HAL
open science

Blocking Expanding Ring Search Algorithm for Efficient Energy Consumption in Mobile Ad Hoc Networks

Incheon Park, Jinguk Kim, Ida Pu

To cite this version:

Incheon Park, Jinguk Kim, Ida Pu. Blocking Expanding Ring Search Algorithm for Efficient Energy Consumption in Mobile Ad Hoc Networks. WONS 2006 : Third Annual Conference on Wireless On-demand Network Systems and Services, INRIA, INSA Lyon, IFIP, Alcatel, Jan 2006, Les Ménuires (France), pp.191-195. inria-00001023

HAL Id: inria-00001023

https://inria.hal.science/inria-00001023

Submitted on 30 Jan 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Blocking Expanding Ring Search Algorithm for Efficient Energy Consumption in Mobile Ad Hoc Networks

Incheon Park, Jinguk Kim, Ida Pu
Department of Computing, Goldsmiths College
University of London, London SE14 6NW, UK
Email: {map01ip, ma201jgk, i.pu}@gold.ac.uk

***Abstract*— Efficient energy consumption is one of the important issues in Ad Hoc Networks. This paper identifies the inefficient elements during the route discovery in well known reactive protocols such as DSR and proposes the new *Blocking-ERS* approach, which demonstrates a substantial energy efficiency at small time expense in comparison to conventional Expanding Ring Search methods.**

I. Introduction

Mobile Ad-Hoc Networks (MANETs) provide an alternative model of communication to conventional wired networks during malfunction or absence of the fixed wired network[1]. They have shown the potential of making significant impacts in situations where the physical infrastructure was either dysfunctional or unavailable, for example, in emergency rescue operations, and in combat zones.

In MANETs, nodes cooperating for delivery of a successful packet form a communication chanel consisting of a *source*, a *destination* and possibly a number of *intermediate nodes* without any fixed base station. Each node communicates directly with the destination or neighbour intermediate nodes within wireless transmission range.

MANETs are hardly realised in practice unless energy efficient communication are in place due to limited battery life of mobile devices. Protocols determine how the nodes communicate and the existing protocols can be modified to become more energy efficient. It is known that the communication between nodes consumes substantial amount of energy in wireless networks [2].

This paper examinates inefficient elements in well known reactive protocols such as DSR [1] and AODV [3] and proposes a new approach for rebroadcasting in Expanding Ring Search. This leads to the *Blocking-ERS* scheme, as we call it, which demonstrates improvement in energy efficiency at the expense of route discovery time in comparison to conventional ERS. We assume that routing protocol can be carried out much faster than the topological changes [4].

The rest of this paper is organised as follows. Section II shortly discuss the existing Expanding Ring Search. Section III describes the general behaviour of TTL sequence-based expanding ring search. Section IV outlines the proposed approach in terms of time delay and energy consumption. Section V discusses the results of this research along with future directions.

II. Expanding Ring Search

Reactive routing protocols in MANETs such as DSR and AODV are often supported by a so-called *Expanding Ring Search* (ERS for short). During the route discovery stage, the RREQ (Route REQest packet) is broadcasted by flooding and propagated from one intermediate node to another to find the route information from the source to the destination node. Figure 1, 2 and 3 show how the broadcasts and propagation form searching 'rings' in such a route discovery process.

In Figure 1, a RREQ is broadcasted by the source and two neighbour intermediate nodes receive the message. Each arrow line represents a send-receive relationship between the broadcasting source and one neighbour node. Each broadcast is issued with a *hop number* which is a serial number indicating the sequence of the nodes alone a route from the source. In Figure 2, for example, a RREQ is broadcasted with a hop number '1' by the source and five intermediate nodes receive the message and are given the same hop number. If none of them has the route information to the destination node, the five nodes rebroadcast the RREQ with an incremental hop number, 2 in Figure 3, for example. In this way, the nodes with the same hop number from the source node form a circle, i.e. the search rings. As the route discovery in progress, the diameter of the searching ring increases.

