An Optimal Algorithm for Relay Node Assignment in Cooperative Ad Hoc Networks

Sushant Sharma, *Student Member, IEEE,* Yi Shi, *Member, IEEE,* Y. Thomas Hou, *Senior Member, IEEE,* Sastry Kompella, *Member, IEEE*

***Abstract*—Recently, cooperative communications, in the form of having each node equipped with a single antenna and exploit spatial diversity via some relay node's antenna, is shown to be a promising approach to increase data rates in wireless networks. Under this communication paradigm, the choice of a relay node (among a set of available relay nodes) is critical in the overall network performance. In this paper, we study the relay node assignment problem in a cooperative ad hoc network environment, where multiple source-destination pairs compete for the same pool of relay nodes in the network. Our objective is to assign the available relay nodes to different source-destination pairs so as to maximize the minimum data rate among all pairs. The main contribution of this paper is the development of an optimal polynomial time algorithm, called ORA, that achieves this objective. A novel idea in this algorithm is a "linear marking" mechanism, which maintains linear complexity of each iteration. We give a formal proof of optimality for ORA and use numerical results to demonstrate its capability.**

***Index Terms*—Cooperative communications, relay node assignment, achievable rate, ad hoc network, optimization.**

I. 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 on a handheld wireless device). To achieve spatial diversity without requiring multiple transceiver antennas on the same node, the so-called *cooperative communications* has been introduced [10], [16], [17]. 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.

We consider two categories of cooperative communications, namely, *amplify-and-forward* (AF) and *decode-and-forward* (DF) [10]. Under AF, the cooperative relay node amplifies 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 critical role in the performance of cooperative communications [1], [2], [24]. As we shall see in Section III, an improperly chosen relay node may offer a smaller data rate for a source-destination pair than that under direct transmission.

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

A. Main Contributions

In this paper, we study how to assign a set of relay nodes to a set of source-destination pairs so as to maximize the minimum achievable data rate among all 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 novel 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 of whether the number of relay nodes in the network is more than or less than the number of source-destination pairs; (ii) the final achievable rate 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.

Manuscript received May 26, 2009; revised January 29, 2010 and August 23, 2010; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Diggavi.

An abridged version of this paper was published in ACM MobiHoc 2008 under the title "Optimal Relay Assignment for Cooperative Communications".

S. Sharma is with the Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061 USA (e-mail: sushant@cs.vt.edu).

Y. Shi and Y.T. Hou are with the Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061 USA e-mail: (yshi@vt.edu; thou@vt.edu).

S. Kompella is with the Information Technology Division, US Naval Research Laboratory, Washington, DC 20375 USA (e-mail: kompella@itd.nrl.navy.mil).

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **23 AUG 2010**	2. REPORT TYPE	3. DATES COVERED **00-00-2010 to 00-00-2010**

4. TITLE AND SUBTITLE **An Optimal Algorithm for Relay Node Assignment in Cooperative Ad Hoc Networks**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Research Laboratory,Information Technology Division,Washington,DC,20375**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
to appear in IEEE/ACM Transactions on Networking, Issue to be determined

14. ABSTRACT
Recently, cooperative communications, in the form of having each node equipped with a single antenna and exploit spatial diversity via some relay node?s antenna, is shown to be a promising approach to increase data rates in wireless networks. Under this communication paradigm, the choice of a relay node (among a set of available relay nodes) is critical in the overall network performance. In this paper, we study the relay node assignment problem in a cooperative ad hoc network environment, where multiple source-destination pairs compete for the same pool of relay nodes in the network. Our objective is to assign the available relay nodes to different source-destination pairs so as to maximize the minimum data rate among all pairs. The main contribution of this paper is the development of an optimal polynomial time algorithm, called ORA, that achieves this objective. A novel idea in this algorithm is a ?linear marking? mechanism, which maintains linear complexity of each iteration. We give a formal proof of optimality for ORA and use numerical results to demonstrate its capability.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**14**	

- We provide a sketch of a possible implementation of the ORA algorithm. Some practical issues and overhead in the implementation are discussed.

B. Paper Organization

In Section II, we discuss related work and contrast them with this paper. Section III gives a brief overview of cooperative communications, so as to set the context of our study. In Section IV, we describe the relay node assignment problem in a cooperative ad hoc network environment. Section V presents our ORA algorithm. In Section VI, we give a proof of optimality for ORA. Section VII presents numerical results, and Section VIII presents a sketch of how ORA can be implemented. Section IX concludes this paper.

II. Related Work

The concept of cooperative communications can be traced back to the three-terminal communication channel (or a relay channel) in [20] by Van Der Meulen. Shortly after, Cover and El Gamal studied the general relay channel and established an achievable lower bound for data transmission [4]. These two seminal works laid down the foundation for the present-day research on cooperative communications that can be broadly classified into the following three categories.

(a) Physical Layer Schemes. Current research on CC aims to exploit distributed antennas on other nodes in the network. This has resulted in several protocols at the physical layer [5], [7], [8], [10], [14], [16], [17]. These protocols describe various ways through which nodes can cooperate at the physical layer.

In [10], Laneman *et al.* studied the mutual information between a pair of nodes using a third cooperating node under the so-called fixed relaying schemes (AF or DF). The underlying physical layer model for CC in this paper is based on these two schemes. In addition to fixed relaying schemes, the authors also presented selection relaying, in which nodes can switch between AF or DF (depending on instantaneous channel conditions), and incremental relaying, which utilizes limited feedback from the receiving node to further improve the performance of CC.

In [7], Gunduz and Erkip studied an opportunistic cooperation scheme in which a feedback channel among cooperating nodes can be used to share channel state information and help perform power control. The authors showed that the performance of DF improves when power control is employed. This kind of opportunistic DF is an alternative to the physical layer fixed DF considered in our work.

Another alternative physical layer scheme could be the delay-tolerant DF presented in [5]. In this delay-tolerant DF scheme, distributed space-time codes are used to address the issue of asynchrony (transmission delay) among cooperative transmitters.

Additionally, in [8] and [14], authors studied multi-hop cooperative protocols that involve cooperation among multiple transmitting nodes along the path. In [16] and [17], the authors performed an in-depth study on the practical issues of implementing user cooperation in a conventional CDMA system.

(b) Network Layer Schemes for Multi-hop Networks. Recent efforts on CC at the network layer include [9], [15], [22]. In [9], Khandani *et al.* studied minimum energy routing problem (for a single message) by exploiting both wireless broadcast advantage and CC. However, their proposed solutions cannot provide any performance guarantee for general ad hoc networks. In [22], Yeh and Berry aimed to generalize the well known maximum differential backlog policy [18] in the context of CC. They formulated a challenging nonlinear program with exponential number of variables that characterizes the network stability region, but only provided solutions for a few simple network topologies. 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 can be improved by using such cooperative clusters. However, problems related with optimal routing and relay node assignment were not discussed in their work.

(c) Relay Node Assignment for Ad hoc Networks. The most relevant research to our work (i.e. relay node assignment) include [1], [2], [13], [21], [24]. In [24], Zhao *et al.* showed that for a single source-destination pair, in the presence of multiple relay nodes, it is sufficient to choose one "best" relay node, instead of multiple relay nodes. 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 have adopted in this paper. In [21], Wang *et al.* showed how game theory can be used by a single session to select the best cooperative relay node. In [1], Bletsas *et al.* proposed a distributed scheme for relay node selection based on the instantaneous channel conditions at the relay node. In contrast to [1], [21], and [24], our paper is not limited to a single-session, and considers the relay node assignment for multiple competing sessions with the goal of maximizing the minimum data rate among all of them. In [13], Ng and Yu studied an important utility maximization problem for the joint optimization of relay node selection, cooperative communications, and resource allocation in a cellular network. However, their solution procedure has non-polynomial running time. In [2], Cai *et al.* studied relay node selection and power allocation for AF-based wireless relay networks, and proposed a heuristic solution. Additionally, both [2] and [13] have different objectives from our work.

III. 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 the destination node d. Due to the broadcast nature of wireless communications, this transmission is also overheard by the relay node r. In the second time slot, node r forwards the data received in the first time slot to node d. Note that such a two-slot structure is necessary for cooperative communications due to the half-duplex nature of most wireless transceivers.

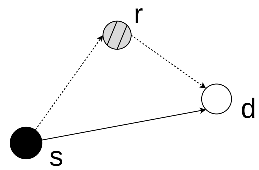

Fig. 1. A three-node schematic for cooperative communication.

In this section, we give expressions for achievable data rate under cooperative communications and direct transmissions (i.e., no cooperation). For cooperative communications, we consider both amplify-and-forward (AF) and decoded-and-forward (DF) modes [10].

Amplify-and-Forward (AF) Under this mode, let h_{sd}, h_{sr}, h_{rd} capture the effects of path-loss, shadowing, and fading between nodes s and d, s and r, and r and d, respectively. Denote $z_d[1]$ and $z_d[2]$ the zero-mean background noise at node d in the first time slot and second time slot, respectively, both with variance σ_d^2. Denote $z_r[1]$ the zero-mean background noise at node r in the first time slot, with variance σ_r^2.

