PAPER

A Reliable Broadcast/Multicast Scheme for Multihop Mobile Ad Hoc Networks*

Jenhui CHEN[†a)], ***Member and*** **Shiann-Tsong SHEU**[††], ***Nonmember***

SUMMARY In wired networks, broadcast and multicast transmissions can be easily achieved by data link layer (layer 2). Nevertheless, it is a big challenge to safely transfer broadcast or multicast data frames over multihop mobile ad hoc networks (MANETs) due to the high bit error rate, the high collision probability and the lack of acknowledgment. Additionally, most of MANET's routing protocols rely on the broadcast function to exchange essential routing packets between mobile nodes and need the multicast function to make more efficient use of network bandwidth for some particular multimedia applications. From our observations, the efficiency of the unicast/multicast routing protocol of finding the path/tree is highly dependent on supported broadcast schemes of the underlying media access control (MAC) protocol. Therefore, in this paper, we illustrate the *uncertain broadcast problem* due to no replying acknowledgment from any recipients when mobile nodes deliver broadcast frames in wireless networks. We, then, propose a novel reliable broadcast scheme to solve this problem as well as a reliable multicast scheme to enhance the network utilization in data link layer. Simulation results show that the proposed scheme, which is still compatible with IEEE 802.11 standard, can efficiently minimize the bandwidth consumption as well as propagation delay.

key words: *ad hoc, broadcast, MAC, multicast, protocol, reliability, wireless LAN*

1. Introduction

A mobile ad hoc network (MANET) [12] is constructed by several mobile handsets or laptops and characterized by multihop wireless connectivity, changing network topology frequently and the need for dynamic routing protocols [7], [13], [15], [17] when there is a packet needed to be delivered. There is no stationary infrastructure or a base station to coordinate packet transmissions and to advertise the information of network topology permanently. According to these characteristics, each mobile node in multihop MANETs must play a role of router to relay data packets for its neighboring mobile nodes. Since the data transmission is operated in the same radio channel, any transmission will interfere with sender's neighbors which also have packets to transmit at the same time. In order to route packets to all members in the network, mobile nodes will perform a *intra-team broadcasting* procedure [9] to exchange essential control messages. Thus the existence of reliable and resource-efficient broadcast protocols [11], [18], [20] in multihop MANET is indispensable due to increased amount of circulating control messages.

In conventional networks, there are many kinds of data needed to be transmitted by using flooding method, e.g., address resolution protocol (ARP), routing information, and advertisement messages, etc. Lack of these schemes, nodes might fail to reach other network devices due to insufficient network information. Moreover, many routing protocols use the broadcast approach to perform their routing procedures, e.g., dynamic source routing (DSR) protocol [7], adhoc on-demand distance vector (AODV) routing protocol [13], [14], and enhanced AODV protocol, named as multicast AODV (MAODV) protocol [14], to provide multicast routing in multihop MANETs. All of them are on-demand and based on the concept of source routing. In order to perform the route discovery process, the source node broadcasts a route request (RREQ) packet, which is flooded through the network in a controlled manner and answered by a unicast route reply (RREP) packet from either the destination node or intermediate nodes that have a route to the destination node. Obviously, the performance of DSR, AODV and MAODV protocols are relying on the efficiency of the broadcast scheme in the data link layer.

In IEEE 802.11 medium access control (MAC) protocol [6] (denoted as IEEE 802.11 for short), regardless of the length of broadcast frame, no acknowledgment (ACK) frame would be replied by any recipients of the broadcast/multicast frame. As a result, the source node has no idea about the status of the transmitted broadcast/multicast frame. Taking DSR, AODV and MAODV protocols for example, the request will be blocked if its source node has not received a valid route(s) within the *route discovery timeout*. Unfortunately, once the route discovery timeout is up, it is very hard to tell the timeout is caused by no path exists or resulted from losing the RREQ packet. Consequently, the *reactive* nature of on-demand routing protocols can not gain any benefit from saving bandwidth than traditional *proactive* routing protocols. Hence, it is desired to design an efficient and highly reliable broadcast transmission scheme for multihop MANETs.

Moreover, the problem of designing an *optimal* broadcasting protocol so that bandwidth consumption or time delay is minimized has been proved as NP-hard in [1], [2]. We therefore resort to heuristics, aiming at providing upper bounded performance with respect to these metrics. Since

Manuscript received January 16, 2004.
Manuscript revised October 23, 2004.
[†]The author is with the Department of Computer Science and Information Engineering, Chang Gung University, Taoyuan, Taiwan, R.O.C.
[††]The author is with the Department of Communication Engineering, National Central University, Jhongli, Taoyuan, Taiwan 320, R.O.C.
*This work was supported in part by the National Science Council, Taiwan, R.O.C., under contract NSC93-2213-E-182-022.
a) E-mail: jhchen@mail.cgu.edu.tw
DOI: 10.1093/ietcom/e89–b.3.867

the broadcasting is a subset of the multicasting that uses point-to-multipoint communication scheme, in this paper, we first discuss and solve the uncertain broadcast problem and then we will enhance the proposed scheme to provide reliable multicast transmissions in complicated multi-hop MANETs.

The remainder of this paper is organized as follows. At first, we shortly take an overview of basic operations of the IEEE 802.11 and describe the uncertain broadcast problem in Sect. 2. Section 3 presents the proposed reliable broadcast transmission scheme for MANETs in detail. In Sect. 4, we further promote the proposed reliable broadcast scheme to achieve reliable multicast transmission in data link layer. We investigate several simulation models and results in Sect. 5. Finally, we give some conclusions in Sect. 6.

2. The IEEE 802.11 MAC Protocol

2.1 MAC Operations

The IEEE 802.11 standard includes a basic distributed coordination function (DCF) and an optional point coordination function (PCF). The DCF uses carrier sense multiple access with collision avoidance (CSMA/CA) as a basic channel access mechanism to transmit asynchronous data in the contention period. When a mobile node desiring to transmit frames, it needs to monitor channel activities before its transmission. If the mobile node perceives the channel is idle for a distributed inter-frame space (DIFS) time period, it will trigger a random backoff delay time before transmission (this is the concept of ‘collision avoidance’ in the CSMA/CA protocol). Otherwise, the mobile node persists on monitoring the channel until it detects channel idle for a DIFS duration. The backoff time is measured in *time slots*, which is defined as the time needed for a node to detect a frame, to accumulate the time needs for the propagation delay, to switch from the receiving state to the transmitting state, and to signal to the MAC layer the state of the channel. The slot time is set as $20\,\mu$s in the direct sequence spread spectrum (DSSS) PHY specification [6]. The random backoff procedure can efficiently minimize the collision probability. However, if more than one mobile node selects a same backoff time, their transmissions will collide with each other. In addition, to avoid channel capture, a node must wait a random backoff time between two consecutive frame transmissions even if the medium is sensed idle for a DIFS period after precedent transmission.

The DCF defines an optional handshaking scheme, which uses the RTS/CTS mechanism to overcome the well-known *hidden terminal problem* [19] and to provide virtual carrier sense for saving battery power [5]. The duration field in the MAC header of a control/data frame is used to carry the information of time period requested for a complete transmission. Any listening mobile node receives this information, it will update its network allocation vector (NAV) which contains the information of the interval time that the channel will remain busy. To prevent the handshaking process from being disturbed by other transmissions, the short inter-frame space (SIFS) is taken to guarantee the control frames to have a higher priority than data frames. However, the handshaking mechanism could not be used in broadcast/multicast transmissions since their receivers are multiple. Therefore, no acknowledgment frame will be replied from any recipient and may lead to unreliable consequences.

