, s_{11}, s_{12}, s_{13}, and s_{15}. As a result, the problem size can be reduced.

Again in Table 3, the objective value $C_{\min}$ is identical (9.0 Mbps) regardless of different initial relay node assignments (I and II). Note that the final relay node assignment under I and II is not identical, although the objective value $C_{\min}$ is the same. The increase of

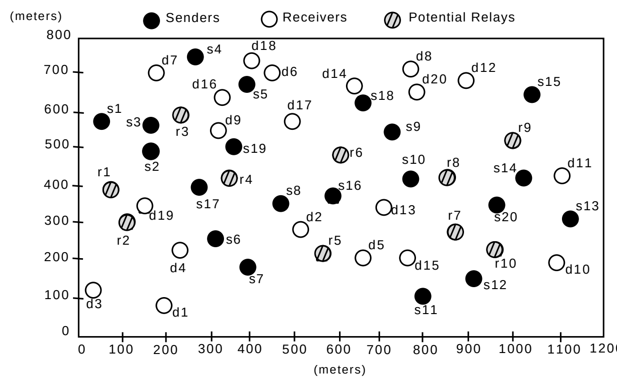

Figure 9: A 50-node network topology for Case 2 ($N_r < N_s$), with $N_s = 20$ and $N_r = 10$.

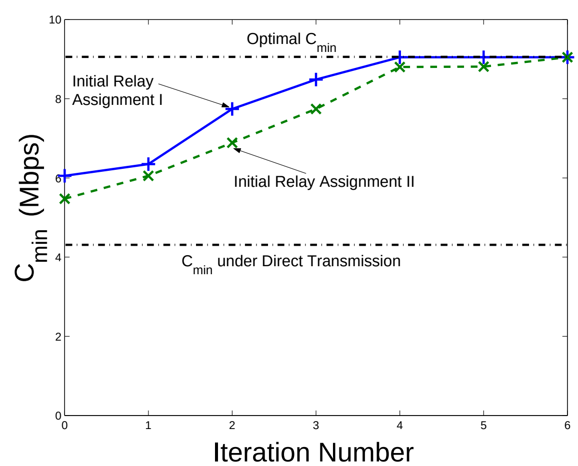

Figure 10: Case 2 ($N_r < N_s$): The objective value $C_{\min}$ at each iteration of the ORA algorithm under two different initial node assignments.

Table 4: An example illustrating the importance of preprocessing.

Sender	Direct Transmission Capacity (Mbps)	Without Preprocessing		
		Initial	Final	Final Capacity (Mbps)
s_1	6.1	r_2	r_2	15.3
s_2	10.6	r_{12}	r_7	21.8
s_3	8.8	r_1	r_1	16.6
s_4	6.4	r_{14}	r_4	$\boxed{10.7}$
s_5	7.3	r_{13}	r_{12}	15.0
s_6	9.6	r_{16}	r_6	21.3
s_7	4.2	r_5	r_8	11.0
s_8	6.9	r_6	r_{10}	11.0
s_9	$\underline{\mathbf{11.3}}$	r_{11}	r_{11}	$\underline{\mathbf{11.0}}$
s_{10}	11.0	r_{15}	r_{19}	22.6
s_{11}	9.5	r_{17}	r_{20}	21.1
s_{12}	7.5	r_{18}	r_{18}	12.8
s_{13}	8.5	r_3	r_{17}	16.9
s_{14}	9.8	r_{10}	r_{15}	12.2
s_{15}	6.0	r_{19}	r_{16}	12.9

C_{min} by cooperative communications over direct transmissions is significant (from 4.2 Mbps to 9.0 Mbps).

Figure 10 shows the objective value C_{min} at each iteration under initial relay node assignments I and II. Again, we observe that in Fig. 10, C_{min} is a non-decreasing function of iteration number under either initial relay node assignments I or II.

Importance of Preprocessing Now we use a set of simulation results to show the significance of preprocessing in our ORA algorithm. We consider the same network in Fig. 7 with 15 source-destination pairs and 20 relay nodes. Now we remove the preprocessing step in ORA algorithm. As an example, the third column of Table 4 shows an initial assignment without first going through the preprocessing step. Although the objective value C_{min} also reaches the same optimal value (10.7 Mbps) as that in Table 2, the final capacity for some non-bottleneck source nodes could be worse than direct transmissions. For example, for s_9, its final capacity is 11.0 Mbps, which is less than direct transmissions (11.3 Mbps). Such event is undetectable without the preprocessing step, as 11.0 Mbps is still greater than the optimal objective value (10.7 Mbps).

On the other hand, when the preprocessing step is employed, ORA can ensure that the final capacity for each source-destination pair is no less than direct transmissions.

7. CONCLUSION

Cooperative communications is a powerful communication paradigm to achieve spatial diversity. However, the performance of such communication paradigm hinges upon the choice of relay node in the network. In this paper, we studied the relay node assignment problem in a network environment, where multiple source-destination pairs compete for the same pool of relay nodes in the network. The main contribution of this paper is a polynomial time algorithm to solve this problem. A key idea in this algorithm is a "linear marking" mechanism, which is able to achieve linear complexity at each iteration. We gave a formal proof of optimality for the algorithm and used numerical results to demonstrate its capability. There are several attractive properties associated with this algorithm, such as its robustness to the number of relay nodes in the network, its guarantee for each source-destination pair to have capacity no less than direct transmissions, and its ability to find the optimal objective regardless of initial assignment.

Acknowledgments

The work of Y.T. Hou, Y. Shi, and S. Sharma has been supported in part by the National Science Foundation (NSF) under Grant CNS-0721421 and Office of Naval Research (ONR) under Grant N00014-08-1-0084. The work of S. Kompella has been supported in part by the ONR.

8. REFERENCES

[1] B. Aazhang, R.S Blum, J.N. Laneman, K.J.R. Liu, W. Su, and A. Wittneben, *IEEE Journal on Selected Areas in Communications – Special Issue on Cooperative Communications and Networking,* vol. 25, no. 2, Feb. 2007.

[2] 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.

[3] J. Cai, S. Shen, J.W. Mark, and A.S. Alfa, "Semi-distributed user relaying algorithm for amplify-and-forward wireless relay networks," to appear in *IEEE Transactions on Wireless Communications.*

[4] T.M. Cover and A. EL Gamal, "Capacity theorems for the relay channel," *IEEE Transactions on Information Theory,* vol. 25, issue 5, pp. 572–584, 1979.

[5] G. Kramer, R. Berry, A. El Gamal, H. El Gamal, M. Franceschetti, M. Gastpar, and J.N. Laneman, *IEEE Transactions on Information Theory – Special Issue on Models, Theory, and Codes for Relaying and Cooperation in Communication Networks,* vol. 53, no. 10, Oct. 2007.

[6] J.N. Laneman, D.N.C. Tse, and G.W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Transactions on Information Theory,* vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[7] T. C-Y. Ng, W. Yu, J. Zhang, and A. Reid, "Joint optimization of relay strategies and resource allocations in cooperative cellular networks," *IEEE Journal on Selected Areas in Communications,* vol. 25, no. 2, pp. 328–339, Feb. 2007.

[8] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity – part I: System description," *IEEE Transactions on Communications,* vol. 51, no. 11, pp. 1927–1938, Nov. 2003.

[9] E.C. van der Meulen, "Three terminal communication channels," *Advances in Applied Probability,* vol. 3, pp. 120-154, 1971.

[10] Y. Zhao, R.S. 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, WA, July 9–14, 2006.