Energy-Efficient Reliable Paths for On-Demand Routing Protocols

Tamer Nadeem, Suman Banerjee, Archan Misra, Ashok Agrawala

Abstract— **We define techniques to compute energy-efficient reliable paths within the framework of on-demand routing protocols. Computation of minimum energy reliable paths for proactive protocols can be performed by simply using the appropriate cost metric in distributed route computation. No additional changes to the protocols are needed. In this work we show that such an approach does not work for on-demand protocols and some additional mechanisms are needed to compute energy-efficient paths.**

In this paper we focus on one specific on-demand routing protocol, namely Ad-hoc On-Demand routing protocol (AODV), and show how it can be enhanced to compute such energy-efficient reliable paths. 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. We identify some specific configurations with low or moderate channel noise, in which an on-demand protocol that is unaware of the noise characteristics will achieve insignificant throughput. Our results show that our proposed variants of on-demand routing protocols can achieve between 10% to orders of magnitude improvement in energy-efficiency of reliable data paths.

I. INTRODUCTION

Battery-power is typically a scarce and expensive resource in wireless devices. Since communication costs for wireless transmissions are high, energy efficient communication techniques are essential for increasing the lifetime of such wireless devices.

A large body of work has addressed energy-efficient link-layer forwarding techniques [16], [11], [5], [4], [12] and routing mechanisms [13], [6], [2], [14], [3], [15] for multi-hop wireless networks. These previously known energy-efficient routing techniques typically address two distinct and complementary objectives:

- *Finding energy-efficient end-to-end routes:* For wireless links, a signal transmitted with power P_t over a link with distance D gets attenuated and is received with power,
$$P_r \propto \frac{P_t}{D^K} \qquad (1)$$
where $K \geq 2$ is a constant that depends on the propagation medium and antenna characteristics. K is typically around 2 for short distances and omni-directional antennae, and around 4 for longer distances. The transmission power for these links are, consequently, chosen proportional to D^K. Protocols that compute energy-efficient end-to-end paths thus choose routes with a large number of small hops [13], [6].
- *Maximizing the lifetime of a network:* Another metric of interest in wireless environments is the lifetime of the network. Techniques for increasing network lifetime include alternating awake and sleep cycles for nodes [15], [3] and heuristic choices for routing traffic flows that balance the residual battery power at different nodes [2], [14].

An end-to-end reliability requirement can significantly af-

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

fect the above choices for both these classes of schemes. In particular, the choice of energy-efficient routes should take into account the channel noise in the vicinity of these nodes and the impact of this noise on transmission errors.

In [1], we had formulated and studied the minimum energy reliable communication problem for multi-hop wireless networks and had shown how standard routing protocols (e.g. link state and distance vector routing protocols) can be adapted to compute such paths. In that formulation, each link is assigned a cost based on two parameters:

- The transmission energy required for a single forwarding attempt across the link, which is an increasing function of the distance and is given by Equation 1.
- The error rate for packets on that link.

A standard (pro-active) routing protocol can periodically distribute such link costs to constituent nodes and then employ a distributed "shortest cost" path algorithm to compute the minimum-energy paths for unicast flows.

In this paper we describe how such minimum energy end-to-end reliable paths can be calculated for reactive (on-demand) routing protocols. On-demand routing protocols, as the name suggests, calculate paths *on-demand*. In these protocols, link costs are not periodically distributed to all other nodes in the network; rather, routes are computed only when needed by particular sessions. Accordingly, it is comparatively more difficult to directly employ metric-based shortest path computation algorithms to obtain minimum-energy routes. The problem becomes significantly harder for mobile networks since the link error rates (channel conditions) also change with node mobility. In the work presented here, we have experimented with the Ad-hoc On-demand Distance Vector Routing protocol (AODV) [10]. Accordingly, this paper describes our experience in developing a minimum energy end-to-end reliable path computation mechanism for AODV. It should, however, become obvious from our description that our technique can be generalized to alternative on-demand routing protocols (e.g., DSR [7] and TORA [9]). 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 present background and overview of our formulation of the minimum energy reliable path computation problem. In Section III we first briefly describe the AODV protocol, and then detail the necessary modifications to AODV behavior that we made to adapt it for minimum energy reliable path computations. In Section IV we present detailed simulation experiments to evaluate the performance of the protocols. In Section V we discuss related work in this area and finally we present our conclusions in Section VI.

II. Minimum Energy Reliable Paths

In [1], we had formulated and provided solutions for computing the minimum energy reliable path problem. Unlike traditional energy-aware routing techniques, our proposed solution 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 $15 - 25\%$.

In [1] we consider two different operating models:

1) *End-to-End Retransmissions (EER)*: where the individual links do not provide link-layer retransmis-

sions and error recovery — reliable packet transfer is achieved only via retransmissions initiated by the source node.

2) *Hop-by-Hop Retransmissions (HHR)*: where each individual link provides reliable forwarding to the next hop using localized packet retransmissions.

It is important to consider the link's error rate as part of the route selection algorithm in both cases. This is because the choice of links with relatively high error rates can significantly increase the energy spent in reliably transmitting a single packet, due to large number of re-transmissions necessary.

For any particular link between a transmitting node and a receiving node, let P_t denote the transmission power and p represent the packet error probability. Assuming that all packets are of a constant size, the energy involved in a packet transmission, E_t, is simply a fixed multiple of P_t.

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 D^K, where D is 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. The bit error rate associated with a particular link is essentially a function of the ratio of this received signal power to the ambient noise.

Like in [1], we consider two scenarios:

1) Fixed Transmission Power: In this case, each node chooses the transmission power to be a fixed constant, which is 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.
2) Variable Transmission Power: In this scenario, a transmitter node adjusts P_t 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}. The transmission power associated with a link of distance D in the variable-power scenario is, therefore, given by:

$$P_t = P_{Th}.\gamma.D^K, \tag{2}$$

where γ is a proportionality constant. Since P_{Th} is typically a technology-specific constant, we can see that the minimum transmission energy needed to sustain communication over such a link varies as:

$$E_{min}(D) \propto D^K. \tag{3}$$

A. Hop-by-Hop Retransmissions

We first consider the HHR case. Consider a link, l, which has a packet error rate, p_l. The number of transmissions (including retransmissions) necessary to ensure the successful transfer of a packet across the link is then a a geometrically distributed random variable X, such that

$$\text{Prob}\{X = k\} \; = p^{k-1} \times (1-p), \;\; \forall \; k$$

The *mean* number of individual packet transmissions for the successful transfer of a single packet is thus $1/(1-p)$. Therefore, the mean energy required for the successful transfer of this packet across the link is given by

$$E_l(\text{HHR}) = \frac{E_{opt}(D)}{1-p} \tag{4}$$

where D denotes the distance between the two nodes.