2.2 The Uncertain Broadcast Problem

In standard, broadcast frames, multicast frames, and RTS frames are sent in the same physical carrier sensing. The key difference between broadcast/multicast frames and unicast frames is the lack of acknowledgments. Since the IEEE 802.11 wireless local area network (WLAN) adapter is designed as half-duplex mode, sender cannot detect collisions or errors on its broadcast/multicast frames. This shortcoming incurs a severe problem for all protocols or applications, which need broadcast control frames to retrieve useful information from networks or transmit the multicast data frames to conserve bandwidth. For examples, as mentioned earlier, DSR, AODV, and MAODV need broadcast RREQs to perform the route discovery procedure. One can imagine that as a RREQ frame travels from a source to various nodes, the frame loss probability (without recovering) is proportional with the number of hops in its journey. Even though some RREQs fortunately survive after passing a number of consecutive contentions, the found available paths by receiving RREP frames from either the destination node or intermediate nodes, which have a valid route to the destination node, may not include the best one.

This means that such routing protocols will work well in wireless networks under the constrain that every node can successfully detect neighbors’ broadcasts without loss as in wired networks. Unfortunately, this discourages us to apply well-known routing protocols for IEEE 802.11 based multi-hop MANETs unless the IEEE 802.11 protocol can provide a reliable broadcast scheme. Similarly, for the multicast service, as soon as the multicast tree is established by route discovery procedure, the reliable multicast transmission becomes another challenge by the same reason. In the following section, we propose a highly reliable broadcast scheme with limited bandwidth consumption to solve the uncertain broadcast problem.

3. Highly Reliable Broadcast Transmission Schemes

Assume the radio link between two neighboring nodes is symmetric. A broadcast sender can receive the same broadcast frames from its neighbor nodes several times after transmitting the broadcast frame as they had successfully received the broadcast frame. Based on this concept, the simple way of a mobile node to recognize whether the broadcast frame has been successfully received by its neighbors can simply accumulate the number of the same broadcast frames, which are broadcasted forward from its neighbors within a specified observing window. In such flooding process, a mobile node only broadcasts forward the broadcast frame at the first time it receives the broadcast and the later arrival identical broadcast frames will be discarded.

If the number of received rebroadcasts is less than the expected number in an observation window, the sender needs retransmit it until the amount is sufficient. However, the problem becomes how to give an appropriate observation window since a shorter observation window will cause excessively redundant retransmission overheads. On the other hand, a longer observation window will increase the retransmission delay and slow down the flooding speed. Owing to mobile nodes with CSMA/CA protocol contending channel in a distributed manner, it is very hard to measure a precise delay of each transmission to help determine the observation window size. Thus we propose two efficient broadcast schemes, which do not need the observation window, to solve the thorny problem.

3.1 Duplicated Broadcast Scheme (DBS)

Upon a node successfully transmitting a broadcast frame, all recipients of the broadcast transmission will forward it as fast as they can. However, it is quite often that neighbor nodes can also hear each others in a small WLAN environment. Thus, expectable severe contentions will make the following forward broadcast transmissions fail potentially. If the broadcast frame is transmitted or forwarded once by each mobile node, the flooding would not cover all members of the network. A simple way to enlarge the flooding area (or can be measured in flooding fraction) is to increase the transmission times per each broadcast frame in every mobile node. If every node transmits a broadcast frame twice, the flooding fraction will become higher than that only transmitting once.

In general, a higher flooding fraction will be obtained if the transmission time is risen. From the network's viewpoint, it is not wise to transmit too many identical broadcast frames in a node since too many redundant transmissions will significantly degrade the network throughput. It is a tradeoff between the flooding fraction (reliability) and the retransmission overhead. Thus, it is worth to design an efficient scheme with minimum broadcast times to achieve an acceptable flooding fraction. In the next subsection, we will introduce an adaptive duplicated broadcast scheme (ADBS) for IEEE 802.11 MAC protocol.

3.2 Adaptive Duplicated Broadcast Scheme (ADBS)

An efficient broadcast scheme should prevent a node from transmitting broadcast frames redundantly. In fact, retransmission is necessary only when any neighbor node does not receive the broadcast frame. To achieve this goal, there are two important information elements that must be obtained by broadcast sender: the number of active neighbors and the number of neighbors which have successfully received the broadcast frame. The former information can be obtained by maintaining a local connectivity table (LCT) in each node. Let $i = 1, 2, \cdots, N$ index the N mobile nodes in the population. By definition, node i "hears" (is connected to) node j if i and j are within range and in line-of-sight of each other. To represent the connections among nodes, we use an $N \times N$ square matrix M such that the element m_{ij} is

$$m_{ij} = \begin{cases} 1, & \text{if } i \text{ hears } j \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Therefore, the number of neighbor nodes of node i is equal to $\sum_{j=1}^{N} m_{ij} - 1$. Each time a node receives a frame, it will update its LCT according to frame's source address. Without losing generality, entries in LCT should be aged by timeout due to mobility.

3.2.1 Broadcast Acknowledgment Scheme

Recall that the uncertain broadcast problem is mainly caused by lack of any replying acknowledgment of the broadcast frame. To ensure the sender be aware of the status of its broadcasting, we slightly modify the IEEE 802.11 MAC protocol to provide broadcast acknowledgment. To avoid extra broadcast overheads, we enforce all recipients to response immediately in a following DIFS by applying the same collision avoidance procedure in CSMA/CA. The $50\,\mu$s DIFS time period, named as the Backoff Acknowledgement Window (BACK_W) in this scheme, is divided into several minislots and each recipient will randomly select one of them to transmit acknowledgment as shown in Fig. 1. Since the transmission time of a formal ACK frame (with necessary physical layer convergence protocol (PLCP) preamble and header) is longer than DIFS, the broadcast acknowledgment (BACK) message must be short enough to accommodate the minislot in the BACK_W. To help nodes recognize the BACK message, we adopt the *busy-tone* concept in the context of packet radio networks [19] to identify the BACK message. The busy-tone is a signal of sine wave and can be persisted for a time of period. Basically, the number of minislots in the BACK_W depends on the length of the minislot as shown in Fig. 1.

These minislots are designed for a receiver to inform sender the reception of the broadcast frame. As long as a node receives a broadcast frame, it will randomly choose a BACK minislot to fill the corresponding grille. Since the WLAN adapter uses half-duplex mode to access channel, the switching delay for sender and receiver to change the transceiver state is required. According to the PHY specification of IEEE 802.11, we need to allocate an enough time period, which is set as equal to SIFS (= $10\,\mu$s), for PHY

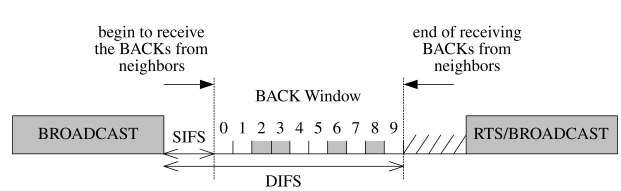


Fig. 1 An illustration of the BACK scheme.

