IEEE 802.11 Fragmentation-Aware Energy-Efficient Ad-Hoc Routing Protocols

Tamer Nadeem, Ashok Agrawala
MIND Lab, UMIACS and Department of Computer Science
University of Maryland
College Park, MD 20742, USA
{nadeem, agrawala}@cs.umd.edu

Abstract

We define techniques to compute energy-efficient paths, using the IEEE 802.11 fragmentation mechanism, within the framework of on-demand routing protocols. We focus on one specific on-demand routing protocol, namely Ad-hoc On-Demand Vector routing protocol (AODV), and show how it should be adapted to compute energy-efficient paths. The choice of energy-efficient paths depends 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 and how it exploits the IEEE 802.11 fragmentation mechanism to generate optimum energy-efficient paths. 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 our proposed variants of on-demand routing protocols can achieve orders of magnitude improvement in energy-efficiency of reliable data paths.

1 Introduction

Minimizing energy consumption in wireless devices during communication is one of the interesting problems in the field of wireless communication. Different techniques and mechanisms have been proposed to reduce the communication cost and increase the power saving of the wireless devices. Large part of the work addresses energy-efficient link-layer forwarding techniques [31, 22, 10, 9, 26, 27] and routing mechanisms [12, 3, 28, 4, 29] 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 a wireless link, the power level of a signal transmitted gets attenuated over the link. The transmission powers for wireless links are chosen proportional to distances of those links. Thus, protocols that compute energy-efficient end-to-end paths choose routes with a large number of small hops [26, 12].

- Maximizing the lifetime of a network: Techniques for increasing network lifetime include alternating awake and sleep cycles for nodes [29, 4] and heuristic choices for routing traffic flows that balance the residual battery power at different nodes [3, 28].

Wireless communication suffers from high transmission errors due to the channel noise. To increase transmission reliability, wireless MAC protocols adopt different error control and reliability mechanisms. IEEE 802.11 standard implements retransmission mechanism in which a packet is retransmitted over a link if no MAC layer acknowledgment is received. In addition, IEEE 802.11 adopts a fragmentation mechanism that partitions large packets into smaller fragments to increase transmission reliability.

Such reliability mechanisms are applied on all transmitted data packets regardless of the used protocol service type (i.e. reliable service (e.g., TCP) or unreliable service (e.g., UDP)). Therefore, routing computations should take into account the different mechanisms provided by the wireless MAC layer to reduce the transmission errors.

Routing protocols in ad hoc networks can be categorized generally to: pro-active and re-active protocols. Pro-active protocols (e.g. link state and distance vector routing protocols) depend on maintaining routing information about the destinations at each node. A route is constructed in an incremental fashion in which each intermediate node, using some cost criteria, select the next link on the route toward the destination. As will be shown in Section 2, the wireless link (hop) error rate is estimated at the receiver end node of the link. In order to incorporate the link error costs in pro-active protocols where the sender node determines which link it transmits on, the receivers need to propagate all the link error information it gathered about
the neighbor links to the sender side nodes to update their cost criteria. Obviously, using link error costs in proactive routing protocols is not scalable due to the large transmission overhead in exchanging link error information between nodes.

On the other hand, re-active (on-demand) routing protocols compute routes only when needed in separate route-discovery phase. In this phase, intermediate nodes participate in selecting the links in which the nodes will receive the packets on. This is contrary to the proactive routing protocols where the intermediate nodes select links to forward the packets on. Hence, the link error computations fits perfectly with the re-active routing protocols in which the intermediate nodes (receiver end nodes) incorporate the estimated link error values in the choice of the route links with no need for data propagations. In this paper, we focus on the re-active protocols for their inherent scalability.

This paper develops a minimum energy end-to-end reliable path computation mechanism for Ad-hoc On-demand Distance Vector routing protocol (AODV) [21]. We assume that the IEEE 802.11 standard is the wireless MAC layer. Therefore, our routing computation takes into account the cross layer interaction with the MAC layer in order to increase the reliability by exploiting the available fragmentation mechanism provided by the IEEE 802.11 layer. It should, however, become obvious from our description that our technique can be generalized to alternative on-demand routing protocols (e.g., DSR [15] and TORA [20]).

2 Wireless Link Error Rates

It is important to explicitly consider the link’s error rate as part of the route selection algorithm to reduce the retransmission cost. This is because the choice of links with relatively high error rates will lead to large number of packet re-transmissions and, hence, significantly increase the energy spent in reliable transmission.

Any signal transmitted over a wireless medium experiences two different effects: attenuation due to the medium, and interference with ambient noise at the receiver. The energy spent in reliable transmission.

Figure 1. Bit Error Rates for different Noise and Distance values using Equations 2 and 1. The parameter values in those equations are defined in Table 1.

The parameter values in those equations are defined in Table 1.

\[
Pr = \begin{cases} \frac{Pt \times G_t \times G_r \times \lambda^2}{(4 \pi \times D \times \lambda)^2} & \text{for } D \leq D_{cross} \\ \frac{Pt \times G_t \times G_r \times h_t^2 \times h_r^2}{D^4 \times L} & \text{for } D > D_{cross} \end{cases}
\]

where \(Pt\) is the transmitted power, \(G_t\) is the transmitter gain, \(G_r\) is the receiver gain, \(D\) is the separation in meters, \(h_t\) is the transmitter height, \(h_r\) is the receiver height, \(L\) is the system loss factor not related to propagation (\(\geq 1\)), \(\lambda\) is the wavelength in meters, and \(D_{cross}\) is calculated as \(D_{cross} = \left\{4 \times \frac{\pi \times h_t \times h_r}{\lambda}\right\}\).

