Energy-Efficient Reliable Paths for On-Demand Routing Protocols

Tamer Nadeem, Suman Banerjee, Archan Misra, Ashok Agrawala [*]

CS-TR-4582 and UMIACS-TR-2004-25

Abstract

We define techniques to compute energy-efficient reliable paths within the framework of on-demand routing protocols. The choice of energy-efficient reliable paths depend on link error rates on different wireless links, which in turn depend on channel noise. We show how our scheme accounts for such channel characteristics in computing such paths. Additionally, we perform a detailed study of the AODV protocol and our energy-efficient variants, under various noise and node mobility conditions. Our results show that even in low noise environments there are specific scenarios where an unmodified on-demand protocol will achieve significantly lower throughput in comparison to our modified variants. Our results show that our proposed variants of on-demand routing protocols can achieve orders of magnitude improvement in energy-efficiency of reliable data paths.

1 Introduction

Battery-power is typically a scarce and expensive resource in wireless devices. Therefore, a large body of work has addressed energy-efficient link-layer forwarding techniques [1, 2, 3, 4, 5] and routing mechanisms [6, 7, 8, 9, 10, 11] for multi-hop wireless networks. However, an end-to-end reliability requirement can significantly affect the choice of data paths in both these objectives. In particular, the choice of energy-efficient routes should take into account the channel noise in the vicinity of these nodes. Such noise would lead to transmission errors and consequent re-transmissions, thus increasing the energy costs for reliable data delivery.

In this paper we describe how such minimum energy end-to-end reliable paths can be calculated *and implemented* for reactive (on-demand) routing protocols. We have experimented with the Ad-hoc On-demand Distance Vector Routing protocol (AODV) [12]. It should, however, become obvious from our description that our technique can be generalized to alternative on-demand routing protocols (e.g., DSR [13]

[*] T. Nadeem A. Agrawala are with the Dept. of Computer Science, University of Maryland, College Park, MD 20742, USA. Emails: {nadeem,suman,agrawala}@cs.umd.edu. S. Banerjee is currently with the Dept. of Computer Science, Wisconsin University, . Email: suman@cs.wisc.edu. A. Misra is with IBM T.J. Watson Research Center, Hawthorne, NY 10532, USA. Email: archan@us.ibm.com.

and TORA [14]). Through our experimentation, we perform a detailed study of the AODV protocol and our energy-efficient variants, under various noise and node mobility conditions. As part of this study, We have identified some specific configurations where an on-demand protocol that does not consider noise characteristics can result in significantly lower throughput, even under conditions of low or moderate channel noise.

The rest of the paper is structured as follows: in the next section we discuss related work in this area. Section 3 presents background and overview of our formulation of the minimum energy reliable path computation problem. In Section 4 we describe the link error rate estimation techniques, which forms an important component of the solution. In Section 5 we first briefly describe the AODV protocol, and then describe the necessary modifications to adapt it for minimum energy reliable path computations. In Section 6 we present detailed simulation experiments to evaluate the performance of the protocols and finally we present our conclusions in Section 7.

2 Related Work

A large number of researchers have addressed the energy-efficient data transfer problem in the context of multi-hop wireless networks. They can be classified into two distinct categories. One group focuses on protocols for minimizing the energy requirements over end-to-end paths. Typical solutions in this approach have ignored the retransmission costs of packets and have therefore chosen paths with a large number of small hops. PAMAS [6] is one such variable energy MAC protocol, where the nodes modify their transmission power based on the distance to the receiver, and where this variable transmission energy is used as the link cost to effectively compute minimum energy routes.A link cost that includes the receiver power as well is presented in [5]. By using a modified form of the Bellman-Ford algorithm, this approach resulted in the selection of paths with smaller number of hops than PAMAS.

An alternative approach focuses on algorithms for increasing the lifetime of wireless nodes, by attempting to distribute the forwarding load over multiple paths. This distribution is performed by either intelligently reducing the set of nodes needed to perform forwarding duties, thereby allowing a subset of nodes to sleep over different durations (e.g, SPAN [10] and GAF [11]), or by using heuristics that consider the residual battery power at different nodes [9, 8, 15] and route around nodes nearing battery exhaustion.

In [16] we formulated the link cost considering the error rates. We showed how can we approximate those costs to fit the proactive routing protocols that use Bellman-Ford algorithm in calculating the best paths. However, in this paper: 1) We study two different mechanisms for bit error rates estimation in practice, 2) We define and implement the modifications needed to compute energy efficient routes in the AODV as an example of the reactive protocols, and 3) We extensively study the performance benefits of our proposed schemes under various wireless environment conditions.

3 Minimum Energy Reliable Paths

To compute minimum energy reliable paths, we need to evaluate candidate paths not merely based on the energy spent in a single transmission attempt across the wireless hops, but rather on the total energy required for packet delivery, *including potential retransmissions due to errors and losses* on the wireless link. Such a formulation is especially relevant in multi-hop wireless networks, where variable channel conditions often cause packet error rates as high as $15 - 25\%$.

For any particular link (l) between a transmitting node and a receiving node, let Pt denote the transmission power and p_l represent the packet error probability. Assuming that all packets are of a constant size, the energy involved in each transmission attempt across of a packet, E_l, is simply a function of Pt.

Any signal transmitted over a wireless medium experiences two different effects: attenuation due to the medium, and interference with ambient noise at the receiver. Due to the characteristics of the wireless medium, the transmitted signal suffers an attenuation proportional to a function of D the distance between the receiver and the transmitter. The ambient noise at the receiver is independent of the distance between the source and distance, and depends purely on the operating conditions at the receiver. We consider two scenarios:

1. Fixed Transmission Power: in which each node chooses the transmission power to be a fixed constant and independent of the link distance. While such a choice is inefficient, most current wireless cards do not provide any mechanism for adaptively choosing the transmission power for each packet. Therefore, in this case the energy required to transmit packets (E_l) is independent of the distance (D).

2. Variable Transmission Power: where a transmitter node adjusts Pt to ensure that the strength of the (attenuated) signal received by the receiver is a constant (independent of D) and is minimally above a certain threshold level, P_{Th}. Since P_{Th} is typically a technology-specific constant, we can see that the transmission energy (E_l) needed for communication over such a link is directly related to D.

