Model and Protocol for Energy Efficient Routing over Mobile Ad Hoc Networks

Jinhua Zhu and Xin Wang *Member, IEEE*

***Abstract*—Many minimum energy (energy efficient) routing protocols have been proposed in recent years. However, very limited effort has been made in studying routing overhead, route setup time, and route maintenance issues associated with these protocols. Without a careful design, an energy efficient routing protocol can perform much worse than a normal routing protocol. In this paper, we first show that the minimum energy routing schemes in the literature could fail without considering the routing overhead involved and node mobility. We then propose a more accurate analytical model to track the energy consumptions due to various factors, and a simple energy-efficient routing scheme PEER to improve the performance during path discovery and in mobility scenarios. Our simulation results indicate that compared to a conventional energy efficient routing protocol, PEER protocol can reduce up to 2/3 path discovery overhead and delay, and 50% transmission energy consumption.**

Index Terms: energy efficient routing, overhead, MAC.

I. Introduction

Wireless ad hoc networks usually consist of mobile battery operated computing devices that communicate over the wireless medium. While the processing capacity and the memory space of computing devices increase at a very fast speed, the battery technique lags far behind. Therefore, it is critical to derive energy conservation schemes to increase the device and network operation time.

In wireless networks, the transmitted signal is attenuated at the rate of $1/d^n$, where d is the distance between a sender and a receiver and n is the path loss exponent with value between 2 and 6 depending on the operational environment. Instead of using the maximum transmission power all the time, with power control, a sender can adjust the transmission power according to d. However, link level power control cannot ensure that the end-to-end energy consumption from a source to a destination is minimum. To conserve energy, many energy efficient routing protocols have been proposed [1]-[9]. These protocols can be generally classified into two categories: *Minimum Energy* routing protocols[1]-[6] and *Maximum Network Lifetime* routing protocols[8][9]. *Minimum Energy* routing protocols search for the most energy efficient path from the source to the destination, while *Maximum Network Lifetime* routing protocols attempt to balance the remaining battery-power at each node when searching for the energy efficient path. Since *Minimum Energy* routing scheme is also an important part in most recent *Maximum Network Lifetime* routing protocols such as *Conditional Max-Min Battery Capacity Routing* (CMMBCR) [8] and *Conditional Maximum Residual Packet Capacity* (CMRPC) routing [9], we will focus on developing more efficient *Minimum Energy* routing protocols in this paper.

Minimum Energy routing protocols can be further divided into three classes based on the types of link costs: *Minimum Total Transmission Power* (MTTP), *Minimum Total TransCeiving Power* (MTTCP), and *Minimum Total Reliable Transmission Power* (MTRTP). MTTP protocols use the transmission power as the link cost metric and search for the path with minimum total transmission power between the source and the destination. Authors in [1] modified Dijkstra's Shortest path algorithm to obtain the minimum total transmission power path. PARO in [5] also used transmission power as the link cost, however it targets to reduce energy consumption between any two neighboring nodes. To reduce transmission energy between two nodes, one or more intermediate nodes elect to forward packets on behalf of the peer nodes. MTTCP protocols use the transmission power plus the receiving power as the link cost. Authors in [3] used distributed Bellman-Ford algorithm to obtain the minimum total transceiving power path. MTTP and MTTCP protocols proposed in the literature, however, did not consider the energy consumption due to data packet retransmissions. Instead, authors in [4] proposed a MTRTP protocol to take into account the energy consumption of packet retransmissions. The total transmission power consumed for reliably transmitting a data packet from one node to its neighboring node is used as the link cost.

Jinhua Zhu is with Ericsson Inc, Boulder, CO 80301; Xin Wang is with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794. E-mail: jinhua.zhu@ericsson.com, xwang@ece.sunysb.edu.

This research was supported by U.S. NSF under grant numbers CNS-0751121 and CNS-0628093.

In existing minimum energy routing protocols, signaling packets are often transmitted at the maximum power to reduce the hidden terminal problem as a result of using asymmetric transmission powers from different neighboring nodes. The signaling packet that experiences more collisions, for example the RTS packet in 802.11, would consume significant amount of power. Without taking into account the energy used for signaling, the path discovered could consume much more energy than a path selected based on a more accurate energy consumption model. In addition, most of literature work focused only on the development of new link cost metric. Once a new link cost is derived, the traditional shortest path routing protocols, such as AODV (*Ad hoc On Demand Distance Vector*) and DSR (*Dynamic Source Routing*) protocols, are modified to search for the minimum cost path. However, such straightforward modification would lead to several problems. First, the routing overhead in route discovery phase is very high, which not only consumes a significant amount of energy but also leads to a long path setup delay. Second, the route maintenance scheme used in conventional shortest path routing protocol is not suitable for maintaining energy efficient path in a mobile environment. We will explain these issues in more details in the next section.

In this paper, we first provide a detailed discussion on the problems in traditional energy efficient routing protocols. We then derive a new link cost model to account for energy consumption due to signaling packets at MAC layer, and provide the schemes for estimating the parameters required for calculating the link cost. Based on the new energy consumption model, we propose a *Progressive Energy Efficient Routing* (PEER) protocol for more timely path setup, and for efficient path maintenance. Contrary to conventional energy-efficient routing protocols that try to find the optimal path during route discovery phase and maintain the route reactively, PEER searches for the more energy efficient path progressively and maintains the route continuously. Particularly, a path closest to the most energy efficient path is established between the source and the destination quickly, and then the transmission path adapts whenever necessary with little overhead to ensure more energy efficient transmissions all the time. Our performance evaluation demonstrates that, as compared to normal minimum energy protocols, PEER could significantly reduce routing overhead and path setup delay, and consume much less energy in both static and mobile scenarios.

The rest of the paper is organized as follows. Section II describes the observation and motivation for this paper. In Section III, we extend our previous work [6] and propose an efficient way to estimate the accurate link cost. The detailed PEER protocol is described in Section IV. Performance evaluation is conducted in V. Section VI concludes the work.

II. Observation and Motivation

There are many existing routing protocols for wireless ad hoc networks. In general, these protocols can be categorized as table-driven, on-demand, and hybrid. In table-driven routing protocols, all nodes need to advertise the routing information periodically to keep an up-to-date view of the network topology. Different from table-driven routing protocols, on-demand routing protocols create a transmission path only when required by the source node. Hybrid protocols combine both approaches. For example, in *Zone Routing Protocol* (ZRP), table-driven routing scheme is used for intra-zone routing and on-demand routing scheme is used for inter-zone routing. Most of energy efficient schemes proposed in the literature modified on-demand routing protocols such as AODV [16] or DSR [17] to build energy efficient path since the routing overhead is very high in table-driven routing protocols [2].

In on-demand routing protocols such as AODV, a node will start a route discovery process if it needs to find a path to a destination. It broadcasts the route request packet and waits for the reply from the destination. The neighboring nodes that receive such route request packet will rebroadcast it, and so on. To reduce the routing overhead, the intermediate nodes will only rebroadcast the first received route request packet and discard the following duplicate ones. In addition, the destination node only replies to the first route request packet. For example, in Fig. 1, both A and B are neighboring nodes of S and D, and S needs a route to D. So if S broadcasts a route request packet, both A and B will receive the packet. Assume A rebroadcast such packet next, node S, B and D will receive the packet. In a conventional on-demand protocol, node S and B will discard the rebroadcast route request packet as they have already received the same route request from S. Therefore the final route discovered is SAD. It is apparent that the overhead for these on demand routing protocols is O(n), where n is the number of nodes in the network.

