

Error-aware and energy-efficient routing approach in MANETs

Liansheng Tan¹, Peng Yang² and Sammy Chan^{3,*},[†]

¹*Research School of Information Sciences and Engineering, The Australian National University, Canberra ACT 0200, Australia*

²*Department of Computer Science and Engineering, University of Nebraska Lincoln, Lincoln, NE 68588, U.S.A.*

³*Department of Electronic Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong*

SUMMARY

The lifetime of a network is the key design factor of *mobile ad hoc networks* (MANETs). To prolong the lifetime of MANETs, one is forced to attain a tradeoff of minimizing the energy consumption and load balancing. In MANETs, energy waste resulting from retransmission due to high bit error rate (BER) and high frame error rate (FER) of wireless channel is significant. In this paper, we propose two novel protocols termed multi-threshold routing protocol (MTRP) and enhanced multi-threshold routing protocol (EMTRP). MTRP divides the total energy of a wireless node into multiple ranges. The lower bound of each range corresponds to a threshold. The protocol iterates from the highest threshold to the lowest one and chooses those routes with bottleneck energy being larger than the current threshold during each iteration. This approach thus avoids overusing certain routes and achieves load balancing. If multiple routes satisfy the threshold constraint, MTRP selects a route with the smallest hop count to further attain energy efficiency. Based on MTRP, EMTRP further takes channel condition into consideration and selects routes with better channel condition and consequently reduces the number of retransmissions and saves energy. We analyze the average loss probability (ALP) of the uniform error model and Gilbert error model and give a distributed algorithm to obtain the maximal ALP along a route. Descriptions of MTRP and EMTRP are given in pseudocode form. Simulation results demonstrate that our proposed EMTRP outperforms the representative protocol CMMBCR in terms of total energy consumption and load balancing. Copyright © 2008 John Wiley & Sons, Ltd.

Received 20 September 2006; Revised 1 May 2008; Accepted 14 June 2008

KEY WORDS: lossy link; load balancing; IEEE 802.11; energy-efficient routing; MANET

*Correspondence to: Sammy Chan, Department of Electronic Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong.

[†]E-mail: eeschan@cityu.edu.hk

Contract/grant sponsor: Research Grants Council of the Hong Kong Special Administrative Region, China; contract/grant number: CityU 1339/04E

1. INTRODUCTION

Mobile *ad hoc* networks (MANET) consist of a set of autonomous mobile wireless nodes distributed in a certain area forming temporary (*ad hoc*) networks without any infrastructure. Thus, every node may have to serve as the intermediate node to relay the packets between a pair of nodes geographically far enough. Any failure of node may result in disconnection between a pair of communicating nodes.

Wireless devices are often battery powered, which means that power should be used extremely efficiently to maintain the connectivity of the network as long as possible. The energy consumption sources within a wireless device include CPU, monitor, hard disk drive, memory, keyboard/mouse, CD drive, floppy disk drive, wireless interface card, etc. Take a Toshiba 410 CDT mobile computer for example, 8% of the total energy is consumed by the wireless interface card alone [1]. From this figure, the communication-related consumption takes a fairly large portion of the total energy consumption. This calls for the design of energy-efficient routing protocols such as power-aware source routing (PSR) [2], localized energy-aware routing (LEAR) [3], online power-aware routing (OPAR) [4], power-aware localized routing (PLR) [5] and PARO [6].

The energy-efficient routing algorithms proposed in the literature mainly consider factors like total transmission power, residual energy or a combination of them. However, in wireless channels, the channel condition also affects the power consumption. For example, if the bit error rate (BER) of a channel is high, packets are retransmitted more frequently, and hence more energy is consumed for retransmission. Therefore, in this paper, we propose an energy-efficient algorithm that also takes into consideration the channel condition of links when searching for routes.

The organization of this paper is as follows. Section 2 briefly examines the current energy-efficient routing protocols. Section 3 presents the energy model used in the paper. Section 4 proposes our energy-efficient routing protocols termed multi-threshold routing protocol (MTRP) and enhanced multi-threshold routing protocol (EMTRP). Section 5 provides the performance comparisons between CMMBCR and the proposed algorithm EMTRP. Section 6 summarizes the paper and brings out the future work.

2. RELATED WORKS

The study of energy-efficient wireless devices focuses mainly on the following aspects: design of low-power consuming hardware, reduction of the computational complexity to reduce the power consumption by CPU/memory, diminishment of communication-related power consumption. Broadly speaking, communication-related power consumption also includes computation-related power consumption. But in this paper we only focus on the power consumed by pure communication operations. Generally, a wireless node in MANETs resides in one of the following states: transmit, receive and standby, where most of the energy is consumed in the transmission state and least in standby state [7]. So the optimization of transmission power is of ultimate importance [8]. The optimization can be carried out at any layer of the protocol stack. For example, at the physical layer an adjustable transmission range can be implemented based on the distance from the next hop to allow the wireless node to use the minimum energy to transmit packets. This not only preserves the energy but also reduces the interference. At data link layer an efficient sleeping scheme is able to further diminish the power consumption when nodes are idle [7]. At network layer there exist several energy-efficient routing protocols being capable of using energy more efficiently. In

addition, load balancing, that implies to use the energy more evenly and thus prevents certain nodes from being overused is also an important consideration.

