Energy-Efficient Reliable Routing Considering Residual Energy in Wireless Ad Hoc Networks

Javad Vazifehdan, R. Venkatesha Prasad, and Ignas Niemegeers

Abstract—We propose two novel energy-aware routing algorithms for wireless ad hoc networks, called reliable minimum energy cost routing (RMECR) and reliable minimum energy routing (RMER). RMECR addresses three important requirements of ad hoc networks: energy-efficiency, reliability, and prolonging network lifetime. It considers the energy consumption and the remaining battery energy of nodes as well as quality of links to find energy-efficient and reliable routes that increase the operational lifetime of the network. RMER, on the other hand, is an energy-efficient routing algorithm which finds routes minimizing the total energy required for end-to-end packet traversal. RMER and RMECR are proposed for networks in which either hop-by-hop or end-to-end retransmissions ensure reliability. Simulation studies show that RMECR is able to find energy-efficient and reliable routes similar to RMER, while also extending the operational lifetime of the network. This makes RMECR an elegant solution to increase energy-efficiency, reliability, and lifetime of wireless ad hoc networks. In the design of RMECR, we consider minute details such as energy consumed by processing elements of transceivers, limited number of retransmissions allowed per packet, packet sizes, and the impact of acknowledgment packets. This adds to the novelty of this work compared to the existing studies.

Index Terms—Energy-aware routing, battery-aware routing, end-to-end and hop-by-hop retransmission, reliability, wireless ad hoc networks

1 Introduction

Energy-efficient routing is an effective mechanism for reducing energy cost of data communication in wireless ad hoc networks. Generally, routes are discovered considering the energy consumed for end-to-end (E2E) packet traversal. Nevertheless, this should not result in finding less reliable routes or overusing a specific set of nodes in the network. Energy-efficient routing in ad hoc networks is neither complete nor efficient without the consideration of reliability of links and residual energy of nodes. Finding reliable routes can enhance quality of the service. Whereas, considering the residual energy of nodes in routing can avoid nodes from being overused and can eventually lead to an increase in the operational lifetime of the network.

During the last decade, various routing algorithms have been proposed aiming at increasing energy-efficiency, reliability, and the lifetime of wireless ad hoc networks (e.g., [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]). We can broadly group them into three categories. The first category includes algorithms that consider the reliability of links to find more reliable routes. For instance, De Couto et al. [1] introduced the notion of expected transmission count (ETX) to find reliable routes that consist of links requiring less number of retransmissions for lost packet recovery. Although such routes may consume less energy since they require less number of retransmissions, they do not necessarily minimize the energy consumption for E2E packet traversal. Furthermore, considering a higher priority for reliability of routes may result in overusing some nodes. If there are some links more reliable than others, these links will frequently be used to forward packets. Nodes along these links will then fail quickly, since they have to forward many packets on behalf of other nodes.

The second category includes algorithms that aim at finding energy-efficient routes (e.g., the proposed algorithms in [2], [3], [4], [5], [6], [7]). These algorithms do not consider the remaining battery energy of nodes to avoid overuse of nodes, even though some of them, namely, [4], [5], [6], [7], address energy-efficiency and reliability together. Apart from this, many routing algorithms—including energy-efficient algorithms proposed in [2], [3], [4], [5], [6], [7]—have a major drawback. They do not consider the actual energy consumption of nodes to discover energy-efficient routes. They only consider the transmission power of nodes (the output power of the power amplifier) neglecting the energy consumed by processing elements of transmitters and receivers. What is considered as energy cost of a path by these algorithms is only a fraction of the actual energy cost of nodes for transmission along a path. As we will show, this negatively affects energy-efficiency, reliability, and the operational lifetime of the network altogether.

The third category includes algorithms that try to prolong the network lifetime by finding routes consisting of nodes with a higher level of battery energy (e.g., the proposed algorithms in [8], [9], [10], [11], [12], [13], [14], [15]). These algorithms, however, do not address the other two aspects, i.e., reliability and energy-efficiency. Discovered routes by these algorithms may neither be energy-efficient nor be reliable. This can increase the overall energy consumption in the network. Thus, the network lifetime may even be reduced.

- *The authors are with the Department of Telecommunications, Delft University of Technology, 2628 CD Delft, The Netherlands. E-mail: {j.vazifehdan, r.r.venkateshaprasad, i.i.g.m.m.niemegeers}@tudelft.nl.*

Manuscript received 1 June 2011; revised 15 Feb. 2012; accepted 20 Dec. 2012; published online 10 Jan. 2013.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2011-06-0289. Digital Object Identifier no. 10.1109/TMC.2013.7.

Published by the IEEE CS, CASS, ComSoc, IES, & SPS

Our in-depth work in this paper considers energy-efficiency, reliability, and prolonging the network lifetime in wireless ad hoc networks holistically. We propose a novel energy-aware routing algorithm, called reliable minimum energy cost routing (RMECR). RMECR finds energy-efficient and reliable routes that increase the operational lifetime of the network. In the design of RMECR, we use an in-depth and detailed analytical model of the energy consumption of nodes. RMECR is proposed for networks with hop-by-hop (HBH) retransmissions providing link layer reliability, and networks with E2E retransmissions providing E2E reliability. HBH retransmission is supported by the medium access control (MAC) layer (more precisely the data link layer) to increase reliability of packet transmission over wireless links. Nevertheless, some MAC protocols such as CSMA and MACA may not support HBH retransmissions. In such a case, E2E retransmission could be used to ensure E2E reliability [4], [5], [16].

Our work has also some important and novel ideas compared to the pioneering studies like [2], [3], [4], [5], [6], [7], which also address the problem of energy-efficient reliable routing in wireless ad hoc networks. 1) We consider the impact of limited number of transmission attempts on the energy cost of routes in HBH systems. This effect has been neglected in [4], [5] and have not been addressed in-depth in [6], [7]. We show that by taking this limitation into account, a shortest-path routing algorithm like Dijkstra's algorithm—which has been considered as an optimum solution in [2], [3], [4], [5], [6], [7] for the problem of minimum energy routing in wireless ad hoc networks—does not provide an optimal solution. It is a heuristic solution, and it can be an optimal solution only if the number of retransmissions on each link is large enough to ensure complete reliability of links. 2) We consider the impact of acknowledgment packets on energy cost of routes in both HBH and E2E systems. This impact has been neglected in [2], [3], [4], [5], [6], [7]. By considering this, we show that in the E2E systems, the energy cost of packet transmission from a source node to an intermediate node depends on both upstream and downstream links of that intermediate node. Neglecting the impact of acknowledgement packets means that we disregard the impact of downstream links on the energy cost. 3) We consider energy consumption of processing elements of transceivers. As mentioned earlier, underestimating the energy consumption of transceivers can severely harm reliability and energy-efficiency of routes. A detailed consideration toward various aspects of the energy consumption of nodes makes our work realistic and thus closer to practical implementations.

Our study here is also exhaustive compared to our earlier studies in [17]. We had only presented RMECR for the HBH system in [17]. Furthermore, the impact of limited number of retransmissions on energy cost of routes, and the in-depth analysis of algorithmic aspects are missing in [17]. These aspects are covered in this paper. Apart from this, based on our detailed and generic analytical model proposed for the design of RMECR in this paper, we devise a state-of-the-art energy-efficient routing algorithm for ad hoc networks called reliable minimum energy routing (RMER). The routes discovered by RMER minimize the consumed energy of the E2E packet traversal in the network. RMER does not consider the remaining battery energy of nodes, and will be used as a benchmark to evaluate energy-efficiency of the RMECR algorithm.

Our simulation studies show that, a considerable energy-efficiency and reliability gain is achieved by the RMER algorithm compared to the energy-efficient routing algorithm proposed in [4] for the HBH system and the algorithm proposed in [5] for the E2E system.[1] On the other hand, while RMECR is not primarily an energy-efficient routing algorithm like RMER, our simulation results verify that energy-efficiency and reliability of routes discovered by RMECR are almost similar to those of RMER. Moreover, RMECR extends the operational lifetime of the network since it considers the remaining battery energy of the nodes. RMECR also outperforms routing algorithms proposed in [4], [5], [9], [15] with regard to energy-efficiency, reliability, and lifetime of the network. It finds reliable routes similar to those discovered by the algorithm proposed in [1] extending the operational lifetime of the network too.

The rest of the paper is organized as follows: In Section 2, we present required preliminaries. In Section 3, we introduce the RMECR and the RMER algorithms for the HBH system, and in Section 4, we introduce them for the E2E system. Section 5 describes some practical issues for deploying RMER and RMECR in ad hoc networks. Section 6 presents simulation studies. We conclude in Section 7.

2 Preliminaries

2.1 Network Model

We represent topology of a wireless ad hoc networks by a graph $G(\mathbb{W}, \mathbb{E})$, where $\mathbb{W}$ and $\mathbb{E}$ are the set of nodes (vertices) and links (edges), respectively. Each node is assigned a unique integer identifier between 1 and $N = |\mathbb{W}|$. Nodes are assumed to be battery powered. The remaining battery energy of node $u \in \mathbb{W}$ is represented by C_u. If the battery energy of a node falls below a threshold C_{th}, the node is considered to be dead [18]. Without loss of generality, we assume $C_{th} = 0$.

A link in the network is denoted by (u, v), in which u and v are sending and receiving nodes, respectively. The criterion for having a link from u to v is as follows: There could be a link from u to v, if the received signal strength by v is above a threshold. This threshold is usually specified in such a way that a targeted link error probability is satisfied. We denote the probability of error-free reception of packets of length x [bit] transmitted by u to v by $p_{u,v}(x)$. In other words, $p_{u,v}(x)$ is the packet delivery ratio (PDR) of (u, v) for packets of size x [bit].

As an essential requirement for energy-efficient routing, we assume nodes support adjustable transmission power. The transmission power from node u to node v is denoted by $P_{u,v}$. $P_{u,v}$ belongs to a finite set of allowable transmission powers for node u specified by $S(u) = \{P_1(u), P_2(u), \ldots, P_{m_u}(u)\}$, where m_u is the number of allowable transmission powers of node u. The discrete set is due to the practical considerations that all the commercially available devices

[1] The energy-efficient routing algorithms proposed in [4], [5] outperform similar algorithms proposed in [2], [3].

are preprogrammed with a set of power settings. Regarding the power adjustment by nodes, we assume: 1) $P_{u,v}$ is the minimum transmission power from $S(u)$ that satisfies the targeted link error probability. 2) By adjusting the transmission power, the data rate of the physical link does not change.

