Information Sciences and Computer Engineering, Vol. 1, No. 1, (2010) 37–44

PHARUS

International Journal of
Information Sciences and Computer Engineering

journal homepage: http://www.ijisce.org

An Efficient ARQ Scheme for Reliable Video Broadcasting in Wireless LANs†

Jun Peng*

Department of Electrical Engineering, University of Texas-Pan American, 1201 W. University Drive, Edinburg, TX 78541

Abstract– **Video streaming has become a popular and important type of network application. Meanwhile, wireless LANs have been gaining popularity too. It has thus become important to address the reliability problem for video streaming in wireless LANs and for video streaming from the Internet to the users in wireless LANs. This paper proposes an efficient Automatic Repeat Request (ARQ) scheme to address the reliability problem for video broadcasting in such scenarios. Based on a novel approach of virtual bitmaps, the proposed scheme achieves high efficiency in recovering broadcasting losses in wireless LANs but does not use out-of-band signals. The comprehensive simulation results in the paper show that, as compared with existing schemes, the proposed scheme improves node broadcast throughput by from several tens to more than one hundred percent in wireless LANs.**

1. Introduction

Both video streaming applications and wireless LANs have been quickly gaining popularity. It has therefore become common that video data are streamed in wireless LANs or from the Internet to users in wireless LANs. Because of the broadcasting nature of wireless transmission, video broadcasting in wireless LANs can significantly improve the efficiency of video streaming in such scenarios. However, packets in wireless LANs may experience heavy losses due to channel errors and packet collisions. Although video streaming may tolerate some losses, excessive losses significantly degrade the viewing experience of users. The reliability problem for video broadcasting in wireless LANs thus needs to be addressed for such applications. The popular IEEE 802.11 wireless LAN standard [3] recommends Automatic Repeat Request (ARQ) for unicast packets at the MAC sublayer of the data link layer. Broadcast packets, however, have not enjoyed the same level of reliability. We propose in this paper a slim ARQ scheme to address that problem for video broadcasting in wireless LANs.

Wireless channels are prone to errors due to multipath fading and interference. In addition, packet collisions can be common in wireless LANs due to the hidden terminal problem [4]. In the popular IEEE 802.11 wireless LAN standard, the Distributed Coordination Function (DCF) [3] uses Network Allocation Vector (NAV) and ARQ to deal with hidden terminals and packet losses, respectively, for *unicast* packets. There are existing efforts extending both NAV and ARQ from the unicast case to the broadcast case [5, 6, 7]. This paper focuses on the ARQ part for two reasons. First, hidden terminal avoidance [4, 8, 9] and packet loss recovery [10] are two independent research topics in general, although they can be used together for improving link reliability [11]. Second, even in the unicast case, the NAV approach has limited effectiveness in dealing with hidden terminals [12, 13].

The challenge in designing an ARQ scheme for reliable broadcasting is for a sender to *efficiently* gather the acknowledgment (ACK) information from the multiple receivers without collisions. An existing common approach is to seek traditional ACK frames from the receivers one by one [5, 7]. ACK frames, however, carry significant overhead due to synchronization and control requirements, which makes the common approach inefficient, particularly when there are a significant number of receivers for each packet. To reduce the control overhead in the data channel, another approach is to use out-of-band tones to deliver the ARQ control information [14, 15], which, however, significantly increases system complexity and cost due to the requirement of out-of-band channels.

To combat the disadvantages of existing approaches to reliable broadcasting, the proposed scheme in this paper uses a new approach of virtual ACK bitmaps. With the new approach, a sender uses a virtual bitmap to collect the packet reception information from its multiple receivers in an efficient way. In particular, each bit in the bitmap indicates the reception status at a receiver. This paper addresses the details on how a virtual bitmap is built in a distributed way and how the bits in a virtual bitmap are filled in by receivers. In addition, this paper analyzes how the parameters of the bitmap should be set so that the proposed scheme works in a fully distributed way. This paper also shows comprehensive simulation results on how the proposed scheme improves broadcasting efficiency as compared to existing schemes for reliable wireless broadcasting.

The rest of the paper is organized as follows. Section 2 presents the details of the proposed ARQ scheme for reliable video broadcasting in wireless LANs. Section 3 analyzes the timeslot synchronization in a virtual bitmap of the proposed scheme. Section 4 shows comprehensive simulation results on the efficiency of the proposed scheme. Finally, Section 5 concludes the paper.

*Corresponding author:
Email address: jpeng@utpa.edu, Ph: +1 9563812688

†The results in this paper are partly based on the results in a conference paper [1] and a letter [2].

2. The Efficient ARQ Scheme for Reliable Video Broadcasting in Wireless LANs

When video data are streamed in wireless LANs or from the Internet to the users in wireless LANs, the reliability problem for video broadcasting in wireless LANs becomes an important topic. Reliability at the transport layer is usually not provided for video streaming and thus the video quality will significantly degrade when wireless links introduce heavy packet losses. To combat this problem for video broadcasting in wireless LANs, the proposed ARQ scheme in this paper sets its design objective as using the minimum control overhead to significantly improve the reliability of video broadcasting in wireless LANs.

To achieve its design objective, the proposed Broadcast ARQ scheme in this paper uses in-band pulses, i.e., ACK pulses, instead of traditional ACK packets, to deliver ARQ control information. In-band pulses are blank, un-modulated carriers of particular lengths. In-band pulses have been used to address various problems in wireless networks, such as collision avoidance [16, 17], service differentiation [18], and channel state inference [19, 17]. In the proposed scheme, when a receiver needs to confirm its reception of a packet with its sender, it transmits an in-band pulse in a timeslot specified by the sender in a virtual bitmap. In the virtual bitmap, the broadcast sender interprets a bit as "1" if it receives a pulse in the corresponding timeslot. Otherwise, the bit is interpreted as "0". The rest of this section presents the details of the proposed scheme.

