International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 1 Issue 3, May - 2012

Minimum Energy Based Efficient Routing Protocol Over MANETs

D. Venkata Siva Prasad, *M. Tech Student*, and Mr. D. Sharath Babu Rao, *Faculty*
Department of Electronics & Communication Engineering
Jawaharlal Nehru Technological University
Anantapur, India

ABSTRACT

Minimum energy (Energy Efficient) routing protocols are very essential for wireless ad hoc networks which usually consist of mobile battery operated computing devices and many such protocols and schemes have been proposed so far. However, few efforts have been spent on issues associated with such protocols such as routing overhead and route setup time and route maintenance. The energy efficient routing protocols could fail without considering the mobility of node and routing overhead involved. In this paper, a more accurate analytical model is proposed to track the energy consumptions due to various factors and impact of packet errors. A simple energy efficient routing scheme called PEER is implemented that significantly improves the performance during path discovery phase and in mobility scenarios. The simulation results show that PEER protocol can reduce up to 2/3 routing overhead during path discovery phase and delay, and 50 percentage transmission energy consumption compared to conventional energy efficient routing protocol.

I. INTRODUCTION

In wireless ad hoc networks, mobile devices are often battery powered. But current battery technique still could not support the devices to work long enough. In addition, changing the battery may not be feasible in some application scenarios, such as sensor networks in hostile environment. Therefore, energy saving schemes are very important in wireless ad hoc networks. Since mobile devices are getting smaller and more energy efficient, communication energy cost becomes a much significant part in the total energy consumed. Therefore energy efficient communication scheme is one of the most effective ways to save energy.

In wireless network, the transmitted signa1 is attenuated at the rate of $1/d^n$, where d is the distance to the sender and n is the path loss exponent between 2 and 6. Then the basic energy efficient scheme would be to adjust the transmission power according to the distance between the sender and the receiver instead of using the constant maximum transmission power. This is called power control scheme. However, this is not optimal in terms of end-to-end energy consumption. To achieve the optimal solution, many energy efficient routing protocols have been proposed [1]-[8]. These protocols can be generally classified into two categories: *Minimum Energy* routing protocols[l]-[6] and *Maximizing Network Lifetime* routing protocols [7][8]. *Minimum Energy* routing protocols try to find the most energy efficient path to transmit the data packets from the source to the destination, while *Maximizing Network Lifetime* routing protocols try to balance the remaining battery power at each node .

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) protocols. MTTP protocols use the transmission power as the link metric and search for the path with minimum total transmission power between the source and the destination. MTTCP protocols use the transmission power as well as the receiving power as the link cost. MTRTP protocol uses the total transmission power for transmitting the data packets from one node to its neighboring node reliably as the link cost.

Most of previous work concentrated on the link costs. Once a new link cost was derived, then the traditional shortest path routing protocols, such as AODV, DSR, and Bellman-ford, can be modified with the new link cost. However, there are some problems with such straightforward modification. First, the routing overhead for the route discovery is very high, which consumes a lot of energy. Second, the route setup time is very long. Third, the route maintenance scheme is not suitable for dynamic environments, such as mobility scenarios.

To address these issues, we propose a *Progressive Energy Efficient Routing* (PEER) protocol. Contrary to other energy efficient routing protocols that try to find the optimal path at one shot and maintain the route reactively, PEER searches for the more energy efficient path progressively and maintains the route continuously. It first finds a path near the most energy efficient path between the source and the destination quickly, and then adjusts the nodes whenever necessary so that the path would be energy efficient all the time. Our performance evaluation shows that PEER achieves less routing overhead, shorter setup time, and great energy efficiency in static scenario as well as the mobile scenario.

II. OBSERVATION AND MOTIVATION

Many routing protocols have been proposed for wireless ad hoc networks. These protocols can be generally categorized as: (a) table-driven, (b) on-demand, and (c) hybrid. For table driven routing protocols, all nodes need to advertise the routing information periodically so that they can have the up-to-date view of the network. *Destination Sequenced Distance Vector* (DSDV), *Wireless Routing Protocol* (WRP), and *Cluster Switch Gateway Routing* (CSGR) belong to this category. Different from table-driven routing protocols, on-demand routing protocols create the route only when desired by the source node. Some on-demand routing protocols are *Ad hoc On-demand Distance Vector* (AODV). *Dynamic Source Routing* (DSR), and *Temporally Ordered Routing Algorithm* (TORA). The *Zone Routing Protocol* (ZRP) is a hybrid protocol with table-driven routing scheme for the intra-zone routing and on-demand routing scheme for the inter-zone routing. Most of energy efficient schemes modified the on-demand routing protocols such as AODV or DSR since there is a lot of routing overhead if using table-driven routing protocols . So, we will only focus on the on-demand energy efficient routing protocols.