We represent a path in the network with h hops between two nodes as a set of nodes $\mathcal{P}(n_1, n_{h+1}) = \{n_1, n_2, \ldots, n_h, n_{h+1}\}$, where $n_k \in \mathbb{W}$ is the identifier of the kth node $(k = 1, \ldots, h+1)$ of the path. Here, n_1 is the source node, n_{h+1} is the destination node, and the rest are intermediate nodes which relay packets from the source to the destination hop by hop. Furthermore, $(n_k, n_{k+1}) \in \mathbb{E}$ is the kth link $(k = 1, \ldots, h)$ of the path.

2.2 Energy Consumption for Packet Transmission over Wireless Links

Let x bit denotes the size of a packet transmitted over the physical link and let $\varepsilon_{u,v}(x)$ [J] denote the energy consumed by a transmitting node u to transmit a packet of length x to a receiving node v through the physical link (u, v). Let $\omega_{u,v}(x)$ [J] denote the energy consumed by the receiving node v to receive and process the packet of length x transmitted by u. The energy consumed by nodes during packet transmission could be abstracted into two distinct parts [19], [20]. The first part represents the energy consumed by the transmission circuit excluding the power amplifier of the transmitter. The second part represents the energy consumed by the power amplifier to generate the required output power for data transmission over the air. On the other hand, the energy consumed by a node to receive a packet could be abstracted by only one part, which is the energy consumed by the receiving circuit including the low noise amplifier (LNA) of the receiver.

Let A_u be the power required to run the processing circuit of the transmitter of node u, $P_{u,v}$ be the transmission power from node u to node v, $0 < \kappa_u \leq 1$ be the power efficiency of the power amplifier of node u, B_v be the power required to run the receiving circuit of the wireless interface at node v, and r [bit/s] be the data rate of the physical link. We can calculate $\varepsilon_{u,v}(x)$ as

$$\varepsilon_{u,v}(x) = \left(A_u + \frac{P_{u,v}}{\kappa_u} \right) \frac{x}{r}, \quad \forall x \geq 0, \ \forall (u, v) \in \mathbb{E}, \tag{1}$$

and $\omega_{u,v}(x)$ as

$$\omega_{u,v}(x) = \frac{B_v}{r} x, \quad \forall x \geq 0, \ \forall (u, v) \in \mathbb{E}. \tag{2}$$

Note that $\varepsilon_{u,v}(x)$ and $\omega_{u,v}(x)$ are the energy consumed during a single transmission of a packet. The impact of packet retransmission will be considered later.

2.3 Hop-by-Hop and End-to-End Retransmission Systems

Wireless links in ad hoc networks are usually prone to transmission errors. This necessitates the use of retransmission schemes to ensure the reliability. We can use either HBH or E2E retransmissions. In the HBH system, a lost packet in each hop is retransmitted by the sender to ensure link level reliability. An acknowledgment (ACK) is transmitted by the receiver to the sender when the receiver receives the packet correctly. If the sender does not receive the ACK (because either the packet or its ACK is lost or corrupted), the sender retransmits the packet. This continues until the sender receives an ACK or the maximum allowed number of transmission attempts is reached. If each link is reliable, the E2E path between nodes will also be reliable.

In the E2E system, the ACKs are generated only at the destination and retransmissions happen only between the end nodes. The destination node sends an E2E ACK to the source node when it receives the packet correctly. If the source node does not receive an ACK for the sent packet, it retransmits the packet. This may happen either because the packet or the ACK is lost. In either case, the source retransmits the packet until it receives an ACK for the packet.

In both HBH and E2E systems, a retransmission occurs after the expiration of a *timer*. We assume that the duration of this timer is long enough to prevent unnecessary retransmissions. We will design energy-aware and reliable routing algorithms optimized for each of the HBH and E2E systems.

2.4 Energy-Aware Reliable Routing

Our objective is to find reliable routes which minimize the *energy cost* for E2E packet traversal. To this end, reliability and energy cost of routes must be considered in route selection. The key point is that energy cost of a route is related to its reliability. If routes are less reliable, the probability of packet retransmission increases. Thus, a larger amount of energy will be consumed per packet due to retransmissions of the packet. By defining two different ways of computing the *energy cost* of routes, we design two sets of energy-aware reliable routing algorithms for HBH and E2E systems. They are called reliable minimum energy cost routing and reliable minimum energy routing (RMER). In RMER, energy cost of a path for E2E packet traversal is *the expected amount of energy* consumed by all nodes to transfer the packet to the destination. In RMECR, the energy cost of a path is *the expected battery cost* of nodes along the path to transfer a packet from the source to the destination. Before we proceed with the design of RMER and RMECR, we first define the *minimum energy cost path*.

Definition 1 (Minimum Energy Cost Path). *The minimum energy cost path (MECP) between a source and a destination node is a path which minimizes the expected energy cost for E2E traversal of a packet between the two nodes in a multihop network.*

3 Energy-Aware Reliable Routing Algorithms for the HBH System

This section presents design of RMER and RMECR algorithms for networks supporting HBH retransmissions. To this end, first, in Section 3.1, we analyze the energy cost of a path for transferring a packet to its destination. Considering the impact of limited retransmissions across each link, the size of data and ACK packets, and the reliability of E2E paths is the added value of our analysis, which distinguishes our work from [4], [5], [6], [7]. Based on this in-depth analysis, in Section 3.2, we design a generic routing algorithm for finding MECP between every two nodes of the network. By defining appropriate link weights, in Section 3.3 RMER and RMECR algorithms are derived as two flavors of this generic routing algorithm.

3.1 Analysis of Energy Cost of a Path

The energy cost of a path is analyzed in four steps:

1. analyzing the expected transmission count of data and ACK packets,
2. analyzing the expected energy cost of a link taking into account the energy cost of retransmissions,
3. analyzing the E2E reliability of a path, ~~and~~
4. formulating the energy cost of a path taking into account the energy cost of links and E2E reliability of the path. This in-depth analysis of the energy cost lays the foundation for designing RMER and RMECR algorithms for the HBH System.

3.1.1 Expected Transmission Count of Data and ACK Packets

We assume that a node u is allowed to transmit a packet only Q_u times (including the first transmission). Thus, due to probabilistic nature of packet loss over wireless links, a packet might be retransmitted a random number of times not greater than $Q_u - 1$. When the receiving node v receives the packet correctly, an ACK is sent to the transmitting node u. If the transmitted ACK is lost, another ACK will be transmitted for the same packet after v again receives the packet correctly (possibly after several attempts). Therefore, an ACK could be transmitted for the same data packet a random number of times not greater than Q_u. It is also possible that no ACK is transmitted for a data packet, if the packet is lost in all Q_u transmission attempts. We assume $E[n_{u,v}(L_d)]$ is the expected number of times that u needs to transmit a packet of length L_d [bit] to deliver it to v (including the first transmission), where $1 \leq n_{u,v}(L_d) \leq Q_u$ is the exact value. Furthermore, we assume $E[m_{v,u}(L_h)]$ is the expected number of ACKs of length L_h [bit] sent by v for the data packet to u, where $0 \leq m_{v,u}(L_h) \leq Q_u$ is the exact value. Note that L_h and L_d are known constant values.

Values of $E[n_{u,v}(L_d)]$ and $E[m_{v,u}(L_h)]$ depend on the quality of forward link (u, v) and reverse link (v, u). The lower the quality of links, the higher is the expected number of retransmissions. We have shown in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2013.7

$$E\big[n_{u,v}(L_d)\big] = \frac{1 - (1 - p_{u,v}(L_d)p_{v,u}(L_h))^{Q_u}}{p_{u,v}(L_d)p_{v,u}(L_h)}. \tag{3}$$

We also have derived an expression for $\Pr\{m_{v,u}(L_h)\} = i$, $i = 0, 1, \ldots, Q_u$ in Appendix A, available in the online supplemental material, as

$$\begin{aligned}&\Pr\{m_{v,u}(L_h) = i\}\\ &= \begin{cases} (1-p)^{Q_u},\ i = 0;\\ p^{Q_u}(1-q)^{Q_u - 1},\ i = Q_u;\\ \sum_{j=i}^{Q_u-1} \binom{j-1}{i-1} p^{i-1}(1-q)^{i-1}(1-p)^{j-i}pq\\ +\binom{Q_u-1}{i-1} p^{i-1}(1-p)^{Q_u-i}(1-q)^{i-1}p\\ +\binom{Q_u-1}{i} p^{i}(1-p)^{Q_u-1-i}(1-q)^{i}(1-p)\\ \qquad \forall i = 1..Q_u - 1,\end{cases}\end{aligned} \tag{4}$$

in which for ease of representation, we have defined $p = p_{u,v}(L_d)$ and $q = p_{v,u}(L_h)$. Using (4), $E[m_{v,u}(L_h)]$ could be calculated as

$$E[m_{v,u}(L_h)] = \sum_{i=0}^{Q_u} i \Pr\{m_{v,u}(L_h) = i\}. \tag{5}$$

We can easily show that if $Q_u \to \infty$, then

$$\begin{cases} E[n_{u,v}(L_d)] \to \frac{1}{p_{u,v}(L_d)p_{v,u}(L_h)},\\ E[m_{v,u}(L_h)] \to \frac{1}{p_{v,u}(L_h)}. \end{cases} \tag{6}$$

3.1.2 Total Energy Consumption across a Link

Let $a_{u,v}(L_d)$ be the total energy consumed by the transmitting node u, and $b_{u,v}(L_d)$ be the total energy consumed by the receiving node v to exchange a packet of length L_d [bit]. Taking into account the impact of HBH retransmissions, we have

$$a_{u,v}(L_d) = E[n_{u,v}(L_d)]\varepsilon_{u,v}(L_d) + E[m_{v,u}(L_h)]\omega_{u,v}(L_h), \tag{7}$$

where $\varepsilon_{u,v}(L_d)$ is the energy consumed by u during a single transmission of the packet, which is computed using (1). Parameter $\omega_{u,v}(L_h)$ is the energy consumed by u during a single reception of the ACK, which is computed using (2). The total energy consumed by the receiving node v is computed as

$$b_{u,v}(L_d) = E[n_{u,v}(L_d)]\omega_{v,u}(L_d) + E[m_{v,u}(L_h)]\varepsilon_{v,u}(L_h), \tag{8}$$

where $\varepsilon_{v,u}(L_h)$ is the energy consumed by v during a single transmission of the ACK, and $\omega_{v,u}(L_d)$ is the energy consumed by v during a single reception of the data packet.