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, \(p\), associated with a particular link is a function of the ratio of the received signal power \((Pr)\) to the ambient noise.

The exact relationship between \(p\) and \(Pr\) depends on the choice of the signal modulation scheme. However, in general, several modulation schemes exhibit the following generic relationship between \(p\) and \(Pr\) is: \(p \propto \text{erfc}\left(\sqrt{\frac{\text{constant} \times Pr}{N}}\right)\) where \(N\) is the noise signal power and \(\text{erfc}(x)\) is defined as the complementary function of \(\text{erf}(x)\) and is given by: \(\text{erfc}(x) = 1 - \frac{1}{2\sqrt{\pi}} \int_0^x \exp(-t^2) dt\).

For the case of BPSK (Binary Phase-Shift Keying) and QPSK (Quadrature Phase-Shift Keying) the bit error is obtained by [23]

\[
p = 0.5 \text{erfc}\left(\sqrt{\frac{Pr \times W}{N \times f}}\right)
\]

where \(f\) is the transmission bit rate and \(W\) is the channel bandwidth (in Hz). Note that the CCK (Complementary Code Keying) used by IEEE 802.11b to achieve the 11 Mbps, which we assume in this paper where the bit rate \(f\) is 11 Mbps and the channel bandwidth \(W\) is 2 MHz, is modulated with the QPSK technology. Figure 1 plots the relation between the bit error rates, distance, and noise where the values of the propagation model parameters of Equation 1 are defined by Table 1.

We assume the transmission power of each node to be
a fixed constant $Pt^1$. For any particular link $l$, the energy required to transmit packets is independent of the distance $D$ and base only on the transmission power $Pt$ and the packet size $k$ bits. Although 802.11 uses a limited number of retransmission trials for a packet, we approximate the mean number of individual packet transmissions for a successful transfer of a single packet as $1/(1 - pt)^k$. This approximation is justified by (1) using of large number of retransmission trials per successful transfer, and (2) the assumption of sources with infinite data packets. The mean energy cost, $C_l$, required for a successful transfer of this packet across the link is given by

$$C_l = \frac{E_l}{(1 - pt)^k}$$

where $E_l$ is the energy consumed by the sender node for each transmission attempt across the link and $pt$ is the bit error rate over that link. Any energy-efficient protocol should consider the cost $C_l$ in their decision of selecting link $l$ or not. Note that 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.

In our proposed mechanism, it is sufficient for each node to estimate only the bit error rate, $p$, on its incoming wireless links from its neighboring nodes. 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 $SNR = 10 \log \frac{R}{N}$. From the SNR value measured by the wireless interface card, we can calculate the ratio $\frac{R}{N}$. Substituting it in Equation 2, we estimate the $p$ experienced by each received packet. This SNR-based error rate estimation technique is useful specially in free space environments where such error models are applicable. For other environments, where signal path characteristics depend more on the location and properties of physical obstacles on the paths, we use an alternative technique that is based on empirical observations of link error characteristics [18]. In this paper we focus on the SNR-based technique.

In practice, a passing mechanism should be used to hand the measured SNR and $Pr$ values from the wireless interface card to the upper routing algorithm. This could be implemented either by allowing the upper layers to pull those information through calls to APIs provided by the wireless card, or by pushing those information up using call-back functions defined by upper layers (e.g., AODV).

From Equation 2, the average energy involved in transmitting packets decreases with reducing the packet size ($k$). On the other hand, using smaller packet sizes increases the transmission overhead which is translated to energy cost. In the following section, we show how to calculate the optimum fragment size over a link to reduce the energy cost.

3 Packet Fragmentation and Energy Efficient Reliable Paths

In this section we describe the IEEE 802.11 fragmentation mechanism and the overhead associated with it. Also, we describe how to calculate the optimum fragment size for a link.

3.1 IEEE 802.11 Fragmentation Mechanism

The IEEE 802.11 standard [1] defines two access methods: the Distributed Coordination Function (DCF) which uses CSMA/CA to allow contended access to the wireless media and the Point Coordination Function (PCF) which provides free contention access via arbitration by a Point Coordinator. The DCF access method is suitable for ad hoc networks where no coordination point exists in such infrastructure. In DCF, a node wishing to transmit senses the channel first; if the medium is still idle for a period of time (DIFS Distributed Inter Frame Space), the node waits for a random contention window (CW) slot times of idle medium and then transmits. If the packet is correctly received, the receiving node sends an ACK frame after another fixed period of time (SIFS Short Inter Frame Space). If this ACK frame is not received by the sending node, a collision is assumed to have occurred. The sending node attempts to send the packet again when the channel is free for a DIFS period augmented of a random amount of time. In addition, the DCF defines the RTS/CTS mechanism, which requires that the transmitter and receiver exchange short Request-To-Send (RTS) and Clear-To-Send (CTS) control frames prior to the actual data frame transmission in order to reduce the collision effects.

