國立交通大學

網路工程研究所

碩 士 論 文

行動隨意網路下之調節式邊緣認知廣播策略

An Adaptive Border-aware Broadcast in Mobile Ad Hoc Networks

研 究 生：林志樺

指導教授：王國禎　教授

中 華 民 國 九 十 六 年 六 月

行動隨意網路下之調節式邊緣認知廣播策略
An Adaptive Border-aware Broadcast in Mobile Ad Hoc Networks

研　究　生：林志樺　　Student：Chih-Hua Lin

指導教授：王國禎　　Advisor：Kuochen Wang

國　立　交　通　大　學
網　路　工　程　研　究　所
碩　士　論　文

A Thesis

Submitted to Institute of Network Engineering

College of Computer Science

National Chiao Tung University

In Partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

行動隨意網路下之調節式邊緣認知廣播策略

學生：林志樺　　指導教授：王國禎 教授

國立交通大學網路工程研究所

摘 要

在行動隨意網路中，對於許多應用而言廣播是一項必要功能，例如，路由搜尋、位址解析等應用。傳統的廣播方法因為重覆廣播、競爭及碰撞等問題而浪費很多網路資源，所以已經有很多方法被提出來解決這些問題。但是很少有方法可以同時適用稀疏式和密集式網路。在本篇論文中，我們提出一個調節式邊緣認知的廣播策略 (ABB)，ABB 只需要 1-hop 節點的資訊，因此可以減少控制訊息過載。我們提出了兩個改進方法：第一個方法稱為稀疏模式改進，在稀疏式網路中這個改進藉由減少取消的範圍來增加到達率。第二個方法稱為密集模式改進，在密集式網路中藉由這個改進可以減少重覆廣播的範圍來獲得高的重覆廣播節省率。評估結果顯示，我們所提出的 ABB 方法在所有網路環境下都可以有較高的到達率，所以無論是在密集或稀疏式行動隨意網路之下，我們的方法均可適用。就平均而言，在到達率方面，ABB 可以達到和 ACB 一樣的水準，然而可以比 DFCN 好 33%。在平均廣播封包的延遲時間方面，ABB 比 ACB 好 33%，而且比 DFCN 好 7%。另外，在重覆廣播節省率方面，ABB 比 ACB 好 16%，但是比 DFCN

差 35%。然而 DFCN 有比較低的到達率，而對於評估廣播方法而言，到達率是一個比較重要的參數。

關鍵詞：廣播、廣播風暴、行動隨意網路

An Adaptive Border-aware Broadcast in Mobile Ad Hoc Networks

Student: Chih-Hua Lin **Advisor: Kuochen Wang**

Department of Computer Science

National Chiao Tung University

Abstract

In mobile ad hoc networks (MANETs), broadcast is an essential function for many applications, such as route discovery and address resolution. Since conventional broadcast in MANETs wastes a lot of resources due to redundant rebroadcasts, contention and collision, several existing broadcast protocols were proposed to resolve these problems. However, few of them are suitable to sparse networks as well as dense networks. In this thesis, we propose an *Adaptive Border-aware Broadcast* (ABB) that requires only 1-hop neighbor information so as to reduce control overhead. Two enhancements are proposed. The first enhancement, called *sparse mode enhancement*, can increase the reachability by decreasing the "cancellation range" in sparse networks. The second enhancement, called *dense mode enhancement*, can reduce the number of rebroadcast nodes in order to have high saved rebroadcast by decreasing the "rebroadcast range" in dense networks. Simulation results show that the proposed ABB can achieve high reachability in all network environments. Therefore, our ABB is suitable for both dense and sparse MANETs. Averagely, ABB is comparable to ACB and is 33% better than DFCN in terms of reachability. For average latency, ABB is 33% better than ACB and 7% better than DFCN. In addition, for saved rebroadcast, ABB is 16% better than ACB but 35% worse than DFCN. However, DFCN has poor reachability, which is a more important metric for evaluating broadcast schemes.

Keywords: broadcast, broadcast storm, mobile ad hoc network (MANET).

Acknowledgements

Many people have helped me with this thesis. I deeply appreciate my thesis advisor, Dr. Kuochen Wang, for his intensive advice and instruction. I would like to thank all the classmates in Mobile Computing and Broadband Networking Laboratory for their invaluable assistance and suggestions. The support by the National Science Council under Grant NSC95-2221-E-009-022 is also grateful acknowledged.

Finally, I thank my Father and Mother for their endless love and support.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

A Mobile Ad Hoc Network (MANET) comprises a set of wireless mobile nodes without any infrastructure support. The wireless mobile nodes are autonomous. No centralized management is required. The topology of a MANET is dynamically changed because mobile nodes can move freely. A mobile node can communicate with other mobile nodes in single-hop or multi-hop fashion. The MANET can be applied to areas where fixed network infrastructures are difficult or unavailable to setup (e.g., battlefield or disaster relief), or to areas for convenience consideration (e.g., interactive conferencing or home networking).

In the MANET, broadcast is an essential function and is frequently used. Some applications, such as route discovery, sending an alarm signal or address resolution, in MANETs require broadcast to disseminate messages to each node. Flooding is a conventional broadcast protocol that each node will rebroadcast the broadcast packet when it receives a broadcast packet for the first time. Flooding is very expensive due to redundant rebroadcasts, contention and collision, which is termed as the broadcast storm problem [1].

Several broadcast protocols were proposed to resolve this problem. The broadcast protocols can be divided into three categories according to the required information: *local knowledge-based broadcast* [1][2][3], *neighbor-based broadcast* [4]-[10] and *location-based broadcast* [11][12]. In the local knowledge-based broadcast protocols, no neighbor information and location information will be maintained. Thus the local knowledge-based broadcast protocols have low control overhead and require less information. However, a disadvantage of local knowledge-based broadcast protocols is that the reachability will degrade sharply in sparse networks. The neighbor-based broadcast protocols have high control overhead because each node needs to periodically send HELLO messages in order to maintain neighbor information. The location-based broadcast protocols need GPS equipment or some localization techniques to get location information. Therefore, the location-based broadcast protocols are very costly.

In this thesis, we propose a neighbor-based knowledge-based broadcast approach, called *Adaptive Border-aware Broadcast* (ABB). The basic idea is that for a node in a sparse (dense) network, the sparse (dense) mode enhancement is enabled to have high reachability (saved rebroadcast). The sparse mode enhancement and dense mode enhancement will be described later. Each node switching to either sparse mode or dense mode is based on its number of neighbors.

The rest of this thesis is organized as follows. Chapter 2 reviews some existing broadcast protocols. In Chapter 3, our approach is proposed. Simulation is shown in Chapter 4. Finally, concluding remarks and future work are given in Chapter 5.

Chapter 2
Related Work

In this chapter, we review some existing broadcast protocols and compare them qualitatively first.

2.1 Categories of Existing Broadcast Protocols

According to the required information, broadcast protocols can be classified into local knowledge-based broadcast, neighbor-based broadcast and location-based broadcast.

2.1.1 Local knowledge-based broadcast

In local knowledge-based broadcast protocols [1][2][3], each node does not need any neighbor information or location information, and uses only itself information to decide whether to rebroadcast the broadcast packet. Most of local knowledge-based broadcast protocols require nodes to wait for a random time interval for collecting more information in order to determine whether to rebroadcast the broadcast packet. That random time interval is termed Random Assessment Delay (RAD) [13]. In the following, we review some representative protocols in this category.

Probabilistic broadcast (PB) [1]: When a node receives a broadcast packet for the first time, it will rebroadcast the packet with a predetermined probability P. If P is set to 1, this scheme is equivalent to flooding.

Counter-based broadcast (CB) [1]: When a node receives a broadcast packet for the first time, it will initial a counter with one and wait for a RAD. During the RAD, if the node receives a redundant broadcast packet, the counter will increase by one. If the counter is equal to a counter threshold C, this node will cancel the rebroadcast of this broadcast packet. Otherwise, after the RAD expires, this node will rebroadcast the broadcast packet.