Route discovery in energy efficient routing protocols is however quite different. The intermediate nodes could not simply discard the duplicate route request packets now as such packets may come from more energy efficient paths. That is, the intermediate nodes need to process and rebroadcast the duplicate route request packets if they come from a more energy efficient path. Therefore, the nodes may need to broadcast the same route request packet many times. For the same example in Fig. 1, node B may need to broadcast both the packets from S and A if the path SAB is more energy efficient than SB. Based on the Bellman-Ford algorithm [14], we can obtain that routing overhead for minimum energy efficient routing protocols is O(n^2) now, which increases dramatically with the number of nodes (n) in the network. After our initial work in [20], similar observations were made in [15] and the observed problem is called *Flooding Waves*.

Fig. 1. A linear topology.

Higher routing overhead causes several issues. The first one is higher energy consumption. As the path discovery packets are very important, they have to be transmitted at the maximum power level. Therefore, even though the size of routing packets is small, the energy consumption for one route discovery packet is comparable to one data packet. The second one is longer route setup delay[1]. There are two main reasons for this. One is that an energy efficient path generally has more intermediate nodes than the shortest path, so it takes longer time for the route request and route reply packets to go through all the intermediate nodes. The other is that extra processing and re-broadcasting of path discovery packets in energy efficient routing protocols would cause higher delay at each link.

To address the Flooding Waves problem, the authors in [15] proposed a *Delayed-Forwarding* scheme, in which the intermediate nodes will wait for a certain period before forwarding the route request packet. The delay period is proportional to the distance between itself and the sender. However, this scheme will increase the route setup time.

Besides the above issues in route discovery phase, there will be also high energy consumption in route maintenance phase. Energy efficient routing protocols tend to use many more intermediate nodes than regular on-demand routing protocols and the length of each link tends to be shorter. This can help reduce the link breakage rate in mobile scenarios as a node has a larger moving range without breaking the link. However, as a result of the mobility, the original minimum energy path may no longer be energy efficient before the link breaks. The worst case would be that the transmission power for each link is at the maximum level when distance between two end nodes of the link reaches the transmission range. In this case, there would be much higher energy consumption by using an energy efficient routing protocol than using a regular on-demand routing protocol since a path built through energy efficient routing protocol has much more hops. The route maintenance schemes in regular on-demand routing protocols such as DSR or AODV generally trigger path rebuilding only when one or more links are broken. Refreshing the path from the source node periodically is also not a good option, as it not only consumes a lot of energy during each path refreshing, but also it is difficult to set an appropriate refreshing cycle. Refreshing the path frequently may waste the energy unnecessarily, while insufficient path update may lead to an inefficient path with more energy consumption.

[1] The time from the source node broadcasts the route request packet until it finds the desired path such as the minimum energy path.

From these observations, we can see that an energy efficient routing protocol should reduce the overhead during route discovery and have more efficient path maintenance scheme. Therefore in this paper, we propose a new link cost metric to facilitate the finding of more energy efficient routing path, and a *Progressive Energy Efficient Routing* (PEER) protocol to speed up path setup while adaptively adjusting the routing path to improve transmission performance and minimize end-to-end energy consumption.

III. New Energy Consumption Model for 802.11

PEER is a cost-based energy efficient routing protocol. In a cost-based routing protocol, the total cost of all the links on each available path between the source node and the destination node will be calculated, and a minimum cost path (meeting certain criteria) will be selected. As link cost is very important in the cost-based energy efficient routing protocols, it is critical to derive an accurate link cost metric to obtain an optimal path. In this section, we will derive the link cost and show how to estimate the parameters needed for link cost calculation. As PEER will run over 802.11 MAC, in the following, we will derive the link cost for 802.11 wireless networks.

A. Energy Consumption Model for 802.11

Two MAC schemes have been specified in IEEE 802.11[10]: DCF (Distributed Coordination Function) and PCF (Point Coordination Function). As PCF is a centralized protocol, we will only consider DCF at MAC layer in this paper as it will be used to work with PEER.

For better describing our link cost model, we first give a brief overview of DCF. IEEE 802.11 DCF is based on CSMA/CA (Carrier Sensing Multiple Access with Collision Avoidance) mechanism. It consists of two carrier sensing schemes, namely physical carrier sensing and virtual carrier sensing. The virtual carrier sensing scheme is implemented with NAV (Network Allocation Vector). If a node receives a packet (such as RTS, CTS and DATA packet), it will update NAV with the duration included in the received packet. The NAV value indicates when the on-going transmission session will end.

If a node has data packets to send to another node, it first checks its NAV. If its NAV is larger than 0, it has to wait until NAV reaches 0. After that, the sender transmits a RTS packet after the channel is available for a period longer than DIFS (DCF InterFrame Space) or the backoff timer reaches zero. The receiver responds with a CTS packet after receiving the RTS packet[2]. If the sender does not receive the CTS packet within a predetermined time interval, it will retransmit the RTS packet. After receiving the CTS, the sender will send out the data packet and the receiver will reply with an ACK packet after receiving the data packet successfully. If the sender doesn't receive the ACK packet within a predefined time period, the whole process will be repeated.

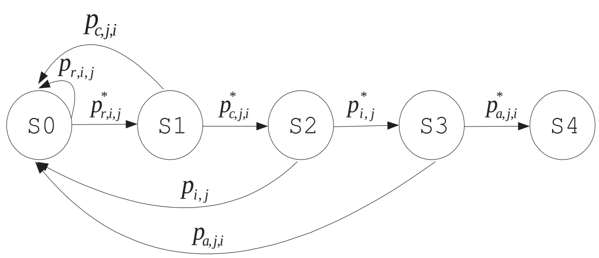

Fig. 2. State diagram for the Four Frame Exchange scheme.

Based on the DCF protocol, in the following, we will develop the link cost model. Denote the packet error rates for RTS, CTS, DATA and ACK packets between node i and j by $p_{r,i,j}$, $p_{c,j,i}$, $p_{i,j}$, and $p_{a,j,i}$. In addition, for a variable x, denote $1-x$ by x^*, and the mean value of x by $\overline{x}$. Then the state diagram for transmitting a data packet from node i to one of its neighboring nodes, node j, is shown in Fig. 2, where $S0$ is the initial state. After node i transmits the RTS packet, the state will change into $S1$ with probability $p^*_{r,i,j}$ or remain in $S0$ with probability $p_{r,i,j}$, depending on whether the RTS packet is received by node j correctly or not. If node j receives the RTS packet, it will send out the CTS packet. With probability $p^*_{c,j,i}$, CTS will be received by node i, and the state will change from $S1$ to $S2$; With probability $p_{c,j,i}$, the state will return to $S0$. Once node i receives the CTS packet, it will transmit the data packet. With probability $p^*_{i,j}$, the data packet will be received by node j, and the state will change from $S2$ to $S3$; With probability $p_{i,j}$, the state will return to $S0$. After node j receives the data packet, it will acknowledge the data packet. With probability $p^*_{a,j,i}$, ACK will be received by node i, and the state will change from $S3$ to $S4$, where the whole process ends; With probability $p_{a,j,i}$, the state returns to $S0$. Note that, we assume the number of retransmissions is unlimited in the above state diagram. Even though the number of retransmissions is limited in 802.11 (e.g., the short retry limit is 7 and long retry limit is 4) [10], this will not affect the model significantly since most of the packet retransmissions will not be over the transmission limit.

[2]If a node receives a RTS but can't reply with a CTS because the channel is busy, we still treat it as a RTS packet error in our analysis even though the RTS packet is received correctly.

