Maintaining Social Links through Amplify-and-Forward in Wireless Networks

Li Qiu
Department of Computer Science and Engineering
The Pennsylvania State University
State College, PA, USA
lyq5023@cse.psu.edu

Guohong Cao
Department of Computer Science and Engineering
The Pennsylvania State University
State College, PA, USA
gcao@cse.psu.edu

ABSTRACT

It is critical to maintain the communication links between important social pairs. However, maintaining the social links between faraway nodes in wireless networks is extremely difficult. Although multi-hop transmission can be used, if two nodes in the routing path are out of the wireless transmission range, a network partition is possible. To address this problem, we adopt a cooperative amplify-and-forward strategy, where nodes (relays) cooperate to improve the signal strength at the destination. We formulate and study two optimization problems for maintaining the required link throughput: Min-Energy and Min-Relay, where the goal of Min-Energy is to minimize the power consumption of the relays, and the goal of Min-Relay is to minimize the number of active relays. Since the Min-Energy problem is a non-convex problem, we solve it based on an approximation technique and prove that our solution is a feasible, in fact optimal solution. We formulate the Min-Relay problem as an integer programming problem, and propose a polynomial-time algorithm which can select the minimum number of relays to maintain the social link. Evaluation results show that Min-Relay can significantly reduce the number of active relays compared to Min-Energy, while achieving comparable power consumption.

CCS CONCEPTS

•Networks →Ad hoc networks;

KEYWORDS

amplify-and-forward, maintain social links, energy saving, relay selection

ACM Reference format:
Li Qiu and Guohong Cao. 2017. Maintaining Social Links through Amplify-and-Forward in Wireless Networks. In *Proceedings of Mobihoc '17, Chennai, India, July 10-14, 2017,* 10 pages.
DOI: http://dx.doi.org/10.1145/3084041.3084055

1 INTRODUCTION

In the past decade, many researchers have designed various routing algorithms for mobile ad hoc networks [10, 16, 22]. In sparse mobile ad hoc networks, where the node density is low, and contacts between the nodes in the network do not occur frequently, it is hard to maintain end-to-end connections. As a result, "carry-and-forward" is used, where mobile nodes physically carry the data, and forward the data when contacting a node with higher forwarding capability. To improve the performance of data forwarding, recent research [3, 8, 12, 23] focuses on exploiting social knowledge for data forwarding, because social knowledge is more reliable and less susceptible to the randomness of node mobility.

Although social cognitive techniques can be exploited to enhance the robustness and performance of mobile ad hoc networks, existing research focuses on improving the performance of data forwarding (or routing) for all nodes in the network, without differentiating which node pairs are more important. However, in many real scenarios, the communications between some nodes are more important than others. For example, in a battle field, for a platoon of soldiers consisting of three squads, the commander of the platoon should maintain good connections with the squad leaders, but not necessarily with all other soldiers. Then, it is more important to maintain the social links between the commander and the squad leaders, even though this may be at the cost of sacrificing some communication performance with other nodes.

There are many research challenges on maintaining important social links in wireless ad hoc networks, especially when the nodes are far away from each other. Although multi-hop transmission can be used, if two nodes in the routing path are out of the transmission range, a network partition is possible. Increasing the transmission power level can solve part of the problem, but this approach has its limitations since there is always a maximum transmission range, out of which two nodes will not be able to communicate. Thus, even with the maximum transmission power level, it is possible that the social links cannot be maintained because two nodes along the routing path are far away from each other.

To address this problem, we adopt a cooperative amplify-and-forward strategy, where nodes around the node that cannot reach the next hop node (or *link source* for simplicity) cooperate to transmit towards the next node in the routing path (or *link destination* for simplicity). More specifically, the source first broadcasts the data to its nearby nodes, which simultaneously amplifies and forwards their received signal to the destination. In this way, the destination can receive a much stronger signal, from which it can decode and obtain the data from the source, and accordingly the social links can be maintained.

There is some existing work [2, 4, 7, 19, 24] on amplify-and-forward. However, most of them focused on maximizing the data throughput and optimizing the power allocation; i.e., given the total

Mobihoc '17, Chennai, India
© 2017 ACM. 978-1-4503-4912-3/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3084041.3084055

transmission power at the relay nodes, how to optimally allocate the power for each relay, so that the data throughput between the source and destination is maximized. Different from existing work, maintaining social link only requires that the data throughput of the social link is greater than a threshold. Furthermore, to save energy, relays should transmit at the lowest power level that enables the required link throughput. Due to these differences, existing solutions on amplify-and-forward can not be directly applied.

In this paper, we study the problem of maintaining social links (i.e., link throughput larger than a threshold) while minimizing the power consumption of the relays, which is referred to as the *Min-Energy* problem. The difficulty of this problem lies in the fact that with many relays simultaneously transmitting to the destination, it is not obvious how increasing the transmission power at one relay will affect the link throughput between the source and destination. We first formulate the Min-Energy problem as a non-convex quadratic programming problem by exploiting the rate-distortion theory, and then solve it based on an approximation technique called semidefinite relaxation (SDR) [13]. We also prove that our solution can minimize the power consumption of the relays.

In our solution to the Min-Energy problem, all relays are involved. This will generate significant synchronization overhead, since relays have to synchronize their clocks for their signals to arrive at the destination simultaneously. To minimize the synchronization overhead, we also study the *Min-Relay* problem, which aims to minimize the number of active relays while maintaining the social link. We formulate the Min-Relay problem as an integer programming problem, and propose a polynomial-time algorithm which can select the minimum number of relays to maintain the social link. The main contributions of this paper are summarized as follows:

- We identify the problem of maintaining social links through amplify-and-forward in wireless networks, and formulate the Min-Energy problem and the Min-Relay problem.
- To solve the Min-Energy problem, we propose an optimal solution which can minimize the power consumption of the relays while maintaining the required link throughput.
- To solve the Min-Relay problem, we propose a polynomial-time algorithm which can efficiently select the minimum number of relays while maintaining the required link throughput.
- Evaluation results show that Min-Relay can significantly reduce the number of active relays compared to Min-Energy, while achieving comparable power consumption.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 presents the model and the problem formulation. We study the Min-Energy problem in Section 4 and the Min-Relay problem in Section 5. Section 6 presents performance evaluations and Section 7 concludes the paper.

2 RELATED WORK

In the past few years, researchers have proposed various routing and data forwarding algorithms for mobile ad hoc networks based on social knowledge. For example, in [8], the authors have exploited the community and centrality social metrics, and proposed a social-based forwarding algorithm to improve the performance of data forwarding. In [23], the authors have identified the existence of transient connected components in mobile social networks, and proposed a data forwarding strategy exploiting the user contacts in transient connected components. However, these existing works focus on improving the performance of data forwarding for all nodes in the network, and none of them considers maintaining important social links through amplify-and-forward.

Amplify-and-forward is a cooperative transmission strategy which has been used to transmit data to faraway nodes. Compared to the other well-known cooperative transmission approach such as decode-and-forward [14, 17] where data is decoded at the relays, amplify-and-forward requires less processing and it saves energy at the relays since the relays simply amplifies and transmits their received signal to the destination without decoding. Moreover, as shown in [20], amplify-and-forward has lower bit error rate since it does not decode the data at the relays.

The power allocation problem in amplify-and-forward has been studied by many researchers recently. Given the total transmission power at the relays, Zhao *et al.* derived the transmission power for each relay, such that the link throughput between the source and destination is maximized [24]. In [2], Ding *et al.* proposed a distributed power allocation strategy which determines the transmission power of each relay, to maximize the data throughput between the source and destination. These existing work [2, 4, 7, 24] focuses on maximizing the link throughput, where relays transmit data at their maximum power. Different from them, in our Min-Energy problem, our goal is to minimize the power consumption of the relays while maintaining the required link throughput.

In these aforementioned solutions, all relays are involved. Jing *et al.* [9] studied the problem of selecting a group of relays to help transmit the data to the destination, such that the SNR at the destination is maximized. Different from them, in our Min-Relay problem, our goal is to minimize the number of active relays, while maintaining the required link throughput. Also, [9] only gives sub-optimal solutions, while we propose a polynomial-time algorithm that can efficiently select the minimum number of relays while maintaining the required link throughput.

