A Multirate MAC Protocol for Reliable Multicast in Multihop Wireless Networks

Claudia Campolo[a], Claudio Casetti[b], Carla-Fabiana Chiasserini[b], Antonella Molinaro[a]

[a]*Dipartimento DIMET, Università Mediterranea di Reggio Calabria*
Loc. Feo di Vito, I-89060, Reggio Calabria (Italy)
e-mail: firstname.lastname@unirc.it
[b]*Dipartimento di Elettronica, Politecnico di Torino*
Corso Duca degli Abruzzi 24, I-10129 Torino (Italy)
e-mail: lastname@tlc.polito.it

Abstract

Many multicast applications, such as audio/video streaming, file sharing or emergency reporting, are becoming quite common in wireless mobile environment, through the widespread deployment of 802.11-based wireless networks. However, despite the growing interest in the above applications, the current IEEE 802.11 standard does not offer any medium access control (MAC) layer support to the efficient and reliable provision of multicast services. It does not provide any MAC-layer recovery mechanism for unsuccessful multicast transmissions. Consequently, lost frames cannot be detected, hence retransmitted, causing a significant quality of service degradation. In addition, 802.11 multicast traffic is sent at the basic data rate, often resulting in severe throughput reduction.

In this work, we address these issues by presenting a reliable multicast MAC protocol for wireless multihop networks, which is coupled with a lightweight rate adaptation scheme. Simulation results show that our schemes provide high packet delivery ratio and, when compared with other state-of-the-art solutions, they also provide reduced control overhead and data delivery delay.

Keywords: 802.11; multicast; reliable; rate adaptation

1. Introduction

Recently, group-oriented services have appeared as one of the primary application classes targeted for wireless multihop networks. Examples abound in both civilian and military applications: in the former, users who share the same interests can form on-demand communities, e.g., for the purpose of file sharing, multi-player online gaming, audio/video streaming, video conferencing; in the latter, nodes working in groups strive toward a common goal, for mission-critical tasks, information dissemination, emergency reporting in rescuing operations.

Multicasting supports data dissemination to a group of hosts, therefore it plays a crucial role as enabler of the above set of applications. Through it, all intended receivers are reached by a single transmission, thus lowering network costs and improving channel efficiency with respect to unicast transmissions to individual group members.

Multicasting has been extensively studied, especially at the transport and routing layers. While multicasting at the transport layer mainly concerns issues on error recovery, routing-layer multicasting focuses on building a tree or a mesh topology, maintaining them in case of mobility of the nodes, ensuring energy efficiency, as extensively surveyed in [1]. Besides, although most network layer multicast protocols work with any MAC scheme, their effectiveness in terms of reliability can be improved by providing a reliable underlying MAC layer. Nevertheless, the salient features of wireless networks, such as random packet drops due to mobility, fading, external interference, bandwidth scarcity, coupled with the lack of network infrastructure in multihop ad hoc networks, make the provision of reliable, efficient, fast MAC-layer multicasting extremely challenging.

Indeed, the current IEEE 802.11 technology [2], based on carrier sense multiple access with collision avoidance (CSMA/CA), has no explicit mechanism to support multicasting at the MAC layer: multicast packets are generally forwarded as one hop broadcast in order to reach all group members in the neighborhood with a single transmission. Thus, two major problems arise when legacy 802.11 is used. Firstly, no MAC-layer recovery for multicast frames is provided: the use of a handshaking procedure, such as request-to-send/clear-to-send (RTS/CTS) frames, or of acknowledgments (ACKs) is not allowed, thus unsuccessfully transmitted packets are never retransmitted. This issue is of particular relevance since, in multihop networks, a transmission failure at any of the forwarding nodes may cause the loss of the packet at any receiver in the downstream multicast tree. Secondly, to ensure high reception probability, multicast traffic is always sent at the basic data rate (i.e., the lowest data rate available in a network using an IEEE 802.11 technology); this often results in severe throughput degradation, especially for multicast streaming.

To solve these shortcomings in 802.11-based multihop networks, various multicast MAC protocols have been recently proposed with the aim to enhance the reliability and efficiency of multicasting. Typically, simple extensions to the IEEE 802.11 Distributed Coordination Function (DCF), which have appeared in the literature, target only reliability, resulting in high control overhead, collisions among the RTS/CTS frames and failure in solving the hidden terminal problem [3, 4]. On the contrary, efficient protocols that are not compliant with the IEEE 802.11 standard are not easily implemented in practical devices [5, 6, 7]. However, to the best of our knowledge, none of the above mentioned approaches is able to simultaneously cope with reliability and efficiency issues, by also ensuring high throughput values. It follows that a good compromise between these two trends needs to be found.

