Power-Balance Broadcast in Wireless Mobile Ad Hoc Networks

Jang-Ping Sheu, Yu-Chia Chang, and Hsiu-Ping Tsai

Department of Computer Science and Information Engineering
National Central University, Chung-Li, 32054,Taiwan

Email: sheujp@csie.ncu.edu.tw, jimchang@axp1.csie.ncu.edu.tw and sptsai@axp1.csie.ncu.edu.tw

***Abstract*—Broadcast is an important operation in a mobile ad hoc network (MANET). Flooding, a straightforward approach, performs the broadcast. Unfortunately, flooding will not only cause problems like redundancy, contention, and collision but may lead to rapid battery exhaustion and reduce the network lifetime. Hence, how to balance remaining battery energy of mobile hosts is a critical issue for the MANETs. Our proposed power-balance broadcast algorithms take the residual battery energy of hosts into consideration. Each host has a rebroadcast probability which is determined by its residual power, number of neighbors, and average residual power of neighbors. Therefore, the hosts with less energy will have lower probability to broadcast than those with more energy. We inhibit low energy hosts from rebroadcasting to balance the host remaining energy and extend the lifetime of the networks. Simulation results demonstrate that our approach can balance the remaining battery energy of hosts and extend network lifetime without scanting the reachability, even in the environment of high mobility and density.**

Keywords: Broadcast, flooding, lifetime, mobile ad hoc network (MANET), power balance.

I. INTRODUCTION

Owing to the advance of wireless communications, portable computing devices have made mobile computing possible. A MANET consists of mobile hosts that may communicate with one another and roam around at will. Broadcast, in a MANET, is a common operation in applications such as supporting network routing protocols and in distributed computing problems. Due to host mobility, broadcasting is expected to be performed more frequently in environments such as paging a particular host, sending an alarm signal, and finding a route to a particular host [1][2][4][8].

In this paper, we assume that mobile hosts in the MANET share a single common channel. Each host is equipped with a CSMA/CA (carrier sense multiple access with collision avoidance) [11] transceiver. Before transmitting packets, a host should confirm if the medium is free during a backoff slot. The broadcast problem has two characteristics. First, the broadcasting is spontaneous. A mobile host can issue a broadcast operation anytime. Second, the broadcast is unreliable, that is, no acknowledgement mechanism will be applied. Although the CSMA/CA technique is used in the wireless network, drawbacks such as redundant rebroadcasts, contentions, and collisions still occur. We refer to the above phenomena as the broadcast storm problem [7]. A series of recent papers [3] [5] [6] [7] [9] have presented plenty of approaches which try to reduce redundant rebroadcasts and solve the broadcast storm problem. The solutions can be classified into two categories. One is sender-based mechanism [3] [6] [10]. The sender will choose the next transmitter in order to relay the broadcast message. The other is receiver-based mechanism, that is, the node which receives the broadcast message determines whether to rebroadcast or not [7] [9]. However, the aforementioned solutions omit the power-balance issue. The *probabilistic* and the *counter-based* schemes proposed in [7] relating to our approach are described below. In the probabilistic scheme, when a host receives a broadcast message for the first time, the host will rebroadcast the message with a probability. In the counter-based scheme, a counter C is used to keep track of the number of times that the broadcast message is received. A counter threshold, *C-threshold*, is chosen. Whenever $C \geqslant$ *C-threshold*, the rebroadcast is inhibited.

In our power-balance broadcast algorithms, each host determines a rebroadcast probability according to its remaining energy, number of neighbors, and average remaining energy of its neighbors. Thus, the hosts with more remaining energy will have higher probability to broadcast than the hosts with less remaining energy. We reduce the rebroadcast probability of the low energy hosts to balance the remaining energy of the hosts and extend the network lifetime. Simulation results show that, as compared to the flooding method, our broadcast algorithms improve network lifetime by 40%, and decrease standard deviation of remaining energy of hosts by 35%. The rebroadcast ratio is saved about 50%. Our proposed algorithms also perform better than the probabilistic and counter-based schemes. Even in a high mobility environment, our approaches can balance the remaining battery energy of the hosts and improve network lifetime with higher reachability.