3 PRELIMINARIES

3.1 Model

Consider a social link, where the source node S wants to transmit data to a distant destination node D. As these two nodes are out of the normal wireless transmission range, node S relies on n other relays ($\{R_1, R_2, \ldots, R_n\}$) around it to help forward the data towards the destination. Let $d_{1,k}$ denote the distance between S and any relay node R_k, and let $d_{2,k}$ denote the distance between R_k and D.

Suppose the source node S sends out a signal s towards D, which follows a Guassian distribution with zero-mean and unit variance. S uses a fixed power P for transmission, while each of the relays can at most transmit at power Q. We consider a Guassian channel, where the received signal is the sum of the faded transmitted signal from all other nodes and additive white Gaussian noise. That is, the received signal Y_j at node j is given by

$$Y_j = \sum_{i \in \tau} \frac{X_i}{d_{ij}^r} + W_j,$$

where τ is the set of nodes that are simultaneously transmitting, and X_i is the signal transmitted by node i, and d_{ij} is the distance between node i and j, r is a positive parameter describing how the signal strength scales with distance, and W_j is the white noise at node j following a normal distribution $\sim \mathcal{N}(0, N)$. To simplify the notations, let α_k denote the signal scaling coefficients from S to the relay node R_k, i.e., $\alpha_k = 1/d_{1,k}^r$, for $k = 1, 2, ..., n$. Besides, let β_k denote the signal scaling coefficient from the R_k to D, which is given by $\beta_k = 1/d_{2,k}^r$, where $k = 1, 2, ..., n$.

In this paper, we consider an amplify-and-forward strategy, which takes two steps to transmit the signal from the source node to the destination node. In the first step, the source node broadcasts the signal to all relay nodes, where the signal received at a relay node R_i is $Z_i = s\alpha_i\sqrt{P} + W_i$.

In the second step, relay nodes amplify their received signal and forward to the destination. If R_i employs power P_i for transmission, the signal transmitted by R_i is

$$X_i = \sqrt{\frac{P_i}{\alpha_i^2 P + N}} Z_i = \sqrt{\frac{P_i}{\alpha_i^2 P + N}} \left(s\alpha_i\sqrt{P} + W_i \right).$$

The signal at the destination node is the sum of the signals from all the relay nodes, which is

$$Y = \sum_{i=1}^{n} \beta_i X_i + W_{n+1} = \sum_{i=1}^{n} \beta_i \sqrt{\frac{P_i}{\alpha_i^2 P + N}} \left(s\alpha_i\sqrt{P} + W_i \right) + W_{n+1},$$

where $W_{n+1} \sim \mathcal{N}(0, N)$ is the white noise at the destination.

3.2 Link Throughput

In this subsection, we quantify how the transmission power at the relay node affects the link throughput between the source and destination.

Based on [1], for a Gaussian source (i.e., the signal that the source sends out follows a Gaussian distribution), the rate distortion function for a squared-error distortion measure is $R(\mathcal{D}) = \frac{1}{2} \log \frac{P}{\mathcal{D}}$. That is, when the source node transmits at power P and the distortion at the destination node is $\mathcal{D}$, the link throughput between the source and destination is at least $\frac{1}{2} \log \frac{P}{\mathcal{D}}$. According to [4], the squared-error distortion between the original signal s and the scaled signal received at the destination node is:

$$\tilde{\mathcal{D}} = \frac{PN}{\frac{\left(\sum_{i=1}^n \alpha_i x_i\right)^2}{1 + \sum_{i=1}^n x_i^2} P + N}, \tag{1}$$

where $x_i = \beta_i \sqrt{\frac{P_i}{\alpha_i^2 P + N}}$. Note that it takes two time slots to transmit the signal under the amplify-and-forward strategy, hence the link throughput between the source and destination is at least

$$R(\tilde{\mathcal{D}}) = \frac{1}{4} \log \frac{P}{\tilde{\mathcal{D}}}, \tag{2}$$

where the distortion $\tilde{\mathcal{D}}$ is given in Eq. (1).

3.3 Problem Formulation

In the **Min-Energy** problem, we seek to minimize the total transmission power at the relays, such that the link throughput between S and D is greater than or equal to a threshold C. Based on Eq. (2), the constraint that the throughput is greater than or equal to C (i.e., $R(\tilde{\mathcal{D}}) \geq C$) is

$$\frac{\left(\sum_{i=1}^n \alpha_i x_i\right)^2}{1 + \sum_{i=1}^n x_i^2} \geq \left(2^{4C} - 1\right) \frac{N}{P}, \tag{3}$$

where $x_i = \beta_i \sqrt{\frac{P_i}{\alpha_i^2 P + N}}$. To simplify the notations, let $\mathcal{T} = \left(2^{4C} - 1\right) \times \frac{N}{P}$. Then, the problem of minimizing the total transmission power at the relay nodes while maintaining the required link throughput is given as follows:

$$\begin{aligned} \textbf{min} \quad & \sum_{i=1}^{n} P_i \\ \textbf{s.t.} \quad & -\left(\sum_{i=1}^{n} \alpha_i x_i\right)^2 + \mathcal{T}\left(1 + \sum_{i=1}^{n} x_i^2\right) \leq 0 \\ & 0 \leq P_i \leq Q, \text{ for } i = 1, \dots, n, \end{aligned} \tag{4}$$

where $x_i = \beta_i \sqrt{\frac{P_i}{\alpha_i^2 P + N}}$. Since $P_i = \frac{x_i^2}{\beta_i^2}(\alpha_i^2 P + N)$, we simplify the above optimization problem as following:

$$\begin{aligned} \textbf{min} \quad & \sum_{i=1}^{n} \frac{x_i^2}{\beta_i^2}(\alpha_i^2 P + N) \\ \textbf{s.t.} \quad & -\left(\sum_{i=1}^{n} \alpha_i x_i\right)^2 + \mathcal{T}\left(1 + \sum_{i=1}^{n} x_i^2\right) \leq 0 \\ & 0 \leq x_i \leq \beta_i \sqrt{\frac{Q}{\alpha_i^2 P + N}}, \text{ for } i = 1, \dots, n. \end{aligned} \tag{5}$$

For Problem (5), P, N, Q, α_i and β_i are known parameters, and x_i ($i = 1, \dots, n$) are unknown variables.

Although the Min-Energy problem provides a theoretical lower bound on the transmission power to maintain the link throughput, it has some strong assumptions. First, the solution involves all relays. This will generate significant synchronization overhead, since the relays have to be synchronized to have their signals arrive at the destination simultaneously. Second, each relay can transmit at any power level below the maximum power Q. However, in practice, there are only limited number of power levels, and it is hard to continuously adjust the power level to all possible values.

To address these limitations, we study a more practical problem called the **Min-Relay** problem. In the Min-Relay problem, the goal is to minimize the number of active relays, where each active relay transmits at power Q, such that the link throughput between S and D is greater than a threshold C. Let $I_i \in \{0, 1\}$ denote whether relay R_i has been chosen to forward the signal ($I_i = 1$ represents R_i is chosen, and $I_i = 0$ represents otherwise), then it is straightforward to formulate the Min-Relay problem as an integer programming problem. Similar to Eq. (3), the constraint that the throughput is greater than or equal to C is

$$\frac{\left(\sum_{i=1}^n I_i \theta_i\right)^2}{1 + \sum_{i=1}^n I_i \delta_i^2} \geq \mathcal{T}, \tag{6}$$

where $\delta_i = \beta_i \sqrt{\frac{Q}{\alpha_i^2 P+N}}$, $\theta_i = \alpha_i \delta_i$. Since minimizing the number of active relays is equivalent to minimizing the sum of I_i, the Min-Relay problem can be formulated as follows:

$$\begin{aligned} \textbf{min} \quad & \sum_{i=1}^{n} I_i \\ \textbf{s.t.} \quad & L = \left(\sum_{i=1}^{n} \theta_i I_i\right)^2 - \mathcal{T} \sum_{i=1}^{n} \delta_i^2 I_i \geq \mathcal{T}, \\ & I_i \in \{0,1\}, \text{ for } i = 1, \dots, n, \end{aligned} \tag{7}$$

where $\delta_i = \beta_i \sqrt{\frac{Q}{\alpha_i^2 P+N}}$, $\theta_i = \alpha_i \delta_i$, $\mathcal{T} = \left(2^{4C} - 1\right) \frac{N}{P}$, and $I_i \in \{0,1\}$ $(i = 1, \dots, n)$ is the unknown variable to denote whether relay R_i has been chosen to help forward the signal.