3.1.3 Link and Path Reliability

We define $R_{u,v}(L_d)$ as the reliability of (u, v) for packets of size L_d bits. The reliability of a link is the probability that a packet is successfully delivered to the receiving node within the number of allowed transmissions. In the HBH system, link reliability is related to the PDR of the link as

$$\begin{aligned} R_{u,v}(L_d) &= 1 - \Pr\{\text{packet lost after } Q_u \text{ transmissions}\}\\ &= 1 - \big[1 - p_{u,v}(L_d)\big]^{Q_u}. \end{aligned} \tag{9}$$

Here, we should notice that the reliability of a link is not affected by the probability of losing the ACK. If the packet is received correctly but its ACK is lost, the packet will be retransmitted after expiration of a timer.[2] If the retransmitted packet is received correctly too, there will be a duplicate packet at the receiver. Duplicate packets are usually discarded silently at the MAC layer, but ACKs are sent for them. This, however, affects the energy consumption of the transmitting and the receiving nodes, which was considered in computing their energy costs in $a_{u,v}(L_d)$ and $b_{u,v}(L_d)$ in (7) and (8).

Considering a path $\mathcal{P}(n_1, n_{h+1}) = \{n_1, n_2, \ldots, n_h, n_{h+1}\}$, we define $R_{n_i}(L_d)$ the E2E reliability of the path from the source node n_1 up to n_i as

[2] This scheme is usually deployed by MAC protocols, for example, IEEE 802.11 MAC.

$$R_{n_i}(L_d) = \begin{cases} 1, & i = 1, \\ \prod_{k=1}^{i-1} R_{n_k,n_{k+1}}(L_d), & i = 2, \dots, h+1, \end{cases} \tag{10}$$

in which $R_{n_k,n_{k+1}}(L_d)$ is the reliability of the link (n_k, n_{k+1}) for a packet of length L_d bit.

3.1.4 Expected Energy Cost of a Path

Let $e_{u,v}(L_d)$ denote the expected energy cost for forwarding a packet of length L_d [bit] over (u, v). Let $\mathcal{C}(\mathcal{P}(n_1, n_{h+1}))$ be the expected energy cost to route a data packet along the path $\mathcal{P} = \{n_1, n_2, \dots, n_h, n_{h+1}\}$ from the source node n_1 to the destination node n_{h+1}. Since the number of retransmissions in each link could be a limited value, a packet may not be delivered to the receiver in one of the hops. Obviously, the remaining nodes in the route after that hop will not consume energy to forward the packet. Thus, in the HBH system, the expected energy cost of a route is computed as

$$\mathcal{C}(\mathcal{P}(n_1, n_{h+1})) = \sum_{i=1}^{h} \left[R_{n_i}(L_d) e_{n_i,n_{i+1}}(L_d) \right], \tag{11}$$

in which $e_{n_i,n_{i+1}}(L_d)$ is the energy cost of $(n_i, n_{i+1}) \in \mathcal{P}$ for packets of length L_d.

3.2 Design of a Routing Algorithm for Finding MECP

Here, we design a generic routing algorithm for finding MECP between every two nodes in the network. Since energy cost is an additive metric, it ~~seems~~ that the Dijkstra's shortest path routing algorithm could be used to find MECP in the HBH system. However, we show that the Dijkstra's shortest path routing algorithm is only a heuristic solution for finding MECP, but under some circumstances it could be the optimal solution.

According to the Dijkstra's algorithm, the cost of a path from s to v is calculated in a recursive way as

$$\mathcal{C}(\mathcal{P}(s, v)) = \mathcal{C}(\mathcal{P}(s, u)) + W(u, v), \tag{12}$$

where u precedes v in $\mathcal{P}(s, v)$, and $W(u, v)$ is the weight of (u, v). To find out whether the Dijkstra's shortest path routing algorithm could be used to find MECP in the HBH system, we inspect the expression derived in (11) for the expected energy cost in this system. This expression could be rewritten as

$$\begin{aligned} \mathcal{C}(\mathcal{P}(n_1, n_{h+1})) &= \left(\sum_{i=1}^{h-1} \left[R_{n_i}(L_d) e_{n_i,n_{i+1}}(L_d) \right] \right) \\ &\quad + R_{n_h}(L_d) e_{n_h,n_{h+1}}(L_d) \\ &= \mathcal{C}(\mathcal{P}(n_1, n_h)) + \frac{R_{n_{h+1}}(L_d)}{R_{n_h,n_{h+1}}} e_{n_h,n_{h+1}}(L_d). \end{aligned} \tag{13}$$

From (13), it could be concluded that the expected energy cost for transferring a packet of length L_d from a source node s to any destination node v can be computed in a recursive way as

$$\begin{cases} \mathcal{C}(\mathcal{P}(s, v)) = \mathcal{C}(\mathcal{P}(s, u)) + R(\mathcal{P}(s, v), L_d) \times \frac{e_{u,v}(L_d)}{R_{u,v}(L_d)} \\ R(\mathcal{P}(s, v), L_d) = R(\mathcal{P}(s, u), L_d) \times R_{u,v}(L_d). \end{cases} \tag{14}$$

In (14), $R(\mathcal{P}(s, v), L_d)$ is the E2E reliability of $\mathcal{P}(s, v)$ for data packets of length L_d.

If we compare the recursive equation in (14) with (12), it is clear that, to be able to use the Dijkstra's algorithm to find MECP in the HBH system, we need to define the link weight $W(u, v)$ as

$$W(u, v) = R(\mathcal{P}(s, v), L_d) \times \frac{e_{u,v}(L_d)}{R_{u,v}(L_d)}. \tag{15}$$

Nevertheless, (15) suggests that the weight of each link depends on the reliability of upstream links from the source node to that link. As a result, the same link may have different weights in different paths between the same source and destination nodes. Dijkstra's algorithm works on the basis of the fact that weight of a link does not depend on the weight of the upstream links between the source node and that link. This fact forms the basis for the proof of correctness of the Dijkstra's algorithm [21].

Theorem 1. *In the Dijkstra's algorithm (see Algorithm 1 in Appendix B, available in the online supplemental material), when a node v is extracted from $\mathcal{Q}$, links in $\mathcal{T}(v)$ form the shortest path from s to v.*

We can show that if the link weight is defined as (15), the Dijkstra's algorithm cannot be used to find MECP in the HBH system.

Lemma 1. *If the link weight in the Dijkstra's algorithm is defined as (15), then Theorem 1 is not valid anymore.*

Proof. Let the path formed by links in $\mathcal{T}(i)\ \forall i \in \mathbb{W}$, be denoted by $\mathcal{P}^*(s, i)$. Suppose that for every node z extracted so far from $\mathcal{Q}$, $\mathcal{P}^*(s, z)$ is the shortest path from s to z. If v is the next node extracted from $\mathcal{Q}$, we can show that there might be another path $\mathcal{P}_1(s, v)$ whose weight is smaller than the weight of $\mathcal{P}^*(s, v)$. Suppose that $\mathcal{P}_1(s, v) = \mathcal{P}_1(s, u) \cup \{v\}$, where $\mathcal{P}_1(s, u)$ is a path between s and u, and u is the node extracted from $\mathcal{Q}$ before v. Since we assumed that $\mathcal{P}^*(s, u)$ is the shortest path from s to u, we have

$$\mathcal{C}(\mathcal{P}^*(s, u)) \leq \mathcal{C}(\mathcal{P}_1(s, u)). \tag{16}$$

If we want $\mathcal{P}^*(s, v)$ to be the shortest path from s to v, then we must have

$$\mathcal{C}(\mathcal{P}^*(s, v)) \leq \mathcal{C}(\mathcal{P}_1(s, v)).$$

Since u has been extracted just before v from $\mathcal{Q}$, then $\mathcal{P}^*(s, v) = \mathcal{P}^*(s, u) \cup \{v\}$. Thus, according to (15), we must have

$$\begin{aligned} &\mathcal{C}(\mathcal{P}^*(s, u)) + R(\mathcal{P}^*(s, v), L_d) \times \frac{e_{u,v}(L_d)}{R_{u,v}(L_d)} \\ &\quad \leq \mathcal{C}(\mathcal{P}_1(s, u)) + R(\mathcal{P}_1(s, v), L_d) \times \frac{e_{u,v}(L_d)}{R_{u,v}(L_d)}. \end{aligned} \tag{17}$$

From (17) and (16), we can conclude that to have $C(\mathcal{P}^*(s, v)) \leq C(\mathcal{P}_1(s, v))$, we must have

$$R(\mathcal{P}^*(s, v), L_d) \leq R(\mathcal{P}_1(s, v), L_d). \tag{18}$$

However, since we did not make any assumption about the reliability of routes with respect to each other, there is no guarantee that the reliability of $\mathcal{P}^*(s, v)$ is smaller than that of $\mathcal{P}_1(s,v)$. It may happen that $R(\mathcal{P}^*(s,v),L_d) > R(\mathcal{P}_1(s,v),L_d)$. In such a case $\mathcal{P}^*(s,v)$ may not be the shortest path from s to v. □

Looking at the proof of Lemma 1, it is clear that the dependency of the link weights to the *E2E reliability of the path* is the reason for not being able to use the Dijkstra's algorithm to find MECP in the HBH system. To be able to find MECP with a complexity as low as that of the Dijkstra's algorithm, we need to remove this dependency and simplify the energy cost of a path as,[3]

$$\mathcal{C}(\mathcal{P}(n_1,n_{h+1})) = \sum_{i=1}^{h} e_{n_i,n_{i+1}}(L_d). \tag{19}$$

The energy cost function $\mathcal{C}(\mathcal{P}(n_1,n_{h+1}))$ in (19) is in fact the energy cost of all nodes along the path to *successfully* transfer a packet from the source to the destination. Thus, the Dijkstra's algorithm could be used as a heuristic solution to find MECP in the HBH system provided that the link weight $W(u,v)$ is defined as

$$W(u,v) = e_{u,v}(L_d). \tag{20}$$

We note that if each link is completely reliable (i.e., its reliability is 1), then (11) naturally reduces to (19). In other words, if each link is completely reliable, then the Dijkstra's algorithm is the exact solution for finding MECP in the HBH system. Theoretically, each link in the HBH system is reliable, if there is no limitation on the number of possible retransmissions of a packet over a link (i.e., $Q_u \to \infty$, $\forall u \in \mathbb{W}$). Practically, a large value for Q_u might also make links reliable. Therefore, from the point of view of energy-efficient routing, a larger value for the number of retransmissions ensures that a low complexity algorithm like Dijkstra's algorithm provides an exact solution for finding MECP in the HBH system. How large the number of retransmission attempts should be is an issue that depends on the typical values of quality of links in the environment where the network is deployed.