The rest of this paper is organized as follows. Section 2 describes our power-balance algorithms. The simulation results are presented in section 3. Section 4 concludes the paper.

II. POWER-BALANCE BROADCAST ALGORITHMS

In this section, we propose two algorithms to balance the residual battery energy of hosts and to prolong the network lifetime. Our algorithms take the remaining mobile hosts energy into consideration and calculate a rebroadcast probability for the mobile host. First, each host periodically collects the remaining energy of its neighbors. Then, we determine the rebroadcast probability of each host based on its remaining energy and the information collected from its neighbors. Low energy hosts will assign low rebroadcast probability to forward the broadcast messages. Hence, the hosts with low energy can reduce power consumption in broadcasting.

Lower average rebroadcast probability can save more energy, but will decrease the reachability. Thus, we would like to observe the relationship between the broadcast reachabilities and the rebroadcast probabilities before presenting our algorithms. Our simulation network is created within a 1000 m x 1000 m space in a random way-point movement model. The broadcasting host is randomly picked and produces the message once per second. The transmission radius is 250 m, the transmission rate is 11 Mb/sec, the broadcast packet size is 280 bytes, the maximum speed of the host is 5 m/sec, and the pause time is 30 seconds. The average rebroadcast probability of the hosts in a network is assigned to 0.5, 0.6, 0.7, 0.8, 0.9, and 1.0 (Flooding). Fig. 1 illustrates that the higher rebroadcast probability will provide higher reachability. The simulation result also explains that the average rebroadcast probability should be kept higher than 0.7 to guarantee the broadcast reachability reaching over 80%. The main goal of our approaches is to balance the remaining battery energy of hosts and keep high reachability.

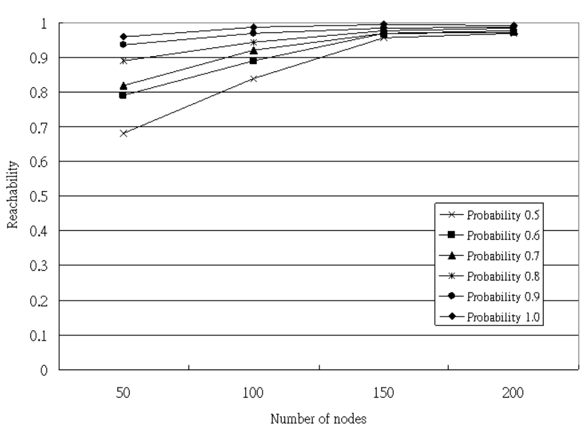

Fig. 1. The reachabilities for various rebroadcast probabilities

A. *Algorithm I*

To begin with, we would like to define some terms used in this paper. Let P_i be the rebroadcast probability of host *i* and *RE(i)* be the remaining energy of host *i*. In order to balance the remaining energy of hosts and to prolong the network lifetime, we first assign the rebroadcast probability of each host according to the host's remaining energy. To maintain a higher reachability, say over 80%, we need to keep the average rebroadcast probability of the hosts higher than 0.7 in a network. The simulation result is shown in Fig. 1. According to host's remaining battery capacity, we use three energy levels and assign different initial rebroadcast probabilities to each level. The initial rebroadcast probability of each host is determined by the following rules. If the remaining energy of host *i*, *RE(i)*, is larger than 70%, the rebroadcast probability P_i is assigned to 0.9. If 40% $<$ *RE(i)* $\leq$ 70%, P_i is assigned to 0.7. Finally, if *RE(i)* $\leq$ 40%, P_i is assigned to 0.5. In general, we can keep the average rebroadcast probability to 0.7 and have a high reachability in a network. The hosts with low energy can decrease their rebroadcast probability and save their battery energy. Thus, the lifetime of the whole network is extended.