Optimization carried out at network layer exists in the following schemes [9]:

1. Minimize the total energy consumed along a route. We denote the energy consumed when transmitting packets between node n_i and n_j as $p(n_i, n_j)$. Thus, the whole route requires energy $P_l = \sum_{i=0}^{D-1} p(n_i, n_{i+1})$, where n_0 and n_D are source and destination, respectively. So the scheme aims to choose the route with minimum energy consumption $P_o = \min_{l \in A} P_l$, where A is the set of all possible routes between the source and destination. We call this scheme *minimum total transmission power routing* (MTPR).
2. Minimize the total battery cost along a route. We define c_i^t to be the residual energy of node n_i at time instant t and $f(c_i^t)$ to be battery cost, which is a decreasing function of c_i^t . Define $r_l = \sum_{i=0}^D f(c_i^t)$ where n_0 and n_D are the source and destination, respectively. This scheme selects a route with minimum total battery cost, that is the route with the total battery cost $r_o = \min_{l \in A} r_l$, where A is the set of all possible routes between the source and destination. We call this scheme *minimum battery cost routing* (MBCR).
3. The third scheme tries to avoid using the node with minimum residual energy. Here the cost of a route is not defined as the sum of individual battery cost of each node, but as the maximum battery cost along that route. This scheme chooses a route with minimum cost. This scheme always tries to avoid using the node with minimum residual energy and thus improves the load balance among all the nodes. But without consideration of the total transmission energy along a route, it may choose a route with large total transmission power. We refer to this scheme as *min-max battery cost routing* (MMBCR).
4. The fourth scheme combines the advantages of both MTPR and MMBCR. It chooses a route whose bottleneck residual energy is larger than a certain threshold. If there is more than one route satisfying this condition, then it selects the one with the minimum total transmission power, as in the case of MTPR. When no route satisfies the condition, similar to MMBCR, it chooses a route with the minimum cost. We call this scheme *conditional min-max battery cost routing* (CMMBCR).

In order to operate appropriately, these schemes require additional information besides route information. For example, to compute the transmission power, the distance between two adjacent nodes on a route is needed. Considering the mobility of MANET, this incurs a certain amount of overhead. The residual energy of every node on a route is required as well, which also brings in overhead. In spite of the overhead, the above schemes and our proposal are still attractive in terms of their merit in prolonging the lifetime of MANET.

All these energy optimization schemes are built upon some existing routing protocols. Since all these schemes require that the source knows all the available routes to destination, usually source routing protocols, such as dynamic source routing (DSR) [10], are adopted. Other related routing protocols include [11, 12].

3. ENERGY MODEL

We assume a homogeneous network in which all wireless nodes have the same wireless interface card, thus the same radio characteristics. We also assume a simple model for the radio hardware

energy dissipation where the sender dissipates energy to run the radio electronics and the power amplifier and the receiver consumes the energy to run the radio electronics. Thus, sender requires an energy of

$$\begin{aligned} E_{Tx}(l, d) &= E_{Tx\text{-elec}}(l) + E_{Tx\text{-amp}}(l, d) \\ &= lE_{\text{elec}} + l\varepsilon_{\text{fs}}d^2 \end{aligned} \quad (1)$$

to send a packet and the receiver requires an energy of

$$E_{Rx}(l) = E_{Rx\text{-elec}}(l) = lE_{\text{elec}} \quad (2)$$

where E_{elec} is the energy dissipated per bit to drive the radio electronics, ε_{fs} is constant depending on frequency and receiving threshold d is the distance between the sender and the receiver and l is the packet length. We assume that $E_{\text{elec}} = 50 \text{ nJ/bit}$, $\varepsilon_{\text{fs}} = 12 \text{ pJ/bit/m}^2$.

In addition, we assume a MAC protocol with a four-frame exchange process and a power control scheme. In the four-frame exchange process, nodes exchange two additional frames before transmitting data packets: RTS and CTS. More specifically, the sender advertises a RTS packet after the channel is available for a period longer than DIFS or the backoff timer reaches zero. The receiver responds with a CTS packet after receiving a RTS packet. If the CTS is not received within a predetermined time interval, the sender retransmits the RTS packet. When the sender receives RTS, it starts to transmit the data packet and waits to receive the ACK. If the ACK fails to arrive, the whole process is repeated again. As with the power control scheme, in this scheme, RTS and CTS are transmitted using the maximum transmission range. DATA and ACK packets are transmitted using the minimum required transmission range.

