Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Network

Yu-Chee Tseng, *Senior Member*, *IEEE*, Sze-Yao Ni, *Member*, *IEEE*, and En-Yu Shih

Abstract—In a multihop mobile ad hoc network, *broadcasting* is an elementary operation to support many applications. In [15], it is shown that naively broadcasting by *flooding* may cause serious redundancy, contention, and collision in the network, which we refer to as the *broadcast storm problem*. Several threshold-based schemes are shown to perform better than flooding in that work. However, how to choose thresholds also poses a dilemma between reachability and efficiency under different host densities. In this paper, we propose several adaptive schemes, which can dynamically adjust thresholds based on local connectivity information. Simulation results show that these adaptive schemes can offer better reachability as well as efficiency as compared to the results in [15].

Index Terms—Broadcast, broadcast storm, communication, mobile ad hoc network (MANET), mobile computing, wireless network.

1 Introduction

The *mobile ad hoc network* (MANET) distinguishes itself from traditional wireless networks by its dynamic changing topology, no base-station support, and multihop communication capability. In a MANET, a mobile host is free to move around and may communicate with other hosts at any time. When a communicating partner is within a host's radio coverage, they can communicate directly in a single-hop fashion. Otherwise, a route consisting of several relaying hosts is needed to forward messages from the source to the destination in a multihop fashion. To support multihop communication in a MANET, a mobile host has to work as a router and cooperate with other hosts to find routes and relay messages. Routing has been studied intensively under a MANET environment (e.g., unicast [2], [3], [6], [8], [9], [10], [13], [18], multicast [4], [7], and geocast [11]). A working group called "manet" has been formed in the Internet Engineering Task Force (IETF) to stimulate research in MANET [5], [14]. Applications of MANETs appear in places where fixed network infrastructures are difficult to build (e.g., fleets in the ocean, air fighters in the sky, and soldiers on the march) or unavailable (e.g., rescue scenes and archaeological or ecological surveys).

In this paper, we study the *broadcasting* problem in a MANET. Broadcasting is a fundamental operation in all kinds of networks; it may be used for discovering neighbors, collecting global information, naming, addressing, and sometimes helping in multicasting. In a MANET particularly, several routing protocols [2], [8], [9], [10], [18] have relied on broadcasting to propagate routing-related information (e.g., the request for a new route to a destination). In most networks (including MANET), a common approach is to broadcast by *flooding*. While most existing works on MANET take flooding as a straightforward and direct solution, we show in [15] that a blind flooding may result in excessive *redundancy*, *contention*, and *collision*. These may lead to lower reachability (to the potential receiving hosts) and longer latency (for the broadcast to complete). (More discussion on this is in Section 2.) We thus refer to this scenario as the *broadcast storm problem*.

To alleviate the broadcast storm, one should inhibit redundant rebroadcasts of the broadcast packet and differentiate the timing of rebroadcasts. Following this guideline, several schemes, called the *counter-based*, *distance-based*, and *location-based* schemes, were proposed in [15]. These schemes rely on various threshold mechanisms to help a mobile host to assert the redundancy of a rebroadcast and decide whether to rebroadcast or not. Results did show that these schemes can effectively relieve the broadcast storm problem by delivering better reachability and lower latency as compared to flooding.

One problem associated with the above threshold-based schemes [15] is that the threshold which is used is a given constant. Given a fixed host density, we can easily determine a best threshold to use. However, since the topology of a MANET may change dynamically and quickly, it is desirable to be able to adjust the threshold on-the-fly. This is what motivates this paper. In this paper, we propose three dynamic solutions, called the *adaptive counter-based*, *adaptive location-based*, and *neighbor coverage* schemes. These schemes all take local connectivity information into account. The first two schemes dynamically choose

- *Y.-C. Tseng is with the Department of Computer Science and Information Engineering, National Chiao-Tung University, Hsin-Chu, 30050, Taiwan. E-mail: yctseng@csie.nctu.edu.tw.*
- *S.-Y. Ni is with DeanSoft Co., Ltd., 2F-2, No. 280, Lioufu Rd., Luju Shiang, Taoyuan, 338, Taiwan. E-mail: neesy@computer.org.*
- *E.-Y. Shih is with the Internet Application Technology Department, Computer & Communication Research Laboratories, Industrial Technology Research Institute, 195, Sec. 4, Chung Hsing Rd., Chutung, Hsinchu, Taiwan 310, ROC. E-mail: enyu@itri.org.tw.*

Manuscript received 19 Dec. 2000; revised 9 July 2001; accepted 28 June 2002. For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 113320.

their threshold values according to a host's number of neighbors. Thus, compared to the work in [15], where all hosts should take the same (constant) threshold, the newly proposed schemes allow hosts to take different thresholds. For each individual host, its threshold can even be changed adaptively according to its neighborhood. This significantly improves the limitation of the earlier work. The last scheme applies two-hop neighborhood information to decide whether a rebroadcast is necessary or not. Through simulations we justify that these schemes can further improve those threshold-based schemes. Since the neighbor coverage scheme relies on the accuracy of neighborhood information, we also study the relationship between host mobility and the interval for a host to send a hello packet to announce its existence (which we call the *hello interval*). Toward this goal, we propose a dynamic scheme to adjust a host's hello interval based on the variation of its neighborhood. We verify that this can reduce unnecessary hello packets and yet still provide up-to-date neighborhood information.

The rest of this paper is organized as follows: Section 2 discusses the broadcast storm problem and reviews some solutions to relieve the storm. Our adaptive schemes are proposed in Section 3. Simulation results are shown in Section 4. Conclusions are drawn in Section 5.

2 Preliminaries

2.1 Broadcasting in a MANET

In this paper, we consider a MANET consisting of a set of cooperating mobile hosts. Each mobile host is equipped with a CSMA/CA (*carrier sense multiple access with collision avoidance*) transceiver which can access the air medium following an IEEE 802.11-like protocol [12]. We consider how to perform broadcasting on top of an IEEE 802.11-like protocol.

A broadcast request can be issued by any source host which has a packet to be distributed to the whole network. This paper focuses on how a broadcast packet is propagated in the network. All other hosts have responsibility to help in propagating the packet by rebroadcasting it. An attempt should be made to successfully distribute the packet to as many hosts as possible without paying too much effort. The broadcast problem considered here is assumed to have the following characteristics:

- **The broadcasting is spontaneous.** Any mobile host can issue a broadcast operation at any time. No global knowledge of the network topology can be collected prior to the broadcast and no global synchronization mechanism can be assumed. Little or no local connectivity information may be collected in advance.
- **The broadcasting is unreliable.** A broadcast message is transmitted via a CSMA/CA manner. No acknowledging mechanism will be used. Note that, in IEEE 802.11 [12], the MAC specification does not allow acknowledging on receiving a broadcast transmission. This is reasonable because, if all receiving hosts send acknowledgments to the sending host, these acknowledgments are very likely to collide with each other at the sender's side, making another "many-to-one" broadcast storm. After receiving a broadcast packet, a host may rebroadcast the message at most once.

In addition, we assume that a host can detect duplicate broadcast packets. This is essential to prevent endless flooding of the packet. One way to do so is to associate with each broadcast packet a tuple (source ID, sequence number) as in [2], [18].

The motivations to solve unreliable broadcast are 1) a host may miss a broadcast message because it is offline, it is temporarily isolated from the network, or it experiences successive collisions, 2) acknowledgments may cause serious medium contention (and, thus, another "storm") surrounding the sender, and 3) in many applications (e.g., the route discovery in [2], [8], [9], [10], [18]), a 100 percent reliable broadcast is unnecessary. However, we remark that reliable broadcasting has also been studied [1], [16], [17], whose goal is to ensure all hosts receive a broadcast message. A lot of high-level acknowledgments between hosts are exchanged. Such protocols are typically accomplished at the application layer and are out of the scope of this paper (however, the result in this paper may serve as an underlying facility to implement reliable broadcast).

Finally, we comment that we do not confine ourselves to broadcasting the *same* packet.[1] What we focus on in this paper is the packet propagation behavior caused by broadcasting in a MANET—the phenomenon where the transmission of a packet will trigger other surrounding hosts to transmit the same (or modified) packet. We shall show that, if flooding is used blindly, many redundant packets will be sent and serious contention/collision will be incurred. Our goal is to solve broadcast with efficiency in mind.