Fig 1: A Linear Topology

For on-demand routing protocols such as AODV, a node will start a route discovery process if it needs a route 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 nodes will only rebroadcast the first route request packet received and discard the following duplicate ones. And the destination node only replies to the first route request packet, too. For example, in Fig 1, both A and B are neighboring nodes of S and D, and S needs a route to D. So S broadcasts the route request packet first, and both A and B receive the packet. Assume A broadcasts such packet next, then node S, B and D receive such packet, however node S and B will discard it as they have already received the same route request packet. Therefore the final route is SAD. It is apparent that the routing overhead for these protocols is $O(n)$, where n is the number of nodes in the network. Things are quite different for energy efficient routing protocols. The nodes could not simply discard the duplicate route request packets now as they may come from more energy efficient paths. That is, they also need to respond to the route request packets 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 , we can obtain that routing overhead for minimum energy efficient routing protocols is $O(n^2)$ now. Such overhead will consume a lot of energy and network resources, especially when the number of nodes in the network is very large.

In addition, the route setup time is much longer than the on-demand routing protocols. There are two main reasons for this. One is that the energy efficient route has more intermediate nodes than the shortest path in general, so it takes longer time for the route request and route reply packets to go through all the intermediate nodes. The other is that the energy efficient routing protocols have much more routing overhead which can cause more delay at each link. The simulations in GlomoSim verify our observation.

From the simulation results in Fig. 5-7, it is clear that the routing overhead, energy consumption for routing overhead, and route setup time for the energy efficient routing protocol increase dramatically with the number of nodes in the network, while only linearly for the on-demand routing protocol.

III. ENERGY CONSUMPTION MODEL FOR 802.11

Link cost is very important in energy efficient routing protocols. Without an accurate link cost the minimum energy routing protocols could not find the optimal route. In this section, we will first present some physical and MAC layer assumptions used in this paper. Then we propose an efficient way to estimate the link cost. PEER requires that each node can adjust the transmission power dynamically and retrieve channel information such as noise and received power level. Both are also common assumptions in most energy efficient routing protocols. In addition, it also desires that the MAC protocol can provide reliable hop-by-hop data transmission as retransmission costs a lot of energy. Therefore we use power control 802.11 for MAC protocol, in which RTS and CTS packets are transmitted at the maximum power while DATA and ACK packets are transmitted at the minimum required power level for the receiver to decode correctly. To avoid some collisions, PEER also requires the nodes to set their NAVs *(Network Allocation Vector)* to the EIF *(Extended InterFrame Space)* duration if they can sense the signal but can not decode it correctly [10]. We derived an accurate energy consumption model for 802.11 in [6]. Denote the packet sizes of RTS, CTS, DATA, and ACK packets by N_r, N_c, N_d, and N_a and 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 $p_{a,ij}$. In addition, for a variable x, denote 1-x by x^*, and the mean value of x by $\bar{x}$. Then the average total transmission power for transmitting a packet from node i to one of its neighboring node, node j , is

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

$$= \frac{P_m (N_r + N_c p^*_{r,i,j})}{p^*_{r,i,j} p^*_{c,j,i} p^*_{i,j} p^*_{a,j,i}} + \frac{N_8 P_{i,j} + N_a P_{j,i} p^*_{i,j}}{p^*_{i,j} p^*_{a,j,i}}$$

where P_m is the maximum power, $P_{i,j}$ and $P_{j,i}$ are the transmission power for DATA and ACK packets respectively. 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} and N_a = N_{ack} + N_{phy} where N_{phy} is the size of physical layer overhead. 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 as

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

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

Assume there are M-1intermediate 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

$$\overline{P_{total}} = \sum_{i=0}^{M-1} (\overline{P_T(i, i+1)} + \overline{P_T(i, i+1)})$$

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 $i+1$.

Most of parameters in this model can be easily obtained except the transmission power and the packet error rates. PEER adopts the transmission power estimation scheme used in [10]. If node A receives a packet transmitted at the maximum power level from node B, such as RTS, CTS and broadcast packets, then node A can calculate the desired transmission power to node B, $P_{desired}$, based on the received power, P_r, and the maximum power level (P_m) as, $P_{desired} = \frac{P_m}{P_r} * P_{rthresh} * c$, where $P_{rthresh}$ is the minimum necessary received signal strength and c is an constant.

IV. PEER PROTOCOL

As a routing protocol, PEER also consists of route discovery process and route maintenance scheme.

A. Route Discovery Process

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 (2hops) paths (SAD, SBD, SED, SFD, SGD, SHD). Among all the shortest paths, it is better to pick the most energy-efficient one. 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

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

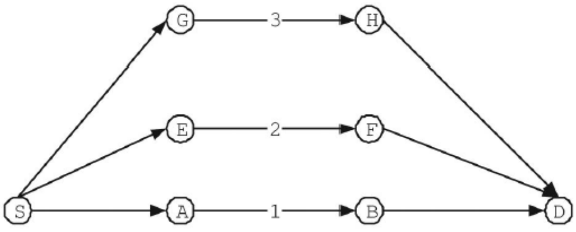


Fig 2: The routes between S and D

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.