4. THE PROPOSED ALGORITHM

4.1. Algorithm for perfect channel

Inspired by the previous work such as CMMBCR, we propose our algorithms termed MTRP for the case of perfect channel. CMMBCR is energy-saving due to the set threshold. However, when all routes do not satisfy this threshold, it reduces to MMBCR. That is, it may use a route, which may require more total transmission power. In MTRP, multiple thresholds are introduced to achieve the tradeoff between balancing of energy consumption and the minimum total transmission power. The main idea here is to classify all the available routes according to their bottleneck energy, i.e. the minimum residual energy among the nodes traversed by the route, and divides the total energy of a wireless node into multiple ranges. The lower bound of each range corresponds to a threshold. The protocol iterates from the highest threshold to the lowest one and chooses those routes with bottleneck energy being larger than the current threshold. If multiple routes are found, then a route with the minimum hop count is chosen. If no route satisfies the current threshold, then the current threshold is lowered and the search is carried out again. Here we assume that all mobile nodes have the same initial energy. Thresholds are set so that the initial energy is evenly divided. For example, suppose the initial energy is 100 and there are 10 thresholds, then the thresholds are 0, 10, 20, 30, 40, 50, 60, 70, 80 and 90. Initially, we set the threshold to 90 and only routes with bottleneck energy larger than 90 are regarded as candidate routes. Among these routes, the one with the smallest hop count is finally chosen. If there is no route with bottleneck energy larger

Table I. Algorithm of MTRP.

Algorithm of MTRP

```

threshold(level): array of thresholds in ascending order
n: the index of current threshold
bottleneck(i): the minimum residual energy of route i
length(i): the hop count of route i

minlength=MAXLENGTH
found=false
n=level-1
while n>=0
  for all available route i do
    e=bottleneck(i)
    if e>threshold(n) then
      found=true
      l=length(i)
      if l<minlength then
        minlength=l
        route=i
      endif
    endif
  endfor
  if found==true then
    break
  endif
  n=n-1
endwhile
return route

```

than 90, the threshold is lowered to 80 and the whole process is repeated again. The pseudocode of MTRP is described in Table I.

Why do we choose the minimum hop count as our energy-saving strategy? As in the indoor environment the energy consumed by radio electronics either to transmit or receive packets takes up a fairly large part of the total energy consumption due to the short transmission distance between two adjacent nodes and this consumption is independent of the distance, but are proportional to the number of hops, which advocate the shortest hop count routing. In addition, the shortest hop count routing also has the smallest overhead to transmit and receive control packets such as RTS, CTS and ACK. So choosing minimum hop counts can closely approximate minimum total transmission power strategy in indoor environment without incurring much overhead. In addition, the smallest-hop-count scheme also brings other benefit such as lower delay, better stability, etc.

4.2. Algorithm for lossy channel

The radical difference between wired and wireless networks is that wireless channel is lossier, which makes packet retransmission more frequent than that in wired network. Packet retransmission certainly wastes already scarce energy resource. So being a circumspect energy-efficient routing protocol, it must take into account the condition of channels on a route and choose a route with better channel condition. To demonstrate this issue, we conduct an experiment to show how big the channel error impacts the energy consumption. A chain of eight nodes is adopted here as the

simulation topology. The distance between every adjacent pair of nodes is 50 m, which ensures that no separate nodes can communicate directly under the parameters of wireless interface card listed in Table II.

A packet flow traverses from node 0 to node 7 as shown in Figure 1. The experiment is conducted under two conditions. One is the perfect channel, one is the lossy channel. For lossy channel, we assume that four channel conditions are randomly distributed to each wireless channel with the BERs and the packet loss probabilities of different kinds of packets listed in Table III. The packet loss probabilities of RTS, CTS, DATA, ACK are determined by the BER and the length, respectively, under a AWGN channel model. Fifty experiments are conducted and the average is calculated for comparison.

The simulation result is presented in Figure 2. From the figure, it is obvious that when channel is lossy, large portion of energy is consumed by the packet retransmission process.

Having confirmed the impact due to channel errors, an enhanced version of MTRP, enhanced MTRP (EMTRP), is proposed. Before presenting the algorithm, we need a metric to describe the error characteristics of a channel. Here, average frame error rate (FER) is used for this purpose. We use two different models to generate error packets. One is a uniform error model; the other is a Gilbert error model [13]. Other related error models can also be employed, such as hierarchical Markov model [14], Markov-based trace analysis (MTA) [15]. Besides various error models, in practical environment, the wireless nodes can also measure the error probability of a channel regularly and use the latest two measurements to calculate the weighted sum over a time period [16], or use some prediction techniques such as the one described in Section 4.3.