3.3 Link Weight in RMECR and RMER Algorithms

Now that we know how the link weight should look like to be able to use Dijkstra's algorithm for finding MECP, we formulate the link weights for the RMER and RMECR algorithm in the HBH system. We start with the RMECR algorithm. We mentioned in Section 2.4 that RMECR defines the energy cost of a path as the expected battery cost of nodes along the path for forwarding the packet. For RMECR, we define the battery cost of a link as

> *"the fraction of the residual battery energy of the two nodes of the link which is consumed to forward the packet."*

To formulate the link weight in RMCER, let C_u be the remaining battery energy of u and C_v be the remaining battery energy of v. As introduced in Section 3.1.2, the energy consumed by u to deliver a packet to v is defined by $a_{u,v}(L_d)$, and the energy consumed by v for receiving the packet is defined by $b_{u,v}(L_d)$. Considering the definition of the battery cost of a link in RMECR, the link weight in this algorithm is obtained as

$$\begin{aligned} W(u,v) &= e_{u,v}(L_d) \\ &= \frac{a_{u,v}(L_d)}{C_u} + \frac{b_{u,v}(L_d)}{C_v} \\ &= \frac{L_d}{r} E[n_{u,v}(L_d)] \left(\frac{A_u + \frac{P_{u,v}}{\kappa_u}}{C_u} + \frac{B_v}{C_v} \right) \\ &\quad + \frac{L_h}{r} E[m_{v,u}(L_h)] \left(\frac{A_v + \frac{P_{v,u}}{\kappa_v}}{C_v} + \frac{B_u}{C_u} \right). \end{aligned} \tag{21}$$

The link weight in RMECR captures the impact of the quality of links, the energy consumption parameters of nodes, and the remaining battery energy of nodes. As we will verify in Section 6, this allows RMECR to find energy-efficient and reliable routes that increase the operational lifetime of the network.

The general approach used to design RMECR allows us to easily define other variants of energy-aware routing algorithms by defining other formulations for energy cost of links. For instance, if we assume that the energy cost of a link is just the total amount of energy consumed by the transmitting and the receiving nodes to exchange a packet, we can devise an energy-efficient routing algorithm. That is, if we define the energy cost associated to a link (u,v) as

$$e_{u,v}(L_d) = a_{u,v}(L_d) + b_{u,v}(L_d),$$

then the link weight $W(u,v)$ is obtained as

$$\begin{aligned} W(u,v) &= e_{u,v}(L_d) \\ &= \frac{L_d}{r} E[n_{u,v}(L_d)] \left(A_u + \frac{P_{u,v}}{\kappa_u} + B_v \right) \\ &\quad + \frac{L_h}{r} E[m_{v,u}(L_h)] \left(A_v + \frac{P_{v,u}}{\kappa_v} + B_u \right). \end{aligned} \tag{22}$$

We name the resulting algorithm RMER, which is an energy-efficient routing algorithm minimizing the total amount of energy consumed to route a packet from a source node to a destination node. Compared to the RMECR, RMER does not consider the remaining battery energy of nodes. We will use RMER as a benchmark algorithm to study the energy-efficiency of RMECR. The RMECR and the RMER algorithms for the HBH system have been summarized in Algorithm 2 in Appendix B, available in the online supplemental material. We will show in Section 6 that RMER and RMECR outperform existing energy-efficient routing algorithms in terms of both energy-efficiency and reliability. This is while RMECR also extends the operational lifetime of the network.

4 Energy-Aware Reliable Routing Algorithms for the E2E System

This section presents design of RMER and RMECR algorithms for networks supporting E2E retransmissions. Similar to the HBH system, we first analyze the energy cost of a path for transferring a packet to its destination (Section 4.1). Here, we also consider the impact of E2E ACKs, which is missing in [4], [5], [6], [7]. Then in Section 4.2, a generic routing algorithm is designed for finding MECP in the E2E system. In Section 4.3, RMER and RMECR algorithms are derived for the E2E system.

[3] Note that the energy cost is still dependent on the reliability of individual links, through expected retransmission count of data and ACK packets. Only the impact of E2E reliability of the path is neglected.

4.1 Analysis of Energy Cost of a Path

In the E2E system, the energy cost of a path depends on the number of times that the packet and its E2E ACK are transmitted. This, in turn, depends on the E2E reliability of the path. To determine the energy cost, we start with formulating the E2E reliability of the path for data packets and E2E ACKs. Then, the expected energy cost is calculated.

4.1.1 Link and Path Reliability

Parameter $R_{n_i}(L_d)$ is defined to be the E2E reliability of the path $\mathcal{P} = \{n_1, n_2, \ldots, n_h, n_{h+1}\}$ for data packets of length L_d [bit] from source node n_1 up to node n_i in the path, $i = 1, \ldots, h+1$. In the E2E system, $R_{n_i}(L_d)$ is calculated as

$$R_{n_i}(L_d) = \begin{cases} 1, & i = 1 \\ \prod_{k=1}^{i-1} p_{n_k,n_{k+1}}(L_d), & i = 2, \ldots, h+1. \end{cases} \tag{23}$$

A data packet may be sent again by the source node if the source node does not receive an ACK from the destination. It is obvious that there must be a reverse path from the destination to the source node to carry the E2E ACK. We assume this reverse path consists of the same links as that of the forward path, but in the opposite direction. Parameter $R^{'}_{n_i}(L_e)$ is defined to be the E2E reliability of the reverse path $\mathcal{P}^{'} = \{n_{h+1}, \ldots, n_1\}$ from the destination node n_{h+1} up to node n_i, which is computed as

$$R^{'}_{n_i}(L_e) = \begin{cases} \prod_{k=i}^{h} p_{n_{k+1},n_k}(L_e), & i = 1, \ldots, h \\ 1, & i = h+1. \end{cases} \tag{24}$$

4.1.2 Expected Transmission Count of Data and ACK Packets

The expected number of times that a data packet of length L_d is transmitted from the source to the destination (including the first transmission) is denoted by $N_{\mathcal{P}}(L_d)$. Furthermore, we denote $M_{\mathcal{P}}(L_e)$ as the expected number of times that an E2E ACK of length L_e [bit] is transmitted for the data packet by the destination node. Since E2E retransmissions are to ensure E2E reliability, we assume that the number of E2E retransmissions is large enough. With this assumption, we can calculate $N_{\mathcal{P}}(L_d)$ and $M_{\mathcal{P}}(L_d)$ as

$$\begin{cases} N_{\mathcal{P}}(L_d) = \frac{1}{R_{n_{h+1}}(L_d) R^{'}_{n_1}(L_e)} \\ M_{\mathcal{P}}(L_e) = \frac{1}{R^{'}_{n_1}(L_e)}. \end{cases}$$

Note that even if there is a limitation on the number of E2E retransmissions, $N_{\mathcal{P}}(L_d)$ and $M_{\mathcal{P}}(L_e)$ could be calculated similar to $E[n_{u,v}(L_d)]$ and $E[m_{v,u}(L_h)]$ in (3) and (5), respectively. The only difference is that PDR of forward and reverse links in (3) and (5) must be replaced by the E2E reliability of forward and reverse paths.

4.1.3 Expected Energy Cost of a Path

In the E2E system, the expected energy cost of path for transferring a data packet from the source node to the destination is the expected energy cost during a single transmission from the source to the destination multiplied by the expected number of times that the source transmits the packet (including the first transmission) [16]. We should notice that a packet could be lost while it is being transferred from the source to the destination. This affects the expected energy cost in a single transmission of the packet. Therefore, in the E2E system we have

$$\begin{aligned} \mathcal{C}(\mathcal{P}(n_1, n_{h+1})) = N_{\mathcal{P}}(L_d) \sum_{i=1}^{h} \left[R_{n_i}(L_d) e_{n_i,n_{i+1}}(L_d) \right] \\ + M_{\mathcal{P}}(L_e) \sum_{i=1}^{h} \left[R^{\prime}_{n_{i+1}}(L_e) e_{n_{i+1},n_i}(L_e) \right], \end{aligned} \tag{25}$$

where $e_{u,v}(L)$, $L \in \{L_d, L_e\}$, $\forall (u,v) \in \mathbb{E}$, is the energy cost of packet transmission over a link in the E2E system. We will formulate $e_{u,v}(L)$ for RMER and RMECR algorithms in the E2E system in Section 4.3.

4.2 Design of a Routing Algorithm for Finding MECP

Here, again the question is whether we can use the Dijkstra's algorithm to find MECP in the E2E system. To answer this question, we inspect the energy cost of a path in the E2E system, expressed in (25), to see if we can find a recursive form for it similar to (12). As shown in Appendix C, available in the online supplemental material, if we replace $R_{n_i}(L_d)$ from (23) and $R^{'}_{n_{i+1}}$ from (24) into (25), we can calculate the expected energy cost for transferring a packet of length L_d from a source node s to any destination node v in a recursive way as

$$\begin{cases} \mathcal{C}((\mathcal{P}(s,v)) = \mathcal{C}_1(\mathcal{P}(s,v)) + \mathcal{C}_2(\mathcal{P}(s,v)) \\ \mathcal{C}_1(\mathcal{P}(s,v)) = \frac{1}{p_{u,v}(L_d) p_{v,u}(L_e)} \left(\mathcal{C}_1(\mathcal{P}(s,u)) + \frac{e_{u,v}(L_d)}{R(\mathcal{P}^{'}(u,s),L_e)} \right) \\ \mathcal{C}_2(\mathcal{P}(s,v)) = \mathcal{C}_2(\mathcal{P}(s,u)) + \frac{e_{u,v}(L_e)}{R(\mathcal{P}^{'}(v,s),L_e)} \\ R(\mathcal{P}^{'}(v,s), L_e) = p_{v,u}(L_e) \times R(\mathcal{P}^{'}(u,s), L_e), \end{cases} \tag{26}$$

where $R(\mathcal{P}^{'}(u,s), L_e)$ and $R(\mathcal{P}^{'}(v,s), L_e)$ are respectively the reliability of reverse paths $\mathcal{P}^{'}(u,s)$ and $\mathcal{P}^{'}(v,s)$ for E2E ACKs.

It is clear that (26) is not characteristically similar to (12). If we use the recursive (26) in Dijkstra's algorithm, using similar approach used to prove Lemma 1, we can show that we cannot find MECP using the Dijkstra's algorithm. Here, the problem is not only the dependency of the path cost to the E2E reliability of the forward and reverse paths, but also the dependency of the path cost to the energy cost of the reverse path for transferring ACKs (referred to as *downstream-links dependency*). In this paper, we propose two heuristic solutions for the E2E system to find MECP using a modified version of Dijkstra's algorithm. To be able to refer them in the sequel, we name them H_1 and H_2.