Border-aware broadcast (BB) [2]: When a node receives a broadcast packet for the first time, the node will wait for a *Directional Assessment Delay* (DAD) which is a time interval calculated according to the distance from its sender for collecting some information in order to determine whether to rebroadcast the broadcast packet. There is an inversely proportional relationship between the distance and the DAD. Before the DAD expires, if the node receives a redundant broadcast packet, it will cancel the rebroadcast of this broadcast packet. Otherwise, after the DAD expires, the node will rebroadcast this broadcast packet.

Hop count-aided broadcasting (HCAB) [3]: When a node receives a broadcast packet for the first time, it will wait for a RAD and record the hop count of this packet. Before the RAD is expired, if a node receives a redundant broadcast packet and the hop count of the received redundant broadcast packet is larger than the hop count received for the first time, this node will cancel the rebroadcast of this broadcast packet. Otherwise, after the RAD is expired, this node rebroadcasts the broadcast packet.

Self-adaptive probability broadcasting (SAPB) [3]: When a node receives a broadcast packet for the first time, it will wait for a RAD. After the RAD is expired, the node will rebroadcast the broadcast packet with a probability. The probability is self-adaptive according to number of received redundant packets during the RAD and the received signal strength. There is an inverse proportional relationship between the probability and the above two information (number of received redundant packets and received signal strength).

2.1.2 Neighbor-based broadcast

In neighbor-based broadcast protocols, each node must maintain its neighbor information for determining whether to rebroadcast the broadcast packet. To maintain the neighbor information, each node will periodically send a HELLO message. Therefore, each node can receive the HELLO messages sent from its neighbors. Then, each node can maintain its neighbor information according to the received HELLO messages.

The neighbor-based broadcast protocols can be divided into *proactive neighbor-based broadcas*t and *reactive neighbor-based broadcast* [14]. In proactive neighbor-based broadcast protocols [4][5], the source node will proactively choose some of its neighbors as "rebroadcast nodes" and send its broadcast packet. When the chosen "rebroadcast nodes" receive the broadcast packet, they will choose some of its neighbors as "rebroadcast nodes" again and rebroadcast the broadcast packet. In reactive neighbor-based broadcast protocols [6][7][8][9][10], the nodes will determine whether to rebroadcast the broadcast packet by themselves.

Most of neighbor-based broadcast protocols require 2-hop neighbor information [4][5][8]; that is, each node must know the neighbors of all its neighbors. The 2-hop neighbor information can be obtained by the received periodic HELLO messages. However, each HELLO message must contain 1-hop neighbor information. This causes high control overhead in 2-hop neighbor-based broadcast protocols.

Few of neighbor-based broadcast protocols require only 1-hop neighbor information to determine whether to rebroadcast the broadcast packet. The 1-hop neighbor-based broadcast protocols have reasonable control overhead compared to 2-hop neighbor-based protocols. There are some 1-hop neighbor-based broadcast protocols [6][7][9][10]:

Adaptive counter-based broadcast (ACB) [6]: The ACB improves the problem of CB [1] which poses a dilemma between reachability and the saved rebroadcast. In ACB, each node dynamically adjusts the counter threshold *C* based on its number of neighbors.

Self-pruning broadcast [7]: Each node will maintain its 1-hop neighbors' IDs via periodic HELLO messages. Upon a node wants to rebroadcast a broadcast packet, this node will piggyback its 1-hop neighbors' IDs in the header of this broadcast packet. When a node receives a broadcast packet, it will compare its 1-hop neighbors' IDs to neighbors' IDs in the header of the received broadcast packet. If all 1-hop neighbors' IDs of the node are covered by the neighbors' IDs in the packet header, the node will cancel its rebroadcast of this broadcast packet; otherwise the node will rebroadcast the broadcast packet.

Adjusted probabilistic broadcast (APB) [9]: The APB improves PB [1] by dynamically changing the probability *P* based on the number of neighbors. If a node's number of neighbors is larger than the average number of neighbors, the node sets a lower *P*. Otherwise, the node sets a higher *P*.

Delayed flooding with cumulative neighborhood (DFCN) [10]: In DFCN, once a node wants to rebroadcast a broadcast packet, it piggybacks its 1-hop neighbors' IDs in the header of each broadcast packet. When a node *s* receives a broadcast packet *m* for the first time, it will wait for a RAD and store the neighbors' IDs retrieved from *m* in a node list termed *K(m)*.

During the RAD, if node *s* receives the redundant broadcast packet *m*, it will cumulate the neighbors' IDs retrieved from *m* and the sender ID in *K(m)*. After the RAD is expired, node *s* will calculate a *benefit*. The benefit is defined as the ratio between the number of neighbors which do not belong to *K(m)* and the total number of neighbors. If the benefit is larger than a threshold value, node *s* will rebroadcast the broadcast packet *m*; otherwise, drop the broadcast packet *m*.

2.1.3 Location-based broadcast

The location-based broadcast protocols need GPS equipment or some localization techniques to get location information. There are some existing location-based broadcast protocols [11][12]:

Optimized broadcast protocol for sensor network (BPS) [11]: BPS assumes the network area can be portioned into many hexagons in ideal conditions and the transmission range of nodes is the hexagons length of sides. The source node is at the center of one of the hexagons. All the nodes at the vertexes of the hexagons (except the source node), which are called strategic locations, will rebroadcast the broadcast packet. However, in real situations, it is impractical to assume each node is located in the vertexes. A simple solution is to select the nearest node as the vertex.

Location-aided flooding (LAF) [12]: Each broadcast packet includes a special field in the packet header termed the *Node List*. The Node List contains the IDs of all the nodes that already have received the broadcast packet. If all 1-hop neighbors' IDs of a node are covered by the Node List, that node does not rebroadcast the broadcast packet. However, including the IDs of all the nodes that already have received the broadcast packet in the Node List is a large overhead. To reduce the overhead, LAF divides the whole network area into several "virtual grids." Each node joins a "virtual grids" depending on its location. The Node List only contains the IDs of the nodes within the same "virtual grid" that already have received the broadcast packet.

2.2 Qualitative Comparison of Different Broadcast Protocols

Some existing broadcast protocols are compared in Table 1. We used the following metrics for comparison: *required information*, control *overhead, piggybacking extra information in packets*, *reachability* [1], *saved rebroadcast* [1], average latency [1] and *RAD or DAD*.

The metric of *required information* indicates what kinds of information the broadcast protocol requires. *Control overhead* indicates that how many extra resources the broadcast protocol needs. DFCN, ACB and ABB have higher control overhead than BB, CB and BPS because they need to maintain 1-hop neighbor information. GSP has the highest control overhead because it needs to maintain 2-hop neighbor information. *Piggybacking extra information in packets* indicates if the broadcast protocol needs to include extra information into packets or not. GSP needs to include 1-hop neighbor information in the HELLO message, whereas BPS needs to include location information in the broadcast packet. DFCN needs to piggyback 1-hop neighbor information in the broadcast packet. *Reachability* (RE) indicates the ratio of the number of nodes that receive the broadcast packet to the number of nodes that are reachable, directly or indirectly from the source node in the network. *Saved rebroadcast* (SRB) is computed as $(r - t)/r$, where r is the number of nodes receiving the broadcast packet, and t is the number of nodes that actually rebroadcast the broadcast packet. *Average latency* indicates the average interval from the time the broadcast is being initiated to the time the last node is finishing its rebroadcast or deciding not to rebroadcast. *RAD or DAD* indicates the node will wait for a RAD or DAD when it receives a broadcast packet for the first time.

Table 1. Qualitative comparison of related work.

Approach	BB [2]	CB [1] (*C*=2)	GSP [8]	BPS [11]	DFCN [10]	ACB [6]	ABB (proposed)
Required information	Local knowledge	Local Knowledge	2-hop neighbors	Location	1-hop neighbors	the number of 1-hop neighbors	the number of 1-hop neighbors
Control overhead	Low	Low	High	Low	Medium	Medium	Medium
Piggybacking extra information in packets	No	No	Yes	Yes	Yes	No	No
Reachability	Medium in sparse networks	Low in sparse networks	High	High	Low in sparse networks	High	High
Saved rebroadcast	Medium	Medium	Medium	Medium	High	Medium in dense networks	High in dense networks
Average latency	Low	Medium	Medium	(not available)	High	Medium	Low
RAD or DAD	DAD	RAD	(not used)	DAD	RAD	RAD	DAD

Chapter 3

Design Approach

The primary objective of our proposed approach is to increase the reachability and the second objective is to increase the saved rebroadcast. In this chapter, we propose an *Adaptive Border-aware Broadcast* (ABB) in which each node enables either the sparse mode enhancement or the dense mode enhancement based on the number of 1-hop neighbors. We first introduce these two enhancements, and then our ABB is described.