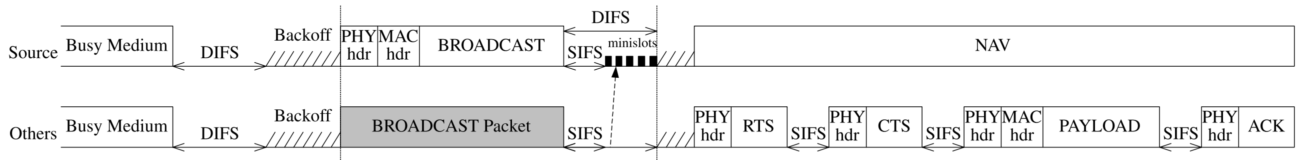


Fig. 2 An illustration of broadcast, RTS/CTS and backoff scheme of DCF.

layer transferring between receiving and transmitting states at the end of broadcast frame as shown in Fig. 1. Consequently, for x-bit minislots used in the n-Mb/sec WLAN, a number of $(\mathrm{DIFS} - \mathrm{SIFS})/(x/n) = (50 - 10)/(x/n)$ minislots will be allocated in the BACK_W. For example, if we use 4-bit minislots in 2 Mb/sec WLAN, we have 20 minislots (i.e., BACK_W = 20). We note that the proposed BACK scheme is compatible with any transmission rates in IEEE 802.11 family (e.g., 802.11b or 802.11g) since different rates only lead to different sizes of BACK_W.

Since the BACK messages are only used for notification, they can be treated as particular control signals between the broadcast sender and receivers. Hence, in this scheme, all mobile nodes can ignore the channel busy caused by these BACK signallings within the DIFS following the broadcast transmission and will contend the channel immediately after passing DIFS as the standard does. As a result, the proposed scheme will not waste any channel resource to acknowledge broadcast frames as shown in Fig. 2. We note that, in the multihop MANETs, it is quite possible that some mobile nodes hidden to broadcast sender may not recognize the BACK signals and, consequently, recognize such signals as channel busy. Therefore, they will defer one more DIFS interval for performing the backoff/contention procedure. This is the potential shortcoming of proposed scheme.

According to the proposed broadcast acknowledgment scheme, the broadcast sender would have sufficient information to help decide the necessity of rebroadcast. Let #(LCT) denote the number of active neighbor nodes of a node. As we mentioned above, the value of #(LCT) of any node i is $\sum_{j=1}^{N} m_{ij} - 1$. In the case of the number of received BACKs is less than #(LCT), the sender needs resend it again. Because the connectivity between nodes is time varying and the BACKs have a chance to be corrupted by noise or collisions, sender may fail to collect sufficient new BACKs no matter how many retransmissions it attempts. Thus, we still need a maximum broadcast retry threshold (MBRT) to minimize the unnecessary bandwidth wastage. Accordingly, a node will retransmit the broadcast frame until either the amount of BACKs is sufficient or the retry count reaches the MBRT. Obviously, a higher flooding fraction and a more bandwidth wastage will be derived when a higher MBRT is used.

3.3 Broadcast Frame Format

To avoid circulating broadcast frames in network, each broadcast frame should contain the following fields:

- Source Address (SA)
- Destination Address (DA)
- Broadcast ID (BID)
- Hop_count
- Retry_flag
- Data Payload

Every mobile node maintains a BID counter and increases the value by one when it has a new broadcast frame and the pair $\langle$SA, BID$\rangle$ uniquely identifies a broadcast packet [7], [14]. The Hop_count of a new broadcast frame is set to zero. Each time the broadcast frame is forwarded by a node, the associated Hop_count will be increased by one. According to the indication of Hop_count, we let the broadcast frame with the largest Hop_count has the highest priority in priority queue to shorten the propagation delay. The Retry_flag is used to identify whether the broadcast frame is a new one or a retransmitted one. This will help the receiver reply a correct BACK back to sender.

In addition, a node needs to maintain two additional counters, a back counter (BC) and a retry counter (RC), for each broadcast frame buffered in queue to make the decision of rebroadcast. The BC stands for a number of expecting repliers, which have successfully received the transmitted broadcast frame. A node will continuously rebroadcast it until the BC is decreased to zero. Initially, the BC is set as #(LCT)−1 or #(LCT) depending on the role which a node plays. It means that each forwarding node should set its BC as #(LCT)−1 except the original source since they receive the broadcast packet from their predecessors. Another counter RC is used to indicate that how many times of a broadcast frame have been transmitted. If the RC reaches the MBRT, the broadcast frame will be discarded immediately no matter how large the BC is.

Since a node may receive the identical broadcast frame from any of its neighbors before its broadcast forward, the bandwidth consumption can be further minimized by smartly reducing the redundant broadcasts. This can be done by detecting the Retry_flag, SA and BID of the received broadcast frame. That is, when a node detects a broadcast frame, which was received earlier, with Retry_flag = False, it decreases the associated BC of broadcast frame buffered in the priority queue if any. As a result, some waiting broadcast frames where BC = 0 could be quickly removed from transmission queue. This is another advantage of the proposed broadcast acknowledgment scheme. The broadcast transmitting and receiving procedures are listed in Figs. 3 and 4, respectively.

```
Procedure TRANSMIT_BROADCAST()
input: BFrame
begin
 set BC(BFrame) := #(LCT)-1; // or #(LCT) in the
   original sender
 set RC(BFrame) := 0;
 set BFrame→Retry_flag := False;
 While (BC(BFrame) > 0 and RC(BFrame) < MBRT )
 begin
    broadcast the BFrame and then wait the replied BACKs;
   receive all replied BACKs in BACK window;
   BC(BFrame) := BC(BFrame) - the number of new
     BACKs;
   BFrame→Retry_flag := True;
 end
 drop this BFrame;
end
```

Fig. 3 The procedure of transmitting broadcast frame.

```
Procedure RECEIVE_BROADCAST()
input: BFrame
begin
 if BFrame→BID > BIDTable[BFrame→SA] then
 // New broadcast frame
   if BFrame→DA = self address then
   // Arrive the destination
     receive the BFrame and response via unicasting;
   else
     if #(LCT) > 1 then //excluding the sender
       BFrame→Hop_count:=BFrame→Hop_count + 1;
       insert BFrame into priority queue and perform
       InsertionSort(Hop_count);
     endif
   endif
    select a random(BACK_W) to reply a new BACK;
    update BIDTable[BFrame→SA] := BFrame→BID;
 else // Duplicate broadcast frame
    if BFrame→BID = BTable[BFrame→SA] then
      if BFrame→Retry_flag = False then
        find the buffered BFrame, say Pkt, from local
        priority queue if any;
        if found Pkt then
          BC(Pkt):= BC(Pkt)-1;
          if BC(Pkt) = 0 then
            remove Pkt from priority queue;
          endif
        endif
      select a random(BACK_W) to reply a duplicate
      BACK;
      else
        drop this BFrame;
      endif
    endif
  endif
end
```

Fig. 4 The procedure of receiving a broadcast frame.

3.4 Priority Queue

In the original IEEE 802.11 MAC broadcast scheme, there is no priority between broadcast frames and ordinary data frames. The broadcast propagation may spend a long buffering delay at intermediate nodes even if the traffic load is light. This is because that each time a broadcast frame relayed by a node needs to wait in the FIFO queue. This is a fatal drawback in multihop communication network, especially when these priority frames have a delay bound. Therefore, the normal data and broadcast packets should be separated as two priority queues as shown in Fig. 5. In the proposed ADBS, broadcast packets will get a higher priority than normal data to be serviced. Moreover, the priority of broadcast packet is ranked by the Hop_count field as mentioned before. We also note that any received broadcast packet should be always inserted behind the packet which is already queued in the head of the buffer and is in the process of contention or transmission even though the Hop_count of the arrival packet is larger than that of the front packet.