However, the destination node D has no such information so that it could not pick the minimum energy shortest path even if it already receives all route request packets from all shortest paths. There are several ways to deal with this issue at the destination node. One option is that the destination sends a route reply packet for each route request packet it receives. This method will waste some energy as the destination will send out many route reply messages and the source node might transmit some data packets on less energy efficient path. The other one is that the destination sets up a timer after receiving first route request packet. If it receives another route request packet before timeout, it will reset the timer. Otherwise, it will select the best path so far and reply with a route reply packet when the timer goes off. This method help reduce the energy consumption, but it may increase the route setup time. In this paper, we use the second one. The minimum energy shortest path may still not be energy efficient enough since it tends to use the long-distance link. Allowing a route to pass through some intermediate nodes may help to save energy. To speed up the route optimization process, this can be done in parallel as the route reply message travels from the destination to the source. When the nodes that are not on the minimum energy shortest path overhear such route reply message, they will check whether they are on a lower energy path between the sender and the receiver.

B. Route Maintenance

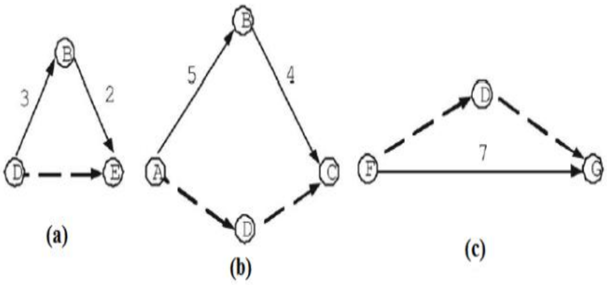
As described in section III, each node can estimate the necessary transmission power and the link cost to one of its neighboring node once it receives RTS, CTS or broadcast packet from such node. PEER requires that each node adds the link cost to the receiver in the IP header as an IP option for each data packet it transmits, and monitors the data packets transmitted in its neighborhood. For each data packet transmitted, received, or overheard by the 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 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: Pl(S1, D1,1), P2(S2, D2,3) and P3(S3, D3,5). The first packet (Pl) uses two-hop path (A→B→C) in D's neighborhood and the total link cost is 9(5+4). The second packet (P2) uses another two-hop path (D→B→E) and the total link cost is 5(3+2). The third packet (P3) uses one-hop path (F→G) and the link cost is 7. Based on the information in the link cost table, each node can help improve the local path as well as its corresponding end-to-end path with the three operations (Remove, Replace, and Insert) illustrated for node D in Fig.3

TABLE 1
A LINK ENERGY TABLE

(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



(a) (b) (c)

Fig 3: Remove, replace and insert

(a) Remove

The rule for Remove operation is as follows:

Assume there is a two-hop path X → A → B with destination D and total link cost *T* in X's link cost table. *If X finds the link cost between X and B is smaller than that of the two-hop path, it will update its routing table by setting the next hop for destination D to B.*

In Fig.3(a), node D has the two-hop path info (D→B→E) from its link energy table with destination D2 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 following packet for destination D2 will go through E directly.

(b) Replace

The rule for Replace operation is as follows:

Assume that there is a two-hop path A→B→C with destination D and total link cost *T* in X's link cost table. *If X finds the total cost for the path A→X→ C is smaller than that of the two-hop path A→B→ C, X will update its routing table by setting the next hop to destination D to C. In addition, it will request A to update A's routing table by setting the next hop to the destination D to itself (X).*

In Fig. 3(b), Node D has the two-hop path info (A→B→C) in its link cost table with the destination D1 and the total link cost (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, then the path A→D→C is more energy efficient than A→B→C. So node D will update its routing table by setting the next hop to destination D1 to C and request A to update A's routing table by setting the next hop to destination D1 to D. If A accepts the request from D, then the following packets for D1 at node A will be transmitted to node D and D will forward them to C. If A does not accept the request from D, the routing info for destination Dl at node D will be purged after some time.

(c) Insert

The rule for Insert operation is as follows:

Assume that there is a one-hop path A → B with destination D and total link cost *T* in X's link cost table. *If X finds the total cost for the path A→X→ B is smaller than that of one-hop path, it will update its routing table by setting the next hop to destination D to B. In addition, X will request A to update A's routing table by setting the next hop to the destination D to itself (X).*

In Fig. 3(c), Node D has the one-hop path info (F→G) in its link cost table with the destination D3 and the total link cost (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 F→D→G is more energy efficient than F→G. So node D will update its routing table by setting the next hop to destination D3 to G and request F to update F's routing table by setting the next hop to destination D3 to D.