4 THE MIN-ENERGY PROBLEM

In this section, we study the Min-Energy problem, and propose an optimal solution which minimizes the transmission power of the relays while maintaining the required link throughput.

4.1 Important Properties

Before solving the Min-Energy problem, we first discuss some of its important properties.

Lemma 4.1. *When the maximum transmission power of the relay nodes Q goes to infinity, a feasible solution to Problem (5) exists if and only if:*

$$\sum_{i=1}^{n} \alpha_i^2 > \mathcal{T},$$

where $\mathcal{T} = \left(2^{4C} - 1\right) \frac{N}{P}$.

Proof. According to Cauchy Schwarz inequality [18], we have

$$\sum_{i=1}^{n} \alpha_i^2 \geq \frac{\left(\sum_{i=1}^{n} \alpha_i x_i\right)^2}{\sum_{i=1}^{n} x_i^2} > \frac{\left(\sum_{i=1}^{n} \alpha_i x_i\right)^2}{1 + \sum_{i=1}^{n} x_i^2}.$$

To satisfy the constraint (3) of Problem (5), it is required that $\sum_{i=1}^{n} \alpha_i^2 > \mathcal{T}$. On the other hand, when $\sum_{i=1}^{n} \alpha_i^2 > \mathcal{T}$, there always exists a feasible solution that satisfies the constraint (3) of Problem (5). □

Given $\mathcal{T}$, the above lemma indicates that even when each relay can employ infinite power to forward its received signal, to maintain a higher throughput, more relays or relays closer to the source node should be involved in the transmission.

For the more practical case of $Q \ll \infty$, the above condition is necessary but not sufficient for the existence of a feasible solution. Then, we have the following lemma.

Lemma 4.2. *For the optimization Problem (5), when a feasible solution exists (i.e., when $\sum_{i=1}^{n} \alpha_i^2 > \mathcal{T}$), the problem is non-convex.*

Proof. The Hessian matrix of constraint (3) is given as follows:

$$H = \begin{bmatrix} 2(\mathcal{T} - \alpha_1^2) & -2\alpha_1\alpha_2 & \dots & -2\alpha_1\alpha_n \\ -2\alpha_1\alpha_2 & 2(\mathcal{T} - \alpha_2^2) & \dots & -2\alpha_2\alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ -2\alpha_1\alpha_n & -2\alpha_2\alpha_n & \dots & 2(\mathcal{T} - \alpha_n^2) \end{bmatrix}.$$

Let $H = A + B$, where $A = 2\mathcal{T} * I_n$ (I_n is the identity matrix of size n), and $B = -2[\alpha_1, \alpha_2, \dots, \alpha_n]^T \times [\alpha_1, \alpha_2, \dots, \alpha_n]$. Matrix B is the product of two vectors, and its rank cannot exceed the rank of a vector. Then the rank of matrix B is 1, and 0 is an eigenvalue of B, and the algebraic multiplicity of eigenvalue 0 is $n-1$. With $H = 2\mathcal{T} * I_n + B$, the eigenvectors of B are also eigenvectors of H. For each of the eigenvectors, its corresponding eigenvalue in B is $2\mathcal{T}$ smaller than its eigenvalue in H. Because 0 is an eigenvalue of B and the eigenvalues of B are $2\mathcal{T}$ smaller than the eigenvalues of H, $2\mathcal{T}$ is an eigenvalue of H, and the algebraic multiplicity of eigenvalue $2\mathcal{T}$ is $n-1$. Matrix H is of size $n \times n$, and its eigenvalue $2\mathcal{T}$ has multiplicity of $n-1$, then H has only one more eigenvalue. Let λ denote the remaining eigenvalue, based on [18], the trace of H equals to the sum of all the eigenvalues of H, then λ can be obtained by:

$$\lambda = \sum_{i=1}^{n} 2(\mathcal{T} - \alpha_i^2) - 2\mathcal{T} * (n-1) = 2\mathcal{T} - 2\sum_{i=1}^{n} \alpha_i^2.$$

Based on lemma 4.1, a feasible solution exists only when $\sum_{i=1}^{n} \alpha_i^2 > \mathcal{T}$, thus eigenvalue λ must be smaller than 0. Accordingly, the Hessian matrix H of constraint (3) is not positive-semidefinite, and Problem (5) is non-convex. □

Based on the above lemma, the Min-Energy problem is a non-convex problem. Compared to convex problems, non-convex problems are generally much more difficult to solve. This is because for non-convex problems, a local optimal point is not guaranteed to be the global optimal point. Consequently, simple algorithms that stop at a local optimal point can not provide optimal solution to non-convex problems. To address such difficulties, we apply semidefinite relaxation (SDR) [13] technique to Problem (5), and prove that our solution can minimize the power consumption of the relays.

4.2 Semidefinite Relaxation

SDR is a computationally efficient technique to provide approximated solutions for difficult optimization problems. It has been widely used to solve non-convex quadratically constrained quadratic programs. Basically, SDR first relaxes the non-convex constraints of the original problem, and transforms the original difficult non-convex problem to a simpler convex problem. By solving the simpler convex problem, an approximated solution to the original problem can be obtained.

To apply SDR to our Min-Energy problem, we present the canonical form of Problem (5) as following:

$$\begin{aligned} \textbf{min} \quad & x^T U x \\ \textbf{s.t.} \quad & x^T V x \leq -\mathcal{T}, \\ & 0 \leq x \leq W, \end{aligned} \tag{8}$$

where $x = [x_1, \dots, x_n]^T \in \mathbb{R}^n$, $V = -[\alpha_1, \dots, \alpha_n]^T \times [\alpha_1, \dots, \alpha_n] + \mathcal{T} \times \mathcal{I}_n$, $W = \left[\sqrt{\frac{\beta_1^2 Q}{\alpha_1^2 P+N}}, \dots, \sqrt{\frac{\beta_n^2 Q}{\alpha_n^2 P+N}}\right]^T$, and U is a $n \times n$ diagonal matrix with diagonal elements $\left[\frac{\alpha_1^2 P+N}{\beta_1^2}, \dots, \frac{\alpha_n^2 P+N}{\beta_n^2}\right]$. To isolate the convex constraints from the non-convex constraints, we introduce a new variable $X = x \times x^T$. Since $x \in \mathbb{R}^n$, X must be a symmetric positive semidefinite matrix and rank$(X) = 1$. For the objective function and the constraint in Problem (8), we have

$$x^T U x = \text{Tr}\left(U x x^T\right) = \text{Tr}\left(UX\right),$$
$$x^T V x = \text{Tr}\left(V x x^T\right) = \text{Tr}\left(VX\right),$$

where Tr(·) is the trace of a matrix. Then, Problem (8) is equivalent to the following problem:

$$\begin{aligned} \min_{X \in \mathbb{S}^n} \quad & \text{Tr}\left(UX\right) \\ \textbf{s.t.} \quad & \text{Tr}\left(VX\right) \le -\mathcal{T}, \\ & \text{diag}\left(X\right) \le W \circ W, \\ & X \succeq 0, \\ & \text{rank}\left(X\right) = 1. \end{aligned} \tag{9}$$

Here $\mathbb{S}^n$ is the set of symmetric matrices of size $n \times n$, diag(X) is the vector formed by the diagonal elements of X, and $W \circ W$ is the Hadamard product (i.e., the pair-wise product that produces a vector of the same size as the original vector), and $X \succeq 0$ represents matrix X is positive semidefinite.

For the objective function and the first two constraints of Problem (9), they only involve linear functions, thus they are all convex. The third constraint (i.e. $X \succeq 0$) is also convex, since for any $0 < \lambda < 1, X_1 \succeq 0$ and $X_2 \succeq 0$, we have $\lambda X_1 + (1-\lambda)X_2 \succeq 0$. Hence, the difficulty of solving Problem (9) lies in the last non-convex constraint, which requires the rank of matrix X equals to 1. Based on SDR, we remove this non-convex constraint and have the following relaxed convex problem:

$$\begin{aligned} \min_{X \in \mathbb{S}^n} \quad & \text{Tr}\left(UX\right) \\ \textbf{s.t.} \quad & \text{Tr}\left(VX\right) \le -\mathcal{T}, \\ & \text{diag}\left(X\right) \le W \circ W, \\ & X \succeq 0. \end{aligned} \tag{10}$$

The above convex problem can be easily solved. In the next subsection, we convert a globally optimal solution $\hat{X}^*$ to Problem (10) into a feasible solution X^* to Problem (8). We then prove that the feasible solution X^* is in fact an optimal solution to Problem (8).