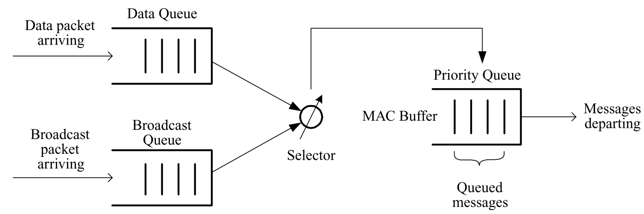


Fig. 5 Model of data queue and broadcast queue in network layer and a priority queue in MAC layer.

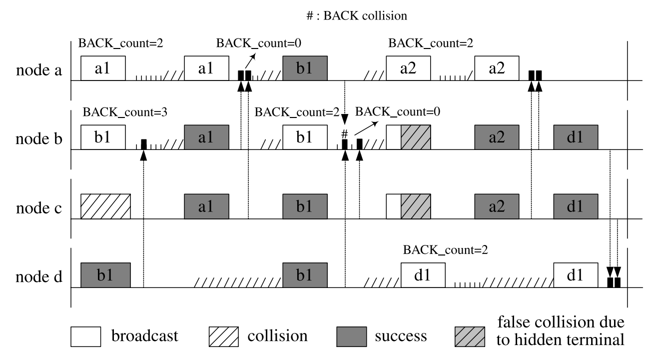


Fig. 6 An example of proposed reliable broadcast scheme.

3.5 An Example of Broadcast Transmission

Figure 6 illustrates an example of broadcast transmission that collisions may resulted from neighbors or hidden terminals. Assume nodes **a**, **b** and **c** can directly communicate with each other and node **d** is the hidden terminal to node **a**. Meantime node **d** can transmit/receive data to/from both nodes **b** and **c** but node **a** (see Fig. 7). At first, assume nodes **a** and **b** have broadcast frames, named as a1 and b1 and they transmit them at the same time. Node **c** will receive a corrupted frame since the signals of frames a1 and b1 are interfering to each other. Node **d** will receive the broadcast frame b1 and randomly choose a BACK minislot to reply an acknowledgment to node **b**. Consequently, after the first broadcast transmission, node **a**'s BC is still 2 and node **b**'s BC becomes 2. After then, we assume node **a** select a shorter backoff window than node **b**'s in the retransmission. Therefore, nodes **b** and **c** will receive frame a1 and return new BACKs. As long as node **a** receives two

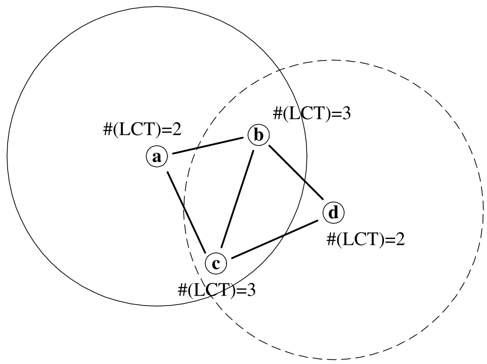


Fig. 7 A wireless network where nodes **a** and **d** are hidden to each other.

expected BACKs, it will stop retransmitting broadcast frame a1 since its BC is decreased to zero. On the contrary, node **b** will keep retransmitting its broadcast frame since its BC (= 2) is not zero.

In the second retransmission of frame b1, we assume that the replies of nodes **a** and **d** collide in the 3rd minislot of the following BACK window and only node **c** succeeds. The overlapped BACKs is also counted as once since nodes could not distinguish the collision in busy-tone approach. Thus, node **b**'s BC is reduced to 0 by the BACKs from nodes **a**, **c** and **d**. This is a drawback of proposed ADBS, however, in our opinions, it is the simple way to provide the broadcast reliability needed for many important upper layer protocols.

After frame b1, we assume nodes **a** and **d** get the right to access channel. If their transmissions (frame a2 and d1) are overlapping, none of the broadcast frames will be received by nodes **b** and **c** and the other retransmissions for them are inevitable. Unfortunately, paper [3] concluded that about over 60% frame transmissions in wireless networks will become failure by the hidden node situation. This implies that conventional broadcast scheme without acknowledgment in IEEE 802.11 MAC protocol will lose lots of broadcast frames.

4. The Reliable Multicast Transmission Scheme

Unlike the broadcast method, multicast transmission specifies particular nodes to forward and receive. Basically, intermediate nodes in multicast tree are responsible for forwarding multicast data to some specified downstream nodes. (In this paper, we do not consider the way of finding the best multicast tree but how to support reliable multicast transmissions underlaying the IEEE 802.11 MAC protocol.) Even though only some neighboring nodes need correctly receive the multicast data, the uncertainty of multicast transmission still exists due to the lack of acknowledgments. For an intermediate node in multicast tree, the reliability of multicast transmission can be easily achieved by setting the MBRT = ∞ and the number of expected BACKs is equal to the number of downstream nodes in multicast tree. As a result, an intermediate node will retransmit the multicast frame until receive all BACKs from its recipients successfully. Certainly, this approach is inefficient.

```
Procedure TRANSMIT_MULTICAST()
input: MPkt
begin
  set all elements in BACK_array(MPkt) as FALSE;
  // The array size is equal to the number of downstream
  // nodes of sending node in the multicast tree;
 While (at least one element in BACK_array(MPkt) is FALSE)
 begin
    broadcast the MPkt and then wait the replied BACKs;
    receive all replied BACKs in BACK window;
     for (each successful BACK) do
        set the corresponding element in BACK_array(MPkt)
        = TRUE;
  end
  drop this MPkt;
end
```

Fig. 8 The procedure of transmitting multicast frame.

```
Procedure RECEIVE_MULTICAST()
input: MPkt
begin
  if MPkt→MGA = self MGA then
    if MPkt→MSN = MSNTable[MPkt→SA]+1 then
    // new multicast frame
      receive the MPkt;
      update MSNTable[MPkt→SA] := MPkt→MSN;
       if this node is an intermediate node in
       multicast tree then
          append MPkt into transmit queue;
       endif
    else // duplicate or non-sequencing multicast frame
       drop this MPkt;
    endif
    reply BACK onto specified minislot in BACK Window;
  else
    drop this MPkt;
  endif
end
```

Fig. 9 The procedure of handling a received multicast frame.

In our scheme, during the multicast tree setup phase, a forwarding node would assign its each successors with a unique identification. These identifications are mapped to minislots in BACK_W and one for each. Thus no BACK collision will occur in the proposed multicast transmission scheme. However, in case of the connection being broken by a successor due to its failure or movement, the predecessor will keep retransmitting forever. Therefore, mobile nodes still need the MBRT, the link failure detection method and the route recovery scheme to solve the potential problem. Since we only focus on the MAC layer, the route maintenance scheme is not discussed and considered here.

The frame format of multicast is different from broadcast and is shown as follows:

- Source Address (SA)
- Multicast Group Address (MGA)
- Multicast Series Number (MSN)
- Data Payload

The multicast series number (MSN) is used for multicast members to reassemble received multicast MPUDs. Each time when a multicast member receives corresponding multicast frames (either the new or the duplicated one), it will reply the BACK onto the specified minislot in the following BACK window. Since the transmissions of BACKs for multicast frame is collision-free, multicast sender can easily decide the necessity of retransmission. Therefore, instead of using the BC, the multicast sender maintains a liner array to identify which downstream node does not reply BACK yet. Hereafter, the Retry_flag and MBRT used for broadcast frame are useless for multicast frame. Based on this scheme, each multicast frame only generates little extra overhead. The multicast transmission and receive procedures are given in Figs. 8 and 9, respectively.

