Flood Filtering and Route Selection for Energy-Efficient On-Demand Routing in Wireless Ad Hoc Networks

Tran Minh Trung and Seong-Lyun Kim

Radio Resource Management & Optimization Laboratory
School of Engineering
Information Communication University (ICU)
Yusong P.O. Box 77, Taejon 305-600, Korea
{trungtm,slkim}@icu.ac.kr

Abstract. This paper presents a new routing algorithm for maximizing the lifetime of a wireless ad hoc network. Our approach is to reduce the energy consumption at routing discovery phase and to establish suitable routing paths with respect to energy-saving. At the routing discovery phase, we use a dynamic flood filtering algorithm to eliminate the nodes that are predicted not to participate in the final routing paths. The prediction rules are based on energy requirement by each connection, energy availability at each node, and neighbor nodes' status. Finally, at the destination node, by using an energy-efficient route selection algorithm, the selected routing path would have small energy consumption with an ample residual energy capacity.

1 Introduction

The wireless ad hoc network is composed of mobile, wireless nodes; its availability depends on status of each node. For that, the failure of any node can divide the network into multiple parts. In particular, if a node runs out of energy, the probability of network partitioning will increase. Since every mobile node has a limited power supply, the energy depletion becomes one of the main threats to the lifetime of the ad hoc network. To cope with the issue, recently proposed solutions have tried to find energy-efficient routing algorithms, with hope that the power consumption can be distributed evenly among the nodes (see [1-8] and references therein).

In this paper, we approach the issue from a different angle. *On-demand routing protocols* such as AODV [9] and DSR [10] are most widely accepted algorithms by wireless ad hoc networks. However, when applying those on-demand routing protocols, the flooding of control messages such as RREQ (route request) packets at the *routing discovery phase* will be out of control. Since the flooding is one of the most energy-intensive operations, the uncontrolled flooding will lead to unnecessary energy consumption at nodes, resulting in serious redundancy, contention and collision (known as *broadcast storm* [11]). Our focus is how to

reduce the energy consumption in the routing discovery phase, while the selected path still has the reasonable energy capacity. Regarding the broadcast storm, there have been some previous researches [11-15] for reducing redundancy of the simple flooding, which can be categorized as follows:

- *Probabilistic-based scheme* [11, 12, 14]: Whenever a node receives an RREQ message for the first time, it will re-broadcast the message with a probability P. This scheme becomes worse (just like the simple flooding) when we want to increase the reach-ability performance by increasing P.
- *Counter-based scheme* [11, 14]: When the medium is busy and the queued messages are many, there is a chance that a node has received the same RREQ message many times before the node starts re-broadcasting it. As a remedy, the node keeps track the number of times the same RREQ message has been received. If it is larger than a threshold, re-broadcasting is inhibited.
- *Distance-based scheme* [13]: Having understood the relationship between the distance and the power, we can even directly replace the role of distances by signal strengths by establishing a signal-strength threshold, and then make re-broadcast decision. The signal strength information was also used to facilitate routing.
- *Location-based scheme* [11, 14]: It uses GPS to get precise location of each node. The location information is used to facilitate the route discovery process.
- *Cluster-based scheme* [11, 13, 15]: This approach is based on a graph-theoretic modeling. All nodes are partitioned into clusters. Each node can communicate with other nodes through a host node, called gateway.

From lifetime maximization point of view, some nodes may not be selected for the final route. For example, a node that has a low residual battery capacity compared to energy requirement by the incoming connections, may be better to be filtered out in the route discovery phase. All of above mentioned flooding schemes, however, did not consider the energy requirement of incoming connections, the residual energy of each node and the impasse zone (non-destination) that routing control messages should not be forwarded to. Consequently, they still have to waste an unnecessary amount of energy for forwarding redundancy packets. Also, we think some of above mentioned algorithms are too complicated for being implemented in the real world. Our goal in this paper is to remedy such drawbacks of the existing flood-filtering algorithms while having the practical applicability.