4.3 Optimal Solution

Let $n \times n$ symmetric matrix $\hat{X}^*$ be an optimal solution to Problem (10), where the element at the i-th row and j-th column of $\hat{X}^*$ is denoted by $\hat{x}^*_{ij}$. As it is required that $\hat{X}^* \succeq 0$, we have $\hat{x}^*_{ii} \ge 0$ for $1 \le i \le n$. Based on $\hat{X}^*$, we construct $X^* = \left[\sqrt{\hat{x}^*_{11}}, \sqrt{\hat{x}^*_{22}}, \dots, \sqrt{\hat{x}^*_{nn}}\right]^T \times \left[\sqrt{\hat{x}^*_{11}}, \sqrt{\hat{x}^*_{22}}, \dots, \sqrt{\hat{x}^*_{nn}}\right]$. Let x^*_{ij} denote the element in the i-th row and j-th column of X^*, then we have $x^*_{ij} = \sqrt{\hat{x}^*_{ii}}\sqrt{\hat{x}^*_{jj}}$. The following lemma compares the value of $\hat{x}^*_{ij}$ and x^*_{ij}.

Lemma 4.3. *$x^*_{ij} \ge \hat{x}^*_{ij}$, $\forall 1 \le i \le n$ and $1 \le j \le n$.*

Proof. When $i = j$, based on the definition of X^*, $x^*_{ij} = \hat{x}^*_{ij}$.

When $i \ne j$, we prove the lemma by contradiction. Assume $\hat{x}^*_{ij} > x^*_{ij}$, consider a $n \times 1$ vector z with all elements equal to 0, except the i-th element $z_i = \sqrt{\hat{x}^*_{jj}}$ and the j-th element $z_j = -\sqrt{\hat{x}^*_{ii}}$. Then we have

$$z^T \hat{X}^* z = 2\sqrt{\hat{x}^*_{ii}\hat{x}^*_{jj}}\left(\sqrt{\hat{x}^*_{ii}\hat{x}^*_{jj}} - \hat{x}^*_{ij}\right).$$

Recall that $x^*_{ij} = \sqrt{\hat{x}^*_{ii}}\sqrt{\hat{x}^*_{jj}}$ and it is assumed that $\hat{x}^*_{ij} > x^*_{ij}$. Then, the right-hand side of above equation is less than 0. For this specific z, we have $z^T \hat{X}^* z < 0$, which contradicts the condition that $\hat{X}^*$ is positive semidefinite. □

Based on the above lemma, we obtain an interesting property of the constructed matrix X^*, which is stated as follows.

Lemma 4.4. *X^* is a feasible, in fact optimal, solution to Problem (10).*

Proof. First, we show that X^* satisfies the first constraint of Problem (10). Recall that $V = -[\alpha_1, \dots, \alpha_n]^T \times [\alpha_1, \dots, \alpha_n] + \mathcal{T} \times I_n$, and let $Y = -[\alpha_1, \dots, \alpha_n]^T \times [\alpha_1, \dots, \alpha_n]$ and let $Z = \mathcal{T} \times I_n$. Note that the diagonal elements of $\hat{X}^*$ and X^* are the same, then we have $\text{Tr}(ZX^*) = \text{Tr}(Z\hat{X}^*)$. Because $\alpha_i \ge 0$ for all i, all the elements of Y are less than or equal to 0. Combining this property of Y with the results of lemma 4.3 (i.e., $x^*_{ij} >= \hat{x}^*_{ij}$), we have $\text{Tr}(YX^*) \le \text{Tr}(Y\hat{X}^*)$. Consequently, $\text{Tr}(VX^*) = \text{Tr}(ZX^*) + \text{Tr}(YX^*) \le \text{Tr}(V\hat{X}^*) = \text{Tr}(Z\hat{X}^*) + \text{Tr}(Y\hat{X}^*)$. Since $\hat{X}^*$ is a feasible solution to Problem (10), $\text{Tr}(VX^*) \le \text{Tr}(V\hat{X}^*) \le -\mathcal{T}$.

Second, we prove that X^* also satisfies the second and third constraints of Problem (10). Since the diagonal elements of X^* and $\hat{X}^*$ are the same, X^* must satisfy the second constraint. For the third constraint, based on the definition of X^*, for any $n \times 1$ vector z, we have

$$\begin{aligned} z^T X^* z = &\left(\left[\sqrt{\hat{x}^*_{11}}, \sqrt{\hat{x}^*_{22}}, \dots, \sqrt{\hat{x}^*_{nn}}\right] z\right)^T \\ &\times \left(\left[\sqrt{\hat{x}^*_{11}}, \sqrt{\hat{x}^*_{22}}, \dots, \sqrt{\hat{x}^*_{nn}}\right] z\right) \ge 0. \end{aligned}$$

Thus, X^* is positive semidefinite, and X^* satisfies the third constraint.

Lastly, we show that by replacing $\hat{X}^*$ with X^*, the value of the objective function is unchanged. Based on problem formulation, U is a diagonal matrix, the condition that the diagonal elements of X^* and $\hat{X}^*$ are the same is sufficient to guarantee that $\text{Tr}(UX^*) = \text{Tr}(U\hat{X}^*)$.

In conclusion, X^* is a feasible solution to Problem (10), and $\text{Tr}(UX^*)$ is equal to the optimal value of the problem. Thus, X^* is an optimal solution to Problem (10). □

Based on the above lemma, we now show that the constructed matrix X^* is actually optimal to Problem (9).

Theorem 4.5. *X^* is a feasible, in fact optimal, solution to Problem (9).*

Proof. Since $X^* = \left[\sqrt{\hat{x}^*_{11}}, \sqrt{\hat{x}^*_{22}}, \dots, \sqrt{\hat{x}^*_{nn}}\right]^T \times \left[\sqrt{\hat{x}^*_{11}}, \sqrt{\hat{x}^*_{22}}, \dots, \sqrt{\hat{x}^*_{nn}}\right]$, $\text{Rank}(X^*) = 1$. Combining this result with lemma 4.4, X^* is a feasible solution to Problem (9).

To show X^* is optimal to Problem (9), we assume there exists another feasible solution $\tilde{X}^* \neq X^*$ to Problem (9), such that $\mathrm{Tr}(U\tilde{X}^*) < \mathrm{Tr}(UX^*)$. Because Problem (10) is a relaxed problem of Problem (8), $\tilde{X}^*$ must also be feasible to Problem (10). Then, the optimal value of Problem (10), which is $\mathrm{Tr}(UX^*)$, can not exceed $\mathrm{Tr}(U\tilde{X}^*)$. This contradicts the condition that $\mathrm{Tr}(U\tilde{X}^*) < \mathrm{Tr}(UX^*)$, thus X^* must be optimal to Problem (9). □

Problem (9) and the original Problem (8) are equivalent, where the only difference is that the unknown variable is x in Problem (8) and the unknown variable is $X = xx^T$ in Problem (9). In theorem 4.5, we have shown that X^* is optimal to (9), and it is known that $X^* = \left[\sqrt{\hat{x}^*_{11}}, \sqrt{\hat{x}^*_{22}}, \dots, \sqrt{\hat{x}^*_{nn}}\right]^T \times \left[\sqrt{\hat{x}^*_{11}}, \sqrt{\hat{x}^*_{22}}, \dots, \sqrt{\hat{x}^*_{nn}}\right]$, and then $\left[\sqrt{\hat{x}^*_{11}}, \sqrt{\hat{x}^*_{22}}, \dots, \sqrt{\hat{x}^*_{nn}}\right]^T$ must be optimal to the original problem (8).

Theorem 4.6. *Given a solution accuracy $\epsilon > 0$, the Min-Energy problem can be solved within $O(n^{4.5}\log(1/\epsilon))$ time, where n is the number of relays in the network.*

Proof. Based on [13, 21], the time complexity of solving Problem (10) is $O(n^{4.5}\log(1/\epsilon))$. Obtaining the solution to Problem (8) based on the solution to Problem (10) takes $O(n)$ time. Therefore, the Min-Energy problem can be solved within $O(n^{4.5}\log(1/\epsilon))$ time. □

5 THE MIN-RELAY PROBLEM

In this section, we study the Min-Relay problem, where the goal is to minimize the number of active relays while maintaining the social link.