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

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

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

$$y_{sr} = h_{sr}x_s + z_r[1] \ . \tag{2}$$

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

$$y_{rd} = h_{rd} \cdot \alpha_r \cdot y_{sr} + z_d[2] \ ,$$

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

$$y_{rd} = h_{rd}\alpha_r \cdot (h_{sr}x_s + z_r[1]) + z_d[2] \ . \tag{3}$$

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

$$\mathbf{Y} = \mathbf{H}x_s + \mathbf{BZ} \ ,$$

where

$$\mathbf{Y} = \begin{bmatrix} y_{sd} \\ y_{rd} \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} h_{sd} \\ \alpha_r h_{rd} h_{sr} \end{bmatrix},$$
$$\mathbf{B} = \begin{bmatrix} 0 & 1 & 0 \\ \alpha_r h_{rd} & 0 & 1 \end{bmatrix}, \quad \text{and} \quad \mathbf{Z} = \begin{bmatrix} z_r[1] \\ z_d[1] \\ z_d[2] \end{bmatrix}. \tag{4}$$

It has been shown in [10] that the above channel, which combines both 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 achievable data rate $C_{\text{AF}}(s, r, d)$ from s to d can be given by

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

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 $C_{\text{AF}}(s, r, d) = \frac{W}{2} \log\left(1 + \frac{P_s}{\sigma_d^2}|h_{sd}|^2 + \frac{P_s|h_{sr}|^2 P_r |h_{rd}|^2}{P_s\sigma_d^2|h_{sr}|^2 + P_r\sigma_r^2|h_{rd}|^2 + \sigma_r^2\sigma_d^2}\right)$. 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

$$C_{\text{AF}}(s, r, d) = W \cdot I_{\text{AF}}(\text{SNR}_{sd}, \text{SNR}_{sr}, \text{SNR}_{rd}) \ , \tag{6}$$

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, and then transmits the estimated data to destination node d in the second time slot. The achievable data rate for DF under the two time-slot structure is given by [10] as

$$C_{\text{DF}}(s, r, d) = W \cdot I_{\text{DF}}(\text{SNR}_{sd}, \text{SNR}_{sr}, \text{SNR}_{rd}) \ , \tag{7}$$

where

$$I_{\text{DF}}(\text{SNR}_{sd}, \text{SNR}_{sr}, \text{SNR}_{rd}) = \frac{1}{2}\min\{\log_2(1 + \text{SNR}_{sr}), \log_2(1 + \text{SNR}_{sd} + \text{SNR}_{rd})\}. \tag{8}$$

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

Direct Transmission When cooperative communications (i.e., relay node) is not used, source node s transmits to destination node d in both time slots. The achievable data rate 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 that cooperative communications is always better than the direct transmission. In fact, a poor choice of relay node could make the achievable data rate under cooperative communications to be lower than that under direct transmission. 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 is also applicable for DF. In this paper, we develop a relay node assignment algorithm for both AF and DF. Table I lists the notation used in this paper.

TABLE I
NOTATION

Symbol	Definition
$C_{\mathrm{R}}(s_i, r_j)$	Achievable rate for s_i-d_i pair when relay node r_j is used
$C_{\mathrm{R}}(s_i, \emptyset)$	Achievable rate for s_i-d_i pair under direct transmission
$C_{\min}$	The minimum rate among all source-destination pairs
h_{uv}	Effect of path-loss, shadowing, and fading from node u to node v
$\mathcal{N}_s$	Set of source nodes in the network
$\mathcal{N}_r$	Set of relay nodes in the network
N_s	$= \lvert\mathcal{N}_s\rvert$, number of source nodes in the network
N_r	$= \lvert\mathcal{N}_r\rvert$, number of relay nodes in the network
N	Number of all the nodes in the network
P	Maximum transmission power
r_j	The j-th relay node, $r_j \in \mathcal{N}_r$
$\mathcal{R}_\psi(s_i)$	The relay node assigned to s_i under ψ
s_i	The i-th source node, $s_i \in \mathcal{N}_s$
$\mathcal{S}_\psi(r_j)$	The source node that uses r_j under ψ
SNR_{uv}	The signal noise ratio between nodes u and v
W	Channel bandwidth
x_s	Signal transmitted by node s
y_{uv}	Received signal at node v (form node u)
$z_v[t]$	Background noise at node v during time slot t
α_r	Amplifying factor at relay r
σ_v^2	Variance of background noise at node v
ψ	A solution for relay node assignment

IV. 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 proposed for cooperative communications [10]. The channel gain from node u to v is captured by variable h_{uv}. 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 transmission where every source node s_i is paired with a destination node d_i, i.e., $N_d = N_s$. We also consider that each node is equipped with a single transceiver and can transmit/receive within one channel at a time. We assume that each node can only serve a unique role of source, destination, or relay. That is, $N_r = N - 2N_s$. Further, we assume that a session utilizes one relay node for CC [24].

Note that a source node may not always get a relay node. There are two possible scenarios in which this may happen. First, there may not be sufficient number of relay nodes in the network (e.g., $N_r < N_s$). In this case, some source nodes will not have relay nodes. Second, even if there are enough relay nodes, a sender may choose not to use a relay node if it leads to a lower data rate than direct transmission (see discussion at the end of Section III).

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

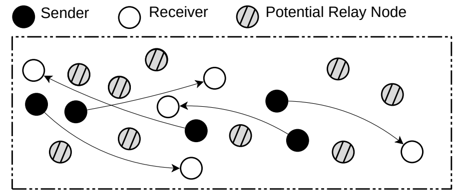

Fig. 2. A cooperative ad hoc network consisting of source nodes, destination nodes, and relay nodes.

More formally, denote $\mathcal{R}(s_i)$ the relay node assigned to s_i, and $\mathcal{S}(r_j)$ as the source node that uses r_j. For both AF and DF, the achievable data rate of the session can be written as (see Section III)

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

with $I_{\mathrm{R}}(\cdot) = I_{\mathrm{AF}}(\cdot)$ when AF is employed, and $I_{\mathrm{R}}(\cdot) = I_{\mathrm{DF}}(\cdot)$ when DF is employed. In case s_i does not use a relay, we denote $\mathcal{R}(s_i) = \emptyset$, and the data rate is the achievable rate under direct transmission, i.e.,

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

Combining both these cases, we have

$$C_{\mathrm{R}}(s_i, \mathcal{R}(s_i)) = \begin{cases} W I_{\mathrm{R}}(\mathrm{SNR}_{s_i,d_i}, \mathrm{SNR}_{s_i,\mathcal{R}(s_i)}, \\ \qquad \mathrm{SNR}_{\mathcal{R}(s_i),d_i}) & \text{if } \mathcal{R}(s_i) \neq \emptyset \\ W \log(1 + \mathrm{SNR}_{s_i,d_i}) & \text{if } \mathcal{R}(s_i) = \emptyset \end{cases} \quad (9)$$

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

Denote $C_{\min}$ as our objective function, which is the minimum rate among all source nodes. That is,

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

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

In subsequent sections, we present a polynomial time solution to the relay node assignment problem along with a correctness proof.

V. An Optimal Relay Assignment Algorithm

A. Basic Idea

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

Initially, the ORA algorithm starts with a random but 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 $C_{\min}$.

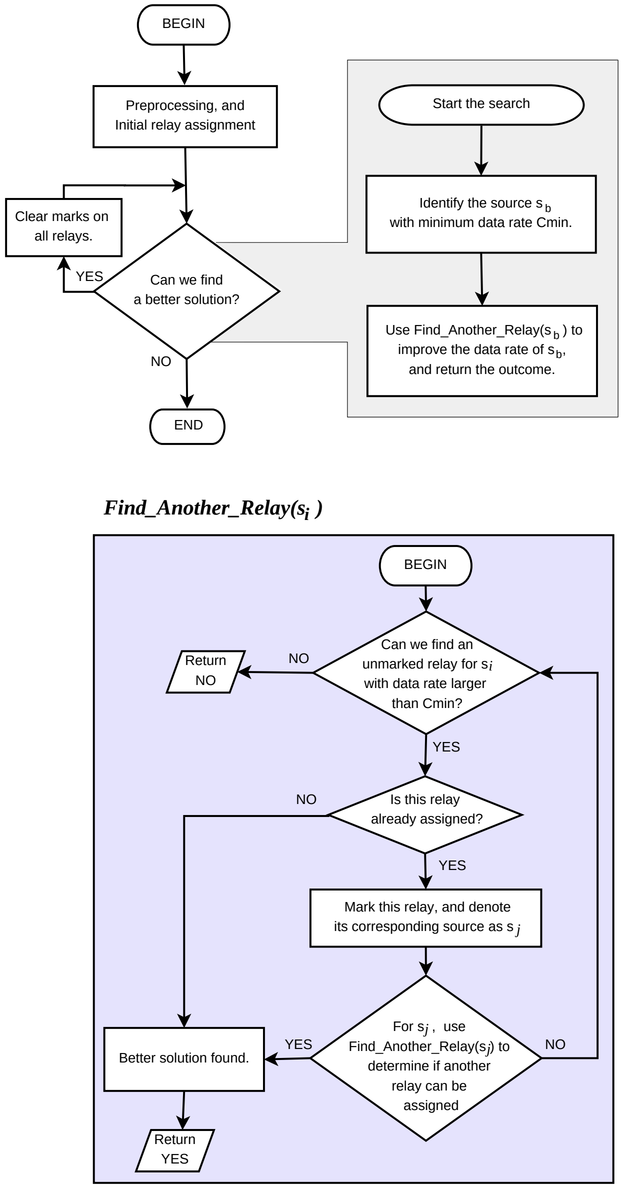

Fig. 3. A flow chart of the ORA algorithm.

Then, ORA helps this source node to search a better relay such that this "bottleneck" data rate can be increased. In the case that the selected relay is already assigned to another source node, further adjustment of relay node for 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 made on a relay node, the affected source node should still maintain a data rate 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

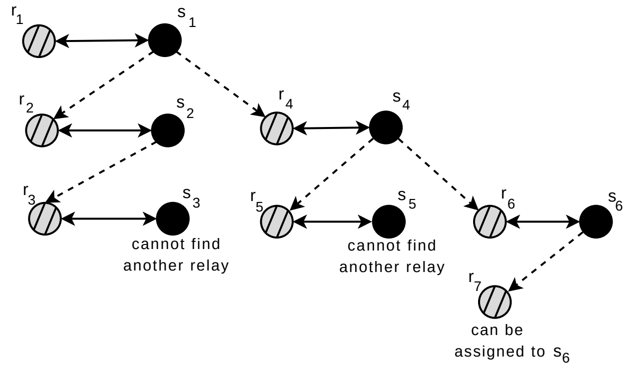

Fig. 4. An example tree topology in ORA algorithm for finding a better solution.