In this paper, we propose so-called the *dynamic flood filtering* that considers four aspects: (i) node residual energy capacity (ii) node's affordability for incoming connections (iii) neighbor nodes' link status, and (iv) practical applicability of the flood filtering. Our idea is to predict whether a node has enough capability for participating in the final routing path or not, based on its available energy capacity and its neighbor's link status. If a node does not have favorable conditions, it will not have to take part in flooding process. Our aim is to apply our idea to AODV or DSR, achieving energy-efficiency while keeping the on-demand

behavior. One minor contribution of this paper is in the new route selection algorithm (after the flooding is finished) that selects a suitable routing path with respect to energy-saving. To see the effectiveness of our idea, we compare it with some other routing algorithms, such as *minimum hop* (MH) and *min-max battery cost routing* (MMBCR) [4], in which AODV route discovery was combined. The remainder of this paper is organized as follows. Section II describes our proposed solution. Our simulation results and evaluation are presented in Sections III. Section IV concludes the paper with remarks on ongoing research.

2 Proposed Solutions

2.1 Basic Ideas

The routing protocol that we choose to apply our proposed solutions is ADOV. In the original on-demand routing protocol, when receiving an RREQ, a node that is unable to be a destination and does not have a routing path to the destination in the routing table, will re-broadcast it to other nodes. This action will be repeated until the RREQ reaches the destination or the node that eventually has a routing path to the destination. Since only few nodes can take part in the selected routing path while many others have to only re-broadcast the RREQ packet, the routing discovery performs many redundant re-broadcast. To solve this problem, we propose a dynamic flood filtering algorithm. We consider three cases in which we can predict whether a node will belong to the final selected routing path or not. From that, we can eliminate unnecessary nodes from the routing discovery phase:

- *Weak node*: A node that does not have enough energy for serving the incoming connection. For example, a source node wants to send a big amount of data, where the minimum energy requirement for sending and receiving this amount of data is 100 joules. However, when the relay node has only 20 joules left, so it drains out energy after a short time. Consequently, the connection will be disconnected and the routing discovery phase will be activated again. That makes unnecessary energy consumption. Figure 1 shows that when the energy requirement of the connection from source (A) to destination (F) is 100 joules, the selected path will be (A, D, E, F). The path (A, B, C, F) was not selected because it contains node B that does not have enough energy for serving the required connection.
- *Congested node*: A node has enough residual energy capacity but it coincidentally belongs to some other active routing paths. So its energy will be drained out very fast. If the energy for serving the current connections and the coming connections is not enough, then the node would be better reject incoming connections. Figure 2 shows that node X has enough energy for serving a connection that requires energy smaller than 120 joules, but it coincidentally belongs to two active routing paths: (B, X, E) and (D, X, C). So as time goes by, the residual energy of node X will go down very fast and the node cannot guarantee enough energy for serving the incoming connections.

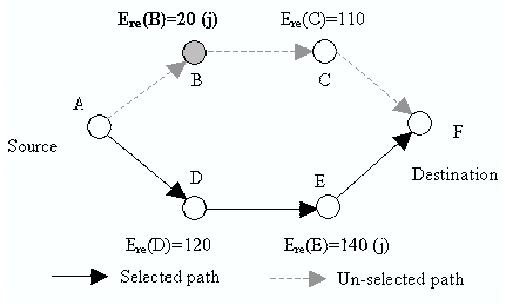

Fig. 1. The node B should be eliminated from routing discovery phase.

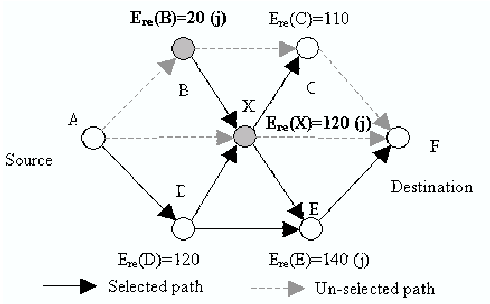

Fig. 2. Node B, X should be eliminated from routing discovery phase.

