

# Energy-balanced cooperative routing in multihop wireless networks

Siyuan Chen · Yang Li · Minsu Huang ·  
Ying Zhu · Yu Wang

Published online: 2 November 2012  
© Springer Science+Business Media New York 2012

**Abstract** Cooperative communication (CC) allows multiple nodes to simultaneously transmit the same packet to the receiver so that the combined signal at the receiver can be correctly decoded. Since the CC can reduce the transmission power and extend the transmission coverage, it has been considered in minimum energy routing protocols to reduce the total energy consumption. However, previous research on cooperative routing only focuses on minimizing the total energy consumption from the source node to the destination node, which may lead to the unbalanced energy distribution among nodes. In this paper, we aim to study the impact of cooperative routing on balancing the energy distribution among nodes. By introducing a new routing scheme which carefully selects cooperative relay nodes and assigns their transmission power, our cooperative routing method can balance the remaining energy among neighboring nodes to maximize the lifetime of the network. Simulation results demonstrate that the proposed cooperative routing algorithm significantly balances the

energy distribution and prolongs the lifetime of the network.

**Keywords** Energy balancing · Cooperative routing · Multihop networks

## 1 Introduction

Multihop wireless networks have various civilian or military applications and have drawn considerable attention in recent years. One of the major concerns in designing multihop wireless networks is to save the energy consumption as the wireless nodes are often powered by batteries only. Energy-aware routing protocols [1–10] have been well-studied. Most of these energy-aware routing protocols consider new energy-related metrics, such as a function of the energy required to communicate on a link [1, 2, 5] or the nodes' remaining lifetime [3] or both [11], instead of classic route metrics such as hop count or delay.

Recently, a new class of communication techniques, cooperative communications (CC) [12, 13], have been introduced to allow single antenna devices to take advantage of the multiple-input–multiple-output (MIMO) systems. This CC explores the broadcast nature of the wireless medium and allows nodes that have received the transmitted signal to cooperatively help by relaying data for other nodes. Previous study has shown significant performance gain of CC in various wireless network applications: broadcasting [14–16], topology control and connectivity [17–20], and throughput maximization [21].

Recent works [22–27] have investigated the impacts of CCs on the problem of minimum power routing. For example, Khandani et al. [23] and Li et al. [22] both study the methods of finding the minimum energy cooperative

---

S. Chen · M. Huang · Y. Zhu · Y. Wang (✉)  
Department of Computer Science, University of North Carolina  
at Charlotte, 9201 University City Blvd., Charlotte, NC 28223,  
USA  
e-mail: yu.wang@uncc.edu

S. Chen  
e-mail: schen40@uncc.edu

M. Huang  
e-mail: mhuang4@uncc.edu

Y. Zhu  
e-mail: yzhu17@uncc.edu

Y. Li  
School of Computer Science, Beijing Institute of Technology,  
5 South Zhongguancun Street, Beijing 100081, China  
e-mail: liyangicecream@gmail.com

route for a wireless network. They have developed several heuristic algorithms to decide the next hop node and the nodes which will participate the CC to the next hop. All these methods only focus on minimizing the total energy consumption of routing the packet from the source node to the destination node. However, it is well known that consistently using the minimum cost path for routing may lead to uneven energy distribution among nodes, which could substantially reduce the lifetime of the network.

Therefore, in this paper, we focus on studying the impact of CC on energy balancing among nodes. We introduce a new cooperative scheme to carefully select cooperative relay nodes from one-hop neighbors around the current node and make smart decisions on their transmission power. It can be applied to any underlying energy-aware routing protocol, and only need local information to perform the optimization on CCs. We formally prove that our cooperative routing method can indeed balance the energy consumption among nodes to prolong lifetime of the network. Notice that Pandana et al. [11] also study how to maximize network lifetime by using cooperative routing. However, they limit the scope of cooperative relay within the nodes on the route and concentrate on minimizing the total energy consumption of all cooperative nodes. This significantly limits the effectiveness of their method on energy balancing. In contrast, we allow all one-hop neighbors around current node to participate the energy-balanced cooperative relay and aim to balance their remaining energy. In addition, under our CC model, one of their methods regresses to the minimum energy path based routing.

The rest of this paper is organized as follows. In Sect. 2, we summarize related works in energy-efficient routing and cooperative routing for wireless networks. In Sect. 3, we introduce the network model and CC model used by our routing protocol. In Sect. 4, we present our energy-balanced cooperative routing (EBCR) protocol and theoretically prove that the protocol guarantees the improvement of balancing remaining energy among nodes in the network. Section 5 presents the simulation results on random networks by comparing the EBCR routing with the minimum energy path routing. Finally, Sect. 6 concludes the paper with a summary and an outlook for future work.

## 2 Related work

Since energy is a scarce resource which limits the lifetime of wireless networks, a number of energy efficient routing protocols [1–10] have been proposed recently using a variety of techniques. Classical routing algorithms may be adapted to energy-related criteria rather than metrics such as delay or hop distance. Most of the proposed energy-aware metrics are defined as a function of the energy

required to communicate on a link [1, 2, 5] or a function of the nodes remaining lifetime [3]. By taking these energy-aware metrics, many centralized minimum energy routing protocols [3–5] and energy efficient localized routing protocols [6–10] have been proposed and widely used in wireless ad hoc networks or sensor networks.

Cooperative communication models introduce a new element in designing energy efficient routing algorithms for wireless ad hoc networks. Recent work [22–27] has investigated the impacts of CCs on the problem of minimizing power routing. Khandani et al. [23] first formulate the problem of finding the minimum energy cooperative route for a wireless network and develop a dynamic-programming-based algorithm as well as two polynomial-time heuristic algorithms. Li et al. [22] also study the finding minimum energy cooperative route problem, but they assume that the last  $L$  predecessor nodes along the path are allowed for cooperative transmission to the next hop. They propose an algorithm which uses the Dijkstra's algorithm with a modified relaxation procedure to reflect the CC cost. In [24], a cooperation-based routing algorithm minimum power cooperative routing (MPCR) algorithm is proposed. This algorithm constructs the minimum-power route using any number of the proposed cooperation-based building blocks that require the least possible transmission power. In [26, 27], the cooperative multi-hop routing under more complex fading model is studied for the purpose of energy savings.

All of the research above focuses on minimizing the total energy consumption in cooperative routing. However, in this paper, we concentrate on balancing the energy among nodes and prolong the lifetime of the network. Pandana et al. [11] in study how to maximize network lifetime of network by using cooperative routing. They define the link cost by normalizing the transmission power with the residual energy at each node. Their methods first find the least cost route based on their link costs, then use the last  $L$  nodes in the route to perform cooperative relay for current node. Even though their methods indeed can balance energy consumption on certain level, they limit the scope of cooperative relay within the nodes on the route, which restricts the effectiveness on balancing energy. In contrast, in this paper, we consider one-hop neighbors around the route and use all possible neighbors to perform cooperative relay aiming at energy balancing. In addition, their method aims to minimize the total normalized energy consumption of all participating nodes, while our study want to balance the remaining lifetime among all participating nodes in CCs. Notice that under our CC model, which is different with their model, simply minimizing the total energy consumption of all nodes on the minimum energy path regresses to the minimum energy path itself without CCs.

In addition, cross-layer cooperative routing problems have been studied recently. Zhang et al. [25] study the joint problem of routing selection in network layer and contention avoidance among multiple links in MAC layer in a CC aware network. They convert the decision problem into an optimization problem which can be solved by linear programming. They also provide a distributed cooperative scheme to approximate the optimal solution. Recently, Sharma et al. [28] study the joint optimization problem of relay node assignment and flow routing for concurrent sessions under CC to maximize the throughput. They model the problem as a mixed integer linear programming problem and provide a novel solution based on the branch-and-cut framework.

### 3 Network models and assumptions

In this section, we briefly introduce the network models and assumptions in our proposed cooperative routing protocol.