3.1 Sparse Mode Enhancement

One problem of BB [2] in the sparse networks is that its reachability will degrade sharply. Although this problem was resolved in [2], it needs to use a decision table to decide if rebroadcast is necessary. We propose a simpler solution to conquer the problem. The low reachability problem is due to too many nodes canceling their rebroadcast. This problem is illustrated in Fig. 1. When node 1 sends a broadcast packet, the nodes in its transmission range will receive the broadcast packet for the first time. The nodes in node 1's transmission range will wait for a DAD. Nodes 2, 3 and 4 are outermost nodes from node 1; thus they wait for a shorter DAD. That is, they will rebroadcast their broadcast packets earlier. Other nodes in node 1's transmission range receive the redundant broadcast packet from nodes 2, 3 or 4. They will cancel the rebroadcast of this broadcast packet. As a result, nodes 5, 6 and 8 will not receive the broadcast packet.

Our solution is to decrease the "cancellation range" in order to increase the number of nodes that rebroadcast the broadcast packet. When nodes receive a redundant broadcast packet, only the nodes which locate within the "cancellation range" will cancel their rebroadcast. As shown in Fig. 2, the "cancellation range" of nodes 2, 3 and 4 are decreased from the circle plotted by the solid line to the circle plotted by the dotted line. Therefore, the rebroadcast of nodes 7 and 9 will not be canceled. Nodes 7 and 9 will rebroadcast the broadcast packet. Therefore, nodes 5, 6 and 8 will receive the broadcast packet.

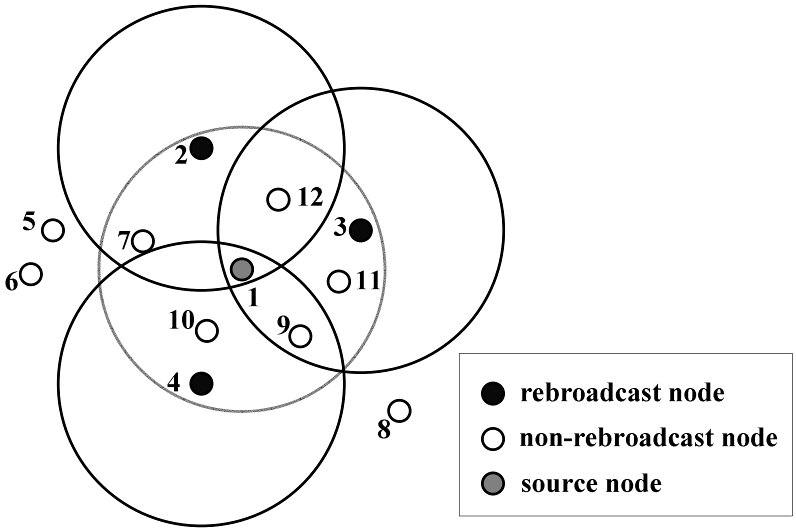

Fig. 1. The reachability problem.

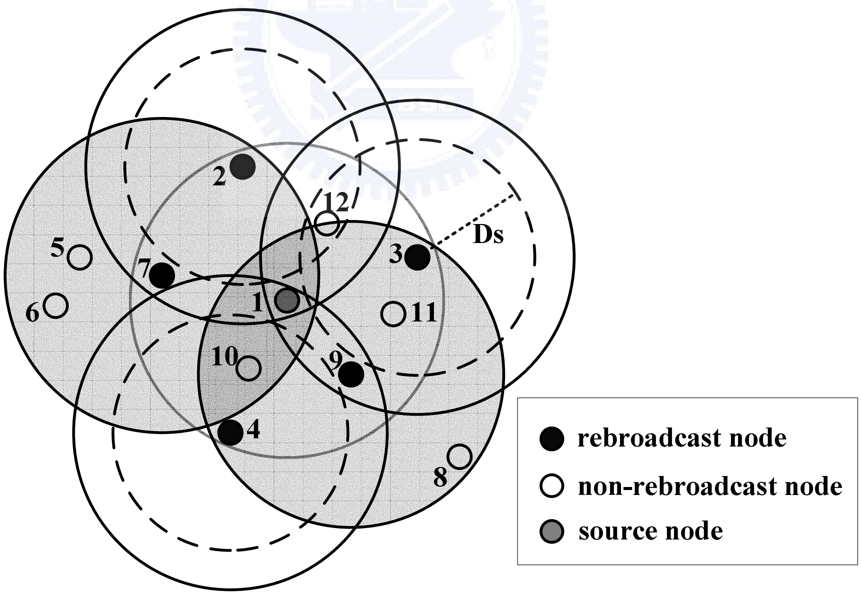

Fig. 2. The sparse mode enhancement.

To implement our sparse mode enhancement, we set a distance threshold, D_s. D_s can be set between 0 to R, where R is the transmission range of nodes. Fig. 3 is the flowchart of nodes in sparse mode enhancement. Each node will run the following procedure (S1 through S5):

S1. On receiving a broadcast packet *P* for the first time, the node sets a DAD.

S2. Wait for the DAD to expire. During the waiting, if the rebroadcast packet *P* is received again, interrupt the waiting and go to S4.

S3. Rebroadcast the broadcast packet *P*. Exit the procedure.

S4. If the distance from the sender < D_s, proceed to S5. Otherwise resume the interrupted waiting in S2.

S5. Cancel the rebroadcast of *P*. The node is inhibited from rebroadcast *P* in the future. Then, exit the procedure.

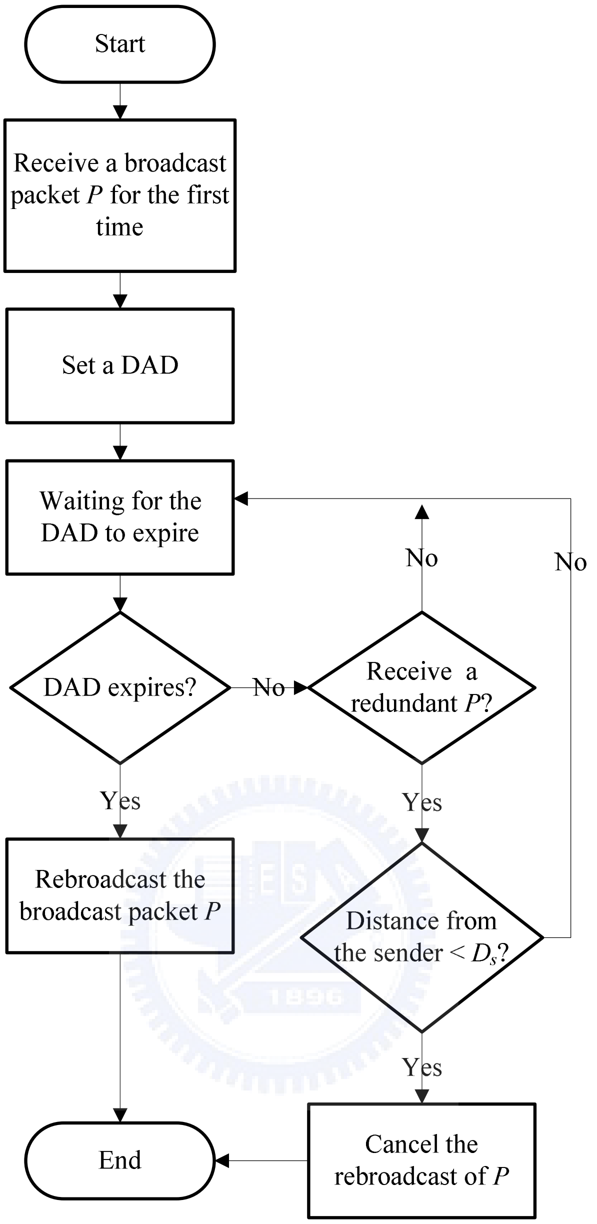

Fig. 3. The flowchart of node operation in sparse mode enhancement.