We consider two different operating models for end-to-end reliable paths:

3.1 Hop-by-Hop Retransmissions (HHR):

Each individual link provides reliable forwarding to the next hop using link-layer retransmissions. The *mean* number of individual packet transmissions for the successful transfer of a single packet is $1/(1 - p_l)$. Therefore, the mean energy cost, C_l, required for the successful transmission of this packet is:

$$C_l = \frac{E_l}{1 - p_l} \tag{1}$$

3.2 End-to-End Retransmissions (EER):

The individual links do not provide link-layer retransmissions and error recovery. Reliable packet transfer is achieved only via retransmissions initiated by the source node.

The actual end-to-end cost energy, C, requirements for a given path with nodes $0, \ldots, n$ in sequence for the EER case is given by:

$$C = \frac{\sum_{i=0}^{n-1} E_{i,i+1}}{\prod_{i=0}^{n-1}(1 - p_{i,i+1})} \tag{2}$$

where, $E_{i,i+1}$ is the energy required for a single transmission across the link $\langle i, i+1 \rangle$ and $p_{i,i+1}$ is the packet error probability of the link.

The above formulas assume unlimited number of retransmission trials for each packet. In practice, wireless cards limit the retransmissions number for HHR case, where as protocols such as UDP don't provide packet reliability. However, those formulas are still valid because: 1) many wireless cards use large number of retransmission trials in case of HHR, and 2) we consider connections in which the sources have infinite or very large number of packets. Also, we do not consider the cost of the control packets, e.g., RTS/CTS/ACK frames of IEEE 802.11, since the cost of the data packets dominates other costs.

4 Estimating Link Error Rate

In order to implement our proposed mechanism, it is sufficient for each node to estimate only the bit error rate, BER, on its incoming wireless links from its neighboring nodes. In this section we discuss the following two possible mechanisms:

4.1 BER using Radio Signal-to-Noise Ratio

The bit error rate, BER, of a wireless channel depends on the received power level, Pr, of the signal. The exact relationship between BER and Pr depends on the choice of the signal modulation scheme. However, in general, several modulation schemes exhibit the generic relationshipbetween BER and Pr:

$$\textsc{Ber} \propto \textit{erfc}(\sqrt{\frac{constant \times Pr}{N}}) \tag{3}$$

where N is the noise density (noise power per Hz) and *erfc*(x) is defined as the complementary function of *erf*(x) and is given by:

$$\textit{erfc}(x) = 1 - (2/\sqrt{\pi}) \int_0^x \exp^{-t^2} dt \tag{4}$$

For the case of BPSK (binary phase-shift keying), which we use to model the error rates in this paper, the bit error is given by

$$\textsc{Ber} = 0.5\, \textit{erfc}(\sqrt{\frac{Pr}{N \times f}}) \tag{5}$$

where f is the transmission bit rate.

Most wireless interface cards typically measure the Signal-to-noise ratio (SNR) for each received packet. SNR is a measure of the received signal strength relative to the background noise and is often expressed in decibels as:

$$SNR = 10 \log \frac{Pr}{N} \tag{6}$$

From the SNR value measured by the wireless interface card we can calculate the ratio Pr/N in Equation 5 for each received packet.

This SNR-based error rate estimation technique is useful specially primarily in free space environments where such error models are applicable. Consequently this is not applicable for indoor environments, where signal path characteristics depend more on the location and properties of physical obstacles on the signal paths. For such environments we use an alternative technique that is based on empirical observations of link error characteristics, which we describe next.

4.2 BER using Link Layer Probes

In this empirical mechanism, we estimate the BER of the incoming links by using link layer probe packets. Each node periodically broadcasts a probe packet within its local neighborhood. Each such packet has a local sequence number which is incremented with each broadcast. Each neighbor of this node receives only a subset of those probes due to channel errors. Each node stores the sequence number of the last correctly received probe from each of its neighbors. On the reception of the next (i^{th}) probe from a node, the receiving node can calculate s_i, the number of probes lost since the last received probe. The packet error rate (p) for a probe packet of length $prob\_size$ bits (assuming independent bit errors) is:

$$p = 1 - (1 - \text{BER})^{prob\_size} \tag{7}$$

Given the packet error rate for probes since the last probe reception as $p = \frac{s_i}{s_i+1}$, the receiving node can compute the incoming link BER as:

$$\text{BER} = 1 - \exp\left(\frac{\log(1 - \frac{s_i}{s_i+1})}{prob\_size}\right) \tag{8}$$

In wireless environments, broadcast packets are more prone to losses due to collision than unicast packets that use channel contention mechanisms, e.g. the RTS/CTS technique employed in IEEE 802.11. Therefore our probe-based BER estimation technique can potentially over-estimate the actual BER experienced by the data packets. However, the probe-based mechanism is still applicable for these reasons:

1. BER is over-estimated in parts of the wireless network with high traffic load. Since our route computation technique is biased against high BER, routes will naturally avoid these areas of high traffic load. This will lead to an even distribution of traffic load in the network, increasing network longevity and decreasing contention.

2. The criteria of selecting optimum route in our algorithm is based on the relative relations between costs of the candidate routes not the actual costs. As the traffic load on the network gets evenly balanced the different links, the BER will

be equally over-estimated for all the links. This implies that the costs relative ordering of the candidate routes is largely unaffected.

Generally, Pr and N in Equation 5 vary with time: N varies due to the environment conditions, and Pr, which changes with distance, varies due to the nodes mobility. Consequently, we can not base the SNR-based BER estimation on a single measurement. Therefore, SNR-based mechanism uses probes packets to calculate those parameters as function of several measurements over a window of time, in order to capture the dynamics of the network. For both the SNR-based and probe-based schemes, each node broadcasts probe packet, at an average period t (one second in the implementation). To avoid accidental synchronization and consequently collisions, t is jittered by up to $\pm 0.25t$. Each node continuously updates its estimate of the BER using an exponentially weighted moving average of the sampled BER values. As in all such averaging techniques, the estimate can be biased towards newer samples depending on the rate at which the noise conditions changes, i.e. increasing node mobility.

4.3 BER Estimation for Variable Power Case

For the fixed transmission power case, both probe and data packets are transmitted with the same constant power. Therefore, the BER experienced by data packets is the same as probe packets However, the same is not true for the variable power transmission case. In the variable power case, the transmission power used for a given data packet depends on the link's length. However, probe packets are broadcasted to all its possible neighbors and is transmitted with the fixed maximum transmission power (Pt_{max}). Equation 5 implies that packets received at a higher power (e.g. probe packets) will experience lower BER than the packets received at lower power (e.g. data packets). Therefore a suitable adjustment is required for BER estimation for data packets.