Utilising the route cache during routing is widely adopted in MANETs to achieve time and energy saving routing. A node in MANETs broadcasts RREQs and the intermediate nodes within the broadcast range cooperate the route discovery process by checking their own route caches for requested route

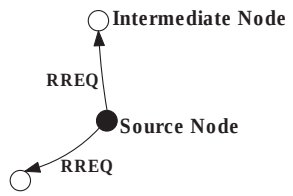

Fig. 1. Propagation of RREQs

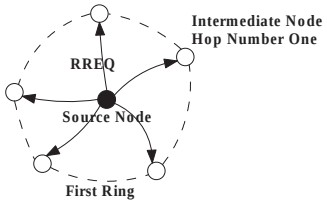

Fig. 2. Formation of a ring

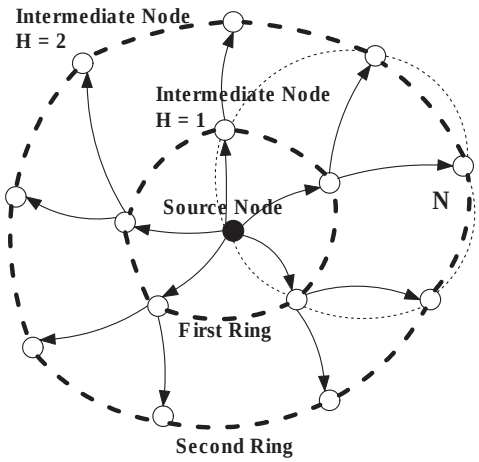

Fig. 3. Expanding search ring

information and maintaining an updated list of known routes. An intermediate node who has the requested route information towards the destination is defined as a *route node* in this paper.

Efficient route discovery relies largely upon how quickly a route node can be found. When a RREQ is received, an intermediate node searches for the requested route information in its route cache. If there is route information to the destination node in its route cache, the route node would stop rebroadcasting the RREQ and sends a RREP to the source node with the complete route information consisting of the cached route in itself and the accumulated route record in the RREQ. In this way, the route may be established much quicker and the total delivery time and energy consumption are saved.

III. TTL Sequence-Based Expanding Ring Search

The goal of the expanding ring search is to find nodes that have the required route information to the destination node in their route cache by propagating RREQs. The propagation of RREQs is, on one hand, an efficient way for route discovery. It, on the other, may lead to ineffective flooding. Nodes in MANETs, for example, can be trapped in a loop of actions and end up asking each other for routing information for a long time without any solutions.

To control the flooding in MANETs, TTL (Time To Live) [1] sequence-based Expanding Ring Search is frequently used. The TTL sequence-based ERS restrains its searching range by giving RREQs a pre-defined TTL number. The TTL number corresponds to the radium of a searching area. Each time it fails to find any node that has route information to the destination node or the destination node itself, the source node rebroadcasts the RREQ in the next round with an increased TTL number to allow the RREQ to reach the remote nodes in further distance.

The TTL number increases linearly with a specified value [5]. The incremental value of TTL is fixed to 2 in [3] and [6], 1 in [7]. Figure 4 shows how TTL controls the RREQ relay range by predefined TTL values of 1, 2 and 3.

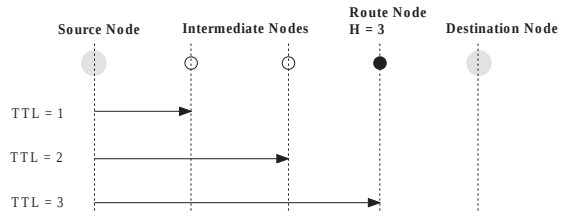

Fig. 4. TTL sequence-based ERS

Historically, TTL sequence-based mechanism was broadly adopted to reach all nodes in the networks to ensure successful route discovery in one round of flooding. Later an optimally chosen set of TTL was introduced to solve the generic minimal cost flooding search problem. However, it has been shown that there is no significant advantage from the optimal TTL sequence-based route discovery compared to the basic route discovery mechanism in terms of route discovery enhancement [8]. In addition, optimal TTL sequence-based discovery causes longer delay than the basic route discovery mechanism.