– *Black zone*: Except the node that it receives the RREQ from, a node may have only one hop neighborhood, all of which as well as itself are not the destination. Figure 3 shows that an RREQ message is broadcast to an impasse (or a black zone), where none of the nodes B, C, D is the destination. Except node A, node B has only nodes C and D as its neighbors.

2.2 Dynamic Flood Filtering Algorithm

In order to incorporate these three aspects into AODV, we consider to modify HELLO and RREQ messages of AODV as follows:

– *HELLO message*: In original AODV, each forwarding node should keep track of its continued connectivity to active neighbor hops (i.e., which next hops or precursors the current node has sent/received packets to or from). This is done by using the HELLO messages that are exchanged periodically among the nodes. In our flood filtering, we suggest to use two more fields in the HELLO message, for constructing/updating the *neighborhood status table* (e.g. Table 1): (i) RESIDUAL_ENERGY field contains information about the residual energy capacity of a node. (ii) LINK_STATUS field contains information about the link status of neighborhood nodes. We use three values

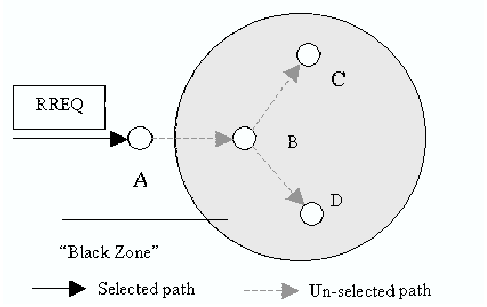

Fig. 3. Node B, C and D should be eliminated from routing discovery phase.

Table 1. The status table of node B of Figure 3. In this table, the LINKS_STATUS value of node A is 2, which means that B has connection up to at least 2 hops via node A, because A has other neighbor nodes other than B.

ID	Residual Energy	Links Status
A	20(j)	2
B	100(j)	0
C	140(j)	1
D	90(j)	1

to indicate the link state of a node. The record that contains "0" value belongs to the node that owns this status table. The value being greater than "0" indicates that the corresponding neighbor node has connection upto one or more than one hops away. It is notable that our modification of the HELLO message costs exchanging few more extra bits among the nodes.

– *RREQ message*: Whenever a source node initiates a routing process for a connection, the information about *incoming energy request* ($E_{future\_request}$) of this connection will be piggybacked into ENERGY_REQUEST field of the RREQ header. The value $E_{future\_request}$ is calculated as follows:

$$E_{future\_request} = N_{packets} \times (E_{tx} + E_{tr}) \tag{1}$$

where E_{tx} and E_{tr} denote energy consumption for transmitting and receiving one packet, respectively. The number $N_{packets}$ denotes the total number of packets that will be transmitted over the connection. Also, we suggest using one more field in RREQ header; ENERGY_CAPACITY field stores the information about *energy capacity* ($E_{capacity}$) of a routing path.

Since the energy capacity of each possible routing path is equal to the *available energy capacity* ($E^t_{available(i)}$) of the weakest node on the path, it is calculated as follows:

$$E_{capacity} = Min_i[E^t_{available(i)}] \tag{2}$$

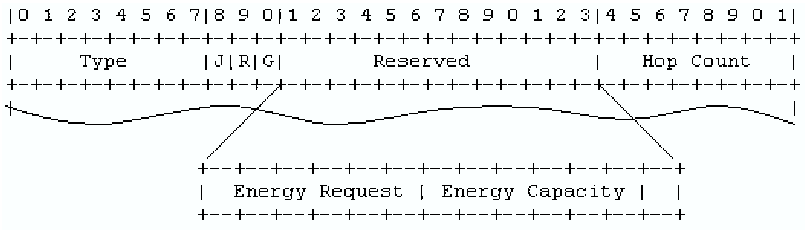

Fig. 4. Modified AODV RREQ Header. 13 bits are reserved for storing ENERGY_REQUEST and ENERGY_CAPACITY information [9].