2.1. Virtual Bitmap Structure

There are multiple important elements of a virtual bitmap, which include the bitmap starting time, the timeslot length, the timeslot order, and the number of timeslots in the bitmap, as shown in Fig. 1. A broadcast sender must specify all the elements for its virtual bitmap so that each receiver can fill in its bit in the right timeslot.

A natural starting time of a virtual ACK bitmap for a receiver is the time when the receiver fully receives the broadcast packet. Note that only the nodes that successfully receive the broadcast packet send back ACK information. Therefore, the nodes that intend to fill in the bitmap can agree on the starting time of the virtual bitmap. However, the signal propagation delays between the sender and its receivers may be different, which means that receivers may not finish receiving the broadcast packet at exactly the same time. Therefore, there must be guard times between the timeslots in the virtual bitmap. The length of the guard time, τ_g, is determined by the transmission range of the sender, R_{tx}. In particular, τ_g must be at least $\frac{2R_{tx}}{c}$, where c is the signal propagation speed and is close to the light speed in the radio communication case (Section 3 shows the details on how propagation delays impact timeslot synchronization in a virtual bitmap). Figure 2 shows the structure of a more practical virtual bitmap. Note that a timeslot length includes the timeslot guard time in our denotation.

The timeslot length of a virtual bitmap should be as small as possible so that the minimum amount of medium time is consumed for delivering the ARQ control information. A timeslot, however, must be big enough to accommodate an ACK pulse. An ACK pulse, on the other hand, must have a size that makes it easy to generate and detect. A pulse of more than multiple microseconds is easy to generate and detect by existing hardware, as indicated by the communication speed and pulse specifications in the IEEE 802.11 standard [3].

The number of timeslots in the virtual ACK bitmap of a sender is determined by the number of neighbors that the sender has, since each receiver must have an associated timeslot. In addition, each receiver must know which timeslot it is assigned in the virtual bitmap, as detailed in the next subsection.

2.2. Timeslot Assignment and Maintenance

Bitmap timeslot assignment and maintenance are critical for the proposed ARQ scheme. With the proposed scheme, each node maintains two timeslot assignment tables. One is called the Receive (R) Table and the other is called the Send (S) Table. The R Table contains the node's own bitmap for receiving ACK bits from its neighbors, while the S Table contains the node's timeslots in its neighbors' bitmaps for the node to send ACK bits to its neighbors when necessary. The formats of the R table and S table are shown in Fig. 3. The first row of each table stores the neighbor addresses, and the second row of each table stores the timeslot assignment information for the addresses in the first row. Each pair, an address in the first row and a timeslot number at the same position in the second row, is called an assignment "entry" in the two tables.

In its initialization, a broadcast sender decides an order for its neighbors in its virtual ACK bitmap, records the order in its R table, and then broadcasts the order information (i.e., timeslot assignment information) to its neighbors in a Timeslot Assignment (TSA) frame, whose format is shown in Fig. 4. When a neighbor receives the TSA frame, it uses the position of its address in the frame to determine its timeslot in the virtual bitmap of the sender and then records the assignment as an entry in its S table. When a neighbor disappears from a node's neighbor list, the node marks the corresponding entry in the R table as empty and deletes the corresponding entry in the S table. The timeslot associated with an empty entry in the R table can be reassigned to other nodes later.

One interesting question is how a broadcast sender ensures that the timeslot assignment information successfully reaches all the intended receivers, since the information is broadcast to the receivers and thus there is also the broadcasting reliability issue. The proposed scheme must solve this bootstrapping problem before it operates correctly.

The proposed scheme solves the bootstrapping problem by using the same virtual ACK bitmap mechanism. In particular, a sender requires all its receivers to fill in a virtual ACK bitmap after they receive a TSA frame (details on how receivers fill in a virtual bitmap are shown in the next subsection). This strategy works because a receiver knows its timeslot as soon as it successfully receives the TSA frame. The sender retransmits the TSA frame until either it receives "1s" in all the timeslots in the ACK bitmap or a retransmission limit is reached. In the latter case, the entries in the R table that are associated with the "0" timeslots are marked as empty. The receivers associated with those empty entries will not be ensured for broadcasting reliability until the neighbor list is updated[1].

[1]Neighbor list maintenance is assumed as an independent function in the network.