IEEE 802.11 fragmentation mechanism creates smaller frames than the original ones to increase reliability by increasing the probability of successful transmission of the original frames in cases where channel characteristics limit reception reliability for longer frames. Fragmentation is accomplished at each immediate transmitter. Only MAC frames with a unicast receiver address will be fragmented. The standards define $aFragmentationThreshold$ as the threshold of fragmentation at which a MAC frames will be fragmented, to frames with length no longer than $aFragmentationThreshold$. The frames resulting from the fragmentation are sent as independent transmissions, each of which is separately acknowledged. This permits transmission retries to occur per fragment, rather than per the original frame. Figure 2 illustrates how IEEE 802.11 transmits the fragments using RTC/CTS mechanism.

---

1 Most current wireless cards do not provide any mechanism for adaptively choosing the transmission power for each packet.
3.2 Optimal Fragment Size Calculation

Fragmentation introduces an overhead associated with transmission of additional bits (additional energy cost) and additional delays (throughput reduction). Although this paper focuses on minimize the energy cost, the experiments show an increase in the throughput as a side effect of our proposed routing mechanism.

Two types of overhead bits are associated with the transmission of each fragment in IEEE 802.11. The bits ($o_1$), which are transmitted separately with each frame and are not considered as a part of the frame bits, represent one type of the overhead bits. As example: the PLCP preamble bits, the PLCP header [1], and the MAC ACK frames. The other type of the overhead bits ($o_2$) is transmitted within each frame. For example, the frame header and the frame CRC field. We assume that the energy necessary to transmit any bit of these types is equal to the energy needed to transmit any single fragment bit, $v$.

Given link $l$, it is required to find the optimal fragment size ($k_l^*$) that is corresponding to the minimum transmission cost. Assume the original packet size to be transmitted over the link is $L$ and it is fragmented to $k_l$ fragments each with size $l$, then the energy cost required for a successful transmission of single fragment, using Equation 3, is

\[ \frac{(o_1 + o_2) \times v}{(1 - p_l)^{k_l}} \]

Since the original packet will be partition into $L/k_l = o_2/o_1$ fragments, the total cost associated with a successful transmission of a packet is:

\[
C_l = \frac{L}{k_l - o_2} \times \frac{(o_1 + k_l) \times v}{(1 - p_l)^{k_l}}
\]

\[= L \times v \times \frac{1 + k_l}{o_1 + k_l} \times (1 - p_l)^{k_l} \tag{4}\]

Figure 3 plots Equation 4. It shows the mean cost of successful single bit delivery with different fragmentation sizes and different $p_l$ values assuming the transmission bit energy, $v$, is one unit. Using small segment sizes, the link transmission cost is very high due to the high overhead included. With increasing the segment size, the cost is decreased until it reaches its minimum value using the optimal segment size ($k_l^*$). Increasing the segment size beyond $k_l^*$ results in increasing the link cost again due to the increase in the retransmission trials. To find $k_l^*$, we differentiate Equation 4 with respect to $k_l$ and equal it to zero to get:

\[
k_l^* = (o_2 - o_1) \beta - \sqrt{(o_2 - o_1)^2 \beta^2 - 4 \beta (o_1 + o_2 - 1)}}{2 \beta} \tag{5}\]

where $\beta = \ln(1 - p_l)$.

Using optimum fragment size over links has two impacts.

1. It reduces the energy cost significantly over individual links. For example, in Figure 3 transmitting a 1500 bytes packet over link with $p = 1.0 \times 10^{-4}$ using fragments of size 300 bytes reduces the cost per bit by 54% from 3.48 energy unit to 1.6 energy unit.

2. It increases the possible alternative routes which gives the flexibility of selecting shorter paths with lower end-to-end energy cost. For example, consider two alternative paths: the first path consist of a single hop with $p = 4.0 \times 10^{-4}$ and cost of 2.6 unit. The other

\[\text{Normalized Energy per Bit}\]

\[\text{Fragment Size (bits)}\]

\[p = 4.0E-00\]

\[p = 2.0E-04\]

\[p = 1.0E-04\]

\[p = 0.5E-04\]

\[p = 0.0E-00\]
path consists of two hops each with \( p = 1.0 \times 10^{-4} \)
and cost of 1.6 unit. Although the individual link costs
on the first path is higher than the links of the second
path, selecting the first path will cost in total 2.6 unit
which is lower than the total cost of the second path
(3.2 units).

In this paper we assume that given the \( p \) value of a
link, the IEEE 802.11 MAC layer will calculate and use
the optimum fragmentation size for packet transmissions in
case the fragmentation mechanism is enabled. In practice, a
passing mechanism between physical/data link layer and the
network layer should be implemented as stated in Section 2
to help in passing information about what fragment size
should be used and when the fragmentation is used between
the layers as needed.

4 AODV and its Proposed Modifications

AODV builds routes using a route request-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 sends 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. Otherwise,
it broadcasts the RREQ. Nodes keep track of the RREQ’s
source IP address and broadcast ID. If a node receives a
RREQ which it has 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. A route is considered active as
long as there are data packets periodically travelling from
the source to the destination along that path. Once the
source stops sending data packets, the links time out and
eventually be deleted from the intermediate node routing
tables. A detailed description of the AODV protocol can be
found in [21].