We consider a connected multihop wireless network with  $n$  nodes  $V = \{v_1, v_2, \dots, v_n\}$ . Every node  $v_i$  can adjust its transmission power  $P(v_i)$  which is limited by a maximum value  $P_{MAX}$ . If a sending node  $v_i$  wants to communicate with node  $v_j$  directly, the transmission power of node  $v_i$  must satisfy

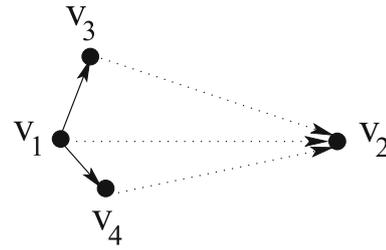
$$P(v_i) \cdot (d(v_i, v_j))^{-\alpha} \geq \tau \quad (P(v_i) \leq P_{MAX}).$$

Here,  $\alpha$  is the path loss exponent (between 2 and 4),  $\tau$  is the minimum average signal-to-noise ratio (SNR) for decoding received data successfully, and  $d(v_i, v_j)$  is the distance between node  $v_i$  and node  $v_j$ . We also assume the transmission time of a packet is one unit time. With  $P_{MAX}$ , we can define a direct communication graph  $G = (V, E)$  where  $V$  is the node set and  $E$  is the direct link set of all possible direct communication among nodes. Let  $N(v_i)$  represent the set of direct neighbors of node  $v_i$ , i.e., for any  $v_j \in N(v_i)$ ,  $(d(v_i, v_j))^{-\alpha} \cdot \tau \leq P_{MAX}$ .

We assume that each node has a distinctive identity and each node knows its position information either through a low-power GPS receiver or localization methods. By a single-hop exchange, each node can gather the location information of its neighbors. By using this position information, a node can calculate the distances to its neighbors and use them to optimize its transmission power.

Our CC model is similar to those of [17–20] but different from those<sup>1</sup> of [11, 24]. CC model takes advantage

<sup>1</sup> In [11, 24], a complete communication from node  $v_i$  to node  $v_j$  can be achieved by CC if  $(\sum_{v_k \in v_i \cup H(v_i)} \sqrt{P(v_k) \cdot (d(v_k, v_j))^{-\alpha}})^2 \geq \tau$ . This model is more realistic than the simplified model which we and [17–20] used. If we use this model in our load balancing scheme, it



**Fig. 1** Cooperative Communication (CC):  $v_1$  together with its helpers  $v_3$  and  $v_4$  transmit simultaneously to  $v_2$  using CC

of the physical layer design that combines partial signals containing the same information to obtain the complete information. Thus, a complete communication from node  $v_i$  to node  $v_j$  can be achieved by CC if  $v_i$  transmits the same signal simultaneously with a set of helper nodes  $H(v_i)$  and their transmission power satisfies

$$\sum_{v_k \in v_i \cup H(v_i)} P(v_k) \cdot (d(v_k, v_j))^{-\alpha} \geq \tau \quad (P(v_k) \leq P_{MAX}).$$

Figure 1 shows a simple example of CC. Node  $v_1$  wants to transmit data to node  $v_2$ .  $v_1$  can first transmit to its nearby neighbors  $v_3$  and  $v_4$ , then the three nodes transmit simultaneously to  $v_2$  using CC. If the combined SNR is larger than  $\tau$ ,  $v_2$  can successfully decode the data. Physical layer techniques for implementation of CC could be found in [29]. Carefully selecting the helper set and using CC can reduce the transmission power.  $v_1$  can directly reach  $v_2$  using at least  $P^d(v_1) = \tau \cdot (d(v_1, v_2))^\alpha$  transmission power. If the total transmission power  $P^{cc}(v_1) + P^{cc}(v_3) + P^{cc}(v_4) < P^d(v_1)$  when using CC, CC can reduce the total transmission power from the direct transmission between  $v_1$  and  $v_2$ . More importantly, CC can also spread the energy consumption among multiple nodes which can benefit the energy balancing in the network. In the above example, if  $P^{cc}(v_1) + P^{cc}(v_3) + P^{cc}(v_4) \geq P^d(v_1)$ , but  $\max\{P^{cc}(v_1), P^{cc}(v_3), P^{cc}(v_4)\}$  is smaller than  $P^d(v_1)$ , CC has potential to balance the energy, more than the direct communication between  $v_1$  and  $v_2$ . In this paper, we apply CC in the one-hop neighborhood along the energy efficient path to balance the energy in the whole network.

We assume that the initial battery level of each node  $v_i$  is  $E^0(v_i)$  and the battery level of  $v_i$  at time  $t$  is  $E^t(v_i)$ . For all

Footnote 1 continued

is difficult to get an explicit solution of the optimal remaining energy  $E_x$  (we will define it in Sect. 4). Notice that the transmission power obtained under the simplified model can always guarantee that the CC transmission succeeds under the realistic model, since  $\sum_{v_k \in v_i \cup H(v_i)} P(v_k) \cdot (d(v_k, v_j))^{-\alpha} \leq (\sum_{v_k \in v_i \cup H(v_i)} \sqrt{P(v_k) \cdot (d(v_k, v_j))^{-\alpha}})^2$ . Therefore, we believe the results obtained by our algorithm can be directly used for the realistic model, though the perfect load balancing may not be guaranteed. Similarly, in our simplified CC model, we do not consider the fading effect since it will make the optimal remaining energy hard to solve.

definitions, when the time  $t$  is clear in context, it could be ignored. When node  $v_i$  transmits a packet using transmission power  $P^t(v_i)$ , its battery level reduces to  $E^{t+1}(v_i) = E^t(v_i) - P^t(v_i) \times 1$ . Here, for simplicity, we ignore the receiving energy cost and assume that the size of a packet is the same and the time to send a packet is unified to one unit time. Thus,  $P^t(v_i) \times 1$  is the energy consumed from transmitting the packet in a unit time. But our proposed cooperative routing protocol can be easily extended to handle the cases without these assumptions. When  $E^t(v_i) \leq 0$ , node  $v_i$  is running out of its battery and dies out. Let  $E_{min}(S) = \min_{v_i \in S} E^t(v_i)$ , which is the minimum remaining energy of a node set  $S$ . As usual, we define the lifetime of the network  $T$  as the time when the first node dies out, i.e.,  $T = \{\min t \mid E^t(V) = 0\}$ .

In this paper, we assume that the underlying routing decision (the route connecting the source and the destination) has been made by certain non-CC routing strategy, such as energy-efficient ad hoc routing protocols or shortest path based routing algorithms. We only focus on how to apply CC technique to further improve the energy balancing along the selected path.

Table 1 lists all the symbols used in the paper.

#### 4 Energy-balanced cooperative routing (EBCR)

In this section, we first introduce how our proposed cooperative routing (EBCR) can balance the energy along a single path from its source  $v_s$  to its destination  $v_d$ , then we

**Table 1** Summary of notations

| Symbol             | Definition   |
|--------------------|--|
| $P_{MAX}$          | Maximum transmission power   |
| $P(v_i), P^t(v_i)$ | Transmission power of node $v_i$ (at time $t$ )  |
| $\alpha$           | Path loss exponent   |
| $\tau$             | Minimum average SNR  |
| $d(v_i, v_j)$      | Distance between node $v_i$ and node $v_j$   |
| $N(v_i)$           | One-hop neighbor nodes of node $v_i$   |
| $E^0(v_i)$         | Initial battery level of $v_i$   |
| $E^t(v_i)$         | Remaining energy of $v_i$ at time $t$  |
| $P_{ik}$           | Energy consumption to support direct link $v_i v_k$  |
| $E_{min}(S)$       | Minimum remaining energy of node set $S$ ,<br>i.e., $E_{min}(S) = \min_{v_i \in S} E^t(v_i)$ |
| $T$                | Lifetime of the network, i.e., $T = \{\min t \mid E^t(V) = 0\}$                              |
| $H(v_i)$           | Helper node set of node $v_i$ for CC   |
| $PH(v_i)$          | Potential helper nodes of $v_i$ for CC   |
| $P^h(v_i)$         | Transmission energy need for $v_i$ to directly reach the farthest helper in $PH(v_i)$        |

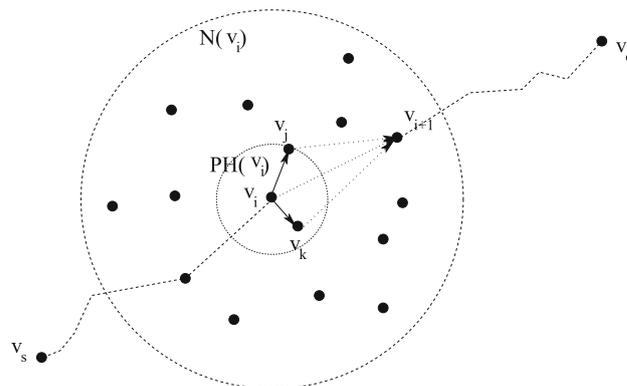
further discuss how the EBCR performs under multiple-flow routing and prove its optimality.