5.1 Relay Selection

The Min-Relay problem (7) is an integer programming problem which chooses the minimum number of relays, until the constraint $L \geq \mathcal{T}$ is satisfied. Let each relay R_i be represented by the parameter (θ_i, δ_i) as defined in Problem (7). Then the n relays in the Min-Relay problem can be viewed as n pairs $(\theta_1, \delta_1), \dots, (\theta_n, \delta_n)$. The problem of choosing the minimum number of relays becomes equivalent to choosing the minimum number of pairs, such that the constraint $L \geq \mathcal{T}$ is satisfied.

The basic idea of our solution is based on a related problem. Consider n packets each with weight W_i, the problem of finding the minimum number of packets, such that their total weight is greater than M can be formulated as follows:

$$\begin{aligned} \textbf{min} \quad & \sum_{i=1}^{n} I_i \\ \textbf{s.t.} \quad & \sum_{i=1}^{n} W_i I_i \geq M, \\ & I_i \in \{0,1\}, \text{ for } i = 1, \dots, n. \end{aligned} \tag{11}$$

For the above problem, the solution is to choose packets in the decreasing order of their weights, until their total weight is larger than M. To have a larger total weight, heavier packets are chosen before lighter packets. If the packets are ranked in the descending order of their weights, packets can be chosen one by one until the total weight grows larger than M. For Problem (7), if a similar ordering of the relays can be obtained, relays can be chosen one by one until constraint $L \geq \mathcal{T}$ is satisfied.

For the integer programming problem (7), it may have multiple optimal solutions; i.e., there may exist more than one $I = \{I_1, I_2, \dots, I_n\}$ that are optimal to Problem (7). We focus on one specific optimal solution I^*, which is defined as follows:

Definition 5.1. Let set S include all optimal solutions of Problem (7). $I^* = \{I^*_1, I^*_2, \dots, I^*_n\}$ is the optimal solution that maximizes L, that is

$$\begin{aligned} I^* &= \arg\max_{I \in S}(L) \\ &= \arg\max_{I \in S}\left(\left(\sum_{i=1}^{n}\theta_i I_i\right)^2 - \mathcal{T}\sum_{i=1}^{n}\delta_i^2 I_i\right). \end{aligned} \tag{12}$$

If any I^*_i in I^* is changed, it leads to either a sub-optimal solution or a decrease of L. Based on I^*, we formally define the selection order of the relays.

Definition 5.2. For any two relays R_i and R_j, R_i should be selected before R_j, denoted by $R_i \geq R_j$, if $I^*_i = 0$ and $I^*_j = 1$ cannot both hold.

In this definition, since $I^*_i = 0$ and $I^*_j = 1$ cannot both hold, it is impossible for R_j to be selected while R_i is not selected. Hence, R_i should always be selected before R_j. However, determining the selection order of the relays is not easy. When a relay R_i is added, the improvement of the link throughput not only depends on the parameter of R_i, but also depends on the relays that have already been selected. Hence, we can not obtain a simple selection order as in Problem (11), where the order only depends on the parameter of R_i. For the selection order of the relays, the order might be different when different relays have already been selected. Let $\mathcal{K}^* = \sum_{i=1}^{n}\theta_i I^*_i$, based on Definition 5.2, the following lemma states how to order two arbitrary relays.

Lemma 5.3. *Consider any two relays R_i and R_j with parameters (θ_i, δ_i) and (θ_j, δ_j) (assume $\theta_j > \theta_i$), and let $\mathcal{K}_{ij} = \frac{\mathcal{T}(\delta_i^2 - \delta_j^2)}{2(\theta_i - \theta_j)}$. If $\mathcal{K}^* > \mathcal{K}_{ij}$, then relay R_j is selected before R_i; otherwise, relay R_i is selected before R_j.*

Proof. We prove the lemma by contradiction. Suppose $I^*_i = 1$ and $I^*_j = 0$, and we replace R_i with R_j to transmit the data (i.e., let $I^*_i = 0$ and $I^*_j = 1$). Then, the value of the objective function remains unchanged, while L differs by:

$$\begin{aligned} \Delta L &= \left(\mathcal{K}^* + \theta_j - \theta_i\right)^2 - \mathcal{K}^{*2} - \mathcal{T}\left(\delta_j^2 - \delta_i^2\right) \\ &= 2\mathcal{K}^*\left(\theta_j - \theta_i\right) + \left(\theta_j - \theta_i\right)^2 - \mathcal{T}\left(\delta_j^2 - \delta_i^2\right). \end{aligned}$$

If $\Delta L > 0$, the new solution has a larger L than I^*, which contradicts the definition of I^*. Thus, when $\Delta L > 0$, $I^*_i = 1$ and $I^*_j = 0$ never hold, and consequently R_j is selected before R_i. Given $\theta_j > \theta_i$, the condition $\Delta L > 0$ is equivalent to

$$\mathcal{K}^* > \frac{\mathcal{T}(\delta_i^2 - \delta_j^2)}{2(\theta_i - \theta_j)} - \frac{1}{2}\left(\theta_j - \theta_i\right).$$

Then, when $\mathcal{K}^* > \mathcal{K}_{ij} = \frac{\mathcal{T}(\delta_i^2 - \delta_j^2)}{2(\theta_i - \theta_j)}$ (note that $\theta_j > \theta_i > 0$), $R_j \geq R_i$ (R_j is selected before R_i).

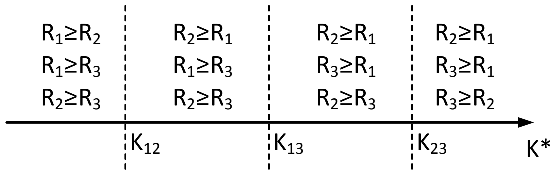


Figure 1: Selection order of the relays

Similarly, suppose $I_i^* = 0$ and $I_j^* = 1$, replacing R_j with R_i will increase L when

$$\mathcal{K}^* < \frac{\mathcal{T}(\delta_i^2 - \delta_j^2)}{2(\theta_i - \theta_j)} - \frac{1}{2}\left(\theta_i - \theta_j\right).$$

That is, when $\mathcal{K}^* < \mathcal{K}_{ij}$, we have $R_i \geq R_j$. □

The above lemma states that for any two relays R_i and R_j, based on $\mathcal{K}_{ij} = \frac{\mathcal{T}(\delta_i^2 - \delta_j^2)}{2(\alpha_i - \alpha_j)}$, their selection order can be determined as follows: when $\mathcal{K}^* > \mathcal{K}_{ij}$, the relay with larger θ value should be chosen first; otherwise, the relay with smaller θ value should be chosen first. If we calculate $\mathcal{K}_{ij}$ for any two of the n relays, there will be $\frac{n(n-1)}{2}$ different $\mathcal{K}_{ij}$. Those different $\mathcal{K}_{ij}$ will divide the feasible region of $\mathcal{K}^*$ into $\frac{n(n-1)}{2} + 1$ small regions, where in each region we know the selection order between any two relays.

Fig. 1 gives an example for the case of three relays R_1, R_2 and R_3 ($\theta_1 < \theta_2 < \theta_3$). When $\mathcal{K}^*$ resides in the leftmost region, $\mathcal{K}^* < \mathcal{K}_{12}$ leads to $R_1 \geq R_2$, $\mathcal{K}^* < \mathcal{K}_{13}$ leads to $R_1 \geq R_3$ and $\mathcal{K}^* < \mathcal{K}_{23}$ leads to $R_2 \geq R_3$. The selection order of the relays in other three regions are also shown in the figure, which is obtained according to the value of $\mathcal{K}^*$, $\mathcal{K}_{12}$, $\mathcal{K}_{13}$ and $\mathcal{K}_{23}$.

However, given the selection order between any two relays, a complete ordering of all relays may not exist. For example, if the ordering is like $R_1 \geq R_2$, $R_2 \geq R_3$, and $R_3 \geq R_1$, then $R_1 \geq R_2$ and $R_2 \geq R_3$ lead to $R_1 \geq R_3$ which contradicts $R_3 \geq R_1$. To show that a complete ordering of the relays always exists, we prove the ordering given in lemma 5.3 is transitive.