Since a node, in the variable power case, chooses the transmission power level such that the power level of the received data packet at the receiver is P_{Th}. Since the noise N can be calculated using the SNR and Pr values measured by the wireless interface card, we can estimate the BER for data packets using the SNR-based technique by substituting P_{Th} for Pr in Equation 5. Hence, the estimated BER for data packets on an incoming link is calculated as:

$$\text{BER} = 0.5 \, \mathit{erfc}(\sqrt{\frac{P_{Th}}{Nf}}) \tag{9}$$

A related but different correction scheme is applied for the BER estimation of data packets when using the probe-based technique. For the sake of simplicity, we approximate the relation between the received power ($Pr >> 0$) and BER, using a data fitting model, as:

$$\text{BER} = exp^{-b.Pr-1} \tag{10}$$

where b depends on the noise level at the receiver nodes. Using this model, we can estimate the BER experienced by data packets from the BER_P estimated using probes as follows:

$$\text{BER} = \text{BER}_P^{\frac{P_{Th}}{Pr_P}} * exp^{\frac{P_{Th}}{Pr_P}-1} \tag{11}$$

where Pr_P is the average signal power level of the received probes at the receiver nodes.

The SNR-based technique can provide very accurate estimations of BER since it allows faster convergence (i.e. a few *Hello* packets are sufficient) to estimate the actual BER value. On the other hand, the probe-based technique needs a large number of Hello packets to accurately estimate BER values.

5 AODV and its Proposed Modifications

AODV builds routes using a route request / route reply query cycle. When a source node desires a route to a destination for which it does not already have a route, it broadcasts a route request (RREQ) packet across the network. Nodes receiving this packet update their information for the source node and set up backwards pointers to the source node in the route tables. A node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has a fresh route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ. Otherwise, it rebroadcasts the RREQ. If a node receives a RREQ which they have already processed, it discards the RREQ and do not forward it. As the RREP propagates back to the source, nodes set up forwarding pointers to the destination. Once the source node receives the RREP, it may begin to forward data packets to the destination. Details of the AODV protocol can be found in [12].

Our proposed modifications adhere to the on-demand philosophy, i.e. paths are still computed on-demand and as long as an existing path is valid, we do not actively change the path. Clearly other alternate designs are possible where even small changes in link error rates can be used to trigger exploration of better (i.e. more energy-efficient) paths. However, we view such a design as a deviation from the on-demand nature. Therefore, the our proposed (energy-efficient) route computation instants are the same as before — either in response to a query for a new route, or to repair the failure of an existing route.

5.1 AODV Messages and Structures

To perform energy efficient route computation for reliable data transfer, we need to exchange information about energy costs and loss probabilities between nodes that comprise the candidate paths. This information exchange is achieved by adding additional fields to existing AODV messages and structures (RREQ, RREP, Broadcast ID table, and Routing table) and does not require the specification of any new message. We describe the relevant changes to the existing message formats and data structures of AODV below:
- **RREQ** Message:

- C_{req}: Stores the average energy cost to transmit a data packet from the source to the current node along the path traversed by the RREQ message.
- E_{req}: Used only for the EER case to store the summation of energy consumed to transmit data packet over the links traversed by RREQ starting from the source

node to the current node. This field is calculated as: $E_{req} = \sum_{\forall l} E_l$, where $\forall l$ denotes the set of links traversed by RREQ.

- Q_{req}: Used only in the EER case to store the probability of transmitting a data packet successfully over the links the RREQ traversed starting from the source node to the current node. It is calculated as: $Q_{req} = \prod_{\forall l} (1 - p_l)$ where p_l is calculated as in Equation 7.

- **RREP** Message:

- C_{rep}: Stores the average energy cost to transmit a *data* packet over the links traversed by RREP starting from the current node to the destination node. Like the C_{req} field, its interpretation is different for the HHR and the EER cases.
- E_{rep}: Analogous to E_{req} (for all links from the current node to the destination node). It is used only in the EER case.
- Q_{rep}: Analogous to Q_{rep} (for all links from the current node to the destination node). It is used only in the EER case.
- Pt_{rep}: This is the transmission power level that the recipient of the RREP message should use in forwarding data packets towards the destination.
- $Bcast_{rep}$: this is the RREQ message ID that uniquely identifies the broadcast RREQ message which led to the generation of this RREP message.

- **Broadcast ID** Table: Each node maintains an entry in the Broadcast ID table for each route request query that is updated with each reception of RREQ.

- H_{bid}: The number of hops that has been traversed by the RREQ starting from the source node to the current node.
- C_{bid}: Stores the value of C_{req} field in the received RREQ message.
- E_{bid}: Stores the value of E_{req} field in the received RREQ message.
- Q_{bid}: Stores the value of Q_{req} field in the RREQ message.
- $Prev_{bid}$: Stores the ID of the node from which the current node received the RREQ message. This entry is updated for each received RREQ message given the RREQ message is forwarded by the current node. In case the received RREQ is dropped, this field is not updated.

- **Route** Table: A node maintains an entry in the route table for each destination to which it knows a route.

- C_{rt}: Stores the value of C_{req} field in the RREQ message or the C_{rep} field in the RREP message received by the current node. If the C_{rep} value is stored in this field, on receiving future RREQ messages for this destination node, it can be used as an estimate of the cumulative downstream cost from this node to the destination node.

- E_{rt}: Stores the value of E_{req} field in the RREQ message or the E_{rep} field in the RREP message received by the current node.
- Q_{rt}: Stores the value of Q_{req} field in the RREQ message or the Q_{rep} field in the RREP message received by the current node.
- Pt_{rt}: Stores the value of Pt_{rep} field in the RREP message.

5.2 Route Discovery

We now describe our modifications to route request and route reply phases.

5.2.1 Route Request Phase

The source node triggers the route discovery by initializing the new added fields in a RREQ message as $C_{req} = 0$, $E_{req} = 1$, and $Q_{req} = 1$ (the latter two are valid for the EER case). RREQ messages are transmitted at the node's maximum power level in order to reach all legitimate one hop neighbors. When an intermediate node n_i receives RREQ message from a previous node n_{i-1}, it calculates the energy (E_l) consumed by node n_{i-1} in a single transmission attempt of a data packet over the link $l = \langle i-1, i \rangle$ as a function of the the transmission power, Pt of node n_{i-1}.