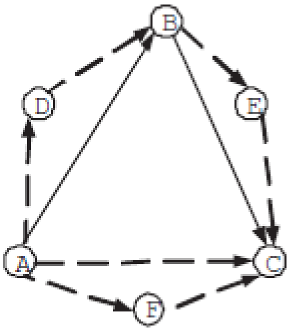


Fig 4: An undesired improvement

Only Replace and Insert operations need the control message. The control messages are only sent out when a better path is noticed so that the maintenance overhead is very low. The control message includes: operation ID, requester ID, destination, next hop, the total link cost for new path.

The control message that D sends to A for Replace operation is [Replace, D, D1, B, the total link cost for ADC]. While the control message that D sends to F for Insert operation is [Insert, D, D3, G, the total link cost for FDG]. Once a node receives a control message, it will first check the routing info 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 been changed. Within these three operations, Insert may have higher priority than the other two since it only needs to check one-hop transmission. This may not be desirable. For example, in Fig 4, node A transmits the data packet to node B. D overhears such data packet so that it sends a packet to A indicating that it can save energy between the link AB. 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. So it would be better to let Remove and Replace 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 has Insert and any other operation request, it will take the other operation. If it has both Remove and Replace operation requests, it will select one by the energy saving percentage. For the same example, node A has the Insert (by node D), Remove, and Replace (by node F) requests, then it will only process Remove and Replace operations. And as AFC is better than AC $(\overline{P_T}(A,F)+\overline{P_T}(F,C)<\overline{P_T}(A,C)$. so it takes the Replace operation.

V. PERFORMANCE EVALUATION

We have simulated PEER, MTRTP, as well as normal AODV protocols in Glomosim. We modified AODV with the new link cost derived in [4] for MTRTP protocol. And the power control scheme is also applied to the normal AODV protocol. The network area is 1200(m)×1200(m.) and the nodes are randomly distributed over the network. The available transmission power levels are 1, 5, 10, 15, 20, 25, 30, 35 mW. The connection arrival rate follows Poison distribution and the connection duration follows Exponential distribution. The application protocol is CBR (Constant Bit Rate) and the source and destination pairs are randomly selected. The mobility model is random waypoint with 30-second pause time. Some other default setup parameters are in Table II.

We first studied the route discovery performance for each protocol, and then the energy consumption as well as the retransmission rate in static as well as the mobile scenarios.

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.5-7.