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.

4.1 Link Error Rates Estimation

As described in Section 2, bit error rates calculation
depends on the power level of the received signal \( (Pr) \)
and the ambient noise \( (N) \) surrounding a node. In order
to estimate the error rates accurately, we need "good”
measurements of \( Pr \) and \( N \), and thus SNR. Generally, \( Pr \)
and \( N \) vary with time: \( N \) varies due to the environment
conditions, and \( Pr \) varies due to the nodes mobility.
Consequently, we can not base our calculations on a
single measurement. Rather we need to calculate those
parameters as function of several measurements over a
window of time, in order to capture the dynamics of the
network. We considered different approaches to calculate
these measurements:

1. Instead of broadcasting single RREQ by each interme-
diate node during route discovery phase, each sender
node broadcasts multiple RREQ packets in sequence
separated by sampling period. To force the AODV
layer of the receiver nodes to discard all those RREQ
packet except the last one, the TTL field of those
RREQ packets is set to zero. The TTL field of the last
RREQ packet is set to the regular TTL value. In this
way, the receiver nodes calculates the parameter values
using the parameter measurements from those multiple
RREQ packets. Although this approach follows the on
demand theme however, it increases the duration of the
route discovery phase.

2. Each node periodically exchanges Hello packets with
all its neighbors. In such method, nodes calculate and
maintain the parameter values during their life time.

We choose to implement the second approach in this
paper in which each node broadcasts Hello packet of
small fixed size, at an average period \( t \) (one second in
the implementation). To avoid accidental synchronization
and consequently collisions, \( t \) is jittered by up to \( \pm 0.25t \).
The receiving nodes measure the SNR value for each
received Hello packet, and uses this value to estimate the
corresponding \( p \) of the incoming links as described in
Section 2. Each node continuously updates its estimate of
the SNR and the corresponding \( p \) using an exponentially
weighted moving average of the sampled SNR values.

Broadcasting of Hello packets could change dynamically
with the network traffics. One possibility, a node could stop
broadcasting the Hello packets when it doesn’t sense any
traffic in the neighborhood and restart to broadcast them
once it detects traffic.

4.2 Messages and Structures

To construct energy efficient route, nodes along the
candidates paths need to exchange information about
energy costs and loss probabilities within the route-
discovery phase. Consequently, we added the following
changes to the structures maintained by AODV (e.g.,
Broadcast ID and Routing tables) and to AODV messages
(e.g., RREQ and RREP).

October 24-27, 2004, Fort Lauderdale, Florida, USA (runner up for Best Paper Award)
- **RREQ Message:**

  - \( C_{req} \): Stores the average energy cost to transmit a single data bit from the source to the current node along the path traversed by the RREQ message.

- **RREP Message:**

  - \( C_{rep} \): Stores the average energy cost to transmit a data bit over the links traversed by RREP starting from the current node to the destination node.

  - \( F_{rep} \): The optimum fragment size, used by the receiving current node of the RREP message, to fragment the transmitted data packets on the next link 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:** Used by each node to maintain an entry 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**.

  - \( Prv_{bid} \): Stores the ID of the node from which the current node received the **RREQ**. This entry is updated for each received and forwarded **RREQ** message by the current node.

- **Routing Table:** A node maintains an entry in the routing table for each destination it has a route for.

  - \( C_{rt} \): 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. This field is used as an estimate of the cumulative upstream/downstream cost from this node to the source/destination node.

  - \( F_{rt} \): Stores the value of \( F_{rep} \) field in the **RREP** message. This value is passed to the IEEE 802.11 MAC layer with each transmitted packet either through API call or a special field within the packet. The MAC layer partitions the packet to fragments each with maximum size equal to \( F_{rt} \).

### 4.3 Route Discovery

AODV Routed discovery consists of two phases: route request phase and route reply phase. We now describe our modifications to these two phases.

### 4.3.1 Route Request Phase

Algorithm 1 describes the steps a node follows when it receives a **RREQ** message in modified AODV route request phase. The source node triggers the route discovery by broadcasting a **RREQ** message initialized with \( C_{req} = 0 \) (other fields are initialized as in the original AODV algorithm). When an intermediate node \( n_i \) receives **RREQ** message from a previous node \( n_{i-1} \), it updates fields in the **RREQ** message (line (i) of Algorithm 1).