2.2 The Broadcast Storm Problem

A straight-forward approach to perform broadcast is by *flooding*. A host, on receiving a broadcast packet for the first time, has the obligation to rebroadcast the packet. Clearly, this costs n transmissions in a MANET of n hosts. In a CSMA/CA network, drawbacks of flooding include:

- **Redundancy:** When a mobile host decides to rebroadcast a broadcast packet to its neighbors, all of its neighbors might already have heard the packet.
- **Contention:** After a mobile host broadcasts a packet, if many of its neighbors decide to rebroadcast the packet, these transmissions (which are all from nearby hosts) may severely contend with each other.
- **Collision:** Because of the deficiency of backoff mechanism, the lack of RTS/CTS dialogue, and the absence of collision detection, collisions are more likely to occur and cause more damage (RTS = request to send; CTS = consent to send).

1. For instance, the routing protocols in [2], [8], [10], [18] rely on broadcasting a UDP packet called *route_request* to search for a route from a source to a particular destination. When propagating such a request, a host generally appends its ID to the request so that appropriate routing information can be collected.

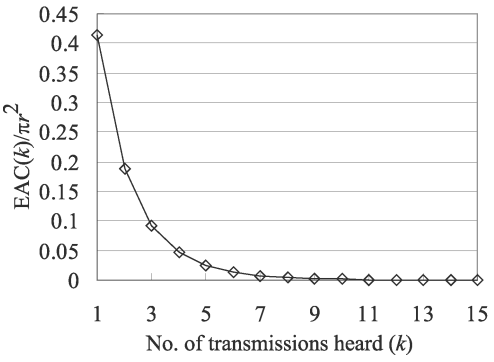


Fig. 1. The expected additional coverage $\mathrm{EAC}(k)$ (divided by πr^2) after a host heard a broadcast packet k times.

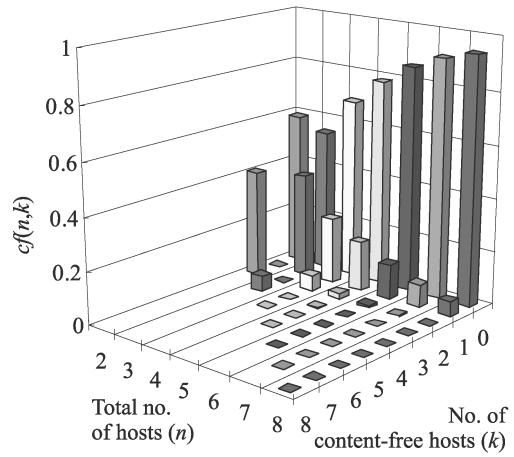


Fig. 2. Analysis on contention: the probabilities of having k contention-free hosts among n receiving hosts.

Collectively, we refer to the above phenomena as the *broadcast storm problem*. In the following, we review some conclusions in [15] in more details.

2.2.1 Redundancy

The main reason for redundancy is that radio signals from different transceivers may overlap with each other seriously. Assume that the area that can be covered by an transceiver forms a circle with a radius r. Let $\mathrm{INTC}(d)$ be the intersection area of two circles of radius r whose centers are distanced by d,

$$\mathrm{INTC}(d) = 4\int_{d/2}^{r} \sqrt{r^2 - x^2}\, dx.$$

On hearing a packet for the first time, the *additional coverage* provided by a host to rebroadcast the packet is $\pi r^2 - \mathrm{INTC}(d)$.

When $d = r$, the additional coverage is the largest, which is about $0.61\pi r^2$. That is to say, a rebroadcast can provide at most 61 percent additional coverage over that already covered by the previous transmission, depending on the value of d. Also, let a rebroadcasting host randomly locate within a transmitter's coverage. We can determine the average additional coverage by integrating $\pi r^2 - \mathrm{INTC}(x)$ over the circle of radius x for x in $[0, r]$:

$$\int_0^r \frac{2\pi x \cdot [\pi r^2 - \mathrm{INTC}(x)]}{\pi r^2}\, dx \approx 0.41\pi r^2.$$

When a host has heard the same broadcast packet from more than one neighbor in its surrounding, the additional coverage, if it does decide to rebroadcast the packet, will become even smaller. Let $\mathrm{EAC}(k)$ denote the *expected additional coverage* provided by a host's rebroadcast after the host has heard the same broadcast packet k times. This can be obtained from simulation by randomly generating k hosts in a host transmission range and calculating the area covered by the latter excluding those already covered by the former k hosts. Fig. 1 shows our simulation result. As can be seen, when $k \geq 4$, the expected additional coverage is below 5 percent.

2.2.2 Contention

Consider that host **A** transmits a broadcast packet and there are n hosts hearing this packet. If all these hosts try to rebroadcast the packet, contention may occur because hosts around **A** are likely to be close and, thus, contend with each other on the wireless medium.

Let's analyze the simpler case of $n = 2$. Let hosts **B** and **C** be the two receiving hosts. Let **B** randomly locate at **A**'s transmission range. In order for **C** to contend with **B**, it must also locate in the area $S_{\mathrm{A}\cap\mathrm{B}}$. So, the probability of having a host **C** contending with **B** is $|S_{\mathrm{A}\cap\mathrm{B}}|/\pi r^2$. Let x be the distance between **A** and **B**. Integrating the above probability over the circle of radius x centered at **A** for x in $[0, r]$, the expected probability of contention is

$$\int_0^r \frac{2\pi x \cdot \mathrm{INTC}(x)/(\pi r^2)}{\pi r^2}\, dx \approx 59\%.$$

Clearly, the contention is expected to be higher as n increases. In [15], a simulation is derived by randomly generating n hosts in **A**'s transmission range. We observe the probability $cf(n, k)$ that k hosts among these n hosts experience no contention in their rebroadcasting (cf stands for *contention-free*). The results are shown in Fig. 2. We can see that the probability of all n hosts experiencing contention (i.e., $cf(n, 0)$) increases quickly over 0.8 as $n \geq 6$. So, the more crowded the area is, the more serious the contention is. On the other hand, the probability of having one contention-free host (i.e., $cf(n, 1)$) drops sharply as n increases. Further, it is very unlikely to have more contention-free hosts (i.e., $cf(n, k)$ with $k \geq 2$). Note that having $k = n - 1$ contention-free hosts implies having n such hosts, so $cf(n, n-1) = 0$.

2.2.3 Collision

In a MANET, there is no base station or access point, therefore we consider only the behavior of the *distributed coordinate function* (DCF) in IEEE Std 802.11. The CSMA/CA mechanism requires a host to start a *backoff* procedure right after the host transmitted a packet or when the host wants to transmit but the medium is busy and the previous backoff has been finished. To perform a backoff, a counter is first set to an integer randomly picked from the host's current backoff window. If the *channel clear assessment* (CCA) mechanism of the transceiver detects no channel

activity during the past *slot* (a short fixed period), the counter is decreased by one. The backoff procedure is finished when the counter reaches zero.

Now, consider the scenario where several neighbor hosts hear a broadcast from host **X**. There are several reasons for collisions to occur. First, if the surrounding medium of **X** has been quiet for enough long, all **X**'s neighbors may have passed their backoff procedures. Thus, after hearing the broadcast packet (and having passed a DIFS period), all hosts may start rebroadcasting at around the same time. This is especially true if carriers cannot be sensed immediately due to things such as RF delays and transmission latency. Second, because the RTS/CTS forewarning dialogue is not used in a broadcast transmission, the collision caused by the *hidden terminal problem* will be more serious. Once collision occurs, without *collision detection* (CD), a host will keep transmitting the packet even if some of its foregoing bits have been garbled, which leads to further waste of bandwidth.

2.3 Review of Some Efficient Broadcasting Schemes

To alleviate the broadcast storm problem, reference [15] suggests two directions: to inhibit redundant rebroadcasts and to differentiate the timing of rebroadcasts. Following these directions, a series of threshold-based broadcasting schemes were proposed. Below, we review two representative schemes: *counter-based* scheme and *location-based* schemes.