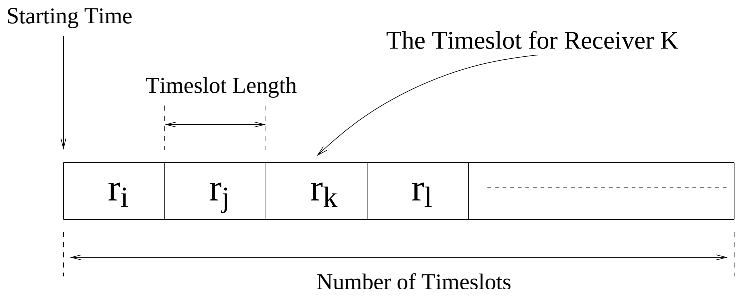


Fig. 1. The Structure of A Virtual Bitmap

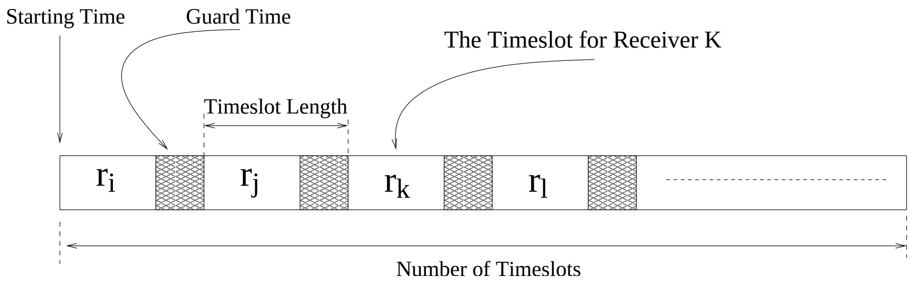


Fig. 2. The Structure of A More Practical Virtual Bitmap

NA_i	NA_j	NA_k	NA_l	– – – – – – – – – – – – – – – – –
RS_i	RS_j	RS_k	RS_l	– – – – – – – – – – – – – – – – –

R Table

NA_i	NA_j	NA_k	NA_l	– – – – – – – – – – – – – – – – –
SS_i	SS_j	SS_k	SS_l	– – – – – – – – – – – – – – – – –

S Table

NA: Neighbor Address RS: Receive Timeslot SS: Send Timeslot

Fig. 3. R Table and T Table for Timeslot Assignments

Frame Control	RA	TA	n	NA_0	NA_1	NA_2	NA_3	– – – – – – – – – – – –	FCS

RA: Receiver Address TA: Transmitter Address NA: Neighbor Address

n: Number of NAs FCS: Frame Check Sequence

Fig. 4. Timeslot Assignment Frame Format

When the neighbor list of a broadcast sender changes, the broadcast sender needs an efficient way to maintain the timeslot assignments. In particular, when a node disappears from the neighbor list, the associated entry in the R table becomes empty. When new nodes are added to the neighbor list, they may be assigned the empty timeslots in the R table or new timeslots need to be created for them in the R table. By maintenance we mean the reassignments of empty timeslots and the creation of new timeslots in the R table.

Although it would be straightforward to use TSA frames too for the maintenance, it is not efficient because the addresses of all the receivers have to be still included in a TSA frame. A possible solution is to add a one-byte space behind each address in the TSA frame to deliver the timeslot number for the address. In such a case, only the addresses that are involved in the assignment maintenance need to be included in a TSA-like frame. The proposed scheme uses a similar approach but of higher efficiency, as introduced below.

With the proposed scheme, a broadcast sender maintains the timeslot assignments in its virtual ACK bitmap only when there are broadcast packets to send. In addition, the sender piggybacks its timeslot maintenance information with the data packets in broadcast frames. For the piggyback purpose, timeslot assignment fields are added into a broadcast data frame between the Transmitter Address (TA) field and the Frame Body field, as shown in Fig. 5. The "n" field is to indicate the number of timeslot assignment pairs following it, and each pair is for the timeslot assignment for one receiver. This maintenance mechanism is feasible because when a receiver needs to send back an ACK pulse, it has correctly received the broadcast frame including the packet and the timeslot maintenance information.

The timeslot maintenance method used by the proposed scheme is efficient for three reasons. First, the timeslot assignments are maintained only when there are broadcast traffic. Second, the timeslot maintenance information does not introduce additional frame overhead such as preambles and headers. Third, no separate ACK information is needed for the timeslot maintenance information.

2.3. The Complete ARQ Scheme

This section presents the complete ARQ scheme. After the initialization process introduced earlier, nodes have their R tables and S tables ready. When a node has a broadcast packet to send, it first conducts medium contention with a MAC protocol such as the IEEE 802.11 DCF. After the sender is successful in a medium contention, it starts to broadcast the packet. Upon finishing transmitting the packet, the sender starts a timeslot timer whose delay is set to T, which is the length of a timeslot including the guard time. After that, the sender waits for a pulse in the current timeslot.

On the other hand, all receivers that have successfully received the broadcast packet start to prepare for filling in their timeslots in the virtual ACK bitmap of the sender. In particular, as soon as finishing receiving the broadcast packet, each such receiver starts an ACK timer whose delay is determined by examining the T table. For example, if receiver i is assigned the m^{th} timeslot[2] in the sender's bitmap, the delay of receiver i's ACK timer is then set to mT. As soon as the ACK timer of a receiver expires, the receiver sends its ACK pulse.

If the sender detects a pulse in its current timeslot, a "1" is recorded for the timeslot in the ACK bitmap. Otherwise, a "0" is recorded. Upon the expiration of its timeslot timer, the sender restarts its timeslot timer with a delay of T, the timeslot length. The sender then waits for the ACK pulse from the next receiver. The sender repeats this process until all the timeslots in its virtual ACK bitmap have been exhausted.

As soon as the timeslots are exhausted in its virtual ACK bitmap, the sender examines the recorded ACK bitmap. If the bitmap has all "1s", the sender will proceed to the next broadcast packet. Otherwise, the sender will retransmit the current packet upon the next successful medium contention. For a retransmitted packet, the sender only examines those "0" timeslots for pulses. However, receivers receiving duplicate packets still send back ACK pulses in their timeslots, since the previous ACK pulses may have been lost. There is a limit on the number of retransmissions of a packet, which may be set according to the delay requirement of a specific video streaming application. When the limit is reached, the packet will be discarded and the sender proceeds to the next packet.