3.2 Dense Mode Enhancement

In dense networks, because the number of nodes is enough, the reachability problem is not the main issue. Therefore, our dense mode enhancement will focus on reducing the number of rebroadcast nodes by decreasing the "rebroadcast range." When nodes receive a broadcast packet for the first time, only nodes within the "rebroadcast range" will wait for a DAD to rebroadcast the broadcast packet whereas nodes not in the "rebroadcast range" cancel their rebroadcast of the broadcast packet. As shown in Fig. 4, the "rebroadcast range" of node

1 is decreased from the transmission range of node 1 (a circle) to the ring filled with gray color. When node 1 sends a broadcast packet, the nodes in the node 1's transmission range will receive the broadcast packet for the first time. Nodes within the dotted circle will cancel their rebroadcast. Since nodes 2, 3 and 4 locate within the "rebroadcast range" and they are the outmost nodes from the sender node 1, they wait for a shorter DAD. That is, they rebroadcast the broadcast packet earlier. Other nodes in the "rebroadcast range" will cancel their rebroadcast because they receive a redundant broadcast packet. As a result, only nodes 2, 3 and 4 rebroadcast the broadcast packet. Note that in the BB, nodes 2, 3, 4 and 5 will all rebroadcast the broadcast packet, whereas in our dense mode enhancement, only nodes 2, 3 and 4 will rebroadcast the broadcast packet. Our dense mode enhancement can help decrease the number of rebroadcast nodes.

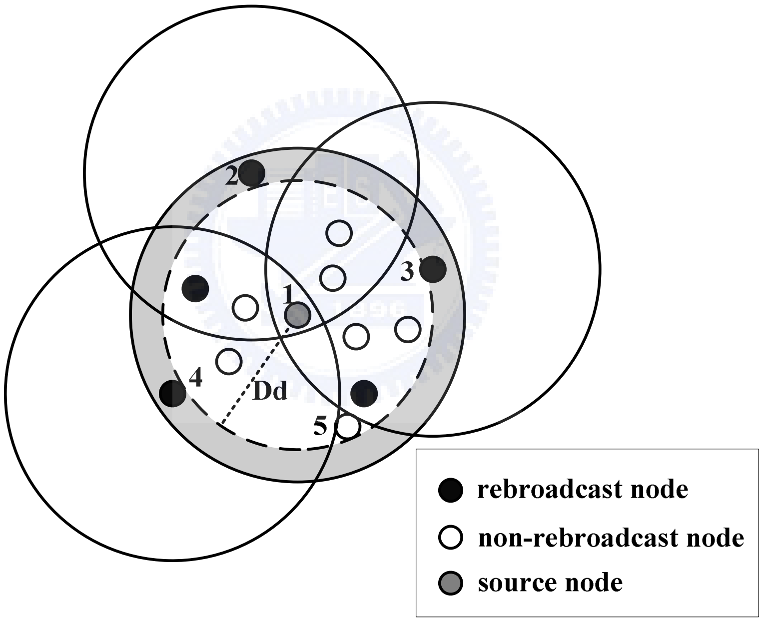

Fig. 4. The dense mode enhancement.

To implement our dense mode enhancement, we set a distance threshold, D_d. However, D_d can not be equal to R. We observed that if D_d is set to R, then no node locates within the "rebroadcast range," and no node will rebroadcast the broadcast packet. For this reason, D_d has an upper bound, termed as $maxD_d$. The $maxD_d$ is smaller than R. D_d can be set between 0 to $maxD_d$. The $maxD_d$ will be derived via simulation in Chapter 4.2. Fig. 5 is the flowchart of nodes in dense mode enhancement. Each node will run the following procedure (S1 through S4):

S1. On receiving a broadcast packet *P* for the first time, if the distance from the sender < D_d, exit the procedure. Otherwise, the node sets a DAD.

S2. Wait for the DAD to expire. During the waiting, if the broadcast packet *P* is received again, interrupt the waiting and go to S4.

S3. Rebroadcast the broadcast packet *P*. Exit the procedure.

S4. Cancel the rebroadcast of *P*. The node is inhibited from rebroadcast *P* in the future. Then, exit the procedure.

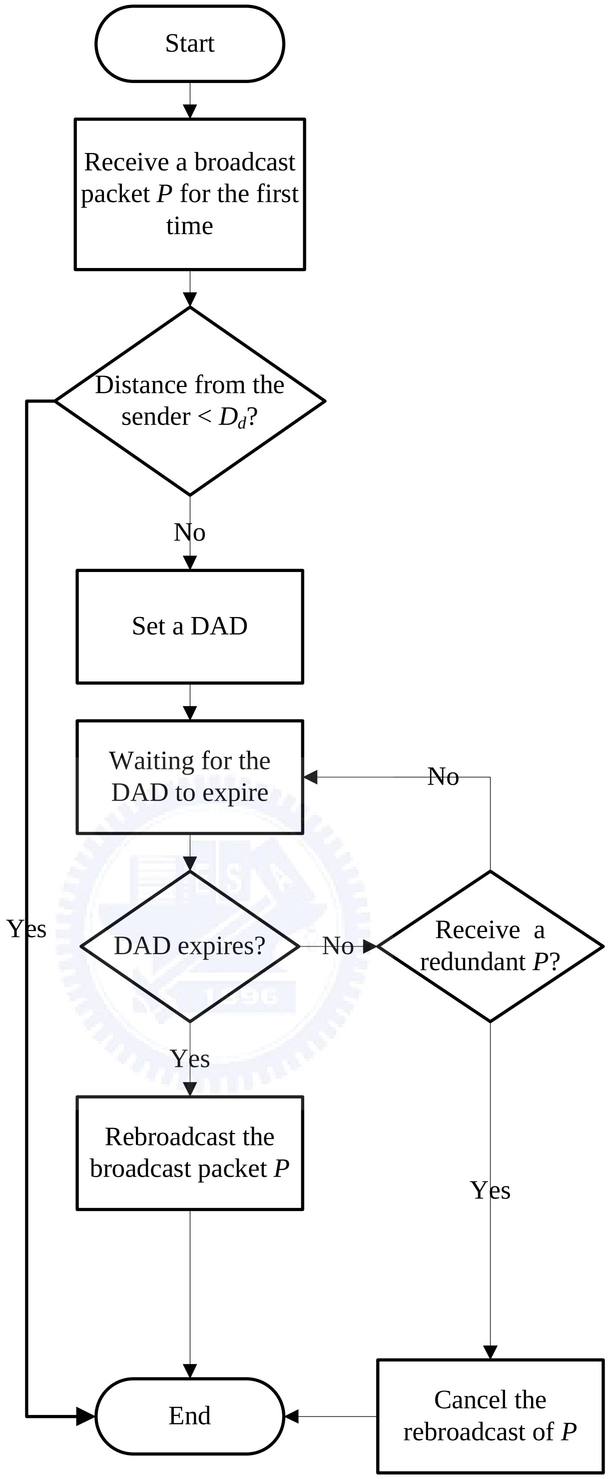

Fig. 5. The flowchart of node operation in dense mode enhancement.

3.3 Adaptive Border-aware Broadcast (ABB)

The proposed sparse mode enhancement and dense mode enhancement can only work well in sparse networks and dense networks, respectively. In addition, both enhancements use fixed thresholds D_s and D_d, regardless of sparse and dense degrees. For better performance, when a node is in a sparser network, D_s should be lower. When a node is in a denser network, D_d should be higher. To work well in all network environments, we propose an *adaptive border-aware broadcast* (ABB) in which each node will dynamically adjust the thresholds, D_s and D_d based on its number of neighbors. We extend fixed distance thresholds D_s and D_d into function $D_s(n)$ and function $D_d(n)$, where n is the number of neighbors. For estimating n, each node can send a HELLO message periodically. Each node can calculate n based on the received HELLO message. Fig. 6 is the flowchart of nodes in ABB, where the two enhancements are integrated. Each node will run the following procedure (S1 through S5):

S1. On receiving a broadcast packet P for the first time, if the distance from the sender < $D_d(n)$, exit the procedure . Otherwise, the node sets a DAD.

S2. Wait for the DAD to expire. During the waiting, if the broadcast packet P is received again, interrupt the waiting and go to S4.