2.3.1 Counter-Based Scheme

From the analysis of redundancy in Section 2.2.1, we see that the more times a host has heard the same broadcast packet, the less additional coverage the host will provide if it rebroadcasts the packet. This can be seen from the descending trend of $\mathrm{EAC}(k)$ as k increases. In the counter-based scheme, a counter c that records the number of times a host has received the same broadcast packet is maintained by each host for each broadcast packet. When c reaches a predefined threshold C, we inhibit the host from rebroadcasting this packet because the benefit (the additional coverage) could be low. It was shown in [15] that a threshold C of 3 or 4 can save many rebroadcasts in a dense network while achieving a reachability better or comparable to that of flooding. A larger threshold of $C > 6$ will provide less saving in a sparse network but behave almost like flooding.

2.3.2 Location-Based Scheme

In the location-based scheme, it is assumed that each host is equipped with a positioning device such as GPS. Thus, a receiver can accurately calculate the additional coverage that it can offer from the location(s) of the source(s) from which it heard the broadcast packet. A predefined threshold A is used to determine whether the receiving host should rebroadcast or not. Since more accurate information is used, the location-based scheme can achieve better performance in terms of both reachability and the amount of saving than the counter-based scheme.

3 Adaptive Broadcasting Schemes

As reviewed earlier, reducing the number of redundant rebroadcasts is our primary means of alleviating the broadcast storm. However, one problem with the schemes in [15] is that the threshold used is a predefined fixed value. We call such schemes *fixed-threshold* solutions. This in fact poses a dilemma between reachability and the amount of saving on rebroadcasts as the host distribution of the MANET changes. It is desirable if a host can dynamically adjust its threshold value on-the-fly. In this section, we propose three adaptive schemes to resolve such a dilemma. Consistently in each scheme, a host will decide whether to rebroadcast a broadcast packet or not based on its local neighborhood information.

3.1 Adaptive Counter-Based Scheme

The counter-based scheme in [15] uses a fixed threshold C to inhibit redundant rebroadcasts. If a host already heard the same broadcast packet more than C times, the host will not rebroadcast the packet because it is unlikely that the rebroadcast will provide anything new to its neighborhood. According to [15], the counter-based scheme does provide significant saving when a small threshold (such as $C = 2$) is used. Unfortunately, the reachability will degrade sharply in a sparse network. Increasing the value of C will improve the reachability, but, once again, the amount of saving will be sacrificed.

To resolve the dilemma between reachability and saving, we propose an *adaptive counter-based scheme* in which each individual host can dynamically adjust its threshold C based on its neighborhood status. Specifically, we will extend the fixed threshold C into a function $C(n)$, where n is the number of neighbors of the host under consideration. Thus, each host will use a threshold $C(n)$ depending on its current value of n to determine whether to rebroadcast or not. Each host now executes the following steps:

- S1. On hearing a broadcast packet P for the first time, the host initializes a local counter $c = 1$.
- S2. Wait for a random number ($0 \sim 31$) of slots. Then, submit P for transmission and wait until the transmission actually starts. However, during the waiting, if P is heard again, interrupt the waiting and go to S4.
- S3. Packet P is on the air. The procedure exits.
- S4. Increase c by one. If $c < C(n)$, resume the interrupted waiting in S2. Otherwise, proceed to S5.
- S5. Cancel the transmission of P. The host is inhibited from rebroadcasting P in the future. Then, the procedure exits.

The function $C(n)$ is undefined yet. To give some intuition, we make two observations below.

Observation 1. When a host has very few neighbors, rebroadcasting a broadcast packet is relatively more important for two reasons. First, the redundancy of its rebroadcast is lower because the host is responsible for covering a larger area. This also costs less because there will be less contention. Second, the host is more likely to be located in a critical position (e.g., an *articulation* point, the removal of which will disconnect the network).

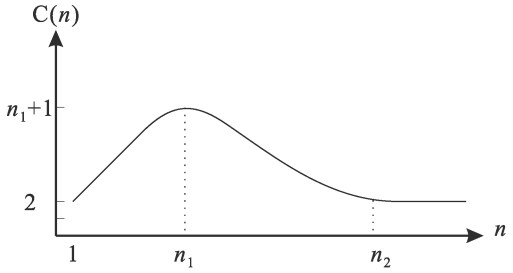


Fig. 3. Abstract shapes of the threshold function $C(n)$ for the adaptive counter-based scheme.

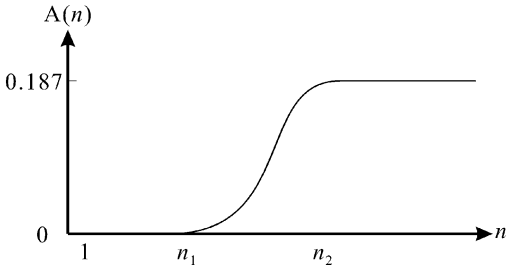


Fig. 4. The abstract shape of the threshold function $A(n)$ for the adaptive location-based scheme.

Inhibiting its rebroadcast may cause a large part of the network to not receive the broadcast packet.

Observation 2. If the neighborhood of a host is crowded enough, we can concentrate more on saving because using a loose threshold will not sacrifice the reachability, but may save many rebroadcasts (and thus alleviate the contention and collision problems).

Intuitively, when n is small, we should use a higher counter threshold because we expect a host to rebroadcast according to **Observation 1**. On the other hand, **Observation 2** suggests that saving is getting more and more important when n increases. The curve in Fig. 3 suggests an abstract shape of $C(n)$ based on the above observations. Before the point n_1, it represents the range where we expect a host to rebroadcast. But, a host with n neighbors is unlikely to hear the same broadcast packet more than n times. So, it is sufficient to let $C(n) = n + 1$ (a $C(n)$ too large would be too strong). Then, we gradually reduce the threshold $C(n)$ after point n_1. After $n \geq n_2$, it is unreasonable to completely prohibit rebroadcasting, so we use the lowest possible threshold $C(n) = 2$. In Section 4.1, we will derive the exact shape of function $C(n)$ through experiments.

Finally, we comment that there should be a neighbor discovery mechanism running at each host to estimate its current n. This can be simply achieved by having each host send a HELLO packet periodically. Such information may be readily available from other protocols (e.g., the routing protocols in [6], [10], [18] all send HELLO periodically).

3.2 Adaptive Location-Based Scheme

The location-based scheme is shown to outperform the counter-based scheme in [15]. However, using a fixed threshold, the scheme still has a dilemma between reachability and saving, especially in a sparse network. In this section, we further extend the location-based scheme to an adaptive one. Specifically, we will extend the fixed threshold A to a function $A(n)$, where n is the number of neighbors of the host under consideration. A host will choose its threshold $A(n)$ based on its current value of n to determine whether to rebroadcast or not. The detailed scheme is spelled out below.

S1. On hearing a broadcast packet P for the first time, the host initializes ac to be the additional coverage provided by its rebroadcast based on the sender's location. If $ac < A(n)$, proceed to S5.

S2. Wait for a random number $(0 \sim 31)$ of slots. Then, submit P for transmission and wait until the transmission actually starts. However, during the waiting, if P is heard again, interrupt the waiting and go to S4.

S3. Packet P is on the air. The procedure exits.

S4. Update ac. If $ac < A(n)$, go to S5. Otherwise, resume the interrupted waiting in S2.

S5. Cancel the transmission of P. The host is inhibited from rebroadcasting P in the future. Then, the procedure exits.

Following **Observations 1** and **2**, Fig. 4 draws an abstract shape of $A(n)$. With few neighbors ($n \leq n_1$), we should use a threshold $A(n) = 0$ to force a host to rebroadcast. Between n_1 and n_2, the threshold should gradually increase to balance saving and reachability. After $n \geq n_2$, a threshold $A(n) = \text{EAC}(2)/\pi r^2 = 0.187$ is used. Intuitively, this is the expected additional coverage after a host received the same broadcast packet twice (recall the counter-based scheme). In Section 4.2, we will derive the exact shape of function $A(n)$ through experiments.

3.3 Neighbor-Coverage Scheme

Although the location-based/adaptive location-based schemes can perform quite well, they are based on a stronger assumption that each mobile host is equipped with a positioning device. In this section, we propose a *neighbor-coverage* scheme that does not count on positioning devices, but on more accurate neighborhood information.

For each host x in the MANET, the following sets are maintained:

- N_x: the set neighbors of x.
- $N_{x,h}$: the set of neighbors of h known by host x, where $h \in N_x$.