**Algorithm 1: Route Request Handler()**

```
procedure recvRequest(RREQ packet)
  VARIABLE :
  BID : The broadcast ID table
  \( n_{i-1} \) : Node transmitted this RREQ
  \( n_i \) : Node receiving this RREQ
  \( l \) : Link \((i-1,i)\) **RREQ** traversed
  \( k_i^* \) : Optimum fragment size over link \( l \) (Equation 5)
  \( H_{req} \) : Number of hops traversed by this **RREQ** message
  ALGORITHM :
  main
  Calculate \( k_i^* \)
  \( C_{req} \leftarrow C_{req} + \frac{v\times(n_i+k_i^*)}{(k_i^*-n_2)(1-p_1)} \) \( (i) \)
  \( H_{req} \leftarrow H_{req} + 1 \)
  Search \( BID \) for \( Bid_{req} \)
  if (Found an entry in \( BID \))
    then {if (\( (C_{req} \geq C_{bid}) \) or \( (C_{req} = C_{bid} \) and \( H_{req} \geq H_{bid} \)) \( (ii) \)
      then return
    }
  else {Add corresponding **RREQ** entry in \( BID \)
    \( H_{bid} \leftarrow H_{req}, C_{bid} \leftarrow C_{req}, Fr_{bid} \leftarrow k_i^*, Prv_{bid} \leftarrow n_{i-1} \)
    Update the reverse route information (as in original AODV)
    if \( n_i \) \( \) (the destination or have path to the destination)
    then Prepare the reply packet **RREP** and initialize its new fields as: \( C_{rt} = C_{rt} + \frac{v\times(n_i+k_i^*)}{(k_i^*-n_2)(1-p_1)} \) + \( C_{rt} \)
      comment: \( C_{rt} = 0 \) in case of \( n_i \) is the destination
      \( F_{rep} \leftarrow k_i^* \), \( Bcast_{rep} \leftarrow Bid_{req} \)
      Send **RREP** to \( Prv_{bid} \) node
    else Forward the **RREQ** to \( n_i \) neighbors
   }
```

Node \( n_i \) examines the broadcast identification number \(^2\) \( Bid_{req} \) stored in the **RREQ** message to check if it has seen any previous **RREQ** message belongs to the same route request phase or not. If this is the first instance for this **RREQ** or the cost associated with this **RREQ** is lower than the known one by the node \( n_i \) (line (ii) of Algorithm 1), the node update the corresponding fields in **Broadcast ID** table entry and then it forward the **RREQ** message.

\(^2\)The broadcast identification uniquely identify all the **RREQ** messages belong to the same route request phase.
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. Although this broadcast costs more energy, the experiments show that this overhead cost is marginal to the total energy saving.

Algorithm 2: ROUTE REPLY HANDLER()

procedure recvReply(RREQ packet)

variable: RT : The routing table
BID : The broadcast table
n_i : Node receiving this RREQ
SeqNo : Sequence number for the destination
H : Number of hops to the destination

algorithm:

main

Search RT for an entry of the destination
	if (Found an entry in RT)
		if ((SeqNo_req < SeqNo_rep) or (C_rep ≥ C_req) or (C_rep = C_req and H_req ≥ H_rep))
			then return

	else

		Add entry in RT for the destination

comment: Update the fields of RT

H_rep ← H_req, C_rep ← C_req, Fr_rep ← Fr_req

if (n_i is the source)
	then return

Get the BID entry corresponding to Bcast_rep

n_{i-1} ← Prevbid

Calculate the k_i, where l is the link (i − 1, i)

C_rep ← \frac{(\alpha_1 + \alpha_2)}{(\beta_1 \times (1 - \beta_1))} + C_{rep}

Fr_rep ← k_i

Send RREQ to n_{i-1} node

4.3.2 Route Reply Phase

Last part in Algorithm 1 shows how the destination node or an intermediate node that has a well-known route to the destination generates and forwards RREP message.

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 expense 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.

Algorithm 2 describes how a node handles a RREP message in the modified route reply phase. Similar to RREQ message, when a node receives a RREP message for the first time or the received one has route with lower cost, it updates the entry in the Routing table corresponding to this RREP. Then, the RREP message is appropriately updated and forwarded to Prevbid node.

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.

5 Simulation Experiments and Performance Evaluation

In this section, we present extensive simulation-based studies on the performance of the AODV protocol, both with and without our 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). We extended ns-2 version 2.1b8a with the full implementation of IEEE 802.11 fragmentation mechanism.

We modelled various scenarios of channel noise, interference between nodes due to channel contention, node mobility and their effects on performance. To study the performance of our suggested schemes, 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 considering the energy cost of a single bit transmission (without retransmission considerations). However, this algorithm selects, among the different candidate routes of the same cost, the one with the highest packet delivery probability.

c) Our Retransmission-Energy Aware (RA): Enhances the AODV protocol as described in this paper. The link cost considers the impact of retransmissions necessary for reliable packet transfer.

We run each of one of the above schemes on IEEE 802.11 fragmentation-disabled version (SD_fix, EA_fix, Fix, EA

October 24-27, 2004, Fort Lauderdale, Florida, USA (runner up for Best Paper Award)
and RA\textsubscript{fix}), as well as fragmentation-enabled version (SD\textsubscript{var}, EA\textsubscript{var}, and RA\textsubscript{var}). For fragmentation-disabled version, packets are transmitted at their original sizes. On the other hand, the MAC layer of the fragmentation-enabled version schemes exploits the fragmentation availability by partitioning the packets, over each link, to the optimum fragment size. Only RA\textsubscript{var} scheme is aware of the fragmentation mechanism and use it in its route computations to obtain the best energy-efficient route.