[2]The first timeslot is the 0^{th} timeslot in our denotations.

Note that video streaming can, in general, tolerate some packet losses. The proposed ARQ scheme is at the MAC sub-layer and does not ensure full reliability. For example, a noise pulse in a timeslot or the retransmission limit may cause a receiver to fail to receive a broadcast packet. If a video streaming application needs full reliability, it must be achieved at the transport layer instead of the MAC sub-layer. However, even in such a case, the improved reliability at the MAC sub-layer with the proposed ARQ scheme will greatly reduce the whole cost of maintaining the full data reliability for such applications.

3. Timeslot Synchronization Analysis

After the preceding section presents the details of the proposed ARQ scheme for video broadcasting in wireless LANs, this section investigates the synchronization of the timeslots in a virtual ACK bitmap of the proposed scheme. By synchronization we mean that the pulses of two receivers assigned adjoining timeslots in a virtual ACK bitmap fall into their own timeslots and do not overlap with each other.

The proposed scheme uses timers to coordinate the receivers of a broadcast sender for them to find their timeslots in the virtual ACK bitmap of the sender. As mentioned earlier, all receivers that intend to send back ACK pulses have successfully received the broadcast packet, so they can agree on the starting time of the virtual ACK bitmap, which is the time when they finish receiving the broadcast packet. The interference factor is, however, that the broadcast packet usually experiences different propagation delays before it reaches different receivers. Therefore, the bitmap may not have exactly the same starting time at the sender and its different receivers.

To analyze how propagation delays may affect the timeslot synchronization among the sender and its receivers, we take a look at three receivers that are assigned three adjoining timeslots in a virtual bitmap, as illustrated in Fig. 6. Particularly, receivers R_j, R_k and R_l are assigned the $(n-1)^{th}$, n^{th} and $(n+1)^{th}$ timeslots in the bitmap, respectively. We may assume that the propagation delays from the sender to R_j, R_k and R_l are τ_j, τ_k and τ_l, respectively. *Note that although Fig. 6 assumes $\tau_j > \tau_k > \tau_l$, the following analysis does not have the assumption.* If the bitmap starts at T_0 at the sender, then the bitmap starts at times $T_0 + \tau_j$, $T_0 + \tau_k$ and $T_0 + \tau_l$ at receivers R_j, R_k and R_l, respectively. However, R_k's pulse must fall into R_k's timeslot in each of the *four* bitmaps viewed by the sender S, the two nodes R_j and R_l that own the two neighboring timeslots, and itself. In such a case, R_k's pulse will not overlap with R_j's or R_l's pulse and will be correctly interpreted by the sender S.

In the sender S's view, the timeslot for R_k is from t_1 to t_2, where

$$t_1 = T_0 + n \times T \tag{1}$$

$$t_2 = T_0 + (n+1) \times T \tag{2}$$

where T is the timeslot length including the guard time (note that the first timeslot is the 0^{th} timeslot in our denotations). In R_l's view, the timeslot for R_k is from t_3 to t_4, where