From the state diagram, we can see that: on average node i needs to transmit $1/p^*_{r,i,j}$ RTS packets so that node j can receive one correctly (from state $S0$ to state $S1$). Similarly, node j needs to transmit $1/p^*_{c,j,i}$ CTS packets (from state $S1$ to state $S2$), node i needs to transmit $1/p^*_{i,j}$ data packets (from state $S2$ to state $S3$), and node j needs to transmit $1/p^*_{a,j,i}$ ACK packets (from state $S3$ to $S4$). Therefore, the average numbers of RTS, CTS, data, and ACK transmissions in the whole process are as follows: RTS: $1/(p^*_{r,i,j}p^*_{c,j,i}p^*_{i,j}p^*_{a,j,i})$, CTS: $1/(p^*_{c,j,i}p^*_{i,j}p^*_{a,j,i})$, data: $1/(p^*_{i,j}p^*_{a,j,i})$, ACK: $1/(p^*_{a,j,i})$.

With power control scheme, RTS and CTS packets are transmitted at the maximum power level P_m in order to reduce hidden terminal problem, while DATA and ACK packets are transmitted at the minimum required transmission power level $P_{i,j}$ between node i and j for energy conservation. In order to reduce collisions as a result of using asymmetric power in control and data packet transmissions [11], the nodes set their NAVs to the EIFS (Extended InterFrame Space) duration if they can sense the signal but can not decode it correctly. Although periodic power raising was proposed in [11] to reduce collision rate further, it will increase the power consumption in the case that there are not many nodes competing for the channel access. As the setting of NAV to EIFS scheme has already reduced the collision rate significantly, in our scheme, the periodic power raising is not assumed.

To simplify the expressions in the analysis, we denote the data size, the 802.11 header size, the RTS packet size, the CTS packet size and ACK packet size by N, N_{hdr}, N_{rts}, N_{cts} and N_{ack} respectively. And we also define the following symbols:

$$N_8 = N + N_{hdr} + N_{phy}, N_r = N_{rts} + N_{phy},$$
$$N_c = N_{cts} + N_{phy}, \text{and } N_a = N_{ack} + N_{phy},$$

where N_{phy} is the size of physical layer overhead. Then the average total transmission power for successfully transmitting a packet from node i to node j is

$$\begin{aligned}\overline{P_T(i,j)} &= \frac{P_m N_r}{p^*_{r,i,j} p^*_{c,j,i} p^*_{i,j} p^*_{a,j,i}} + \frac{P_m N_c}{p^*_{c,j,i} p^*_{i,j} p^*_{a,j,i}} \\ &+ \frac{P_{i,j} N_8}{p^*_{i,j} p^*_{a,j,i}} + \frac{P_{j,i} N_a}{p^*_{a,i,j}} \\ &= \frac{P_m (N_r + N_c p^*_{r,i,j})}{p^*_{r,i,j} p^*_{c,j,i} p^*_{i,j} p^*_{a,j,i}} + \frac{N_8 P_{i,j} + N_a P_{j,i} p^*_{i,j}}{p^*_{i,j} p^*_{a,j,i}} \quad (1)\end{aligned}$$

In addition, denoting the receiving power as P_r, then the average total receiving power for successfully receiving a packet from node i to node j is

$$\overline{P_R(i,j)} = P_r \frac{\frac{N_r}{N_8} + (\frac{N_c}{N8} + p^*_{i,j} + \frac{N_a}{N_8} p^*_{i,j} p^*_{a,j,i}) p^*_{c,j,i}}{p^*_{c,j,i} p^*_{i,j} p^*_{a,j,i}}. \quad (2)$$

Assume there are $M-1$ intermediate nodes between a source and a destination. Let the nodes along the path from the source to the destination be numbered from 0 to M in that order. Then the average total power for reliable transmission along the path from the source (node 0) to the destination (node M) is

$$\overline{P_{total}} = \sum_{i=0}^{M-1} \left(\overline{P_T(i,i+1)} + \overline{P_R(i,i+1)} \right). \quad (3)$$

Based on this formula, it is apparent that $(\overline{P_T(i,i+1)} + \overline{P_R(i,i+1)})$ would be the link cost between node i and $i+1$.

B. Parameter Estimation for Link Cost

Most parameters in the link cost model (Equ. (1)) can be easily obtained except the transmission powers ($P_{i,j}$ and $P_{j,i}$) and the packet error rates ($p_{r,i,j}$, $p_{c,j,i}$, $p_{i,j}$ and $p_{a,j,i}$). In this section, we will show how to estimate these parameters.

For parameter estimation purpose, we make the following assumptions: (1) the path loss between two nodes is symmetric on both directions; (2) the physical layer can provide the information on the average power level of a packet (such as RTS/CTS) received and the average interference level to the MAC layer. These are common assumptions made in many power control schemes as well as energy efficient routing protocols.

Since the wireless signal is attenuated at the rate of $\frac{1}{d^n}$ (d is the distance and n is the path loss exponent), the received power level (P_r) at the receiver is proportional to $\frac{P_t}{d^n}$, where P_t is transmission power level. That is,

$$P_r = K \frac{P_t}{d^n}$$

where K is a factor depending on the environment. With this formula, a node can send a packet at a known power level, and calculate the desired transmission power for other packets based on the received power level of the known packet and the target receiving power. For example, if node A receives a packet P_e (e.g., RTS, CTS and broadcast packets) at per bit power level P_r, and it knows that the packet was sent by B using maximum per bit transmission power (P_m), then node A can calculate the necessary per bit transmission power node B needs to use to transmit other packets to A with the following equations:

$$\begin{cases} P_r = K \times P_m / d^n \\ P_r^{th} = K \times P_t(B,A)/d^n \end{cases}$$

where P_r^{th} is the minimum necessary received power level. It is easy to obtain $P_t(B,A)$ as

$$P_t(B,A) = P_r^{th} \times \frac{P_m}{P_r}.$$

As we assume that the path loss is the same on both directions, $P_t(A,B)$ for a packet from A to B will be the same as $P_t(B,A)$. That is, node A can estimate its necessary transmission power as well as the necessary transmission power B should use to transmit a packet to itself.

Packet error is mainly caused by collision, interference and noise. Here we distinguish the concept of collision and interference by the carrier sensing zone. If the error is caused by the nodes within the carrier sensing zone, we call it collision, otherwise interference.

It is easy to obtain the interference and noise level since each node can monitor it when the channel is free. With interference and noise level measured, we can then calculate the bit error rate (BER) based on the received power level and modulation scheme [4]. Once we get the BER, we can calculate the packet error rate (PER) caused by the interference and noise (assuming there is no error correction scheme) as $PER = 1-(1-BER)^L$, where L is the number of bits in the packet.

For 802.11, most collisions happen during the transmission of RTS. Therefore, we only need to consider the packet error rate caused by the collision of RTS packet. Authors in [12] presented a simple way to estimate the collision probability by counting the number of busy/idle slots:

$$p_c(t+1) = \alpha p_c(t) + \frac{1-\alpha}{N} \sum_{i=0}^{N-1} C_{t-i},$$

where $p_c(t)$ is the estimated collision probability at time t, α is the remembering rate, and C_{t-i} with $i = 0, ..., N-1$ are the last N slot samples. C_i is equal to 0 if the i-th slot is free or the node transmits successfully in such slot; otherwise C_i is 1.