As it can be seen from Figure 4, in TTL sequence-based ERS, the source node issues and rebroadcasts a RREQ with increased TTL number if no RREP is received after a certain amount of time. The route discovery procedure repeats until the whole network is covered or until it finds the nodes that have the route information to the destination node or the destination node itself. This can also cause the overheads over the network area. More energy will inevitably be consumed during the route discovery if the route discovery process starts from the source node every time. It is especially costly when a larger area of the network needs to be searched.

IV. Blocking-ERS

We propose an alternative ERS scheme to support reactive protocols such as DRS and AODV, and it is called *Blocking Expanding Ring Search* (*Blocking-ERS* for short).

The Blocking-ERS integrates, instead of TTL sequences, a newly adopted control packet, *stop_instruction* and a hop number (H) to reduce the energy consumption during route discovery stage. We derive new route discovery algorithms (Algorithm 1, 2 and 3 for the source, intermediate and destination node respectively, and Algorithm 4 for blocking procedure).

The basic route discovery structure of Blocking-ERS is similar to that of conventional TTL sequence-based ERS. One of the differences from TTL sequence-based ERS is that the Blocking-ERS does not resume its route search procedure from the source node every time when a rebroadcast is required. The

rebroadcast can be initialised by any appropriate intermediate nodes. An intermediate node that performs a rebroadcast on behalf of the source node acts as a *relay* or an *agent* node. Figure 5 shows an example of our Blocking ERS approach in which the rebroadcasts are initialised by and begins from a relay node M in rebroadcast round 2, and another relay node N in round 3, and so on.

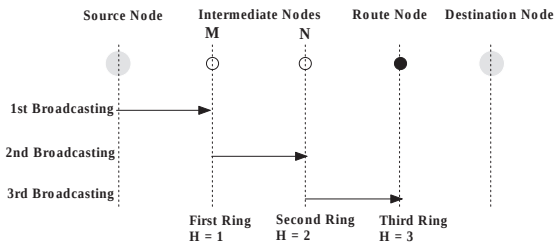

Fig. 5. Blocking-ERS

In Figure 5, the source node broadcasts a RREQ including a hop number (H) with an initial value of 1. Suppose that a neighbour M receives the RREQ with H=1, and the first ring was made. If no route node is found, that is, no node has the requested route information to the destination node, the nodes in the first ring rebroadcast the RREQ with an increased hop number, for example, RREQ with H=2 is rebroadcast in this case. The ring is expanded once again just like the normal expanding ring search in DSR [1] or AODV [9] (see also Figure 3), except with an extended waiting time.

We define the waiting time as

$$WaitingTime = 2 \times HopNumber$$

The nodes in Blocking-ERS receiving RREQs need to wait for a period of $2H$, i.e. $2 \times$ their hop-number *unit time* before they decide to rebroadcast, where the 'unit time' is the amount of time taken for a packet to be delivered from one node to one-hop neighboring node.

In the example (Figure 5), the second ring has been made because there is no route node is found in the first ring. The node N receives RREQ and waits for, in this case, a period of 4 unit time since the hop number of the RREQ packet that the node N received is 2.

Two 'stop' signals are used in Blocking-ERS to control the flooding. One is the RREP, which can be sent by any route node to the source. The other is called 'stop_instruction' which can only be sent by the source node. The RREP informs the source node that 'a route node has been found' and therefore 'flooding should be stopped', while the 'stop_instruction' is an order to everyone involved in the flooding to terminate the route discovery process.

If there is no 'stop_instruction' message after 4 unit time has passed, it means that no node has the route information to the destination node in the second ring either. Then, every node in the second ring rebroadcasts to make the third ring. If a node finds the route information to the destination node in its route cache, this intermediate node is a route node, and it broadcasts the RREP message to the source node, and the source node broadcasts a 'stop_instruction' message to all the nodes involved in flooding. To improve the time efficiency, a 'stop_instruction' is sent via conventional TTL scheme. The automatic flooding takes place until the 'stop_instruction' message reaches all the nodes that have the same hop number as the route node that originated the RREP in the first place.