However, the reachability will decrease in some special cases. For example, if a host and its neighbors have low battery energy, these hosts will be assigned to low rebroadcast probabilities and result in low network reachability. So, we need to raise the rebroadcast probability of these hosts to keep the higher reachability. On the contrary, if a host and its neighbors have higher remaining energy larger than 70%, the rebroadcast probability of these hosts is assigned to 0.9. Although the network reachability can keep over 80%, it will cause broadcast storm problem and waste much energy in broadcasting since too many hosts would like to rebroadcast the broadcast messages. Thus, we need to adjust the rebroadcast probability of the hosts with high remaining energy according to the number of neighbors. When the number of neighbors is less than *N_threshold*, the rebroadcast probability assignment is hold as before. Otherwise, a square function is used to smoothly decrease the rebroadcast probability of the hosts with high remaining energy. A simulation is used to determine the value of *N_threshold*. Fig. 2 illustrates that if *N_threshold* is greater than or equal to 4, the reachability is over 80%. The higher value of *N_threshold* is, the bigger chance of a host keeping the high rebroadcast probability is. Since a higher *N_threshold* will spend more battery energy in broadcasting, the value of *N_threshold* is set to 4 in our algorithms to save the battery energy and keep the network reachability over 80%.

In the following, we will adjust the initial rebroadcast probability P_i of each host based on the following three cases. In case 1, when the average remaining energy of the neighbors of host *i* is denoted as *Avg_NE(i)* $\leq$ 40%, we assign the new rebroadcast probability P_i according to its remaining energy, *RE(i)*. If *RE(i)* $>$ 70%, P_i is kept in 0.9. If 40% $<$ *RE(i)* $\leq$ 70%, let P_i equal to 0.8. Otherwise, P_i is raised to 0.7. In case 2, when 40% $<$ *Avg_NE(i)* $\leq$ 70%, the new rebroadcast probability P_i is held as the initial assigned value. In case 3, when *Avg_NE(i)* $>$ 70%, the new rebroadcast probability P_i will take the number of neighbors into consideration. If the number of neighbors $<$ *N_threshold*, P_i is kept as before. If the number of neighbors $\geq$ *N_threshold*, we use a square function, $(3/Neighbor\_No(i))^{0.5}$, where *Neighbor_No(i)* denotes the number of neighbors of host *i*, to smoothly decrease the rebroadcast probability of P_i.

Algorithm I:

Initial: Each host periodically collects the remaining energy

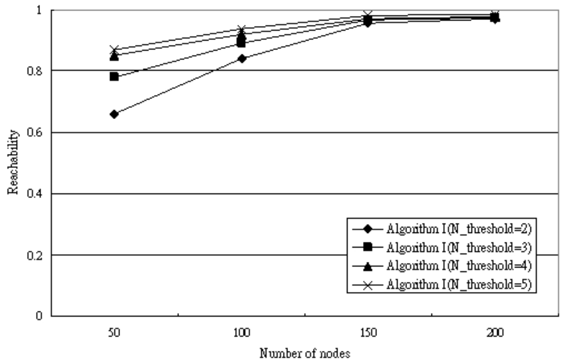

Fig. 2. The reachability of Algorithm I with different *N_threshold*.

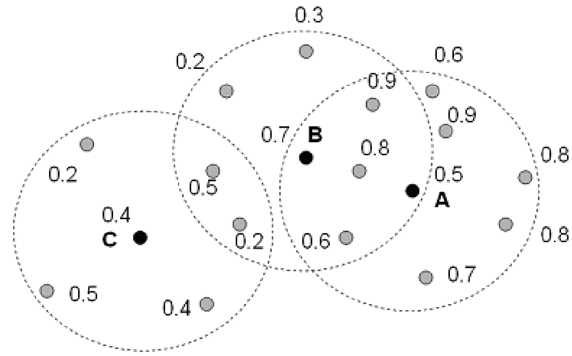

Fig. 3. The rebroadcast probability assignment according to Algorithm I.

of its neighbors and computes the average remaining energy, $Avg\_NE(i)$. Each host *i* is assigned an initial rebroadcast probability P_i based on the following rules.