where $E^t_{available(i)}$ is the predicted available energy capacity of node i at time t and the time t denotes the instance that the RREQ packet passes through node i. This available energy capacity is calculated based on its residual energy capacity and the energy requirement by the current active routing paths. Let $E^t_{current\_request(j)}$ be the energy request by active routing path j at time t then:

$$E^t_{available(i)} = E^t_{residual(i)} - \sum_j E^t_{current\_request(j)} \tag{3}$$

Each time the RREQ message passes by a node, the value $E_{capacity}$ in ENERGY_CAPACITY field will be compared with $E^t_{available(i)}$ of that node. If it is greater than $E^t_{available(i)}$ then ENERGY_CAPACITY field will be updated with $E^t_{available(i)}$. By performing this method, the ENERGY_CAPACITY field always contains the minimum available energy capacity of the nodes on the path that the RREQ packet has traversed. The modified format of RREQ message header is shown in Figure 4, in which we use 6 reserved bits for storing energy request information and the other 6 reserved bits for storing energy capacity information.

In the original AODV routing protocol, at routing discovery phase, when a node receives an RREQ, it determines whether it has received RREQ:s from the same originator IP address, during at least the last PATH_DISCOVERY_TIME. If such an RREQ has been received, the node silently discards the newly received RREQ. Else, if this node is not the destination or it does not have an active routing path to the destination, it will re-broadcast the RREQ message to all neighboring nodes. In order to eliminate unnecessary nodes from the routing discovery phase, we add two checking processes into the *processing and forwarding route request step* as follows:

- *Weak and congested node*: Whenever a node receives an RREQ message, it will compare its *available energy capacity*, $E^t_{available(i)}$ with the *energy request*, $E_{future\_request}$ of the incoming connection. If the former is smaller, then this node will silently discard the RREQ packet. Figure 5 shows that, node (5) discards an RREQ message because its available energy capacity is smaller than the energy request. While node (10) has enough residual energy

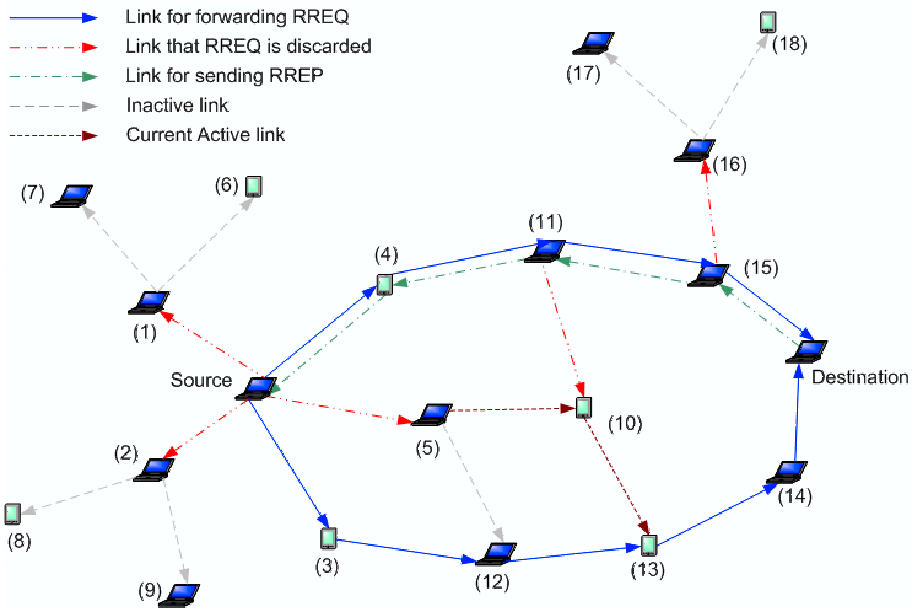

Fig. 5. A sample RREQ flooding with 7 forward nodes and 11 non-forwarding nodes.

capacity but it is on-service for two ongoing connections: (11, 10, 12) and (5, 10, 13) at the same time. As a result, its available energy may not be enough for the incoming connection; it will discard the RREQ packet. In the case that all the neighbor nodes do not have enough available energy capacity, the RREQ will broadcast with $E_{future\_request}$ value equaling to the maximum available energy capacity of the neighbor nodes.