For the fixed power case, this transmission power, Pt is a globally known constant. In the variable power case, the control messages, e.g. probe packets and RREQ messages, are sent with a fixed maximum transmission power Pt_{max}, which is globally known. Therefore, node n_i can calculate the transmit power to be used by node n_{i-1} for data packets as:

$$Pt = P_{Th} \times \frac{Pt_{max}}{Pr_{max}} \tag{12}$$

where that Pr_{max} is the power level at which control messages from n_{i-1} are received at n_i.

Subsequently, node n_i updates fields in the RREQ message as:

- HHR case:

$$C_{req} = C_{req} + \tfrac{E_l}{1-p_l}$$

- EER case:

$$\begin{aligned} E_{req} &= E_{req} + E_l \\ Q_{req} &= Q_{req} \times (1 - p_l) \\ C_{req} &= \tfrac{E_{req}}{1-Q_{req}} \end{aligned}$$

The packet error rate (p_l) is calculated by node n_i using the data size known in advance and the estimated $\textsc{Ber}_l$ as $p_l = 1 - (1 - \textsc{Ber}_l)^{data\_size}$.

Node n_i examines the broadcast identification number [1] stored in the RREQ message to check if it has seen any previous RREQ message belongs to the same route

[1] The broadcast identification uniquely identify all the RREQ messages belong to the same route request phase.

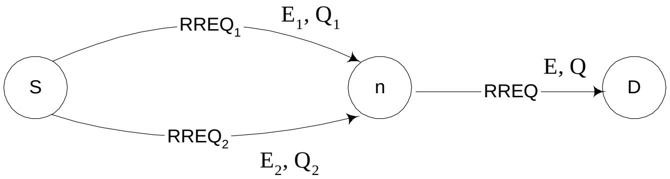

Figure 1: Calculating the energy cost for the EER case.

request phase or not. If this is the first instance for this RREQ, node n_i adds a new entry in its *Broadcast ID* table and initializes its values as $H_{bid} = H_{req}$, $C_{bid} = C_{req}$, $E_{bid} = E_{req}$, $Q_{bid} = Q_{req}$, and $Prev_{bid} = n_{i-1}$ where H_{req} is the number of hops traversed by this RREQ messages. Otherwise a previous RREQ message has been seen by the node n_i. In this case it compares the updated cost value in the RREQ message with that stored in the table entry. In the HHR case, if the boolean expression

$$(C_{req} < C_{bid}) \text{ OR } (C_{req} = C_{bid} \text{ AND } H_{req} < H_{bid}) \tag{13}$$

is true, then this RREQ message is further forwarded. Otherwise the currently best known route has lower cost than the new route discovered by this RREQ message, and so is discarded.

For a correct formulation in the EER case, the comparison rule (Expression 13) does not apply. This is because the cost function is not linear in the EER case. Consider Figure 1 in which node n receives two RREQ messages through two different paths from the source. The end-to-end energy costs for the two paths are $(E_1+E)/(Q_1 \times Q)$ and $(E_2 + E)/(Q_2 \times Q)$ respectively. The node n should choose the path defined by $RREQ_1$ if and only if

$$\frac{E_1 + E}{Q_1 \times Q} < \frac{E_2 + E}{Q_2 \times Q} \tag{14}$$

However, at the node, n, information on E and Q are not available and so this inequality cannot be evaluated. In case of $Q_{bid} \geq Q_{req}$, the Inequality 14 holds whatever the value of E is. In that case, node n could drop the RREQ message safely and no updates in the *Broadcast ID Table* is required. On the other hand, if $Q_{bid} < Q_{req}$, Inequality 14 will be hold only for

$$E \leq \frac{E_{req} * Q_{bid} - E_{bid} * Q_{req}}{Q_{req} - Qbid} \tag{15}$$

Based on the size of the network, average number of routes hops, and the average node transmission energy, node n estimate an energy threshold, E_{Th}, that is consumed over the rest of the path that is expected to be traversed by RREQ. E_{Th} value is monotonically decreasing with the remaining number of hops that could be estimated using H_{req}.

When node n receive the $RREQ_2$, it calculates the right side of the Inequality 15, E_{pass}, and compare with the estimated E_{Th}. If it is greater than E_{Th}, node n drops the $RREQ_2$ and no update is required in *Broadcast ID Table* since, with the high probability, the E on the rest of the route to the destination is equal to or lower than E_{Th}. On the other hand, if E_{pass} is lower than E_{Th} then with the high probability the energy E consumed by in the rest of the path is higher than the E_{Th} that in turn is higher than E_{pass}. In such case, node n_i forwards the $RREQ_2$ and updates *Broadcast ID Table* with the values of $RREQ_2$. As we can see, in EER case we can not guarantee to find the optimal route between the source node and the destination nodes since we may drop the RREQ message that traverse the optimum route in the middle. With the good estimation for E_{Th} we can get routes as close as, if not, the optimum

To optimally compute energy-efficient routes in the EER case, each separate RREQ message needs to be forwarded towards the destination. To do this, we need to uniquely identify each RREQ message belonging to the same route request phase, and also we need to maintain a separate entry in the Broadcast ID table for each RREQ message. Those entries can potentially lead to an exponential growth in the size of the table, and hence is not practical. Therefore, in practice we propose the same forwarding mechanism as used in the HHR case. To improve the quality of the chosen paths, we increase the state maintained in the Broadcast ID table to store information about the five RREQ messages with the lowest costs at each node.

As described in our modification, the intermediate nodes may broadcast multiple RREQ messages for the same route request phase, as an opposite to a single RREQ message in original AODV.

one.

5.2.2 Route Reply Phase

In AODV, the route reply (RREP) message can be generated by either the destination, or by an intermediate node that is aware of *any* path to the destination. In our modified version of AODV, the generation of RREP message is based on the cost of the candidate paths. If the destination node receives a set of RREQ messages from different paths, it chooses the path with the lowest cost among these alternatives and generates a RREP message along this path. Since the destination node receives multiple RREQ messages it has two choices: 1) Immediately reply with a RREP message for each better (i.e. more energy-efficient) route discovered by a new RREQ message, or 2) Wait for a small timeout to allow all RREQ messages to discover routes, and then send a single RREP response for the best discovered route. Clearly, the former approach will allow the destination node to select the optimum route at the expenses of transmitting multiple RREP messages, The latter approach results in just a single transmission of RREP message at the expense of higher route setup latency. In this paper we choose to implement the first approach of sending multiple RREP messages.