$$t_3 = T_0 + n \times T + \tau_l \tag{3}$$

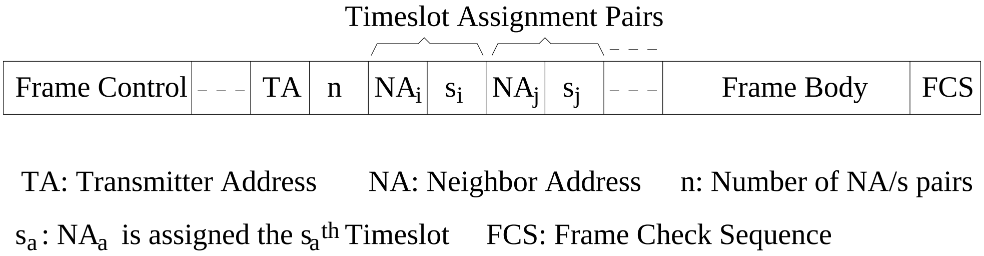


Fig. 5. Timeslot Assignment Fields in a Data Frame

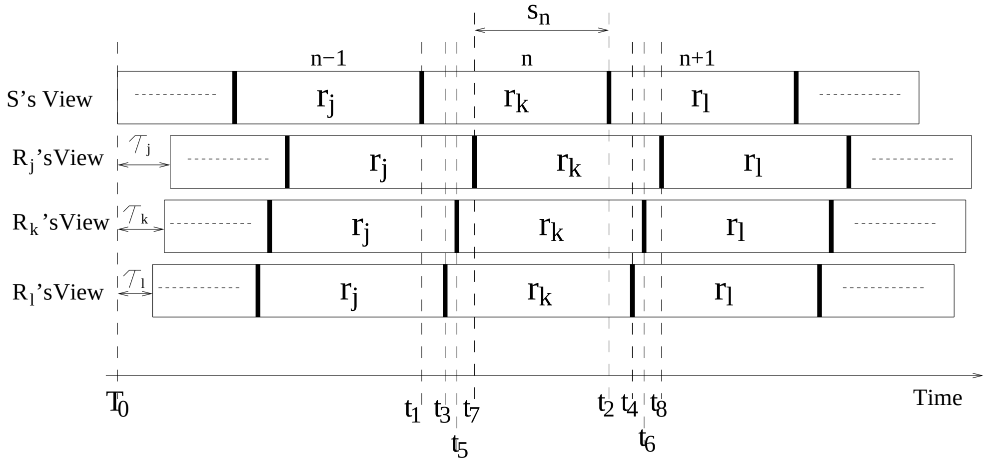


Fig. 6. Impact of Propagation Delays on Timeslot Synchronization

$$t_4 = T_0 + (n + 1) \times T + \tau_l \tag{4}$$

In R_k's own view, its timeslot is from t_5 to t_6, where

$$t_5 = T_0 + n \times T + \tau_k \tag{5}$$

$$t_6 = T_0 + (n + 1) \times T + \tau_k \tag{6}$$

In R_j's view, however, the timeslot for R_k is from t_7 to t_8, where

$$t_7 = T_0 + n \times T + \tau_j \tag{7}$$

$$t_8 = T_0 + (n + 1) \times T + \tau_j \tag{8}$$

Since R_k's pulse must fall into each of the four R_k's timeslots viewed by the sender S, the two nodes R_j and R_l, and itself, we need to find the *intersection* of the four timeslots. The four timeslots are from $t1$ to $t2$, from $t3$ to $t4$, from $t5$ to $t6$, and from $t7$ to $t8$, respectively. Therefore,

$$s_n = [t_1\ t_2] \cap [t_3\ t_4] \cap [t_5\ t_6] \cap [t_7\ t_8] \tag{9}$$

where s_n is the actual n^{th} timeslot that can be used by R_k's pulse.

To accommodate R_k's pulse, the length of s_n, $|s_n|$, must be greater than or equal to the length of R_k's pulse plus its propagation delay, i.e., $|s_n| \geq \lambda + \tau_k$, where λ is the pulse length. From Eq. 9, we have $T \geq |s_n|$. Thus, if $|s_n| \geq \lambda + \tau_k$, then $T \geq \lambda + \tau_k$. To meet the condition of $|s_n| \geq \lambda + \tau_k$, we only need to meet the following conditions:

$$t_1 \leq \min(t_4,\ t_6,\ t_8) - \lambda - \tau_k \tag{10}$$

$$t_3 \leq \min(t_2,\ t_6,\ t_8) - \lambda - \tau_k \tag{11}$$

$$t_5 \leq \min(t_2,\ t_4,\ t_8) - \lambda - \tau_k \tag{12}$$

$$t_7 \leq \min(t_2,\ t_4,\ t_6) - \lambda - \tau_k \tag{13}$$

After solving Equations 10, 11, 12 and 13, we have

$$T \geq \max(\tau_j,\ \tau_k,\ \tau_l) + \lambda + \tau_k \tag{14}$$

Equation 14 gives the timeslot length that is required for the proper timeslot synchronization for R_k.

If Equation 14 is met, we will be able to find the beginning time, $t_{s_n 1}$, and the end time, $t_{s_n 2}$, for s_n. In particular,

$$t_{s_n 1} = \max(t_1,\ t_3,\ t_5,\ t_7) \tag{15}$$

$$t_{s_n 2} = \min(t_2,\ t_4,\ t_6,\ t_8) \tag{16}$$

From Eqs. 1, 3, 5, and 7, $t_1 < \min(t_3, t_5, t_7)$, so Eq. 15 becomes

$$t_{s_n 1} = \max(t_3,\ t_5,\ t_7) \tag{17}$$

Similarly, from Eqs. 2, 4, 6, and 8, $t_2 < \min(t_4, t_6, t_8)$, so Eq. 16 becomes

$$t_{s_n 2} = t_2 \tag{18}$$

Now with Eqs. 17 and 18, we can find the guard time needed for s_n in R_k's view. If we denote the guard time at the beginning of the timeslot as τ_{g_1} and that at the end of the timeslot as τ_{g_2}, then

$$\begin{aligned}\tau_{g_1} &= t_{s_n1} - t_5 \\ &= \max(t_3,\ t_5,\ t_7) - t_5 \\ &= \max(\tau_j,\ \tau_k,\ \tau_l) - \tau_k \end{aligned} \quad (19)$$

$$\tau_{g_2} = t_6 - t_{s_n2} = t_6 - t_2 = \tau_k \quad (20)$$

As show by Eq. 19, the guard time needed at the beginning of a timeslot is determined by the propagation delays of three nodes, which are the owner of the timeslot and the two nodes owning the two neighboring timeslots. However, as shown by Eq. 20, the guard time at the end of a timeslot is determined only by the propagation delay of the owner of the timeslot.

Moreover, when the propagation delay to a node is greater than the propagation delays to the two nodes owning the two neighboring timeslots, the node's timeslot needs no guard time at the beginning of the timeslot. If the propagation delay to a node is smaller than the propagation delay to one of the two nodes owning the two neighboring timeslots, then some guard time is needed at the beginning of the node's timeslot. Figure 6 shows a case where $\tau_j > \tau_k > \tau_l$. In such a case, the beginning time of R_k's timeslot is pushed backward by R_j's view of the bitmap, as shown in the figure. In particular, the beginning time of R_k's timeslot is determined by t_7, which is the beginning time of R_k's timeslot in R_j's view. On the other hand, the end time of R_k's timeslot is pushed forward by the sender's view of the bitmap. In particular, t_2 determines the end time of R_k's timeslot.