In this work, we design a fully-distributed reliable and efficient multicast MAC protocol, which introduces as few modifications as possible with respect to the IEEE 802.11 MAC specifications and is suitable for infrastructured, multihop ad hoc and mesh network topologies. We name our scheme *RM^3, Reliable Multirate Multicast MAC*. RM^3 does not rely on any centralized entity that instructs feedback transmissions by multiple receivers or handles them on behalf of other receivers. In addition, it relies on a rate adaptation mechanism, which avoids the indiscriminate use of the basic data rate and allows a transmitter to send traffic at the highest data rate that is acceptable to all multicast receivers. This clearly increases the overall network throughput. Finally, RM^3 is able to ensure high packet delivery ratio as well as reduced data delivery delay and control overhead.

The remainder of the paper is organized as follows. Section 2 describes the basic principles of the 802.11 DCF scheme, while Section 3 discusses related work. In Section 4, we provide an overview of our solution for multirate reliable multicasting. In Section 5, a detailed description of the transmission scheme of RM^3 is given. The rate adaptation mechanism and the NAV updating procedures are described in Section 6 and Section 7, respectively. Section 8 compares through ns2 simulations the performance of RM^3 and RAMP, our early proposal in [8], against the legacy 802.11 and the MMP protocol presented in [9]. Finally, Section 9 concludes the paper.

2. Background: The IEEE 802.11 DCF Scheme

In 802.11 networks [2] time is divided into time intervals called *time slots* and the channel access is based on the CSMA/CA mechanism. DCF exploits both a physical and a virtual channel sensing, to determine whether the channel is idle or busy. Virtual sensing is implemented by including in all transmitted frames an indication of their duration so that the non-destination stations overhearing a transmission are aware of the time interval during which the channel will remain busy. A counter, called NAV (Network Allocation Vector), is set accordingly to keep track of the channel status. Once a station has set its NAV, it remains in overhearing state for the whole duration of the transmission.

When a station wishes to access the channel, the physical and virtual carrier sense mechanisms are checked. If both of them detect the channel as idle for a time duration equal to DIFS (Distributed Inter Frame Space) seconds, the node transmits. Otherwise, the station waits for the channel to become idle; then, within an interval of DIFS, it randomly selects a backoff value from the range called Contention Window (CW) and sets its backoff counter to this value times the slot duration. The value of contention window is doubled at every failed transmission attempt. The backoff counter is decremented at the end of each idle slot; as the backoff counter reaches zero, the node accesses the channel.

For unicast transmissions, correctly received data frames are acknowledged by the intended destination by sending an ACK frame after SIFS (Short Inter Frame Space) seconds from the data frame reception.

To increase the reliability level of unicast transmissions and avoid the hidden terminal problem, the access scheme described above can include a control frame handshaking between sender and receiver. In this case, the sender first transmits an RTS frame and sends the data frame only after it receives a CTS frame from the intended destination. Once the handshaking takes place successfully, both the sender's and the receiver's neighbors are informed about the upcoming data transmission and will refrain from accessing the channel, thus avoiding collisions.

The current IEEE 802.11 technology has no explicit mechanism to support multicasting at the MAC layer: multicast packets, regardless of their length, are generally forwarded as one-hop broadcast in order to reach all the receivers in the neighborhood with a single transmission. RTS/CTS cannot be used for sending broadcast/multicast frames and such frames are never acknowledged. Thus, lost frames cannot be detected, hence retransmitted. To compensate for the inherent unreliability of such an approach, packets are transmitted at the basic data rate (e.g., typically 1 Mbps for 802.11b), also when a much higher rate could be acceptable to the multicast receivers.

3. Related Work

Since our aim is to design a rate-adaptive reliable and efficient MAC protocol for multicasting, we first discuss some of the main solutions proposed in the literature regarding both reliability and efficiency in 802.11-based networks. Then, we review some multicast MAC protocols that provide adaptation of the data rate.