ORA addresses both problems successfully. Specifically, we show the complexity of the ORA algorithm is polynomial in Section V-D. We will also give a correctness proof of its optimality in Section VI.

B. 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 data rate $C_{\mathrm{R}}(s_i, r_j)$ by (9). Each source node s_i also computes the rate $C_{\mathrm{R}}(s_i, \emptyset)$ by (9) 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 in its data rate compared to direct transmissions, i.e., those relays with $C_{\mathrm{R}}(s_i, r_j) > C_{\mathrm{R}}(s_i, \emptyset)$. Obviously, it only makes sense to consider these relays for CC. In the case that no relay can offer any increase of data rate 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 pre-processing step, we have already identified the list of relay nodes for each source node that can increase its data rate compared to direct transmission. 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 there is no relay node available to a source node, then this source node will simply employ direct transmission as its initial assignment. Upon the completion of this assignment, each source node will have the data rate no less than that under direct transmission.

The next step in the ORA algorithm is to find a better assignment, which represents an iteration process. This is the key step in the ORA algorithm. The detail of this step is shown in the bottom portion of Fig. 3. As a starting point of this step, ORA algorithm identifies the smallest data rate $C_{\min}$ among all sources. ORA algorithm aims to increase this minimum rate for the corresponding source node, while having all other source nodes maintain their data rates 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 rate $C_{\min}$ under the current assignment (with relay node r_1). Then

s_1 examines other relays with a rate larger than C_{min}. If it cannot find such a relay, then no better solution is found and the ORA algorithm terminates.

In case of a tie, i.e., when two or more source nodes have the same smallest data rate, the tie is broken by choosing the source node with the highest node index.

- Otherwise, i.e., if there are better relays, we consider these relays in the *non-increasing* order in terms of data rate (should it be assigned to s_1). That is, we try the relay that can offer the maximum possible increase in data rate first. In case of a tie, i.e., when two or more relay nodes offer the same maximum data rate, the tie is broken by choosing the relay node with the highest node index.
- Suppose that source node s_1 considers relay node r_2. 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 transmission) while making sure that such new assignment still has its data rate larger than C_{min}. This process is identical to what we have done for s_1, with the only (but important) difference that s_2 will not consider a relay that has already been "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. Moreover, if the relay under consideration by s_2 is the one that is being used by the source node that initiated the iteration, i.e., relay r_1, then it is easy to see that r_1 can be taken away from s_1. A better solution, where r_1 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. We also note that s_2 can consider direct transmission if it offers a data rate larger than C_{min}.
- Suppose that s_3 cannot find any "unmarked" relay that offers a data rate larger than C_{min}, and its data rate under direct transmission is no more than C_{min}. Then s_2 cannot use r_3 as its relay.
- If any "unmarked" relay that offers a data rate 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 rate 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 the "linear marking" mechanism. These marks will only be cleared when the current iteration terminates and before the start of the next iteration. A pseudocode for the ORA algorithm is shown in Fig. 5.

Main algorithm

1. Perform preprocessing and an initial relay node assignment.
2. Set all the relay nodes in the network as "unmarked".
3. Denote s_b the source node with C_{min},* the smallest data rate among all source nodes. The corresponding destination node of s_b is d_b and the corresponding relay node is $\mathcal{R}(s_b)$.
4. Find_Another_Relay $(s_b, \mathcal{R}(s_b), C_{\text{min}})$.
5. If s_b finds a better relay, then go to line 2.
6. Otherwise, the algorithm terminates.

Subroutines

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

7. 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 $C_{\text{R}}(\mathcal{S}(r_j), r_k)$.*
8.   Run Check_Relay_Availability(r_k, C_{min}).
9.   If r_k is available, then do the following:
10.   Remove relay node r_j's assignment to $\mathcal{S}(r_j)$;
11.   Assign relay node r_k to $\mathcal{S}(r_j)$.
12.   Otherwise, continue on to next r_k and go to line 8.
13. If all relays are unavailable, then $\mathcal{S}(r_j)$ cannot find another relay.

Check_Relay_Availability(r_j, C_{min}):

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

* A tie is broken by choosing the node with the largest node index.

Fig. 5. Pseudocode for the ORA algorithm.

We now use an example to illustrate the operation of the ORA algorithm, in particular, its "linear marking" mechanism. Readers who already understood the ORA algorithm can skip this example.

Example 1: Suppose that there are seven source-destination pairs and seven relay nodes in the network.

Table II(a) shows the data rate for each source node s_i when relay node r_j is assigned to it. The symbol $\emptyset$ indicates direct transmission. Also shown in Table II(a) is an initial relay node assignment, which is indicated by an underscore on the intersecting row (s_i) and column (r_j). Note that the preprocessing step before the initial assignment ensures that the data rate for each source-destination pair in the initial assignment is no less than that under direct transmission.

Under the initial relay node assignment in Table II(a), source s_3 is identified as the bottleneck source node s_b with the smallest rate of $C_{\text{min}} = 13$. Since consideration of relay nodes is performed in the order of non-increasing (from largest to smallest) data rate for the source node under consideration, r_4 is therefore considered for s_3. Since r_4 is already assigned to source node s_2, we "mark" r_4 now. Now s_2 needs to find another relay. But any other relay (or direct transmission) will result in a data rate 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 next relay node that offers the second largest data rate value, i.e., relay node r_7. Since r_7 is already assigned to sender s_4, we "mark" r_7 now. Next, ORA algorithm will check to see if s_4 can find another relay. It turns out that none of the relay

TABLE II
AN EXAMPLE.

(a) Initial relay node assignment.

	$\emptyset$	r_1	r_2	r_3	r_4	r_5	r_6	r_7
s_1	14	7	24	5	14	$\underline{15}$	17	9
s_2	9	8	10	11	$\underline{20}$	10	12	11
$\rightarrow s_3$	11	10	$\underline{13}$	17	21	8	9	19
s_4	12	8	9	12	11	10	9	$\underline{18}$
s_5	10	9	18	$\underline{19}$	24	9	13	23
s_6	7	18	12	6	11	11	$\underline{17}$	20
s_7	$\underline{16}$	1	9	4	14	19	8	12

(b) Assignment after the first iteration.

	$\emptyset$	r_1	r_2	r_3	r_4	r_5	r_6	r_7
$\rightarrow s_1$	14	7	24	5	14	$\underline{15}$	17	9
s_2	9	8	10	11	$\underline{20}$	10	12	11
s_3	11	10	13	$\underline{17}$	21	8	9	19
s_4	12	8	9	12	11	10	9	$\underline{18}$
s_5	10	9	$\underline{18}$	19	24	9	13	23
s_6	7	18	12	6	11	11	$\underline{17}$	20
s_7	$\underline{16}$	1	9	4	14	19	8	12

(c) Assignment after the second iteration.

	$\emptyset$	r_1	r_2	r_3	r_4	r_5	r_6	r_7
s_1	14	7	24	5	14	15	$\underline{17}$	9
s_2	9	8	10	11	$\underline{20}$	10	12	11
s_3	11	10	13	$\underline{17}$	21	8	9	19
s_4	12	8	9	12	11	10	9	$\underline{18}$
s_5	10	9	$\underline{18}$	19	24	9	13	23
s_6	7	$\underline{18}$	12	6	11	11	17	20
$\rightarrow s_7$	$\underline{16}$	1	9	4	14	19	8	12

(d) Final assignment upon termination.

	$\emptyset$	r_1	r_2	r_3	r_4	r_5	r_6	r_7
s_1	14	7	24	5	14	15	$\underline{17}$	9
s_2	9	8	10	11	$\underline{20}$	10	12	11
$\rightarrow s_3$	11	10	13	$\underline{17}$	21	8	9	19
s_4	12	8	9	12	11	10	9	$\underline{18}$
s_5	10	9	$\underline{18}$	19	24	9	13	23
s_6	7	$\underline{18}$	12	6	11	11	17	20
s_7	16	1	9	4	14	$\underline{19}$	8	12

nodes except r_7 can offer a data rate larger than the current $C_{\min}$ to s_4. As a result, r_7 cannot be taken away from s_4. Source node s_3 will now check for the relay node that offers next largest rate, i.e., r_3. Since r_3 is already assigned to sender s_5, we "mark" r_3 now. Next, ORA algorithm checks to see if s_5 can find another relay. Then s_5 checks relay nodes in non-increasing order of data rate values. Since r_4 (with largest rate), r_7 (with the second largest rate), and r_3 (with the third largest rate) are all marked, they will not be considered. The relay with the fourth largest rate is r_2, which offers a rate of $18 > C_{\min} = 13$. Moreover, r_2 is the relay node assigned to $s_b = s_3$. Thus, s_5 can choose r_2. The new assignment after the first iteration is shown in Table II(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 identify s_1 as the source node with a minimum data rate in the network. The algorithm will then perform a new search for a better relay node for source s_1. Similar to the first iteration, the assignments for other source nodes can change during this search process, but all assignments should result in data rates larger than 15.

The iteration continues and the final assignment upon termination of ORA algorithm is shown in Table II(d), with the optimal (maximum) value of $C_{\min}$ being 17. ■

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 data rate of each source in this subset under direct transmission is less than the data rate 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 faster due to a smaller problem size.