Note that Eq. 14 only gives the timeslot length required for the proper timeslot synchronization for one receiver, which is R_k. If all receivers of the sender are considered, Eq. 14 basically indicates that the timeslot length must be greater than or equal to two times of the maximum propagation delay plus the pulse length, i.e.,

$$T \geq 2\max(\tau_0,\ \tau_1,\ \tau_2, \ldots) + \lambda \quad (21)$$

Finally, the propagation delay, say τ, from a sender to one of its receivers is bounded by the transmission range of the sender, say T_{tx}. In particular, $\tau \leq \frac{T_{tx}}{c}$ where c is the signal propagation speed. Therefore, Eq. 21 becomes

$$T \geq \frac{2T_{tx}}{c} + \lambda \quad (22)$$

and the guard time for a timeslot is

$$\tau_g = |T - \lambda| \geq \frac{2T_{tx}}{c} \quad (23)$$

4. Simulation Results

This section presents the comprehensive simulation results obtained by using the ns-2 simulator [20]. For easy denotation, the proposed scheme is named Multiple-receiver ARQ (MARQ). For comparison, simulation results are also included for the ARQ scheme in BMMM [7], which is a broadcast ARQ scheme using traditional control frames[3].

One-hop, saturation traffic is assumed in the simulations. In particular, each node always has broadcast packets to send and the packets are for its neighbors. In addition, static networks are used in the simulations. Broadcast ARQ schemes are designed to ensure the broadcast reliability to the nodes in a sender's neighbor list. The maintenance of the neighbor list of a node is, however, usually not a part of a broadcast ARQ scheme. To exclude the interference factor of possibly imprecise neighbor lists, static networks are used in the simulations.

However, the scheme control overhead that is related to neighbor list changes should be included in the scheme evaluations for the ARQ schemes. For such a purpose, the MARQ scheme is also simulated for the worst case (W.C.) scenario where all timeslots of an ARQ bitmap need to be reassigned for each broadcast packet. In other words, in the W.C. scenario, each broadcast packet always carries the addresses of all the nodes in the neighbor list of the sender (details in Section 2), which introduces the highest possible overhead for the MARQ scheme in practice.

There are some other configuration details. CSMA [3] is used for medium access control in the simulations and RTS/CTS frames are never used. The contention window (CW) of each node is adjusted dynamically. In particular, whenever a node needs to *retransmit* a packet, it doubles its CW. When a packet is successfully delivered to all the intended receivers or when a packet is discarded due to excessive retransmissions, the CW is reset to its minimum size. All broadcast packets have a size of 512 bytes and the transmission rate is 1Mb/s in the LAN of a size of 1000 by 1000 meters. In addition, the timeslot length in a virtual ARQ bitmap in MARQ is $35\mu s$, which includes a guard time of $10\mu s$. Finally, if not specified otherwise, default ns-2.30 configurations are used in the simulations at all layers.

Figure 7 shows the packet delivery ratio against the number of nodes in the wireless LAN for the case where the packet retransmission limit is four (i.e., the default configuration in ns-2). The packet delivery ratio is calculated at each node and what shown in the figure is the average over all the nodes in the network. As shown in Fig. 7, when there are less than ten nodes in the LAN, one hundred percent of the broadcast packets reach all the receivers. However, when the number of nodes in the LAN reaches ten, the packet delivery ratio is below one hundred percent with both ARQ schemes due to high packet collision rate.

Figure 8 shows the node throughput against the number of nodes in the LAN for the same case of a packet retransmission limit of four. The node throughput is calculated at each node and the figure shows the average over all the nodes in the network. As shown in Fig. 8, the average node throughput with MARQ has gains of from several tens to more than one hundred percent as compared with that with BMMM. MARQ has higher relative throughput gains over BMMM when there are more nodes in the LAN, as expected. These throughput results show that MARQ introduces much less ARQ control overhead than BMMM does in the network.

Figure 9 shows the packet delivery ratio against the number of nodes in the LAN for the case where the packet retransmis-

[3]The RAK frames carrying the NAV information and RTS/CTS frames are not used in the simulations.