LEMMA 5.4. *For a given $\mathcal{K}^*$, $R_i \geq R_j$ and $R_j \geq R_k$ will lead to $R_i \geq R_k$.*

PROOF. Based on the value of θ_i, θ_j and θ_k, there are 6 different cases. Consider the first case of $\theta_i > \theta_j > \theta_k$. $R_i \geq R_j$ and $\theta_i > \theta_j$ lead to $\mathcal{K}^* \geq \mathcal{K}_{ij}$, which is

$$\mathcal{K}^* \geq \frac{\mathcal{T}(\delta_i^2 - \delta_j^2)}{2(\theta_i - \theta_j)}. \tag{13}$$

Similarly, with $R_j \geq R_k$ and $\theta_j > \theta_k$, we have

$$\mathcal{K}^* \geq \frac{\mathcal{T}(\delta_j^2 - \delta_k^2)}{2(\theta_j - \theta_k)}. \tag{14}$$

For $\mathcal{K}_{ik}$, we have

$$\begin{aligned}\mathcal{K}_{ik} &= \frac{\mathcal{T}(\delta_i^2 - \delta_k^2)}{2(\theta_i - \theta_k)} = \frac{\mathcal{T}(\delta_i^2 - \delta_j^2)}{2(\theta_i - \theta_k)} + \frac{\mathcal{T}(\delta_j^2 - \delta_k^2)}{2(\theta_i - \theta_k)} \\ &= \frac{\mathcal{T}(\delta_i^2 - \delta_j^2)}{2(\theta_i - \theta_j)}\frac{(\theta_i - \theta_j)}{(\theta_i - \theta_k)} + \frac{\mathcal{T}(\delta_j^2 - \delta_k^2)}{2(\theta_j - \theta_k)}\frac{(\theta_j - \theta_k)}{(\theta_i - \theta_k)}\end{aligned}$$

Combining the above equation with Eq. (13) Eq. (14), we can get

$$\mathcal{K}_{ik} \leq \mathcal{K}^* \frac{(\theta_i - \theta_j)}{(\theta_i - \theta_k)} + \mathcal{K}^* \frac{(\theta_j - \theta_k)}{(\theta_i - \theta_k)} = \mathcal{K}^*.$$

Thus $R_i \geq R_k$. The other five cases can be similarly proved by comparing the value of $\mathcal{K}_{ik}$ and $\mathcal{K}^*$ based on $\mathcal{K}_{ij}$ and $\mathcal{K}_{jk}$. □

For each of the small region bounded by two consecutive $\mathcal{K}_{ij}$ or infinity, we can get a complete ordering of the relays. For example, in the leftmost region of Fig. 1, the ordering will be $R_1 \geq R_2 \geq R_3$. That is, R_1 should be chosen first, and R_2 should be chosen second, and R_3 should be chosen third.

To sum up, we can find the selection order of the relays based on the value of $\mathcal{K}^*$. If the value of $\mathcal{K}^*$ is known, relays can be selected one by one according to the selection order, until the constraint is met. Unfortunately, the value of $\mathcal{K}^*$ is not known a priori, and we have to try each of the $\frac{n(n-1)}{2} + 1$ selection order. By comparing all the different solutions, we can obtain an optimal solution to the Min-Relay problem. Next, we present the details of this solution, which is referred to as the *Min-Relay algorithm*.

5.2 The Min-Relay Algorithm

Algorithm 1 shows the formal description of the algorithm. Recall that δ_i and θ_i are calculated based on the signal scaling factor α_i and β_i, where $\delta_i = \beta_i\sqrt{\frac{Q}{\alpha_i^2 P + N}}$, $\theta_i = \alpha_i \delta_i$, and we have $\delta_i > 0$ and $\theta_i > 0$. Parameter $\mathcal{T}$ is related to the throughput between the source and destination, which is given by $\mathcal{T} = \left(2^{4C} - 1\right)\frac{N}{P}$.

Algorithm 1 Min Relay Algorithm

$S_{opt} \leftarrow \emptyset$, $Q_{Relay} \leftarrow$ initQueue()
$Q_{\mathcal{K}} \leftarrow$ initPriorityQueue()
for any R_i and R_j **and** $i \leq j$ **do**
  $\mathcal{K}_{ij} \leftarrow \mathcal{T}(\delta_i^2 - \delta_j^2)/2(\theta_i - \theta_j)$
  $Q_{\mathcal{K}}$.push($\{R_i, R_j, \mathcal{K}_{ij}\}$) {tuple with smaller $\mathcal{K}_{ij}$ has higher priority}
end for
$Q_{\mathcal{K}}$.push({NULL, NULL, inf})
for each R_i, in the increasing order of θ_i **do**
  Q_{Relay}.push($\{R_i, \theta_i, \delta_i\}$)
end for
while $Q_{\mathcal{K}}$.notEmpty() **do**
  $\{R_i, R_j, \mathcal{K}_{upper}\} \leftarrow Q_{\mathcal{K}}$.poll(), $Q_{ranking} \leftarrow Q_{Relay}$
  $\mathcal{K} \leftarrow 0$, $B \leftarrow 0$, $S_{temp} \leftarrow \emptyset$
  while $Q_{ranking}$.notEmpty() **and** $\mathcal{K} \leq \mathcal{K}_{upper}$ **and** $\mathcal{K}^2 - \mathcal{T}B < \mathcal{T}$ **do**
    $\{R_k, \theta_k, \delta_k\} \leftarrow Q_{ranking}$.poll()
    $S_{temp} \leftarrow S_{temp} \bigcup \{R_k\}$, $\mathcal{K} \leftarrow \mathcal{K} + \theta_k$, $B \leftarrow B + \delta_k^2$
  end while
  if $\mathcal{K}^2 - \mathcal{T}B \geq \mathcal{T}$ **and** ($|S_{temp}| < |S_{opt}|$ **or** $S_{opt} = \emptyset$) **then**
    $S_{opt} \leftarrow S_{temp}$
  end if
  Call *update-ordering* (Q_{relay}, R_i, R_j)
end while

In the algorithm, S_{opt} records the relays chosen in the optimal solution, and $Q_{\mathcal{K}}$ stores all the small regions divided by $\mathcal{K}_{ij}$, and Q_{Relay} stores the current ordering of relays. The algorithm first calculates $\mathcal{K}_{ij}$ for each pair of relays, and push the tuple $(i, j, \mathcal{K}_{ij})$

Procedure 1 Update-Ordering (Q_{relay}, R_i, R_j)

```
Q_temp ← initQueue()
while Q_relay.notEmpty() do
    {R_k, θ_k, δ_k} ← Q_relay.poll()
    if R_k = R_i then
        Q_temp.push({R_j, θ_j, δ_j})
    else if R_k = R_j then
        Q_temp.push({R_i, θ_i, δ_i})
    else
        Q_temp.push({R_k, θ_k, δ_k})
    end if
end while
Q_relay ← Q_temp
```

into the priority queue $Q_{\mathcal{K}}$, where the tuple with smaller $\mathcal{K}_{ij}$ has higher priority. After that, one more tuple is pushed into the queue (in Line 7). It corresponds to the rightmost region. Line 8-10 gives the ordering of the relays for the first region (i.e., the leftmost region), where simply relays with lower θ_i are first chosen. The while loop between Line 11-22 finds the solution when $\mathcal{K}^*$ resides in each of the small region. Basically, in each small region, the algorithm chooses relays one by one until either the constraint $L \geq \mathcal{T}$ is met, or for the relays already chosen, $\mathcal{K}$ grows out of the bounds of the region (i.e., $\mathcal{K} = \sum_{i=1}^{n} \theta_i I_i > \mathcal{K}_{upper}$). If a solution is obtained, in Line 18-20, the algorithm compares the solution with the current best solution, and updates the current best solution if needed ($|\cdot|$ in Line 18 is the cardinality of a set). In Line 21, the algorithm calls the *Update-Ordering Procedure* to update the ordering of the relays. In this way, as the while loop between Line 11 and 22 starts over (i.e., compute the solution for the next region), Q_{relay} will store the correct ordering for the next region. The details of the update-ordering procedure are given in Procedure 1.

Note that in Procedure 1, what we have achieved is simply exchange R_i and R_j in the ordering. Because in the next region, as $\mathcal{K}^*$ crosses $\mathcal{K}_{ij}$, only the ordering between R_i and R_j is reversed, and the ordering of all others are kept unchanged. For instance, in fig. 1, when comparing the leftmost region and the second left region, the only difference is the ordering of R_1 and R_2 is reversed. Hence, procedure 1 gives the correct ordering for the next region.