- If *RE(i)* $> 70\%$, $P_i = 0.9$.
- If $40\% <$ *RE(i)* $\leq 70\%$, $P_i = 0.7$.
- If *RE(i)* $\leq 40\%$, $P_i = 0.5$.

Step 1: On receiving a broadcast message, the host *i* calculates a new rebroadcast probability P_i according to the following three cases if the message is received in the first time. Otherwise, the host is inhibited from rebroadcasting the message.

- Case 1: $Avg\_NE(i) \leq 40\%$
  If *RE(i)* $> 70\%$, assign P_i = 0.9.
  If $40\% <$ *RE(i)* $\leq 70\%$, assign P_i = 0.8.
  If *RE(i)* $\leq 40\%$, assign P_i = 0.7.
- Case 2: $40\% < Avg\_NE(i) \leq 70\%$
  Keep P_i as the initial assigned value.
- Case 3: $Avg\_NE(i) > 70\%$
  If $Neighbor\_No(i) <$ *N_threshold*, P_i is held as before.
  If $Neighbor\_No(i) \geq$ *N_threshold*, assign P_i = P_i * $(3/Neighbor\_No)^{0.5}$.

Step 2: Generate a random number *RN* over [0, 1]. If *RN* $\leq P_i$, rebroadcast the received message; otherwise, drop it.

An example to illustrate the operation of Algorithm I is shown in Fig. 3. The number beside each host represents the remaining battery energy of the host and the dotted cycles represent the transmission ranges of the hosts *A*, *B*, and *C*. For host *A*, the $Neighbor\_No(A) = 9 > 4$ and the $Avg\_NE(A) = 76\% > 70\%$, so P_A = P_A * $(3/Neighbor\_No(A))^{0.5}$ = 0.404. For host *B*, the $Avg\_NE(B) = 50\%$ and *RE(B)* = 70%, thus P_B is assigned to 0.7. For host *C*, $Avg\_NE(C) = 36\% < 40\%$ and *RE(C)* = 40%, we have P_C = 0.7.

B. *Algorithm II*

When a host tries to rebroadcast a message, the message may be blocked by a busy medium, the back-off procedure, and other queued messages. There is a chance for the host to hear the same message again and again from other hosts before the host actually starts to transmit the message. A host rebroadcasts a message after receiving the message *k* times, the additional coverage is expected to be lower when *k* increases. The authors in [7] have discussed the proper value of *k*. If $k \geq 3$, the expected additional coverage is below 5%. Algorithm II combines the idea of Algorithm I and the counter-based scheme to save the number of redundant rebroadcast. The rebroadcast is inhibited if a message is received more than three times. Algorithm II is explored in detail below.

Algorithm II:
Let *Rec_No(i)* be the number of times the broadcast message *M* is received.
Initial: The rebroadcast probability P_i is calculated based on Algorithm I.
Step 1: On receiving a broadcast message *M*, go to Step 2 if the message is received in the first time. Otherwise, the host is inhibited from rebroadcasting *M*.
Step 2: Generate a random number *RN* over [0, 1]. If *RN* $\leq P_i$, go to Step 3; otherwise, drop the message.
Step 3: The host will not rebroadcast *M* if *Rec_No(i)* > 3. Otherwise, rebroadcast the message.

The advantage of Algorithm II is to balance the remaining energy among the hosts, reduce redundant rebroadcast, and keep the higher reachability.

III. Simulation Results

We have developed a simulator using Glomosim 2.03[12]. Our simulation network is created within a 1000 m x 1000 m space with a random way-point movement model. One broadcast is requested per second. The broadcasting host is randomly picked. In our simulations, we assume that the transmission radius is 250 m, the transmission rate is 11 M bits/sec, the broadcast packet size is 280 bytes, the maximum speed of a host is 5 m/sec, the pause time is 30 seconds, the transmission power is 3.63 W, and the receiving power is 2.54 W. Each node's battery energy is selected at the beginning of the simulation, and uniformly distributed between one and two Joules.