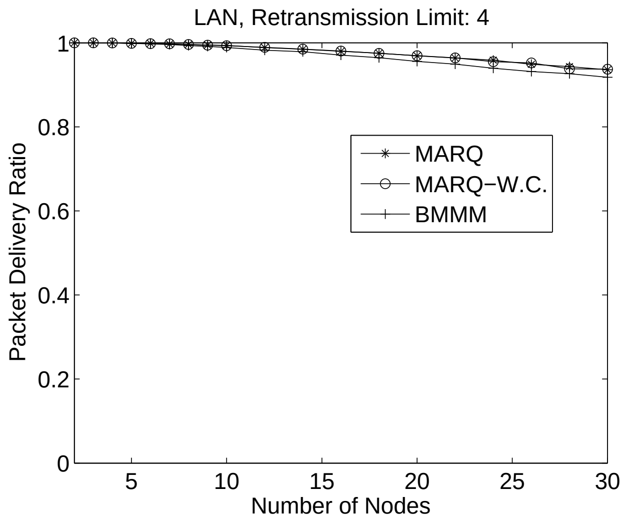


Fig. 7. Packet Delivery Ratio vs. Number of Nodes

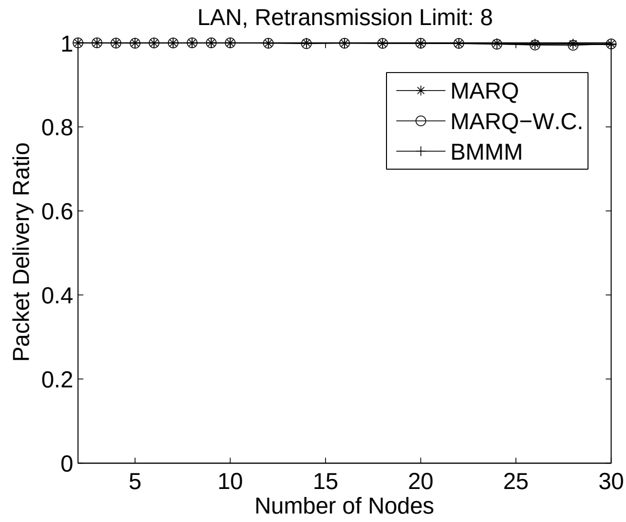


Fig. 9. Packet Delivery Ratio vs. Number of Node

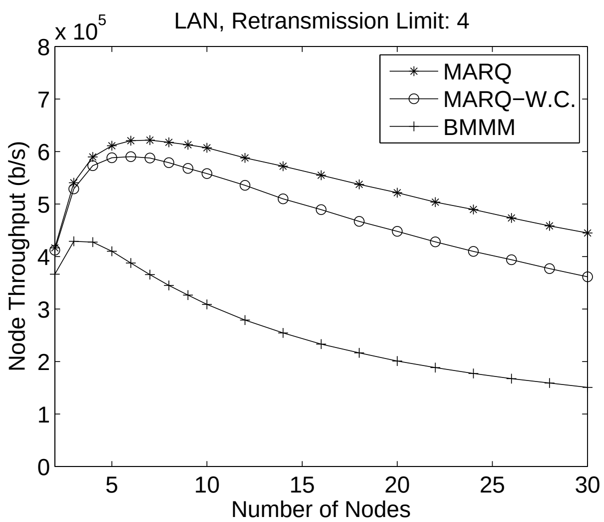


Fig. 8. Node Throughput vs. Number of Nodes

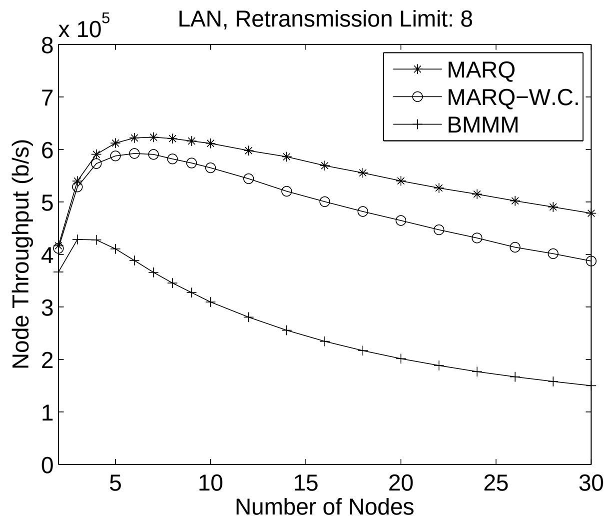


Fig. 10. Node Throughput vs. Number of Nodes

sion limit increases from four to eight. As shown in the figure, all packets reach all the receivers successfully in this case of a higher packet retransmission limit. Similarly, as shown in Fig. 10, MARQ doubles the node throughput of BMMM when there are a significant number of nodes in the LAN.

One interesting thing shown by the comparison of Fig. 8 with Fig. 10 is that after the packet retransmission limit increases from four to eight, the node throughput may increase instead of decreasing. Further investigation found that this happens because the increase of the retransmission limit enables a node to adjust its CW size in a larger range. When a node is able to adjust its CW size in a larger range, the average CW size of the node increases and thus the collision rate in the LAN decreases.

Figure 11 shows the average CW size of a node against the number of nodes in the LAN. The CW size is averaged over all packets at a node and is then averaged over all nodes in the LAN. The CW size associated with a packet is the CW size when the packet is either successfully delivered to all the intended receivers or is discarded due to excessive retransmissions. As shown in Fig. 11, the average CW size of a node increases when the packet retransmission limit increases from four to eight. The packet collision rate in the LAN is thus reduced when the packet retransmission limit increases from four to eight, as shown in Fig. 12. The retransmission ratio shown in Fig. 12 is the ratio of the number of *retransmissions* at a node over the total number of packet transmissions at the node in the simulations.

5. Conclusion

This paper proposes an efficient ARQ scheme, the MARQ scheme, for reliable video broadcasting in wireless LANs. The MARQ scheme is an in-band scheme that does not use expensive out-of-band control signals. However, unlike existing in-band ARQ schemes for reliable broadcasting, the MARQ scheme is highly efficient in its ARQ control information delivery due to its use of a new approach of virtual ACK bitmaps. In particular, in a fully distributed way, a sender allocates a timeslot in its virtual ACK bitmap for each of its receivers and a receiver uses