THEOREM 5.5. *The time complexity of the Min Relay Algorithm is $O(n^3)$, where n is the number of relays in the network.*

PROOF. For the while loop between Line 11 and Line 22, it repeats at most $O(n^2)$ times, since there are in total $O(n^2)$ small regions. The Update Ordering Procedure takes $O(n)$ time, and the while loop between Line 14 and Line 17 repeats at most $O(n)$ times. Hence, it takes $O(n^3)$ to run the algorithm. □

6 PERFORMANCE EVALUATIONS

In this section, we evaluate the performance of the proposed Min-Energy solution and the Min-Relay algorithm in terms of power consumption and the number of active relays.

6.1 Simulation setup

Our evaluation is based on the UCSD trace [15], which is collected at a campus scale with WiFi enabled PDAs. These devices search for nearby WiFi Access Points (APs), and a contact is detected when two devices detect the same AP. The trace records the contacts between 275 users every 20 seconds for a period of 77 days. Each user can detect multiple APs, and the user location is inferred based on the location of the detected APs and the AP signal strength [6].

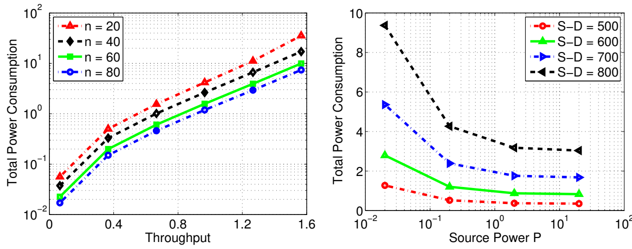
Because the social relationship between the users is not reported in the trace, to identify the important social links, we construct a weighted directed social graph upon the nodes in the network similar to [11]. Let f denote the total contact frequency of the whole trace, f_i denote the total contact frequency of node i, and f_{ij} denote the contact frequency between node i and j. Since users connected by stronger ties are more likely to contact frequently with each other, and users with more social ties are more likely to meet with others [5], the social graph is built as follows. First, we generate the node degrees following the power-law distribution, where the power-law coefficient is set to 1.6. Second, we assign the node degrees to the nodes in the network. For the largest node degree, it is assigned to node i with a probability of $\frac{f_i}{f}$. Then, this process is repeated for the remaining degrees and nodes. Third, for the social ties of each node, we generate weight for each of the social tie, where the weight is uniformly distributed in $[0, 1]$. Finally, we connect the social ties of each node to other nodes. Specifically, for the strongest tie of node i, it is connected to another node in a way that node j's probability to be connected is $\frac{f_{ij}}{f_i}$, and this is repeated for the remaining social ties and the nodes have not been connected to i.

In the simulation, node locations are generated based on a snapshot of the UCSD trace; i.e., we use the location of n nodes in that trace at a specific time to generate the node location. The important social link corresponds to link with the strongest social tie in the network. In the simulations, we focus on cases where multi-hop transmission can not be maintained due to network partition, and amplify-and-forward is used to help transmit the data. The link source is the node in the routing path that can not maintain the required link throughput between itself and its next hop node (called link destination), even with the highest transmission power. To simplify the notations, we use S-D to represent the distance between the link source and the link destination.

In the simulation, the link source uses power $P = 2$W to transmit the data to all relay nodes. For Min-Energy, each relay can employ any power level below the maximum power $Q = 2$W for data transmission. For Min-Relay, all relays use power $Q = 2$W for data transmissions. The noise at each node follows a normal distribution $\sim \mathcal{N}(0, N)$, where $N = 10^{-10}$W.

6.2 Power Consumption

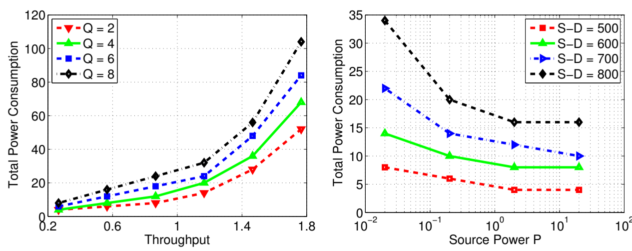
Fig. 2 evaluates how various parameters such as the link throughput, the number of relays, and the distance between the source and destination affect the power consumption of Min-Energy. Fig. 2(a) shows the effects of link throughput C and the number of relays n. As expected, a higher link throughput requires higher transmission power at the relays. We can also see that the power consumption



(a) Total power consumption vs. throughput (S-D = 400)

(b) Total power consumption vs. source transmission power (n = 20, $C = 0.25$)

Figure 2: The power consumption of Min-Energy



(a) Total power consumption vs. throughput ($n = 30$, S-D = 400)

(b) Total power consumption vs. source transmission power (n = 20, $C = 0.44$)

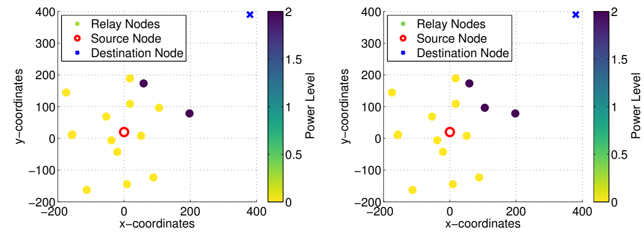
Figure 3: The power consumption of Min-Relay

decreases when the number of relays increases. This is because with more relays, the SNR at the destination can be improved, and consequently the relays can use lower power to achieve the required link throughput.

Fig. 2(b) shows the power consumption as a function of P and the distance between the source and destination. As shown in the figure, the transmission power at the relays increases when the distance between the source and destination increases. This is because the signal strength at the destination drops quickly with increasing distance. To maintain the link throughput, the relays have to employ higher power for data transmission.

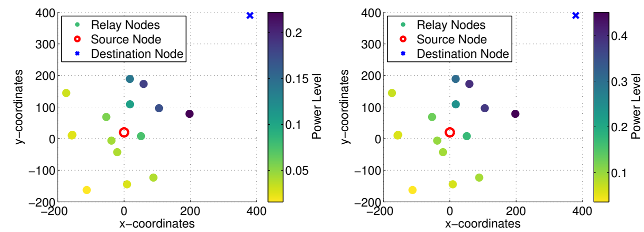
We can also see that the power consumption decreases as P grows. This is because increasing P increases the signal strength at the relays, and consequently improves the SNR at the destination. When P is large, further increasing P hardly affects the transmission power. This is because for large P, the noise at the relay is negligible compared to P, and then the SNR at the destination is only determined by the relay transmission power and the white noise at the destination.

Fig. 3 illustrates how the power consumption of Min-Relay is affected by the parameters including the link throughput, the relay transmission power, and the distance between the source and destination. Fig. 3(a) shows how the transmission power at the relays varies with link throughput, given different relay transmission power Q. As can be seen, when Q increases, the total transmission power increases and the number of active relays decreases. For example, when $C = 1.5$ and $Q = 2$W, the total transmission power is 28W, and there are $28/2 = 14$ active relays; when $C = 1.5$ and $Q = 8$W, the total transmission power is 56W, and 7 relays are involved in the transmission. The number of active relays decreases with increasing Q, because with higher Q, each node can provide a



(a) Min-Relay ($C = 0.46$)

(b) Min-Relay ($C = 0.68$)



(c) Min-Energy ($C = 0.46$)

(d) Min-Energy ($C = 0.68$)

Figure 4: Relay Selection

stronger signal strength at the destination, and consequently less number of relays are required to maintain the link.

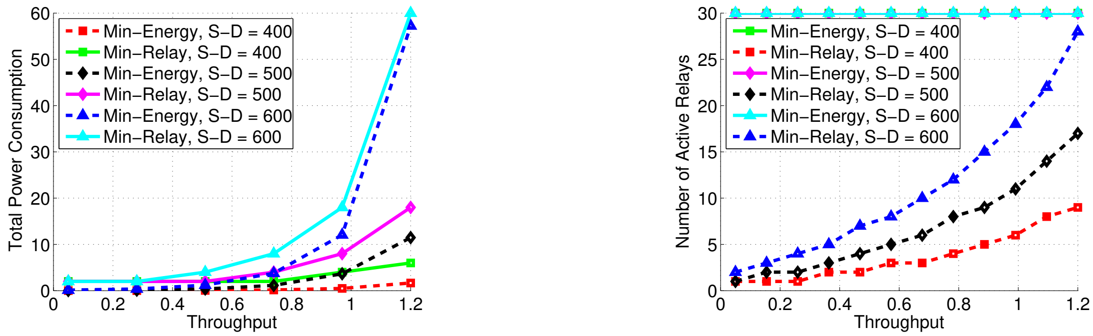
Fig. 3(b) shows the power consumption as a function of P, when the distance between the source and destination varies. As can be seen, increasing P reduces the transmission power at the relays, and the number of active relays. When the distance between the source and destination increases, the transmission power increases and more relays are involved.