This analysis suggests that we should assign each link, l, a cost, E_l, equivalent to the mean energy required to successfully transmit a packet across the link. Any standard distributed routing protocol based on this cost can then be used to compute the minimum energy reliable paths.

B. End-to-End Retransmissions

It is not possible to compute minimum energy paths in the EER case using a distributed routing protocol with a single cost metric. Therefore in [1], we proposed an approximate cost metric which can be used for the EER case. This approximate cost metric for the EER case is given by:

$$E_l(\text{EER}) = \frac{E_{opt}(D)}{(1-p)^L} \tag{5}$$

where L is some small constant. Simulations have shown significant performance benefits using this proposed cost metric. The actual end-to-end energy requirements for a given path with nodes $0, \ldots, n$ in sequence for the EER case is given by:

$$E(\text{EER}) = \frac{\sum_{i=0}^{n-1} E_{i,i+1}}{1 - \prod_{i=0}^{n-1}(1 - p_{i,i+1})} \tag{6}$$

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.

III. AODV and its Proposed Modifications

The Ad hoc On Demand Distance Vector (AODV) routing protocol is an on-demand routing protocol designed for ad hoc mobile networks. AODV not only builds routes only when necessary, but also maintains such routes only as long as data packets actively use the route. AODV uses sequence numbers to ensure the freshness of routes.

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. In addition to the source node's IP address, current sequence number, and broadcast ID, the RREQ also contains the most recent sequence number for the destination of which the source node is aware. A node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has a route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ. If this is the case, it unicasts a RREP back to the source. Otherwise, it rebroadcasts the RREQ. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ which they have already processed, they discard 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. If the source later receives a RREP containing a greater sequence number or contains the same sequence number with a smaller hop-count, it may update its routing information for that destination and begin using the better route.

As long as the route remains active, it will continue to be maintained. A route is considered active as long as there are data packets periodically traveling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually be deleted from the intermediate node routing tables. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination(s). On receiving such an RERR, the source node will reinitiate route discovery, if it is still interested in a route to that destination node. A detailed description of the AODV protocol can be found in [10].

We now describe the set of modifications to the AODV protocol that are required to select energy-efficient paths for reliable data transfer. To implement an energy-efficient AODV for reliable data transfer, we need to add two simple, but fundamental, capabilities at the wireless nodes:

1) Estimation of Bit Error Rates (BERs) and transmis-

sion power for the different links. As we will describe, the BER estimation technique depends on the scenario — fixed transmission power case or variable transmission power case.

2) On-demand computation of energy-efficient reliable routes.

A. Estimating Links Bit Error Rate (BER)

Each node in the AODV protocol monitors and maintains state about all other nodes that are in its vicinity and can therefore serve as neighbors on the data path. To detect such neighbor connectivity information, each node periodically exchanges "Hello" packets with all such neighbors. Based on this exchange, each node maintains status information of each of its active neighbors in a *Neighbor List* table. The Hello packets are always transmitted by nodes using the maximum transmission power level. While this power level always equals the power level used for data packet transmission in the fixed power case, the two power levels may well be different in the variable power case. The maximum power level is employed since the job of the Hello packets is to exchange keepalives with *all potential* one-hop neighbors, i.e., all nodes with which a node can legitimately communicate over a direct link.

We first obtain the BER experienced by the Hello packets across the wireless link. As we shall explain later, the data packets, however, do not necessarily experience the same BER as the Hello packets due to the possible difference in their transmission power levels. We use the BER estimate of the Hello packets to obtain an estimate of the BER of the data packets. For our technique, it is sufficient for each node to estimate only the error rate on its incoming wireless links from its neighbors.

Calculating BER for Hello packets: Each node broadcasts a local sequence number within the Hello packet. The sequence number is incremented with each broadcast. A neighbor of this node receives only a subset of these broadcasted Hello packets. The remaining are lost due to channel errors. We define the time period between successive correctly received Hello packets as an *epoch*. Correct reception of a Hello packet terminates an epoch. Each node stores the sequence number of the last correctly received Hello packet from each one of its neighbors. On the reception of the next (i^{th}) Hello packet from a node, the receiving node can calculate l_i, the number of Hello packets lost in the last epoch. The total number of Hello packets broadcasted in this epoch is $l_i + 1$. Note that the packet error rate (PER) for a packet of length $packet\_size$ is given by:

$$PER = 1 - (1 - BER)^{packet\_size} \quad (7)$$

The packet error rate for Hello packets over the last epoch can also be calculated as $l_i/(l_i + 1) = PER_{Hello} = 1 - (1-BER_{Hello})^{hello\_packet\_size}$, where $hello\_packet\_size$ is the size of the Hello packet (in bits). Therefore, the receiving node can compute the BER of the last epoch, as

$$BER^{new}_{Hello} = 1 - exp(\frac{log(1 - \frac{l_i}{l_i+1})}{hello\_packet\_size})$$

The receiving node then updates its estimate of the BER of Hello packets for this incoming wireless link as follows:

$$BER_{Hello} = \alpha.BER^{old}_{Hello} + (1 - \alpha).BER^{new}_{Hello}. \quad (8)$$

The α factor is used to weight the sum of the BER estimated from this epoch and the previously estimated BER. α is a parameter that should be chosen based on how aggressively the BER estimate should depend on the new sample. A low value of α gives a larger weight to the new sample, and vice versa. Therefore, one way to choose the α parameter is based on the relative mobility pattern of the nodes. In scenarios where the relative mobility between nodes is high, link characteristics change very rapidly and therefore faster adaptation to new samples of the BER is required. Hence, in high mobility scenarios, a low value of α should be chosen. Similarly, in low mobility scenarios, a high value of α should be used.