A uniform error model is simple. It has only one parameter i.e. average FER. When the parameter is specified, it can generate error packet according to uniform distribution. Gilbert model, as shown

Table II. Parameters of wireless interface card.

Parameters	Description
RXThresh=9.3266e-16	Receiver's threshold
CSThresh=9.3266e-17	Carrier sense threshold
CPTresh=10.0	Capture threshold
pathlossExp=2.0	Path loss exponent
freq=2.472e9	Radio frequency
Pt=3.1360e-8	Maximum transmission energy



Figure 1. Chain pattern.

Table III. Channel conditions.

BER	RTS (44B)	CTS (38B)	DATA (1212B)	ACK (38B)
0.0	0.0	0.0	0.0	0.0
1e-6	0.0004	0.0003	0.0096	0.0003
1e-5	0.0035	0.003	0.0924	0.003
5e-5	0.0174	0.0151	0.3842	0.0151

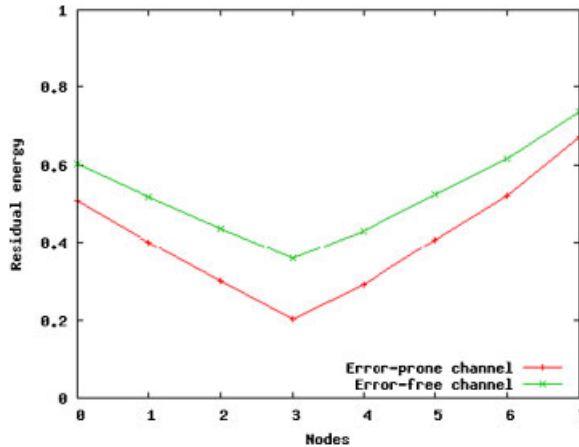


Figure 2. Residual energy comparison between lossy and perfect channel.

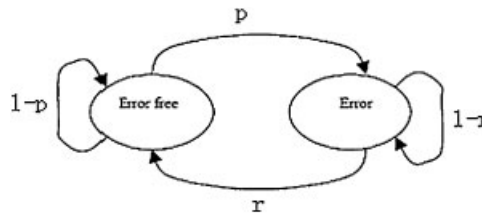


Figure 3. Gilbert error model.

in Figure 3, is essentially a two-state Markov chain. One of the states is error state; the other is error-free state. When the process is in the error state, the probability that a packet gets corrupted is high; while in error-free state packets never get corrupted or get corrupted with a low probability. The channel remains in each state for a period of time and then transfers to another state. Let p denote the transition probability from error-free state to error state, r denote the transition probability from the error state to error-free state.

The enhancement made in EMTRP is that it takes the channel condition into account when selecting the routes. We assume that no error correction mechanism is used. Thus, when a packet is corrupted, it is lost. So the average loss probability (ALP) and average error probability are used interchangeably.

To incorporate channel condition, we assign each wireless channel a price that is a function of channel's ALP. A choice is to use the average energy consumption of one hop transmission under the packet loss as the metric and choose a route with smallest total energy consumption. However, this requires the sending node to know the distance between itself and its neighbor which is not very worth for the same reason discussed in Section 4.1. So, we use the expected number of transmission of that channel as its price. The expected number of transmission is the transmission attempts of all packets including the control packets in order to transmit a data packet to its downstream node. Under the energy model we use, this metric is a close approximation of average energy consumption. In order to calculate the expected number of transmission, we formulate the four-frame exchange process in the state diagram of Figure 4.

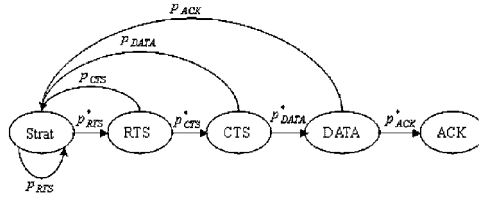


Figure 4. The state diagram of the MAC layer.

In the state diagram, P_{RTS} , P_{CTS} , P_{DATA} , P_{ACK} are the ALPs for RTS, CTS, DATA, ACK packets, respectively, and $P^* = 1 - P$. From the state diagram, we have the expected number of transmission given by the following equation [17]:

$$E_t = \frac{1}{P_{RTS}^* P_{CTS}^* P_{DATA}^* P_{ACK}^*} + \frac{1}{P_{CTS}^* P_{DATA}^* P_{ACK}^*} + \frac{1}{P_{DATA}^* P_{ACK}^*} + \frac{1}{P_{ACK}^*} \quad (3)$$

From (3) the transmission attempt is a function of the channel's ALP. A large loss probability will create a large expected number of transmission and according to the state diagram shown in Figure 4, this means a larger number of four-frame exchange process. In the light of the guideline in Section 4.1, the expected number of transmission can be used as a reliable link price.

Having defined the channel price, we can select a route with small aggregate price in the route cache. The pseudocode of EMTRP is described in Table IV.