5. Simulation Model and Results

In order to evaluate the proposed adaptive broadcast and reliable multicast schemes, which are based on the DCF of IEEE 802.11 [6] WLAN, we considered the realistic system parameters listed in the DSSS physical specification as shown in Table 1. The 802.11 DCF uses RTS/CTS exchange precedes data frame transmission for performing the *virtual carrier sensing* as well as channel reservation to reduce the impact of the hidden terminal problem [19]. Data frame transmission is followed by an ACK and the RTS/CTS frames are sent using physical carrier sensing. Broadcast/Multicast frame transmission follows by a number of BACKs (denoted as BACK_W) can be treated as control frame in proposed scheme. The radio model uses characteristics similar to a commercial radio interface, Lucent's WaveLAN [4], [21]. The WaveLAN adapter is modeled as a shared-media radio with a nominal bit rate of 2 Mb/sec and a nominal radio range of 100 m.

5.1 Simulation Models

In our simulations, we simulated a scenario of N mobile nodes active in a square area of 300 m × 300 m. The initial location of each node is assigned randomly within the area. Excepting the first node, the other nodes will be reallocated if they do not have at least one neighbor. This ensures that the simulated network topology is a 'connected' graph and the flooding behavior is meaningful. For the sake of comparisons, nodes are assumed to stay at its original spot during the simulation duration. Each mobile node has one transceiver and its transmission range is 100 m (in 2 Mb/sec). The background data packets arrival rate of each mobile node follows the Poisson distribution with a mean λ_d. We set the $\lambda_d = 10^{-5}$ frames/slot/node (0.5 frames/sec/node) throughout all simulations. The packet length is an exponential distribution with a mean of L time slots. The mean length of packets is set according to the analyzed average packet length on ordinary LAN [8], which is about 50–150 bytes, i.e., about 10~30 time slots in 2 Mb/sec transmission rate. These popular TCP/UDP packets occupy overall traffic loading over 74%. Thus, we assume the unicast/multicast data frame length L = 47 time slots (including PHY and MAC headers ≈ 17 time slots) in our simulations. The broadcast (or multicast) request arrival rate λ_b of each mobile node also follows the Poisson distribution, and the request frame length is a fixed length of 25 octets. The λ_b is considered from 10^{-5} to 10^{-4} frames/slot/node (equal to 0.5 to 5 frames/sec/node) in a step of 10^{-5}. Each node maintains an infinite *waiting buffer* (priority queue) in MAC layer. It contains all data frames and broadcast/multicast request frames waiting for transmission, in which, broadcast/multicast request frames have a higher priority than data frames. Each simulation run lasts 60 seconds ($\approx 3\times10^6$ time slots) and each simulation result is obtained by averaging the results from one hundred independent simulation runs.

Table 1 System parameters in simulations.

Parameter	Normal Value
Channel bit rate	2 Mb/sec
Transmission Range (2 Mb/sec)	100 m
RTS frame length	160 bits
CTS frame length	112 bits
ACK frame length	112 bits
Broadcast request frame length	25 Octets
Unicast/Multicast data frame length	200 Octets
Preamble and PLCP header	192 μs
MAC header	34 octets
A slot time (τ)	20 μs
SIFS	10 μs
DIFS	50 μs
aCWmin	31 slots
aCWmax	1023 slots
Air propagation delay (δ)	1 μs
Density 1	30 nodes
Density 2	60 nodes
Density 3	100 nodes

5.2 Simulation Results

Four important performance metrics are investigated:

- ***Broadcast flooding fraction***—The successfully received broadcast frames to number of mobile nodes in networks per each broadcast frame.
- ***Broadcast retry overhead***—The number of broadcast retransmissions to all broadcast transmissions in networks.
- ***Multicast request success probability***—The probability of successfully receiving the multicast request frame by all multicast members of a multicast group.
- ***Multicast data retry overhead***—The number of multicast retransmissions to all multicast transmissions in networks.

In the first simulation, we consider three different network densities, which are generated by allocating 30, 60 and 100 mobile nodes into a fixed square area 300 m × 300 m. In order to evaluate the effect of broadcast flooding fraction under different network densities, we adopt a light broadcast load $\lambda_b = 10^{-6}$ frames/slot/node. We first investigate the efficiency of broadcast flooding by using IEEE 802.11 (with-

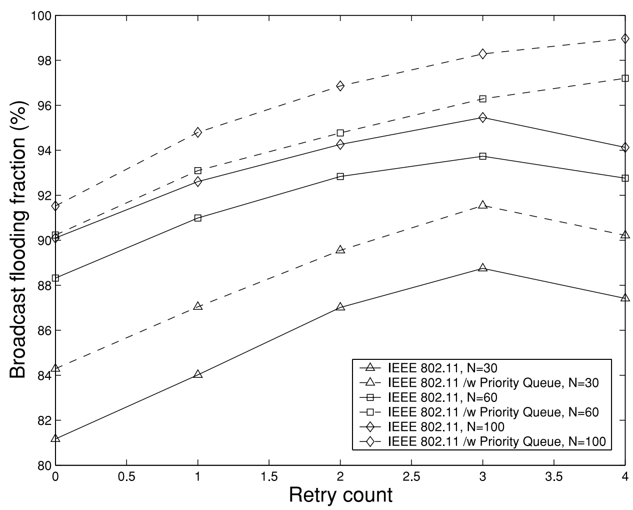


Fig. 10 Comparisons of derived broadcast flooding fractions by the IEEE 802.11 with and without priority queue under different retry counts in DBS. The broadcast frame arrival rate λ_b is 10^{-6} frames/slot/node.

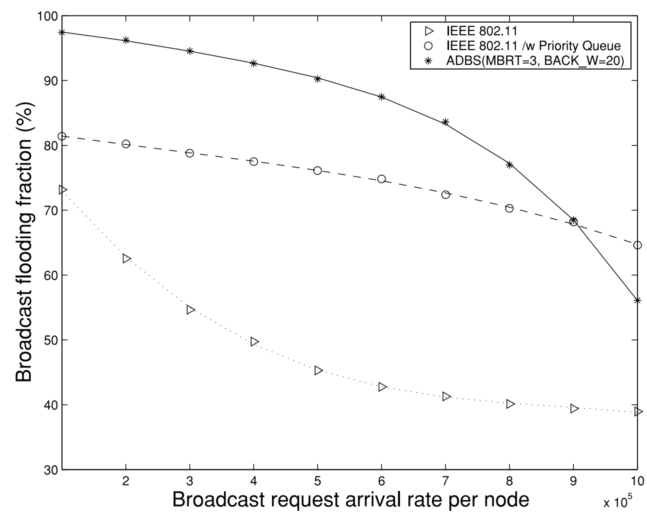


Fig. 11 Comparisons of the derived broadcast flooding fractions by different schemes under different network loads.

out acknowledgement) and vary the retry count in DBS to observe broadcast flooding results. From the results, shown in Fig. 10, we can see that with a lower network density (i.e., with fewer nodes in networks), a worse flooding fraction will be obtained. For example, without the priority queue (the traditional broadcast scheme), the best flooding fractions in cases $N = 30$, $N = 60$ and $N = 100$ as retry count is zero (MBRT = 0) are about 81%, 88% and 90%, respectively. This indicates that a higher density network topology will obtain a higher flooding fraction since the average link degree of a node in lower density is smaller than that of a network with higher density. In other words, in a higher density network, even though a broadcast frame fails in reaching some nodes, these nodes still have a higher probability to receive the broadcast frame from their neighbors, which are also neighboring to the sender. This result also shows that the flooding capability is linear proportional (contra-proportional) with the number of retries when the network load including the extra control overhead is under (beyond) the saturated load.