When the destination node receives the first RREQ, it updates its corresponding *Broadcast ID Table* entry to reflect the cost of the route traversed by this RREQ message as described in the previous section. For each duplicate RREQ message received by alternate path from the source, the destination node compares the cost traversed by this new RREQ with the one stored locally in its *Broadcast ID Table* using Expression 13. If the expression is true, then the cost of the path traversed by the new RREQ is lower. In such a case, the node updates its local information and responds with a RREP message for the path traversed by this new RREQ. Otherwise, it ignores the RREQ message. An intermediate node that receives a RREQ message for a destination can also generate a RREP message if it has a well-known route the destination[2] and the cost of the partial path traversed by this RREQ is less than the cost stored locally.

The node generating the RREP message copies the RREQ ID to the $Bcast_{rep}$ in the RREP message. For the variable power case, it also calculates the transmission power to be used by the previous hop node to transmit the data packets. This value is computed using Equation 12 and is stored in the Pt_{rep} field of the RREP message. The different fields of the RREP message are updated as:

- HHR case:

$$C_{rep} = \frac{E_l}{1-p_l} + C_{rt}$$

- EER case:

$$\begin{aligned} E_{rep} &= E_l + E_{rt} \\ Q_{rep} &= (1-p_l) \times Q_{rt} \\ C_{rep} &= \frac{E_{rep}}{1-Q_{rep}} \end{aligned}$$

where p_l is the packet error rate of the previous hop for data packets. If the node generating the RREP message is the destination itself, then $C_{rt} = 0$, $E_{rt} = 0$, and $Q_{rt} = 1$. The node forwards the RREP message to node $Prev_{bid}$ stored in the corresponding *Broadcast ID Table* entry to this RREQ.

When a node receives a RREP message for the first time, it creates an entry in the *Route Table* corresponding to this RREP. It initializes the fields of this entry as $C_{rt} = C_{rep}$, $E_{rt} = E_{rep}$, $Q_{rt} = Q_{rep}$, and $Pt_{rt} = Pt_{rep}$. If such an entry already exists, the node compares the cost values as described in the route request phase. If the new path has lower cost, then the *Route Table* entries are updated and the entries in the RREP message are appropriately updated and forwarded.

To update the RREP message, the node calculates E_l and p_l values corresponding to the link between this node and the node $Prev_{bid}$ stored in corresponding entry of *Broadcast ID Table*. The node updates the Pt_{rep} field using Equation 12. It also updates the other fields of the RREP message (i.e. C_{rep}, E_{rep}, and Q_{rep}) in the same way as done for a RREQ message.

As described above, the node may forward multiple RREP messages in response to better routes found by successive RREQ messages that indicate progressively lower-cost routes.

6 Simulation Experiments and Performance Evaluation

In this section, we report on extensive simulation-based studies on the performance of the AODV protocol, both with and without our energy-aware modifications. The performance comparisons were done using the *ns-2* simulator, enhanced with the CMU-wireless extensions (the underlying link layer is IEEE 802.11 with 11 Mbps data rate).

[2] By ”well known” we mean that the cost of the route from the current node to the destination is known.

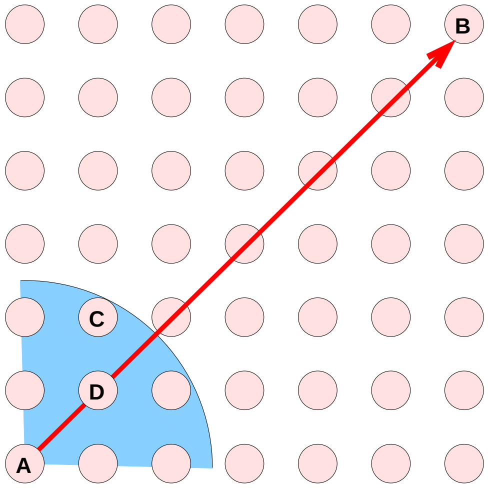

Figure 2: The 49-node grid topology. The shaded region marks the maximum transmission range for the node, A. $A \rightarrow B$ is one of the example flows used on this topology.

The simulations, therefore, model various scenarios of channel noise, interference between nodes due to channel contention, node mobility and their effects on performance. While the primary goal of this study was to study the benefits of a re-transmission aware routing scheme for on-demand protocols, our simulations also helped us quantify the performance of unmodified AODV under different noisy conditions on the wireless channels.

We have studied the performance of the different schemes for both HHR and EER cases, under both fixed and variable transmission power scenarios. In this paper, we will, however, focus only on the HHR case. This is because all practical link-layer protocols for multi-hop wireless attempt to provide some degree of reliable forwarding through the use of retransmissions or error control coding strategies.

To study the performance of our suggested scheme, we implemented and observed three separate routing schemes:

a) The Shortest-Delay (**SD**): The original AODV routing protocol that selects the route with the minimum latency.

b) The Energy-Aware (**EA**): Enhances the AODV protocol by associating a cost with each wireless link which is the energy required for a single packet transmission without the retransmission considerations. In this formulation of wireless link cost, the link error rates are ignored.

c) Our Retransmission-Energy Aware (**RA**): Which enhances the AODV protocol as described in this paper. As discussed previously, the link cost now considers the impact of retransmissions necessary for reliable packet transfer.

6.1 Network Topology and Link Error Modeling

For our experiments we use different topologies having 49 nodes randomly distributed over a 700×700 square region. The maximum transmission radius of a node is 250 units. We present results for three different topology scenarios:

- *Static Grid:* Nodes are immobile and equi-spaced along each axis as shown in Figure 2.
- *Static Random:* Nodes are immobile and uniformly distributed over the region.
- *Mobile Random:* Initially, nodes are distributed uniformly at random over the region. During the simulation, nodes move around the region using the random waypoint model [13] with zero pause time.

In all our simulations we had a set of 12 flows that were active over the duration of the experiment. We use both TCP and UDP flows for different experiments. For the TCP flows, we use its NewReno variant. For the UDP flows, we choose the traffic sources to be constant bit rate (CBR) sources at rate of 5 packets per second. The UDP packets and TCP segments were 1000 bytes each. Each of the simulations was run for a fixed duration of 250 seconds. Each point in the results is the average of 10 runs. When hop-by-hop reliability (HHR) is used, the UDP and TCP flows have similar performance patterns as was confirmed in our experiments. Therefore, we only present the results for the UDP flows.