Note that the function $price(i)$ returns the aggregate price of i . In order to obtain $price(i)$, the ALP is needed. For uniform error model, ALP is directly specified as an explicit parameter and is easily obtained. For Gilbert error model, however, ALP is not explicitly specified. Assume that the error-free period is t_{good} , error period is t_{bad} , loss probability in each state in Figure 3 is p_{good} and p_{bad} , respectively. The transition matrix R is given by

$$R = \begin{pmatrix} 1-p & p \\ r & 1-r \end{pmatrix} \quad (4)$$

So the duration of error state is a geometric random variable with mean $(1/r) \times t_{bad}$ and similarly the mean time in error-free state is $(1/p) \times t_{good}$. The steady-state probability of the channel being in error state is given by

$$\pi_{bad} = \frac{\frac{1}{r} \times t_{bad}}{\frac{1}{r} \times t_{bad} + \frac{1}{p} \times t_{good}} \quad (5)$$

The steady-state probability of the channel being in error-free state is given by

$$\pi_{good} = \frac{\frac{1}{p} \times t_{good}}{\frac{1}{r} \times t_{bad} + \frac{1}{p} \times t_{good}} \quad (6)$$

Table IV. The pseudocode of EMTRP.

Algorithm of EMTRP

threshold(level): array of thresholds in ascending order
 n : the index of current threshold
bottleneck(i): the minimum residual energy of route i
length(i): the hop count of route i
price(i): the aggregate price of route i

```

minlength = MAXLENGTH
minprice = MAXPRICE
found = false
n = level - 1
while n ≥ 0
  for all available route i do
    e = bottleneck(i)
    if e > threshold(n) then
      found = true
      l = length(i)
      p = price(i)
      if p < minprice or (p == minprice and
        l < minlength) then
        minlength = l
        minprice = p
        route = i
      endif
    endif
  endfor
  if found == true then
    break
  endif
  n = n - 1
endwhile
return route

```

From Equations (5) and (6), the steady-state error rate or equivalently ALP is given by

$$\varepsilon = p_{\text{bad}} \times \pi_{\text{bad}} + p_{\text{good}} \times \pi_{\text{good}} \quad (7)$$

If $p_{\text{good}} = 0$, then (7) can be simplified to

$$\varepsilon = p_{\text{bad}} \times \pi_{\text{bad}} \quad (8)$$

Note that, MTRP is actually a special case of EMTRP when the channel condition is perfect. In this case, $p_{\text{RTS}}^* = p_{\text{CTS}}^* = p_{\text{DATA}}^* = p_{\text{ACK}}^* = 1$. The expected number of transmission of one hop $E_t = 4$, i.e. four-frame exchange. So minimizing the route price amounts to minimizing the hop count.

4.3. Overhead analysis and implementation issues

In the perfect channel case, the overhead of the proposed algorithms is less than that of CMMBCR, because only residual energy information is required. The distance-related information is eliminated, since we use minimum hop count as the metric instead of total transmission energy.

For lossy channel, besides residual energy, the ALP is also required. Like distance, ALP must be monitored periodically. But the overhead is comparable to that of CMMBCR in this case.

In DSR, the routing information is stored in a route cache at the source and is initialized during the route discovery phase. When a packet is about to be sent, the routing information stored in the route cache together with other information, i.e. residual energy and ALP, are extracted as the input, the proposed algorithms are then performed.

In practical environment when the error characteristic cannot be modeled well by a predefined error model, channel condition can be monitored by a channel estimator. A straightforward method is to monitor packet loss by attaching a sequence number to each packet and counting packets lost in the channel. Then the channel estimator can receive this information for error probability calculation and feedbacks it to the source.

We adopt a receiver-initiating feedback mechanism in which the receiver can periodically send probing packets along some routes in its route cache. The probing packet records the channel condition and node's residual energy at each hop. The nodes on that the route may or may not extract out the information stored in the probing packet depending on whether it is a sender who sends packets to that receiver. If it is, then the sender knows exactly the channel condition and bottleneck energy of the routes to the receiver in its own route cache. It sounds strange at first because the probing packets are sent along the routes in the receiver's route cache, how does the sender know the channel condition and bottleneck energy of the routes in its own route cache? Since in DSR, the route in sender's route cache is also in the receiver's route cache but in a reverse order (here we assume that all wireless links are symmetric links), thus this scheme works. In addition, if the transport layer protocol used by the nodes are ACK-enabled protocol, then ACK can be used for probing if there are some ACK packets waiting for sending, otherwise the receiver sends probing packet.

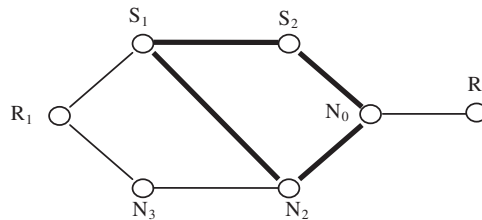


Figure 5. Example topology of feedback mechanism.