Therefore, the packet error rates for CTS, DATA and ACK packets are calculated based on the interference and noise power, the receiving power, and the packet size. While for RTS packet, we need to take into account the packet error rate caused by both interference and collision. Denote the packet error rate due to interference and noise by p_{int}, the packet error rate due to collision by p_c, the packet error rate of RTS packet can be calculated as:

$$p_{r,i,j} = p_{int} + p_c - p_{int} * p_c.$$

IV. PEER Protocol

As discussed in Section II, the existing minimum energy based routing schemes often introduce big overhead during path discovery and the path setup time is very long. On the other hand, a routing strategy should not get some arbitrary path quickly and rely on a route maintenance scheme to adjust the path later to an energy efficient one, as it may take much more time and create a larger overhead to adapt the route and there is no guarantee that such adaptation could find a path that leads to energy saving comparable to the minimum energy one. Therefore, a good strategy is to find a path close to the minimum energy one quickly and then use a maintenance scheme to adjust the path for further energy reduction.

Taking this into consideration, PEER searches for the energy efficient path quickly during route discovery process, and maintains the route actively so that it can respond to topology and channel changes quickly. In the following, we show how PEER achieves both goals.

A. Route Discovery Process

In this section, we introduce the route discovery strategy of PEER. The quickest way to find a path between two nodes would be through a shortest path routing scheme. However, there may exist a few shortest (smallest number of hops) paths between the source node and destination node. For example, in Fig. 3, assuming all the intermediate nodes (A, B, E, F, G, H) are the neighboring nodes of both S and D while S and D are beyond transmission range, then there are six shortest (2 hops) paths (SAD, SBD,SED, SFD, SGD, SHD). Among all the shortest paths, it is better to pick the most energy efficient one (we call it *minimum energy shortest path*).

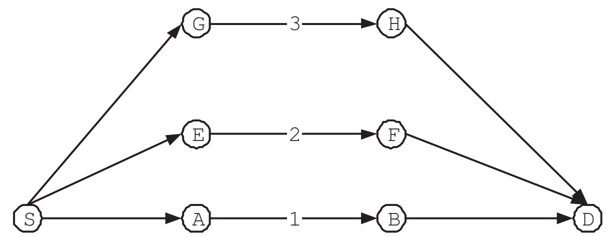

Fig. 3. Three routes between node S and D.

Denote the set of paths between the source and the destination by L, the number of hops for path l by N_l, and the energy consumption for link i in path l by $E_{l,i}$, then the set of shortest paths L_s would be

$$L_s = \arg\min(N_l), l \in L.$$

The set of minimum energy shortest paths L_{ms} would be

$$L_{ms} = \arg\min(\sum_{i=1}^{N_l} E_{l,i}), l \in L_s.$$

Even though there may be more than one minimum energy shortest path in L_{ms}, the routing protocol can pick a unique one by some criterion, such as route request packet arriving time.

Based on the previous definition, the basic searching algorithm would be: (1) search for all shortest (fewest hops) paths; (2) pick the minimum energy path(s) among the shortest paths in (1). To implement this algorithm, the route request packet should carry two pieces of information: one is the hop count, the other is the energy consumption. The source node first broadcasts the route request packet with both hop count and energy consumption set to 0. Once an intermediate node receives a route request packet, it first updates the hop count (increased by 1) and energy consumption (increased by the energy consumption between the sender and itself) information in the route request packet. Then it will rebroadcast such packet only if one of the following conditions holds:

1) The node hasn't received such a packet before or the packet comes from a shorter (smaller number of hops) path;
2) The packet comes from a path with the same number of hops as the best path so far, but the energy consumption is lower.

The first condition ensures that the shortest path is selected, while the second condition selects the minimum energy path from all the shortest ones.

This algorithm, however, has similar path selection issues as other energy efficient routing protocols. That is, the destination node may receive many route request packets from different possible minimum energy shortest paths, but it could not tell which one is the best until it receives all possible packets. As the destination node has no knowledge on the number of route request packets it will receive, it may not be able to make the decision even if it has already received all the route request packets. For example, assuming all six shortest paths (SAD, SBD, SED, SFD, SGD, SHD) in Fig. 3 have the same energy consumption and the destination D has received all of them, D may still not be able to select the best one as it does not know when is the best time to make the decision. There are several ways to deal with this issue at the destination node. One option is that the destination sends a route reply for each route request it receives. As the destination may need to send out many route reply messages, this method will waste energy. Also, the source node might transmit some data packets on less energy efficient path before the best one is found. Another option is that the destination sets up a timer after receiving a route request packet. If it receives another route request before the timer goes off, it will reset the timer. Otherwise, it will select the best path found before the timer goes off and reply the source with a route reply packet. The third option is to set up a time window, and the destination will select the best path within the time window. The last two methods help reduce the energy consumption, but it may increase the route setup time. In this paper, we use the second one as it can adapt to the number of arriving route request packets. If there are only very few route request packets arriving at the destination, the destination can send back route reply packet quickly to reduce the route setup time. On the other hand, it can wait for a period of time for the route request packet from a more energy efficient path to arrive if there are more route request packets arriving at the destination and there is no significant time difference between two consecutive request packets.

B. Route Maintenance

The route obtained in path discovery phase is suboptimal and may still lead to a higher end-to-end energy consumption than that of the minimum energy path. In addition, the network environment can change dramatically due to node movements and dynamic channel conditions, and the previous energy efficient route may no longer be efficient as time goes on. Therefore, the route maintenance phase is very critical for energy efficient routing protocols.

TABLE I
AN EXAMPLE LINK ENERGY TABLE RECORDED BY A NODE D

(a)	(b)	(c)	(d)	(e)	(f)	(g)
A	B	5	S1	D1	1	0
B	C	4	S1	D1	1	1
D	B	3	S2	D2	3	3
F	G	7	S3	D3	5	4
B	E	2	S2	D2	3	5

As extra signaling messages will consume more energy, the route maintenance scheme of PEER will not use additional periodic messages. Instead, an observing node will passively monitor data packets exchanged in its neighborhood and collaborate with its neighbors to look for a more energy-efficient path. As described in Section III, each node can estimate the necessary transmission power and the link cost to a neighboring node once it receives RTS, CTS or broadcast packet from this node. In PEER, each forwarding node will insert the link cost into the IP header of the packet targeted for its next-hop receiver as an IP option, and every node will monitor the data packets exchanged in its neighborhood to intercept the corresponding link costs and use these link costs to estimate the cost of a path segment. For each data packet transmitted, received, or overheard by a node, it will record the following information into a link cost table: (a) sender; (b) receiver; (c) link cost between the sender and the receiver; (d) source; (e) destination; (f) IP header ID; (g) the current time. Among these parameters, (a) and (b) can be obtained from the MAC header, while (c) to (f) can be obtained from the IP header. The information for a link will be kept only for a short time for accurate information and reducing storage overhead.

From the link cost table, a node can know how a packet passes through its neighborhood and the total link cost for that. For example, node D's link energy table is shown in Table I. As the parameters (source, destination, and IP header ID) can identify a packet, we can see in the table that node D records the path info for three packets: P1(S1, D1, 1), P2(S2, D2, 3) and P3(S3, D3, 5). The first packet (P1) goes through a two-hop path segment (A$\rightarrow$ B $\rightarrow$ C) in D's neighborhood and the total cost of the path segment is 9 $(5 + 4)$. The second packet (P2) goes through another two-hop path segment (D$\rightarrow$ B $\rightarrow$ E) and the total cost of the path segment is 5 $(3 + 2)$. The third packet (P3) goes through a one-hop path segment (F$\rightarrow$ G) and the link cost is 7.