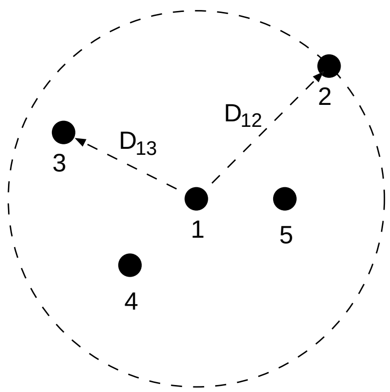

Fig. 1. Calculating the BER for the variable power case.

Each node stores its estimate of the BER of an incoming link in the corresponding entry in the *Neighbor List*. In the variable power case, each node also stores the power with which the Hello messages are received. This information is necessary in estimating the BER of data packets.

Calculating BER for data packets: For the fixed transmission power case, both Hello packets and data packets are transmitted with the same constant power by all nodes. Therefore, for a specific pair of transmitter and receiving nodes, the BER experienced by data packets is the same as the Hello packets. This implies that the BER estimate in Equation 8 computed for Hello packets is equally applicable for data 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 is given by Equation 2 and depends on the distance of the link. However, Hello packets sent by a node is broadcast to all possible neighbors and is transmitted with fixed transmission power, sufficient to reach all such neighbors. For example, in Figure 1, where $D_{1,2}$ represents the maximum transmission range of node 1, node 1 would transmit a Hello packet with power $P_{t,hello} \geq P_{Th}.\gamma.D_{1,2}^{K}$. It will, however, transmit a data packet to node 3 with the power $P_{t,data} = P_{Th}.\gamma.D_{1,3}^{K}$. Clearly, $P_{t,hello} > P_{t,data}$ in this example. Therefore, at node 3, the Hello packet from 1 is received with a higher received signal strength than the data packet sent from 1 to 3. The BER at a link is typically modeled as:

$$BER \propto erfc(\sqrt{\frac{constant.P_r}{N}})$$

where, N is the noise spectral density and P_r is the received power of the signal. The different constants depend on the choice of modulation scheme. $erfc(x)$ is the complementary function of the $erf(x)$ function, where

$$erfc(x) = 1 - \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

This implies that for packets received with higher received power (e.g. Hello packets), BER will be lower than other packets that are received with lower received packets (e.g. data packets). Note that Equation 8 provides an estimate of the BER for Hello packets. Therefore a suitable adjustment is required to estimate the BER for data packets in the variable power scenario.

For the sake of simplicity, we approximate the relation between the received power ($P_r >> 0$) and BER, using a data fitting model, as:

$$BER = exp^{-b.P_r - 1} \tag{9}$$

where b depends on the noise level at the receiving nodes. Using this relationship, we can calculate the BER of data packets from the BER of Hello packets as follows:

$$BER_{data} = BER_{hello}^{\frac{P_{r,data}}{P_{r,hello}}}.exp^{\frac{P_{r,data}}{P_{r,hello}} - 1} \tag{10}$$

Note that for the fixed transmission case, we will have $BER_{data} = BER_{hello}$.

B. Route Computation

To perform energy efficient route computation for reliable data transfer, we needed to exchange some 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 (RREQ and RREP) and does not require the specification of any new message. We describe the relevant changes to existing message formats and data structures below:

- **RREQ message**
  - $energy_{RREQ}$: This field stores the amount of energy consumed to send a data packet from the source to the current node. Its interpretation is different for the HHR and the EER cases.
  - q_{RREQ}: This field is used only in the EER case. It stores the probability of successful packet transmission from the source node to the current node.
- **RREP message**
  - $energy_{RREP}$: This field stores the amount of energy consumed to send a data packet from the current node to the destination. Like the $energy_{RREQ}$ field, its interpretation is different for the HHR and the EER cases.
  - q_{RREP}: This field is used only in the EER case. It stores the probability of successful packet transmission from the current node to the destination.
  - trx_power_{RREP}: The node receiving this RREP message (from the immediately downstream node) uses this value as the transmission power (P_t) for the data packets to the next hop (the transmitter of this RREP message) on the route.
  - $bcast\_id_{RREP}$: This is the RREQ message's ID that uniquely identifies the broadcast RREQ message which led to the generation of this RREP message.
- **BroadcastID table (bid)** A node maintains an entry in the BroadcastID table for each route request query to help in forwarding RREQ messages. We add the following fields to the BroadcastID entries to help in the route discovery phase, as shown later.
  - $request_{bid}$: The number of RREQ messages the node forwarded or replied to.
  - $hops_{bid}$: The hop count between the source and this intermediate node that the RREQ message traversed.
  - $energy_{bid}$: equal to the $energy_{RREQ}$ field in the received RREQ message.
  - q_{bid}: equal to the q_{RREQ} field in the RREQ message.
  - $from_{bid}$: The ID of the node from which the RREQ message was received.
- **Route Table (rt)** A node maintains an entry in the route table for each destination it knows a route to it. The following additional fields are required:
  - $energy_{rt}$: equal to the $energy_{RREP}$ field in the received
  - q_{rt}: equal to the q_{RREP} field in the RREP message.
  - trx_power_{rt}: equal to the trx_power_{RREP} field in the RREP message.

We now describe the operations of route discovery (generation and processing of RREQ messages) and route reply (generation and processing of RREP messages). We shall see that the determination of minimum energy routes requires enhancements not only to the source and destination node behavior, but also to the processing logic at intermediate nodes.

Route Discovery Phase: A RREQ message is initialized with $q_{RREQ} = 1$ and $energy_{RREQ} = 0$. This is broadcasted by the source node to initiate a route query. Like the Hello messages, all RREQ message is also transmitted at the maximum possible transmission power, so as to reach all legitimate one-hop neighbors of the transmitting node..

When an intermediate node, i, receives this RREQ message from a previous node, $i-1$, it updates the q_{RREQ} and $energy_{RREQ}$ fields and forwards it downstream, if appropriate.