To limit the flooding in Blocking-ERS in case there is no route node is found, the source node sends a maximum journey value (max_j) with a RREQ packet. When an intermediate node receives a RREQ, the node compares the maximum journey value of the RREQ with its H. If the H is bigger than the max_j, then the intermediate node drops the RREQ packet.

To develop these ideas further, we have derived four al-

Algorithm 1 Source node

broadcast 'RREQ, $H = 1$ and max_j'
wait until a RREP is received
broadcast the 'stop_instruction' and H to everyone within the ring where it sent following conventional TTL scheme
use the 1st RREP for the data packet and save 2nd RREP as a backup
drop any later RREPs

Algorithm 2 Intermediate node

```
listen to RREQ
check the max_j after receiving the RREQ
if the H is bigger than the max_j then
    drop the RREQ
else
    check the route cache after receiving the RREQ
    if there is route information in the cache (i.e. being the
    Route Node) then
        send a RREP and H to the source node
    else
        wait for a period of 'waiting time'
        (i.e. 2 × HopNumber)
        while waiting do
            if receive a 'stop_instruction' then
                call the blocking procedure (see Algorithm 4)
                erase the source-destination pair in the route
                cache
            else if receives a 'RREP' then
                forward it to the source node
            end if
        end while
        if receives no 'stop_instruction' then
            increase the hop serial number by 1 and rebroad-
            cast RREQ
        end if
    end if
end if
```

Algorithm 3 Route or destination node

1: wait for the first arriving RREQ
2: **if** receive a RREQ **then**
3: send the RREP and the H to the source route (contained in the RREQ packet)
4: **end if**

Algorithm 4 Procedure of blocking

1: compare H_r with H
2: **if** H_r is bigger **then**
3: forward the stop_instruction and
4: erase the source-destination pair in the route cache
5: **else**
6: drop the stop_instruction and
7: stop rebroadcasting and
8: erase the source-destination pair in the route cache
9: **end if**

gorithms Algorithm 1, 2 and 3 for the source, intermediate and destination (or route) nodes respectively, and Algorithm 4 gives the details of the blocking procedure.

The source node in Algorithm 1 covers the actions of a source node for the route discovery process. This includes initialising a route discovery process by first sending a RREQ (line 1), sending a 'stop_instruction' after a RREP is received (line 3) and handling the route information in RREPs (line 5, 6).

Similarly, Algorithm 2 summerises the actions taken by intermediate nodes which can be classified into route nodes and the rest. Once the route node is identified, a RREP will be sent with the current hop number to the source node (line 7, 8). Other intermediate nodes need to wait for a period of 'waiting time' (line 11–18) and start flooding if no 'stop_instruction' is received (line 19–21). During the waiting-time period, the intermediate nodes need to forward a 'stop_instruction' (line 11–12) or the RREP (line 8–10) because it might be the 2nd RREP for the source node as a backup.

Finally, Algorithm 3 highlights the actions by a route or destination node. It simply completes route information and sends it together with the RREP to the source (line 2–4).

V. Results

The Blocking-ERS proposed in this paper takes an approach of controlled flooding and continues its route discovery process from where it was left in the previous round each time after it fails to find a Route node (see Figure 5). This avoids the repeated network-wide flooding in the conventional TTL sequence-based ERS.

The source node broadcasts the next RREQ in the following round with an increased TTL number after the current route discovery is failed in TTL sequence-based ERS. Many intermediate nodes on the partial route that has been established so far are abandoned and the route needs to be rebuilt over again from the source node. The nodes may receive redundant RREQs repeatedly. Whenever no route to the destination node is found, the route discovery process starts from the very beginning (see Figure 4).

A clear contrast can be seen from Figure 4 and 5 and comparisons can be made between the proposed Blocking-ERS and the conventional TTL sequence-based ERS.