We adopted RTS/CTS mechanism in the IEEE 802.11 MAC layer to factors out the effect of collisions from our results. Also RTS/CTS is used in the AODV as a detection mechanism for link failure (absence) rather than using data packets. The link is triggered as broken if no CTS frame is received for a number of consecutive trials of a RTS frame (in our case we set such number to 4). Table 1 summarizes the parameter used in our simulation\textsuperscript{4}.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet Payload</td>
<td>1500 bytes</td>
<td>data frame payload length</td>
</tr>
<tr>
<td>MAC header</td>
<td>28 bytes</td>
<td>MAC layer overhead</td>
</tr>
<tr>
<td>PLCP Preamble</td>
<td>144 bits</td>
<td>PLCP Preamble overhead length</td>
</tr>
<tr>
<td>PLCP Header</td>
<td>48 bits</td>
<td>PLCP Header overhead length</td>
</tr>
<tr>
<td>ACK</td>
<td>14 bytes</td>
<td>ACK frame length</td>
</tr>
<tr>
<td>RTS</td>
<td>20 bytes</td>
<td>RTS frame length</td>
</tr>
<tr>
<td>CTS</td>
<td>14 bytes</td>
<td>CTS frame length</td>
</tr>
<tr>
<td>Retran\textsubscript{max}</td>
<td>6</td>
<td>maximum retransmission trials</td>
</tr>
<tr>
<td>Fragment\textsubscript{min}</td>
<td>150 bytes</td>
<td>minimum fragmentation size</td>
</tr>
<tr>
<td>(f)</td>
<td>11 Mbps</td>
<td>data transmission rate</td>
</tr>
<tr>
<td>(W)</td>
<td>(22 \times 10^6) Hz</td>
<td>channel bandwidth</td>
</tr>
<tr>
<td>(Pt)</td>
<td>0.281838 W</td>
<td>transmission power level</td>
</tr>
<tr>
<td>(G_t)</td>
<td>1.0</td>
<td>transmitter gain</td>
</tr>
<tr>
<td>(G_r)</td>
<td>1.0</td>
<td>receiver gain</td>
</tr>
<tr>
<td>(h_t)</td>
<td>1.5 m</td>
<td>transmitter height</td>
</tr>
<tr>
<td>(h_r)</td>
<td>1.5 m</td>
<td>receiver height</td>
</tr>
<tr>
<td>(L)</td>
<td>1.0</td>
<td>system loss factor</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>0.125 m</td>
<td>signal wavelength</td>
</tr>
</tbody>
</table>

Table 1. The parameter values used in simulation in addition to the standards values defined in [1].

5.1 Network Topology and Link Error Modeling

For our experiments, we used different topologies each having 49 nodes 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 4.
- **Static Random**: Nodes are immobile and uniformly distributed over the region.
- **Mobile Random**: Nodes are distributed uniformly at random over the region and allowed to move around using the random waypoint model [15] 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 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. For the TCP flows, we used its NewReno variant. The UDP packets and TCP segments were 1500 bytes each. Each of the simulation was run for a fixed duration of 250 seconds including a warm up period of 50 seconds. Transmission flows start in serial with gap of 5 seconds between consecutive flows. Each point in the results is the average of 10 runs. For all the simulations, the energy cost to transmit single bit on a single attempt over a link was chosen to be 60μJ.

All the control packets, e.g., probe packets, RREQ, RREP messages, IEEE 802.11 RT/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 2. 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 \(\mu\) mean and standard deviation \(\sigma\). The noise mean \(\mu\) for the different small square grids was chosen to vary between two configurable parameters, \(N_{\text{min}}\) and \(N_{\text{max}}\) corresponding to minimum and maximum noise respectively, while the noise standard deviations \(\sigma\) was chosen to be equal to \((0.1 \times \mu)W\). We used different distributions for the \(\mu\) over the entire region for different experiments. In this paper, we focus only on the following extreme cases:

1. **Fixed noise environment**: \(N_{\text{min}}\) is equal to \(N_{\text{max}}\) and their values vary between 0.0W and 20.0 \(\times\) \(10^{-11}\)W.
2. **Random noise Environment**: We fix \(N_{\text{min}}\) to 0.0W and vary \(N_{\text{max}}\) between 0.0W and 20.0 \(\times\) \(10^{-11}\)W.

5.2 Metrics

For evaluation, we observed the following metrics:

1. **Average Energy**: Computed per data bit delivered to the destination. It includes energy consumption due to

---

\textsuperscript{4}Note that the PLCP preamble, PLCP header, RTS frame, and CTS frame are sent at the basic access rate.
control packets (e.g. RREQ, RREP messages, IEEE 802.11 RTS/CTS packets etc.) as well as the data packets. The cost of periodic Hello packets is included only in our modified schemes (i.e. RA\textsubscript{fix} and RA\textsubscript{var}). This metric is plotted in the logarithmic scale. Note that we plot the transmission energy cost only and not the reception energy cost since the reception cost is a scale of the transmission cost.

2. **Effective Reliable Throughput**: Counts the number of packets reliably delivered to the destinations.

3. **Average Path Length**: Shows the average number of hops traversed by a data packet.

4. **Average Path Lifetime**: Counts the average time in which a path is active and carry data packets. Time needed for route discovery phase or route maintenance phase is not included in this metric.

5.3 **Static Grid Topologies**

Figure 4. Figures 5 and 6 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 (RA\textsubscript{fix} and RA\textsubscript{var}) show significant benefits. It is interesting to note that for EA and SD schemes, the effective reliable throughput does not decrease monotonically. This is an interesting phenomena that is related to the relative size of the RREQ and the data packets.