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

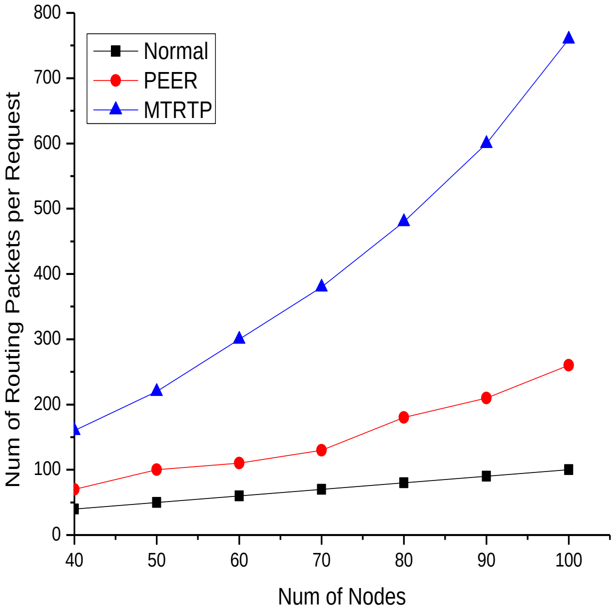


Fig 5: Routing overhead

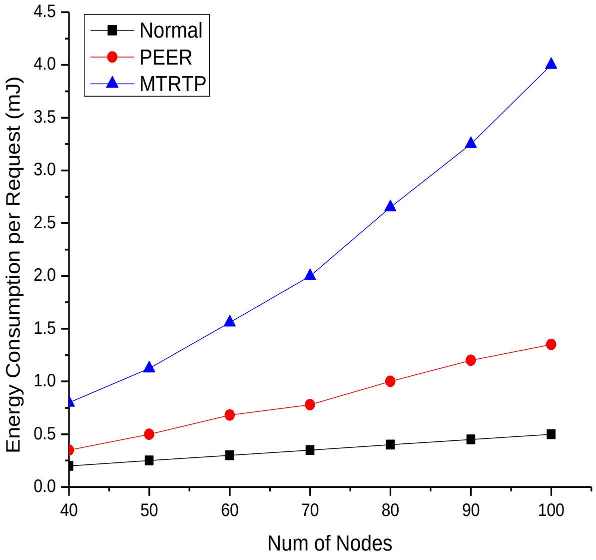


Fig 6: Energy Consumption for Routing overhead

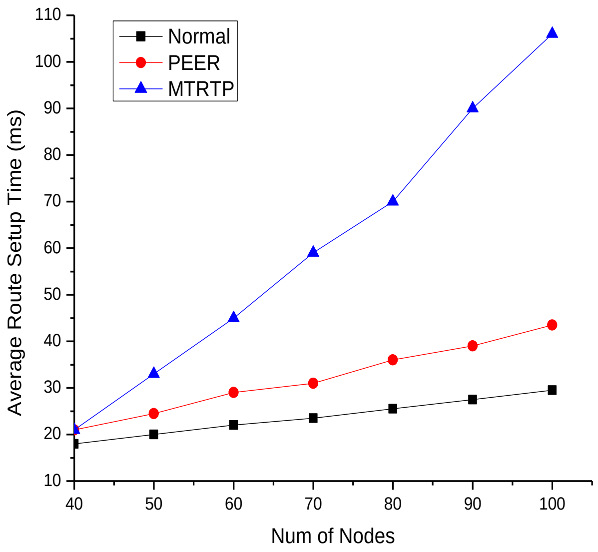


Fig 7: Route setup time

It is clear from the results that the normal on-demand routing protocol performs the best in terms of routing overhead, energy consumption for routing overhead, and setup time, followed by PEER and minimum energy routing protocol. Both the routing overhead and setup time for the minimum energy routing protocol are much more than the on-demand routing protocol, and increase dramatically with the number of nodes, That is because the routing overhead for minimum energy routing protocol is $O(n^{2})$ *(n* is the number of nodes) as discussed in Section Ⅱ. Therefore the minimum energy routing protocol could not scale well with the number of nodes. While for PEER protocol, the performance is quite well. Even though both the routing overhead and route setup time are still higher than the on-demand routing protocol, they are much less than the minimum energy routing protocol. Most importantly, both routing overhead and route setup time increase very close to linearly with the number of nodes in the network. So PEER has high scalability with the number of nodes.

Static Scenario

In the static scenario, we studied the energy consumption and RTS retransmission rate performance for each protocol in three different groups: different density, different packet size, and different connection arrival rate. The simulation time for each protocol is 5 hours. We monitored the total energy consumption, the total number of packets received at all destination nodes, and the total number of RTS retransmission for each simulation. The two metrics we used to evaluate the protocols are:

Energy Consumption per Picket: 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 RTS Retransmission per Data Packet: It is defined by the total number of RTS retransmission divided by the total number of packets received. As the RTS packet is transmitted at the maximum power level and the packet size is very small, most of RTS retransmission is because of collision. 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.

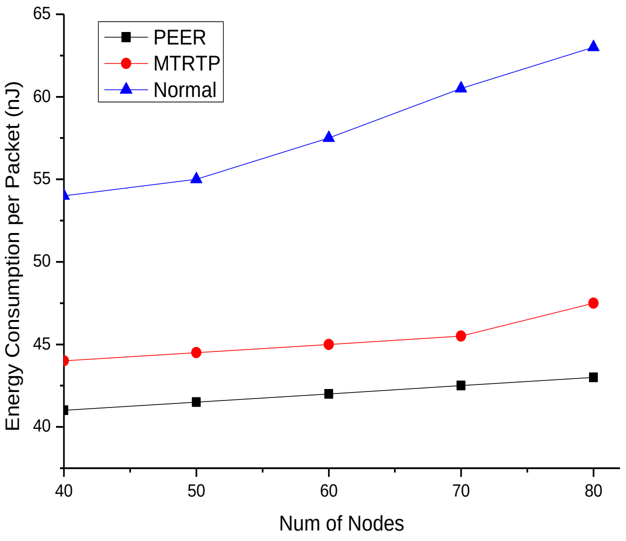


Fig 8: Different density (static)

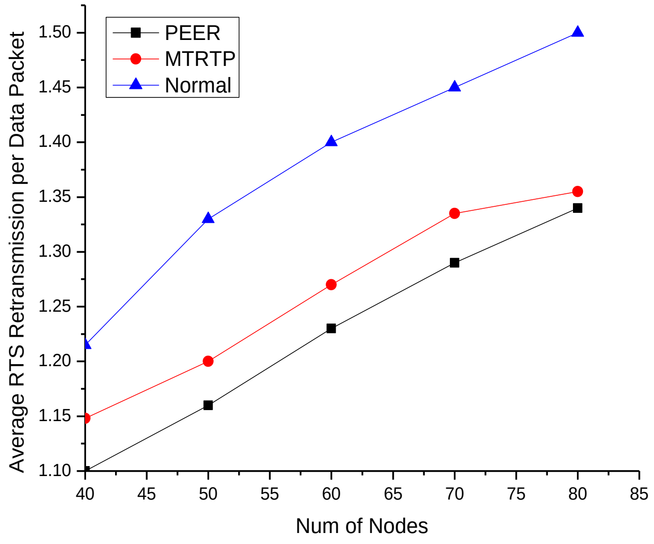


Fig 9: Different density

The simulation results are in Fig. 8-13. 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.