Reliable and Efficient Multicast MAC Protocols. To solve the shortcomings of 802.11-based networks in supporting multicasting, some early schemes [3, 4] extended the basic 802.11 control mechanisms used for unicast frames, such as RTS/CTS and ACK, to broadcast/multicast transmissions. However, these solutions were unable to coordinate the transmission of CTS frames from multiple receivers, which makes collisions among CTS frames highly probable. To avoid CTS collisions, later proposals either make receivers send their CTS frame at different time instants [10, 11] or replace these frames with one CTS frame sent by a leader node [12]. The result is an increased reliability level in multicast delivery, although these solutions still suffer from the hidden terminal problem, thus often incurring in excessive delay or inefficiency in channel utilization.

To counteract the hidden terminal problem, in [5, 6, 7] busy tones are used to signal negative ACK/negative CTS; however, such an approach requires additional transceivers in the wireless devices to handle the busy tones.

A low-overhead solution is proposed in [13]. When a multicast member receives a data packet from the sender, it allocates a symbol on a pre-assigned Orthogonal Frequency Division Multiple Access (OFDMA) subcarrier, which acts as a positive/negative ACK for the packet. Thus, after SIFS, all receivers can send simultaneously their feedback and the scheme incurs the same overhead as in unicast transmissions. However, since the virtual carrier sensing procedure is not deployed, the probability of collisions due to hidden terminals may still be high; furthermore, the protocol lacks of a proper retransmission policy for failed multicast transmissions.

Two interesting solutions, working in 802.11-based ad hoc environments and not requiring additional hardware are in [9, 14]. The multicast-aware MAC protocol (MMP), proposed in [9], adds an Extended Multicast Header (EMH) to each packet, reporting the identifier of the next-hop neighbors that are supposed to receive the packet and send back an ACK, following their order of appearance in the EMH. If not all of the expected ACKs are received, the sender retransmits the packets by using a handshake mechanism, where the RTS frame reports the identifiers of the nodes which did not send the ACK, and a CTS is expected from each of them, again following their order of appearance in the RTS frame. The solution in [9], although ensuring high packet delivery ratio, implies a significant overhead because the packet size is increased to carry all next-hop MAC addresses. To overcome this problem, the scheme in [14] exploits RTS/CTS frames similarly to [9], but limits the number of next-hop neighbors to four. It also complements the data packet header with the indication of which next-hop neighbors have sent a CTS, so that neighboring nodes can set their NAV accordingly. Finally, an early version of our solution, the so-called RAMP scheme, was presented in our conference paper [8]. RAMP implements the same transmission procedures as RM^3, however it does not implement any rate adaptation for multicast data transmission and always uses the basic data rate.

Multirate Multicast MAC Protocols. Since the IEEE 802.11 physical layer supports multi-rate transmissions, several unicast protocols have been proposed to exploit this capability. In [15], senders increase the data rate after consecutive successful transmissions and reduce the rate after consecutive transmission failures. In [16], the proposed Receiver Based Auto Rate (RBAR) protocol lets the receiver measure the perceived channel quality, decide the transmission rate, and then notify it to the sender before the data packet transmission.

In multicast communications, each intended receiver may experience different (and variable) channel states, thus making rate adaptation at the source a challenging task. Only few works exist that tackle this topic.

Rate Adaptive Multicast (RAM) [17] exploits RTS/CTS control frames to allow multicast receivers to perform channel estimation and rate selection. The sender node transmits an RTS frame and the members of the multicast group measure the Received Signal Strength (RSS) of the received RTS frame, depending on which they choose a suitable data rate. Then, all multicast receivers simultaneously send a variable length dummy CTS frame, whose length corresponds to a selected data rate. According to the duration of the CTS transmissions, the multicast sender can predict the lowest data rate to use for transmitting the data frame.

A similar approach is followed in HIMAC [18]. Although the RAM and the HIMAC protocols can ensure high throughput by enabling the sender to transmit at the maximum achievable data rate among the receivers, they cannot ensure full reliability. Indeed, the duration of the data packet transmission cannot be advertised before the data transmission starts, thus the nodes that are hidden to the sender cannot set their NAV.

To the authors' knowledge, no solution has been recorded in the literature that is able to provide reliable multicast communications while ensuring high throughput. Thus, by drawing on the solutions presented in [9] and [14] targeting full reliability, and by borrowing the main features of the receiver-based rate adaptation scheme proposed in [16] for unicast transmissions, we design a rate-adaptive protocol which aims at providing high throughput and reliability of multicast communications in multihop 802.11 wireless networks.

4. Outline of RM^3

We consider a wireless multihop network and assume that MAC-layer multicasting is managed as a mapping of network-layer multicasting. This implies that the routing protocol, which handles group membership by addition/removal of nodes to/from a multicast group, has the additional task of identifying the next-hop nodes of a multicast transmission through their network layer addresses, prior to mapping them into their MAC layer addresses.