Table V. Routing tables.

Routing table of S_1	
$S_1 \rightarrow S_2 \rightarrow N_0$	
$S_1 \rightarrow N_2 \rightarrow N_0$	
Routing table of S_2	
$S_2 \rightarrow N_0$	
Routing table of N_0	
$N_0 \rightarrow S_2 \rightarrow S_1 \rightarrow R_1$	*
$N_0 \rightarrow N_2 \rightarrow N_3 \rightarrow R_1$	
$N_0 \rightarrow N_2 \rightarrow S_1 \rightarrow R_1$	*
$N_0 \rightarrow R_2$	

Take the topology in Figure 5 for example. In the figure, node S_1 and S_2 send packet to node N_0 . N_0 sends packet to R_1 and R_2 . The routing table of each sender is shown in Table V. We now discuss how N_0 sends the probing packets. First, N_0 checks each route in its route cache to see whether this route contains the sender who sends packets to it. If the route contains at least a sender it sends a probing packets on that route. In this example, N_0 will send probing packet on route 1 and route 3, which are marked with asteroids. The destination field of each probing packet is set to the farthest sender on that route. So the probing packet on route 1 has the destination field of S_1 . This way of setting the destination field can eliminate extra energy consumption that could occur if we set the destination field to the end of each route. The routes on which N_0 sends probing packets are thickened in Figure 5, and from the thick lines we know that the sender S_1 and S_2 have all necessary information including the channel conditions and bottleneck energy of each route in their route caches, respectively.

5. SIMULATION RESULTS

In simulations, we assume that 49 wireless nodes are randomly distributed over an $130\text{m} \times 130\text{m}$ area with a random waypoint mobility model. Each node initiates a transmission session with another randomly chosen node with a random probability. This means that only some nodes can initiate transmission session. We conduct a series of simulations using the ns2 simulator [18]. The simulation parameters are those shown in Tables II and III. Note that Table III shows the ALPs of four channel conditions that are randomly distributed to each receiving channel. For uniform error model, the ALP is the only parameter required. For Gilbert error model, we use the parameters of Table VI and the transition matrix of Equation (9) to achieve these ALPs

$$R = \begin{pmatrix} 0.9 & 0.1 \\ 0.8 & 0.2 \end{pmatrix} \quad (9)$$

Two metrics are compared, i.e. residual energy, node death speed. Node death speed traces how many nodes are still alive after certain time. From load balancing perspective, the number of nodes alive after certain time should remain as large as possible, thus the upper curve in the diagram indicates better load balancing than the curve below it.

Since MTRP is only a special case of EMTRP, only EMTRP and CMMBCR are compared.

First, the residual energy of two schemes is compared. The simulation time is 100 s. We assign each node of an initial energy of 3.0 to ensure that no node is dead after simulation ends. The threshold for CMMBCR is set to 1.5. There are 10 levels in EMTRP, i.e. 2.7, 2.4, ..., 0.0. The results for two error models are shown in Figures 6 and 7. From the figures we can see that residual energy of two schemes is comparable. The reason is that although we consider the channel condition in

Table VI. Parameters for Gilbert error model.

t_{bad}	t_{good}	p_{bad}	p_{good}
N/A	N/A	0	0
0.01	0.01	$7.2\text{e}-5$	0
0.03	0.03	$7.2\text{e}-4$	0
0.04	0.04	$3.6\text{e}-3$	0

EMTRP, CMMBCR uses energy more efficiently by choosing the route with smallest transmission energy consumption from all routes whose bottleneck energy is larger than the threshold. While in EMTRP, when the bottleneck energy of some good routes is smaller than the current threshold, we are forced to choose some suboptimal routes to achieve the load balancing effect. However, as we can see in Figure 7, the residual energy of all nodes when using EMTRP is still larger than that of CMMBCR.

Next, the node death speed is compared. The simulation time is 100 s. We assign each node of an initial energy of 1.0 to ensure that large part of nodes is dead after simulation ends. The threshold for CMMBCR is set to 0.5. There are 10 levels in EMTRP, i.e. 0.9, 0.8, ..., 0.0. The results are shown in Figures 8 and 9, respectively. As Figure 8 shows, when EMTRP is used, the node death speed is much lower than using CMMBCR. Also, when simulation ends, there are 25

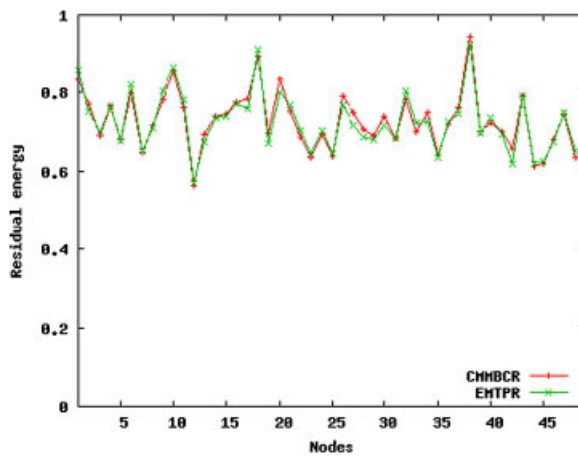


Figure 6. Residual energy (uniform error model).