4.2.1 Heuristic Solution H_1

Let us neglect the effect of the E2E ACK on the expected energy cost. With this assumption, (25) reduces to

$$\begin{aligned} \mathcal{C}(\mathcal{P}(n_1, n_{h+1})) &= \frac{1}{R_{n_{h+1}}} \sum_{i=1}^{h} R_{n_i} e_{n_i,n_{i+1}}(L_d) \\ &= \sum_{i=1}^{h} \frac{e_{n_i,n_{i+1}}(L_d)}{\prod_{j=i}^{h} p_{n_j,n_{j+1}}(L_d)} \\ &= \frac{1}{p_{n_h,n_{h+1}}(L_d)} \left[C(\mathcal{P}(n_1, n_h)) + e_{n_h,n_{h+1}}(L_d) \right]. \end{aligned} \tag{27}$$

On the basis of (27), we can extract the following recursive expression for computing the energy cost of a path from a source node s to any destination node v,

$$\begin{cases} C(\mathcal{P}(s,v)) = \frac{1}{p_{u,v}(L_d)}\left[C(\mathcal{P}(s,u)) + W(u,v)\right] \\ W(u,v) = e_{u,v}(L_d), \end{cases} \tag{28}$$

where $e_{u,v}(L_d)$ is the energy cost of a link in the E2E system. If we compare the recursive equation in (28) with (12), we realize that by considering $R_{u,v}(L_d) = 1$, (28) simplifies to (12). Equation (28) lays the foundation for devising a generalized version of the Dijkstra's algorithm to find MECP in the E2E system. We only need to use the recursive equation in (28) instead of (12) in the Dijkstra's algorithm.

4.2.2 Heuristic Solution H_2

H_1 neglects the effect of the E2E ACK on the energy cost of a path. If the size of the E2E ACK is not small compared to the size of the data packet, this assumption may not be a valid assumption. H_2, on the other hand, considers the impact of the E2E ACK on the energy cost. In H_2, the energy cost of packet transfer from a source node s to any destination node v is calculated in a recursive way as

$$\begin{cases} C(\mathcal{P}(s,v)) = \frac{1}{p_{u,v}(L_d)p_{v,u}(L_e)}\left[C(\mathcal{P}(s,u)) + W(u,v)\right] \\ W(u,v) = e_{u,v}(L_d) + e_{v,u}(L_e), \end{cases} \tag{29}$$

where $e_{u,v}(L_d)$ is the energy cost for a single transmission of a data packet over (u,v), and $e_{v,u}(L_e)$ is the energy cost for a single transmission of the E2E ACK over (u,v).

The main characteristics of H_2 is that it considers the reliability of links in the reverse path as well as the energy consumed to forward the E2E ACK along the reverse path. As (29) suggests, H_2 resolves the problem of downstream-node dependency that (26) suffers from, because in H_2 the energy cost from the source node to an intermediate node, only depends on the reliability of links in the forward and reverse paths till that intermediate node (i.e., only upstream links). On the basis of (29), we can design another generalized version of the Dijkstra's shortest-path routing algorithm for finding MECP for the E2E system. To this end, we need to use the recursive equation in (29) instead of (12) in the Dijkstra's algorithm.

4.3 Link Weight in RMECR and RMER Algorithms

Similar to the HBH system, in the E2E system, the energy cost of a link $e_{u,v}(L_d)$ in RMECR is defined as the fraction of the remaining battery energy of the two end nodes consumed to forward a packet across a link. In the E2E system, the energy consumed by the transmitting node to forward a packet of length L_d is $\varepsilon_{u,v}(L_d)$, which is defined by (1). The energy consumed by the receiving node is $\omega_{u,v}(L_d)$, which is defined by (2).

For the heuristic solution H_1, the link weight in RMECR is obtained as

$$\begin{aligned} W(u,v) &= e_{u,v}(L_d) \\ &= \frac{\varepsilon_{u,v}(L_d)}{C_u} + \frac{\omega_{u,v}(L_d)}{C_v} \\ &= \frac{L_d}{r}\left(\frac{A_u + \frac{P_{u,v}}{\kappa_u}}{C_u} + \frac{B_v}{C_v}\right). \end{aligned} \tag{30}$$

For heuristic solution H_2, the link weight in RMECR is obtained as

$$\begin{aligned} W(u,v) &= e_{u,v}(L_d) + e_{v,u}(L_e) \\ &= \frac{L_d}{r}\left(\frac{A_u + \frac{P_{u,v}}{\kappa_u}}{C_u} + \frac{B_v}{C_v}\right) + \frac{L_e}{r}\left(\frac{A_v + \frac{P_{v,u}}{\kappa_v}}{C_v} + \frac{B_u}{C_u}\right). \end{aligned} \tag{31}$$

As (30) and (31) show, the link weights defined for the RMECR algorithm capture the impact of remaining battery energy and the energy consumption characteristics of nodes. Note that although the reliability of links is not captured in computing the link weight in (30) and (31), RMECR still considers the reliability of links in computing the total energy cost (see (28) and (29)).

On the basis of the general approach used in the design of RMECR for the E2E system, we can devise an energy-efficient routing algorithm for the E2E system by defining the energy cost of a link as the actual amount of energy consumed by the two end nodes of the link to exchange the packet. In this way, we introduce the RMER algorithm for the E2E system. For the heuristic solution H_1, the link weight for RMER is obtained as

$$W(u,v) = \frac{L_d}{r}\left(A_u + B_v + \frac{P_{u,v}}{\kappa_u}\right). \tag{32}$$

For the heuristic solution H_2, the link weight for RMER is obtained as

$$W(u,v) = \frac{L_d}{r}\left(A_u + B_v + \frac{P_{u,v}}{\kappa_u}\right) + \frac{L_e}{r}\left(A_v + B_u + \frac{P_{v,u}}{\kappa_v}\right).$$

Algorithm 3 in Appendix B, available in the online supplemental material, summarizes RMER and RMECR algorithms for the E2E system.

5 Practical Issues

In this section, we describe some practical issues to provide an insight about implementation of RMECR and RMER algorithms. These two algorithms describe the procedure that each node should undertake to find MECP, for which they require each node to have a complete image of the network topology. In ad hoc networks, this could be achieved using a link state proactive routing protocol such as optimized link state routing (OLSR) [22]. In OLSR, each node periodically shares its view of the network topology with other nodes. This is done by the use of so-called topology control messages, which are flooded in the network. Nodes also use periodic beacons to detect their neighboring nodes.

As we observed, RMECR and RMER were designed based on the Dijkstra's algorithm. The Dijkstra's algorithm is a centralized algorithm for finding the shortest-path between nodes. We could also design RMECR and RMER algorithms based on the Bellman-Ford algorithm. Bellman-Ford algorithm could be implemented in a distributed way using a distance-vector routing protocol. However, the Bellman-Ford algorithm has a higher computational complexity than the Dijkstra's algorithm. To be able to use the Bellman-Ford algorithm to find MECP, we only need to calculate link weight and route cost in this algorithm according to link weight and energy-cost of routes defined for RMECR or RMER algorithms.

For implementation of RMECR and RMER, PDR of a link, i.e., $p_{u,v}(x)$ must be known to compute the energy cost of that link. Depending on whether HBH ACK or E2E ACK is supported, $p_{u,v}(L_d)$ and either $p_{u,v}(L_h)$ or $p_{u,v}(L_e)$ must be known to be able to calculate $n_{u,v}(L_d)$ as well as $n_{u,v}(L_e)$ and $m_{u,v}(L_h)$. In wireless ad hoc networks, PDR of a link could be estimated using a link quality estimation technique. It could be a packet-based technique in which periodic beacons [1], periodic unicast packets [23], or data traffic [24] are used to estimate the packet delivery ratio of links. Alternatively, we can use SNR-based techniques [25], [26] which uses the received signal strength to determine the expected PDR of links using SNR-to-PDR profile mapping.

Nevertheless, we should notice that the estimated PDR is for a specific packet size (e.g., size of broadcast beacons or unicast packets). In general, data packets, HBH ACKs, and E2E ACKs might have different sizes. Thus, their delivery ratio might also be different. Apart from this, link weights in RMER and RMECR algorithms directly depend on the packet size L_d (see for example (21) and (22)). It is important that the routing metric used by a routing algorithm should be independent of the packet size, because packets with different sizes might be routed through the same path. These dependencies could be resolved as follows.

Suppose L_{est} is the size of packets used by the link quality estimation method to estimate PDR of links. To have accurate estimation L_{est} should be close to the size of typical data packets transmitted between nodes. This, in turn, depends on the network application. Let $p_{u,v}(L_{est})$ be the estimated PDR of the link. We can obtain an estimation of energy cost of the link for RMER and RMECR algorithms, if we set $p_{u,v}(L_d) = p_{u,v}(L_{est})$, $p_{u,v}(L_e) = p_{u,v}(L_{est})$ and $p_{u,v}(L_h) = p_{u,v}(L_{est})$.

To resolve the direct dependency of link weights to the packet size, we can replace L_d with L_{est} when we calculate the energy cost of a link. Note that in the HBH system, L_h could take its exact value which is known a priori for the wireless technology being used. For instance, $L_h = 240$ bits in IEEE 802.11 standard and $L_h = 64$ bits in IEEE 802.15.4 standard. In the E2E system, the size of the E2E ACK, L_e, is also a fixed value, which depends on the networking protocol used. For instance, if we use IEEE 802.11 standard and TCP/IPv4 suite on top, and the E2E ACK is assumed to have the same format as that of a TCP acknowledgment, then $L_e = 768$ bits.[4] Values of L_{est}, L_e, and L_h could be stored as configurable parameters of the routing protocol at each node.

Another important issue is to determine the minimum power required by nodes for reliable packet transmission to each of their neighboring nodes, i.e., $P_{u,v}$. To determine $P_{u,v}$, we can use the proposed algorithm in [27]. In this algorithm, each node sends a number of packets (e.g., a number of beacons) to its neighboring nodes to measure the minimum transmission power required for reliable packet transmission to each of them. The neighboring nodes send back the measured values to the transmitting node. That is, the value of $P_{u,v}$ is measured by v and is sent back to u.

6 Simulation Results

6.1 Simulation Setup

To evaluate the performance of RMECR and RMER algorithms, we consider a network in which nodes are uniformly distributed in a square area. Nodes are assumed to be static. In our simulations, we compute the probability of error-free reception of packets of size x bit over a link as

$$p_{u,v}(x) = (1 - p_c)(1 - \lambda_{u,v})^x,$$

where $\lambda_{u,v}$ is the bit error rate of link (u, v) and p_c is the collision probability. To generate different error probabilities for different links, we choose p_c for each link randomly from the interval $[0, p_{c_{max}}]$. By changing the value of $p_{c_{max}}$ in each experiment, we are able to control the average quality of links in the network and also introduce the varying nature of quality of various links.