##### 4.1 Balancing energy along single path

As explained in last section, our cooperative routing algorithm starts from a path generated by an underlying non-cooperative routing protocol. Assume the path is  $\pi = v_0, v_1, v_2, \dots, v_h$ , where  $v_s = v_0$  and  $v_d = v_h$ . Our goal is to perform CC for each hop  $v_i v_{i+1}$  along the path, such that after the transmission  $v_i$  and its helpers  $H(v_i)$  have the balanced remaining energy and the minimum remaining energy among them is maximized. Therefore, we hope that the energy along the path is balanced too. See Fig. 2 for illustration. Here, we assume that each node  $v_i$  exchanges the information of its remaining energy and position with neighbors so that  $v_i$  also has information of its neighbors. We also assume that the energy consumed by exchanging control packets is ignorable.

In order to apply CC, we need to pick the helper set for current node  $v_i$  which will send the same packet simultaneously to  $v_{i+1}$ . We first define a potential cooperative helper set  $PH(v_i) \subset N(v_i)$ . Obviously, in  $PH(v_i)$ , we only consider those neighbors of  $v_i$  which are closer to  $v_i$  than  $v_{i+1}$  (here, the closeness is defined based on the Euclidean distance between the neighbor to the current node  $v_i$ , and this can be easily obtained since each node knows its own position and positions of its neighbors); otherwise directly sending the packet to  $v_{i+1}$  is more energy efficient. Therefore,  $PH(v_i)$  could be any subset of  $\{v_k \mid v_k \in N(v_i) \text{ and } d(v_i, v_k) < d(v_i, v_{i+1})\}$ .

Given a potential helper set  $PH(v_i)$  with size  $k$ , we first introduce an algorithm to calculate the remaining energy of  $v_i$  and its helper set  $H_k(v_i)$  under CC model. The basic idea



**Fig. 2** Illustration of EBCR to pick the helper set  $H(v_i)$  (including  $v_j$  and  $v_k$  in this example) for current node  $v_i$  which will send the same packet simultaneously to next hop node  $v_{i+1}$ . The large circle represents neighborhood  $N(v_i)$  of  $v_i$  and the small circle represents one of the potential cooperative helper sets  $PH(v_i)$  of  $v_i$

of the algorithm is as follows. We first calculate the transmission power  $P^h(v_i)$  needed to reach the farthest neighbor inside  $PH(v_i)$ , then use it to update the remaining energy of  $v_i$  by  $E^{t+1}(v_i) = E^t(v_i) - P^h(v_i)$ . For other nodes  $v_j$  in  $PH(v_i)$ , the remaining energy is  $E^{t+1}(v_j) = E^t(v_j)$ . Then we sort all these  $k + 1$  nodes (defined as set  $A$ ) in the descending order by its remaining energy  $E^{t+1}(v_j)$ . Assume that the sorted set is  $\{v'_1, v'_2, \dots, v'_{k+1}\}$ . Our algorithm will greedily pick those nodes with larger remaining energy to be helpers of  $v_i$ , until their cumulated signal strength at  $v_{i+1}$  is larger than or equal to  $\tau$ , i.e.,  $\sum_{j=1}^w (E^{t+1}(v'_j) - E^{t+1}(v'_{w+1})) (d(v'_j, v_{i+1}))^{-\alpha} \geq \tau$ . If the cumulative power strength of the first  $w$  nodes in  $PH(v_i)$  is enough to reach  $v_{i+1}$ , our algorithm will just use these  $w$  nodes as  $v_i$ 's helpers during CC. We try to balance every helper's remaining energy after the transmission. Let  $E_x$  be the remaining energy of all  $w$  nodes. We need

$$\sum_{j=1}^w (E^{t+1}(v'_j) - E_x) (d(v'_j, v_{i+1}))^{-\alpha} = \tau.$$

Thus,

$$E_x = \frac{\sum_{j=1}^w E^{t+1}(v'_j) (d(v'_j, v_{i+1}))^{-\alpha} - \tau}{\sum_{j=1}^w (d(v'_j, v_{i+1}))^{-\alpha}}.$$

If the energy consumption of  $v_i$  with CC ( $E^{t+1}(v_i) - E_x$ ) +  $P^h(v_i)$  is less than the energy consumption of direct transmission  $\tau \cdot (d(v_i, v_{i+1}))^\alpha$ , we return  $E_x$  as the estimated energy and the first  $w$  nodes  $\{v'_1, v'_2, \dots, v'_w\}$  as the helper set  $H_k(v_i)$ . If  $E^{t+1}(v_i) - E_x + P^h(v_i) \geq \tau \cdot (d(v_i, v_{i+1}))^\alpha$ , then CC with nodes in  $PH(v_i)$  is not useful and  $v_i$  directly sends the packet to  $v_{i+1}$ . In that case, we return  $E_k^{t+1}(v_i) =$

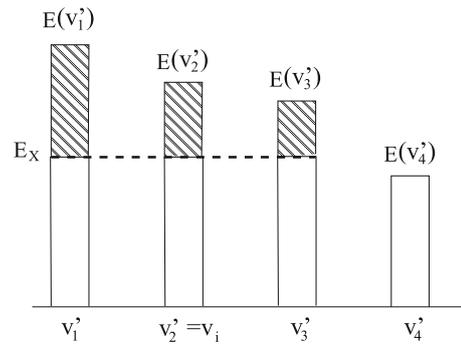


Fig. 3 Example for Algorithm 1

$-\infty$  and  $H_k(v_i) = \{v_i\}$ . Algorithm 1 shows the detailed algorithm. By running this algorithm, for a given  $HP(v_i)$ , we can decide which nodes have to involve into the cooperative routing and what are their transmission power during CC.

Figure 3 illustrates a simple example where there are 4 nodes inside  $A = PH(v_i) \cup \{v_i\}$ . Under CC model, the sender  $v_i$  first sends the packet to other nodes in  $A$ . Then  $v_i$  refreshes its remaining energy. The sorted remaining energy of the nodes in  $A$  is shown in the figure. The sender  $v_i$  is assumed to have the second largest remaining energy, i.e.  $v'_2 = v_i$ . Based on Algorithm 1, we try to find first  $w$  nodes whose cumulated signal strength at  $v_{i+1}$  is strong enough as  $\tau$ . In this example,  $w = 3$ . Then a target remaining energy  $E_x$  is calculated. As the output of Algorithm 1,  $v'_1$  and  $v'_3$  will help  $v'_2$  ( $v_i$ ) to perform CC transmission. After the transmission, the remaining energy of these three nodes is  $E_x$ .