– *Black zone*: In order to prevent the RREQ packet from being broadcast to a black zone (Figure 3), each node receiving an RREQ will check its neighbors' link status. If all of the neighbor nodes, except the node that the RREQ is sent from, are not the destination and have LINK_STATUS value smaller than '2', it will discard the RREQ packet. In Figure 5, we can see a black zone starting at node (1) so that the node will discard all RREQ packets. This case is the same as those of nodes (2) and (16).

2.3 Energy-Saving Route Selection Algorithm

During a receiving period, each received RREQ packets will be saved in a lexicographic order with two entries: (HOP_COUNT, ENERGY_CAPACITY). After the period, the RREP (route reply) will be generated, with respect to the first RREQ packet in this lexicographic order. By this way, we hope that the selected routing path would have smallest energy consumption for transmitting/receiving data among the possible routing paths. Due to the filtering, the selected path has

reasonable energy capacity for serving the incoming connection. Figure 5 illustrates our flooding process. Instead of 16 nodes re-broadcasting RREQ packets, there are only 7 nodes performing RREQ re-broadcast. The possible routing paths at the destination are: (S, 4, 11, 15, D) with 4 hops and (S, 3, 12, 13, 14, D) with 5 hops. The final selected path is (S, 4, 11, 15, D) with smaller hops (energy consumption) for transmitting and receiving data.

3 Simulation Results and Evaluations

To get the insights on the proposed solutions, we compare it with two other routing algorithms; one is the minimum hop (MH) and the other is MMBCR [4]. All of them are combined with the RREQ flooding of AODV. Each algorithm is implemented by the NS2 tool [16]. In our simulation model, 40 mobile nodes are generated randomly in an area of 500m×500m. The moving speed of each node is 10m/s. A number of connections with different levels of energy requirement are established during 900 seconds. In this simulation we choose the energy requirement of each generated connection is equal with $\frac{3}{4}$ of the residual energy capacity of the source node. For instance, if at simulation time t, a source node has residual energy capacity is 10 joules, then the energy requirement of the connection generated from this node at time t will be $\frac{3}{4}\times 10 = 7.5$ joules.

There are two criteria which will be measured: The first one is the network lifetime. An ad hoc network will be dead whenever any node cannot send data to its destination. This case happens when any node runs out of energy, making the network partitioned. The second one is the energy consumption when the network is flooded with routing control messages. The simulation results are shown in Figures 6-9, in which we use the labels as follows: AODV-MH, AODV-MMBCR, and AODV-FF&RS stand for the MH, MMBCR and our flood filtering and route selection, respectively. As the name says, all the algorithms are combined with AODV with respect to the routing discovery phase.

3.1 Network Lifetime

First, we generate 10 connections (source-destination) with different energy requirement levels. The initiating time for each connection is chosen randomly. Figure 6 shows that our proposed routing algorithm achieved the longest network lifetime in the case of 10 connections. The network lifetime of AODV-FF&RS is up to 880s, while AODV-MH and AODV-MMBCR last 660s and 810s, respectively. In our algorithm (AODV-FF&RS), the time until the first node runs out of its energy is 630s, while those numbers in AODV-MMBCR and AODV-MH are 300s and 90s, respectively. Figure 6 also shows that the AODV-FF&RS curve is more slopping than AODV-MH and AODV-MMBCR. This means that the energy consumption is distributed more equally in case of AODV-FF&RS, making the deviation among nodes in AODV-FF&RS smaller.