Based on these assumptions, we devise the RM^3 solution with the aim of improving the packet delivery ratio while curbing control overhead and packets' delay.

The key points of the proposed RM^3 scheme are that (i) sender and receivers carry out an *efficient*, bare-bone handshaking procedure to ensure reliable multicast data delivery; (ii) the handshaking procedure is enhanced in such a way as to let the sender choose the optimal data rate to transmit multicast packets, by learning the current channel state at all receivers; (iii) shortened node identifiers are used instead of full-length node addresses to save bandwidth and reduce control overhead, and (iv) nodes receiving (or overhearing) a control or a data frame on the channel update their NAV accordingly so as to increase the channel utilization without increasing the collision probability on the wireless medium.

The main steps of the proposed protocol can be summarized as follows:

1) When a node has to transmit the first multicast data packet, it sends a multicast RTS (MRTS) frame to the multicast receivers.

2) On receiving the MRTS, each receiver sequentially sends a properly modified CTS frame (MCTS), which includes its highest acceptable data rate, based on the channel quality measured while receiving the MRTS. The MCTS transmissions follow the sequential order specified in the MRTS frame.

3) On receiving the MCTSs, the sender computes the minimum of the advertised data rates and chooses this value as the actual data rate for the data transmission.

4) On receiving the data frame from the sender, every receiver sends a proper ACK frame (MACK) in the sequential order as specified in the data frame.

The details of our RM^3 protocol and the associated rate adaptation scheme are described in the next sections.

For clarity of presentation, henceforth we refer to the members of a multicast group as *multicast final receivers* and to the next-hop neighbors of a given upstream node as *multicast MAC-layer receivers*. The latter ones act as forwarders in order to reach the intended multicast final receivers, and on their turn they may belong to the multicast group as well. As an example, in Figure 1, nodes represented as thick circles are the multicast final receivers, while the other ones, with the expection of the sender node S, are multicast MAC-layer receivers. In particular, node E which with A is a multicast MAC-layer receiver for S also belongs to the multicast group.

5. The RM^3 Transmission Procedures

The RM^3 considers some new control frames, which are modified versions of the respective frames specified by the 802.11 standard for unicast transmissions: Multicast RTS (MRTS), Multicast CTS (MCTS), and Multicast ACK (MACK); in addition, RM^3 makes use of a new Multicast DATA (MDATA) frame. Note that, the aforementioned frames can be simply identified by setting the subfield SUBTYPE in the

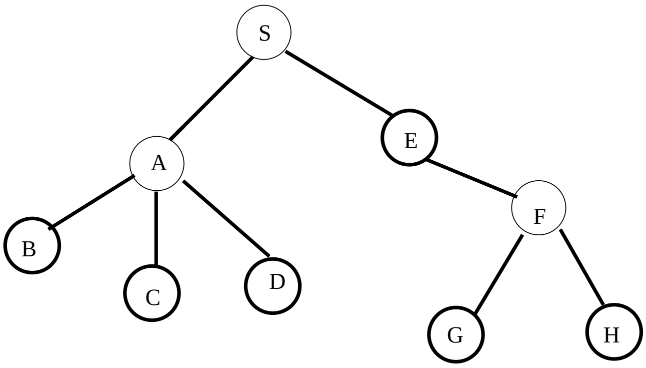

Figure 1: Example multicast tree

FRAME_CONTROL field of the MAC header, without altering the 802.11 standard MAC frame's structure.

5.1. First Packet Transmission

Let us assume that a source node, S, has to deliver a data flow to the members of a multicast group M. To do so, S transmits the first packet of the flow to its next-hop neighbors and operates as depicted in Figure 2 and described below; note that also intermediate nodes follow the same behavior as the one adopted by the source node when acting as forwarders towards their next-hop neighbors. Nodes having a single next-hop neighbor follow unicast transmission procedures.

S first transmits an MRTS frame following the standard 802.11 DCF channel sensing and backoff procedures. The MRTS frame differs from the standard RTS, because it has a variable-length destination address field, as in [9], which contains the MAC addresses (each 6-byte long) of the next-hop neighbors provided by the routing layer through the Address Resolution Protocol (ARP). Hence, being N the number of next-hop neighbors of S, the length of the MRTS frame exceeds the one of the standard RTS by $(N-1) \times 6$ bytes, as shown in Figure 3(a), where *RA*$_i$ represents the MAC address of receiver i. The order of appearance of the *RA*$_i$ addresses in the MRTS frame determines the order to be followed by the receivers to send back their MCTS frames.