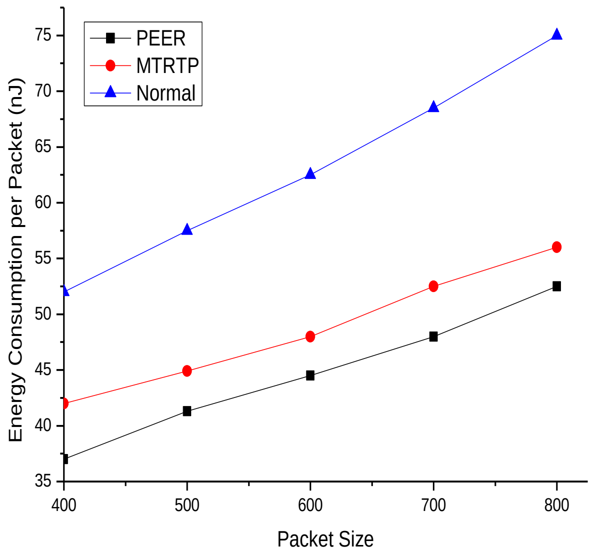


Fig 10: Different packet size (static)

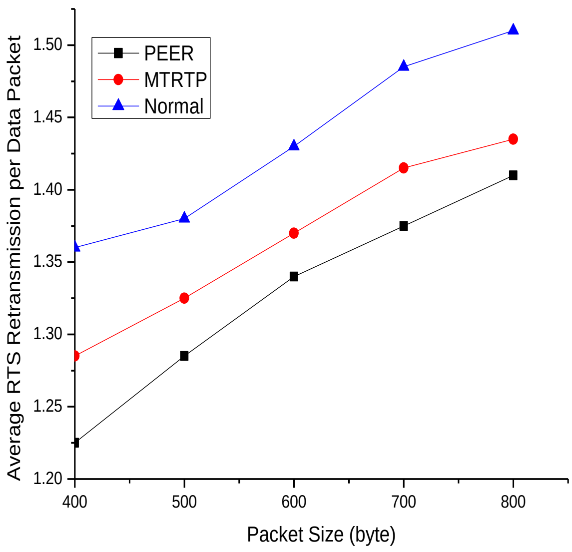


Fig 11: Packet size (static)

Both PEER and MTRTP protocol search for energy efficient path instead of shortest path in normal protocol so that they can perform better in terms of energy consumption. PEER performs better than MTRTP in terms of energy consumption. There are several reasons for that. First, PEER protocol uses a more accurate link cost. Second, there is a lot of routing overhead in MTRTP that the route request packet from the most energy efficient path has higher probability of being lost in some intermediate node, Third, PEER protocol can adapt the path with the environment change quickly. With power control scheme in all three protocols, RTS retransmission is mainly caused by asymmetric power. For normal protocol, the distance on each link can be quite different, ranging from very small up to the transmission range. While the two energy efficient routing protocols try to use some short distance links. Therefore, the retransmission rate is higher for normal protocol than the energy efficient routing protocols. As the link cost for MTRTP underestimates the real energy consumption, it tends to use larger number of hops. This will also increase the chance of RTS packets being lost and hence the retransmissions. So PEER protocol performs the best in terms of RTS retransmission rate. It is interesting to observe that the RTS retransmission rate increases with the density in Fig. 9 for all protocols, while the energy consumption per packet in Fig. 8 has no such trend. This is because even though higher retransmission rate can cause more energy consumption, it can be compensated by the more energy efficient paths found by the routing protocols with higher number of nodes.

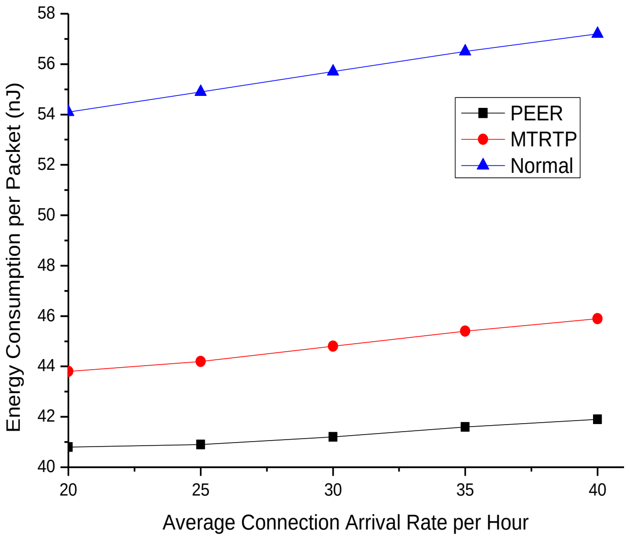


Fig 12: Different connection arrival rate (static)