S3. Rebroadcast the broadcast packet P. Exit the procedure.

S4. If the distance from the sender < $D_s(n)$, proceed to S5. Otherwise resume the interrupted waiting in S2.

S5. Cancel the rebroadcast of P. The node is inhibited from rebroadcast P in the future. Then, exit the procedure.

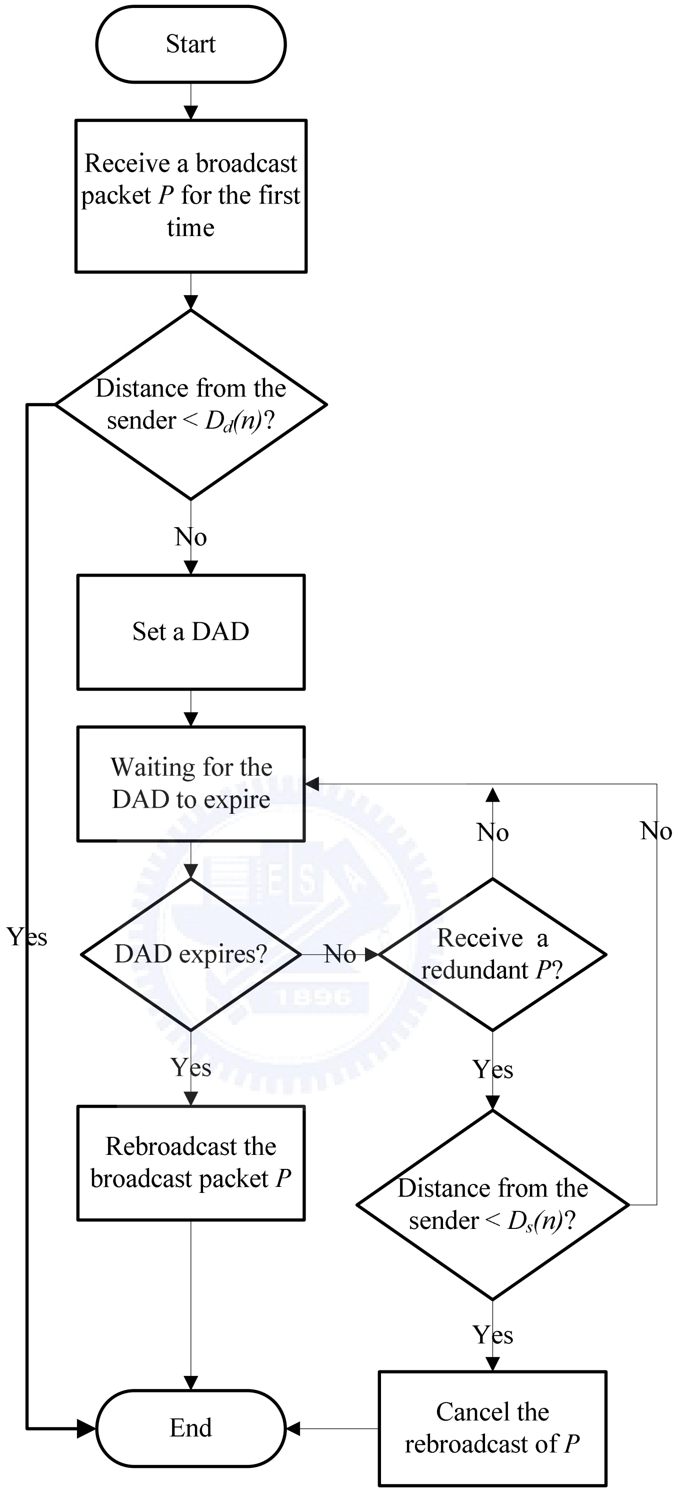

Fig. 6. The flowchart of node operation in ABB.

Compared to the sparse mode enhancement and dense mode enhancement, we observed that when the function $D_d(n) = 0$, ABB is identical to the sparse mode enhancement. When the function $D_s(n)= R$, ABB is equal to the dense mode enhancement. We assume when number of neighbors of a node is smaller than n_1, which will be derived via simulation, the node is in the sparse network. Otherwise, the node is in the dense networks. Therefore, if number of neighbors are smaller than n_1, the function $D_d(n)$ of the nodes will be set to 0. If number of neighbors are larger than n_1, the function $D_s(n)$ of the nodes will set to R. Intuitively, fewer number of neighbors of a node means that the node is in the sparser network. The function value of $D_s(n)$ will be lower. On the contrary, more number of neighbors of a node means that the node is in the denser network. The function value of $D_d(n)$ will be higher. Based on he above observations, we define the functions of D_s(n) and D_d(n) as follows:

$$D_s(n) = \begin{cases} \left[\dfrac{1 \times (n-1)}{n_1 - 1} \right] \times R, & 1 \le n \le n_1 \\ R, & n > n_1 \end{cases} \tag{1}$$

$$D_d(n) = \begin{cases} 0, & n < n_1 \\ \dfrac{(n - n_1)}{n_2 - n_1} \times maxD_d, & n_1 \le n \le n_2 \\ maxD_d, & n > n_2 \end{cases} \tag{2}$$

The curves in Fig. 7 are the shapes of functions D_s(n) and D_d(n). In Chapter 4.3, we will derive the values of n_1 and n_2 via simulation [6].

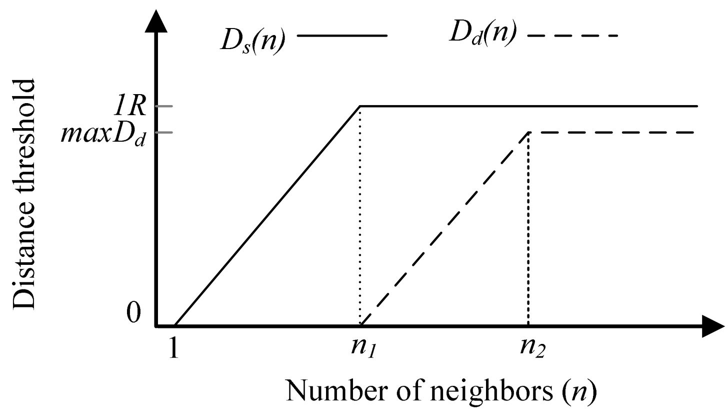

Fig. 7. The functions of $D_s(n)$ and $D_d(n)$.

Chapter 4

Simulation Results and Discussion

To evaluate our proposed approach, ABB, three performance metrics are observed: REachability (RE), Saved ReBroadcast (SRB) and average latency. RE, SRB and average latency have been defined in Chapter 2.2; in this chapter, we evaluate our proposed approaches: sparse mode enhancement, dense mode enhancement and ABB based on the above three performance metrics and compare them with existing methods, BB [2], ACB [6] and DFCN [10]. In addition, we will derive the best values of n_1 and n_2 for ABB. Our simulation was implemented in the GloMoSim simulator [15] (version 2.03), which is a scalable simulation environment for wireless and wired systems.

4.1 Simulation Model

In our simulation model, the fixed parameters [6] are DSSS physical layer timing, the transmission range of nodes (500 meters), the broadcast packet size (280 bytes) and the transmission rate (1Mbps). The interval to send the HELLO message is 40 seconds. First, to evaluate sparse and dense networks, the network sizes can be 1 × 1, 3 × 3, 5 × 5, 7 × 7, 9 × 9, or 11 × 11 [6], where a unit is 500 meters and 100 nodes are randomly placed in the network area [6]. Second, to evaluate the effect of a different number of nodes, the network size is fixed to 5 × 5 and the number of nodes can be 40, 60, 80, … ,160, 180 and 200. For mobility, we employed the random waypoint mobility model [16], in which each node randomly chooses a destination and moves toward the destination with a randomly chosen speed (from 0 to a given maximum speed). The maximum speed is 10 × n km/hour in the n × n network where n = 1, 3, 5, 7, 9 and 11 [6].

4.2 Performance Evaluation of Sparse Mode and Dense Mode Enhancements

First, we compare the BB [2] with our sparse mode enhancement and dense mode enhancement under different network sizes.