Moreover, the approach of combining IEEE 802.11 with DBS and priority queue can easily achieve a higher flooding fraction in all cases since the simple priority queue approach speeds up the propagation speed for an ongoing flooding. This is because that higher priority frames should be queued in front buffer and performed prior (speeded up) to avoid unnecessary timeout due to longer queuing delay in buffer or a limited buffer size (no available space for store and would be dropped). We also show that the derived broadcast flooding fraction by this approach increases with the increment of retry count. The highest flooding fraction can be up to 98% in high network density where $N = 100$ and retry count is 4. From Fig. 10, we concluded two results: 1) the broadcast retry may raise or degrade the flooding fraction depending on the generated control overheads; 2) the proposed priority queue approach can substantially enhance the flooding fraction despite the network density. In the following simulations, we only consider the light network density case (30 nodes) since the worse performance occurs in this case as shown in Fig. 10.

Figure 11 shows the derived flooding fractions by original IEEE 802.11 broadcast scheme, the IEEE 802.11 with priority queue, and the proposed ADBS under different broadcast request loads. We find that the flooding fraction of IEEE 802.11 is contra-proportional with the broadcast request load and the flooding fraction may be down to only 39% when $\lambda_b = 10^{-4}$. This implies that over 60% mobile nodes can not be notified via broadcasting and, consequently, routing protocols like DSR and AODV become useless in IEEE 802.11 based multihop MANETs. Contrarily, the proposed ADBS scheme can reach about 97% flooding fraction when $\lambda_b = 10^{-5}$, MBRT = 3 and BACK_W = 20. Even though when $\lambda_b = 10^{-4}$, our scheme still has 60% flooding fraction. From our observations, the performance degradation is mainly caused by the generated extra broadcast retry overhead and the original broadcast request load. We also note that IEEE 802.11 broadcast scheme with priority queue will outperform the proposed ADBS when $\lambda_b = 10^{-4}$. This is because that no retry overhead will be generated by the priority queue approach. Therefore, the IEEE 802.11 with priority queue will sustain an acceptable flooding fraction even when the network load is heavy.

Figure 12 shows the generated retry overheads under different numbers of expected BACKs and different MBRTs in the ADBS with BACK_W = 5. We can find that a larger number of expected BACKs will result in a higher broadcast retry overhead. The retry overhead will finally saturate at (MBRT)/(MBRT + 1) by the MBRT. A broadcast retry overhead y% means that the number of broadcast transmissions contains y% redundant (retransmitted) broadcast frames. For example, if $y = 50$, one retransmission is necessary for each broadcast transmission in networks.

In order to investigate how the MBRT and the

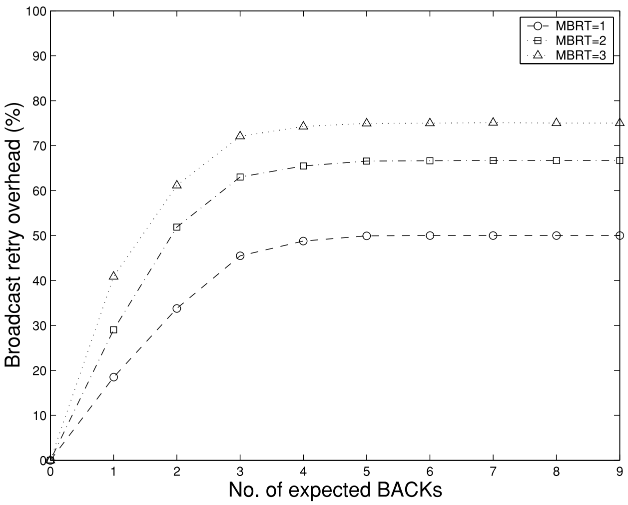


Fig. 12 The derived retry overheads by proposed ADBS scheme under different numbers of expected BACKs.

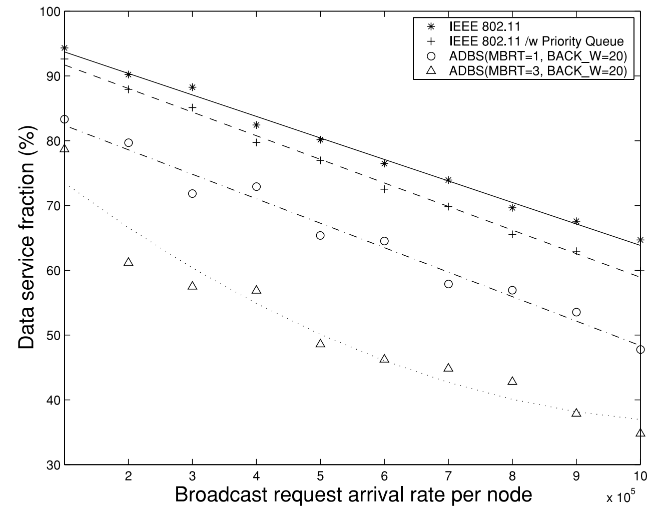


Fig. 14 Comparisons of the derived data fraction by IEEE 802.11 and proposed ADBS under different MBRTs, BACK_Ws and network loads.

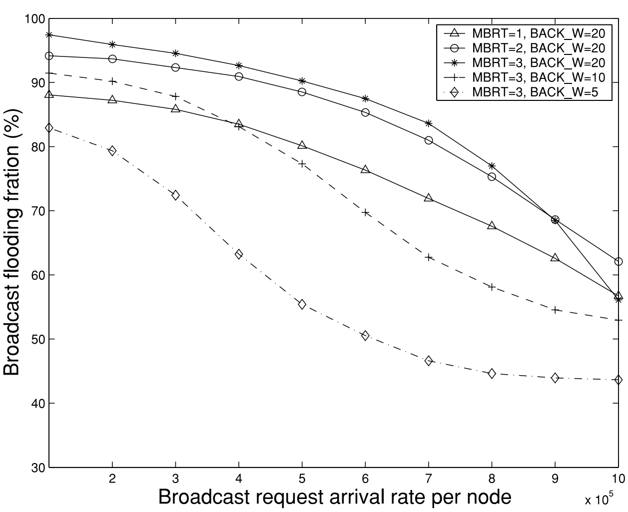


Fig. 13 Comparisons of the derived broadcast flooding fractions by proposed ADBS under different MBRTs, BACK_Ws and network loads.

BACK_W affect the efficiency of proposed ADBS, we consider different combinations of MBRT and BACK_W in simulations. Figure 13 illustrates that a larger MBRT or BACK_W will result in a higher flooding fraction. Moreover, the improvement of flooding fraction by enlarging the BACK_W is more obvious than enlarging the MBRT. If the BACK_W size is small, a lot of BACKs will collide with each other and the sender will rebroadcast as many times as possible. Consequently, these rebroadcasts will consume network bandwidth, increase queuing length and slow down the propagation speed of flooding to reach all nodes. We also find that a larger MBRT should be used to derive a higher flooding fraction if the number of minislots in BACK window is sufficient (e.g., BACK_W = 20).