C. Caveat on the Proposed Marking Mechanism

We now re-visit the marking mechanism in the 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 an exponential complexity for each iteration. Then we discuss our proposed marking mechanism and show its 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 is being considered by s_1 in the new solution and is used by s_2 in the current 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 data rate 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 mark r_4 and 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 on 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, r_4 and r_5 should be unmarked so that they can be considered as candidate relay nodes for s_6, in addition to r_7. It is not hard to show that such marking/unmarking mechanism will consider all possible assignments and can guarantee to find an optimal solution upon termination. However, the complexity of such approach is exponential within each iteration.

In contrast, under the 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. 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. Unmarking for all nodes is performed only at the end of an iteration so that there is a clean start for the next iteration.

An immediate question regarding our marking mechanism is: how could such a "linear marking" 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 VI, where we will prove that ORA can guarantee that its final solution is optimal.

D. Complexity Analysis

We now analyze the computational complexity of ORA algorithm. 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 maximum number of iterations that ORA can execute. The number of improvements in data rate that an individual source node can have is limited by N_r. As a result, in worst case, the number of iterations that the algorithm can go through are $O(N_s N_r)$. This makes the overall complexity of ORA algorithm to be $O(N_s N_r^2)$.

VI. Proof of Optimality

In this section, we give a correctness proof of the ORA algorithm. That is, upon the termination of the 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 $\mathcal{R}_\psi(s_i)$. Conversely, for ψ, denote the source node that uses relay node r_j as $\mathcal{S}_\psi(r_j)$.

We now assume that there exists a solution $\hat{\psi}$ better 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 $\hat{\psi}$, we denote the relay node assigned to source node s_i as $\mathcal{R}_{\hat{\psi}}(s_i)$. Conversely, for $\hat{\psi}$, we denote the source node that uses relay node r_j as $\mathcal{S}_{\hat{\psi}}(r_j)$.

The key idea in the proof is to exploit the marking status of relay nodes at the end of its last iteration, which is a non-improving iteration. Specifically, in the beginning of this last iteration, ORA will select a "bottleneck" source node, which we denote as s_b. ORA will then try to improve the solution by searching for a better relay node for this bottleneck source node. Since the last iteration is a non-improving iteration, ORA will not find a better solution, and thus will terminate. We will show that $\mathcal{R}_\psi(s_b)$ is not marked at the end of the last iteration of ORA. On the other hand, by assuming that there exists a better solution $\hat{\psi}$ than ψ, we will show that $\mathcal{R}_\psi(s_b)$ will be marked at the end of the last iteration of ORA. This leads to a contradiction and thus $\hat{\psi}$ cannot exist. We begin our proof with the following fact.

Fact 1: For the bottleneck source node s_b under ψ, its relay node $\mathcal{R}_\psi(s_b)$ is not marked at the end of the last iteration of the ORA algorithm.

Proof: In the ORA algorithm, a relay node r_j is marked only if $r_j \neq \mathcal{R}_\psi(s_b)$ (see Check_Relay_Availability() in Fig. 5). Thus, $\mathcal{R}_\psi(s_b)$ cannot be marked at the end of the last iteration of the ORA algorithm. ■

Fact 1 will be the basis for contradiction in our proof for Theorem 1, the main result of this section.

Now we present the following three claims, which *recursively* examine relay node assignment under $\hat{\psi}$. First, for the relay node assigned to s_b in $\hat{\psi}$, i.e., $\mathcal{R}_{\hat{\psi}}(s_b)$, we have the following claim.

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

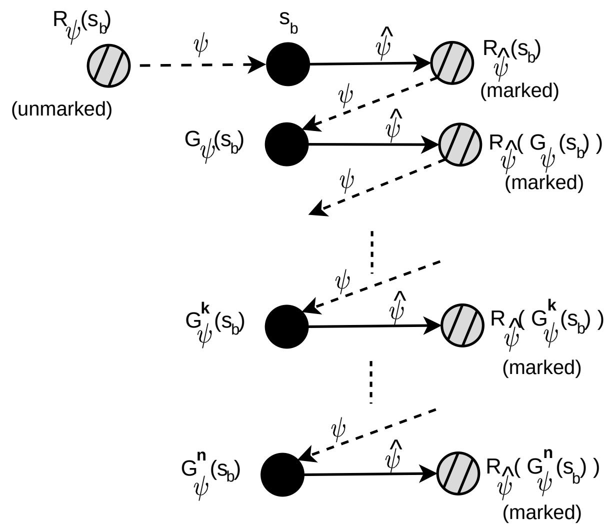

Fig. 6. The sequence of nodes under analysis in the proof of optimality.

Proof: Since $\hat{\psi}$ is a better solution than ψ, we have $C_{\text{R}}(s_b, \mathcal{R}_{\hat{\psi}}(s_b)) \geq \hat{C}_{\text{min}} > C_{\text{min}}$. Thus, by construction, ORA will consider the relay node $\mathcal{R}_{\hat{\psi}}(s_b)$'s availability for s_b in its last iteration. Since ORA algorithm cannot find a better solution in its last iteration, relay $\mathcal{R}_{\hat{\psi}}(s_b)$ should be marked and then the outcome for checking $\mathcal{R}_{\hat{\psi}}(s_b)$'s availability must be unavailable. By "linear marking", the mark on $\mathcal{R}_{\hat{\psi}}(s_b)$ will not be cleared throughout the search process in the last iteration. Thus, the relay node $\mathcal{R}_{\hat{\psi}}(s_b)$ is marked at the end of the last iteration of ORA algorithm.

We now prove the second statement by contradiction. If $\mathcal{R}_{\hat{\psi}}(s_b)$ is $\emptyset$, then s_b will choose $\emptyset$ in the last iteration since it can offer $C_{\text{R}}(s_b, \mathcal{R}_{\hat{\psi}}(s_b)) > C_{\text{min}}$. But this contradicts to the fact that we are now in the last iteration of ORA, which is a non-improving iteration. So $\mathcal{R}_{\hat{\psi}}(s_b)$ cannot be $\emptyset$. Further, since we proved that $\mathcal{R}_{\hat{\psi}}(s_b)$ is marked at the end of the last iteration of the ORA algorithm, it must be assigned to some source node already. ■

By the definition of $\mathcal{S}_\psi(\cdot)$, we have that $\mathcal{R}_{\hat{\psi}}(s_b)$ is assigned to source node $\mathcal{S}_\psi(\mathcal{R}_{\hat{\psi}}(s_b))$ in solution ψ. To simplify notation, define function $\mathcal{G}_\psi(\cdot)$ as

$$\mathcal{G}_\psi(\cdot) = \mathcal{S}_\psi(\mathcal{R}_{\hat{\psi}}(\cdot)) \ . \tag{10}$$

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

Since $\mathcal{R}_{\hat{\psi}}(s_b) \neq \mathcal{R}_\psi(s_b)$, they are assigned to different source nodes in ψ, i.e., $\mathcal{G}_\psi(s_b) \neq s_b$. Now, we recursively investigate the relay node assigned to source $\mathcal{G}_\psi(s_b)$ under solution $\hat{\psi}$, i.e., $\mathcal{R}_{\hat{\psi}}(\mathcal{G}_\psi(s_b))$. We have the following claim (also see Fig. 6).

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

The proofs for both statements in this claim follow the same token as that for Claim 1.

Again, by the definition of $\mathcal{S}_\psi(\cdot)$, we have that relay node $\mathcal{R}_{\hat{\psi}}(\mathcal{G}_\psi(s_b))$ is assigned to source node $\mathcal{S}_\psi(\mathcal{R}_{\hat{\psi}}(\mathcal{G}_\psi(s_b)))$ in solution ψ. By (10), we have source $\mathcal{S}_\psi(\mathcal{R}_{\hat{\psi}}(\mathcal{G}_\psi(s_b))) = \mathcal{G}_\psi(\mathcal{G}_\psi(s_b))$. To simplify the notation, we define function $\mathcal{G}^2_\psi(\cdot)$ as

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

Thus, relay node $\mathcal{R}_{\hat{\psi}}(\mathcal{G}_\psi(s_b))$ is assigned to source node $\mathcal{G}^2_\psi(s_b)$ in ψ. Now we have two cases: source node $\mathcal{G}^2_\psi(s_b)$ may or may not be a node in $\{s_b, \mathcal{G}_\psi(s_b)\}$. If source node $\mathcal{G}^2_\psi(s_b)$ is a node in $\{s_b, \mathcal{G}_\psi(s_b)\}$, then we terminate our recursive procedure. Otherwise, we can further consider its relay node in $\hat{\psi}$.

In general we can use the following notation.

$$\begin{aligned} \mathcal{G}^0_\psi(s_b) = &\quad s_b, \\ \mathcal{G}^k_\psi(s_b) = &\quad \mathcal{G}_\psi(\mathcal{G}^{k-1}_\psi(s_b)) \quad (k \geq 1). \end{aligned} \tag{11}$$

Since the numbers of source nodes are finite, our recursive procedure will terminate in finite steps. Suppose that we terminate at $k = n$.

Following the same token for Claims 1 and 2, we can obtain a similar claim for each of the relay nodes $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^2_\psi(s_b))$, $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^3_\psi(s_b)), \cdots, \mathcal{R}_{\hat{\psi}}(\mathcal{G}^k_\psi(s_b)), \cdots, \mathcal{R}_{\hat{\psi}}(\mathcal{G}^n_\psi(s_b))$ (see Fig. 6). Thus, we can generalize the statements in Claims 1 and 2 for relay node $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^k_\psi(s_b))$ and have the following claim.

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

Proof: Since $\hat{\psi}$ is a better solution than ψ, we can say that $C_{\text{R}}(\mathcal{G}^k_\psi(s_b), \mathcal{R}_{\hat{\psi}}(\mathcal{G}^k_\psi(s_b))) \geq \hat{C}_{\text{min}} > C_{\text{min}}$. Note that $\mathcal{G}^k_\psi(s_b)$ is some source node in the solution ψ obtained by ORA, whereas $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^k_\psi(s_b))$ is the relay node assigned to this source node in the hypothesized better solution $\hat{\psi}$. Our goal is to show that ORA should have marked this relay node in its last iteration.