A. Energy Consumption

Energy consumption during the transmission of RREQs can be saved by using the Blocking-ERS scheme.

Let the amount of energy consumption on each node for one broadcast be the same *unit energy* consumption, denoted by *UnitEnergy*. We assume that each action of broadcasting a RREP, RREQ or 'stop_instruction' consumes the same amount of 1 UnitEnergy.

This can be easily shown by the difference of the energy consumption between the conventional TTL sequence-based ERS and the Blocking-ERS scheme.

1) One route case: We first consider only the energy consumption along the route from the source to the route node (Figure 6).

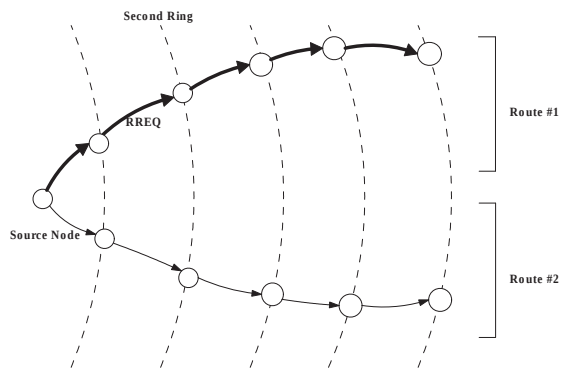

Fig. 6. Energy consumption for one route

The energy consumption for the TTL based ERS and for the Blocking-ERS can be described by the following formula respectively, where H_r is the hop number of the route node:

$$E_{TTL-ERS} = H_r + \sum_{i=1}^{H_r} i \text{ (UnitEnergy)}$$

$$E_{Blocking-ERS} = 3H_r \text{ (UnitEnergy)}$$

The difference of the amount of energy consumption is more visible from Figure 7, where the amount of energy consumption by the two ERS approaches are plotted against the number of rings (i.e. the searching distance between the source to the route node). The Blocking-ERS curve is below the TTL based ERS curve after ring 1. As we can see, the difference of the amount of energy consumption between these two mechanisms becomes larger as the distance increases between the source and the route node.

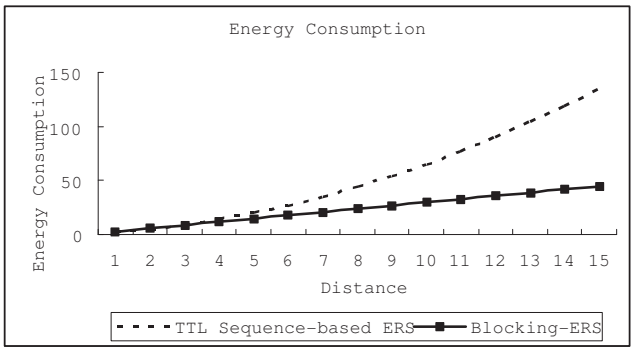

Fig. 7. Comparison of Energy Consumption for One Route

2) General case: We now consider the general case.

For the Blocking-ERS, the energy consumption during the route discovery process can be considered as the total energy consumption in three stages: (a) searching for the route node, (b) retrun RREP and (c) sending the 'stop_instruction'.

For the conventional TTL based ERS, the energy consumption during the route discovery process includes that in two stages: (a) searching for the route node, and (b) return RREP.

The energy consumed for '(b) returning a RREP' is H_r UnitEnergy for both routing schemes, and H_r UnitEnergy is consumed for the Blocking-ERS stage '(c) sending the stop_instruction'.

In the stage of '(a) searching for the route node', the energy consumption is different for the two methods.

Each ring contains a number of nodes that rebroadcast to form the next ring. Let n_i be the number of nodes in ring i and the hop number of the route node be H_r (see Figure 6 and 8).