All the control packets, e.g., probe packets, RREQ, RREP messages, IEEE 802.11 RTC/CTS/ACK frames, as well as the data packet experience the same bit error rate (BER) of a wireless link which depends on the ambient noise level as shown in Equation 5. We partitioned the entire square region into small square grids (50×50 units each). We model the ambient noise of each of these small square regions as independent identically distributed white Gaussian noise of μ mean and standard deviation σ. The noise mean μ for the different small square grids was chosen to vary between two configurable parameters, N_{min} and N_{max} corresponding to minimum and maximum noise respectively, while the noise standard deviations σ was chosen to be equal to $(0.1 \times \mu)W$. We used different distributions for the μ over the entire region for different experiments. In this paper, we focus only on the following extreme cases:

1. *Fixed noise environment:* N_{min} is equal to N_{max} and their values vary between $0.0W$ and $1.0 \times 10^{-11}W$.
2. *Random noise Environment:* We fix N_{min} to $0.0W$ and vary N_{max} between $0.0W$ and $1.0 \times 10^{-11}W$.

6.2 Metrics

To study the energy efficiency of the different schemes, we observed two different metrics:

1. **Average energy:** Computed per data packet by dividing the total energy expenditure (over all the nodes in the network) by the total data units (sequence

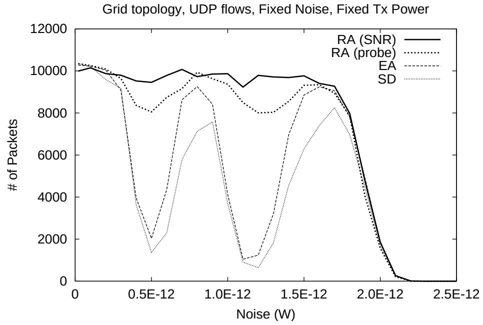

Figure 3: Effective reliable throughput

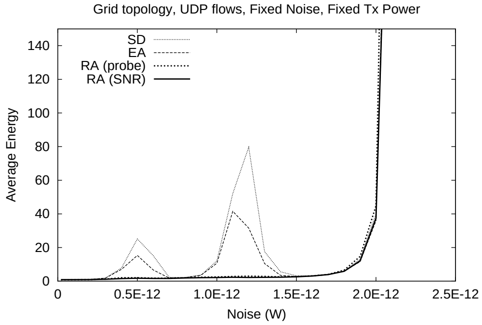

Figure 4: Average energy costs

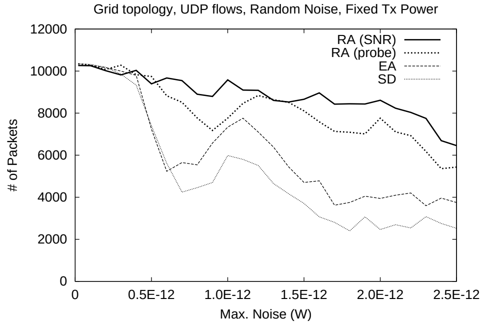

Figure 5: Effective reliable throughput

number for TCP and packets for UDP) received correctly at destinations. Note that this cost includes energy consumption due to control packets as well as the data packets. This metric is measured in Jouls.

2. **Effective Reliable Throughput:** This metric counts the number of the reliably transmitted packets from the source to the destination, over the simulated duration. Note that different schemes are able to transfer a different number of packets over an identical time interval. Since all the experiments have been performed over identical durations, we do not actually divide this packet count by the simulation duration. Instead we simply compare the total number of packets successfully transferred over this duration.

We choose 100μJ to be the energy cost for single attempt transmission of a bit a over a link for the fixed transmission power experiments, and to be the maximum energy cost corresponding to the maximum transmission power in case of the variable transmission power experiments.

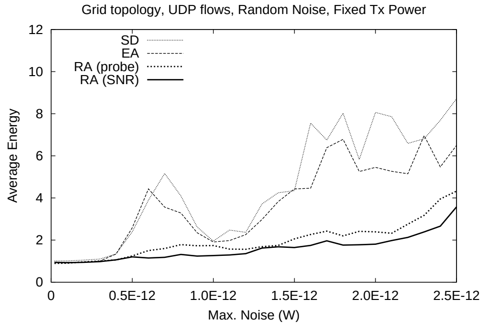

Figure 6: Average energy costs

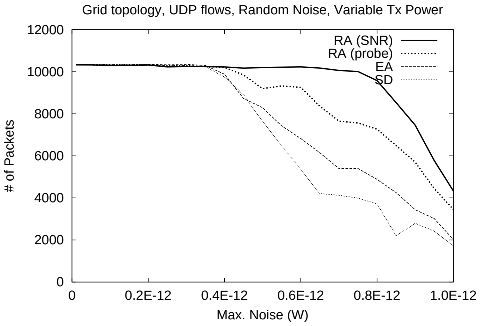

Figure 7: Effective reliable throughput

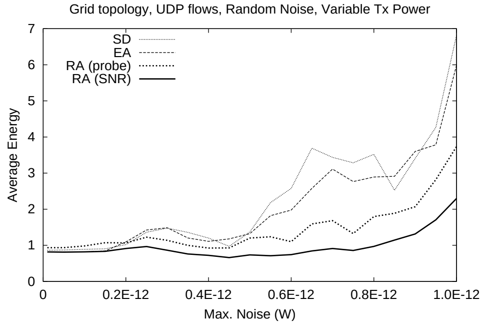

Figure 8: Average energy costs

6.3 Static Grid Topologies

Figures 3 and 4 show the effective reliable throughput and the average energy cost for experiments with fixed noise environments for UDP flows. Note that each data point on the plot corresponds to an experiment with a specified fixed noise value for the entire square region. Clearly for very low noise environments, all schemes are equivalent. However, as the noise in the environment starts to increase, the RA schemes show significant benefits. The SNR-based link error estimation has a superior performance than the probe-based technique. It is interesting to note that for both EA and SD schemes, the effective reliable throughput does not decrease monotonically. Instead at certain intermediate noise values (e.g. $5.0 \times 10^{-13} W$ and $1.2 \times 10^{-12} W$) the throughput goes to zero. This is an interesting phenomena that is related to the relative size of the RREQ packets and the data packets.