6.3 Comparing Min-Energy and Min-Relay

Fig. 4 illustrates what relays will be selected when the link throughput C increases from 0.46 to 0.68 in Min-Energy and Min-Relay. In the figure, the red circle near $(0, 0)$ denotes the source, and the blue cross near $(400, 400)$ denotes the destination. There are 15 relays in the network (i.e., $n = 15$), which are denoted by the solid circles. The color of the solid circle represents the transmission power of the relay, where dark color means high power and light color means low power.

For Min-Relay, as shown in Fig. 4(a), when $C = 0.46$, only two relays are active, each with a transmission power of 2W. In Fig. 4(b), C increases to 0.68. As a result, one more relay is selected to support higher link throughput. For Min-Energy, as shown in Fig. 4(c), when $C = 0.46$, all relays are involved, although most of them only use a relatively low power (about 0.25W) for transmission. When C increases to 0.68 (see Fig. 4(d)), relays employ higher power for transmission, e.g., most of them transmit with a power of 0.5W. From the figure, we can easily see that the number of active relays in Min-Relay is much less than that in Min-Energy.

In Fig. 5(a), we compare the power consumption of Min-Relay and Min-Energy as the link throughput C increases, given different distance between the source and destination. The results of Min-Energy are shown by the dashed lines, while the results of Min-Relay are shown by the solid lines. As can be seen, the power consumption of Min-Energy and Min-Relay is comparable. Min-Energy always performs better than Min-Relay, because it has less constraints and has the minimum power consumption for maintaining the required link throughput. We can also see that the power



(a) Total power consumption **(b) Number of active relays**

Figure 5: Min-Relay vs. Min-Energy (n = 30)

consumption of Min-Relay converges to Min-Energy as C and S-D both increase (e.g., the teal and blue line at $C = 1.25$). This is because when C and S-D are large, all relays are required to transmit at their highest power to maintain the required link throughput.

Fig. 5(b) compares the number of active relays in Min-Energy and Min-Relay. The results for Min-Relay are shown by the dashed lines, and the results for Min-Energy are shown by the solid lines. For Min-Energy, all 30 relays are involved in the transmission, and the three lines overlap. For Min-Relay, when the throughput increases or the distance between the source and destination increases, more relays are involved. In general, Min-Relay has much less number of active relays.

7 CONCLUSION

In this paper, we applied the amplify-and-forward strategy to maintain social links in wireless networks. To save energy at the relays, we formulated and studied the Min-Energy problem. We formulated it as a non-convex quadratic programming problem by exploiting the rate-distortion theory, and solved it based on an approximation technique. We also proved that our solution can minimize the power consumption of the relays. The Min-Energy problem involves all relays, which could generate significant synchronization overhead. To minimize the synchronization overhead, we also studied the Min-Relay problem which aims to minimize the number of active relays while maintaining the social link. We formulated the Min-Relay problem as an integer programming problem, and proposed a polynomial-time algorithm which can select the minimum number of relays to maintain the social link. Evaluation results showed that Min-Relay can significantly reduce the number of active relays compared to Min-Energy, while achieving comparable power consumption.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation (NSF) under grant CNS-1421578, and by Network Science CTA under grant W911NF-09-2-0053.

REFERENCES

[1] T. M. Cover and J. A. Thomas. 2012. *Elements of information theory*. John Wiley & Sons.

[2] Z. Ding, W. H. Chin, and K. K. Leung. 2008. Distributed beamforming and power allocation for cooperative networks. *IEEE Transactions on Wireless Communications* 7, 5 (2008), 1817–1822.

[3] W. Gao, Q. Li, B. Zhao, and G. Cao. 2009. Multicasting in delay tolerant networks: a social network perspective. In *ACM MobiHoc*.

[4] M. Gastpar and M. Vetterli. 2002. On the capacity of wireless networks: The relay case. In *IEEE INFOCOM*.

[5] M. S. Granovetter. 1973. The strength of weak ties. *American journal of sociology* 78, 6 (1973), 1360–1380.

[6] W. G. Griswold, P. Shanahan, S. W. Brown, R. Boyer, M. Ratto, R. B. Shapiro, and T. M. Truong. 2004. ActiveCampus: experiments in community-oriented ubiquitous computing. *Computer* 37, 10 (2004), 73–81.

[7] I. Hammerstrom, M. Kuhn, and A. Wittneben. 2004. Impact of relay gain allocation on the performance of cooperative diversity networks. In *Vehicular Technology Conference*.

[8] P. Hui, J. Crowcroft, and E. Yoneki. 2011. Bubble rap: Social-based forwarding in delay-tolerant networks. *IEEE Transactions on Mobile Computing* 10, 11 (2011), 1576–1589.

[9] Y. Jing and H. Jafarkhani. 2009. Single and multiple relay selection schemes and their achievable diversity orders. *IEEE Transactions on Wireless Communications* 8, 3 (2009), 1414–1423.

[10] Y. Ko and N. H. Vaidya. 2000. Location-Aided Routing (LAR) in mobile ad hoc networks. *Wireless networks* 6, 4 (2000), 307–321.

[11] Q. Li, S. Zhu, and G. Cao. 2010. Routing in socially selfish delay tolerant networks. In *IEEE INFOCOM*.

[12] Z. Lu, Y. Wen, W. Zhang, Q. Zheng, and G. Cao. 2016. Towards information diffusion in mobile social networks. *IEEE Transactions on Mobile Computing* 15, 5 (2016), 1292–1304.

[13] Z. Luo, W. Ma, A. M. So, Y. Ye, and S. Zhang. 2010. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine* 27, 3 (2010), 20.

[14] R. Madan, N. B. Mehta, A. F. Molisch, and J. Zhang. 2008. Energy-efficient cooperative relaying over fading channels with simple relay selection. *IEEE Transactions on Wireless Communications* 7, 8 (2008), 3013–3025.

[15] M. McNett and G. M. Voelker. 2005. Access and mobility of wireless PDA users. *ACM SIGMOBILE Mobile Computing and Communications Review* 9, 2 (2005), 40–55.

[16] M. Musolesi, S. Hailes, and C. Mascolo. 2005. Adaptive routing for intermittently connected mobile ad hoc networks. In *IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*.

[17] H. Ochiai, P. Mitran, H. V. Poor, and V. Tarokh. 2005. Collaborative beamforming for distributed wireless ad hoc sensor networks. *IEEE Transactions on Signal Processing* 53, 11 (2005), 4110–4124.

[18] G. Strang. 2011. *Introduction to linear algebra*. Wellesley-Cambridge Press.

[19] Z. Wei, X. Zhu, S. Sun, Y. Huang, L. Dong, and Y. Jiang. 2015. Full-duplex versus half-duplex amplify-and-forward relaying: which is more energy efficient in 60-ghz dual-hop indoor wireless systems? *IEEE Journal on Selected Areas in Communications* 33, 12 (2015), 2936–2947.

[20] S. Yang and J. Belfiore. 2007. Towards the optimal amplify-and-forward cooperative diversity scheme. *IEEE transactions on information theory* 53, 9 (2007), 3114–3126.

[21] Y. Ye. 2011. *Interior point algorithms: theory and analysis*. Vol. 44. John Wiley & Sons.

[22] Q. Yuan, I. Cardei, and J. Wu. 2009. Predict and relay: an efficient routing in disruption-tolerant networks. In *ACM MobiHoc*.

[23] X. Zhang and G. Cao. 2014. Efficient data forwarding in mobile social networks with diverse connectivity characteristics. In *IEEE ICDCS*.

[24] Y. Zhao, R. Adve, and T. J. Lim. 2006. Improving amplify-and-forward relay networks: optimal power allocation versus selection. In *IEEE International Symposium on Information Theory*.