---

**Algorithm 1** Calculate Remaining Energy and Helper Set

---

**Input:** Potential helper set  $PH(v_i)$  and its size  $k = |PH(v_i)|$ .

**Output:** Estimated remaining energy  $E_k^{t+1}(v_i)$  of  $v_i$  and its corresponding helper set  $H_k(v_i)$ .

- 1: Let  $A = PH(v_i) \cup \{v_i\}$ .
  - 2: Calculate transmission energy of  $v_i$  need to reach the farthest helper:  $P^h(v_i) = \max_{v_j \in A} (d(v_i, v_j))^\alpha \cdot \tau$ .
  - 3: Update the estimated remaining energy:  $E^{t+1}(v_i) = E^t(v_i) - P^h(v_i)$  and  $E^{t+1}(v_j) = E^t(v_j)$  for every other node  $v_j \in A$ .
  - 4: Sort all elements in  $A$  in the descending order of its remaining energy  $E^{t+1}(v_j)$ . Assume that the sorted set is  $\{v'_1, v'_2, \dots, v'_{k+1}\}$ .
  - 5:  $w = 1$ .
  - 6: **while**  $w \leq k$  and  $\sum_{j=1}^w (E^{t+1}(v'_j) - E_x) (d(v'_j, v_{i+1}))^{-\alpha} < \tau$  **do**
  - 7:      $w++$ .
  - 8: **end while**
  - 9: Let  $E_x = \frac{\sum_{j=1}^w E^{t+1}(v'_j) (d(v'_j, v_{i+1}))^{-\alpha} - \tau}{\sum_{j=1}^w (d(v'_j, v_{i+1}))^{-\alpha}}$
  - 10: **if**  $E^{t+1}(v_i) - E_x + P^h(v_i) < \tau \cdot (d(v_i, v_{i+1}))^\alpha$  **then**
  - 11:     return  $E_k^{t+1}(v_i) = E_x$  and  $H_k(v_i) = \{v'_1, v'_2, \dots, v'_w\}$
  - 12: **else**
  - 13:     return  $E_k^{t+1}(v_i) = -\infty$  and  $H_k(v_i) = \{v_i\}$
  - 14: **end if**
-

Now we are ready to present our main algorithm (Algorithm 2) of the proposed EBCR. Basically, it tries all possible initial setting of  $PH(v_i)$  by running Algorithm 1 for each setting and picks the solution with the largest remaining energy as the final decision. If the best solution  $E_{k^*}^{t+1}(v_i) \geq 0$ , we perform CC with its helper set  $H_{k^*}(v_i)$ , otherwise, send the packet directly to  $v_{i+1}$ . Notice that the proposed EBCR algorithm only use 1-hop neighbor information around node  $v_i$  and its time complexity is only  $O(m^2)$  where  $m$  is the number of neighbors whose distances from  $v_i$  are less than  $v_j$ 's. Since usually the number of neighbors of a node is small unless the network is very crowded, the additional computation cost of the proposed method over the routing task is very limited and the requirement of computation capability on each individual node is very low.

Our cooperative routing problem can efficiently balance the remaining energy among nodes. Figure 4 illustrates an example with just three nodes. Assume that all nodes can communicate with each other directly and have the same initial energy  $32\tau$ , i.e.,  $E^0(v_1) = E^0(v_2) = E^0(v_3) = 32\tau$ . The distances among them are marked in the figure. Let  $\alpha = 2$  and the underlying routing is minimum energy routing. Assume that we want to send a packet from  $v_1$  to  $v_3$ . Without CC,  $v_1$  need transmit at power of  $16\tau$ , thus  $E_{v_1} = 32\tau - 16\tau = 16\tau$ . Then  $E_{min}(V) = 16\tau$ . If with our cooperative routing,  $v_1$  will choose  $v_2$  as its helper. First,  $v_1$  needs to send the packet to  $v_2$  using the transmission power of  $4\tau$ . Then  $v_1$  and  $v_2$  transmit the packet simultaneously to  $v_3$  with transmission power of  $6\tau$  and  $10\tau$  respectively, so that the power strength at  $v_3$  is  $6\tau \cdot 4^{-2} + 10\tau \cdot 4^{-2} = \tau$ . After the transmission, the remaining energy of both  $v_1$  and  $v_2$  is  $22\tau$ . Thus, the minimum remaining energy of the network  $E_{min}(V) = 22\tau$ , which is larger than the one without CC. In addition, the remaining energy among all the three nodes are more balanced.

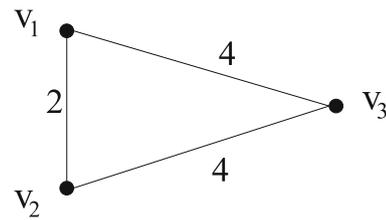


Fig. 4 Example of cooperative routing with a single flow

Next we formally prove that our proposed EBCR can indeed balance remaining energy of the network. Recall that the minimum remaining energy of a network is the minimum of remaining energy of individual nodes in the network.

**Theorem 1** For a routing task between a pair of source and destination nodes, the proposed EBCR can improve the minimum remaining energy of the network.

*Proof* Assume that  $E_{min}(V)$  is the remaining energy of the network  $V$  with the proposed cooperative routing, while  $E'_{min}(V)$  is the remaining energy of the network without using the proposed cooperative routing. In the network, there are three types of nodes: nodes in the original path  $\pi$  (denoted as  $R$ ), nodes acting as helpers during CC (denoted as  $H$ ), and nodes which do not participate in any packet forwarding process even under CC (denoted as  $U$ ). So  $V = R \cup H \cup U$ . Assume that  $E_{min}(S)$  and  $E'_{min}(S)$  are the minimum remaining energy of node set  $S$  with and without cooperative routing, respectively. Then we consider the minimum remaining energy of these three types of nodes separately. First, the remaining energy of  $U$  does not change, i.e.,  $E_{min}(U) = E'_{min}(U)$ . Second, the remaining energy of helpers  $H$  may be reduced by using CC ( $E_{min}(H) \leq E'_{min}(H)$ ). However, since our cooperative routing guarantees that all helpers' remaining energy is not less than the sender's remaining energy after cooperative routing,  $E_{min}(H) \geq E_{min}(R)$ . This

---

**Algorithm 2** Energy-Balanced Cooperative Routing (EBCR)

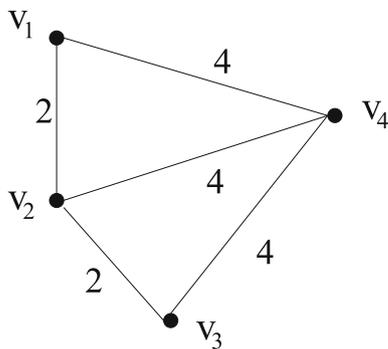
---

- 1: Sort all nodes  $v_k \in N(v_i)$  in increasing order of  $d(v_i, v_k)$ . Assume that the ordered list is  $v'_1, v'_2, \dots, v'_m, v'_{m+1}, \dots, v'_{|N(v_i)|}$  and  $v'_{m+1} = v_j$ .
  - 2: **for**  $k = 1, 2, \dots, m$  **do**
  - 3:   Let  $PH(v_i) = \{v'_1, v'_2, \dots, v'_k\}$ .
  - 4:   Run Algorithm 1 which returns  $E_k^{t+1}(v_i)$  and  $H_k(v_i)$ .
  - 5: **end for**
  - 6: Among  $m$  outputs, let  $E_{k^*}^{t+1}(v_i)$  be the largest remaining energy of  $v_i$  and  $H_{k^*}(v_i)$  be the corresponding helper set.
  - 7: **if**  $E_{k^*}^{t+1}(v_i) \geq 0$  **then**
  - 8:   Node  $v_i$  sends the packet to nodes in  $H_{k^*}(v_i)$ .
  - 9:   Assign all nodes  $v_j \in H_{k^*}(v_i)$  transmit the packet simultaneously to  $v_{i+1}$  using power  $E^t(v_j) - E_{k^*}^{t+1}(v_i)$ .
  - 10: **else**
  - 11:    $v_i$  send packet to  $v_{i+1}$  directly.
  - 12: **end if**
-

also shows  $E'_{min}(H) \geq E_{min}(R)$ . Third, the remaining energy of  $R$  must be extended by using CC, thus  $E_{min}(R) \geq E'_{min}(R)$ . Since  $E_{min}(V) = \min \{E_{min}(R), E_{min}(H), E_{min}(U)\} = \min \{E_{min}(R), E_{min}(U)\}$  and  $E'_{min}(V) = \min \{E'_{min}(R), E'_{min}(H), E'_{min}(U)\}$ , we now have  $E_{min}(V) \geq E'_{min}(V)$ . Therefore, using cooperative routing can indeed lead to larger minimum remaining energy of the network.