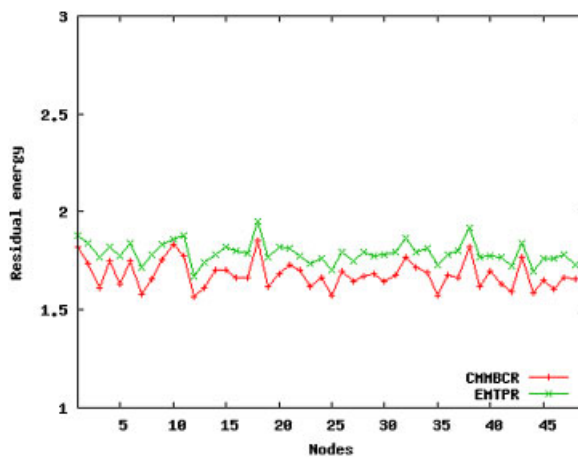


Figure 7. Residual energy (Gilbert error model).

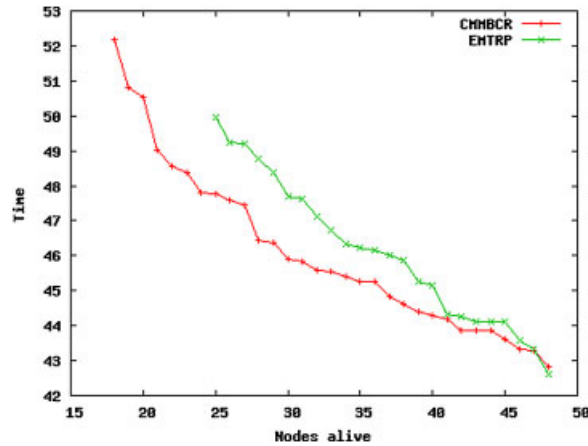


Figure 8. Node death speed (uniform error model).

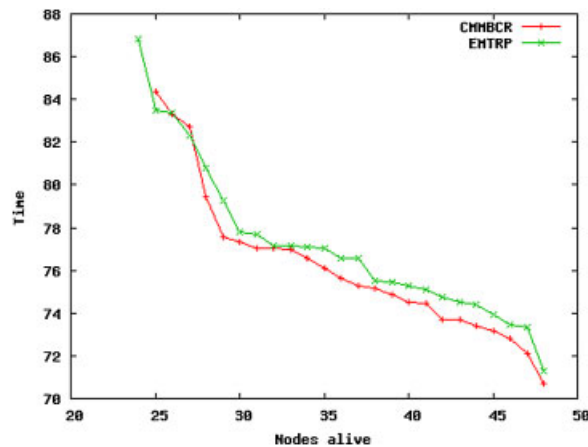


Figure 9. Node death speed (Gilbert error model).

nodes still alive using EMTRP, while there are 17 nodes alive using CMMBCR. Although when Gilbert error model is used, the advantage of EMTRP is not as obvious as the situation under uniform error model, there is still some improvement.

6. CONCLUSIONS AND FUTURE WORK

This paper has proposed MTRP and its enhanced version EMTRP that achieve a tradeoff between energy efficiency and load balancing. Through simulations, we have found that such multi-threshold schemes consume energy in a more balanced manner. Also the error-aware feature of EMTRP helps to reduce the energy consumption caused by retransmission of packets due to high FER of wireless channel.

It is still an interesting topic for future research of setting the number of energy levels. If there are too many thresholds, then the number of available routes, which satisfy the current threshold, will reduce quickly and leads to a relatively suboptimal selection in terms of minimizing the energy consumption. If there are too few thresholds, the benefit of load balancing will diminish. Therefore, in future study, we will use fuzzy-based algorithms to dynamically adjust the number of thresholds.

ACKNOWLEDGEMENTS

This work was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 1339/04E).