To calculate $\lambda_{u,v}$, we assume the wireless channel is Rayleigh, and nodes use DBPSK modulation.[5] The bit error rate of DBPSK over a Rayleigh fading channel is [28]

$$\lambda_{u,v} = \frac{1}{2(1 + \bar{\gamma}_{u,v})},$$

where $\bar{\gamma}_{u,v}$ is the average received signal to interference and noise ratio (SINR) per bit. $\bar{\gamma}_{u,v}$ is related to the transmission power of a node $P_{u,v}$ as

$$\bar{\gamma}_{u,v} = \frac{g_1 P_{u,v}}{d_{u,v}^{\eta}} \times \frac{1}{\mathcal{N}},$$

where g_1 is a constant which depends on the gain of transmitting and receiving antennas, and $\mathcal{N}$ is the power of noise and interference. In our simulations, we chose the value of $\frac{g_1}{\mathcal{N}}$ in such a way that when a node transmits with the maximum transmission power, $P_{max} = 150$ mW, packet error rate for packets of length 128 octets is 0.08 at the border of the transmission range.

The packet format in our simulation model is based on IEEE 802.11 standard. Each transmitted packet on the physical link consists of three parts: a preamble, a physical layer header, and the payload which includes user data and headers from higher layers. Packet reception is also compatible with IEEE 802.11b/g and IEEE 802.15.4 standards. That is, if an error occurs in preamble or header, the packet is dropped and the payload is not detected (i.e., no energy is consumed to detect the payload). If there is no error in the header and preamble, the payload is detected. Nevertheless, if the payload is detected erroneously, the packet will be dropped.

For each transmitted packet by u, $(A_u + \frac{P_{u,v}}{\kappa_u})\frac{x}{r}$ is deducted from its battery energy, where x is the packet size. For each received packet by v, $B_v \frac{x_1}{r}$ is deducted from its battery energy, where $0 \leq x_1 \leq x$ is the size of detected part of the packet. Furthermore, nodes consume a small amount of energy when they are idle (i.e., they do not

[4] We do not analyze the TCP session in this paper. We simply assume that there is an E2E ACK which would acknowledge the packets received.

[5] DBPSK is one of the modulation schemes used in IEEE 802.11b standard.

transmit or receive any data or control packet) and when they sense the medium. For the sake of simulations, the consumed energy at the idle mode and during channel sensing is assumed to be a fraction of the energy that a node consumes during reception of a packet. More specifically, we assume the energy consumption at the idle mode is $k_{idle}\frac{B_u}{r}T_{idle}$, where T_{idle} is the duration that a node is idle, and k_{idle} is a constant. We also assume that the energy consumption during channel sensing is $k_{sense}\frac{B_u}{r}T_{sense}$, where T_{sense} is the duration of sensing the channel and k_{sense} is a constant. The deployed routing protocol is OLSR in which Hello messages are sent periodically every T_{hello} seconds and topology control messages are transmitted every T_{tc} seconds. For each node u, we consider 10 levels of transmission power starting from 15 mW and increasing in steps of 15 mW up to the maximum transmission power $P_{max} = 150$ mW. Values of various parameters used in the simulations are listed in Table 1.

6.2 Performance of the RMER Algorithm

We first compare the energy-efficiency and the reliability of routes discovered by the energy-efficient routing algorithm RMER with that of routes discovered by similar schemes from the literature, which will be introduced accordingly.

6.2.1 The HBH System

We compare energy-efficiency and reliability of routes discovered by the RMER designed for the HBH system with those discovered by the energy-efficient routing algorithm proposed in [4], which considers both energy consumption of nodes and reliability of links in route selection (similar to RMER). We refer to this scheme as traditional minimum energy routing (TMER). TMER neglects energy consumption of processing elements and the impact of HBH ACK.

TABLE 1
Values of Various Parameters Used in Simulations

Parameter	Value
Initial battery energy of each node (B)	100 [J]
Network area	350×350 [m^2]
Path-loss exponent (η)	3
Data rate (r)	100 [Kbps]
Power consumption of transmitter circuit (P_t)	100 [mW]
Power consumption of receiver circuit (P_r)	100 [mW]
Maximum transmission power (P_{max})	150 [mW]
Minimum transmission power (P_{min})	15 [mW]
Power efficiency of transmission amplifier (κ)	10%
Maximum # of transmissions in HBH system (Q_u)	7
Transmission range (d_{max})	70 [m]
Data packet size (L_d)	512 [byte]
MAC ACK packet size (L_h)	240 [bit]
E2E ACK packet size (L_e)	96 [byte]
Hello packet size (L_{hello})	96 [byte]
Battery death threshold (B_{th})	0
Maximum collision probability ($p_{c_{max}}$)	0.3
Channel sensing time (T_{sense})	50 [μs]
k_{idle}	0.2
k_{sense}	0.4
T_{hello}	10 [s]
T_{tc}	20 [s]

In our simulation setup, the considered values for quality of links are such that each link becomes almost reliable after the required number of retransmissions (maximum of 6 retransmissions). Thus, for fair comparison between algorithms, we consider probability of error-free reception of packets over wireless links in a single transmission to determine reliability of a route. Fig. 1a shows the expected amount of energy consumed to route a data packet along a route discovered by RMER and TMER. Fig. 1b shows the average E2E reliability of discovered routes by each of

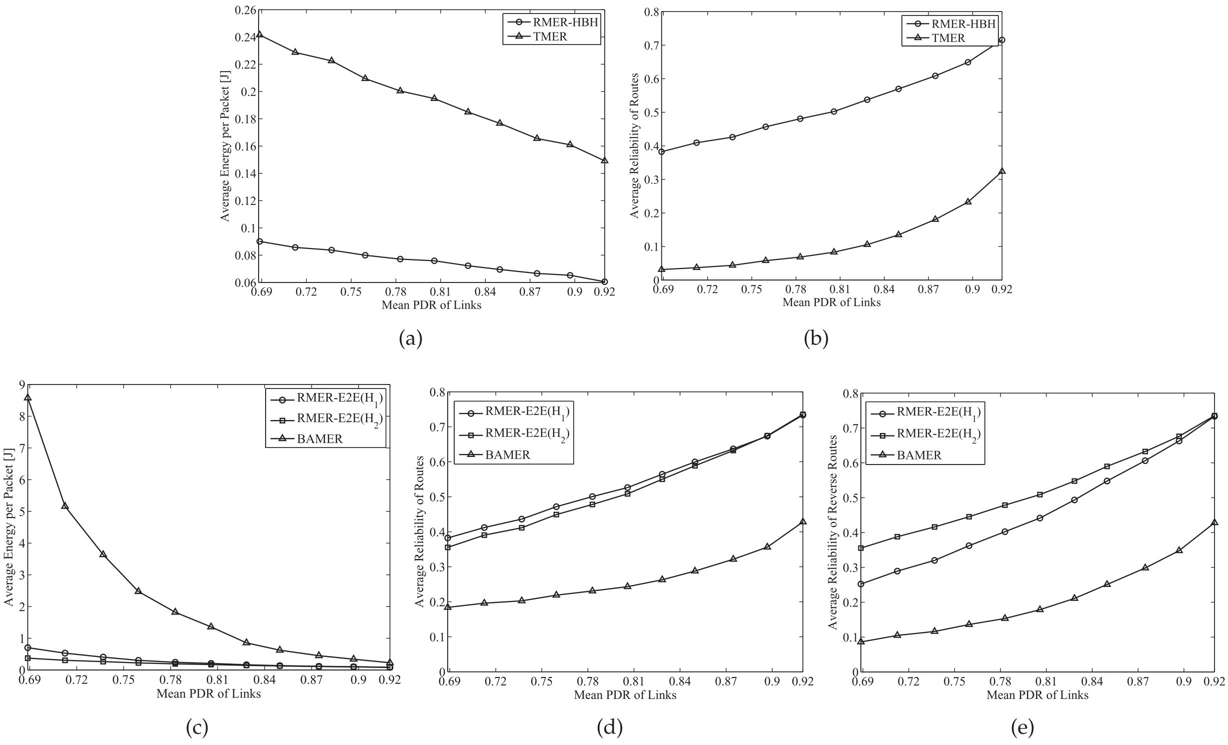
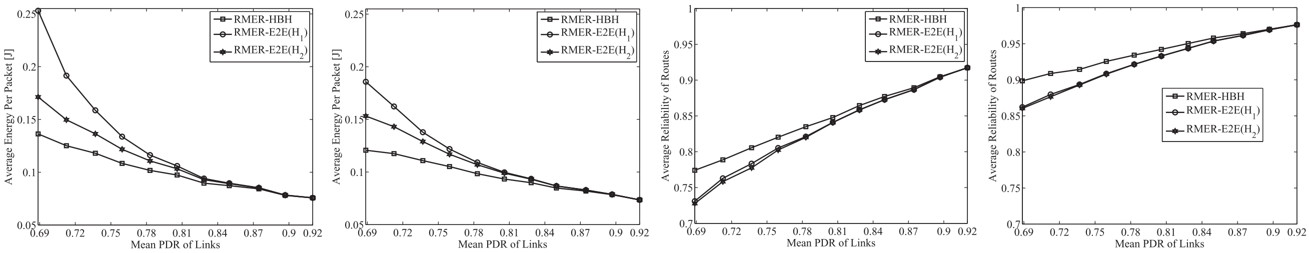


Fig. 1. Energy-efficiency and reliability of routes discovered by RMER and other algorithms. The number of nodes in the network is 200.



(a) $Q_u = 3$ (b) $Q_u = 4$ (c) $Q_u = 3$ (d) $Q_u = 4$

Fig. 2. Energy-efficiency and reliability of routes discovered for the hybrid system by variants of the RMER algorithm. The number of nodes in the network is 200.

them. Results have been depicted in terms of mean PDR of links in the network.