#### 4.2 Balancing energy along multiple routes

So far, we only consider to cooperatively route a single flow in the network. But the proposed cooperative routing method can also handle multiple flows by serving them one by one. However, different serving order may lead to different remaining energy of the network. Notice that the remaining energy of nodes changes after serving one flow, thus the selected helper sets for each flow may vary under different serving orders. Figure 5 illustrates such an example. There are two flows from node  $v_1$  and  $v_3$  to  $v_4$ , respectively. Assume that the initial energy of the nodes are  $E^0(v_1) = 32\tau$ ,  $E^0(v_2) = 44\tau$ ,  $E^0(v_3) = 24\tau$ , and  $E^0(v_4) = 36\tau$ , respectively. If we apply our cooperative routing, both  $v_1$  and  $v_3$  will use  $v_2$  as its helper to perform cooperative routing. However,  $v_2$  helps which flow first will affect the remaining energy of the network. Case 1:  $v_2$  helps  $v_1$  to transmit its packet to  $v_4$  first. After four rounds' transmission, the final remaining energy of the four nodes are  $E(v_1) = 28\tau$ ,  $E(v_2) = E(v_3) = 16\tau$ , and  $E(v_4) = 36\tau$ . Thus, the minimum remaining energy of the network is  $16\tau$ . Case 2:  $v_2$  helps  $v_3$  to transmit its packet to  $v_4$  first. After four rounds' transmission, the final remaining energy of all nodes are  $E(v_1) = E(v_2) = E(v_3) = 20\tau$ , and  $E(v_4) = 36\tau$ . Thus, the remaining energy of the network is  $20\tau$ . Therefore, the remaining energy level of the network is different with different order. But, for both cases, the remaining energy of the network is better than without cooperative routing which is  $8\tau$ .



**Fig. 5** Example of EBCR routing with multiple flows. Different serving order of CCs will lead to different remaining energy level of the network

Fortunately, with any serving order our cooperative routing protocol can guarantee the improvement of minimum remaining energy of the network compared with the routing algorithms without using cooperative routing.

**Theorem 2** For a routing task between  $k$  pairs of source and destination nodes, the proposed EBCR can improve the minimum remaining energy of the network.

*Proof* Again let  $E_{min}(S)$  and  $E'_{min}(S)$  be the minimum remaining energy of node set  $S$  with and without cooperative routing, respectively. When  $k = 1$ , we already prove that  $E_{min}(V) \geq E'_{min}(V)$  in Theorem 1. Now we consider that  $k > 1$ , i.e., there are  $k$  packet flows in the network. We want to prove that after all of these  $k$  packets  $p_1, p_2, \dots, p_k$  arrive to their final destinations,  $E_{min}(V) \geq E'_{min}(V)$ .

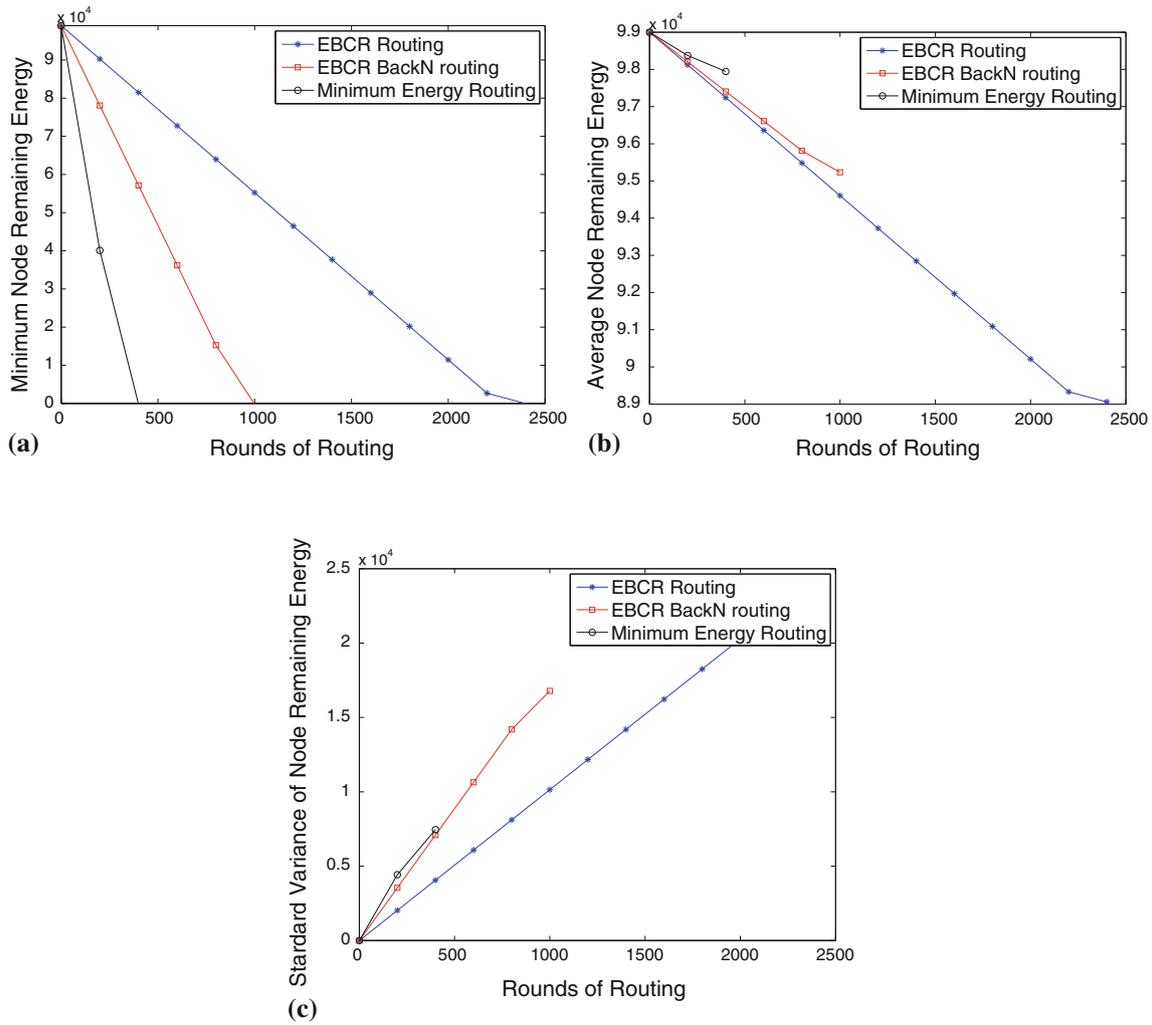
The proving technique is similar with the one used in Theorem 1. We now divide the all nodes  $V$  into  $k + 1$  disjoint node sets. With the cooperative routing, we divide  $V$  into node sets:  $R_1, R_2, \dots, R_k, U$ . Here,  $R_i$  includes a subset of nodes participate in the cooperative routing of packet  $p_i$  and  $U$  is the set of nodes which do not participate any route. If a node  $v_k$  participates multiple flows, it only belongs to the set  $R_i$  in the last flow it participates (i.e.,  $p_i$  is the last packet it transmits). Obviously,  $V = R_1 \cup R_2 \cup \dots \cup R_k \cup U$ . Similarly, we can define  $k + 1$  node sets for the case without cooperative routing, and let them be  $R'_1, R'_2, \dots, R'_k$  and  $U'$ .  $V = R'_1 \cup R'_2 \cup \dots \cup R'_k \cup U'$ .

First,  $U \subset U'$  since some nodes in  $U'$  will be helpers in cooperative routing. Thus,  $E_{min}(U) \geq E'_{min}(U')$ .

Assume that  $v_{min}$  is the node with least remaining energy in set  $R_j$ . It is either a node on the original route of packet  $p_j$  or a helper for a node on that route. If it is a node on the original route of packet  $p_j$ , the remaining energy of  $v_{min}$  must be larger than or equal to the remaining energy of the same node in the set  $R'_j$ , because our algorithm can guarantee to enlarge the remaining energy of node  $v_i$  at each step. Thus,  $E_{min}(R_j) = E(v_{min}) \geq E'(v_{min}) \geq E_{min}(R'_j)$ . If the node  $v_{min}$  is a helper of a node  $v_p$  on that route of packet  $p_j$ , there are three cases and we discuss them one by one.

**Case 1** The last packet handled by  $v_p$  is  $p_j$ , i.e.,  $v_p \in R_j$ . Based on our algorithm,  $E(v_{min}) \geq E(v_p)$ , since  $v_{min}$  is the helper of  $v_p$ . Notice that the remaining energy of  $v_p$  must also be larger than or equal to the remaining energy of the same node in the set  $R'_j$ . Consequently,  $E_{min}(R_j) = E(v_{min}) \geq E(v_p) \geq E'(v_p) \geq E'_{min}(R'_j)$ .