REFERENCES

1. Jones CE, Sivalingam KM, Agrawal P, Chen JC. A survey of energy efficient network protocol for wireless networks. *Wireless Networks* 2001; **7**:343–358.
2. Maleki M, Dantu K, Pedram M. Power-aware source routing protocol for mobile *ad hoc* networks (ISLPED '02). presented at *Proceedings of the 2002 International Symposium on Low Power Electronics and Design*, Monterey, CA, U.S.A., 2002.
3. Woo K, Yu C, Lee D. Non-blocking, localized routing algorithm for balanced energy consumption in mobile *ad hoc* networks. presented at *International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2001)*, Cincinnati, OH, U.S.A., 2001.
4. Li Q, Aslam JA, Rus D. Online power-aware routing in wireless *ad hoc* networks. *Mobile Computing and Networking* 2001; 97–107.
5. Stojmenovic I, Lin X. Power-aware localized routing in wireless networks. *IEEE Transactions on Parallel and Distributed Systems* 2001; **12**:1122–1133.
6. Gomez J, Campbell AT, Naghshineh M, Bisdikian C. Conserving transmission power in wireless *ad hoc* networks. *Ninth International Conference on Network Protocols*, Riverside, CA, U.S.A., 2001.
7. Singh S, Raghavendra CS. PAMAS-power aware multi-access protocol with signaling for *ad hoc* networks. *ACM SIGCOMM* 1998; **28**:5–26.
8. Krunz M, Muqattash A, Lee SJ. Transmission power control in wireless *ad hoc* networks: challenges, solutions, and open issues. *IEEE Networks* 2004; **18**(5):8–14.
9. Toh CK. Maximum battery life routing to support ubiquitous mobile computing in wireless *ad hoc* networks. *IEEE Communication Magazine* 2001; **39**:138–147.
10. Johnson DB, Maltz DA. Dynamic source routing in *ad hoc* wireless networks. In *Mobile Computing*, Imielinskiand T, Korth H (eds). Kluwer Academic: Dordrecht, 1996; 153–181.
11. Chakeres I, Belding-Royer E, Perkins C. Mobile *ad hoc* networks working group. *Dynamic MANET on-demand routing*. draft-ietf-manet-dymo-03, Internet Draft, July 2005 (<http://moment.cs.ucsb.edu/pub/draft-ietf-manet-dymo-00.txt>).
12. Clausen T. The optimized link-state routing protocol. *Internet Draft*, August 2005 (<http://bgp.potaroo.net/ietf/ids/draft-ietf-manet-olsrv2-00.txt>).
13. Gilbert EN. Capacity of a burst-noise channel. *Bell System Tech Journal* 1960; **39**:1253–1266.
14. Khayam S, Karande S, Radha H, Loguinov D. Performance analysis and modeling of errors and losses over 802.11b LANs for high-bitrate real-time multimedia. *Signal Processing: Image Communication* 2003; **18**(7): 575–595.
15. Konrad A, Zhao BY, Joseph AD, Ludwig R. A Markov-based channel model algorithm for wireless networks. *Wireless Networks* 2003; **9**(3):189–199.
16. Jay BT. Monitoring link quality in a mobile *ad hoc* network. *U.S. Patent 6894985*, 17 May 2005.
17. Gardener WA. *Introduction to Random Process*. McGraw-Hill: New York, 1990.
18. USC/ISI, Los Angeles, CA. The NS simulator and the documentation [Online]. Available from: <http://www.isi.edu/nsnam/ns/>.

AUTHORS' BIOGRAPHIES



Liansheng Tan is now with Research School of Information Sciences and Engineering, The Australian National University, Canberra ACT 0200, Australia. Dr Tan received his PhD degree from Loughborough University in the U.K. in 1999. He has done research in computer communication network in School of Information Technology and Engineering at University of Ottawa, Ontario, Canada as a postdoctoral research fellow and a visiting research scientist in 2001. During 2002 till 2006, he worked at the Department of Computer Science, Central China Normal University, China as a professor and Head of Department. During that period, he also held a number of visiting research positions at Loughborough University, University of Tsukuba, City University of Hong Kong and University of Melbourne. Dr Liansheng Tan is currently an Associate Editor of *Dynamics of Continuous, Discrete & Impulsive Systems (Series B: Applications & Algorithms)*, he is also on the Editorial Board of *International Journal of Communication Systems*. He has published over 70 referred papers in international journals and conferences. His research interests include modeling, congestion control analysis and performance evaluation of computer communication networks, resource allocation and management of wireless and wireline networks and routing and transmission control protocols.



Peng Yang has received his Bachelor and Master degree in Central China Normal University, PR China and now is pursuing his Doctoral degree in the University of Nebraska Lincoln, U.S.A. His research interests include wireless communication, transmission control protocols, utility-based rate control.



Sammy Chan received his BE and MEngSc degrees in Electrical Engineering from the University of Melbourne, Australia, in 1988 and 1990, respectively, and a PhD degree in Communication Engineering from the Royal Melbourne Institute of Technology, Australia, in 1995. From 1989 to 1994 he was with Telecom Australia Research Laboratories, first as Research Engineer, and between 1992 and 1994 as a Senior Research Engineering and Project Leader. Since December 1994, he has been with the Department of Electronic Engineering, City University of Hong Kong, where he is currently an Associate Professor.