The first set can be obtained by having each host broadcasting a HELLO packet periodically. The second set can be obtained by having host h append its set N_h to its HELLO packet. Note that these sets may not be completely accurate. depending on the host's mobility. We also remark that such information is readily available in some routing protocols, such as CBRP (Cluster Based Routing Protocol [10]) or ZRP (Zone Routing Protocol [8]).

The basic idea of this scheme is as follows: Host x will be allowed to rebroadcast a broadcast packet only if it believes that there exists at least one neighbor $h \in N_x$ who may not have received the packet yet. This is achieved by keeping track of a set T containing the pending hosts in x's

neighborhood who have not received the broadcast packet. Whenever a broadcast packet from a host, say h, is received, the set $N_{x,h}$ is subtracted from T. Once T becomes empty, host x's rebroadcast will be inhibited. More formally, each host x runs the following steps:

S1. On hearing a broadcast packet P for the first time, the host initializes a local set $T = N_x - N_{x,h} - \{h\}$, where h is the host from which the packet was received. If $T = \emptyset$, proceed to S5.
S2. Wait for a random number of slots. Then, submit P for transmission and wait until the transmission actually starts. However, during the waiting, if P is heard again, interrupt the waiting and go to S4.
S3. Packet P is on the air. The procedure exits.
S4. Let h be the host from which the same P is heard again. Update $T = T - N_{x,h} - \{h\}$. If $T = \emptyset$, go to S5. Otherwise, resume the interrupted waiting in S2.
S5. Cancel the transmission of P. The host is inhibited from rebroadcasting P in the future. Then, the procedure exits.

4 Performance Simulations

To test our new schemes, we developed a simulator using C++. Central to the simulator is a discrete event-driven engine designed to simulate systems that can be modeled by processes communicating through signals. The MAC specification in IEEE 802.11 Standard is followed to simulate the CSMA/CA behavior among hosts.

The fixed parameters in our simulations are the transmission radius of an transceiver (500 meters), the broadcast packet size (280 bytes), the transmission rate (1M bits per second), and the DSSS physical layer timing (backoff window size = $31 \sim 1,023$ slots, slot time = 20 μsec, SIFS = 10 μsec, DIFS = 50 μsec, PLCP preamble = 144 μsec, and header length = 48 μsec, as suggested in IEEE 802.11).

In each simulation, 100 mobile hosts in a geometric area called a *map* are simulated. Ten thousand broadcast requests are simulated in each simulation where the interarrival time between broadcasts is a uniform distribution between 0 to 2.0 seconds to the whole map. The broadcasting host is randomly picked for each request. To simulate sparse and dense host distributions, a map can be of size 1×1, 3×3, 5×5, 7×7, 9×9, or 11×11, where a unit is of length 500 meters (the transmission radius). Each host will roam around randomly in the map during the simulation. The roaming pattern of each host consists of a series of *turns*. In each turn, the direction, speed, and time interval are randomly generated. The direction is uniformly distributed from 0° to 360°, the time interval from 1 to 100 seconds, and the speed from 0 to a given maximum speed. Unless otherwise specified, the maximum speed is 10 km/hour in the 1×1 map, 30 km/hour 3×3 map, 50 km/hour 5×5 map, etc. Intuitively, this is to make a host move through a wider range in a larger map.

The performance metrics to be observed are:

- *REachability* (RE): r/e, where r is the number of hosts receiving the broadcast packet and e is the number of mobile hosts that are reachable, directly or indirectly, from the source host at the moment when the broadcast is taken.[2]
- *Saved ReBroadcast* (SRB): $(r - t)/r$, where r is the number of hosts receiving the broadcast packet and t is the number of hosts actually rebroadcasting the packet.
- *Average latency*: the interval from the time the broadcast is being initiated to the time the last host is finishing its rebroadcasting or deciding not to rebroadcast.

In this paper, we take RE as the primary goal. In the following, we first discuss each scheme separately. At last, an overall comparison will be given.

4.1 Performance of the Adaptive Counter-Based Scheme

In the following, we first present how we determine the best threshold function $C(n)$. Then, we compare our adaptive counter-based scheme to the fixed-threshold counter-based scheme.

Fig. 3 already gives an abstract shape of $C(n)$. Let's denote the thresholds by a sequence of integers, $x_1x_2x_3\ldots$, i.e., $C(1) = x_1, C(2) = x_2, C(3) = x_3$, etc. The way we determine the best $C(n)$ is in fact through extensive simulations and refinements. The steps are outlined below.

1. **Determine the slope before $n \le n_1$.** As discussed earlier, before n_1 we should enforce a host to rebroadcast. We compare three sequences: $C(n) = 22233344455555\ldots$ (slope = 1/3), $C(n) = 22334455555\ldots$ (slope = 1/2), and $C(n) = 23455555\ldots$ (slope = 1). Fig. 5a shows that the reachability of $C(n) = 234555\ldots$ is the best in sparser maps (7×7, 9×9, and 11×11). This justifies our **Observation 1** that enforcing a host to rebroadcast is essential to improve reachability in sparser networks. The next question is: To what extent should we enforce rebroadcasting?
2. **Determine the value of n_1.** We use the function $C(n) = n + 1$ when $n \le n_1$, and $C(n) = n_1 + 1$ when $n > n_1$ by varying the value of n_1. In Fig. 5b, four functions are tested: $C(n) = 233\ldots$, $C(n) = 2344\ldots$, $C(n) = 23455\ldots$, and $C(n) = 234566\ldots$. The results indicate that $n_1 = 4$ and 5 both give satisfactory reachability. By further taking SRB into consideration, using $n_1 = 4$ will give better saving.
3. **Determine the value of n_2.** As discussed earlier, after n_2 we should use the smallest possible threshold of 2. To find the best n_2, we fix n_1 at 4, vary n_2 (= 8, 12, or 16), and let $C(n)$ linearly decrease between n_1 and n_2. From the results in Fig. 5c, we see that setting $n_2 = 12$ gives better reachability and saving. In particular, $n_2 = 12$ gives the best reachability at sparse networks.
4. **Tune the thresholds between n_1 and n_2.** We already know that the threshold should gradually decrease from n_1 to n_2. As a final step, we try different decreasing functions for this range, as shown in