**Case 2** The last packet handled by  $v_p$  is not  $p_j$  but another packet  $p_s$ , and  $v_p$  is on the route of packet  $p_s$ . Thus,  $v_p \in R_s$  and  $v_p \in R'_s$ . Recall that when  $v_p$  involves into  $p_j$ 's forwarding, the remaining energy of  $v_{min}$  is already larger than or equal to the remaining energy of  $v_p$ , thus  $E(v_{min}) \geq E(v_p)$ . And in the end,



**Fig. 6** Results on a 200-node random network with a fixed source-destination pair. **a** Minimum node remaining energy, **b** average node remaining energy, **c** standard variance of node remaining energy

**Table 2** Lifetime (in rounds) of random networks with a fixed source-destination pair

| Minimum energy routing | EBCR routing | EBCR back-N routing |
|------------------------|--------------|---------------------|
| 951                    | 1,682        | 951                 |
| 1,083                  | 1,311        | 1,083               |
| 457                    | 1,250        | 839                 |
| 477                    | 1,304        | 477                 |
| 1,132                  | 2,882        | 1,132               |
| 336                    | 2,261        | 945                 |
| 731                    | 1,614        | 731                 |
| 737                    | 1,591        | 1,046               |
| 754                    | 967          | 815                 |

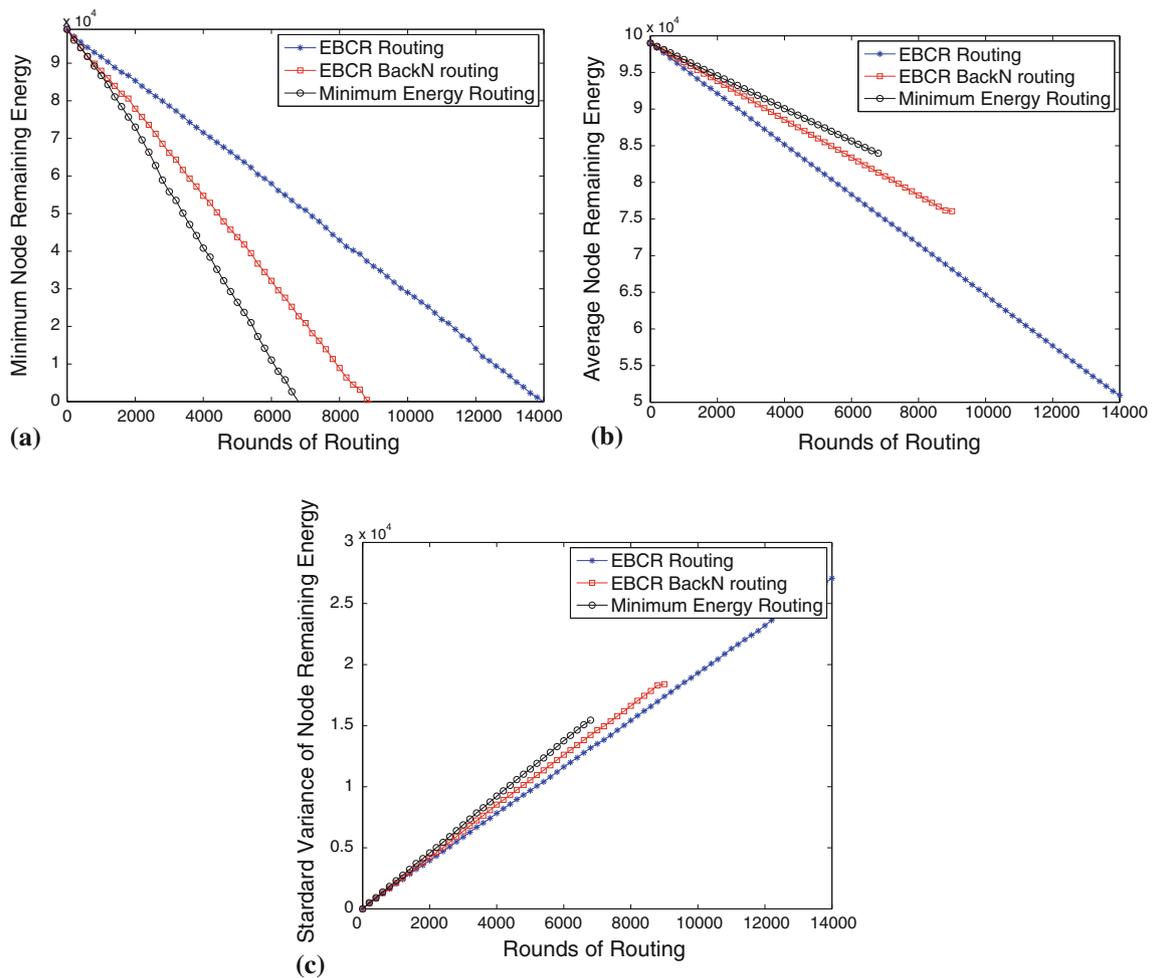
the remaining energy of  $v_p$  is also greater than or equal to the remaining energy of the same node in  $R'_s$  at that time. Therefore,  $E_{min}(R_j) = E(v_{min}) \geq E(v_p) \geq E'(v_p) \geq E'_{min}(R'_s)$ .

**Case 3** The last packet handled by  $v_p$  is not  $p_j$  but another packet  $p_s$ , and  $v_p$  is a helper for a node  $v_q$  on the route of packet  $p_s$ . Based on our cooperative routing algorithm,  $E(v_p) \geq E(v_q)$  and  $E(v_q) \geq E'(v_q)$ . Hence,  $E_{min}(R_j) = E(v_{min}) \geq E(v_p) \geq E(v_q) \geq E'(v_q) \geq E'_{min}(R'_s)$ .

In summary, for any set  $R_i$ , we can always find a set  $R'_j$  such that  $E_{min}(R_i) \geq E'_{min}(R'_j)$ . Since  $E_{min}(U) \geq E'_{min}(U')$ ,  $E_{min}(V) = \min(E_{min}(R_1), \dots, E_{min}(R_k), E_{min}(U))$  and  $E'_{min}(V) = \min(E'_{min}(R'_1), \dots, E'_{min}(R'_k), E'_{min}(U'))$ , we know that  $E_{min}(V) \geq E'_{min}(V)$ .

### 5 Simulation

In this section, we evaluate our proposed EBCR protocol by comparing its performances with the minimum energy path based routing protocol. In addition, we also implement a variation of our EBCR, EBCR back-N, in which the