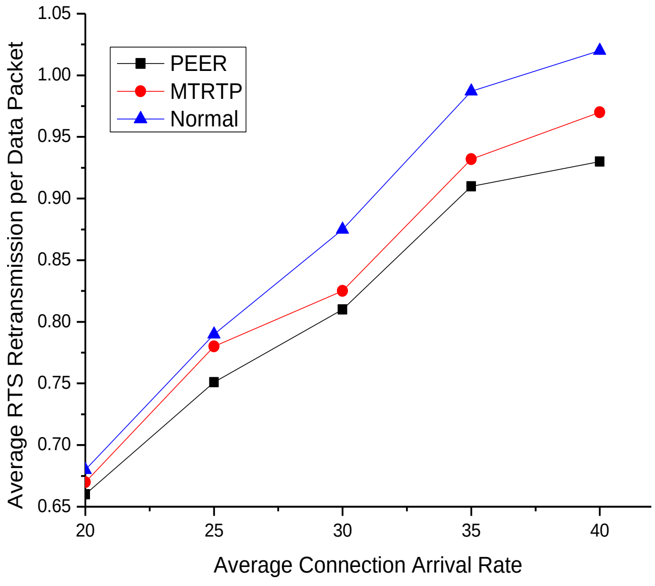


Fig 13: Different connection arrival rate (static)

Mobile Scenario

For mobile scenario, we also studied the same metrics as in static scenarios for each protocol. And the three groups of simulations are different speed, different packet size, and different connection rate. The simulation results are in Fig. 14- 19. 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*.

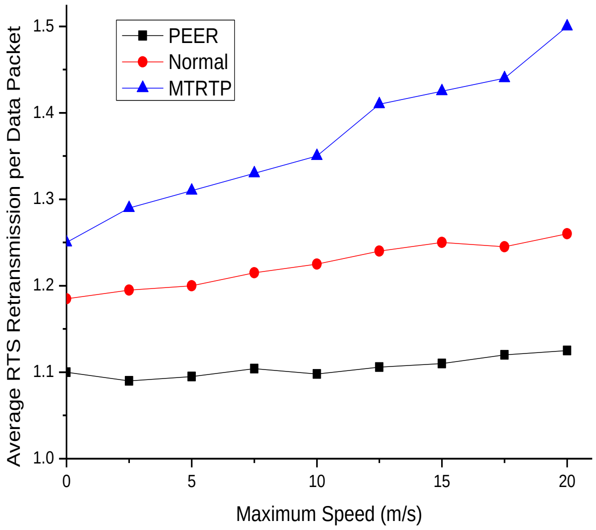


Fig 14: Different speed (mobile)

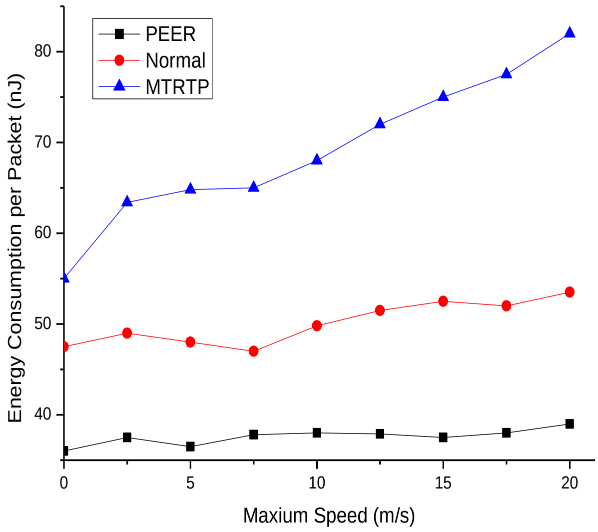


Fig 15: Different speed (mobile)

MTRTP performs the worst in terms of energy consumption, as its route maintenance scheme could not adapt with the mobility well. So the original minimum energy path would not be energy efficient any more because of node mobility. MTRTP even consumes much more energy than normal protocol as its path normally has more hops. As PEER adapts the path with the mobility, it could get an energy efficient path all the time. Therefore, it performs much better than normal protocol and consumes several times lower energy as compared to MTRTP.

As mentioned in static scenarios, the RTS retransmission is mainly caused by asymmetric power. Because of node mobility, MTRTP will have similar asymmetric power issue as normal protocol now. In addition, due to larger number of hops, the RTS retransmission rate is larger for MTRTP than normal protocol. Again, because PEER protocol could adapt the path with the mobility, it still tries to use some short distance link in spite of node mobility. So it performs better than normal protocol.