Based on the information recorded in its link cost table, a node can help reduce the cost of a local path segment and hence the cost of the end-to-end path between a source and a destination with the use of the following three operations: Remove, Replace, and Insert. Fig. 4 illustrates how the three operations work around a node D. In the following, we will explain the three operations in details.

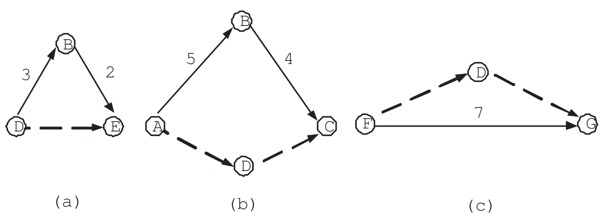

Fig. 4. Remove, Replace, and Insert.

(a) Remove

Assume there is a two-hop path segment X→ A → B on the path to a destination Z in node X's link cost table and the total cost of the path segment is T.

If X finds the link cost between X and B is smaller than the cost of the two-hop path segment, it will update its routing table by setting the next hop for the destination Z to B.

In Fig.4(a), node D has the two-hop path info (D→ B → E) from its link energy table with destination D2 and and the total link cost (5) for such path. If node E is one of D's neighboring nodes, D can estimate the link cost to E ($\overline{P_T(D,E)}$) from the RTS or CTS packets transmitted by node E. If $\overline{P_T(D,E)} < 5$, then D will update its routing table by setting the next hop for destination D2 to E. The subsequent packets for destination D2 will go through E directly.

(b) Replace

Assume there is a two-hop path segment A → B → C on the path to a destination Z in node X's link cost table and the total cost of the path segment is T.

If X finds the total cost for the path segment A→ X → C is smaller than that of the two-hop path segment A→ B → C, X will update its routing table by setting its next hop for the destination Z to C. In addition, it will request A to set its next hop for the destination Z to X.

In Fig. 4(b), Node D has information on the two-hop path segment (A→ B → C) in its link cost table for destination D1 and the total cost of the path segment is 9. If both A and C are D's neighboring nodes, D can estimate the link costs to them ($\overline{P_T(D,A)}$, $\overline{P_T(D,C)}$). If $\overline{P_T(D,A)} + \overline{P_T(D,C)} < 9$, the path A→ D→ C is more energy efficient than A→ B→ C. So node D will set its next hop for destination D1 to C and request A to update its next hop for destination D1 to D. If A accepts the request of D, A will forward D the subsequent packets destined to D1, and D will forward them to C. If A does not accept the request of D, the routing information for destination D1 at node D will be purged after a timeout period.

(c) Insert

Assume there is a one-hop path segment A → B on the path to a destination Z in node X's link cost table and the total cost of the path segment is T.

If X finds the total cost of the path segment A→ X → B is smaller than that of the one-hop path segment, it will update its routing table by setting its next hop for the destination Z to B. In addition, X will request A to set its next hop for the destination Z to X.

In Fig. 4(c), Node D records the link cost of the one-hop path segment (F→ G) destined to D3 as 7. If both F and G are D's neighboring nodes, D can estimate the link costs to them ($\overline{P_T(D,F)}$, $\overline{P_T(D,G)}$). If $\overline{P_T(D,F)} + \overline{P_T(D,G)} < 7$, then the path segment F→ D→ G is more energy efficient than F→ G. So node D will update its routing table by setting its next hop for destination D3 to G and request F to update its next hop for destination D3 to D.

It is worthwhile to point out that both Replace and Remove operations can be applied to a path segment with more than two hops. However, to estimate the link cost without incurring extra signaling cost, all the nodes on the path segments monitored should be neighbors of the monitoring node. The probability of having a path segment longer than two hops and with all the nodes on the path segment within then direct monitoring range is very small. In addition, the maintenance operations on a path segment with more than two hops sometimes can be replaced by several operations on a one-hop or a two-hop path segment. Therefore, we restrict both operations to two-hop path segments only.

In the proposed maintenance scheme, a monitoring node only needs to send out control messages in Replace and Insert operations to facilitate path change. As the control messages are only sent out when a better path is detected, so the maintenance overhead is very low. The control message includes: operation type, requester ID, destination, next hop on the old path segment, the total cost for the new path segment. In our example shown above, the control message that D sends to A for Replace operation is [Replace, D, D1, B, the total cost of new path segment ADC], while the control message that D sends to F for Insert operation is [Insert, D, D3, G, the total cost of new path segment FDG]. Once a node receives a control message, it will first check its routing information for the destination in its routing table. If the next hop for such destination is different from that in the control message, it will discard such control message since the route has changed and the estimation based on the old path is no longer valid.

Within these three operations, Insert may be more easily requested than the other two since it only needs to check one-hop transmission. However, this may not be desirable. For example, in Fig 5, node A transmits a data packet to node B. D overhears the packet transmission and finds out it could save energy if it is inserted into the single hop path segment between nodes A and B. Similarly, node E may be inserted between nodes B and C. Therefore, the final path will be ADBEC. However, there are two more options, AC and AFC, and AFC is the best path segment. It would be better to let Remove and Replace to have higher priority than Insert. In PEER, each node receiving Remove or Insert requests will wait for some time before making the decision. If it receives an Insert request and also an operation request of Replace or Remove, it will take the other operation. If it has both Remove and Replace operation requests, it will select the one which allows for a higher percentage of energy saving. For the same example, node A receives the Insert (by node D), Remove (by node C), and Replace (by node F) requests, it will only perform Remove and Replace operations. As taking path segment AFC will save more energy than taking path segment AC ($\overline{P_T(A,F)} + \overline{P_T(F,C)} < \overline{P_T(A,C)}$), it selects the Replace operation.

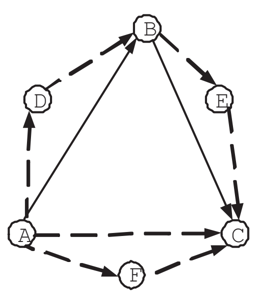

Fig. 5. An undesired improvement

V. Performance Evaluation

We have simulated PEER, minimum energy protocol MTRTP, as well as normal AODV protocols in Glomosim. To implement MTRTP protocol, we modified AODV to search for the minimum cost path using the new link cost derived in [4]. The reference per hop transmission distance is 250m. For energy conservation purpose, many smaller hops are taken. Power control is used in all three protocols, including normal AODV protocol, in which a transmitter adjusts the transmission power based on its actual distance to the next-hop receiver. The network area in the simulation is set to $1200(m)X1200(m)$ and the nodes are randomly distributed in the network. The available transmission power levels are $1, 5, 10, 15, 20, 25, 30, 35$ mW. The P_m is set to 35mW. The session arrival rate follows Poisson distribution and the session duration follows Exponential distribution. The application protocol is CBR (Constant Bit Rate) and the source and destination pairs are randomly selected. The mobility follows modified random waypoint model [19] with 30-second pause time. For each CBR session, fifty packets are sent per second. The path loss and collision rate are estimated using method in [12]. The remembering rate, which is called filter memory in [11], is set to 0.99. A simulation result was gained by averaging over 20 runs with different seeds. Some other default setup parameters are in Table II.

TABLE II
Default Setup Parameters

Parameter	Value	Parameter	value
Number of Nodes	60	Packet Size(byte)	512
Connection Arrival Rate	30	Connection Duration(min)	6
Max Speed(m/s)	10	Min Speed(m/s)	0.5

We assume that there is no power saving mode for the nodes, and accordingly, a node will spend energy in monitoring the channel even if it doesn't receive a packet. A node also consumes energy when overhearing packet transmissions. Therefore, the receiving power cannot be actively controlled. In the simulations, we thus ignore the receiving power and focus only on the comparison of transmission power. We first evaluate the accuracy of the proposed cost model, we then study the performance of route discovery for each protocol, and finally we consider energy consumption as well as RTS retransmissions in both static and mobile environment.