2. This is to take network partitioning into account.

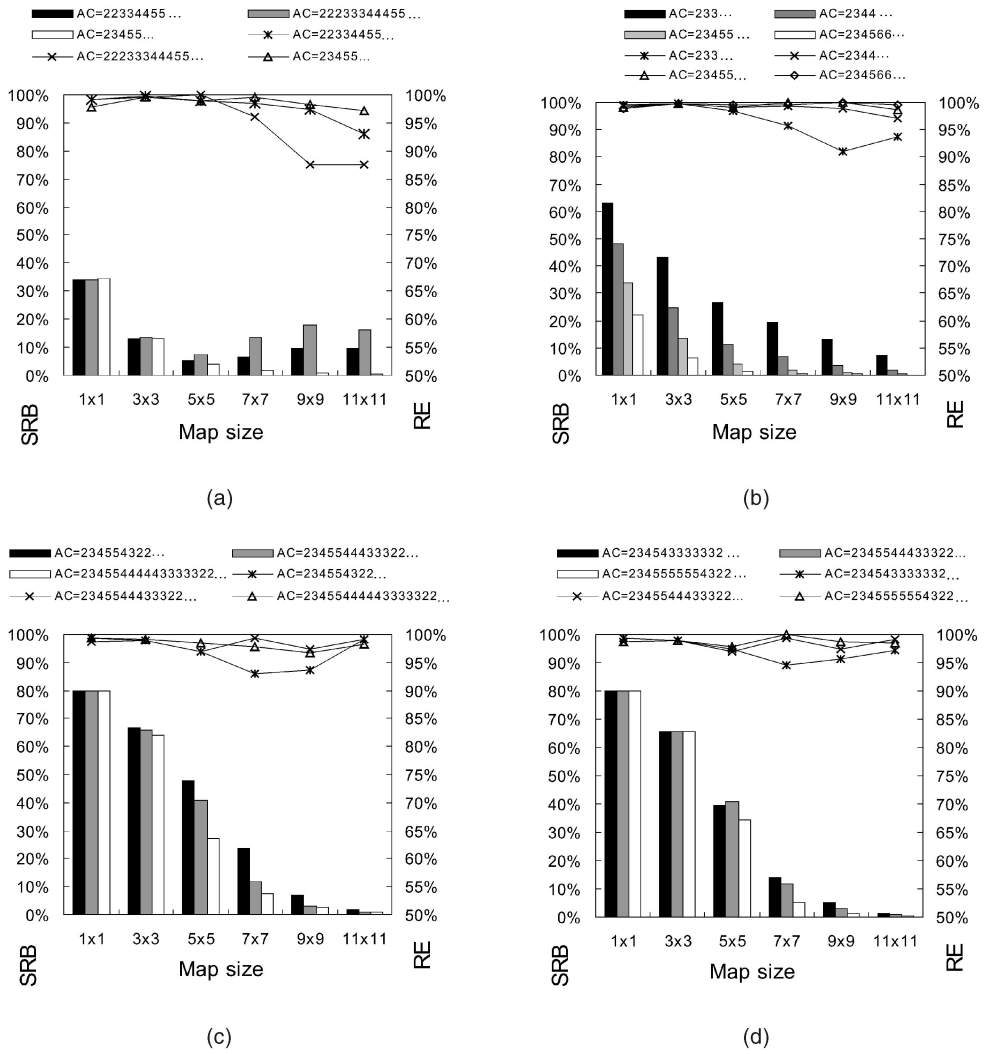


Fig. 5. Tuning the threshold function $C(n)$ for the adaptive counter-based scheme: (a) determining the slope before n_1, (b) determining the value of n_1, (c) determining the value of n_2, and (d) tuning the thresholds between n_1 and n_2. RE (reachability) is shown in lines and SRB (saved rebroadcast) in bars. The thresholds used are represented by a sequence $x_1x_2x_3\ldots$, i.e., $C(1) = x_1, C(2) = x_2, C(3) = x_3$, etc.

Fig. 6, by letting $n_1 = 4$ and $n_2 = 12$. The results are shown in Fig. 5d.

From the above steps, we suggest that the solid line in Fig. 6 should be used by the adaptive counter-based scheme by taking both RE and SRB into consideration.

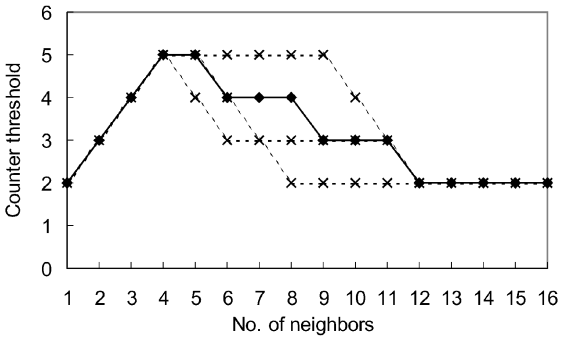
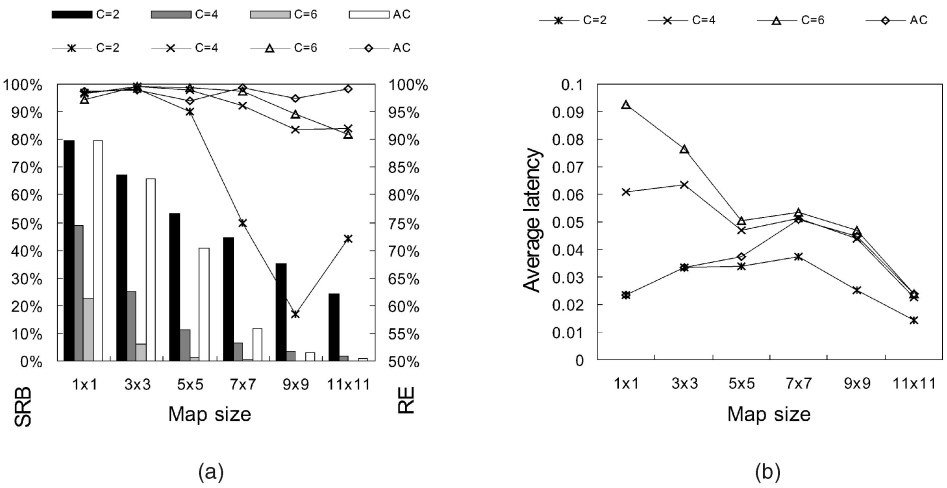


Fig. 6. The functions used to tune $C(n)$ between n_1 and n_2 for the adaptive counter-based scheme.

Finally, we compare the fixed-threshold counter-based scheme ($C = 2, 4, 6$) to our adaptive counter-based scheme using the above suggested threshold function (denoted by AC). Fig. 7a shows the obtained RE and SRB. As can be seen, the fixed-threshold scheme does have the dilemma between RE and SRB. Using a small threshold (such as 2) can give both satisfactory RE and SRB in denser maps (1×1, 3×3, 5×5), but RE degrades sharply if the host distribution is sparser. On the other hand, using a larger threshold (such as 6) indeed raises RE, but SRB degrades in all maps. The proposed scheme can resolve the dilemma effectively. RE is always maintained at high level and, except in very sparse networks, the SRB is significant.

Fig. 7b compares the broadcast latency. The latency of AC is the smallest in maps 1×1 and 3×3. In other maps, the latency of AC is slightly larger than that of $C = 2$ because AC is targeting at higher RE in sparser maps. These results justify the effectiveness of our adaptive counter-based scheme over the fixed counter-based scheme.



(a) (b)

Fig. 7. Comparison of the adaptive counter-based scheme (AC) and fixed-threshold counter-based scheme ($C = 2, 4, 6$). (a) RE (shown in lines) and SRB (shown in bars) and (b) average broadcast latency (in seconds).

4.2 Performance of the Adaptive Location-Based Scheme

In the following, we first show how we determine the best threshold function $A(n)$ for our adaptive location-based scheme. Then, we compare it to the fixed-threshold location-based scheme.

Fig. 4 already indicates an abstract curve for $A(n)$. Before n_1, we should enforce a host to rebroadcast by setting $A(n) = 0$ and, after n_2, we tend to inhibit a host from rebroadcasting by setting $A(n) = 0.187$ (this is the average additional coverage after a host hears the same broadcast packet twice). In our experiments, we will fix n_1 and n_2 and simply make $A(n)$ a linear function between n_1 and n_2. Fig. 8 shows the curves for $A(n)$ used in our experiments.

Each curve in Fig. 8 can be denoted by (n_1, n_2), depending on the values of n_1 and n_2 chosen for use. We then conducted extensive simulations on different maps to compare these threshold functions. The result is in Fig. 9 (for clarity, we partition the results into four subfigures). It indicates that $(6, 12)$, $(8, 12)$, and $(8, 10)$ all deliver quite satisfactory RE. By further considering their SRB, we would suggest picking $(6, 12)$ (observe the SRB of $(8, 12)$, and $(8, 10)$ on sparser maps).

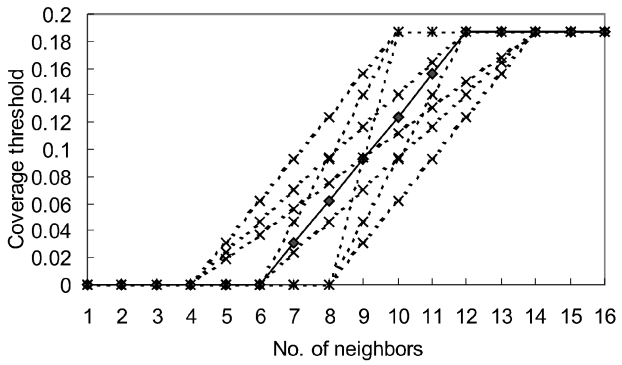


Fig. 8. The functions used to tune $A(n)$ for the adaptive location-based scheme.

Based on the above result, we then compare the fixed-threshold location-based scheme ($A = 0.1871$, $A = 0.0469$, and $A = 0.0134$, which are used in [15]) to our adaptive location-based scheme (using $n_1 = 6$ and $n_2 = 12$). As shown in Fig. 10a, the RE of the fixed-threshold scheme will degrade significantly in sparser maps. The problem can be conquered by our adaptive scheme. Moreover, the SRB will not be sacrificed as high RE is achieved. Fig. 10b compares the broadcast latency. On denser maps, the adaptive scheme has the lowest latency and, on sparser maps, the latency is slightly higher than that of $A = 0.1871$ so as to maintain high RE.