Consider the flow $A - B$ in Figure 2. Both SD and EA schemes for the fixed transmission power case chooses a path with minimum number of hops. Therefore, the first hop for this flow will be the link $\langle A, C \rangle$. For a static link , the BER is constant and depends on the noise value and the received power. but the packet error rates (PER) is not. PER depends on the size of the packets and is smaller for RREQ packets than the data packets. When the noise on the grid is $5.0 \times 10^{-13} W$, the BER for the $\langle A, C \rangle$ link is 0.0008. The corresponding PER for RREQ packets is about 0.5. Therefore

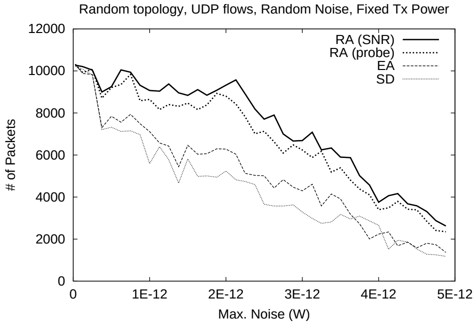

Figure 9: Effective reliable throughput

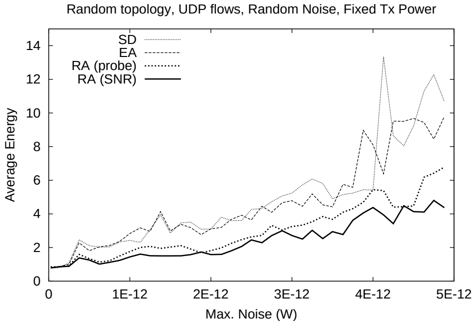

Figure 10: Average energy costs

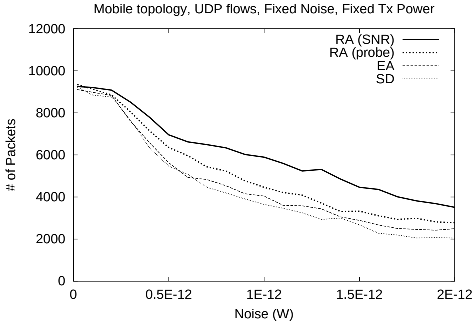

Figure 11: Effective reliable throughput

RREQ packets sent by node A is correctly received at C in about 50% of the cases and the link $\langle A, C\rangle$ is chosen by both SD and EA scheme using their usual cost metrics. However, the PER experienced by the data packets on the same link is nearly 1. This causes significant losses for data packets and therefore the throughput achieved is close to zero. However, when the noise level increases (i.e. say $6.0 \times 10^{-13}W$), the BER on the link goes up (i.e. to 0.00186). This causes the PER for RREQ packets to increase to 0.8. Therefore most of these RREQ packets get lost across link $\langle A, C\rangle$. Consequently both SD and EA schemes start to shift to paths with shorter hops with lower BER and their performance starts to approach the RA scheme.

The RA scheme do not suffer from this anomalous behavior. This is because the RA schemes choose routes based on the BER. Therefore, it automatically avoid links with high PER for data packets. This behavior is clearly visible in the grid topology, since the number of alternatives routes are discrete and few. Figures 5 and 6 show the corresponding plots for the random noise environment. The EA and SD schemes consume about 140% more energy per successfully transferred data unit than the RA schemes, when the maximum noise in the environment is $> 1.5 \times 10^{-12}W$ and still achieves only half the throughput of the RA schemes.

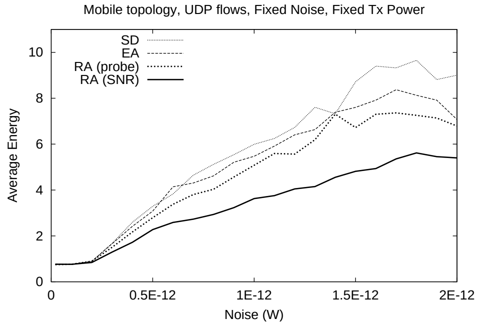

Figure 12: Average energy costs

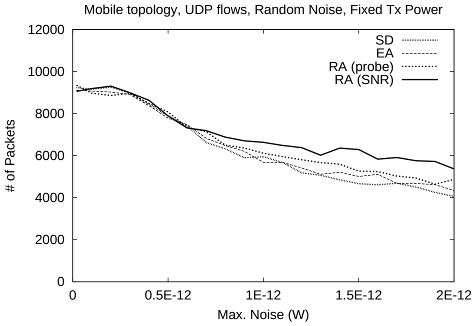

Figure 13: Effective reliable throughput

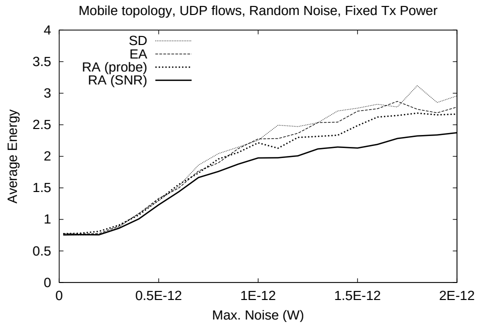

Figure 14: Average energy costs

For the sake of completeness, we present sample results for the variable transmission power case. In this experiment, we chose a power threshold at the receiver, P_{Th}, to be 1.0×10^{-9} W. The transmission power needs to be chosen such that the receiving node receives the packet with this power. Figures 7 and 8 show the effective reliable throughput and the average energy costs for UDP flows on a grid topology in a random noise environment for the variable transmission power case. As before, we can see performance benefits of the RA schemes over the EA and SD schemes.

6.4 Static Random Topologies

We next present results of the randomly generated static topologies (Figures 9 and 10). As before, the RA schemes provide significant performance benefits over the SD and EA schemes. For low noise environments, the SD and EA schemes consume 50-100% more energy for each reliably delivered data unit, while for high noise environments the energy requirements for the SD and EA schemes are about 2-3 times higher.