For this, we need to first evaluate the energy required for a single transmission attempt of a data packet, $E_{i-1,i}$, across the link $\langle i-1, i \rangle$. For the fixed transmission case, this is a fixed and globally known constant value. In the variable transmission case, the control messages, e.g. Hello and RREQ messages, are sent with a fixed transmission power $P_{t,control}$, which is globally known. The data messages are sent so that the received power is minimally above the threshold, ie. equal to P_{th}. Therefore, the transmission power for data packets is given by,

$$P_{t,data} = P_{Th} \cdot \frac{P_{t,control}}{P_{r,control}} \quad (11)$$

Note that $P_{r,control}$ is locally known to the receiving intermediate node, i, while P_{Th} and $P_{t,control}$ are globally known constants (or optionally can be included as additional fields in the packet header). The receiving intermediate node can thus calculate $P_{t,data}$ and use it to obtain the corresponding energy, $E_{i-1,i}$, required to transmit the data packet at this power. Therefore the receiving node updates the fields in the RREQ message as follows:

- HHR case:

$$energy_{RREQ} = energy_{RREQ} + \frac{E_{i-1,i}}{1 - PER_{i-1,i}} \quad (12)$$

- EER case:

$$\begin{aligned} energy_{RREQ} &= energy_{RREQ} + E_{i-1,i} \quad (13) \\ q_{RREQ} &= q_{RREQ} \cdot (1 - PER_{i-1,i}) \quad (14) \end{aligned}$$

The $energy_{RREQ}$ field thus contains the cumulative transmission power for reliable delivery in the HHR case. In the EER case, this field contains the total transmission energy for only a single transmission attempt; the intermediate node must combine this field with the value in q_{RREQ} to obtain the effective energy for reliable transmission. The packet error rate $PER_{i-1,i}$ is calculated by node i as in Equation 7 using the BER estimate for data packets obtained using Equation 10 and stored in the *Neighbor List*.

If this is the first time the node, i, see this RREQ (identified by the BroadcastID), it adds an entry for this BroadcastID into the BroadcastID table. It initializes the fields ($request_{bid}$, $hops_{bid}$, $energy_{bid}$, $prob_{bid}$, and $from_{bid}$) appropriately. Alternatively, if this is not the first RREQ for the specific BroadcastID, the fields in the BroadcastID table are appropriately updated. For the HHR case, if the cost of the partial route discovered by the RREQ message is lower than the previous discovered routes, then the node forwards this RREQ message. The message is otherwise discarded at this node [1].

As we can see, *the intermediate may forward multiple RREQ message for the same route request query (arriving by different paths) multiple times* in contrast to the original AODV route discovery phase where it drops any RREQ duplicate. In the AODV protocol, each intermediate node forward only one RREQ message for every unique route request generated by the source node. While the suppression of duplicate RREQ messages significantly ameliorates the "broadcast storm" problem, this restricts the AODV routing protocol to the "shortest delay" route: the path taken by the first RREQ to reach the destination node. Clearly, the "shortest delay" path may not be the minimum cost path, when alternative metrics (such as our energy-aware metrics in Equations 4or 5) are considered. Therefore, our path discovery mechanism must allow multiple RREQ messages to be forwarded by the same intermediate node, as long as a later RREQ corresponds to a potentially "lower cost" path. While the route discovery phase allows us to discover a set of candidate paths, it is the des-

[1] For the EER case, such a choice does not necessarily lead to minimum energy paths. Therefore, we use a heuristic where RREQs with large number of hops are discarded.

tination that chooses the lowest energy path from among these multiple alternatives. Note that the number of RREQ messages is not unbounded because we drop the messages that have worse cost than the already discovered ones. We can also use other techniques may be also used to decrease the number of RREQ messages, e.g. based on hop count thresholds.

Route Reply Phase: In AODV, the Route Reply message can be generated by either the destination, or by an intermediate node that is aware of *a*ny path to the destination. In our energy-aware version of AODV, the generation of RREP message is based on the cost of the paths. If the destination node receives a set of RREQ messages, it chooses the path with the lowest cost among these alternatives and generates a RREP message along this path. Therefore, the destination node uses a small timeout value to receive the different RREQ messages that may follow the first one before generating the RREP message. Clearly, this approach should result in the selection of a more energy efficient path, at the expense of possibly greater route setup latency. Alternatively, if an intermediate node receives a RREQ message for a destination, it can generate a RREP message if it has a well-known route to the destination [2]. If this is a duplicate RREQ message (possibly received by an alternate path from the source), then the intermediate node will have a corresponding entry for this route request in its BroadcastID table. If the partial path cost from the source to this intermediate node is lower than the cost stored in the BroadcastID table, and if there exists a well-known route to the destination, then the RREP message is generated.

The node generating the RREP message copies the RREQ ID to the $bcast\_id_{rp}$ 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 11 and is put in the trx_power_{RREP} field of the RREP message. $energy_{RREP}$ and q_{RREP} are initialized to the $energy_{hop}$ and q_{hop} respectively (the latter only for the EER case) for the last hop traversed by the triggering RREQ packet. The node forwards the RREP to the next hop $from_{bid}$ defined in the corresponding BroadcastID. When an intermediate node receive an RREP message for the first time it stores the cost of the route from this node to the destination. If such an entry already exists, then the fields are appropriately updated. It also appropriately updates the $energy_{rp}$ and q_{RREP} fields of the packet. The RREP message is forwarded to the node that is stored in the $from_{bid}$ field of the BroadcastID table. A node may send multiple RREP messages in response to better routes found by successive RREQ messages that arrive by progressively lower-cost routes.

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

IV. 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. 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 perform experiments using both TCP and UDP traffic sources to study the effect of our routing schemes on these transport layer mechanisms. For the TCP flows, we used its NewReno variant. In UDP flows, packets were inserted by the source at regular intervals. 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 on only the HHR case, since 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 schemes, we implemented and observed three separate routing schemes:

a) The Shortest-Delay (**SD**) routing protocol. This is the original AODV routing protocol without any modification. The algorithm selects the route with the minimum number of hops.

b) The Energy-Aware (**EA**) routing protocol, enhances the AODV protocol by associating a cost with each wireless link which is the energy required to transmit a single packet (without retransmission considerations) across that link. In this formulation of wireless link cost, the link error rates are ignored. As shown in [1], the EA scheme is equivalent to the SD scheme in the fixed power case.