4.3 Performance of the Neighbor Coverage Scheme and Dynamic Hello Interval

The neighbor coverage scheme can adaptively assess the redundancy of a host's rebroadcast based on its neighborhood information and the sources from which it received the same broadcast packet. However, as shown below, the accuracy of the neighborhood information may have significant effect on this scheme. To collect neighborhood information, HELLO packets are sent periodically by each host. A host x enlists another host h as its one-hop neighbor when a HELLO is received from h. If no HELLO has been received from h for the past two hello intervals, host x deletes h as its one-hop neighbor. Below, we first discuss the effect of the hello interval and then propose a *dynamic hello interval* scheme on top of our neighbor coverage scheme.

Obviously, a shorter hello interval will make neighborhood information more up-to-date. To study this effect, we compare the RE achieved by different hello intervals (1,000, 5,000, 10,000, 20,000, and 30,000 milliseconds) under different host mobility (20, 40, 60, and 80 km/hour) at different maps in Fig. 11. As can be seen, on sparser maps (such as Fig. 11c and Fig. 11d), a long hello interval may significantly degrade RE, especially when the host mobility is high. Note that only in smaller maps (such as 5×5 in Fig. 11a) does host mobility have little impact on RE because hosts hardly roam too far away from the source host.

Although a lower hello interval can offer better RE, too many HELLOs may also hurt the network by taking up too

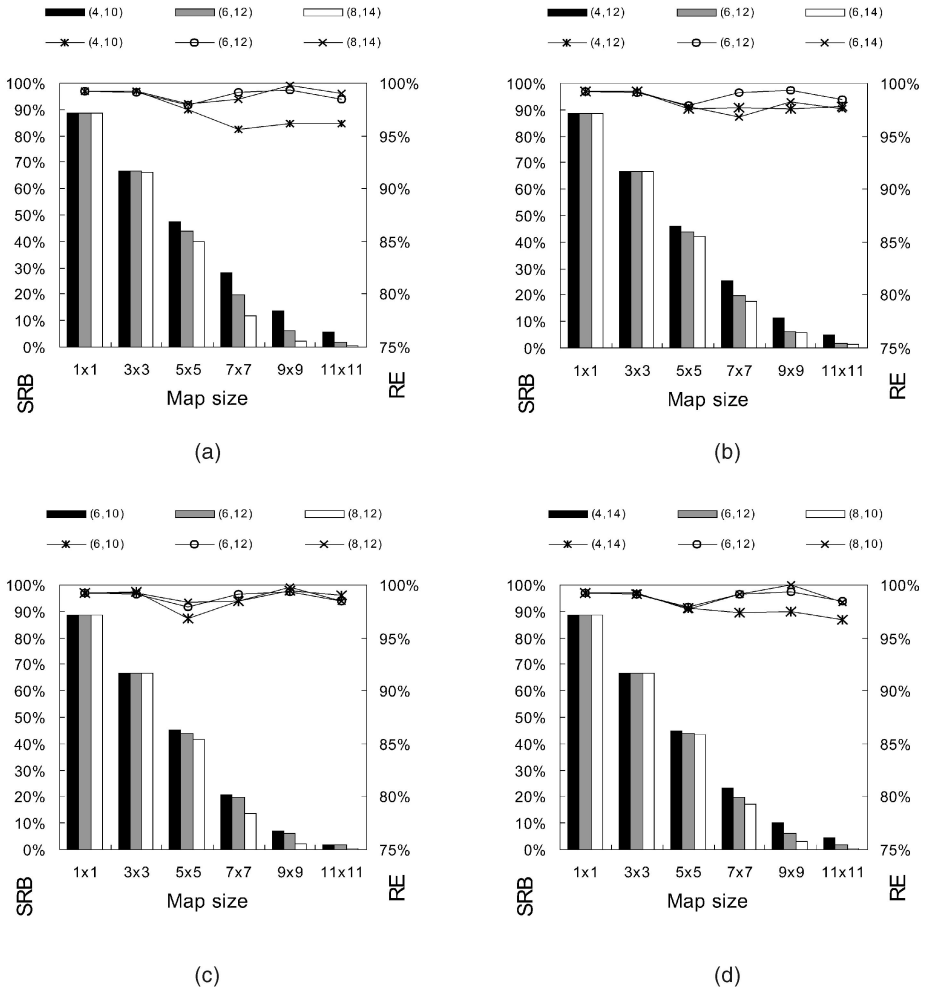


Fig. 9. Comparison of the adaptive location-based scheme using the threshold functions in Fig. 8.

much network bandwidth, especially in a crowded environment. Below, we propose a way to dynamically adjust the hello interval by each host. First, we define a host x's *neighborhood variation* nv_x as

$$nv_x = \frac{\text{the number of hosts joining or leaving the set } N_x \text{ in the past 10 sec.}}{|N_x| \times 10}.$$

Intuitively, nv_x is a quantitative estimation of the change at x's neighborhood learned in the near past. Note that the change may be caused by x's neighbors or be x itself.

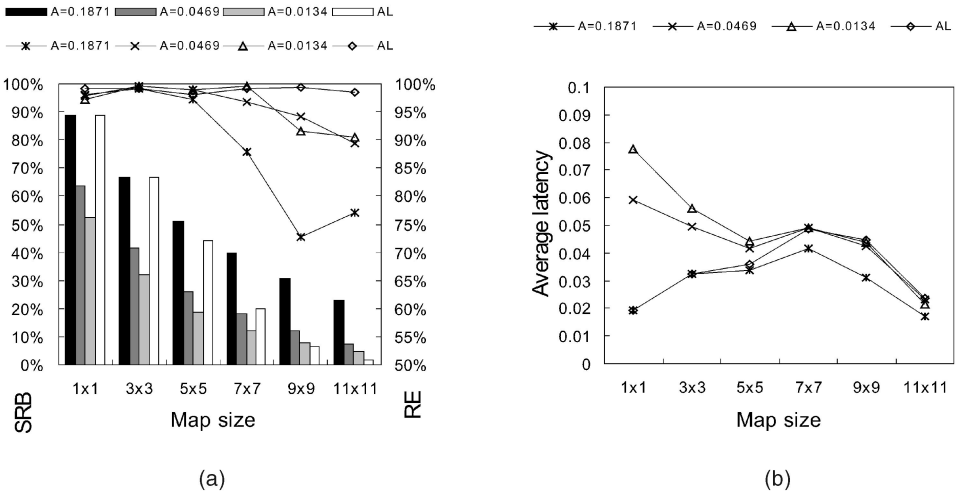


Fig. 10. Comparison of the adaptive location-based scheme (AL) and fixed-threshold location-based scheme ($A = 0.1871, 0.0469, 0.0134$). (a) RE and SRB, and (b) average broadcast latency (in seconds).

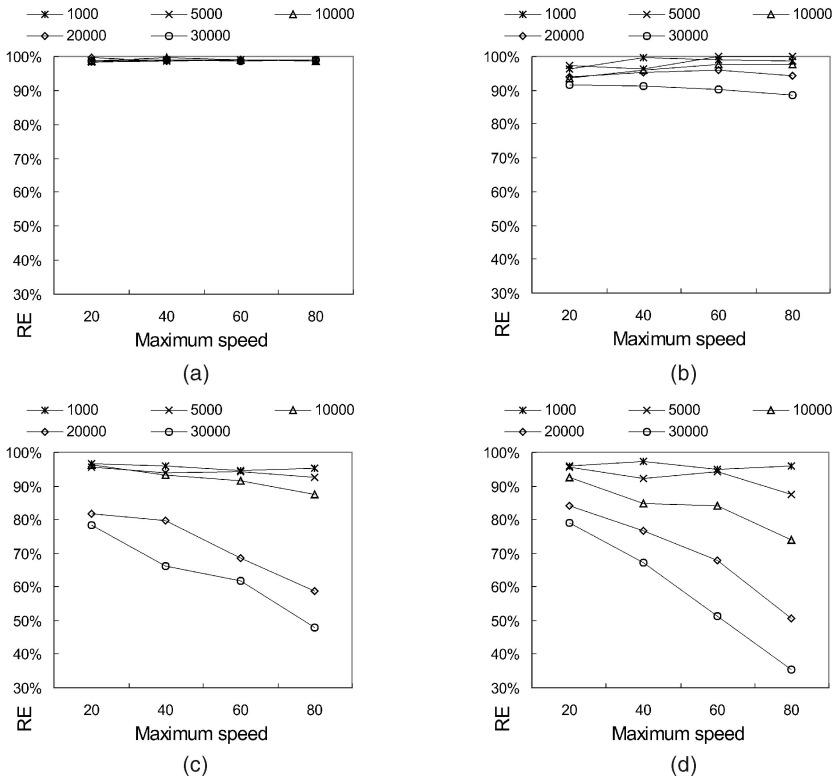


Fig. 11. Comparison of the neighbor coverage scheme under different hello intervals (1,000, 5,000, 10,000, 20,000, and 30,000 milliseconds) at different host speeds. (a) 5×5, (b) 7×7, (c) 9×9, and (d) 11×11 maps.