To explain this phenomena, consider the flow $A \rightarrow B$ in Figure 4. Both SD and EA schemes try to choose 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 $p$ is constant and depends on the noise value and the received power, but the packet error rate is not. Packet error rate depends on the size of the packets and is smaller for RREQ packets than the data packets. When the noise on the grid is $1.25 \times 10^{-11}$W, the $p$ for the $\langle A, C \rangle$ link is 0.0008. The corresponding packet error rate for RREQ packets is about 0.5. Therefore 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 schemes. However, the packet error rate 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 lower. However, when the noise level increases (i.e., say $1.80 \times 10^{-11} W$), the $p$ on the link goes up (i.e., to 0.00186). This causes the packet error rate 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 shift to paths with shorter hops (which also has lower $p$) and their performance starts to increase again.

The RA schemes do not suffer from this anomalous behavior. This is because the RA schemes choose routes based on the $p$. Therefore, it automatically avoids links with high packet error rates for data packets. Both EA and SD schemes are oblivious of link errors and cannot make such intelligent choices. This behavior is clearly visible in the grid topology since the number of alternative paths are discrete and few. Since the number of path alternatives are discrete and few, RA_var has marginal benefit, both in energy and throughput, over RA_fix at low noise values. At noise values greater than $4.30 \times 10^{-11} W$, RA_fix performance degrades rapidly and faster than RA_var.

Figures 7 and 8 show the corresponding plots for the random noise environment. The EA and SD schemes consume about 140% more energy per successfully transferred data bit than the RA schemes, when the maximum noise in the environment is bigger than $2.50 \times 10^{-11} W$ and still achieves only half the throughput of the RA schemes. Clearly, due to high number of available alternative for route selection, RA_var perform much better than RA_fix scheme. In high noise environments, the RA_var scheme consumes about 77% less energy than RA_fix while maintaining about double the throughput.

Experiments with TCP flows show a similar performance. We show only the case for fixed noise environment in Figures 9 and 10. It is interesting to observe the different behavior of the effective reliable throughput metric for the different schemes (UDP and TCP). For TCP flows, the number of packets transmitted reliably for SD and EA schemes is dropped rapidly to zero for long ranges of noise. The decreasing trend in both these schemes is due to the increasing link error rates with the increase in noise. As the link error rates increase, packets see an increase in end-to-end delays due to the overhead delays spent in the increased number of retransmissions needed to ensure reliability. This indicates the effect of our scheme has impressive effect on the TCP flows more than the UDP flows.

Figures 11 and 12 show the average number of hops per

---

flow for fixed and random noise environment respectively. Both EA and SD schemes produce curves with average number of hops less than those of RA_fix and RA_var. This is because both techniques try to minimize number of hops. Figure 11 shows that RA_var performs better than RA_fix in specific regions of noise. But Figure 12 shows that RA_var outperforms RA_fix for almost all noise regions because of the large number of alternatives for route in random noise environments. In general, decreasing number of hops per flow reduces number of active links, which in turn reduces the number of active nodes. Therefore, RA_var in comparison to RA_fix has the following impacts: (1) reduces the network load, (2) increases the network lifetime, and (3) scales better with number of flows.

Figures 13 and 14 show the average path lifetime per flow for fixed and random noise environment respectively. Note that this is a static topology in which links are not broken due to mobility but only due to dropping frames because of high error rates. Figure 13 shows that the path lifetimes of all schemes are similar to each other except at certain noise values for SD and EA schemes. As explained earlier, SD and EA schemes select short paths of links with high error rates. In this case, packets are dropped due to noise and consequently, AODV layer in the nodes at the receiver side of those links perceive those links as idle. Later, those links time out, and thus break the paths. This behavior occurs in small range values of noise as it appears in the curve notches in Figure 13. In case of random noise environments, SD and EA schemes have more alternatives of short paths of links with high error rates. Therefore, the broken paths behavior occurs more frequently over a wide range of high noise values (contrary to the fixed noise case). This reduces their path lifetime as shown in Figure 14. We experimented our schemes on randomly generated static topologies. Similar results to grids are obtained [19].

5.4 Mobile Topologies

We experimented with different maximum speeds\(^5\): 5, 10, 15, and 20 m/s. In this paper we show the results for the case when the maximum speed of the wireless nodes is 20 m/s. Figures 15 and 16 show the effective reliable throughput and the average energy per reliable delivered data bit respectively in the fixed noise environment. Figures 17 and 18 are the corresponding plots for the random noise environment.

A comparison with the static topologies indicates that mobility reduces reliable data throughput. In particular we also 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 5% lower than the corresponding static topologies. As the channel noise increases (e.g., maximum noise of $3.50 \times 10^{-11}$ W) the data throughput achieved for the mobile topologies is significantly lower (e.g., about 40% less than the corresponding static topologies).

Figures 19 and 20 are the corresponding plots for the TCP flows in a random noise environment. Similar to UDP flows, the RA_var outperforms the other schemes both in energy cost and throughput. Comparing with the UDP flows, the end-to-end delays has a significant effect on the TCP flows. This explains why the TCP throughput goes down faster than the UDP with the increase in the noise environment. As in the static topology, the average number of hops per flow for RA schemes is higher than the other schemes while RA_var maintains shorter paths than RA_fix. This is shown in Figures 21 and 22. Nodes mobility increases the chances of having minimum energy short paths, which explains the large difference between RA_var and RA_fix curves in comparison with the static topology.