We further observe how the broadcast load affects the data service rate in networks. For observing the maximal data service rates in different broadcast schemes, we assume every arrival data frame only destine to the neighboring node. Figure 14 shows the derived data service fractions derived by original IEEE 802.11 MAC protocol and our proposed scheme under different broadcast request loads. The data service fraction is defined as the ratio of the number of successfully transmitted data frames to total number of generated data frames in networks. The amount of broadcast requests can be treated as the background traffic in network while servicing data frames. Instructively, a heavy broadcast load or a heavy broadcast retry overhead will reduce the speed of servicing data. We can see that the IEEE 802.11 provides a higher data service rate than proposed broadcast scheme since the original IEEE 802.11 generates little broadcast overhead. However, referring to Fig. 11, IEEE 802.11 has the lowest probability of finding an available route from source to destination in realistic multihop MANETs environment, that will significantly downgrade the actual data service rate.

Figure 15 shows the derived multicast request success probabilities derived by original IEEE 802.11, the IEEE 802.11 with priority queue and the proposed ADBS under different multicast group sizes (MGS) and multicast request loads. If a multicast request with a larger MGS, the probability of this request reaches all multicast members in networks will becomes smaller as shown in Fig. 15. When MGS = 4, the probability of IEEE 802.11 successfully delivers the multicast request frame to corresponding nodes is only 29% when multicast request load per node is 10^{-5}. On the contrary, our scheme can reach up to 90% that implies the ADBS can support multicast tree discovery needs of multicast routing protocols (e.g., MAODV protocol).

In the following simulations, we do not consider the way of establishing the multicast tree for a multicast request, but put our attentions on the individual impact of an intermediate node while forwarding multicast data frames to the downstream nodes in a multicast tree. Assume a multicast tree with X nodes (including all multicast members and in-

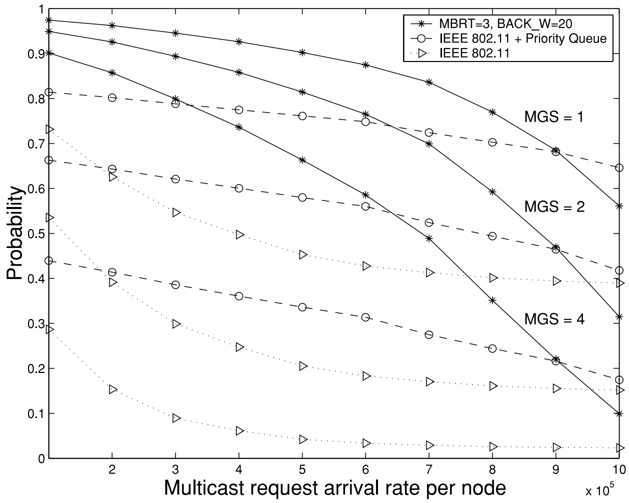


Fig. 15 Comparisons of the derived multicast request success probabilities by proposed broadcast schemes and IEEE 802.11 under different MGSs and multicast request loads.

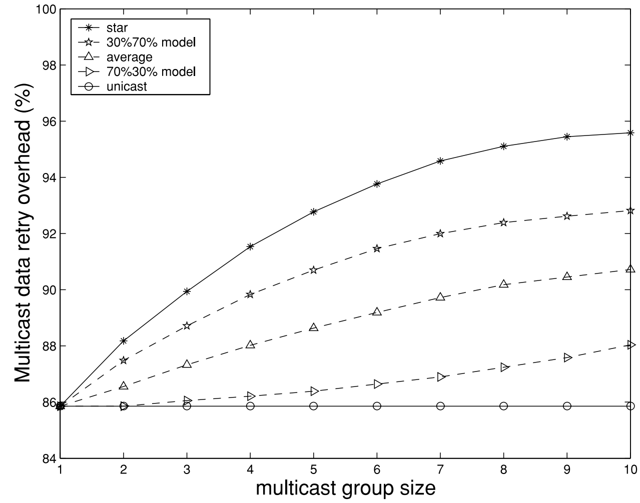


Fig. 16 Comparisons of the derived multicast data retry overheads under different multicast tree topologies and multicast group sizes.

Table 2 Multicast simulation model (degrees of nodes).

Multicast size	70–30%	average	30–70%
1	1	1	1
2	1+1	1+1	2
3	1+1+1	1+2	1+2
4	1+1+1+1	1+2+1	1+3
5	1+1+1+1+1	1+2+2	1+4
6	1+1+1+1+1+1	1+2+3	1+5
7	1+1+1+1+1+2	1+2+3+1	2+5
8	1+1+1+1+1+1+2	1+2+3+2	2+6
9	1+1+1+1+1+1+1+2	1+2+3+3	2+7
10	1+1+1+1+1+1+1+3	1+2+3+4	3+7

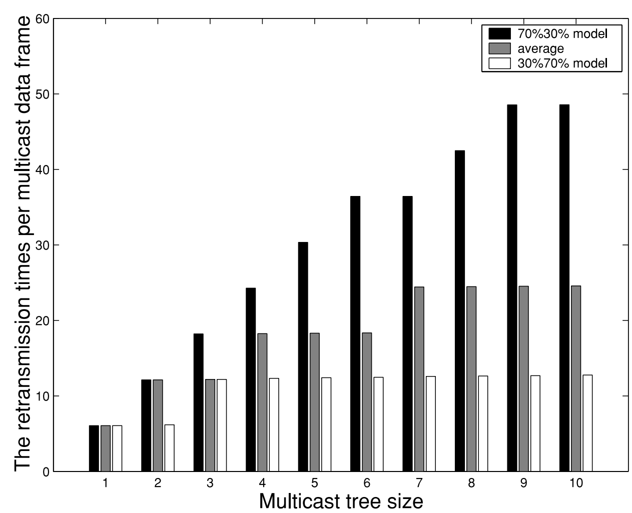


Fig. 17 The number of generated retransmissions times per multicast data frame under multicast tree topologies and multicast group sizes.

termediate nodes), we have MGS $\leq X$ and the number of non-multicast nodes in tree is equal to $X -$ MGS. In our simulations, we consider three different multicast tree topologies: 70–30% model, average model and 30–70% model. The 70–30% model means that 70% nodes are linked one by one in one line and the remaining 30% nodes are linked as multicast. The considered multicast tree somewhat likes a broom. The average model makes the numbers of intermediate nodes with different numbers of downstream nodes as equal as possible. The 30–70% model constructs a multicast tree with at most two intermediate nodes. One intermediate node links 70% nodes in multicast tree and the other links the remaining nodes. Three different models for multicast tree topology are listed in Table 2.

Figure 16 shows the average generated multicast data retry overheads by each intermediate node under different multicast tree sizes. The multicast data generation rate λ_m of the multicast source is set 10^{-4} frames/slot/node and the background traffic is set 10^{-5} frames/slot/node. The highest and lowest curves show the overheads generated by the multicast tree with star (single broadcast) and linked-list (unicast) topologies, respectively. The 30–70% model and 70–30% model derive the second highest and second lowest multicast data retry overheads. From these curves, we can easily find that the increased overheads will get smaller and smaller as increasing the multicast link degree per intermediate node.

We also note that the lowest curve is flat since the multicast data is dealt as unicast data. This means that at least 86% retry overhead is needed for achieving one hundred percent data delivery. In other words, for each multicast transmission in an intermediate node, a number of $86/(100 - 86) = 6.14$ retransmissions will be generated. Accordingly, if the multicast data is transmitted by several independent unicast links either in star or linked-list topology, the number of repeated transmission times will be the number of unicast links multiplies by 6.14. Thus, we emphasize that a tree topology with higher multicast data retry overhead does not mean that it will generate more transmissions in network. In a word, combining the retry overhead (in percentage) with the number of intermediate nodes of tree, the smallest number of multicast retransmissions will be derived when the multicast tree size is equal to MGS (i.e., MGS = X) and the height of the multicast tree is minimal. This conclusion can be found in Fig. 17.