In the Blocking-ERS, the energy consumed in each ring is as blow:

Ring i	Energy Consumed
0	1
1	n_1
2	n_2
$\vdots$	
$H_r - 1$	n_{H_r-1}

In the TTL based ERS, the energy consumed in each ring is as follows:

Ring i	Energy Consumed
0	1
1	$1 + n_1$
2	$1 + n_1 + n_2$
$\vdots$	
$H_r - 1$	$1 + n_1 + n_2 + \cdots + n_{H_r-1}$

Therefore, the total energy consumption by the Blocking-ERS is

$$E_{B-ERS} = 2(1 + \sum_{i=1}^{H_r-1} n_i) + E_{RREP} \text{ (UnitEnergy)}$$

Similiarly, the total energy consumption by the conventional TTL sequence based ERS is

$$E_{TTL_ERS} = H_r + \sum_{i=1}^{H_r-1} \sum_{j=1}^{i} n_j + E_{RREP} \text{ (UnitEnergy)}$$

The difference between the E_{B_ERS} and E_{TTL_ERS} is

$$E_{saved} = H_r - 2 + \sum_{i=1}^{H_r-1} ((\sum_{j=1}^{i} n_j) - 2n_i) \text{ (UnitEnergy)}$$

Clearly, when $n_i = 1$, for $i = 1, \cdots, H_r$. The above formula represent the energy consumption for a single route. This indicates that the energy consumption saving achieved by the Blocking-ERS for a single route is the *minimum* amount of energy saving. In reality, with a high possibility that more than one route are involved with the flooding (Figire 8). More energy saving can, therefore, be achived during the route discovery period.

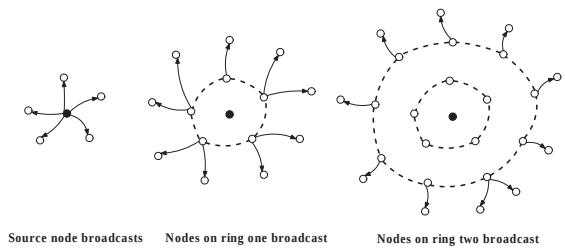

Fig. 8. Broadcasting nodes in each ring in Blocking-ERS

B. Time Delay

We consider the *time delay* for the route discovery period, during which from the RREQ is broadcasted for the first time, transmitted from the source node to the route node possibly via flooding. That is the total time taken from when the source node broadcasts the first RREQ for the first time until after a Route Node is found and the source node receive the RREP from the Route Node.

Let the *UnitTime* be the one-hop transmission time, which is the time taken for a RREQ from a broadcasting node to one of its neighbour nodes.

In case of TTL sequence based ERS, in Figure 4, for example, suppose $H = 3$, that is, the route node is 3 hops distant from the source node. The total time includes the time when $TTL = 1, 2$ and 3. The final TTL number equals to the hop number of the route node. This gives the following formula of total time delay for the TTL sequence-based ERS:

$$T_{TTL_ERS} = 2\sum_{i=1}^{H_r} i \text{ (UnitTime)}$$

Now consider the time delay in the Blocking-ERS. The total time includes the time for three stages: (a) searching for the

route node, (b) returning the RREP and (c) broadcasting the 'stop_instruction'. For stage (a), the time consists of the time to for broadcasting and the waiting time. The broadcasting time for 1 hop distance is 1 UnitTime. The waiting time depends on the hop number of the node. In Figure 5, for example, the route node is 3 hops distant from the source node. Each node needs to wait for $2H$ before rebroadcasting. At ring 1, the node waits for $2 \times 1 = 2$ UnitTime, and at ring 2, the node waits for $2 \times 2 = 4$ UnitTime, and at ring 3, the node waits for $2 \times 3 = 6$, so the total waiting time for the '(a) stage of searching for the route node' is $2 + 4 + 6 = 12$, and the total time for stage (a) is $12 + H_r = 12 + 3 = 15$. The time for stage (b) and (c) are H_r and this gives $2H_r = 2 \times 3 = 6$. Therefore, the total time for the route discovery and flooding control is $15 + 6 = 21$ UnitTime. Mathematical formula is presented below, where H represents the hop number of a route node.