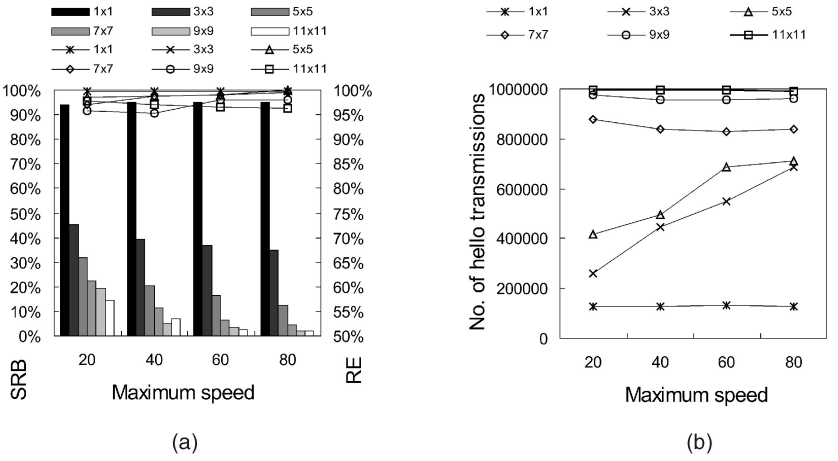


Fig. 12. Performance of the neighbor coverage scheme with dynamic hello interval at various host speeds. (a) RE and SRB. (b) The number of hello packets sent.

Then, we use the neighborhood variation to adjust the hello interval hi_x of x:

$$hi_x = max\left(hi_{min}, \frac{nv_{max} - nv_x}{nv_{max}} \times hi_{max}\right),$$

where nv_{max} is the predefined maximum neighborhood variation and hi_{min} and hi_{max} the predefined shortest and longest hello intervals, respectively. Intuitively, the hello interval is inverse to the value of nv_x and is guarded by a minimal and a maximal value. We comment that, since each host's hello interval may change dynamically, this value should be appended to its HELLO packets so that surrounding hosts can determine its existence. We also comment that, in AODV [18], a *Hello Interval Extension* is also specified, but without much detail.

To verify the effectiveness of the above scheme, we conducted some simulations using $nv_{max} = 0.02$, $hi_{min} = 1,000\text{ms}$, and $hi_{max} = 10,000\text{ms}$. Fig. 12a shows that using the dynamic hello interval can maintain high RE independent of the host mobility and host density. The SRB is also very significant compared to other schemes. This is a merit that cannot be achieved by either the adaptive counter-based scheme or the adaptive location-based scheme. Fig. 12b evaluates the number of HELLO packets sent on different scenarios. Generally speaking, on sparser

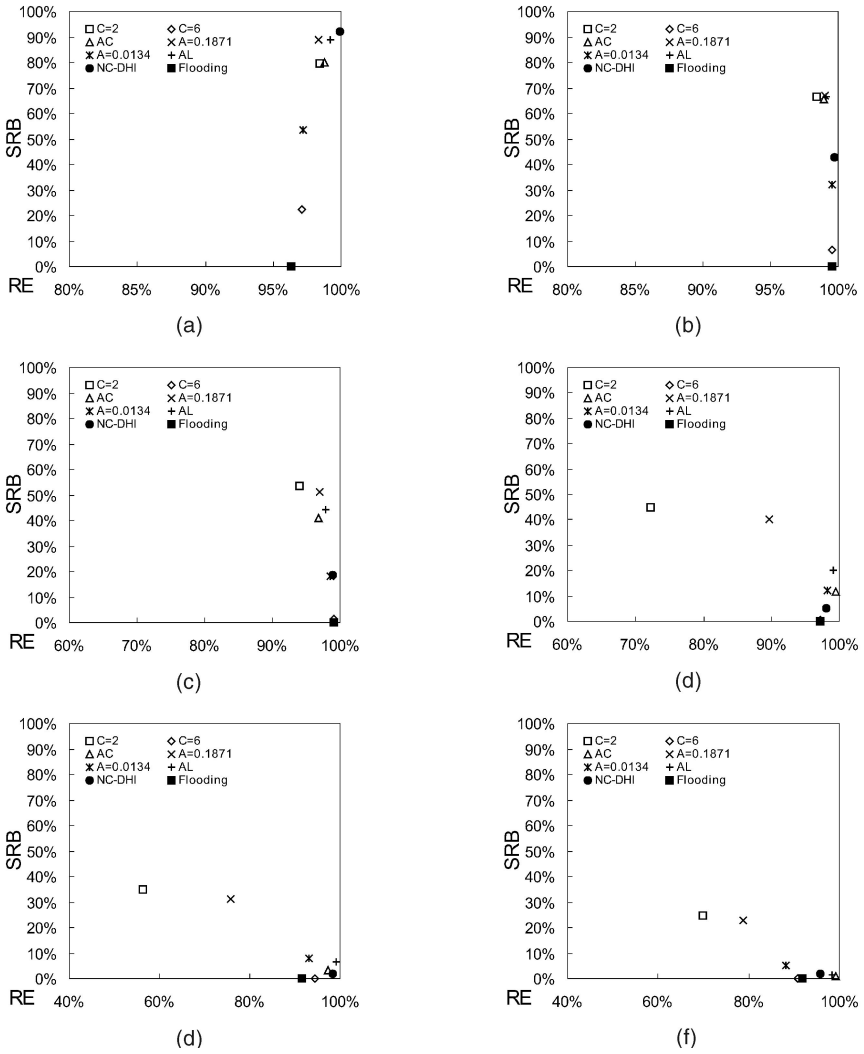


Fig. 13. Overall comparison of the counter-based ($C = 2, 6$), adaptive counter-based (AC), location-based ($A = 0.1871, 0.0134$), adaptive location-based (AL), neighborhood coverage with dynamic hello interval (NC-DHI), and flooding schemes. (a) 1×1, (b) 3×3, (c) 5×5, (d) 7×7, (e) 9×9, and (f) 11×11 maps.

maps (9×9 and 11×11), the neighborhood variation will be higher (around 0.02 in our simulations). So, most hosts will pick the smallest hello interval of $hi_{min} = 1,000\text{ms}$, which is the reason for raising RE. On 3×3 and 5×5 maps, the number of hellos will increase with higher host mobility. On the smallest 1×1 map, there is almost no neighborhood variation, so the hello interval is very close to $hi_{max} = 10,000\text{ms}$.

4.4 Overall Comparison

This section shows an overall comparison on the counter-based (denoted by C), adaptive counter-based (denoted by AC), location-based (denoted by A), adaptive location-based (denoted by AL), neighbor coverage (denoted by NC-DHI), and flooding schemes. For each scheme, we pick the best available threshold or threshold function. The neighbor coverage scheme is coupled with our dynamic hello interval scheme. RE is our primary goal and SRB is our secondary goal. (We presume that RE is always a more important issue than SRB because, by the semantic of "broadcast," one would like to disseminate a message as much as possible.) The results are shown in Fig. 13. Intuitively, points toward the upper-right corner of the figure would be the best choices. The maximum speed of hosts is 10 km/hour in the 1×1 map, 30 km/hour in the 3×3 map, 50 km/hour in the 5×5 map, etc. (Such setting is to take the different map sizes into consideration.)