For the numbers of connections, 15 and 20, Figures 7 and 8 show that the network lifetime becomes shorter. In the case of 15 connections, the network

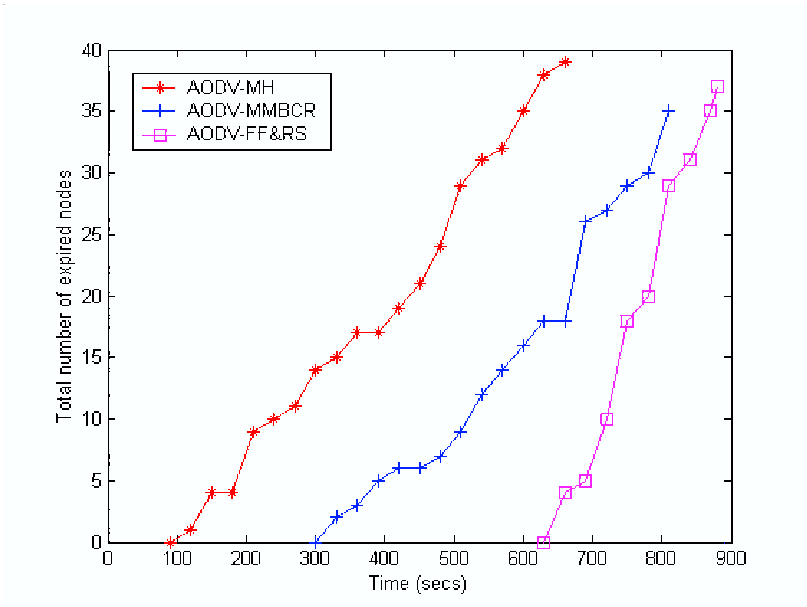

Fig. 6. The number of expired nodes as a function of simulation time (10 connections).

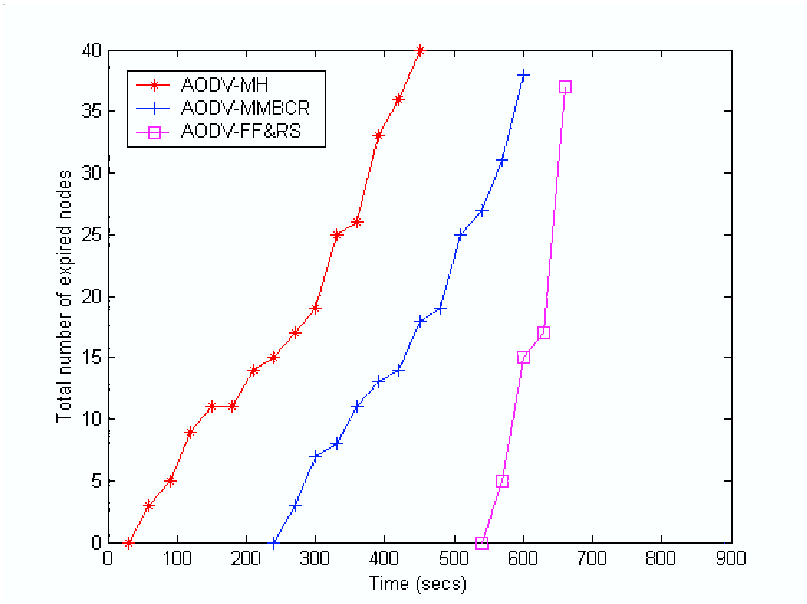

Fig. 7. The number of expired nodes as a function of simulation time (15 connections).