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

A. *Network Topology and Link Error Modeling*

For our experiments we used different topologies having 49 nodes randomly distributed over on a 700×700 square region, to study the effects of various schemes on energy requirements and throughputs achieved. In each case, we chose the maximum transmission radius of a node to be 250 units. We present results for three different topology scenarios:

- *Static Grid topology:* of 49 nodes is shown in Figure 2. The nodes are separated 100 units apart along each axis. Thus, each node had either 7 or 8 neighboring nodes on this topology. There was no node mobility in this case.
- *Static Random topology:* in which the 49 nodes were distributed uniformly at random over the square region. There was no node mobility in this case.
- *Mobile Random topology:* in which the 49 nodes were distributed uniformly at random over the square region. Additionally the nodes moved around the square region. We used the random waypoint model [7] for node mobility.

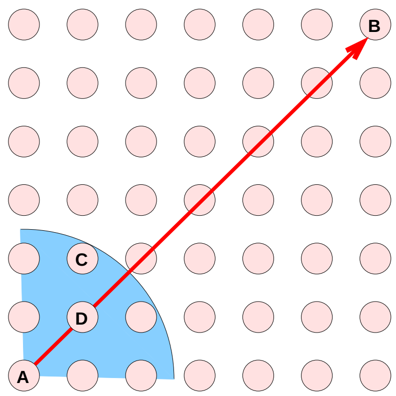

Fig. 2. The 49-node grid topology. The shaded region marks the maximum transmission range for the node, A. There is a single flow from each of the 4 corner nodes.

Each of the routing algorithms were then run on these topologies to derive the appropriate paths to each destination node. In our simulations for the static grid topology, each of the corner node had 1 active flow to its opposite corner node, providing a total of 4 flows. In the random and mobile topologies, we chose 4 random source-destination pairs from the entire set of nodes. We used both TCP and UDP flows for different experiments. 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 with a warm up period of 50 seconds. The four flow are started at times 50, 65, 80, and 95.

For different experiments we varied the noise at different points on the topologies. We partitioned the entire square region into small square grids (50×50 units each). Each of these small square regions was assigned a single noise level. Note that the bit error rate of a wireless link depends on the noise level and regions with higher noise has higher bit error rates for the corresponding wireless links. The noise 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.

We experimented with different noise distributions over the entire region. In this paper, we focus only on the following extreme cases:

1) *Fixed noise environment:* In this case, N_{min} is equal to N_{max}. For different experiments we varied $N_{max} = N_{min}$ between $0.0W$ $10.0 \times 10^{-12}W$.
2) *Random noise Environment:* In this scenario, we chose N_{min} to be $0.0W$ and varied N_{max} to be between $0.0W$ and $1.0 \times 10^{-11}W$ in different experiments.

Our results show that the other schemes are as good as the RA scheme only in zero noise environments. For all other cases, the RA scheme shows significant performance improvement, with the performance gain becoming larger with increasing levels of noise.

B. Metrics

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

1) **Average energy:** We compute the average energy per data packet by dividing the total energy expenditure (over all the nodes in the network) by the total data units (sequence number for TCP and packets for UDP) received at any destination.
2) **Effective Reliable Throughput:** This metric counts the number of packets that was reliably transmitted 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.

Fixed Transmission Power Case

In this paper, we will focus primarily on the results for the fixed transmission power case. The performance of the RA scheme provides a greater improvement over the EA and SD schemes for the variable transmission power case. Due to space constraints, we will show performance comparisons for the variable power case for only a few sample experiments later in this section.

For all the fixed transmission power experiments we choose a transmit power of 20 mW to be used by all the nodes for packet transmission.

C. Static Grid Topologies

Our static grid topology of 49 nodes is shown in Figure 2.

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 scheme shows significant benefits. 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. $6.0 \times 10^{-13}W$ and $1.2 \times 10^{-12}W$) the throughput goes

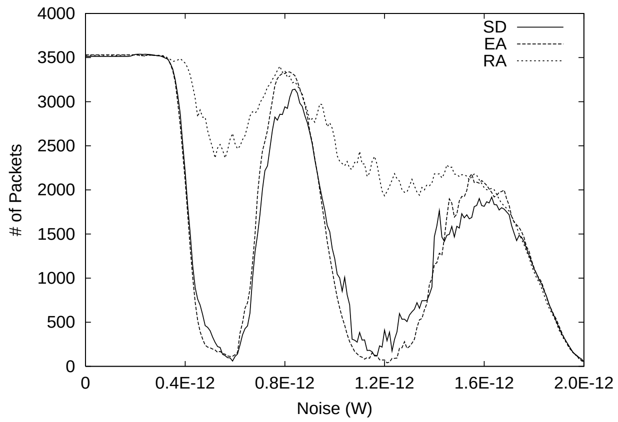

Fig. 3. Effective reliable throughput for UDP flows (Grid Topology, Fixed Transmission Power in Fixed Noise Environment).

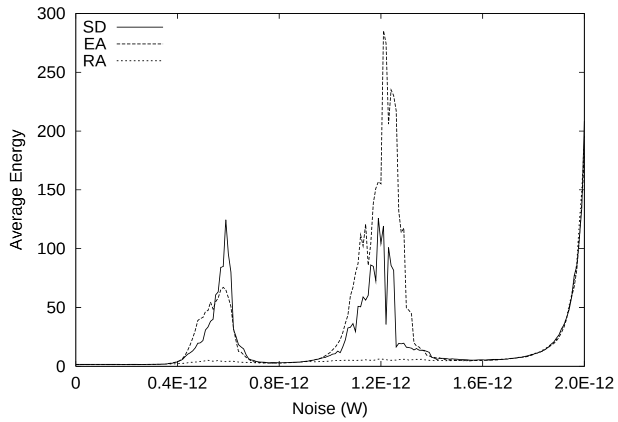

Fig. 4. Average energy costs for UDP flows (Grid Topology, Fixed Transmission Power in Fixed Noise Environment).