Algorithms I and II need to periodically broadcast beacons in order to exchange battery energy among the neighbors. We assume that each host broadcasts a 64 bytes beacon in every 10 seconds. Four performance metrics are observed :

- Reachability: The number of mobile hosts that receives the broadcast message divided by the number of surviving hosts.
- Network lifetime: We consider two kinds of network lifetimes. One is the duration of time till the first node fails due to all battery exhaustion. The other is the average lifetime of all the nodes in a network.
- Saved rebroadcast ratio (SRR): *(r - t) / r*, where *r* is the number of hosts that receive the broadcast message, and *t*, the number of hosts which actually transmit the message.
- Standard deviation: The standard deviation of the residual power among the hosts.

A. *Reachability Analysis*

Fig. 4 shows the results of reachability. This simulation totally broadcast 1000 times. The result shows that reachability increases when network density increases, regardless of what kind of the algorithms is used. The flooding algorithm has the best performance in reachability, reaching nearly 1. The second one is counter-based scheme. Counter-based scheme would not be easily influenced by mobility. The performance of Algorithm I shows that the reachability is above 85% in any density of the network. The reachability of Algorithm II is lower than Algorithm I. In all network densities, the reachability of both Algorithm I and Algorithm II perform better than the probabilistic scheme with the probability assigned to 0.7. In higher density networks, i.e., 150 hosts and above, the reachability of our approach and flooding are very close. The reachability is close to 100%.

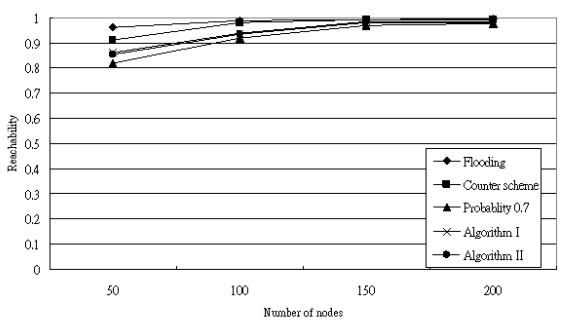

Fig. 4. The reachability of each algorithm.

B. *Analysis of Lifetime*

Fig. 5 shows the network lifetime improvement for each algorithm compared with flooding. Algorithm I and Algorithm II outperform the counter-based and probabilistic schemes as the power-balanced scheme is considered. In low-density network of 50 nodes, our two algorithms improve the network lifetime 10% than flooding. The improving rate of Algorithm II is 43% in high-density networks with 200 nodes, and the improving rate is 32% for Algorithm I. The counter-based scheme improves network lifetime from 7% to 26% and the probabilistic scheme improves from 5% to 13%. Therefore,

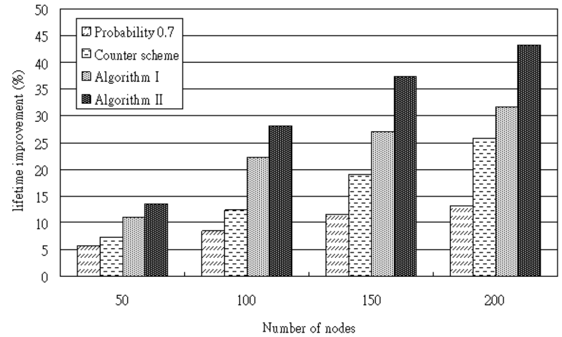

Fig. 5. The network lifetime improvement.

we conclude that our approaches have the better network lifetime than the probabilistic and the counter-based schemes.

Fig. 6 illustrates the surviving rate, the number of survived nodes divided by the total number of nodes, of each broadcast algorithm in low-density network of 50 nodes. Algorithm II has the best surviving rate and Algorithm I is the second one. The counter-based scheme has a better surviving rate than the probabilistic scheme and flooding. To broadcast by flooding, the first host in the network exhausts battery at 1200 second. In Algorithm I or Algorithm II, the first host is dead at 1400 second.