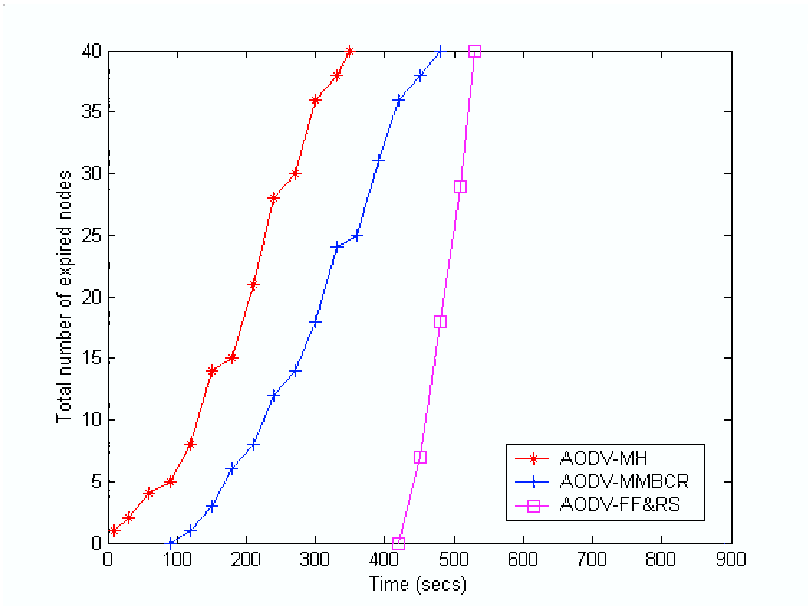

Fig. 8. The number of expired nodes as a function of simulation time (20 connections).

lifetime of APDV-FF&RS is 660s while those of AODV-MMBCR and AODV-MH are 600s and 450s, respectively. This means that the network lifetime of our algorithm is longer than AODV-MMBCR and AODV-MH by 10% and 46%, respectively. With 20 connections, these increments are 11% and 56%. This convinces us that the more traffic load (number of connections) the better effect that our algorithm gets.

As mentioned above, an ad hoc network is dead whenever any node cannot send data to its destination. In the cases of 10 and 15 connections, the simulation results show that, although the network is dead, there are still some alive nodes with AODV-MMBCR and AODV-FF&RS. This means that when the network is partitioned, the still-living nodes cannot send and receive data. However, in the case of 20 connections, the ad hoc network is dead, because of all nodes running out of energy.

3.2 Routing Energy Consumption

To evaluate the effectiveness of new flooding algorithm, we measure the total energy consumption for flooding routing control messages in the cases of 10, 15 and 20 connections. In each case, we calculate the total energy that all nodes have to spend for sending and receiving routing control messages. Figure 9 shows that with 10 connections the energy consumption of AODV-FF&RS, AODV-MMBCR and AODV-MH are 14, 59, 64 joules, respectively. With 15 connections, they are 28, 70, 65; with 20 connections, 38, 98, 92 joules. It means that our algorithm

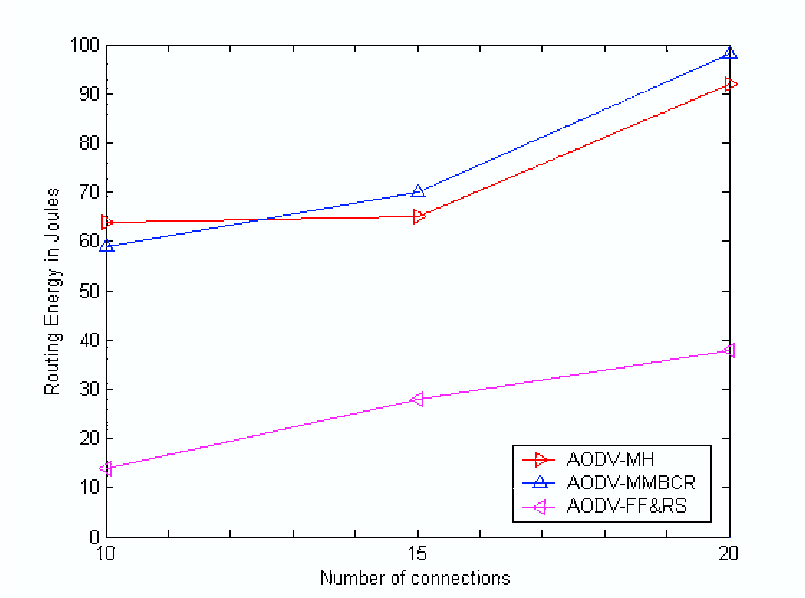

Fig. 9. Routing energy consumption.

can save energy consumption in the routing discovery phase, from 60% upto 78%, compared to other mentioned algorithms.