6.5 Mobile Topologies

In Figures 11 and 12 we show the effective reliable throughput and the average energy per reliable delivered data unit respectively in the fixed noise environment. Figures 13

	Fixed Noise Env.		Random Noise Env.	
Max. speed	Noise: 2.0×10^{-12} W		Max. noise: 2.0×10^{-12} W	
(m/s)	$\frac{T_{probe}}{T_{SNR}}$	$\frac{E_{probe}}{E_{SNR}}$	$\frac{T_{probe}}{T_{SNR}}$	$\frac{E_{probe}}{E_{SNR}}$
1	0.90	1.08	0.90	1.07
5	0.84	1.19	0.95	1.09
10	0.72	1.29	0.85	1.26
15	0.74	1.32	0.89	1.29

Table 1: Comparison of SNR-based and the probe-based versions of the RA scheme. T_{probe} and T_{SNR} are the reliable throughput of the probe-based and the SNR-based schemes respectively. Similarly E_{probe} and E_{SNR} are the average energy for the two schemes.

and 14 are the corresponding plots for the random noise environment. In both these cases, the maximum speed nodes is 10 m/s.

We observe that the impact of mobility increases with increase in the channel noise. For example, in absence of channel noise, the reliable throughput achieved for the mobile topologies is about 10% lower than the corresponding static topologies. As the channel noise increases the data throughput achieved for the mobile topologies is significantly lower.

From the energy consumption plots (Figures 12 and 14) it is clear that the RA scheme that uses the SNR-based link error estimation performs significantly better than all the other schemes. This is because mobility impacts the link error rates, and the SNR-based scheme is able to quickly adapt its estimates of the continuously changing link error rates. In contrast, the probe-based technique needs a longer duration to converge to the accurate link error estimate. Hence paths chosen by the RA (probe) scheme are less optimal than the ones computed by the RA (SNR) scheme. Consequently the RA (SNR) scheme performs significantly better than all the other schemes (upto 100% lower energy costs).

This effect is clearly visible in Table I where we compare the performance of the two RA schemes over different mobile topologies with varying speeds. As the mobility speed increases, we can see that the RA (SNR) scheme is more energy-efficient in comparison to the RA (probe) scheme (the benefits increase from 7% to 29% as the maximum speed of nodes increase from 1 m/s to 15 m/s in the random noise environment). It also achieves higher reliable throughput.

7 Conclusions

In this paper we have extensively studied the performance of the AODV protocol under varying wireless noise conditions. We have shown how AODV can be modified, through simple extensions to existing AODV messages, to compute minimum-energy routes, rather than ”shortest delay” routes. The modifications require each intermediate node to potentially forward multiple RREQ packets corresponding to a single unique route request (as long as the subsequent RREQs correspond to shorter-cost routes).

Our simulation studies show that the energy-aware modification of AODV can result in a significant (sometimes orders of magnitude) reduction in total energy consumption per packet, with the added benefit of higher throughput as well. In essence, the higher overheads of our energy-aware route establishment process (e.g., forwarding of multiple RREQs) are more than compensated for by the lower energy consumed in data forwarding. We presented two schemes to estimate the link error rates based on the characteristics of the environment. Our simulations show that the performance gains are impressive for both RA (SNR) and RA (probe) schemes compared with other schemes.

The performance benefits of the RA (probe) scheme diminish with increase in node mobility. This is because node mobility impacts link error rates and the performance benefits of our proposed scheme depends on the accurate estimation of link error rates. The probe-based link error estimation technique cannot quickly adapt estimation with increasing node mobility. We can achieve significant performance benefits even under high mobility scenarios as long as we can continue to maintain fast and accurate estimates of link error rates under such mobility conditions. In our experiments we show that the SNR-based link error estimation technique (applicable in free-space environments) has this property and achieves significantly better performance than all other schemes under a range of mobility conditions.

References

[1] W. Ye, J. Heidemann, and D. Estrin, "An energy efficient mac protocol for wireless sensor networks," in *Proceedings of Infocom*, June 2002.

[2] B. Prabhakar, E. Uysal-Biyikoglu, and A. El Gamal, "Energy-efficient Transmission over a Wireless Link via Lazy Packet Scheduling," in *Proceedings of IEEE Infocom*, Apr. 2001.

[3] J.H. Gass Jr., M.B. Pursley, H.B. Russell, and J.S. Wysocarski, "An adaptive-transmission protocol for frequency-hop wireless communication networks," *Wireless Networks*, vol. 7, no. 5, Sept. 2001.

[4] A. El Gamal, C. Nair, B. Prabhakar, E. Uysal-Biyikoglu, and S. Zahedi, "Energy-efficient Scheduling of Packet Transmissions over Wireless Networks," in *Proc. of IEEE Infocom*, 2002.

[5] K. Scott and N. Bambos, "Routing and channel assignment for low power transmission in PCS," in *Proceedings of ICUPC*, Oct. 1996.

[6] S. Singh and C.S. Raghavendra, "Pamas-power aware multi-access protocol with signaling for ad hoc networks," in *ACM Comm. Review*, July 1998.

[7] J. Gomez-Castellanos, A. Campbell, M. Naghshineh, and C. Bisdikian, "PARO: A power-aware routing optimization scheme for mobile ad hoc networks, draft-gomez-paro-manet-00.txt," *IETF*, Mar. 2001.

[8] J.-H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *Proceedings of Infocom*, Mar. 2000.

[9] C.K. Toh, H. Cobb, and D. Scott, "Performance evaluation of battery-life-aware routing schemes for wireless ad hoc networks," in *Proceedings of ICC*, June 2001.

[10] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-Efficient coordination Alogrithm for Topology Mainte nance in Ad Hoc Wireless Networks," *ACM Wireless Networks Journal*.

[11] Y. Xu, J. Heidemann, and D. Estrin, "Geographically-informed Energy Conservation for Ad Hoc Routing," in *Proc. of ACM Mobicom*, July 2001.

[12] C.E. Perkins and E.M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the* 2^{nd} *IEEE Workshop on Mobile Computing Systems and Applications*, Feb. 1999.

[13] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, 1996, pp. 153–181.

[14] V. Park and S. Corson, "Temporally-ordered routing algorithm (tora) version 1: Functional specification, draft-ietf-manet-tora-spec-04.txt," .

[15] A. Misra and S. Banerjee, "MRPC: Maximizing network lifetime for reliable routing in wireless environments," in *Proceedings of WCNC*, Mar. 2002.

[16] S. Banerjee and A. Misra, "Minimum energy paths for reliable communication in multi-hop wireless networks," in *Proc. of Mobihoc*, June 2002.