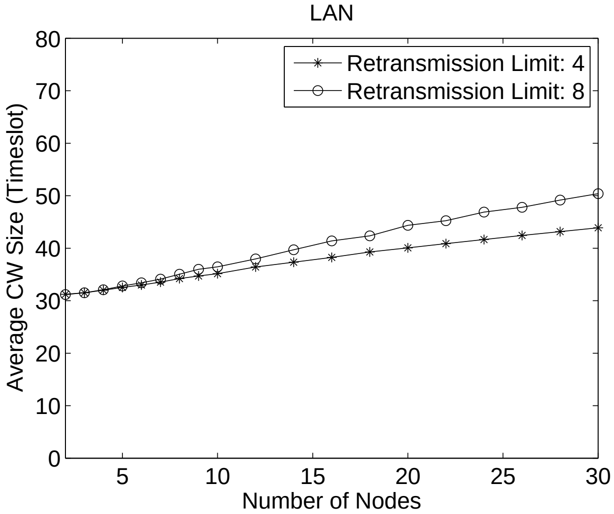


Fig. 11. Average CW Size vs. Number of Nodes

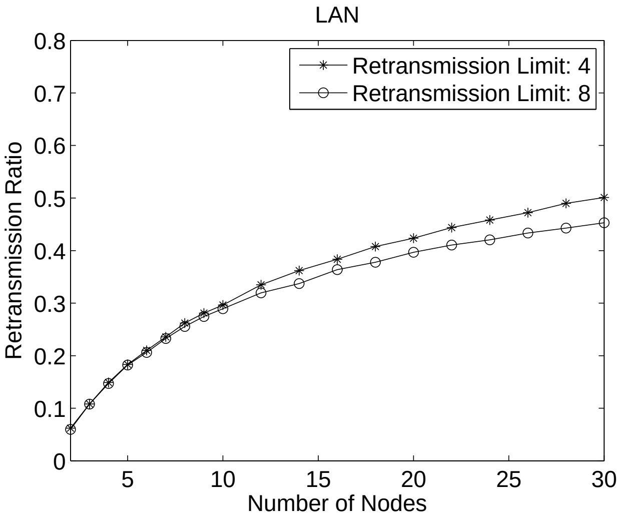


Fig. 12. Retransmission Ratio vs. Number of Nodes

its allocated timeslot to send back an in-band ACK pulse when it successfully receives a packet. Our comprehensive simulation results show that the MARQ scheme is highly efficient. In particular, as compared with existing ARQ schemes using traditional control frames, MARQ has throughput gains from several tens to more than one hundred percent.

References

[1] J. Peng, "Slim ARQ for Reliable Broadcasting in Wireless LANs," in *IEEE WCNC 2008*, Las Vega, Nevada, March 2008.

[2] ——, "A New ARQ Scheme for Reliable Broadcasting in Wireless LANs," *IEEE Communications Letters*, vol. 12, no. 2, pp. 146–148, February 2008.

[3] IEEE 802.11 wireless local area networks. [Online]. Available: http://grouper.ieee.org/groups/802/11/

[4] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part II - the hidden terminal problem in carrier sense multiple access and the busy tone solution," *IEEE Transactions on Communications*, vol. 23, pp. 1417–1433, 1975.

[5] K. Tang and M. Gerla, "MAC reliable broadcast in ad hoc networks," in *The MILCOM*, 2001.

[6] J. Kuri and S. K. Kasera, "Reliable multicast in multiaccess wireless LANs," *Wireless Networks*, vol. 7, pp. 359–369, 2001.

[7] M. T. Sun, L. Huang, A. Arora, and T. H. Lai, "MAC layer multicast in IEEE 802.11 wireless networks," in *The International Conference on Parallel Processing (ICPP)*, 2002.

[8] A. Colvin, "CSMA with collision avoidance," *Computer Commun.*, vol. 6, pp. 227–235, 1983.

[9] P. Karn, "MACA - a new channel access method for packet radio," in *Proc. of the 9th ARRL Computer Networking Conference*, Ontario, Canada, 1990.

[10] R. Benice and A. Frey, "An Analysis of Retransmission Systems," *IEEE Transactions on Communications*, vol. 12, pp. 135–145, Dec 1964.

[11] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: a medium access protocol for wireless LANs," in *Proc. of the ACM SIGCOMM*, London, United Kingdom, August 1994.

[12] K. Xu, M. Gerla, and S. Bae, "How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks?" in *Proc. of the IEEE GLOBECOM*, Taipei, Taiwan, November 2002.

[13] J. Peng and L. Cheng, "Exploring a new approach to collision avoidance in wireless ad hoc networks," in *IFIP Networking 2007*, Atlanta, GA, May 2007.

[14] S. Gupta, V. Shankar, and S. Lalwani, "Reliable Multicast MAC Protocol for Wireless LANs," in *The IEEE ICC*, 2003.

[15] W. Si and C. Li, "RMAC: A reliable multicast MAC protocol for wireless ad hoc networks," in *The IEEE ICPP*, 2004.

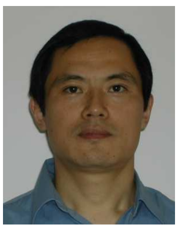
[16] G. A. Halls, "HIPERLAN: the high performance radio local area network standard," *Electronics and Communication Engineering Journal*, vol. 6, pp. 289–296, Dec. 1994.

[17] J. Peng, "A New Scheme to Avoid Collisions for Broadcast Packets in Wireless LANs," *IEEE Communications Letters*, vol. 11, no. 9, pp. 762–764, September 2007.

[18] J. L. Sobrinho and A. S. Krishnakumar, "Real-time traffic over the IEEE 802.11 medium access control layer," *Bell Labs Technical Journal*, pp. 172–187, 1996.

[19] A. Chen, G. Chandrasekaran, D. Lee, and P. Sinha, "HIMAC: High Throughput MAC Layer Multicasting in Wireless Networks," in *IEEE MASS*, Oct. 2006.

[20] The network simulator - ns-2. [Online]. Available: http://www.isi.edu/nsnam/ns/



Jun Peng received his Ph.D. degree in electrical engineering from the Electrical, Computer, and Systems Engineering Department at Rensselaer Polytechnic Institute (RPI), Troy, NY, USA in 2004. He is currently an assistant professor at the Electrical Engineering Department of University of Texas - Pan American, Edinburg, TX, USA. His general research interests are in computer communications networks. He has contributed to the areas of medium access control, congestion control, error control, multicasting, and broadcasting. He has recently been conducting research in the general area of distributed wireless networks. He is a senior member of IEEE.