4 Concluding Remarks

The lifetime of an ad hoc network can be increased by efficiently and wisely controlling the power consumption of each individual node. In this paper, we propose an adaptive routing algorithm that can help control the flooding of routing control messages and give all nodes active right to save their energy. In our approach, every node can decide by itself whether to take part in a connection section or not by performing a set of prediction rules. These prediction rules are based on energy requirement by each connection, energy availability at each node, and neighbor nodes' status. Finally, the route selections algorithm enable the destination node to choose the most suitable routing path, which has enough energy capacity and small energy consumption. Our simulation results show that the solution can significantly last the network lifetime. Moreover, we see that it is also easy to implement by modifying an existing routing protocol such as AODV. This encourages us to implement our algorithm on a real testbed, which is our current research focus.

Acknowledgements

This research is supported by the ITRC program, Ministry of Information and Communications of Korea.

References

1. R. Jäntti and S.-L. Kim: Energy-efficient routing in wireless ad hoc networks under mean rate constraionts, in Proc. IEEE VTC-Spring, Jeju, Korea, 2003.
2. W. Cho and S.-L. Kim: A Fully distributed routing algorithm for maximizing lifetime of a wireless ad hoc network, in Proc. IEEE WCNC, Stockholm, Sweden, 2002.
3. S. Singh, M. Woo, and C. S. Raghavendra: Power-aware routing in mobile in mobile ad hoc networks, in Proc. ACM MobiCom, Dallas, Texas, USA, 1998.
4. C. K. Toh, H. Cobb, and D. A. Scott: Performance evaluation of battery-life aware routing schemes for wireless ad hoc networks, in Proc. IEEE ICC, Amsterdam, Netherlands, 2001.
5. V. Rodoplu, T. H. Meng: Minimum energy mobile wireless ad hoc networks, Selected Areas in Communications, IEEE Journal on, Volume: 17 Issue: 8, Pages: 1333 -1344, 1999.
6. M. Maleki, K. Dantu and M. Pedram: Power-aware source routing protocol for mobile ad hoc networks, in Proc. ISLPID, Monterey, California, USA, 2002.
7. Y. X. John, H. D. Estrin: Geography-informed energy conservation for ad hoc routing, in Proc. ACM MobiCom, Rome, Italy, 2001.
8. B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris: An Energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks, in Proc. ACM MobiCom, Rome, Italy, 2001.
9. C. E. Perkins, M. Belding-Royer, and D. Samir: Ad hoc on demand distance vector (AODV) routing, IETF Internet draft, draft-ietf-manet-aodv-10.txt, 2002.
10. D. Johnson and D. Maltz: Dynamic source routing in ad hoc wireless networks, Mobile Computing, edited by T. Imielinski and H. Korth, Kluwer Academic Publishers: Chapter 5, pages 153-181, 1996.
11. S. Y. Ni: The broadcast storm problem in a mobile ad hoc network, in Proc. ACM MobiCom, Seattle, Washington, USA, 1999.
12. Y. Sasson, D. Cavin, A. Schiper: Probabilistic broadcast for flooding in wireless mobile ad hoc networks wireless communications and networking, in Proc. IEEE WCNC. New Orleans, Louisiana, USA, 2003.
13. Y. Yunjung, M. Gerla: Efficient flooding in ad hoc networks, a comparative performance study, in Proc. IEEE ICC, Anchorage, Alaska, USA, 2003.
14. S. Min-Te, L. Ten-Hwang: Location aided broadcast in wireless ad hoc network systems, in Proc. IEEE WCNC, Orlando, Florida, USA, 2002.
15. C.R. Lin and M. Gerla: Adaptive Clustering for Mobile Wireless Networks, Selected Areas in Communications, IEEE Journal, Volume: 15 Issue: 7, Pages: 1265-1275, 1997.
16. The Network Simulator - http://www.isi.edu/nsnam/ns/, DARPA, 1995.