Fig. 8 shows the sparse mode enhancement can achieve higher RE than BB, but the SRB will be sacrificed. On the other hand, the dense mode enhancement can provide higher SRB than BB, but the RE will decrease sharply in sparse networks. Therefore, the sparse mode enhancement indeed can work well in sparse networks, and the dense mode enhancement can work well in dense networks.

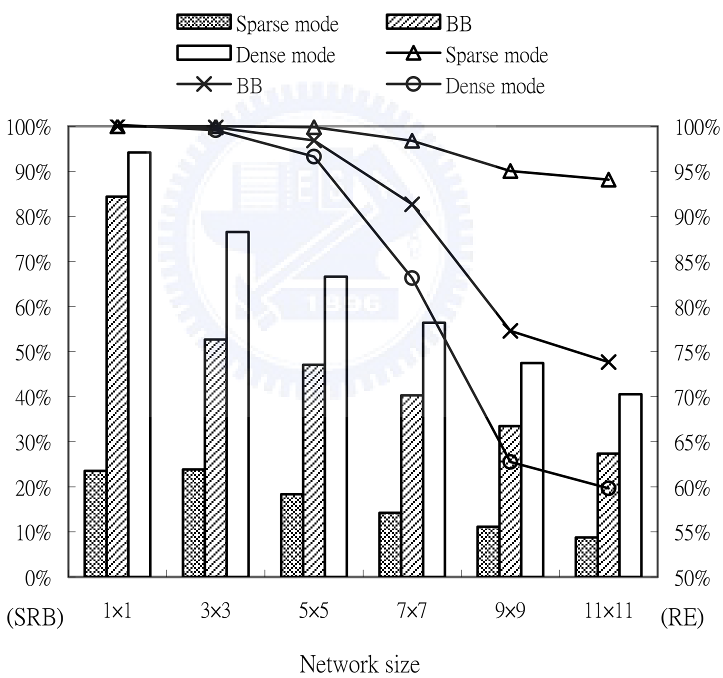

Fig. 8. Comparison of the sparse mode enhancement ($D_s = 0.5R$), dense mode enhancement ($D_d = 0.5R$) and BB for SRB (shown in bars) and RE (shown in lines) under different network sizes.

Fig. 9 shows the average latency of the sparse mode enhancement, dense mode enhancement and BB. We found that when the network size increases, average latency increases except that in sparser networks. Because there are too many isolated nodes in sparser networks, average latency will decrease sharply. The sparse mode enhancement has longer average latency than BB because the sparse mode enhancement increases the number of nodes for rebroadcasting the broadcast packet in order to resolve the reachability problem. The dense mode enhancement also increases the average latency compared to BB. Because the dense mode enhancement decreases the "rebroadcast range" of each node, the DAD is more sensitive to the distance. Averagely, the DAD of dense mode enhancement is larger than that of BB.

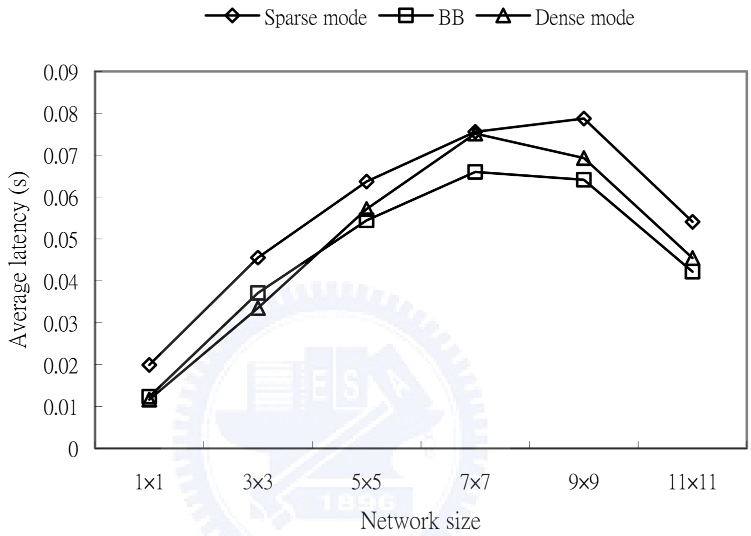

Fig. 9. Comparison of the sparse mode enhancement (D_s = *0.5R*), dense mode enhancement (D_d = *0.5R*) and BB for average latency (in seconds) under different network sizes.

In the following, we determine the value of $maxD_d$ mentioned in Chapter 3.2.

Fig. 10 shows that the SRB and RE of our dense mode enhancement under different D_d When D_d is larger, SRB is higher and RE is lower. However, when D_d is equal to R, the RE will decrease sharply in dense networks. That is, our dense mode enhancement can not work well when the D_d is close to R. For this reason, we set the value of $maxD_d$ is *0.9R*.

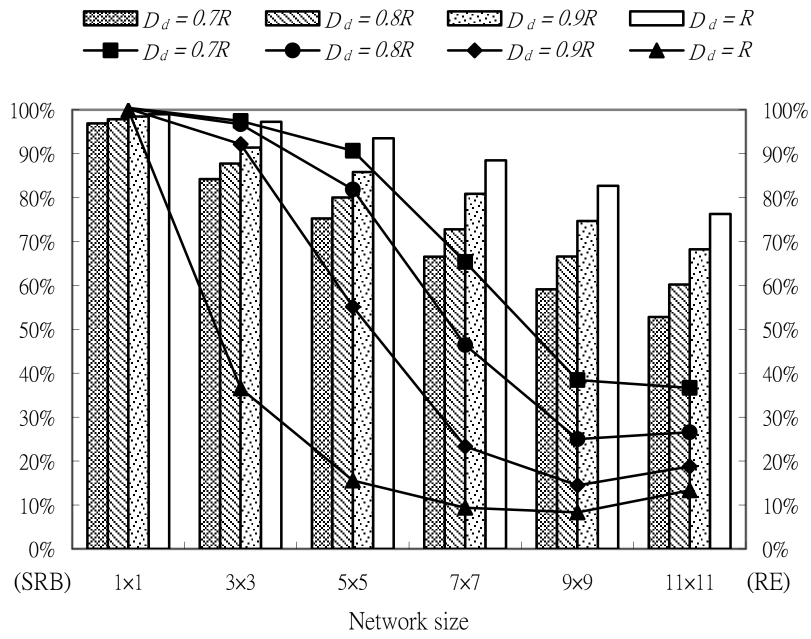

Fig. 10. Determining the value of $maxD_d$ for dense mode enhancement.

4.3 Performance Evaluation of the Adaptive Border-aware Broadcast

In the following, we show how to determine the best values of n_1 and n_2 for the proposed ABB. Firstly, we determine the value of n_1. In Fig. 11, we varied the value of n_1 (= 8, 10, 12 and 14) and did not consider n_2. The function $D_s(n)$ defined in Chapter 3.3 was used, but the function $D_d(n)$ was fixed to 0. The results in Fig. 11 indicate that $n_1 = 12$ or 14 can get the best RE. To have a higher SRB, we set the value of n_1 to 12.

Secondly, we determined the value of n_2 in Fig. 12. We fixed the value of $n_1 = 12$ and varied the value of n_2 (= 26, 28 and 30). Again, the functions $D_s(n)$ and $D_d(n)$ defined in Chapter 3.3 were used. Observing the results, we found that when the value of $n_2 = 26$, the RE will decrease when the network size is 3 × 3. Therefore, only $n_2 = 28$ and $n_2 = 30$ can achieve a higher RE. To have a higher SRB, we set the value of n_2 to 28. Finally, we compare our proposed ABB with DFCN [10] and ACB [6], which are also 1-hop neighbor-based broadcast protocols for MANETs.