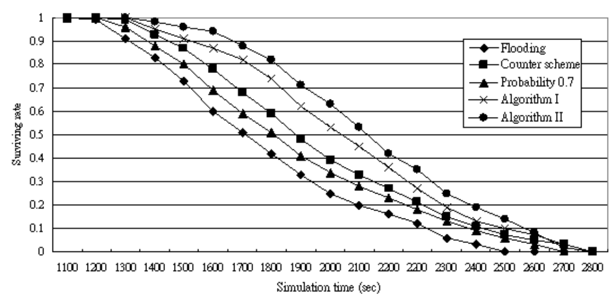

Fig. 6. The surviving rate of each algorithm in low-density network with 50 nodes.

Fig. 7 demonstrates the surviving rate of each broadcast algorithm in high-density network of 200 nodes. Comparing Fig. 6 to Fig. 7, we find out that the surviving rate of Algorithms I and II are outstanding in high-density networks. The counter-based scheme saves many rebroadcasts in high-density networks because it is easier to receive duplicated broadcast messages for a host in a higher density networks. Hence, Algorithm II is well-performed in high-density networks due to the use of the counter-based scheme.

C. *Saved Rebroadcast Ratio*

Fig. 8 explores the saved rebroadcast ratio (SRR) of each algorithm. The SRR of Algorithm I is 32% in low-density networks (50 nodes) and 42% in high-density networks (200 nodes). The SRR of Algorithm II is 33% in low-density network and 52% in high-density network. The SRR of the probabilistic scheme with the probability assigned to 0.7 in any

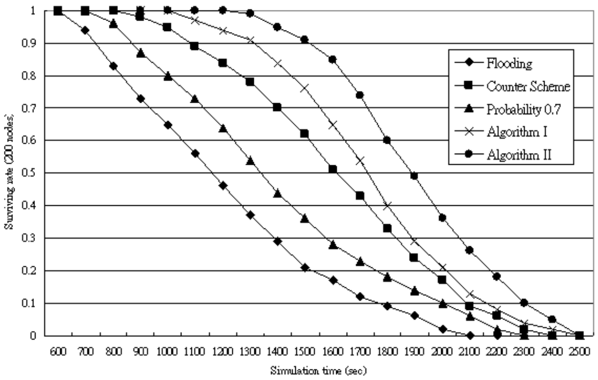

Fig. 7. The surviving rate of each algorithm in high-density network with 200 nodes.

density of network is around 30%. The SRR of the counter-based scheme increases in the high-density network. (e.g., the SRR increases from 30% to 43% in the high-density networks). In high-density networks of 200 hosts, the SRR of the counter-based scheme is a little higher than Algorithm I. Finally, we can see that Algorithm II performs the best in various network densities.

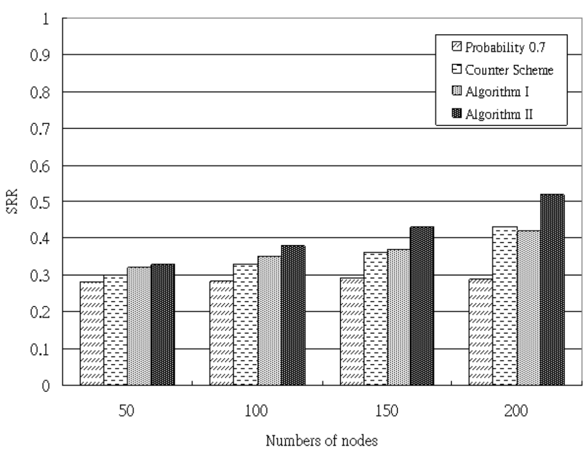

Fig. 8. The save rebroadcast ratio (SRR) of each algorithm.

D. *The Standard Deviation of Remaining Battery Energy*

Smaller standard deviation means that the remaining energy of all hosts are similar, and can prolong the whole network lifetime. In Fig. 9, the simulation results demonstrate that the Algorithm I has the lowest standard deviation among all hosts. The standard deviation of Algorithm I is 0.25 and the average standard deviation of Algorithm II is 0.3. Although the network lifetime of Algorithm II is greater than Algorithm I, the standard deviation is little higher than Algorithm I. The average standard deviation of the probabilistic and the counter-based schemes are both 0.38 and it is almost identical with flooding because both schemes do not consider the power-balance issue.