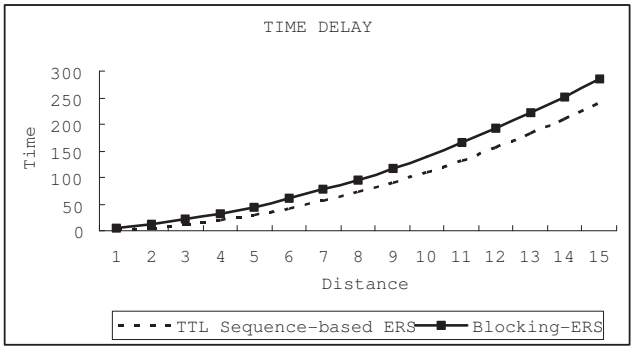

Fig. 9. Comparison of the time delay

The formula of the time delay in the Blocking-ERS is given

$$T_{B-ERS} = 3H_r + 2\sum_{i=1}^{H_r} i \text{ (UnitTime)}$$

Compare this to the TTL sequence based ERS:

$$T_{TTL_ERS} = 2\sum_{i=1}^{H_r} i \text{ (UnitTime)}$$

It is clear that the difference between the two is $3H_r$, three times of the hop serial number of the route node, depending only on the distance between the source node to the Route node.

The time difference is caused by the fact that it is the intermediate nodes, instead of the source node, that wait for the 'stop_instruction' in the Blocking-ERS. If there is no 'stop_instruction after a certain amount of time, the intermediate node received RREQ recently will rebroadcasts RREQ automatically to continue the route discovery procedure.

The time delay in both approaches is compared in Figure 9. As illustrated, the Blocking-ERS takes slightly more time than the conventional TTL sequence-based ERS for the route discovery process.

VI. Conclusion

We have introduced a new method 'Blocking-ERS' for route discovery. The analysis demonstrates a substantial improvement in energy consumption that can be achieved by the Blocking-ERS at a margin cost of slightly longer time. The amount of the energy saving by the Blocking-ERS depends on the number of nodes involved in flooding and the number of searching rings in general case. The minimum energy saving achieved by the Blocking-ERS can be seen clearly from the one route case and even the minimum energy saving looks promising.

Acknowledgment

The authours wish to thank the anonymous referees for helpful comments, and to Seungwoo Yang and Alan Huang for useful discussion.

References

[1] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, T. Imlellnski and H. Korth, Eds. Kluwer, 1996, pp. 153–181.

[2] M. Stemm and R. H. Katz, "Measuring and reducing energy consumption of network interfaces in hand-held devices," *EICE (Institute of Electronics, Information and Communication Engineers) Transactions on Communications*, vol. E80-B(8), pp. 1125–1131, 1997.

[3] C. Perkins, E. Royer, and S. Das. (2003) Ad hoc on-demand distance vector (aodv) routing. IETF Request for Comment. [Online]. Available: www.ietf.org/rfc/rfc3561.txt

[4] B. Liang and Z. Haas, "Virtual backbone generation and maintenance in ad hoc network mobility management," in *In Proc. InfoCom 2000*, 2000, pp. 1293–1302.

[5] J. Hassan and S.Jha, "On the optimazation trade-offs of expanding ring search," in *Proc. of IWDC 2004*, 2004, pp. 489–494.

[6] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proceeding of the ACM Sigmetrics Conference*, Mariana del Rey,CA, 2002.

[7] E. Royer, "Routing in ad-hoc mobile networks: On-demand and hierarchical strategies," Ph.D. dissertation, University of California at Santa Barbara, 2000.

[8] D. Koutsonikolas, S. Das, H. Pucha, and Y. C. Hu., "On optimal ttl sequence-based route discovery in manets," in *Proc. of the 2nd ICDCS International Workshop on Wireless Ad Hoc Networking (IEEE WWAN 2005)*, Columbus, Ohio, 2005.

[9] C. Perkins and E. Royer, "Ad -hoc on-demand distance vector routing," in *The 2nd Annual IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, 1999, pp. 90–100.