Plots in Figs. 1a and 1b clearly show that RMER not only is able to find more energy-efficient routes compared to TMER, but it is also able to find more reliable routes. As mentioned before, TMER does not consider the energy cost of processing elements of transceivers. It only considers the transmission power, which decays with distance (d^{η}). TMER favors routes consisting of many short hops to routes consisting of few long hops, because short hops are more energy-efficient than long hops. However, when the processing power is considered, routes consisting of many short hops are not energy-efficient anymore, because at each hop a fixed amount of energy is consumed by processing elements. These fixed amounts can neutralize energy-efficiency of routes discovered by TMER. RMER, on other hand, considers the actual energy consumption of nodes, which enables it to find more energy-efficient routes. Regarding reliability, we notice that although transmission error in short hops is lower compared to long hops, the collision probability in wireless networks may even result in a lower PDR for short hops. As a result, when the packet has to be forwarded through many short hops probability of packet loss might even be greater compared to the case where a few long hops are used. This is why we can expect to have more reliable routes in RMER rather than TMER. *In summary, we can state that underestimating the energy consumption of nodes in wireless ad hoc networks can severely affect the reliability of energy-efficient routes.*

6.2.2 The E2E System

In this case, we consider RMER algorithm devised on the basis of the heuristic solutions H_1 and H_2 and an energy-efficient algorithm from the literature which has been designed for the E2E system called basic algorithm for minimum energy routing (BAMER) [5]. We again study the average reliability of routes and the average amount of energy consumed to route a packet. Furthermore, we consider the average reliability of reverse routes which carry E2E ACKs form destination nodes to source nodes. Results are shown in Figs. 1c, 1d, and 1e. These figures clearly show that RMER (both H_1 and H_2) can find more energy-efficient and reliable routes compared to BAMER. Similar to TMER, BAMER does not consider energy cost of processing elements of transceivers, while RMER considers these sources of energy consumption.

We also observe in Figs. 1c and 1d that energy-efficiency and reliability of routes discovered by H_1 and H_2 are almost the same (on average). However, reliability of reverse routes for H_2 is higher than the reliability of reverse routes for H_1. This is due the fact that H_2 considers reliability of both forward and reverse links, while H_1 only considers reliability of forward link. In summary, we can state that H_2 provides a better compromise between energy-efficiency and reliability of routes as well as reliability of reverse routes compared to H_1.

6.2.3 The Hybrid System

So far, we studied the performance of the RMER algorithm when either HBH or E2E retransmissions are supported. We observed that the E2E system has the drawback of increased energy cost in the network, since the PDR of routes drop exponentially with the number of hops. E2E systems, however, can ensure E2E reliability between a source and a destination. On the other hand, a HBH system with limited number of retransmissions in each hop may fail to provide E2E reliability, since packets might be lost in each hop. Here, we consider a situation in which both HBH and E2E retransmissions are supported (henceforth called the hybrid system). In the hybrid system, each link supports a limited number of HBH retransmissions while unlimited number of E2E retransmissions ensures complete reliability between the source and the destination. An immediate question that arises is, which of the RMER algorithms should be used to find energy-efficient routes for the hybrid system: 1) the RMER algorithm designed for the HBH system, or 2) the RMER algorithm designed for the E2E system based on heuristic solution H_1, or 3) the RMER algorithm designed for the E2E system based on heuristic solution H_2? Plots in Fig. 2 show the energy cost and reliability of routes discovered by variants of RMER algorithm for the hybrid system. The figure shows that RMER-HBH finds more energy-efficient and more reliable routes for the hybrid system. However, if links are of good quality (on the average), RMER-E2E algorithms perform similar to the RMER-HBH algorithm.

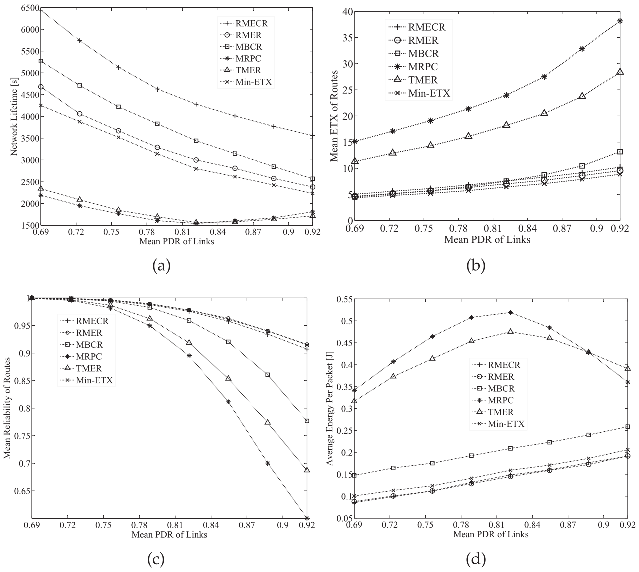
6.3 Performance of the RMECR Algorithm

Now, we compare the reliability and the energy-efficiency of discovered routes by the RMECR algorithm with that of other routing algorithms. We also compare the network lifetime when each of them is deployed in the network. We compare different algorithms in a completely similar setting. We deploy a network randomly in each simulation run. We then create several replicas of the deployed network. In each replica, a different routing algorithm is used to find routes between nodes. Similar traffic sessions are generated between randomly chosen source-destination nodes in all replicas of the network. The interarrival time of sessions is exponentially distributed with a mean of 20 s. The session duration is also exponentially distributed with a mean of 200 s. The source node of the session transmits data packets with the constant rate of 1 packet/s. We recalculate routes in each replica of the network every 40 s. With time, the battery energy of nodes reduces. Route recalculation is required to prevent nodes from being overused. We repeated this procedure 300 times to achieve a minimum of 95 percent confidence level and plotted the average values for each algorithm.[6] The network lifetime in our simulations is defined as the time that the first node failure happens in the network due to battery depletion. Delay in the failure of first node indicates that failure of other nodes is delayed as well. Achieving a higher network lifetime by a routing algorithm shows its capability to avoid nodes being overused.

Here, we consider RMECR and RMER with HBH retransmissions, maximum residual packet capacity (MRPC) [15] which similar to RMECR considers both link reliability and battery energy of nodes in route selection, minimum battery capacity routing (MBCR) [9], which only considers the remaining battery energy of nodes in route selection, and TMER, which is an energy-efficient routing algorithm like RMER. We also consider a routing algorithm which finds the path with the minimum accumulated ETX [1]. We refer to this algorithm as Min-ETX. By finding paths with the minimum accumulated ETX, Min-ETX can find links which have better quality. Min-ETX is an example of a routing algorithm which only considers the quality of links in route selection.

Fig. 3 shows the performance of various routing algorithms as a function of the mean PDR of links for data packets. Fig. 3a clearly shows that RMECR can significantly delay the first node failure compared to the other algorithms. This shows the capability of RMECR to avoid nodes being overused, which in turn increases the network lifetime. As shown in Fig. 3a, the next best performing algorithm is MBCR, and MRPC has the worst performance even though MRPC considers battery energy of nodes in routing. We also observe that the network lifetime when RMER algorithm is used is much higher than the network lifetime when TMER is deployed.

As Fig. 3b shows, RMECR is also able to find routes which their accumulated ETX is very close to the accumulated ETX of routes selected by the Min-ETX algorithm (the optimal value). This is also true for the RMER algorithm. The figure also shows that routes selected by MRPC have a higher accumulated ETX. The reason behind this performance of MRPC is that MRPC uses a max-min route selection scheme which increases the hop count of the selected route unboundedly.



(a)

(b)

(c)

(d)

Fig. 3. Performance of various routing schemes in terms of (a) average number of packets delivered to destination nodes before the first node failure occurs in the network, (b) average ETX of selected routes, (c) average E2E reliability of selected routes, and (d) average energy consumed to route a packet from a source node to a destination node. The total number of nodes is 100.

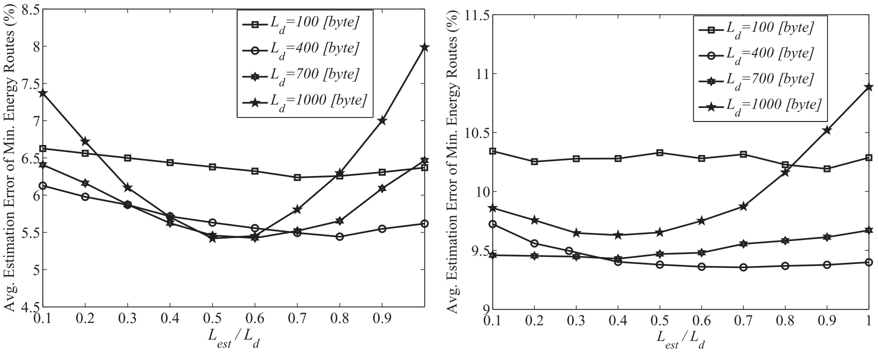
Fig. 3c shows another important feature of the RMECR algorithm. Similar to the Min-ETX and the RMER algorithms, RMECR is able to find more reliable routes. Finally, we observe in Fig. 3d that similar to the RMER, RMECR algorithm is also able to find energy-efficient routes which consume less amount of energy (on the average) to route a packet. In summary, while RMECR can increase the operational lifetime of the network, it is also able to find energy-efficient and reliable routes between nodes.

6.4 Impact of Packet Size on Energy-Efficient Routes

The routing metric in RMER and RMECR algorithms depends on the quality of links. As stated in Section 5, we can use a single packet size L_{est} to estimate the quality of links and consequently the energy cost of the routes. This may impact the optimality of routes in terms of energy-efficiency. Here, we study the effect of this approximation on the energy-efficiency of discovered routes by RMER (energy-efficiency of RMECR is almost similar to RMER). Our objective is to realize how sensitive the minimum energy cost is to the estimation error. To this end, we compare energy-efficiency of the optimal route between two nodes when it is computed knowing the exact and the estimated value of energy cost of routes. The average relative error of the estimation is computed by choosing 300 pairs of nodes in each of 50 randomly deployed networks with 200 nodes.

Fig. 4a shows the relative estimation error for the HBH system as a function of the ratio between the L_{est} and the size of data packets L_d. Fig. 4b shows the estimation error for the E2E system when our proposed heuristic solution H_2 is deployed. Simulation results show that the estimation error of the minimum energy cost for various packet sizes is at most 8 percent in the HBH system and at most 11 percent in the E2E system. Hence, the optimality of the minimum energy route—as discovered by RMER—is not very sensitive to the error in estimating the size of data packets. However, we can still chose an optimum value for L_{est} minimizing the estimation error when data packets are

[6] We do not show the confidence interval in the graph to enable easy comparison.



(a) RMER-HBH

(b) RMER-E2E (H_2)

Fig. 4. Average relative error of estimating the minimum energy cost between two nodes using a single packet size L_{est} for various sizes L_d of data packets.

relatively large. For small data packets, the estimation error remains almost unchanged as $\frac{L_{est}}{L_d}$ changes from 0 to 1.

Here, we specify a guideline on how this optimal packet size could be chosen. Note that the quality of links for both data and ACK packets is estimated using a single packet size L_{est}. ACK packets are relatively small packets. Thus, when the size of data packets is relatively large, a small value for L_{est} will underestimate the quality of links for data packets. On the other hand, a large value for L_{est} will underestimate the quality of links for ACK packets. We can choose a value for L_{est} between these two extremes as an optimum value to minimize the estimation error of energy cost, for example, $L_{est} \approx \frac{L_d}{2}$. This, however, is a rough estimation. Analyzing the optimal value of the L_{est} remains an open issue, which could be addressed in a separate study.