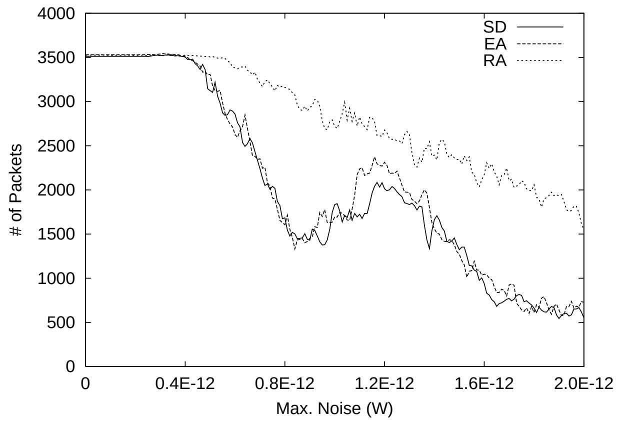

Fig. 5. Effective reliable throughput for UDP flows (Grid Topology, Fixed Transmission Power in Random Noise Environment).

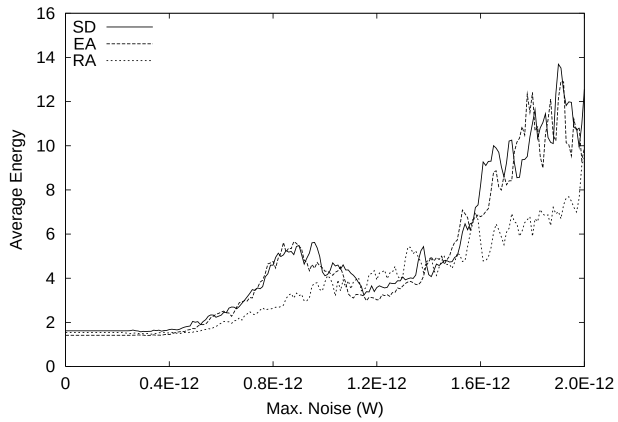

Fig. 6. Average energy costs for UDP flows (Grid Topology, Fixed Transmission Power in Random Noise Environment).

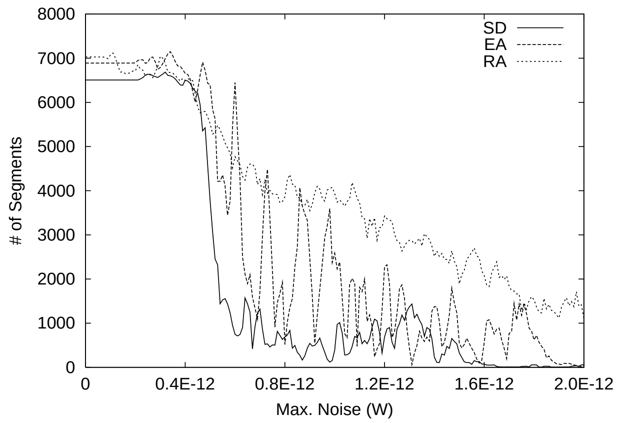

Fig. 7. Effective reliable throughput for TCP flows (Grid Topology, Fixed Transmission Power in Random Noise Environment).

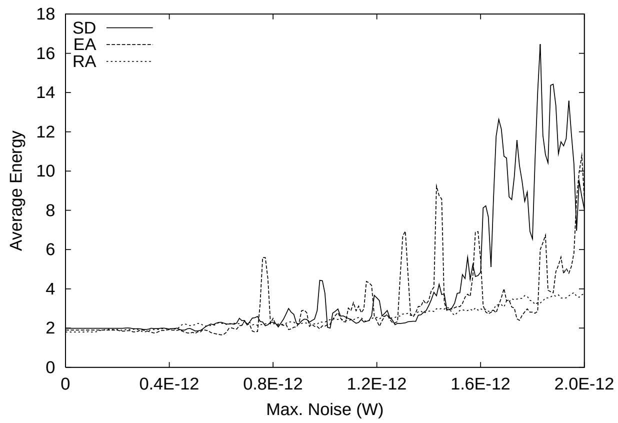

Fig. 8. Average energy costs for TCP flows (Grid Topology, Fixed Transmission Power in Random Noise Environment).

to zero. This is an interesting phenomena that is related to the relative size of the Hello 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 wireless hop for this flow will be the link $\langle A, C\rangle$. The bit error rate (BER) on any link depends on the noise value and the received power. Thus for a static link (e.g. $\langle A, C\rangle$) the BER is constant, but the packet error (PER) rate is not. PER depends on the size of the packets and is smaller for smaller sized packets. Therefore the PER is lower for Hello packets than the data packets. When the noise on the grid is $6.0 \times 10^{-13}W$, the BER for the $\langle A, C\rangle$ link is 0.00186. The corresponding PER for Hello packets is about 0.8. A link is considered inactive by the AODV protocol if it fails to receive three consecutive Hello packets. For a PER of 0.8, at least one of three successive packets sent by node A is correctly received at C in about 50% of the cases. Therefore, the link $\langle A, C\rangle$ is considered active by both SD and EA schemes and are chosen as first hops 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.

When the noise level increases (i.e. say $7.0\times10^{-13}W$), the BER on the link goes up (i.e. to 0.0036). This causes the PER for Hello packets to increase to 0.97 and three consecutive Hello packets are lost about 90% of the time. Therefore, the link $\langle A, C\rangle$ is considered inactive by the AODV protocol for routing purposes. Therefore both SD and EA schemes shift to paths with shorter hops (which also has lower BER) and their performance starts to approach the RA scheme.

The RA scheme does not suffer from this anomalous behavior. This is because the RA scheme chooses routes based on the *PER estimate for data packets.* Therefore, it is automatically avoid links with high PER for data packets. Both EA and SD schemes are oblivious of link errors and cannot make such intelligent choices. This specific behavior is clearly visible only in the grid topology, since the number of alternatives are discrete and few.

Figures 5 and 6 show the corresponding plots for the random noise environment. The EA and SD schemes consume about 40% more energy per successfully transferred data unit than the RA scheme, when the maximum noise in the environment is $> 1.6 \times 10^{-12}W$ and still achieves only half the throughput of the RA scheme.