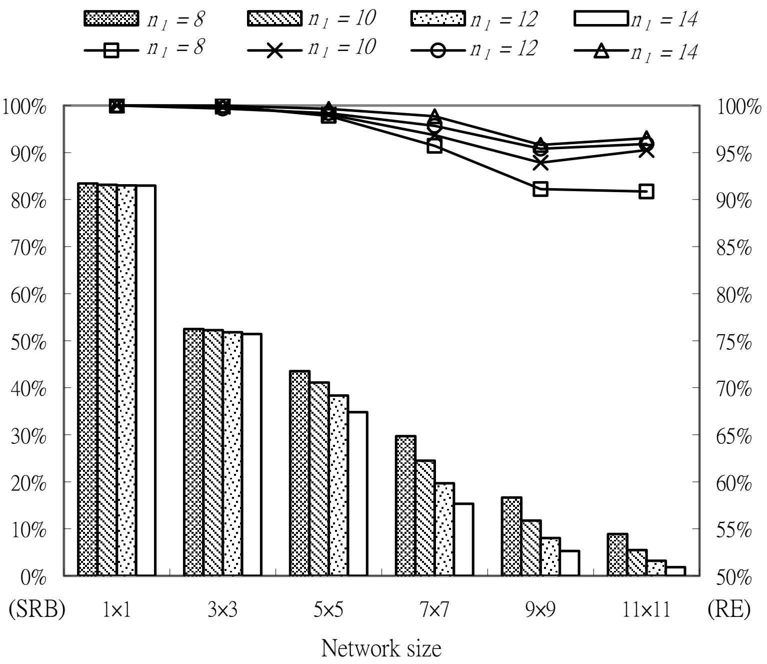

Fig. 11. Determining the value of n_1 for ABB.

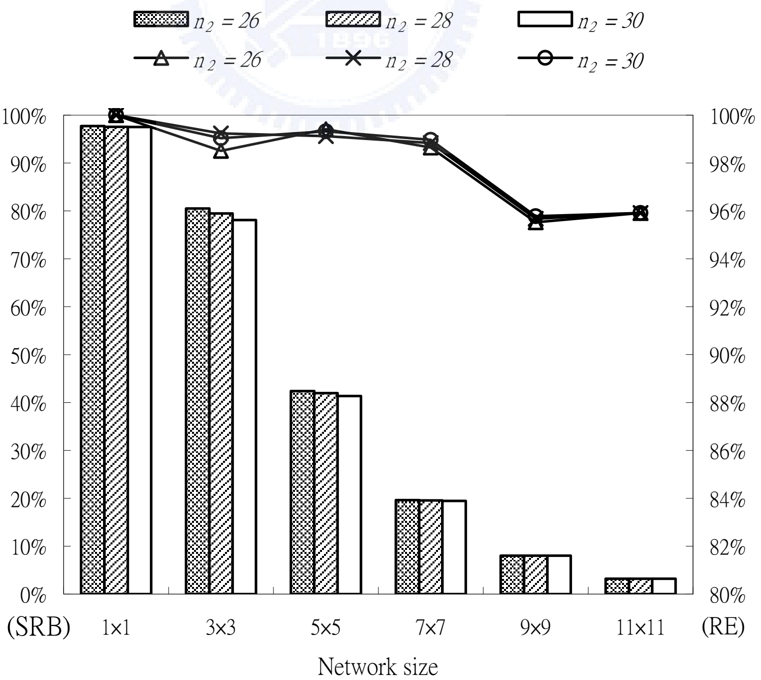

Fig. 12. Determining the value of n_2 for ABB.

Fig. 13 shows that our ABB is comparable to ACB and is 33% better than DFCN in terms of RE. For SRB, our ABB is 16% better than ACB but 35% worse than DFCN. Although the DFCN can maintain high SRB in all network sizes, the RE of DFCN decreases sharply in sparse networks because DFCN uses a fixed threshold value. Because the RE of DFCN is too low in sparse networks, DFCN can only work well in dense networks. Note that the proposed ABB has 4% better SRB than DFCN in dense networks averagely.

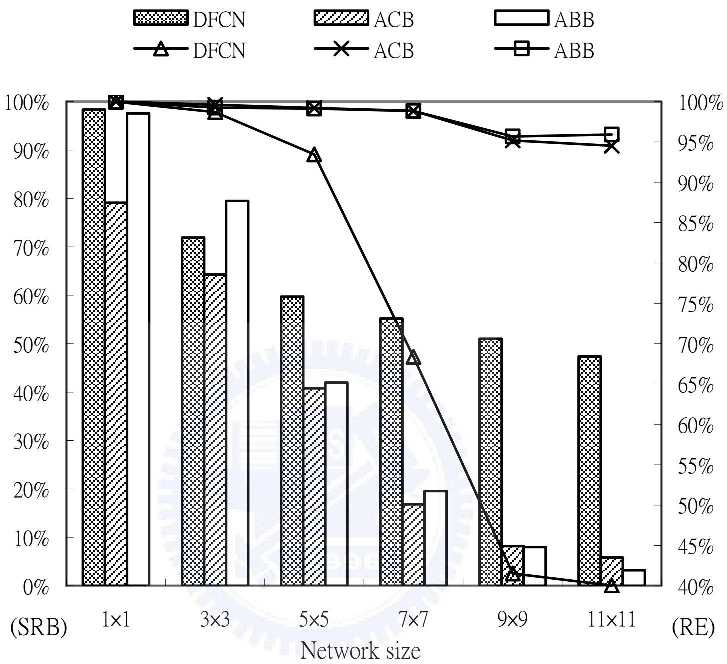

Fig. 13. Comparison of ABB, ACB and DFCN for SRB (shown in bars) and RE (shown in lines) under different network sizes.

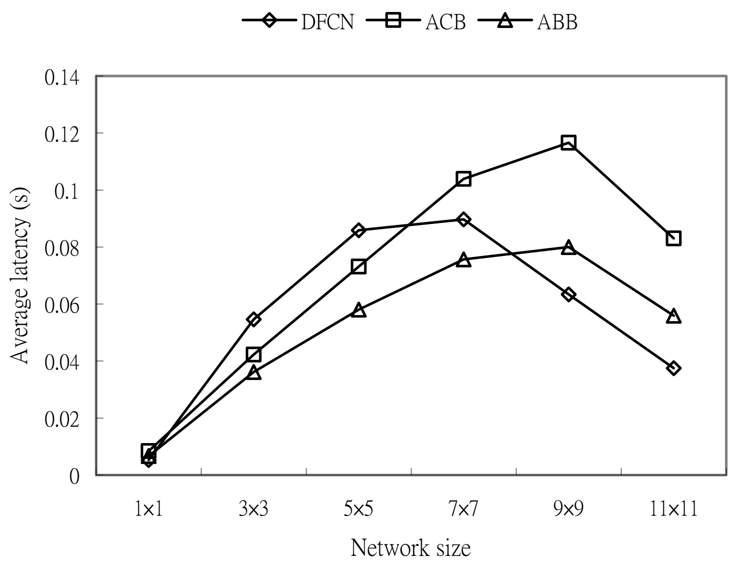

Fig. 14. Comparison of the ABB, ACB and DFCN for average latency (in seconds) under different network sizes.

Fig. 14 compares the average latency of ACB, DFCN and the proposed ABB. Because our ABB uses DAD instead of RAD, the node which is farther from its sender rebroadcasts the broadcast packet faster. As a result, the average latency of our ABB is shorter than ACB. The average latency of DFCN is longer than that of ACB and ABB in dense networks because the transmission time of DFCN is prolonged due to its piggybacking of 1-hop neighbors' IDs in the broadcast packet. However, we found that DFCN has shorter average latency than ACB and our ABB in sparse networks. This is because DFCN has a lower RE in sparse networks; thus the number of nodes which receive the broadcast packet is lower. As a result, the number of nodes which rebroadcast the broadcast packet is also lower. Therefore, the average latency is shorter as well. Nevertheless, our proposed ABB is 33% better than ACB and 7% better than DFCN in terms of average latency for both sparse and dense networks as a whole.