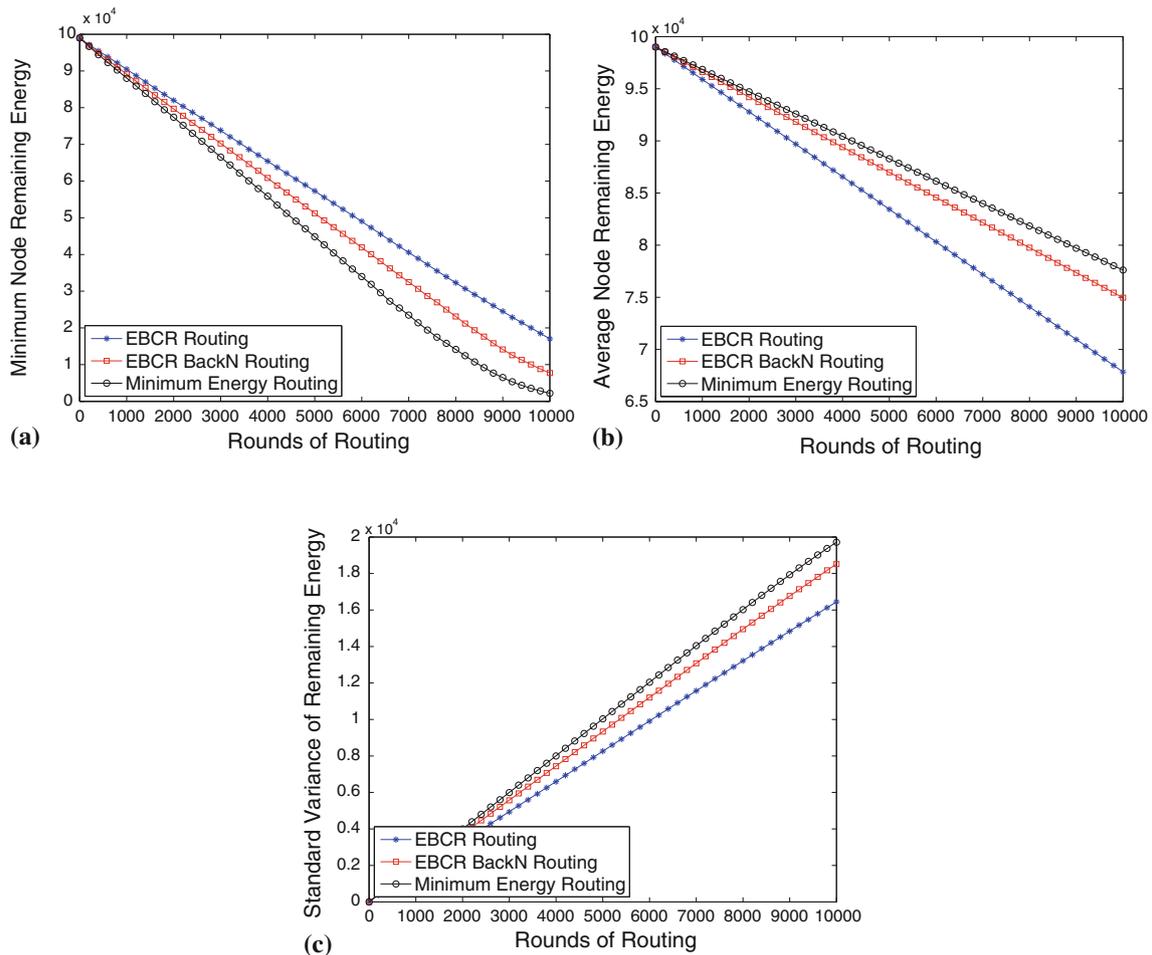
**Fig. 7** Results on a 200-node random network with random source-destination pairs. **a** Minimum node remaining energy, **b** average node remaining energy, **c** standard variance of node remaining energy

**Table 3** Lifetime (in rounds) of random networks with random source-destination pairs

| Minimum energy routing | EBCR routing | EBCR back-N routing |
|------------------------|--------------|---------------------|
| 7,483                  | 11,629       | 7,483               |
| 9,002                  | 15,815       | 13,606              |
| 10,056                 | 12,284       | 10,856              |
| 8,103                  | 11,421       | 9,786               |
| 8,454                  | 13,508       | 10,380              |
| 8,485                  | 11,411       | 8,485               |
| 8,977                  | 12,218       | 8,977               |
| 6,759                  | 13,962       | 8,842               |
| 9,247                  | 12,502       | 9,247               |

helpers of  $v_i$  during CC are selected from the last  $N$ -hop nodes along the minimum energy path instead of one-hop neighbors of  $v_i$ . However, the method to select helpers and assign transmission power is the same with EBCR. The

underlying wireless networks are randomly generated and have 200 nodes. For convenience, we set the path loss factor  $\alpha = 2$  and the SNR threshold  $\tau = 1$ . The initial energy level of each node  $E^0(v_i)$  is set to 99,000 units. Simulations are run in rounds. Each round we pick one source and one destination and perform one round of routing using different routing algorithms. We ignore the energy consumption of control packets and only consider the energy consumed by data packets, since the size of control packets is relevantly smaller than the size of data packets. Notice that we do consider the energy consumed by sending the data packets to the selected helper set at each node during CC operations, which is the major energy overhead of the proposed method. In addition, as shown in Sect. 4, the computational complexity of the proposed algorithm is just  $O(m^2)$  and  $m$  is smaller than the number of neighbors of the current node which is usually a small number. Thus, we also ignore the energy consumed from computations.



**Fig. 8** Results over 10,000 routes on 50 200-node random networks with random source- destination pairs in each round. **a** Minimum node remaining energy, **b** average node remaining energy, **c** standard variance of node remaining energy

In the simulations, we take two metrics as the performance measurements:

- *Node Remaining Energy*: current energy level of each node  $E^t(v_i)$ . We report the average, minimum, and standard variance of node remaining energy of all nodes in the network.
- *Lifetime*: the lifetime of the network is defined as the total number of rounds of routing before the first node runs out of its power.

In the first set of simulations, we generate nine 200-node random networks. For each network, we randomly pick a pair of nodes as the source and the destination and keep running the routing between them using different routing methods, until the first node runs out of its power. In other words, this is a single flow case. The minimum energy routing protocol sends the packet directly along the minimum energy path between the source and the destination, while our EBCR (or EBCR back-N) method uses the

neighbors (or last  $N$ -hop) of the nodes on the minimum energy path to cooperatively forward the packet. Figure 6 shows the results of one random network. Figure 6(a) plots the minimum node remaining energy of the network at each 200 rounds. With more routing task completed, the minimum node remaining energy of the network reduces. From the simulation results, the EBCR algorithm has the best performance among the three algorithms. The lifetime of EBCR is over six time of the one of the minimum energy routing. This confirms our theoretical proof in Section 4. Notice that EBCR back-N performs worse than EBCR, but still better than the minimum energy routing. Figure 6(b) shows the average node remaining energy of the network at each 200 rounds. Since the minimum energy routing protocol aims to minimize the total transmission energy, its average node remaining energy is better than the other two methods. However, its lifetime is much shorter than EBCR's. Figure 6(c) shows the standard variance of remaining energy among nodes. The standard variance of

node remaining energy of our EBCR routing is the minimum among those of all methods. This shows that it can balance remaining energy among nodes very well. Table 2 shows the lifetime of the nine different networks. The conclusions are consistent.

In the second set of simulations, we again generate nine 200-node random networks. Instead of using one fixed source-destination pair, we randomly pick a source-destination pair in each round. Thus, it is a multi-flow case. For each network, we keep running the routing for random source-destination pairs, until the first node runs out of its power. Figure 7 shows the results of one network. Clearly, the lifetime of the network is much longer than the one in the first set of simulation. From the results in Fig. 7, EBCR performs best among all methods in term of minimum remaining energy and energy balancing. Again, the curves of EBCR back-N and minimum energy routing are much shorter than those of EBCR. This confirms that our proposed method can prolong the lifetime significantly. Table 3 also shows the lifetime of all nine networks. Clearly, EBCR can prolong the lifetime of the network in multi-flow networks too.

For the third set of simulations, we again test all algorithms for the multi-flow cases over 50 different networks. For each network, we run 10,000 rounds (i.e., 10,000 routes) and record the node remaining energy after each 200 rounds. Results over 50 networks are averaged and plotted in Fig. 8. The advantage of EBCR over other two methods is obvious too. In summary, our EBCR algorithm can always balance the remaining energy among nodes and prolong the lifetime of the whole networks.

## 6 Conclusion

In this paper, we study the impact of CC on energy balancing in multihop routing. We introduce a novel routing scheme (EBCR) to select cooperative relay nodes and their transmission power for each hop. It can be applied to any underlying energy-aware routing protocol, and only need local information to perform the optimization on CCs. We formally prove that our cooperative routing method EBCR can indeed balance the energy among nodes and prolong the remaining lifetime of the network. Simulation results confirm the nice performance of our proposed method over the minimum energy routing. Even though our proposed method can balance the energy over any energy-aware routing protocols, it has several limitations too: (1) our proposed scheme does not work for mobile networks, since node movement will break the optimal selection of cooperative relays; (2) we assume a simplified CC model where fading effect is not considered; (3) in our proposed scheme, each node needs to know its position information, which

requires certain devices (such as GPS) equipped or additional communication costs. We leave further improvement and implementation of our proposed cooperative routing methods in real networking systems as our next steps.