Figures 7 and 8 plot these metrics for TCP flows for random noise environments. As expected the RA scheme achieves significantly higher throughputs than the EA and SD schemes for TCP flows. Note that in both SD and EA schemes, the paths are chosen without taking into account the error rates on individual links. Therefore, the chosen end-to-end paths for these schemes have higher losses. This increases end-to-end delays, which significantly affect the throughput of TCP flows. In fact, when the maximum noise in the environment approaches $2.0\times10^{-12}W$ the throughput achieved by both SD and EA schemes approach zero.

D. Static Random Topologies

We next present results of the experiments on randomly generated static topologies for UDP flows. The benefits of using the RA scheme over the SD and EA schemes is lower for randomly generated topologies than the grid topologies. However even in such cases, the energy requirements for the RA scheme is about half the energy requirements for the SD or EA schemes for a maximum noise level of $4.0\times10^{-12}W$. Note that the energy requirements of the SD and EA schemes significantly fluctuates with small changes in the noise level and is usually 10-150% higher than the RA scheme in all such cases.

E. Mobile Topologies

Finally we present results of the experiments on randomly generated mobile topologies. Node mobility was based on the random waypoint model [7]. In our simulation, we use a pause time of zero, which means that the

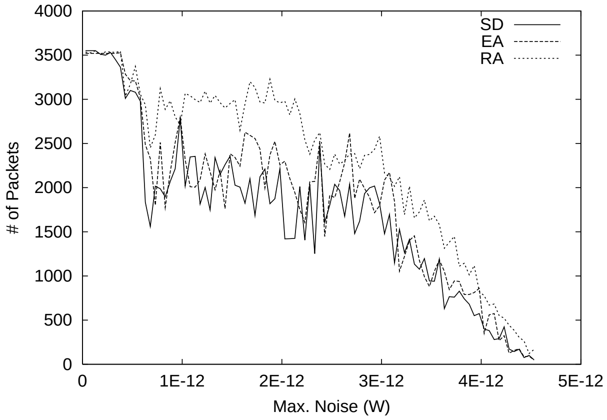

Fig. 9. Effective reliable throughput for UDP flows (Random Topology, Fixed Transmission Power in Random Noise Environment).

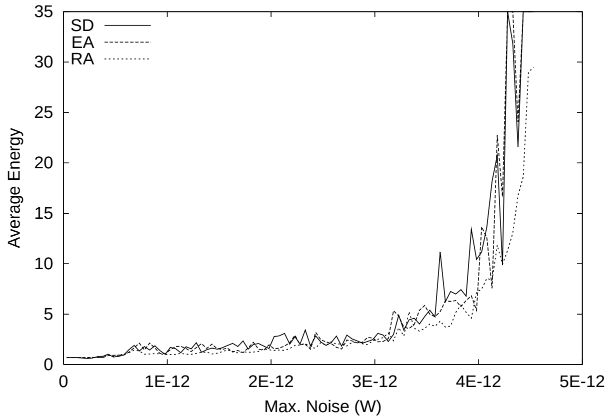

Fig. 10. Average energy costs for UDP flows (Random Topology, Fixed Transmission Power in Random Noise Environment).

nodes keep moving over the entire duration of the simulation. In Table IV-E we compare the performance of the three schemes for the fixed transmission power case implemented in the AODV protocol for UDP flows using the random noise environment. It can be noted that the RA scheme chooses low error paths and achieves at least 20% higher throughput that the other schemes. It also achieves this with lower energy costs. The benefits of the RA scheme over the other schemes is not as significant as in static grid topologies. This is because node mobility leads to continuously changing link distances, which in turn continuously change the link error rates. This makes it difficult to arrive at accurate estimates of link error rates at any time.

Max. Noise	Effective Reliable Throughput			Average Energy		
(W)	SD	EA	RA	SD	EA	RA
1.0×10^{-12}	2374	2442	2860	2.01	1.86	1.64
3.0×10^{-12}	1394	1446	1828	2.77	2.13	2.42
5.0×10^{-12}	578	604	686	3.58	2.91	3.43
7.0×10^{-12}	228	238	376	2.20	2.05	2.56

TABLE I

COMPARISON OF EFFECTIVE RELIABLE THROUGHPUT AND AVERAGE ENERGY COSTS FOR MOBILE TOPOLOGIES FOR THE FIXED TRANSMISSION POWER CASE IN RANDOM NOISE ENVIRONMENT USING UDP FLOWS. THE AVERAGE SPEED OF NODES WAS 5 M/S. THE MOVEMENT WAS BASED ON THE RANDOM WAYPOINT MODEL, WITH ZERO PAUSE TIME.

Variable Transmission Power Case

For the variable transmission power case we have chosen a power threshold at the receiver, P_{Th}, to be 5×10^{-11} W. The transmission power needs to be chosen such that the receiving node receives the packet with this power.

Figures 11 and 12 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. Unlike the fixed transmission power case, the energy requirements of the SD scheme is much higher than the EA scheme, which in turn is higher than the RA scheme. This is because for the variable power case the SD scheme chooses paths with a small number of large hops, while the EA scheme chooses paths with a larger number of small hops. When the maximum noise in the environment is greater than $1.2 \times 10^{-12} W$, the EA scheme incurs about 50% higher energy cost of the RA scheme, while the SD scheme incurs more than three times the cost. A similar performance can be observed for the TCP flows under the same conditions in Figures 13 and 14.

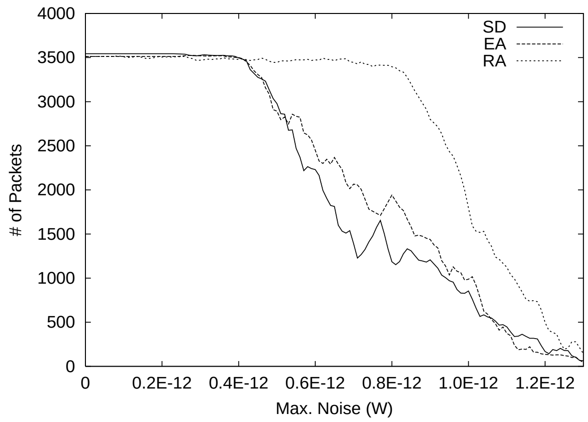

Fig. 11. Effective reliable throughput for UDP flows (Grid Topology, Variable Transmission Power in Random Noise Environment).