A. Accuracy of Energy Consumption Model

In this study, we evaluate the accuracy of our model as well as that of MTRTP. In our simulation, the transmission power level is set to 1mW for data packets, and 5mW for RTS and CTS packets. To exclude the impact of finding a route on the energy consumption, we use static route and consider only one path from the source (numbered as node 0) to a destination node that is 2 to 6 hops away along the path (numbered as nodes 2 to 6, respectively).

We use CBR (Constant Bit Rate) to transmit 65,536 data packets. The packet error rate is set to 0.001. The simulation results and the energy consumption estimated by each model are shown in Figs. 6. From the results we can see that energy consumption based on our cost model matches the simulation result very well, while

MTRTP underestimates the energy consumption and the difference gets larger as the number of intermediate nodes increases.

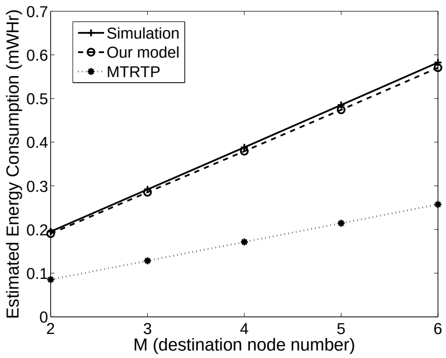

Fig. 6. Estimated energy consumption vs. simulation results.

B. Routing Overhead and Setup Time

In this study, we simulated 10,000 connection requests for each protocol and collected the total number of routing packets, total energy consumption, and total setup time on each simulation. The simulation results are in Fig. 7-9.

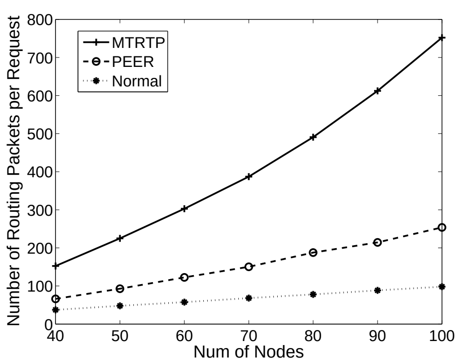

Fig. 7. Routing Overhead.

It is clear from the results that the normal on-demand routing protocol performs the best in terms of routing request, energy consumption for routing overhead, and path setup time, followed by PEER and MTRTP. Both the routing overhead and setup time for MTRTP are much higher than those of the on-demand routing protocol, and the overhead and setup time increase dramatically as the number of nodes increases. This is because the routing overhead for MTRTP is $O(n^2)$ (n is the number of nodes) as discussed in Section II. Therefore MTRTP could not scale well with the number of nodes in the network.

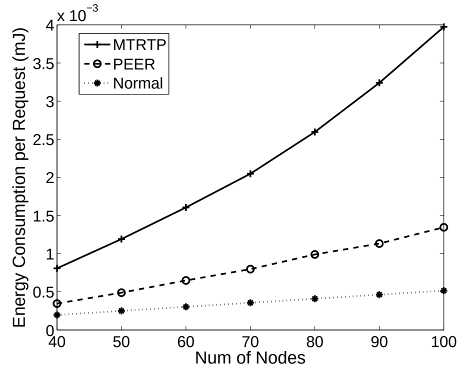

Fig. 8. Energy consumption for routing overhead.

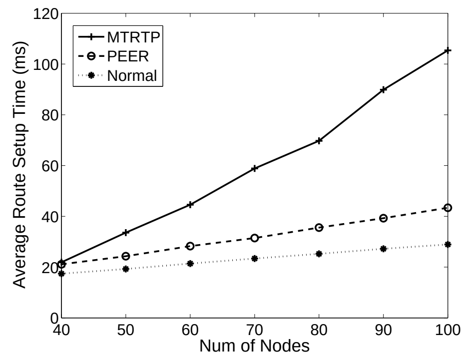

Fig. 9. Route setup time.

The PEER protocol is seen to have significantly better performance than MTRTP. Most importantly, instead of having a rapid increase in both routing overhead and route setup time as that of MTRTP, these performance metrics of PEER increase only linearly with the number of nodes in the network. So PEER has a higher scalability with the number of nodes. When the number of nodes reaches 100, PEER can reduce about 2/3 routing overhead, and hence the corresponding energy consumption and path setup delay.

C. Performance in Static Scenario

In static scenario, we compared the energy consumption and the average number of RTS retransmissions of the three protocols by varying the following parameters: node density, average packet size, and connection arrival rate. The simulation time for each protocol is 5 hours. We monitored the total energy consumption of all the packets received, the total number of packets received at all destination nodes, and the total number of RTS retransmissions for each simulation. The two metrics we used to evaluate the protocols are:

- *Energy Consumption per Packet:* It is defined by the total energy consumption divided by the total number of packets received. This metric reflects the energy efficiency for each protocol.

- *Average number of RTS Retransmissions per Data Packet:* It is defined by the total number of RTS retransmissions divided by the total number of packets received. The RTS packet is transmitted at the maximum power level and the packet size is very small. Most of RTS retransmissions are due to collisions, including the collisions of both RTS messages and data packets. Therefore, this metric can reflect the collision rate for each protocol. Higher collision rate will cause more energy consumption, higher end-to-end delay, and lower throughput.

The simulation results are shown in Fig. 10-15. For all three different groups of studies, PEER protocol performs the best in terms of *Energy Consumption per Packet* as well as *Average RTS Retransmission per Data Packet*, followed by MTRTP protocol and normal protocol.

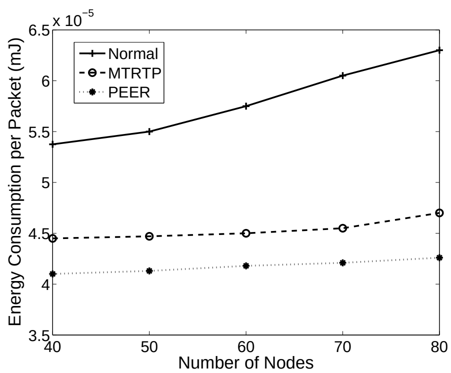

Fig. 10. Different density (static).

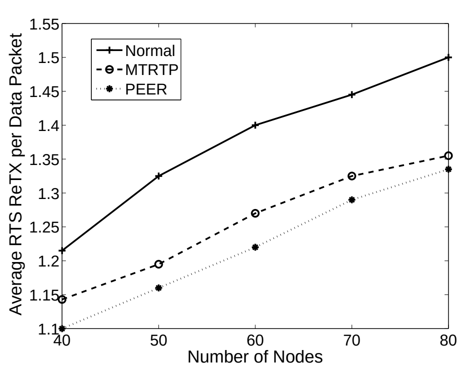

Fig. 11. Different density (static).

Both PEER and MTRTP protocols search for energy efficient path instead of shortest path in normal AODV

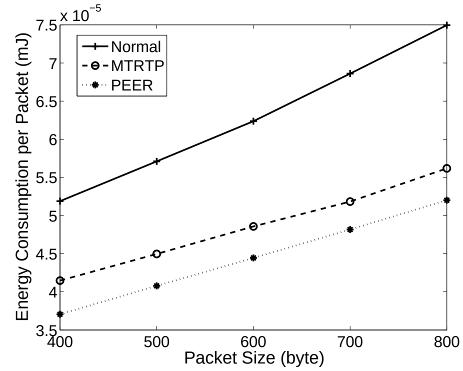

Fig. 12. Different packet size (static).