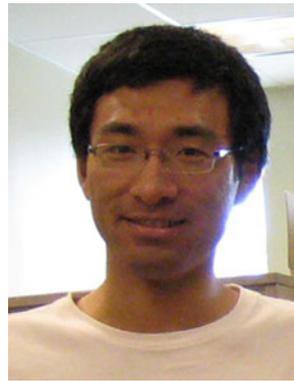
**Acknowledgments** This work is supported in part by the US National Science Foundation (NSF) under Grant No. CNS-0721666, CNS-0915331, and CNS-1050398, and by Tsinghua National Laboratory for Information Science and Technology (TNList). This work was partially done when Y. Li visited the Department of Computer Science, University of North Carolina at Charlotte, with a scholarship from the China Scholarship Council. The authors also greatly appreciate the anonymous reviewers and the editor for their constructive comments for improving the paper.

## References

- Doshi, S., Bhandare, S., & Brown, T. X. (2002). An on-demand minimum energy routing protocol for a wireless ad hoc network. *SIGMOBILE Mobile Computing and Communication Review*, 6(3), 50–66.
- Kar, K., Kodialam, M., Lakshman, T., & Tassiulas, L. (2003). Routing for network capacity maximization in energy-constrained ad-hoc networks. In *Proceedings of IEEE conference on computer communications (INFOCOM)*.
- Chang, J. H., & Tassiulas, L. (2000). Energy conserving routing in wireless ad-hoc networks. In *Proceedings of IEEE conference on computer communications (INFOCOM)*.
- Li, Q., Aslam, J., & Rus, D. (2001). Online power-aware routing in wireless ad-hoc networks. In *Proceedings of the seventh annual international conference on mobile computing and networking*. Rome, Italy: ACM Press, pp. 97–107.
- Dong, Q., Banerjee, S., Adler, M., & Misra, A. (2005). Minimum energy reliable paths using unreliable wireless links. In *MobiHoc '05: proceedings of the 6th ACM international symposium on mobile ad hoc networking and computing*. New York, NY, USA: ACM Press, pp. 449–459.
- Stojmenovic, I., & Lin, X. (2001). Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(10).
- Melodia, T., Pompili, D., & Akyildiz, I. F. (2004). Optimal local topology knowledge for energy efficient geographical routing in sensor networks. In *Proceedings of IEEE conference on computer communications (INFOCOM)*.
- Stojmenovic, I., & Datta, S. (2004). Power and cost aware localized routing with guaranteed delivery in wireless networks. *Wireless Communications and Mobile Computing*, 4(2), 175–188.
- Seada, K., Zuniga, M., Helmy, A., & Krishnamachari, B. (2004). Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks. In *Proceedings of ACM conference on embedded networked sensor systems (Sensys)*.
- Li, C.-P., Hsu, W.-J., Krishnamachari, B., & Helmy, A. (2005). A local metric for geographic routing with power control in wireless networks. In *Proceedings of annual IEEE communications society conference on sensor, mesh, and ad hoc communications and networks (SECON)*.
- Pandana, C., Siriwongpairat, W., Himsoon, T., & Liu, K. (2006). Distributed cooperative routing algorithms for maximizing network lifetime. In *Proceedings of IEEE wireless communications and networking conference (WCNC)*.
- Laneman, N., Tse, D., & Wornell, G. (2004). Cooperative diversity in wireless networks: Efficient protocols and outage behavior. *IEEE Transactions on Information Theory*, 50(12), 3062–3080.

13. Nosratinia, A., Hunter, T. E., & Hedayat, A. (2004). Cooperative communication in wireless networks. *IEEE Communications Magazine*, 42(10), 74–80.
14. Agarwal, M., Cho, J., Gao, L., & Wu, J. (2004). Energy efficient broadcast in wireless ad hoc networks with hitch-hiking. In *Proceedings of IEEE conference on computer communications (INFOCOM)*.
15. Wu, J., Cardei, M., Dai, F., & Yang, S. (2006). Extended dominating set and its applications in ad hoc networks using cooperative communication. *IEEE Transactions on Parallel and Distributed Systems*, 17, 851–864.
16. Jakllari, G., Krishnamurthy, S., Faloutsos, M., & Krishnamurthy, P. (2007). On broadcasting with cooperative diversity in multi-hop wireless networks. *IEEE Journal on Selected Area in Communications*, 25, 484–496.
17. Cardei, M., Wu, J., & Yang, S. (2006). Topology control in ad hoc wireless networks using cooperative communication. *IEEE Transactions on Mobile Computing*, 5(6), 711–724.
18. Yu, J., Roh, H., Lee, W., Pack, S., & Du, D. -Z. (2010). Cooperative bridges: Topology control in cooperative wireless ad hoc networks. In *Proceedings of the 29th IEEE conference on computer communications (IEEE INFOCOM 2010)*.
19. Zhu, Y., Huang, M., Chen, S., & Wang, Y. (2012). Energy-efficient topology control in cooperative ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 23(8), 1480–1491.
20. Wang, L., Liu, B., Goeckel, D., Towsley, D. & Westphal C. (2008). Connectivity in cooperative wireless ad hoc networks. In *MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, Hong Kong, China.
21. Shi, Y., Sharma, S., & Hou, Y. (2008) Optimal relay assignment for cooperative communications. In *MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, Hong Kong, China.
22. Li, F., Wu, K., & Lippman, A. (2006). Energy-efficient cooperative routing in multi-hop wireless ad hoc networks. In *Proceedings of IEEE international performance computing and communications conference (IPCCC 2006)*.
23. Khandani, A., Abounadi, J., Modiano, E., & Zheng, L. (2007). Cooperative routing in static wireless networks. *IEEE Transaction on Communications*, 55(11), 2185–2192.
24. Ibrahim, A., Han, Z., & Liu, K., (2008). Distributed energy-efficient cooperative routing in wireless networks. *IEEE Transactions on Wireless Communications*, 7(10), 3930–3941.
25. Zhang, J., & Zhang, Q. (2008). Cooperative routing in multi-source multi-destination multi-hop wireless networks. In *Proceedings of IEEE conference on computer communications (INFOCOM)*.
26. Maham, B., Narasimhan, R., & Hjørungnes, A. (2009). Energy-efficient space-time coded cooperative routing in multihop wireless networks. In *Proceedings of 28th IEEE conference on global telecommunications, (GLOBECOM 2009)*.
27. Madan, R., Mehta, N., Molisch, A., & Zhang, J. (2009). Energy-efficient decentralized cooperative routing in wireless networks. *IEEE Transactions on Automatic Control*, 54, 512–527.
28. Sharma, S., Shi, Y., Hou, Y., Sherali, H. D., & Kompella, S. (2010). Cooperative communications in multi-hop wireless networks: Joint flow routing and relay node assignment. In *Proceedings of IEEE conference on computer communications (INFOCOM)*.
29. Jakllari G., Krishnamurthy, S. V., Faloutsos, M., Krishnamurthy, P. V., & Ercetin, O. (2006). A framework for distributed spatio-temporal communication in mobile ad hoc networks. In *Proceedings of IEEE conference on computer communications (INFOCOM)*, Barcelona, Spain.

## Author Biographies



**Siyuan Chen** received his B.S. degree from Peking University, China in 2006. He is currently a Ph.D. student in the University of North Carolina at Charlotte, majoring in computer science. His current research focuses on wireless networks, ad hoc and sensor networks, and algorithm design.



**Yang Li** is currently a Ph.D. student in computer science at Beijing Institute of Technology, China. She was a visiting Ph.D. student in the University of North Carolina at Charlotte, USA during 2010–2011. Her current research focuses on wireless networks, mobile ad hoc networks, and delay tolerance networks.



**Minsu Huang** received his B.Eng. degree in mechanical engineering from Central South University in 2003 and his M.S. degree in computer science from Tsinghua University in 2006. He is currently a PhD student in the University of North Carolina at Charlotte, majoring in computer science. His current research focuses on wireless networks, ad hoc and sensor networks, and algorithm design.



**Ying Zhu** received her B.Eng. degree in automatic control and M.Eng. degree in pattern recognition from University of Electronic Science and Technology of China in 2002 and 2005, respectively. She is currently a Ph.D. student in the University of North Carolina at Charlotte, majoring in computer science. Her current research focuses on wireless networks, delay tolerant networks, and algorithm design.