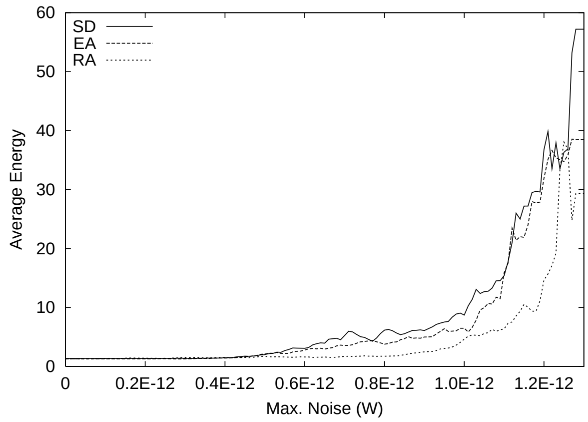

Fig. 12. Average energy costs for UDP flows (Grid Topology, Variable Transmission Power in Random Noise Environment).

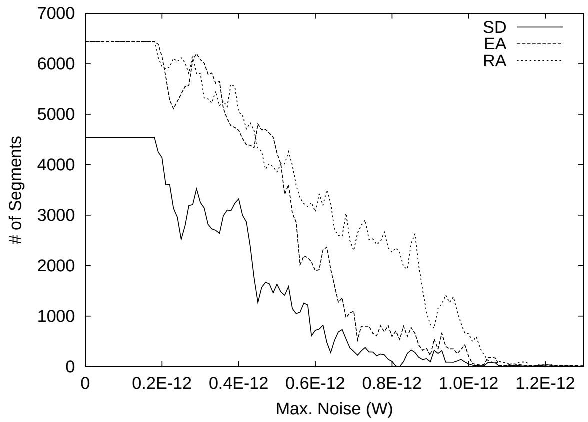

Fig. 13. Effective reliable throughput for TCP flows (Grid Topology, Variable Transmission Power in Random Noise Environment).

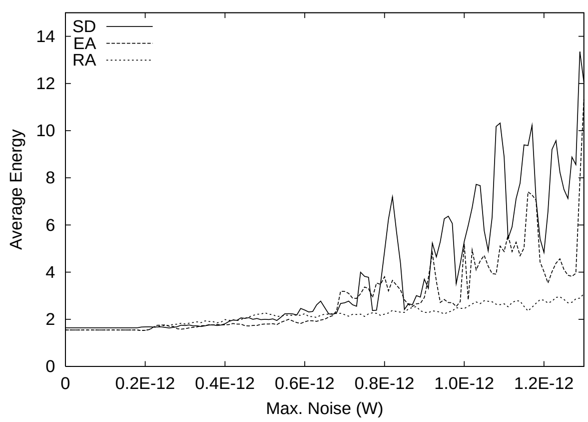

Fig. 14. Average energy costs for TCP flows (Grid Topology, Variable Transmission Power in Random Noise Environment).

V. Related Work

A large number of researchers have addressed the energy-efficient data transfer problem in the context of multi-hop wireless networks. As described in Section I, they can be classified into two distinct categories. One group focusses 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 [13], is one such minimum total transmission energy protocol, where the link cost was set to the transmission power and Dijkstra's shortest path algorithm was used to compute the path that uses the smallest cumulative energy. A link cost that includes the receiver power as well is presented in [12]. 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 focusses 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 [3] and GAF [15]), or by using heuristics that consider the residual battery power at different nodes [14], [2], [8] and route around nodes nearing battery exhaustion.

However, all of these schemes are typically defined as distributed *proactive* protocols. This paper defines the modifications needed to compute energy efficient (minimum energy) routes using reactive (on-demand) protocols, and studies the performance benefits of using a retransmission-aware energy metric in multi-hop wireless environments.

VI. 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). The destination node also needs to wait for a certain interval to collect cost information from potentially multiple alternatives before responding with an RREP for the minimum-energy path. In addition to changes in AODV behavior, our energy-aware framework recognizes the fact that Hello and data packets may have different bit and packet error rates, and requires each node to maintain estimates of the BER for data transmission from each of its one-hop neighbors.

Our simulation studies show that the energy-aware modification of AODV behavior can result in a significant (sometimes as high as 70-80%) reduction in total energy consumption per packet, often with the added benefit of higher throughput as well. In essence, the higher overheads of our energy-aware route establishment process (e.g., the forwarding of multiple RREQs) are more than compensated for by the lower energy consumed in data forwarding. Our simulations also show that our performance gains are more impressive in static or low-mobility networks: when nodes exhibit very high speed movement, the rapid changes in link noise levels and transmission powers can rapidly change the longevity of our minimum-energy paths. There are, however, several realistic ad-hoc networking scenarios (for example, rooftop radio-based community networks or handheld device based peer-to-peer clouds) with fairly low mobility–such scenarios can significantly benefit from the use of an energy-aware on-demand routing protocol.

References

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

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

[3] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An Energy-Efficient coordination Alogrithm for Topology Mainte nance in Ad Hoc Wireless Networks. *To appear in ACM Wireless Networks Journal*, 8(5), September 2002.

[4] A. El Gamal, C. Nair, B. Prabhakar, E. Uysal-Biyikoglu, and S. Zahedi. Energy-efficient Scheduling of Packet Transmissions over Wireless Ne tworks. In *Proceedings of IEEE Infocom*, June 2002.

[5] 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*, 7(5):487–495, September 2001.

[6] 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, work in progress. *IETF*, March 2001.

[7] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, pages 153–181, 1996.

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

[9] V. Park and S. Corson. Temporally-ordered routing algorithm (tora) version 1: Functional specification, draft-ietf-manet-tora-spec-04.txt, work in progress. *IETF*, July 2001.

[10] 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*, February 1999.

[11] 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*, April 2001.

[12] K. Scott and N. Bambos. Routing and channel assignment for low power transmission in PCS. In *Proceedings of ICUPC*, October 1996.

[13] S. Singh and C.S. Raghavendra. Pamas-power aware multi-access protocol with signaling for ad hoc networks. In *ACM Communications Review*, July 1998.

[14] 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.

[15] Y. Xu, J. Heidemann, and D. Estrin. Geographically-informed Energy Conservation for Ad Hoc Routing. In *Proceedings of ACM Mobicom*, July 2001.

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