Since $C_{\text{R}}(\mathcal{G}^k_\psi(s_b), \mathcal{R}_{\hat{\psi}}(\mathcal{G}^k_\psi(s_b))) > C_{\text{min}}$ and $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^k_\psi(s_b))$ is not assigned to $\mathcal{G}^k_\psi(s_b)$ in the last iteration of ORA, then by construction of ORA, ORA must have checked $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^k_\psi(s_b))$'s availability for $\mathcal{G}^k_\psi(s_b)$ during the last iteration, then marked it, and then determined it to be unavailable for $\mathcal{G}^k_\psi(s_b)$. Moreover, due to "linear marking", this mark on $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^k_\psi(s_b))$ should be there after the last iteration of ORA. Thus, we can conclude that $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^k_\psi(s_b))$ is marked at the end of the last iteration of the ORA algorithm.

We now prove the second statement by contradiction. If $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^k_\psi(s_b))$ is $\emptyset$, then $\mathcal{G}^k_\psi(s_b)$ will choose $\emptyset$ in the last iteration since it can offer $C_{\text{R}}(\mathcal{G}^k_\psi(s_b), \mathcal{R}_{\hat{\psi}}(\mathcal{G}^k_\psi(s_b)) > C_{\text{min}}$, and finally s_b will be able to get a better relay node. But this contradicts with the fact that this last iteration is a non-improving iteration. So, $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^k_\psi(s_b))$ cannot be $\emptyset$. Further, since we proved that $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^k_\psi(s_b))$ is marked at the end of the last iteration of the ORA algorithm, it must be assigned to some source node already. ■

Referring to Fig. 6, we have Claim 3 for a set of relay nodes $\mathcal{R}_{\hat{\psi}}(s_b), \mathcal{R}_{\hat{\psi}}(\mathcal{G}_\psi(s_b)), \cdots, \mathcal{R}_{\hat{\psi}}(\mathcal{G}^n_\psi(s_b))$. Our recursive procedure terminates at $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^n_\psi(s_b))$ because its assigned source node in solution ψ is a node in $\{s_b, \mathcal{G}_\psi(s_b), \cdots, \mathcal{G}^n_\psi(s_b)\}$. 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: Under Claim 3, we proved that the relay node $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^n_\psi(s_b))$ is assigned to some source node in solution ψ obtained by ORA. Since our recursive procedure terminates at $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^n_\psi(s_b))$, its assigned source node in solution ψ is a node in $\{s_b, \mathcal{G}_\psi(s_b), \cdots, \mathcal{G}^n_\psi(s_b)\}$. But we also know that under ψ, source nodes $\mathcal{G}_\psi(s_b)$, $\mathcal{G}^2_\psi(s_b)$, $\mathcal{G}^3_\psi(s_b)$, $\cdots$, $\mathcal{G}^n_\psi(s_b)$ have relay nodes $\mathcal{R}_{\hat{\psi}}(s_b)$, $\mathcal{R}_{\hat{\psi}}(\mathcal{G}_\psi(s_b))$, $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^2_\psi(s_b))$, $\cdots$, $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^{n-1}_\psi(s_b))$, respectively. Thus, $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^n_\psi(s_b))$ is the only relay node that can be assigned to s_b in solution ψ. On the other hand, relay node assigned to s_b in solution ψ is denoted by $\mathcal{R}_\psi(s_b)$. Thus, we have $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^n_\psi(s_b)) = \mathcal{R}_\psi(s_b)$.

Now, Claim 3 states that $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^n_\psi(s_b))$ must be marked after the last iteration, whereas Fact 1 states that the relay node assigned to the bottleneck source node, i.e., $\mathcal{R}_\psi(s_b)$, cannot be marked. Since both $\mathcal{R}_\psi(s_b)$ and $\mathcal{R}_{\hat{\psi}}(\mathcal{G}^n_\psi(s_b))$ are the same relay node, we have a contradiction. Thus our assumption that there exists a solution $\hat{\psi}$ better than ψ does not hold. The proof is complete. ■

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

Corollary 1.1: Under any feasible initial relay node assignment, the ORA algorithm can find an optimal relay node assignment.

VII. Numerical Results

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

A. Simulation Setting

We consider a 100-node cooperative ad hoc network. The location of each node is given in Table III. 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 30 source-destination pairs and 40 relay nodes. While in the second case, we have 40 source-destination pairs and only 20 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, with details given in Table III.

For the simulations, we assume $W = 10$ MHz bandwidth for each channel. The maximum transmission power at each node is set to 1 W. Each relay node employs AF for cooperative communications. 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. Note that the working of the ORA algorithm does not depend on the mode of CC and the channel gain model. As long as channel gains and achievable rates are known, ORA will give optimal assignment. For the AWGN channel, we assume the variance of noise is 10^{-10} W at all nodes.

TABLE III
LOCATIONS AND ROLES OF ALL THE NODES IN THE NETWORK.

Node Location	Role Case 1	Role Case 2	Node Location	Role Case 1	Role Case 2	Node Location	Role Case 1	Role Case 2
$(75, 500)$	s_1	s_1	$(220, 190)$	d_4	d_4	$(380, 370)$	r_7	s_{31}
$(170, 430)$	s_2	s_2	$(660, 190)$	d_5	d_5	$(300, 350)$	r_8	r_8
$(170, 500)$	s_3	s_3	$(430, 630)$	d_6	d_6	$(410, 650)$	r_9	s_{33}
$(250, 650)$	s_4	s_4	$(180, 620)$	d_7	d_7	$(470, 500)$	r_{10}	d_{40}
$(400, 550)$	s_5	s_5	$(750, 625)$	d_8	d_8	$(660, 525)$	r_{11}	s_{39}
$(340, 230)$	s_6	s_6	$(310, 480)$	d_9	d_9	$(600, 425)$	r_{12}	s_{40}
$(390, 150)$	s_7	s_7	$(1100, 180)$	d_{10}	d_{10}	$(510, 200)$	r_{13}	s_{38}
$(460, 280)$	s_8	s_8	$(1110, 360)$	d_{11}	d_{11}	$(575, 325)$	r_{14}	r_{14}
$(700, 500)$	s_9	s_9	$(875, 600)$	d_{12}	d_{12}	$(750, 560)$	r_{15}	r_{15}
$(750, 360)$	s_{10}	s_{10}	$(700, 300)$	d_{13}	d_{13}	$(800, 360)$	r_{16}	r_{16}
$(800, 90)$	s_{11}	s_{11}	$(650, 550)$	d_{14}	d_{14}	$(860, 260)$	r_{17}	r_{17}
$(900, 160)$	s_{12}	s_{12}	$(740, 170)$	d_{15}	d_{15}	$(980, 450)$	r_{18}	r_{18}
$(1125, 300)$	s_{13}	s_{13}	$(410, 810)$	d_{16}	d_{16}	$(950, 310)$	r_{19}	r_{19}
$(1000, 340)$	s_{14}	s_{14}	$(550, 1100)$	d_{17}	d_{17}	$(950, 200)$	r_{20}	d_{37}
$(1025, 540)$	s_{15}	s_{15}	$(150, 790)$	d_{18}	d_{18}	$(100, 1000)$	r_{21}	s_{32}
$(100, 1120)$	s_{16}	s_{16}	$(210, 1110)$	d_{19}	d_{19}	$(310, 980)$	r_{22}	r_{12}
$(150, 920)$	s_{17}	s_{17}	$(530, 720)$	d_{20}	d_{20}	$(250, 800)$	r_{23}	d_{32}
$(330, 1110)$	s_{18}	s_{18}	$(800, 1140)$	d_{21}	d_{21}	$(460, 1010)$	r_{24}	r_{13}
$(450, 890)$	s_{19}	s_{19}	$(1080, 1100)$	d_{22}	d_{22}	$(610, 930)$	r_{25}	d_{34}
$(650, 1050)$	s_{20}	s_{20}	$(940, 790)$	d_{23}	d_{23}	$(680, 760)$	r_{26}	s_{34}
$(700, 640)$	s_{21}	s_{21}	$(1360, 640)$	d_{24}	d_{24}	$(700, 900)$	r_{27}	r_{20}
$(820, 880)$	s_{22}	s_{22}	$(1280, 1120)$	d_{25}	d_{25}	$(910, 1120)$	r_{28}	d_{35}
$(1150, 1060)$	s_{23}	s_{23}	$(1260, 350)$	d_{26}	d_{26}	$(970, 970)$	r_{29}	s_{35}
$(1480, 1120)$	s_{24}	s_{24}	$(1500, 50)$	d_{27}	d_{27}	$(1360, 910)$	r_{30}	r_9
$(1160, 720)$	s_{25}	s_{25}	$(1450, 605)$	d_{28}	d_{28}	$(1200, 920)$	r_{31}	r_{11}
$(1050, 50)$	s_{26}	s_{26}	$(1030, 910)$	d_{29}	d_{29}	$(1250, 690)$	r_{32}	d_{36}
$(1350, 450)$	s_{27}	s_{27}	$(1150, 230)$	d_{30}	d_{30}	$(1290, 180)$	r_{33}	r_{10}
$(1380, 110)$	s_{28}	s_{28}	$(80, 370)$	r_1	d_{31}	$(150, 360)$	r_{34}	r_5
$(1500, 800)$	s_{29}	s_{29}	$(110, 280)$	r_2	r_2	$(1380, 380)$	r_{35}	r_7
$(1500, 300)$	s_{30}	s_{30}	$(160, 300)$	r_3	r_3	$(1220, 60)$	r_{36}	s_{37}
$(200, 50)$	d_1	d_1	$(280, 520)$	r_4	r_4	$(1190, 510)$	r_{37}	s_{36}
$(520, 240)$	d_2	d_2	$(375, 580)$	r_5	d_{39}	$(500, 40)$	r_{38}	d_{38}
$(40, 100)$	d_3	d_3	$(385, 450)$	r_6	r_6	$(50, 805)$	r_{39}	d_{33}
						$(1510, 920)$	r_{40}	r_1