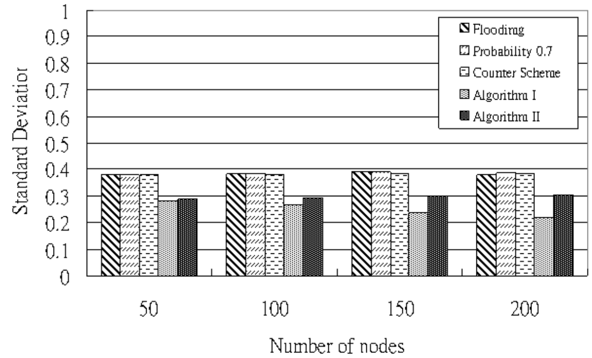

Fig. 9. The standard deviation of the remaining energy among all hosts.

IV. Conclusions

In this paper, we propose two power-balance broadcast algorithms in wireless networks to extend the network lifetime. We determine the rebroadcast probability by considering both of the remaining energy information and the network density. In order to prolong the network lifetime, we increase the rebroadcast probability of high-energy hosts and decrease the rebroadcast probability of low-energy hosts to balance the residual battery energy among all hosts. Compared with the flooding approach, the simulation results show that our proposed power-balance broadcast algorithms can improve the network lifetime to 42%, decrease standard deviation to 35%, and save half of the rebroadcasts. Regarding the network lifetime, our algorithms also perform better than the probabilistic and the counter-based schemes. Our approaches are not only successfully balance the battery energy of hosts but can extend the network lifetime without scanting the reachability, even in the environment of high mobility and density.

Acknowledgement

This work was supported by the Ministry of Education, the Republic of China, under Grant A-92-H-FA07-1-4 (Learning Technology).

References

[1] J. Broch, D. B. Johnson, and D. A. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," *Internet Draft*, draft-ietf-manet-dsr-00.txt, 1998.

[2] Z. J. Haas and M. R. Pearlman, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," *Internet Draft*, draft-haas-zone-routing-protocol-00.txt, 1997.

[3] P.-K. Hung, J.-P. Sheu, and C.-S. Hsu, "Scheduling of Broadcasts in Multihop Wireless Networks," *in Proceeding of the International Conference on European Wireless 2002*, pp. 163-169, 2002.

[4] M. Jiang, J. Li, and Y. C. Tay, "Cluster Based Routing protocol (CBRP) functional specification," *Internet Draft*, draft-ietf-manet-cbrp-spec-00.txt, 1998.

[5] W. Lou and J. Wu, "On Reducing Broadcast Redundancy in Ad Hoc Wireless Networks," *in Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, pp. 305-314, 2003.

[6] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint Relay:An Efficient Technique for Flooding in Mobile Wireless Networks," *INRIA Research Report*, RR-3898, 2000.

[7] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The Broadcast Storm Problem in a Wireless Mobile Ad Hoc Network," *in Proceedings of the International Conference on Mobile Computing and Networking*, pp. 151-162, Aug. 1999.

[8] C. E. Perkins and P. Bhagwat, ”Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers,” *ACM SIGCOMM ’94 Computer Communications Review 24(4)*: pp. 234 - 244, Oct. 1994.
[9] Y.-C. Tseng, S.-Y. Ni, and E.-Y. Shih, ”Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multi-hop Mobile Ad Hoc Network,” *in Proceedings of the 21st International Conference on Distributed Computing Systems*, pp. 481-488, 2001.
[10] C.-C. Yang and C.-Y. Chen, ”A Reachability-Guaranteed Approach for Reducing Broadcast Storms in Mobile Ad Hoc networks,” *in Proceedings of the 56th International Conference on Vehicular Technology*, pp. 1036-1040, vol.2, 2002.
[11] LAN MAN Standards Committee of the IEEE Computer Society, editor. IEEE STD 802.11-1997, ”Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” Nov, 1997.
[12] ”The Network Simulator-Glomosim”, http://pcl.cs.ucla.edu/projects/glomosim/