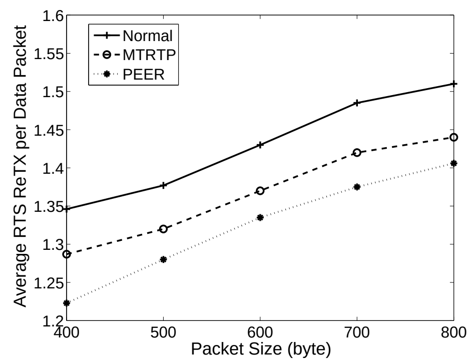

Fig. 13. Different packet size (static).

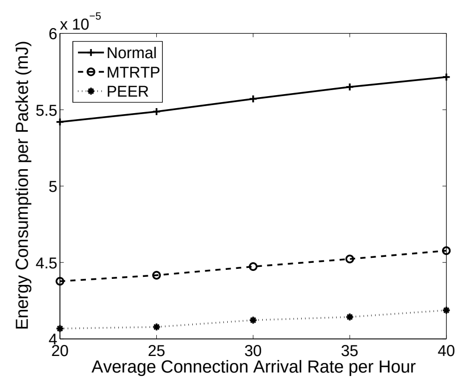

Fig. 14. Different connection arrival rate (static).

protocol, therefore they can perform better in terms of energy consumption. There are several reasons that PEER performs better than MTRTP in terms of energy consumption. First, PEER protocol uses a more accurate link cost, and can search for a more energy efficient path. Second, as the routing overhead in MTRTP is high, the path search request from the most energy efficient path has a higher probability of being lost by the intermediate nodes. The higher routing overhead also contributes to the higher energy consumption. Third, PEER protocol

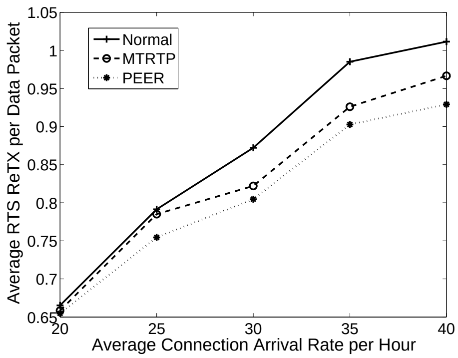

Fig. 15. Different connection arrival rate (static).

can adapt the path to the environment change quickly, and better maintain an energy efficient path.

The use of maximum transmission power for RTS/CTS will reduce hidden terminal problems and allow nodes within the transmission ranges of the sender and receiver to set NAV values correctly. However, with use of power control in all three protocols, when a node A reduces its sending power for data or ACK packets, it will also reduce the sensing range for other nodes to detect the transmission of the node. The nodes not being able to sense the transmission of node A could become hidden nodes when transmitting at a larger range (up to the maximum transmission range) and collide with the receiving at A. For the normal protocol, the range of data transmissions can vary from very small up to the transmission limit. The big difference in transmission ranges will cause serious hidden node problems and hence collisions. As the two energy efficient routing protocols try to use some shorter distance links, there will be fewer collisions due to a hidden node sending data packets. Although the sending of RTS/CTS packets at the maximum power from a hidden node could still collide with on-going transmission, as RTS/CTS is generally much smaller than data packets, the collision probability due to sending of RTS/CTS packets is much smaller than that of sending data packets. Therefore, the retransmission rate of normal protocol is higher than that of the energy efficient routing protocols. As the link cost of MTRTP does not include energy consumption due to signaling, which is significant due to its large signaling overhead as just demonstrated, MTRTP underestimates the link cost and tends to have a path with a larger number of hops. This will also increase the chance of RTS packets being lost and hence the retransmissions. Therefore, PEER protocol has the lowest RTS retransmission rate among all three protocols in all simulation scenarios.

It is interesting to observe that RTS retransmission rate increases with node density in Fig. 11 for all protocols, while the energy consumption per packet in Fig. 10 has no such a trend. In a high density network, more energy efficient path (with a larger number of nodes and shorter hop distance) could be found, which compensates the extra energy consumed due to a larger number of retransmissions.

D. Performance in Mobile Scenario

In mobile scenario, we also compared the same metrics as in static scenarios for all three protocols by varying the average node speed, packet size, and connection arrival rate. The simulation results are in Fig. 16- 21. For all three different groups of studies, PEER protocol performs the best in terms of *Energy Consumption per Packet* as well as *Average RTS Retransmission per Data Packet*.

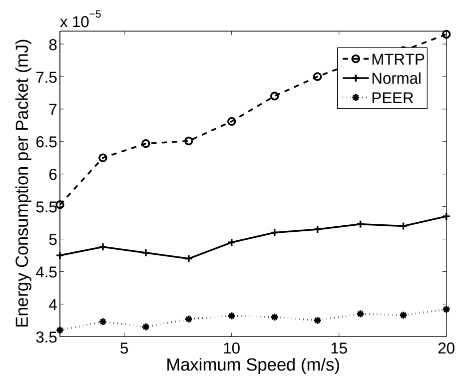

Fig. 16. Different speed (mobile).

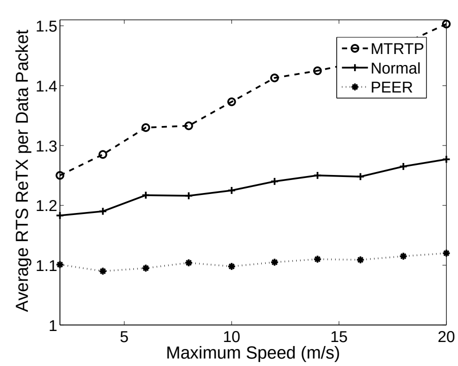

Fig. 17. Different speed (mobile).

As expected, MTRTP performs the worst in terms of energy consumption. As its path could not adapt well to the mobility, the minimum energy path found at the path setup time may no longer be energy efficient, and can

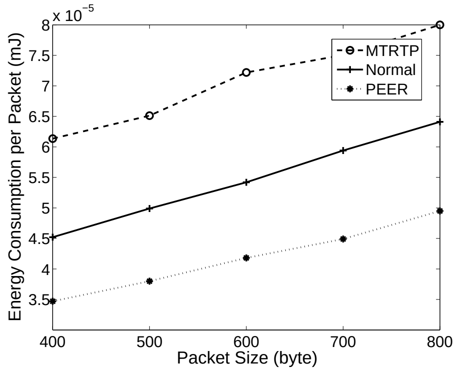

Fig. 18. Different packet size (mobile).

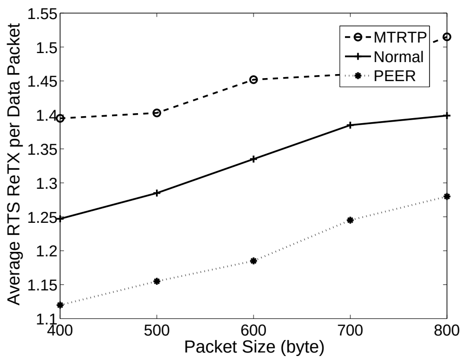

Fig. 19. Different packet size (mobile).