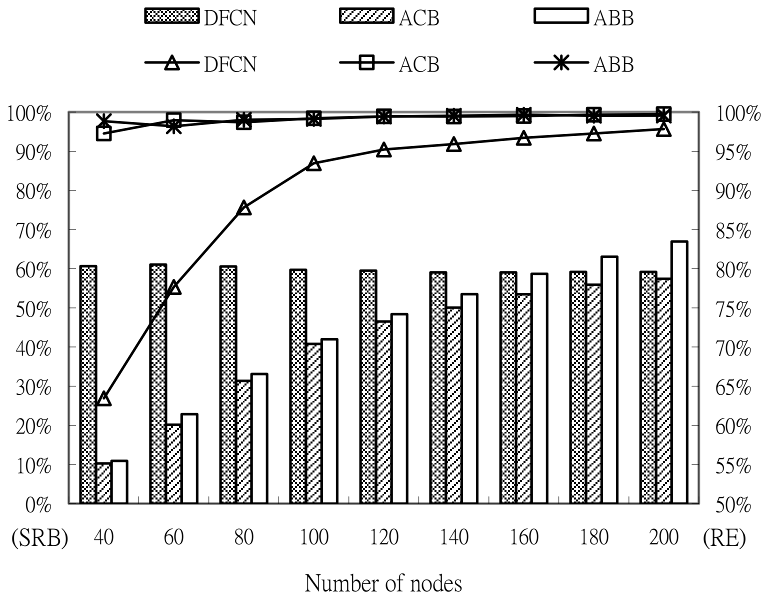

Fig. 15. Comparison of ABB, ACB and DFCN for SRB (shown in bars) and RE (shown in lines) under different number of nodes.

To evaluate the effect of a different number of nodes on broadcast performance, the network size was fixed to 5 × 5 and the number of nodes was varied from 40 to 200. In Fig. 15, it shows that ABB is comparable to ACB and is 14% better than DFCN in terms of RE. For SRB, our ABB is 8% better than ACB but 33% worse than DFCN. However, when number of nodes is small, the RE of DFCN decreases sharply. Therefore, DFCN can only work well when number of nodes is large enough.

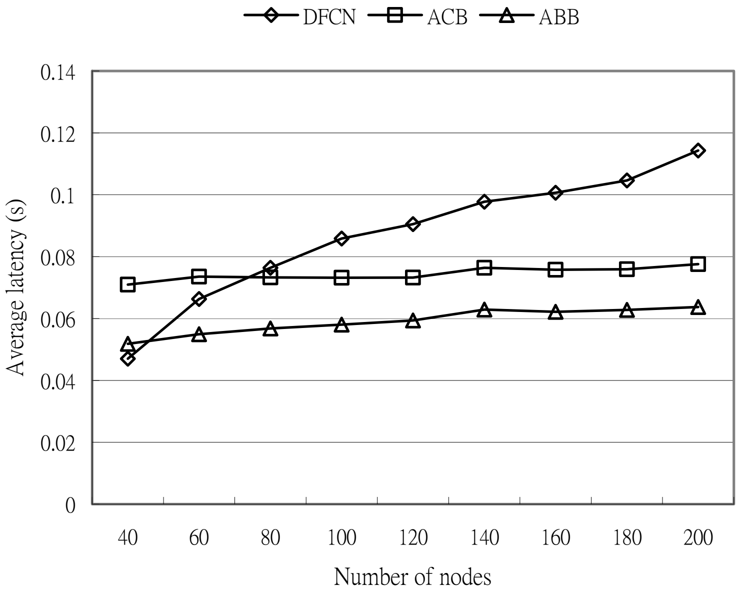

Fig. 16. Comparison of the ABB, ACB and DFCN for average latency (in second) under different number of nodes.

Fig. 16 compares average latency. The average latency of the proposed ABB is the shortest because of DAD. When the number of nodes increases, the average latency of ACB and our ABB increases slowly because of the fixed network size, while the average latency of DFCN increases faster because RE and transmission time increase. Averagely, our proposed ABB is 22% better than ACB and 31% better than DFCN in terms of average latency.

Chapter 5
Conclusion and Future Work

5.1 Concluding Remarks

Broadcast is an essential function for many applications in MANETs, such as route discovery or address resolution. In this thesis, we have presented a neighbor-based broadcast approach, called A*daptive Border-aware Broadcast* (ABB). The basic idea of our approach is that for a node in a sparse (dense) network, the sparse (dense) mode enhancement is enabled. The simulation results reveal that our ABB can achieve higher RE in all network environments and has higher SRB in dense networks. This implies that our ABB is suitable for both dense and sparse MANETs. In summary, the propose ABB is comparable to ACB [6] and 33% better than DFCN [10] in terms of reachability. For average latency, ABB is 33% better than ACB and 7% better than DFCN. In addition, for saved rebroadcast, ABB is 16% better than ACB but 35% worse than DFCN. However, DFCN has poor reachability, which is a more important metric for evaluating broadcast schemes.

5.2 Future Work

To estimate number of neighbors, each node can send a HELLO message periodically. Each node can calculate number of neighbors based on the received HELLO message. However, collision of HELLO messages may happen especially in dense networks. For this reason, the estimated number of neighbors may be lower than the actual number of neighbors. To have a better performance, we can further study functions $D_s(n)$ and $D_d(n)$ by considering the collision of HELLO messages problem. In addition, we can also consider energy-aware broadcast to prolong the overall lifetime of MANETs.

Bibliography

[1] Y.C. Tseng, S.Y. Ni, Y.S. Chen, and J.P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless Networks*, vol. 8, nos. 2-3, pp. 153-167, Mar.-May 2002.

[2] C. Zhu, M.J. Lee, and T. Saadawi, "A border-aware broadcast scheme for wireless ad hoc network," in *Proc. IEEE Consumer Communications and Networking Conference*, Jan. 2004, pp. 134-139.

[3] Q. Huang, Y. Bai, and L. Chen, "Efficient lightweight broadcasting protocols for multi-hop ad hoc networks," in *Proc. IEEE Int. Symp. on Personal, Indoor and Mobile Radio Communications*, Sept. 2006, pp. 1-5.

[4] W. Peng and X. Lu., "AHBP: An efficient broadcast protocol for mobile ad hoc networks," *Journal of Science and Technology – Beijing*, China, 2002.

[5] W. Lou and J. Wu, "On reducing broadcast redundancy in ad hoc wireless networks," in *Proc. 36th Annual Hawaii International Conference on System Sciences*, pp. 10-20, Jan. 2003.

[6] Y.C. Tseng, S.Y. Ni, and E.Y. Shih, "Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network," *IEEE Transactions on Computers*, vol. 52, no. 5, pp. 545-557, May 2003.

[7] H. Lim and C. Kim, "Multicast tree construction and flooding in wireless ad hoc networks," in *Proc. ACM Int'l Workshop Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Aug. 2000, pp 61-68.

[8] J. Wu and F. Dai, "Broadcasting in ad hoc networks based on self-pruning," in *Proc. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 2240-2250, Mar.-Apr. 2003.

[9] L. M. M. M. Bani-Yassein, M. Ould-Khaoua, and S. Papanastasiou., "Performance analysis of adjusted probabilistic broadcasting in mobile ad hoc networks," *International Journal of Wireless Information Networks*, pp. 1-14, Mar. 2006.

[10] L. Hogie, M. Seredynski, F. Guinand, and P. Bouvry, "A bandwidth-efficient broadcasting protocol for mobile multi-hop ad hoc networks," in *Proc. International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies*, pp. 71-79, Apr.

2006.

[11] A. Durresi, V. Paruchuri, S. S. Iyengar, and R. Kannan, “Optimized broadcast protocol for sensor networks, *IEEE Transactions on Computers*, pp. 1013-1024, Aug. 2005.

[12] H. Sabbineni and K. Chakrabarty, "Location-aided flooding: An energy-efficient data dissemination protocol for wireless sensor networks," *IEEE Transactions on Computers*, vol. 54, no. 1, pp. 36-46, Jan. 2005.

[13] B. Williams and T. Camp, “Comparison of broadcasting techniques for mobile ad hoc networks,” *in Proc. 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2002, pp. 914-205.

[14] Y. Cai, K. A. Hua, and A. Phillip, ”Leveraging 1-hop neighborhood knowledge for efficient flooding in wireless ad hoc networks,” *in Proc. IEEE Int'l Performance Computing and Communications Conference*, Apr. 2005, pp. 347- 354.

[15] X. Zeng, R. Bagrodia, and M. Gerla, “GloMoSim: a library for parallel simulation of large-scale wireless networks,” in *Proc. 1998 12th Workshop on Parallel and Distributed Simulations*, May 1998, pp. 154-161.

[16] C. Bettstetter, G. Resta, and P. Santi, “The node distribution of the random waypoint mobility model for wireless ad hoc networks,” *IEEE Transactions on Mobile Computing*, vol. 2, no.3, pp. 257-269, July-Sept. 2003.