7 Conclusion

We presented an in-depth study of energy-aware routing in ad hoc networks, and we proposed a new routing algorithm for wireless ad hoc networks, namely, reliable minimum energy cost routing (RMECR). RMECR can increase the operational lifetime of the network using energy-efficient and reliable routes. In the design of RMECR, we used a detailed energy consumption model for packet transfer in wireless ad hoc networks. RMECR was designed for two types of networks: those in which hop-by-hop retransmissions ensure reliability and those in which end-to-end retransmissions ensure reliability. The general approach that we used in the design of RMECR was used to also devise a state-of-the-art energy-efficient routing algorithm for wireless ad hoc networks, i.e., reliable minimum energy routing (RMER). RMER finds routes minimizing the energy consumed for packet traversal. RMER does not consider the remaining battery energy of nodes, and was used as a benchmark to study the energy-efficiency of the RMECR algorithm. Extensive simulations showed that RMER not only saves more energy compared to existing energy-efficient routing algorithms, but also increases the reliability of wireless ad hoc networks. Furthermore, we observed that RMECR finds routes that their energy-efficiency and reliability are almost similar to that of routes discovered by RMER. However, RMECR also extends the network lifetime by directing the traffic to nodes having more amount of battery energy. We are in the process of implementing the proposed algorithms on a test bed to study the impact of varying conditions on the performance of these algorithms.

Acknowledgments

The authors would like to thank TRANS, a research cooperation between TU Delft, TNO, and the Royal Dutch KPN for supporting this work. The second author thanks the EU FP7 iCORE Project.

References

[1] D.S.J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing," *Proc. ACM MobiCom,* pp. 134-146, 2003.

[2] S. Singh and C. Raghavendra, "PAMAS—Power Aware Multi-Access Protocol with Signalling for Ad Hoc Networks," *ACM Computer Comm. Rev.,* vol. 28, pp. 5-26, 1999.

[3] J. Gomez, A.T. Campbell, M. Naghshineh, and C. Bisdikian, "PARO: Supporting Dynamic Power Controlled Routing in Wireless Ad Hoc Networks," *Wireless Networks,* vol. 9, no. 5, pp. 443-460, 2003.

[4] S. Banerjee and A. Misra, "Minimum Energy Paths for Reliable Communication in Multi-Hop Wireless Networks," *Proc. ACM MobiHoc,* pp. 146-156, June 2002.

[5] Q. Dong, S. Banerjee, M. Adler, and A. Misra, "Minimum Energy Reliable Paths Using Unreliable Wireless Links," *Proc. ACM MobiHoc,* pp. 449-459, May 2005.

[6] X.-Y. Li, Y. Wang, H. Chen, X. Chu, Y. Wu, and Y. Qi, "Reliable and Energy-Efficient Routing for Static Wireless Ad Hoc Networks with Unreliable Links," *IEEE Trans. Parallel and Distributed Systems,* vol. 20, no. 10, pp. 1408-1421, Oct. 2009.

[7] X. Li, H. Chen, Y. Shu, X. Chu, and Y.-W. Wu, "Energy Efficient Routing with Unreliable Links in Wireless Networks," *Proc. IEEE Int'l Conf. Mobile Adhoc and Sensor Systems (MASS '06),* pp. 160-169, 2006.

[8] S. Singh, M. Woo, and C.S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," *Proc. ACM MobiCom,* Oct. 1998.

[9] C. Toh, "Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks," *IEEE Comm. Magazine,* vol. 39, no. 6, pp. 138-147, June 2001.

[10] D. Kim, J.J.G. Luna Aceves, K. Obraczka, J. Carlos Cano, and P. Manzoni, "Routing Mechanisms for Mobile Ad Hoc Networks Based on the Energy Drain Rate," *IEEE Trans. Mobile Computing,* vol. 2, no. 2, pp. 161-173, Apr.-June 2003.

[11] J.-H. Chang and L. Tassiulas, "Maximum Lifetime Routing in Wireless Sensor Networks," *IEEE/ACM Trans. Networking,* vol. 12, no. 4, pp. 609-619, Aug. 2004.

[12] A. Nagy, A. El-Kadi, and M. Mikhail, "Swarm Congestion and Power Aware Routing Protocol for Manets," *Proc. Sixth Ann. Comm. Networks and Services Research Conf.,* May 2008.

[13] A.B. Mohanoor, S. Radhakrishnan, and V. Sarangan, "Online Energy Aware Routing in Wireless Networks," *Ad Hoc Networks,* vol. 7, no. 5, pp. 918-931, July 2009.

[14] D.J. Vergados, N.A. Pantazis, and D.D. Vergados, "Energy-Efficient Route Selection Strategies for Wireless Sensor Networks," *Mobile Networks and Applications,* vol. 13, nos. 3-4, pp. 285-296, Aug. 2008.

[15] A. Misra and S. Banerjee, "MRPC: Maximizing Network Lifetime for Reliable Routing in Wireless Environments," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '02),* pp. 800-806, 2002.

[16] J. Zhu, C. Qiao, and X. Wang, "On Accurate Energy Consumption Models for Wireless Ad Hoc Networks," *IEEE Trans. Wireless Comm.,* vol. 5, no. 11, pp. 3077-3086, Nov. 2006.

[17] J. Vazifehdan, R. Prasad, and I. Niemegeers, "Minimum Battery Cost Reliable Routing in Ad Hoc Wireless Networks," *Proc. Eighth IEEE Consumer Comm. and Networking Conf.,* Jan. 2011.

[18] S. Gold, "A Pspice Macromodel for Lithium-Ion Batteries," *Proc. 12th Ann. Battery Conf. Applications and Advances,* pp. 215-222, Jan. 1997.

[19] Q. Wang, M. Hempstead, and W. Yang, "A Realistic Power Consumption Model for Wireless Sensor Network Devices," *Proc. Third Ann. IEEE Comm. Soc. Sensor and Ad Hoc Comm. and Networks (SECON '06),* pp. 286-295, Sept. 2006.

[20] P. Liaskovitis and C. Schurgers, "Energy Consumption of Multi-Hop Wireless Networks under Throughput Constraints and Range Scaling," *Mobile Computing and Comm. Rev.,* vol. 13, no. 3, pp. 1-13, 2009.

[21] T.H. Cormen and C.S. Charles, E. Leiserson, and R.L. Rivets, *Introduction to Algorithms,* second ed. MIT Press, 2001.

[22] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," IETF RFC 3626, www.ietf.org/rfc/rfc3626.txt, 2003.

[23] K.-H. Kim and K.G. Shin, "On Accurate Measurement of Link Quality in Multi-Hop Wireless Mesh Networks," *Proc. ACM MobiCom,* pp. 38-49, 2006.

[24] H. Zhang, A. Arora, and P. Sinha, "Link Estimation and Routing in Sensor Network Backbones: Beacon-Based or Data-Driven?" *IEEE Trans. Mobile Computing,* vol. 8, no. 5, pp. 653-667, May 2009.

[25] M. Senel, K. Chintalapudi, D. Lal, A. Keshavarzian, and E. Coyle, "A Kalman Filter Based Link Quality Estimation Scheme for Wireless Sensor Networks," *Proc. IEEE Global Telecomm. Conf. (GlobeCom '07),* pp. 875-880, Nov. 2007.

[26] L. Verma, S. Kim, S. Choi, and S.-J. Lee, "Reliable, Low Overhead Link Quality Estimation for 802.11 Wireless Mesh Networks," *Proc. IEEE Fifth Ann. Comm. Soc. Conf. Sensor, Mesh and Ad Hoc Comm. and Networks (SECON '08),* June 2008.

[27] A. Sheth and R. Han, "Adaptive Power Control and Selective Radio Activation for Low-Power Infrastructure-Mode 802.11," *Proc. 23rd Int'l Conf. Distributed Computing Systems,* May 2003.

[28] J. Proakis, *Digital Communications,* fourth ed. McGraw-Hill, 2000.

Javad Vazifehdan received the BSc degree in electrical engineering from the Iran University of Science and Technology, Tehran, in 2002, the MSc degree in electrical engineering from the University of Tehran in 2005, and the PhD degree (cum laude) from the Department of Telecommunications, Delft University of Technology, The Netherlands, in 2011. In 2004, he joined a private company in Iran working on the development of communication protocols for industrial automation. He is currently a postdoctoral guest researcher at the Delft University of Technology and a design engineer at ASML, a world-class international company producing lithography machines. His main research interests include energy-efficiency in wireless and mobile networks.

R. Venkatesha Prasad received the BE degree in electronics and communication engineering and the MTech degree in industrial electronics from University of Mysore, India, and the PhD degree from the Indian Institute of Science in 1991, 1994, and 2003, respectively. Since 1996, he has been working as a consultant for the ERNET Lab of ECE at the Indian Institute of Science. From 1999 to 2003, he also worked as a consultant for CEDT, IISc, Bangalore, for VoIP application developments. In 2003, he was a team leader at the Esqube Communication Solutions Pvt. Ltd., Bangalore, for the development of various real-time networking applications. From 2005 through 2012, he was a senior researcher, and since 2012, he has been an assistant professor at Delft University of Technology. He is a TPC member of ICC, GlobeCom, ACM MM, and ACM SIGCHI. He was the TPC cochair of the CogNet Workshop in 2007, 2008, and 2009, and the TPC chair for E2Nets at IEEE ICC-2010. He has been running the PerNets Workshop since 2006 with IEEE CCNC. He was the tutorial cochair of CCNC 2009 and 2011 and the demo chair of IEEE CCNC 2010.

Ignas Niemegeers received a degree in electrical engineering from the University of Ghent, Belgium, in 1970, the MScE degree in computer engineering in 1972, and the PhD degree from Purdue University in West Lafayette, Indiana, in 1978. From 1978 to 1981, he was a designer of packet switching networks at Bell Telephone Manufacturing Cy, Antwerp, Belgium. From 1981 to 2002, he was a professor in the Computer Science and the Electrical Engineering Departments, University of Twente, Enschede, The Netherlands. From 1995 to 2001, he was the scientific director of the Centre for Telematics and Information Technology at the University of Twente. Since May 2002, he has been the chair of the Wireless and Mobile Communications Group at Delft University of Technology, where he heads the Telecommunications Department. He is a member of the Expert Advisory Group of the European technology platform eMobility and IFIP TC-6 on Networking. He has (co)authored nearly 300 scientific publications and coauthored a book on personal networks.