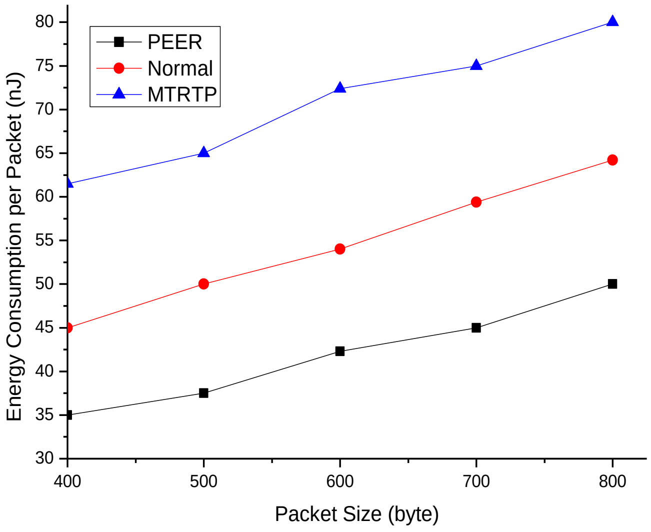


Fig 16: Different packet size (mobile)

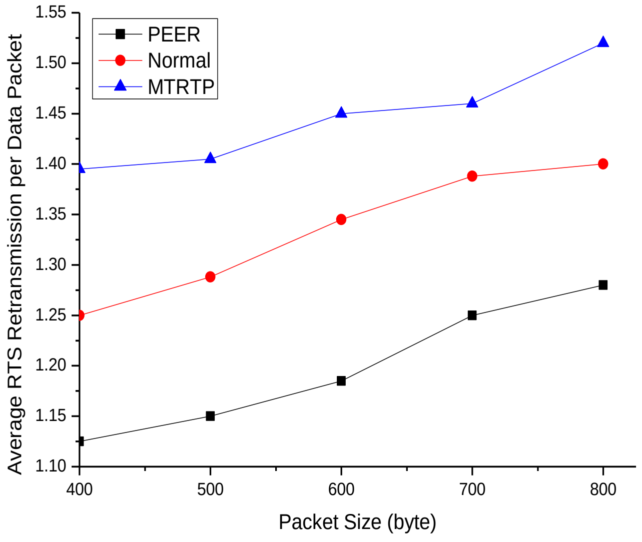


Fig 17: Different packet size (mobile)

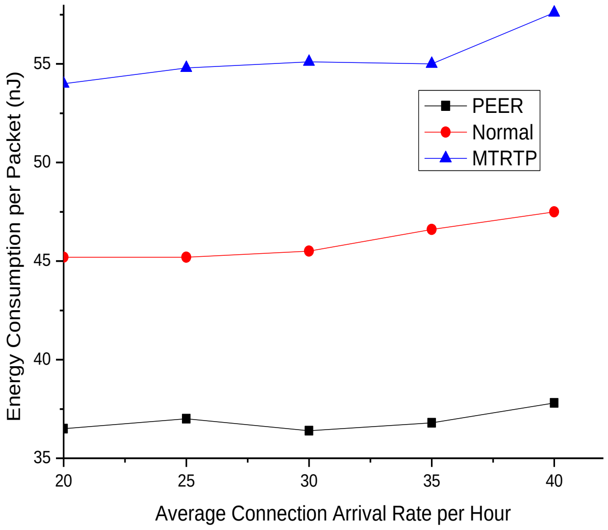


Fig 18: Different connection arrival rate (mobile)

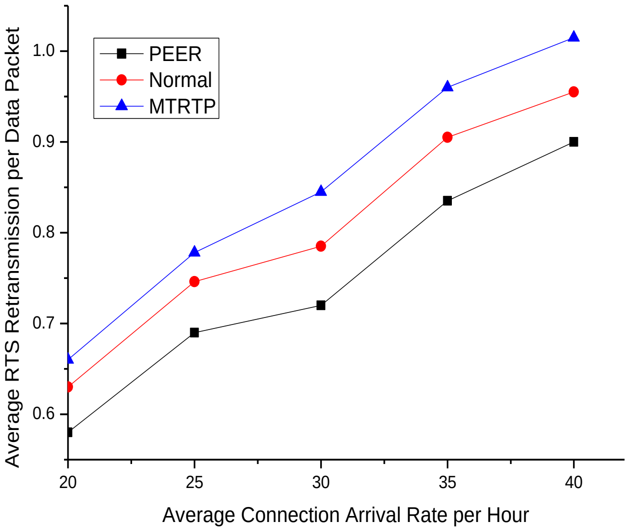


Fig 19: Different connection arrival rate (mobile)

VI. CONCLUSION

It is important to design energy efficient routing protocols for mobile ad hoc networks. 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. 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 reduces about 2/3 routing overhead and path setup delay and reduce transmission energy consumption up to 50% in all simulation cases compared to the conventional energy efficient routing protocol MTRTP.

ACKNOWLEDGMENT

Siva Prasad would like to thank Mr. D. Sharath Babu Rao, who had been guiding through out to complete the work successfully, and would also like to thank the HOD, ECE Department and other Professors for extending their help & support in giving technical ideas about the paper and motivating to complete the work effectively & successfully.

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 Ondemand 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 Lifetimefor 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.