We observe that the flooding approach only offers satisfactory RE in denser maps. Its SRB is always 0. On the contrary, almost all other schemes provide certain levels of SRB and sometimes better RE than that of flooding. The main reason for a lot of hosts missing the broadcast message is collision. As has been discussed earlier, this is the effect of broadcast storm caused by flooding.

In terms of SRB, on denser networks (such as 1×1 and 3×3 maps), significant saving can be obtained by our schemes. This is because messages can be easily distributed in dense networks. So, hosts should inhibit their rebroadcasts. On the contrary, on sparser networks (such as 9×9 and 11×11 maps), each host is only connected to a very few hosts. If any host does not rebroadcast, a disaster may happen (i.e., a lot of hosts may miss the broadcast message).

So, in this case, hosts should try harder to rebroadcast (and SRB will become a less important issue in this case). Apparently, there are trade offs which justify the need of adaptive approaches, especially when the network density is an unpredictable factor.

Further, in terms of reachability, our schemes outperform flooding in almost all cases. The only exception is when the network size is 5×5, where flooding also provides good reachability. This is perhaps because the network is just the "right size" such that collision is not a problem.

Overall, the neighbor coverage scheme performs the best on denser maps and the adaptive counter-based and adaptive location-based schemes perform the best on sparser maps. Their REs are consistently above 95 percent in all cases. Except on the 5×5 map, where the flooding scheme has good RE, it is always rewarding to use our schemes considering both RE and SRB. To summarize, the adaptive location-based scheme could be the best choice. However, if positioning devices are unavailable, we would suggest using the neighbor coverage scheme, which also performs quite well. The adaptive counter-based scheme has merit on its simplicity and it in fact performs quite comparably to the other two schemes.

5 Conclusions

Broadcasting in a MANET has quite different characteristics from that in other networks. It could cause serious redundancy, contention, and collision. We have proposed three adaptive broadcasting schemes, namely, adaptive counter-based, adaptive location-based, and neighbor-coverage schemes, for the broadcast storm problem in a MANET. The first two adaptive schemes have effectively resolved the dilemma between reachability and efficiency in the fixed-threshold counter-based and location-based schemes that we proposed in [15]. If positioning devices are not available, the last scheme is also a very good choice. Because of their adaptive nature, these schemes can be used independent of the current host density and host mobility. A different direction deserving investigation is to use unicast-based broadcast; one such example is in a recent paper [19]. We have also discussed how to dynamically adjust the intervals to send HELOO messages. We note that many recently proposed routing protocols also rely on periodic HELLO messages for various route discovery and maintenance purposes. The HELLOs used here may be combined with those for routing purposes. The dynamic HELLO scheme can potentially be used by such routing protocols to reduce the corresponding cost.

Acknowledgments

This work is cosponsored by the Lee and MTI Center for Networking Research at the National Chiao Tung University, and the Program of Excellence Research, Ministry of Education, Taiwan (contract numbers 89-E-FA-04-4, 89-E-FA04-1-4, and 89-H-FA07-1-4). Part of this work was done while Y.-C. Tseng was supported by the Institute of Information Science, Academia Sinica, Taiwan, as a visiting scholar. A preliminary version of this paper has appeared in the *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*, 2001.

References

[1] S. Alagar, S. Venkatesan, and J. Cleveland, "Reliable Broadcast in Mobile Wireless Networks," *Military Comm. Conf., MILCOM Conf. Record,* vol. 1, pp. 236-240, 1995.

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

[3] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *Proc. Int'l Conf. Mobile Computing and Networking (MOBICOM),* pp. 85-97, 1998.

[4] C.-C. Chiang, M. Gerla, and L. Zhang, "Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks," *ACM-Baltzer J. Cluster Computing,* vol. 1, no. 2, 1998.

[5] S. Corson and J. Macker, "Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," 1999. Request for Comments: 2501.

[6] R. Dube, C.D. Rais, K.-Y. Wang, and S.K. Tripathi, "Signal Stability Based Adaptive Routing for Ad Hoc Mobile Networks," *IEEE Personal Comm.,* Feb. 1997.

[7] M. Gerla, C.-C. Chiang, and L. Zhang, "Tree Multicast Strategies in Mobile, Multihop Wireless Networks," *ACM/Baltzer Mobile Networks and Applications,* vol. 4, no. 3, pp. 193-207, 1998.

[8] Z.J. Haas and M.R. Pearlman, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," 1998. Internet draft, draft-ietf-manet-zone-zrp-01.txt, work in progess.

[9] P. Jacquet, P. Muhlethaler, and A. Qayyum, "Optimized Link State Routing Protocol," 2000. Internet draft, draft-ietf-manet-olsr-01.txt, work in progess.

[10] M. Jiang, J. Li, and Y.C. Tay, "Cluster Based Routing Protocol (CBRP) Functional Specification," 1998. Internet draft, draft-ietf-manet-cbrp-spec-00.txt, work in progess.

[11] Y.-B. Ko and N.H. Vaidya, "Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms," *Proc. IEEE Workshop Mobile Computing Systems and Applications (WMCSA '99),* Feb. 1999.

[12] LAN MAN Standards Committee of the IEEE CS, *IEEE Std 802.11-1997, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,* Nov. 1997.

[13] W.-H. Liao, Y.-C. Tseng, and J.-P. Sheu, "GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks," *Telecomm. Systems,* vol. 18, no. 1, pp. 37-60, 2001.

[14] J.P. Macker and M.S. Corson, "Mobile Ad Hoc Networking and the IETF," *ACM Mobile Computing and Comm. Rev.,* vol. 2, no. 3, pp. 7-9, July 1998.

[15] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *Proc. Int'l Conf. Mobile Computing and Networking (MOBICOM),* Aug. 1999.

[16] E. Pagani and G.P. Rossi, "Reliable Broadcast in Mobile Multihop Packet Networks," *Proc. Int'l Conf. Mobile Computing and Networking (MOBICOM),* pp. 34-42, 1997.

[17] E. Pagnni and G.P. Rossi, "Providing Reliable and Fault Tolerant Broadcast Delivery in Mobile Ad-Hoc Networks," *ACM/Baltzer Mobile Networks and Applications,* vol. 4, pp. 175-192, 1999.

[18] C.E. Perkins and E.M. Royer, "Ad Hoc on Demand Distance Vector (AODV) Routing," 1999. Internet draft, draft-ietf-manet-aodv-03.txt, work in progess.

[19] P. Sinha, R. Sivakumar, and V. Bharghavan, "Enhancing Ad Hoc Routing with Dynamic Virtual Infrastructure," *Proc. IEEE INFOCOM,* pp. 1763-1772, 2001.

Yu-Chee Tseng received the BS and MS degrees in computer science from National Taiwan University and the National Tsing-Hua University in 1985 and 1987, respectively. He worked for D-LINK Inc. as an engineer in 1990. He obtained the PhD degree in computer and information science from Ohio State University in January of 1994. From 1994 to 1996, he was an associate professor in the Department of Computer Science, Chung-Hua University. He joined the Department of Computer Science and Information Engineering, National Central University in 1996 and became a full professor in 1999. In August 2000, he became a full professor in the Department of Computer Science and Information Engineering, National Chiao-Tung University, Taiwan. Dr. Tseng has served as a program committee member on many international conferences, as a program chair for the Wireless Networks and Mobile Computing Workshop in the years 2000 and 2001, as an associate editor of *The Computer Journal*, and as a guest editor of *ACM Wireless Networks*, *IEEE Transactions on Computers*, *Journal of Internet Technology*, and *Wireless Communications and Mobile Computing*. His research interests include wireless communication, network security, parallel and distributed computing, and computer architecture. He is a senior member of the IEEE.

Sze-Yao Ni received the PhD degree in computer science and information engineering from the National Central University, Taiwan, Republic of China. He is a software architect at DeanSoft Co., Ltd. in Taiwan. His research interests include computer architecture, parallel and distributed computing, and mobile computing. He is a member of the IEEE, the ACM, and the Phi Tau Phi Society.

En-Yu Shih received the BS and MS degrees in computer science and information engineering from the Ta-tung Institute of Technology and the National Central University in 1998 and 2000, respectively. He is now a software engineer at the Industrial Technology Research Institute.