The effects of mobility on the path lifetime are shown in Figure 23 and 24 for fixed and random noise environments respectively. The lifetime of the paths degrades gracefully with the increase in the noise level. However, the average path lifetime in both RA_fix and RA_var is larger than the other schemes with an explanation similar to the grid topologies. An interesting observation from the curves is that the path lifetime in RA_var scheme is shorter than the corresponding time in RA_fix scheme which mean the rate of broken paths in RA_var is higher than the rate in RA_fix scheme. An explanation to that is RA_VAR tends to build shorter paths than RA_fix scheme as shown in Figures 21 and 22. Therefore, the average hop distance in RA_var paths is longer than the RA_fix paths and consequently, RA_var paths are more vulnerable to be broken because of node mobility than the RA_fix paths.

6 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 1, 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 [26, 12]. For example, the proposed protocol in [26] is one such variable energy protocol using a modified form of the Bellman-Ford algorithm, where the nodes modify their transmission power based

---

\(^5\)Since our simulations were performed over a relatively short duration of up to 5 minutes, and so we were not affected by the long term slowdown behavior of the random waypoint model [32].
on the distance to the receiver, and where this variable transmission energy is used as the link cost to effectively compute minimum energy routes.

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 idle periods or different durations (e.g., PAMAS [27], SPAN [4], and GAF [29]), or by using heuristics that consider the residual battery power at different nodes [28, 3, 17] and route around nodes nearing battery exhaustion. However, none of the these protocols has considered the link quality and the MAC layer retransmission effect in their computations.

Yarvis et al. [30] observe that hop-count performs poorly as a routing metric for a sensor network, and present the results of using a loss-aware metric. While this metric is likely to use low-loss paths with many hops and doesn’t consider situations where a path with a smaller number of higher loss links would perform better, the cost function in our schemes handles such situation perfectly. A number of existing ad hoc wireless routing algorithms collect per-link signal strength information and apply a threshold to avoid links with high loss ratios ([5], [6], [8], [11], [14], [16]). While this approach may eliminate links that are necessary for connectivity, our method select such links if there is no possible paths. Papers [7] and [2] introduce a method for route selection using metrics accounts for link loss ratios. Authors in [2] assume that each node is aware about the error rates for its outgoing links with no mechanism description about how to acquire such information. They studied the minimum energy reliable communication problem for the standard pro-active routing protocols in static topologies only.

The metric in [7] combines the loss ratios in the two directions over a link. In consequence, the method selects a single path between two nodes regardless of the direction of the communication. This method doesn’t work in situations when the optimum path for one direction is not the same for the other direction. Our cost function consider the cost only on the direction of the communications which allow it to calculate the optimum path on each direction. Another difference, the [7] protocol appends the cost all the links along the route in the route construction packets while our method append only fixed number of values (3 values) regardless of the number of links. Also, they experimented with static topologies only.

None of the above schemes consider the effect of the features provided account for exploiting the features

October 24-27, 2004, Fort Lauderdale, Florida, USA (runner up for Best Paper Award)
provided by the MAC layer as our schemes make use of the fragmentation feature in the IEEE 802.11 MAC layer. Finally, this paper does not assume using of sophisticated hardware to allow variable transmission power levels to minimize energy consumption required to successfully deliver data as in [13] and [24].

7 Conclusion and Future Work

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 and computations, to compute minimum-energy routes, rather than "shortest delay" routes. We showed how our routing computations take into account the link error rates and its IEEE 802.11 retransmission consequences. Also it take into account the cross layer interaction with the IEEE 802.11 layer by exploiting the available fragmentation mechanism in order to increase the reliability. From our description, however, it is obvious that our modifications and techniques can be ported and easily implemented in any alternative on-demand routing protocols (e.g., DSR and TORA).

Our simulation studies show that the energy-aware modification of AODV behavior 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 overhead of our energy-aware route establishment process (e.g., the periodic Hello packets, the forwarding of multiple RREQ and RREP) is more than compensated for by the lower energy consumed in data forwarding. The results, also, show that using packet fragmentation in routing in addition to retransmission cost (RA_var scheme) outperforms the routing with no fragmentation (RA_fix scheme) in terms of energy, throughput, and network load. Although our simulations are conducted using medium scale networks, the performance gains of our schemes will be magnified as the average path length becomes large as in the case of using large scale networks (hundreds or thousands of nodes).

As future work, we will study and compare other mechanisms than periodic Hello packets for link error rates estimation. One possibility is to proceed the route discovery phase by a link error rates estimation phase. With the such on-demand like phase, the overhead of using periodic packets is eliminated in return of additional delays in the route construction phase. Further, in this paper we assumed the mean values of the network noises are fixed during the network life time. Such assumption may not be true in some environments. Therefore, we need to develop mechanisms that discover and redirect the current flow to a new optimum path as soon as it becomes available.

8 Acknowledgments

This work was supported in part by the Maryland Information and Network Dynamics (MIND) Laboratory, its founding partner Fujitsu Laboratories of America, and by the Department of Defense through a University of Maryland Institute for Advanced Computer Studies (UMIACS) contract.

References