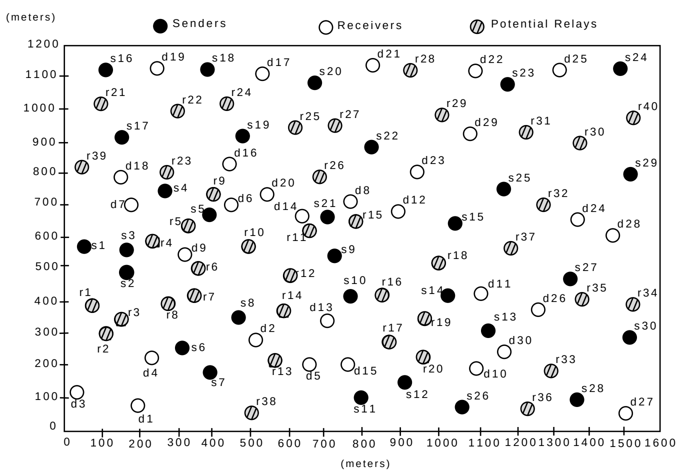

Fig. 7. Topology for a 100-node network for Case 1 ($N_r \geq N_s$), with $N_s = 30$ and $N_r = 40$.

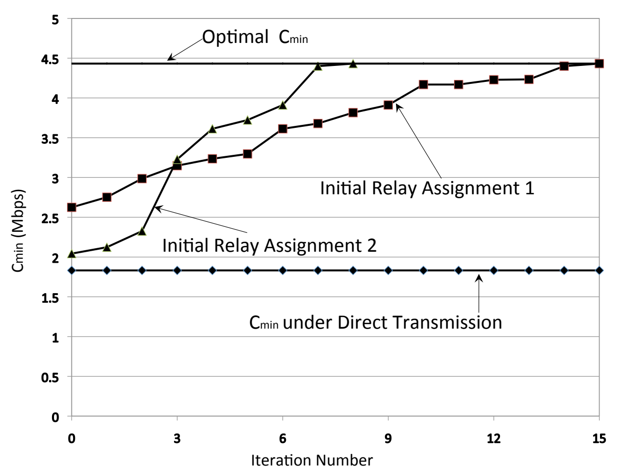

Fig. 8. Case 1 ($N_r \geq N_s$): The objective value $C_{\min}$ at each iteration of ORA algorithm under two different initial relay node assignments.

B. Results

Case 1: $N_r \geq N_s$**.** In this case (see Fig. 7), we have 30 source-destination pairs and 40 relay nodes.

Under ORA, after preprocessing, we start with an initial relay node assignment in the first iteration. Such initial assignment is not unique. But regardless of the initial relay node assignment, we expect the objective value to converge to the optimum (by Corollary 1.1). To validate this result, in Table IV, we show the results of running the ORA algorithm under two different initial relay node assignments, denoted as I and II (see Table IV).

In Table IV, the second column shows the data rate for each source-destination pair under direct transmissions. Note that the minimum rate among all pairs is 1.83 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 that initial relay node assignments I and II are different. As a result, the final assignment is different under I and II. However, the final

TABLE IV
OPTIMAL ASSIGNMENTS FOR CASE 1 ($N_r \geq N_s$) UNDER TWO DIFFERENT INITIAL RELAY NODE ASSIGNMENTS.

Session	C_D (Mbps)	Relay Assignment I			Relay Assignment II		
		Initial	Final	Final Rate (Mbps)	Initial	Final	Final Rate (Mbps)
s_1	2.62	$\emptyset$	r_3	6.54	r_3	r_3	6.54
s_2	4.60	r_8	r_7	9.46	r_8	r_7	9.46
s_3	3.81	$\emptyset$	r_2	8.73	r_1	r_1	7.21
s_4	2.75	$\emptyset$	r_4	4.66	r_4	r_4	4.66
s_5	3.15	$\emptyset$	r_{14}	6.47	r_7	r_{14}	6.47
s_6	4.17	$\emptyset$	r_6	9.25	r_{10}	r_6	9.25
s_7	$\boxed{1.83}$	r_6	r_8	4.76	r_6	r_8	4.76
s_8	2.99	$\emptyset$	r_{12}	7.22	r_{16}	r_{12}	7.22
s_9	4.92	r_{12}	r_{10}	9.81	r_{12}	r_{10}	9.81
s_{10}	4.80	r_{18}	$\emptyset$	4.80	$\emptyset$	$\emptyset$	4.80
s_{11}	4.13	r_{16}	r_{20}	9.13	r_{17}	r_{20}	9.13
s_{12}	3.23	$\emptyset$	r_{19}	5.89	r_{18}	r_{18}	5.55
s_{13}	3.68	$\emptyset$	r_{18}	4.84	r_{19}	r_{17}	7.32
s_{14}	4.23	$\emptyset$	r_{16}	7.87	r_{15}	r_{15}	5.29
s_{15}	2.62	r_{17}	r_{17}	4.86	r_{20}	r_{19}	5.84
s_{16}	3.30	$\emptyset$	r_{22}	7.29	r_{22}	r_{22}	7.29
s_{17}	4.17	$\emptyset$	r_{24}	5.62	r_{24}	r_{24}	5.62
s_{18}	6.03	r_{21}	r_{21}	7.37	r_{23}	r_{23}	6.26
s_{19}	8.76	$\emptyset$	$\emptyset$	8.76	$\emptyset$	$\emptyset$	8.76
s_{20}	6.95	$\emptyset$	$\emptyset$	6.95	$\emptyset$	$\emptyset$	6.95
s_{21}	1.90	r_{27}	r_{27}	4.90	r_{27}	r_{27}	4.90
s_{22}	7.65	r_{28}	r_{28}	8.71	r_{28}	r_{28}	8.71
s_{23}	7.55	r_{29}	r_{29}	11.26	r_{29}	r_{28}	11.26
s_{24}	2.12	r_{40}	r_{40}	$\boxed{4.43}$	$\emptyset$	r_{40}	$\boxed{4.43}$
s_{25}	3.90	$\emptyset$	r_{30}	5.87	$\emptyset$	r_{30}	5.87
s_{26}	6.08	r_{36}	r_{36}	6.81	r_{36}	r_{36}	6.81
s_{27}	3.61	$\emptyset$	r_{34}	5.44	$\emptyset$	r_{34}	5.44
s_{28}	2.04	r_{35}	r_{35}	5.29	$\emptyset$	r_{35}	5.29
s_{29}	2.32	r_{30}	r_{31}	4.68	$\emptyset$	r_{31}	4.68
s_{30}	6.60	r_{34}	r_{33}	9.65	r_{34}	r_{33}	9.65

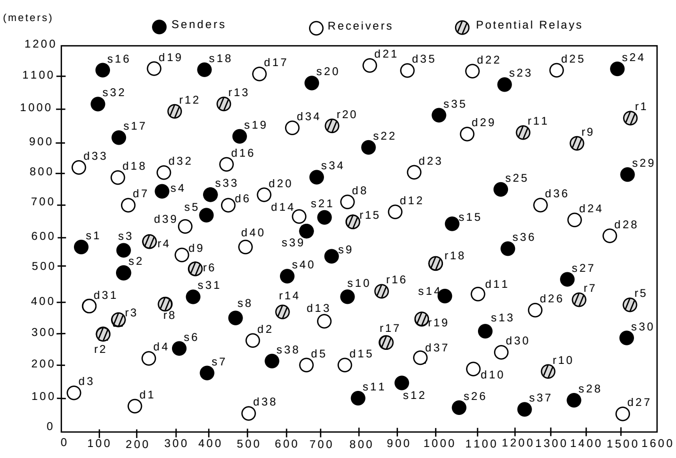

Fig. 9. Topology for a 100-node network for Case 2 ($N_r < N_s$), with $N_s = 40$ and $N_r = 20$.

objective value (i.e., $C_{\min}$) under I and II is identical (4.43 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. Note that a higher initial value of $C_{\min}$ does not mean that ORA will converge faster. The increase of $C_{\min}$ by cooperative communications over direct transmissions is significant (from 1.83 Mbps to 4.43 Mbps).

Case 2: $N_r < N_s$. In this case (see Fig. 9), we have 40

TABLE V
OPTIMAL ASSIGNMENTS FOR CASE 2 ($N_r < N_s$) UNDER TWO DIFFERENT INITIAL RELAY NODE ASSIGNMENTS.