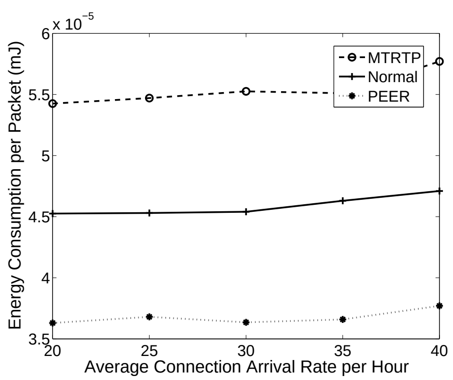

Fig. 20. Different connection arrival rate (mobile).

consume even more energy than normal routing protocol as its path normally has a larger number of hops. This is confirmed by the simulation results. MTRTP is seen to consume much more energy than the normal routing protocol. With an efficient path maintenance scheme, PEER can adapt its path with the mobility to maintain an energy efficient path all the time. Therefore, PEER performs much better than normal on-demand routing protocol and consumes significant less energy as compared to MTRTP in all test scenarios. Compared to the normal

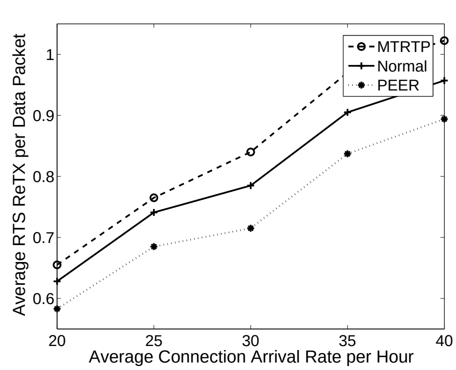

Fig. 21. Different connection arrival rate (mobile).

routing protocol (which is extended with power control capability as mentioned in the performance setup), PEER can reduce energy consumption up to 27%, 25%, and 25% respectively when varying the moving speed, packet size and average connection arrival rate. Compared to MTRTP, PEER can reduce energy consumption up to 51% 40%, 40% respectively.

As mentioned in the static scenario, the RTS retransmission is mainly caused by asymmetric power. Because of node mobility, the distance between two nodes in MTRTP can increase up to the transmission range, which will cause similar asymmetric power issue as in normal protocol. In addition, due to its use of larger number of hops, RTS retransmission rate in MTRTP is larger than in normal protocol. On the contrary, PEER protocol could adapt the path as nodes move, so it can maintain an energy efficient path in spite of node mobility. Therefore, it has a smaller number of RTS retransmissions, and consumes less energy than normal protocol as expected.

VI. Conclusion

It is important to design energy efficient routing protocols for mobile ad hoc networks. However, without a careful design, an energy efficient routing protocol could have much worse performance than a normal routing protocol. Specially, an energy efficient routing protocol could incur much higher control overhead and path setup delay as demonstrated by our simulations, and consume even more energy than a normal routing protocol in mobile environment.

In this paper, we first derived a new link cost model to more accurately track the energy consumption due to various factors. We then discussed the issues in path discovery and route maintenance associated with the minimum energy routing protocols. Based on these observations and our new link cost metric, we propose a progressive energy efficient routing (PEER) protocol with a quick and low overhead path discovery scheme and an efficient path maintenance scheme for reducing energy consumption especially in mobile environment.

Our performance studies show that PEER protocol reduces about 2/3 routing overhead and path setup delay as compared to a conventional energy efficient routing protocol, and is highly adaptive to the environment change. PEER performs much better than normal energy efficient protocol in both static scenario and mobile scenario, and under all circumstances in terms of node mobility, network density and load. In mobile scenarios, PEER can reduce transmission energy consumption up to 50% in all simulation cases compared to the conventional energy efficient routing protocol MTRTP.

REFERENCES

[1] K. Scott and N. Bambos, "Routing and Channel Assignment for Low Power Transmission in PCS", ICUPC '96 , Oct. 1996

[2] S. Doshi, S. Bhandare, and T. X Brown, "An On-demand Minimum Energy Routing Protocol for a Wireless Ad Hoc Network", ACM Mobile Computing and Communications Review, vol. 6, no. 3 , July 2002

[3] V. Rodoplu and T. Meng, "Minimum Energy Mobile Wireless Networks", IEEE Journal on Selected Areas on Communications, vol. 17, Aug. 1999.

[4] S. Banerjee and A. Misra, "Minimum Energy Paths for Reliable Communication in Multi-hop Wireless Networks", MOBIHOC'02, June. 2002.

[5] J. Gomez, A. T. Campbell, M. Naghshineh, and C. Bisdikian, "Conserving Transmission Power in Wireless Ad Hoc Networks", IEEE Conference on Network Protocols, Nov. 2001

[6] J. Zhu, C. Qiao and X. Wang, "A Comprehensive Minimum Energy Routing Protocol for Wireless Ad Hoc Networks", INFOCOM'04 , Mar. 2004

[7] C. K. Toh, H. Cobb and D. Scott, "Performance Evaluation of Battery-Life-Aware Routing Schemes for Wireless Ad Hoc Networks", ICC'01, June 2001

[8] A. Misra and S. Banerjee, "MRPC: Maximizing Network Lifetime for Reliable Routing in Wireless Environments", WCNC'02, Mar. 2002

[9] ANSI/IEEE Std 802.11, 1999 Edition

[10] E. Jung and N. H. Vaidya, "A Power Control MAC Protocol for Ad Hoc Networks", MOBICOM'02, Sept. 2002.

[11] G. Bianchi and I. Tinnirello, "Kalman Filter Estimation of the Number of Competing Terminals in an IEEE 802.11 network", INFOCOM'03, 2003

[12] C-K Toh, "Ad Hoc Mobile Wireless Networks Protocols and Systems", Prentice Hall, 2002

[13] T. H. Cormen, C. E. Leiserson and R. L. Rivest, "Introduction to Algorithms", MIT Press, 1998

[14] S. Gobriel, D. Moss and R. Melhem, "Mitigating the FloodingWaves Problem in Energy-Efficient Routing for MANETs", ICDCS'06, 2006

[15] C. E. Perkins and E. M. Royer, "Ad hoc On-Demand Distance Vector Routing", Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, February 1999

[16] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks", Mobile Computing, Kluwer Academic Publishers, 1996

[17] DS. PalChaudhuri and D. B. Johnson, "Power Mode Scheduling for Ad Hoc Networks", ICNP, 2002.

[18] W. Navidi and T. Camp, "Stationary Distributions for the RandomWaypoint Mobility Model", IEEE Trans. on Mobile Computing, vol. 3, no. 1, pp. 99- 108, 2004.

[19] J. Zhu and X. Wang, "PEER: A Progressive Energy Efficient Routing Protocol for Wireless Ad Hoc Networks", INFOCOM'05 , Mar. 2005

Jinhua Zhu received his Ph.D. degree in Computer Science and Engineering from University at Buffalo in 2007. He is currently a service engineer in Ericsson Inc. His research interests include wireless sensor networks and cognitive radio.

Xin Wang (M'01/ACM'04) received her Ph.D. degree in electrical and computer engineering from Columbia University, New York, NY.

She is currently an Assistant Professor in the Department of Electrical and Computer Engineering of the State University of New York at Stony Brook, Stony Brook, NY. Before joining Stony Brook, she was a Member of Technical Staff in Bell Labs Research, Lucent Technologies, New Jersey, and an Assistant Professor in the Department of Computer Science and Engineering of the State University of New York at Buffalo, Buffalo, NY. Her research interests include wireless networks and communications, mobile and distributed computing. Dr. Wang has served in numerous technical communities of top networking conferences.