6. Conclusions

This paper pointed out the uncertain broadcast or multicast problem in IEEE 802.11 multihop MANETs. Without a robust broadcast scheme, some well-known multihop routing protocols will become inefficient in MANETs. Thus, in this paper, we propose a novel broadcast acknowledgment scheme and priority queue to enhance the reliability and efficiency of data broadcasting by slightly modifying the IEEE 802.11 MAC protocol. These two schemes, which are still compatible with the standard, can substantially minimize unnecessary broadcast retry overheads and propagation delay of broadcast frames in MANETs. Furthermore, the erroneous multicast data transmissions can be recovered by using this scheme. Simulation results showe that, with moderate network load, the proposed broadcast scheme can provide an acceptable flooding fraction as well as multicast request success probability. These results encouraged us to realize the IEEE 802.11 based multihop MANETs by proposed schemes.

References

[1] I. Chlamtac and S. Kutten, "On broadcasting in radio networks-problem analysis and protocol design," IEEE Trans. Commun., vol.COM-33, no.12, pp.1240–1246, Dec. 1985.

[2] I. Chlamtac and S. Kutten, "Tree based broadcasting in multihop radio networks," IEEE Trans. Comput., vol.C-36, no.10, pp.1209–1223, Oct. 1987.

[3] J. Deng and Z.J. Haas, "Dual busy tone multiple access (DBTMA): A new medium access control for packet radio networks," Proc. IEEE Int. Conf. Universal Personal Commun. (ICUPC'98), vol.2, pp.973–977, Oct. 1998.

[4] D. Eckhardt and P. Steenkiste, "Measurement and analysis of the error characteristics of an in-building wireless network," Proc. ACM SIGCOMM'96, pp.243–254, Oct. 1996.

[5] T.A. ElBatt, S.V. Krishnamurthy, D. Connors, and S. Dao, "Power management for throughput enhancement in wireless ad-hoc networks," Proc. IEEE ICC 2000, vol.3, pp.1506–1513, June 2000.

[6] IEEE 802.11 Working Group, "Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," ANSI/IEEE Std. 802.11, Sept. 1999.

[7] D.B. Johnson and D.A. Maltz, "Dynamic source routing in ad-hoc wireless networks," in Mobile Computing, eds. T. Imielinski and H. Korth, chapter 5, pp.153–181, Kluwer Academic Publishers, 1996.

[8] K.M. Khalil, K.Q. Luc, and D.V. Wilson, "LAN traffic analysis and workload characterization," Proc. 15th Conf. Local Computer Networks, pp.112–122, 1990.

[9] I. Koutsopoulos, D. Connors, A. Savvides, and S.K. Dao, "Intra-team multihop broadcasting (ITMB): A MAC layer protocol for efficient control signaling in wireless ad-hoc networks," Proc. IEEE ICC 2000, vol.3, pp.1723–1727, New Orleans, June 2000.

[10] J. Kuri and S.K. Kasera, "Reliable multicast in multi-access wireless LANs," ACM/Kluwer Wireless Networks, vol.7, no.4, pp.359–369, July 2001.

[11] W. Lou and J. Wu, "On reducing broadcast redundancy in ad hoc wireless networks," IEEE Trans. Mobile Computing, vol.1, no.2, pp.111–122, April–June 2002.

[12] J. Macker and S. Corson, "Mobile ad hoc networks (MANET)," IETF WG Charter., http://www.ietf.org/html.charters/manet-charter.html, 1997.

[13] C.E. Perkins and E.M. Royer, "Ad-hoc on-demand distance vector routing," Proc. 2nd IEEE Wksp. Mobile Comp. Sys. and App., pp.90–100, Feb. 1999.

[14] C.E. Perkins, E.M. Royer, and S.R. Das, "Ad hoc on-demand distance vector (AODV) routing," Internet Draft draft-ietf-manet-aodv-08.txt, March 2001.

[15] S.-T. Sheu and J. Chen, "A novel delay-oriented shortest path routing protocol for mobile ad hoc networks," Proc. IEEE ICC 2001, vol.6, pp.1930–1934, Helsinki, Finland, June 2001.

[16] S.-T. Sheu, Y. Tsai, and J. Chen, "Delay-oriented routing protocol for wireless ad hoc networks," IEICE Trans. Commun., vol.E84-B, no.6, pp.1581–1587, June 2001.

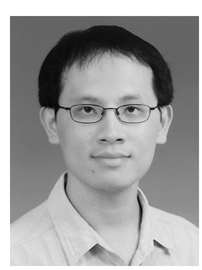
[17] S.-T. Sheu, Y. Tsai, and J. Chen, "MR^{2}RP: The multi-rate and multi-range routing protocol for IEEE 802.11 ad hoc wireless networks," ACM/Kluwer Wireless Networks, vol.9, no.2, pp.165–177, March 2003.

[18] S.-T. Sheu, Y. Tsai, and J. Chen, "A highly reliable broadcast scheme for IEEE 802.11 multi-hop ad hoc networks," Proc. IEEE ICC 2002, vol.1, pp.610–615, New York, April/May 2002.

[19] F.A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part II-the hidden terminal problem in carrier sense multiple-access models and the busy-tone solution," IEEE Trans. Commun., vol.COM-23, no.12, pp.1417–1433, Dec. 1975.

[20] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," ACM/Kluwer Wireless Networks, vol.8, no.2–3, pp.153–167, 2002.

[21] B. Tuch, "Development of waveLAN, an ISM band wireless LAN," AT&T Tech. J., vol.72, no.4, pp.27–33, July/Aug. 1993.



Jenhui Chen received his B.S. and Ph.D. degrees in Computer Science and Information Engineering (CSIE) from Tamkang University in July 1998 and January 2003, respectively. In the spring of 2003, he joined the faculty of Computer Science and Information Engineering Department at Chang Gung University and served as the Assistant Professor. Dr. Chen served as the referee for several famous international journals including IEEE Transactions on Computers, IEEE Transactions on Fuzzy Systems, IEEE Transactions on Mobile Computing, IEEE Transactions on Wireless Communications, IEEE Transactions on Parallel and Distributed Systems, and ACM/Kluwer Mobile Networks and Applications (MONET). He served as a Program Committee Member for the 2005 IEEE International Conference on Parallel and Distributed Systems and a referee for the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems. Dr. Chen was a Local Chair of Mobile Computing Workshop 2005. His main research interests include design, analysis, and implementation of communication and network protocols, wireless networks, robotics, and artificial intelligence. He is a member of IEEE.

Shiann-Tsong Sheu received his B.S. degree in Applied Mathematics from National Chung Hsing University in 1990, and obtained his Ph.D. degree in Computer Science from National Tsing Hua University in May of 1995. From 1995 to 2002, he was an Associate Professor at the Department of Electrical Engineering, Tamkang University. From February 2002 to July 2005, he was a Professor at the Department of Electrical Engineering, Tamkang University. He is currently a Professor of Department of Communication Engineering, National Central University. Dr. Sheu received the outstanding young researcher award by the IEEE Communication Society Asia Pacific Board in 2002. His research interests include next-generation wireless communication, optical networks, protocol designs and intelligent control algorithms.