Session	C_D (Mbps)	Relay Assignment I			Relay Assignment II		
		Initial	Final	Final Rate (Mbps)	Initial	Final	Final Rate (Mbps)
s_1	2.62	$\emptyset$	r_2	6.62	r_3	r_3	6.54
s_2	2.60	$\emptyset$	$\emptyset$	4.60	$\emptyset$	$\emptyset$	4.60
s_3	3.81	$\emptyset$	$\emptyset$	3.81	$\emptyset$	$\emptyset$	3.81
s_4	2.75	$\emptyset$	r_5	4.66	r_8	r_5	5.20
s_5	3.15	$\emptyset$	r_6	$\boxed{3.80}$	r_{14}	r_6	$\boxed{3.80}$
s_6	4.17	$\emptyset$	$\emptyset$	4.17	$\emptyset$	$\emptyset$	4.16
s_7	$\boxed{1.83}$	$\emptyset$	r_8	4.76	r_6	r_8	4.76
s_8	2.99	$\emptyset$	r_{14}	4.43	$\emptyset$	r_{14}	4.43
s_9	4.92	$\emptyset$	$\emptyset$	4.92	$\emptyset$	$\emptyset$	4.92
s_{10}	4.80	$\emptyset$	$\emptyset$	4.80	$\emptyset$	$\emptyset$	4.80
s_{11}	4.13	$\emptyset$	$\emptyset$	4.13	$\emptyset$	$\emptyset$	4.13
s_{12}	3.23	$\emptyset$	r_{18}	5.55	$\emptyset$	r_{18}	5.55
s_{13}	3.68	$\emptyset$	r_{19}	8.04	$\emptyset$	r_{19}	8.04
s_{14}	4.23	$\emptyset$	$\emptyset$	4.23	$\emptyset$	$\emptyset$	4.23
s_{15}	2.62	$\emptyset$	r_{16}	5.60	$\emptyset$	r_{16}	5.60
s_{16}	3.30	$\emptyset$	r_{12}	7.30	$\emptyset$	r_{12}	7.30
s_{17}	4.17	$\emptyset$	$\emptyset$	4.17	$\emptyset$	$\emptyset$	4.17
s_{18}	6.03	$\emptyset$	$\emptyset$	6.03	r_{13}	$\emptyset$	6.03
s_{19}	8.76	$\emptyset$	$\emptyset$	8.76	r_{12}	r_{13}	8.97
s_{20}	6.95	$\emptyset$	$\emptyset$	6.95	r_{20}	$\emptyset$	6.95
s_{21}	1.90	$\emptyset$	r_{20}	4.90	$\emptyset$	r_{20}	4.90
s_{22}	7.65	$\emptyset$	$\emptyset$	7.65	$\emptyset$	$\emptyset$	7.65
s_{23}	7.55	$\emptyset$	$\emptyset$	7.55	r_{11}	$\emptyset$	7.55
s_{24}	2.12	$\emptyset$	r_9	5.15	$\emptyset$	r_9	5.15
s_{25}	3.91	$\emptyset$	$\emptyset$	3.91	$\emptyset$	$\emptyset$	3.91
s_{26}	6.08	$\emptyset$	$\emptyset$	6.08	r_{10}	$\emptyset$	6.08
s_{27}	3.61	$\emptyset$	r_{10}	5.27	$\emptyset$	r_{10}	5.27
s_{28}	2.04	$\emptyset$	r_7	5.29	$\emptyset$	r_7	5.29
s_{29}	2.32	$\emptyset$	r_{11}	4.68	$\emptyset$	r_{11}	4.68
s_{30}	6.60	$\emptyset$	$\emptyset$	6.60	$\emptyset$	$\emptyset$	6.60
s_{31}	11.06	$\emptyset$	$\emptyset$	11.06	$\emptyset$	$\emptyset$	11.06
s_{32}	17.47	$\emptyset$	$\emptyset$	17.47	$\emptyset$	$\emptyset$	17.47
s_{33}	4.86	$\emptyset$	$\emptyset$	4.86	$\emptyset$	$\emptyset$	4.86
s_{34}	31.34	$\emptyset$	$\emptyset$	31.34	$\emptyset$	$\emptyset$	31.34
s_{35}	37.87	$\emptyset$	$\emptyset$	37.87	$\emptyset$	$\emptyset$	37.87
s_{36}	29.79	$\emptyset$	$\emptyset$	29.79	$\emptyset$	$\emptyset$	29.79
s_{37}	10.65	$\emptyset$	$\emptyset$	10.65	$\emptyset$	$\emptyset$	10.65
s_{38}	38.27	$\emptyset$	$\emptyset$	38.27	$\emptyset$	$\emptyset$	38.27
s_{39}	12.10	$\emptyset$	$\emptyset$	12.10	$\emptyset$	$\emptyset$	12.10
s_{40}	41.70	$\emptyset$	$\emptyset$	41.70	$\emptyset$	$\emptyset$	41.70

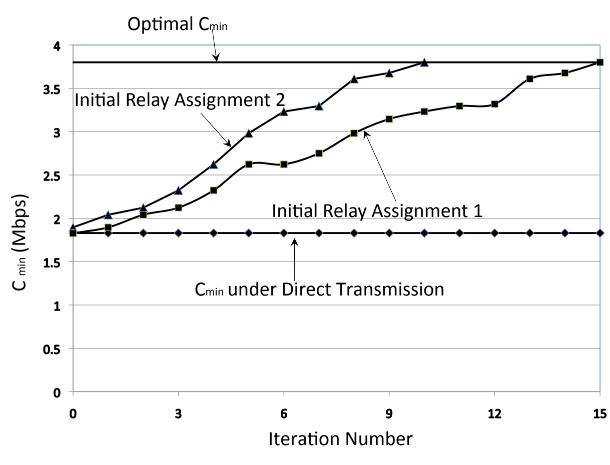

Fig. 10. Case 2 ($N_r < N_s$): The objective value $C_{\min}$ at each iteration of ORA algorithm under two different initial node assignments.

source-destination pairs and 20 relay nodes.

Table V shows the results of this case under two different initial relay node assignments I and II. The second column

TABLE VI
AN EXAMPLE ILLUSTRATING THE IMPORTANCE OF PREPROCESSING.

Sender	C_D (Mbps)	Without Preprocessing		
		Initial	Final	Final Rate (Mbps)
s_1	2.62	r_3	r_3	6.54
s_2	4.60	$\emptyset$	$\emptyset$	4.60
s_3	3.81	$\emptyset$	r_2	8.73
s_4	2.75	r_8	r_4	4.66
s_5	3.15	r_{14}	r_{14}	6.47
s_6	4.17	$\emptyset$	r_6	9.25
s_7	1.83	r_6	r_8	4.76
s_8	2.99	$\emptyset$	r_{12}	7.22
s_9	4.92	$\emptyset$	$\emptyset$	4.92
s_{10}	4.80	$\emptyset$	$\emptyset$	4.80
s_{11}	4.13	$\emptyset$	r_{20}	9.13
s_{12}	3.24	$\emptyset$	r_{18}	5.55
s_{13}	3.68	$\emptyset$	r_{17}	7.32
s_{14}	4.23	$\emptyset$	r_{16}	7.87
s_{15}	2.62	$\emptyset$	r_{19}	5.84
s_{16}	3.30	$\emptyset$	r_{22}	7.30
s_{17}	4.17	$\emptyset$	r_{24}	5.62
s_{18}	6.03	r_{23}	r_{21}	7.37
s_{19}	**8.76**	r_{39}	r_{39}	**4.81**
s_{20}	6.95	r_{26}	r_{26}	7.25
s_{21}	1.90	$\emptyset$	r_{27}	4.90
s_{22}	7.65	r_{28}	r_{28}	8.71
s_{23}	7.55	r_{29}	r_{29}	11.26
s_{24}	2.12	$\emptyset$	r_{40}	$\boxed{4.43}$
s_{25}	3.91	$\emptyset$	r_{30}	5.87
s_{26}	6.08	r_{33}	r_{33}	7.55
s_{27}	3.61	$\emptyset$	r_{34}	5.45
s_{28}	2.04	$\emptyset$	r_{35}	5.29
s_{29}	2.33	$\emptyset$	r_{31}	4.68
s_{30}	6.60	$\emptyset$	$\emptyset$	6.60

in Table V lists the data rate under direct transmissions. As discussed at the end of Section V-B, for the case of $N_r < N_s$, it is only necessary to consider relay node assignment for $N_r = 20$ source nodes corresponding to the 20 smallest achievable rates under direct transmission.

Again in Table V, the objective value $C_{\min}$ is identical (3.80 Mbps) regardless of different initial relay node assignments (I and II). Note that despite the difference in final relay node assignments under I and II, the objective value $C_{\min}$ is identical. The increase of $C_{\min}$ by cooperative communications over direct transmissions is significant (from 1.83 Mbps to 3.80 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 both initial relay node assignments I and II.

Significance of Preprocessing: Now we use a set of numerical results to show the significance of preprocessing in our ORA algorithm. We consider the same network in Fig. 7 with 30 source-destination pairs and 40 relay nodes. Now we remove the preprocessing step in the ORA algorithm. As an example, the third column of Table VI shows an initial assignment without first going through the preprocessing step. Although the objective value $C_{\min}$ also reaches the same optimal value (4.43 Mbps) as that in Table IV, the final data rate for some non-bottleneck source nodes could be worse than direct transmissions. For example, for s_{19}, its final rate is 4.81 Mbps, which is less than its direct transmission rate (8.76 Mbps). Such event is undetectable without the preprocessing step, as 4.81 Mbps is still greater than the optimal objective value (4.43 Mbps).

On the other hand, when the preprocessing step is employed, ORA can ensure that the final rate for each source-destination pair is no less than that under direct transmission, as shown in Table IV.

VIII. A SKETCH OF A POSSIBLE IMPLEMENTATION

In this section, we present a sketch of a possible implementation of the ORA algorithm. This implementation follows the link-state approach. Note that although a link-state approach is not considered fully distributed, it is nevertheless a viable implementation, as evidenced by the widespread deployment of OSPF [12] in the Internet and acceptance of OLSR [3] in wireless ad hoc networks.

A. Ensuring Identical Optimal Solution at Source Nodes

In the presentation of the ORA algorithm in Section V, we have learned that the ORA algorithm can start with any random initial relay node assignment and can still obtain an optimal solution. However, in the link-state based implementation, each source node in the network will run ORA independently on its own. As such, the randomness in initial relay assignment must be removed in implementation so as to ensure that each source node can obtain an identical optimal solution. Otherwise, we may run into a situation that the same relay node may be assigned to multiple source nodes.

A simple way to ensure identical initial relay node assignment is to have each source node choose direct transmission, i.e. $\emptyset$ as its initial relay assignment. Given such identical initial assignment and that ORA is a deterministic algorithm, each source node will obtain an identical final optimal solution.

B. Some Implementation Details

Under such implementation, each relay node collects its link state information with its neighboring source nodes; each destination node also collects its link state information with its source node and neighboring relay nodes. To do this, each source node sends a broadcast packet to its neighboring relay nodes and its destination node; each relay node sends a broadcast packet to its neighboring destination nodes. As shown in [6] by Gollakota and Katabi, this broadcast packet transmission can be used by the receiver of each wireless link to accurately determine the link state. It was also shown in [6] that such an approach is practically feasible, and was demonstrated in their implementation of 802.11 receivers. However, we point out that in an uncontrolled environment, estimating channel state is not trivial.

Upon obtaining the link-state information, each relay and destination node will distribute such information to all the source nodes in the network. This will ensure that each source node will have global link-state information. Such link-state dissemination can be achieved by using one of the many efficient flooding techniques (see e.g. [23]) for wireless networks.

The overhead of this operation is small when compared to potential gain in achievable rate in optimal assignment (see Section VIII-C).

Once each source node has global link-state information, it can now run ORA locally, with an identical initial assignment as discussed in Section VIII-A. As discussed, the final optimal solution obtained at each source node will be identical.

C. Overhead

An important consideration in our implementation is the overhead incurred in distributing link-state information in the network. This can be measured by comparing such overhead with the potential gain in achievable rate in the optimal solution. We now analyze such overhead and show that the ratio between the two is small, thus affirming the efficacy of our proposed implementation.

First of all, Mandke *et al.* [11] conducted extensive experiments and showed that the CSI for the 10 MHz band in 2.4 GHz spectrum changes every 300 mSec on average. Their experiments showed that CSI distribution does not need to be performed more frequently than every 300 mSec (or 0.3 Sec).

To run the ORA algorithm locally, each source node must obtain global link-state information. This can be done by having all relay nodes and destination nodes flood their local link-state information in the network. To estimate an upper bound for such flooding overhead, we assume that 32 bits (commonly used for floating variables) are used to represent each link-state value. Then the total link-state information collected at each relay node has $O_r = N_s \times 32$ bits. Similarly, the total link-state information collected at each destination node has $O_d = (N_r + 1) \times 32$ bits. As a result, the total overhead (in b/s) due to flooding at every node is:

$$O = \frac{N_r \cdot O_r + N_d \cdot O_d}{0.3}. \quad (12)$$

As an example, for Case 1 in the numerical results in Section VII-B, it can be shown that the total overhead is 170.83 Kb/s at each node. On the other hand, the gain in the bottleneck data rate by ORA is $4.43 - 1.83 = 2.6$ Mb/s. The ratio between the two is only 6.4%. That is, the overhead is much less than the gain of CC.

We acknowledge that in some environments, the overhead could be large if CSI in the network varies on a smaller time scale. Under such environment, fast and efficient dissemination of CSI remains an open problem.

IX. Conclusion

Cooperative communications is a powerful communication paradigm to achieve spatial diversity. However, the performance of such communication paradigm hinges upon the assignment of relay nodes in the network. In this paper, we studied this problem in a cooperative ad hoc network environment, where multiple source-destination pairs compete for the same pool of relay nodes. Our objective is to assign the available relay nodes to different source-destination pairs so as to maximize the minimum data rate among all the pairs. The main contribution of this paper is a polynomial time optimal algorithm that achieves this objective. A novel 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 efficacy.

Although we offered a sketch of a possible implementation of ORA, a number of issues remain challenging in practice. In particular, fast and efficient method for collecting and disseminating CSI in moderate and large sized networks remain an open problem. Nevertheless, the theoretical results presented here can be used as a performance benchmark for other proposed solutions in practice.

Acknowledgments

The authors thank Editor Suhas Diggavi, who handled the review of this paper. The authors also thank the anonymous reviewers who offered very constructive comments on improving the presentation of this paper. Y.T. Hou, S. Sharma, and Y. Shi were supported in part by National Science Foundation under Grant CCF-0946273 and Naval Research Laboratory under Grant N00173-10-1-G-007. S. Kompella has been supported in part by the Office of Naval Research.

References

[1] 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.

[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 Transactions on Wireless Communication,* vol. 7, no. 4, pp. 1348–1357, April 2008.

[3] T. Clause and P. Jacquet, “Optimized link state routing protocol,” *IETF RFC 3626,* October 2003.

[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] M.O. Damen and A.R. Hammons, “Delay-tolerant distributed-TAST codes for cooperative diversity,” *IEEE Transactions on Information Theory,* vol. 53, no. 10, pp. 3755–3773, Oct. 2007.

[6] S. Gollakota and D. Katabi, “Zigzag decoding: combating hidden terminals in wireless networks,” In *Proceedings of ACM SIGCOMM,* pp. 159–170, Seattle, WA, August 17–22, 2008.

[7] D. Gunduz and E. Erkip, “Opportunistic cooperation by dynamic resource allocation,” *IEEE Transactions on Wireless Communications,* vol. 6, no. 4, pp. 1446–1454, April 2007.

[8] O. Gurewitz, A. de Baynast, and E.W. Knightly, “Cooperative strategies and achievable rate for tree networks with optimal spatial reuse,” *IEEE Transactions on Information Theory,* vol. 53, no. 10, pp. 3596–3614, Oct. 2007.

[9] A.E. Khandani, J. Abounadi, E. Modiano, and L. Zheng, “Cooperative routing in static wireless networks,” *IEEE Transactions on Communications,* vol. 55, no. 11, pp. 2185–2192, Nov. 2007.

[10] 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.

[11] K. Mandke, R.C. Daniels, S. Choi, S.M. Nettles, and R.W. Heath Jr., “Physical concerns for cross-layer prototyping and wireless network experimentation,” In *Proc. ACM Int’l Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization* (in conjunction with ACM MobiCom 2007), pp. 11–18, Montreal, Quebec, Canada, Sept. 10, 2007.

[12] J. Moy, “Open shortest path first version 2,” *IETF RFC 2328,* April 1998.

[13] T. C-Y. Ng and W. Yu, "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.

[14] S. Savazzi, and U. Spagnolini, "Energy aware power allocation strategies for multihop-cooperative transmission schemes," *IEEE Journal on Selected Areas in Communications,* 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 Magazine,* vol. 23, no. 5, pp. 18–29, Sept. 2006.

[16] 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.

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

[18] L. Tassiulas and A. Ephremides. "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control,* vol. 37, no. 12, pp. 1936–1948, 1992.

[19] E.C. van der Meulen and P. Vanroose. "The capacity of a relay channel, both with and without delay," *IEEE Transactions on Information Theory,* vol. 53, no. 10, pp. 3774–3776, October 2007.

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

[21] 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. IEEE INFOCOM*, pp. 544–552, Anchorage, AL, May 6–12, 2007.

[22] E.M. Yeh and R.A. Berry, "Throughput optimal control of cooperative relay networks," *IEEE Transactions on Information Theory,* vol. 53, no. 10, pp. 3827–3833, Oct. 2007.

[23] Y. Yi, M. Gerla, and T.J. Kwon, "Efficient flooding in ad hoc networks: a comparative performance study," *IEEE International Conference on Communications,* vol. 2, pp. 1059-1063, May 11-15 2003.

[24] 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.

Sushant Sharma (S'06) received his B.E. degree in computer engineering from the University of Pune, India, in 2002, and the M.S. degree in computer science from the University of New Mexico, Albuquerque, in 2005. He is currently pursuing the Ph.D degree in Computer Science at Virginia Polytechnic Institute and State University ("Virginia Tech"), Blacksburg, VA. His research interests include developing algorithms to solve cross-layer optimization problems in ad hoc wireless networks.

Yi Shi (S'02–M'08) received his B.S. degree from the University of Science and Technology of China, Hefei, China, in 1998, a M.S. degree from Institute of Software, Chinese Academy of Science, Beijing, China, in 2001, a second M.S. degree from Virginia Polytechnic Institute and State University ("Virginia Tech"), Blacksburg, VA, in 2003, all in computer science, and a Ph.D. degree in computer engineering from Virginia Tech, in 2007. He is currently a Research Scientist in the Department of Electrical and Computer Engineering at Virginia Tech. Dr. Shi's research focuses on algorithms and optimization for cognitive radio networks, MIMO and cooperative communication networks, sensor networks, and ad hoc networks. He was a recipient of IEEE INFOCOM 2008 Best Paper Award. He was a recipient of Chinese Government Award for Outstanding Ph.D. Students Abroad in 2006. While an undergraduate, he was a recipient of Meritorious Award in International Mathematical Contest in Modeling in 1997 and 1998, respectively. He served as a TPC member for many major international conferences (including ACM MobiHoc 2009 and IEEE INFOCOM 2009–2011).

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

Prof. Hou's research interests are cross-layer design and optimization for cognitive radio wireless networks, cooperative communications, MIMO-based ad hoc networks, and new interference management schemes for wireless networks. He is a recipient of an Office of Naval Research (ONR) Young Investigator Award (2003) and a National Science Foundation (NSF) CAREER Award (2004) for his research on optimizations and algorithm design for wireless ad hoc and sensor networks.

Prof. Hou is currently serving as an Area Editor of *IEEE Transactions on Wireless Communications*, and Editor for *IEEE Transactions on Mobile Computing, IEEE Wireless Communications, ACM/Springer Wireless Networks (WINET),* and *Elsevier Ad Hoc Networks.* He was a past Associate Editor of *IEEE Transactions on Vehicular Technology.* He was Technical Program Co-Chair of IEEE INFOCOM 2009.

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

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