The sender sets the DURATION field in the MRTS frame as follows:

$$N \cdot (2 \cdot SIFS + T_{MCTS} + T_{MACK}) + T_{MDATA} + SIFS \tag{1}$$

where the T_{MCTS} and T_{MACK} are the transmission delays of the MCTS and MACK frames, respectively, and T_{MDATA} is the transmission delay of the multicast data frame. Note that all these values are computed by assuming the basic rate as the transmission data rate, since, upon sending the first packet, the sender cannot know which data rate is suitable for its next-hop nodes.

The next-hop nodes, for which the MRTS is intended, sequentially send an MCTS frame back to S. Let us define as next-hop identifier for node i (NHID$_i$) the position of a next-hop neighbor in the order of MAC address fields (*RA*$_i$ with $i = 1, \ldots, N$)

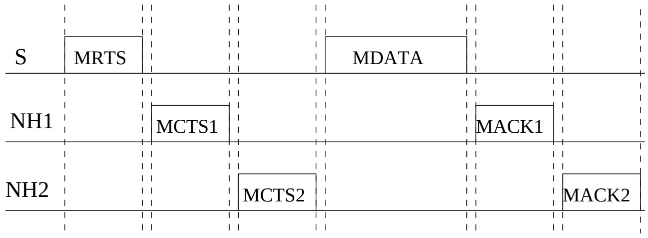

Figure 2: Transmission procedure used in RM^3 for the first packet of the multicast flow and for packet retransmissions

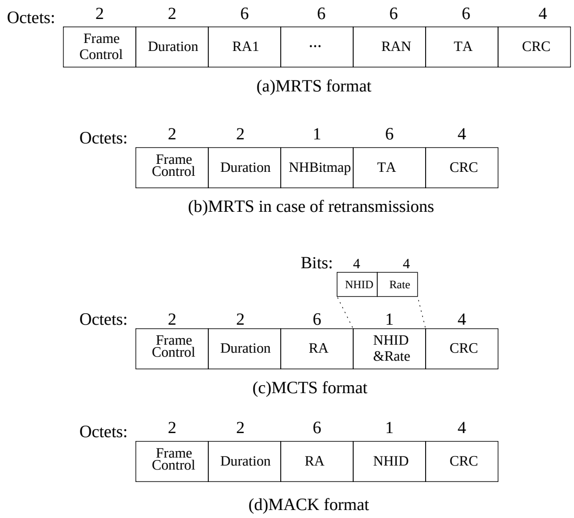

Figure 3: Control frames format

carried in the MRTS frame of the first data packet of the multicast flow[1]. The NHID size is determined by the maximum number of allowed next-hop neighbors and is much shorter than a MAC address (e.g., 4 bits can be a reasonable choice). Unlike previous proposals, our solution provides for the next-hop neighbors receiving an MRTS to store their NHID[2]. Also, to increase the efficiency of our scheme, we let MCTS (and MACK) frames carry a field with the NHID of the node issuing these control frames, as shown in Figure 3(c) (and 3(d)), thus allowing the sender S to differentiate between the MCTS (MACK) frames it receives from the different neighbors. Clearly, using the NHID in

[1] In case of addition/removal of members to/from a multicast group notified by the network layer, a new MRTS with extended RA_i addresses fields is transmitted by the sender (forwarder) node, whose next-hops are changed, with the aim to assign new NHIDs.

[2] Since a node may have more than one upstream node (although this is not the case for tree-based multicast routing protocols), it has to store different NHIDs. Thus, in order to ensure the uniqueness of the NHID, each node maintains the mapping between the upstream node and the NHID assigned by it.

Octets:	2	2	6	6	6	2	6	0−2312	4
	Frame Control	Duration	Address 1	Address 2	Address 3	Seq Control	NHBitmap	Frame Body	CRC

Figure 4: MDATA frame format

the frames exchange between S and its next-hop neighbors, and vice versa, instead of the node MAC address, significantly reduces the protocol overhead.

Collisions among MCTS frames are avoided by letting the next-hop nodes access the channel sequentially, according to their NHIDs. For example, a next-hop neighbor with NHID $= i$ ($i \geq 1$), will transmit its MCTS after $i \cdot SIFS + (i-1)T_{MCTS}$ seconds from the reception of the MRTS. The i-th next hop neighbor will set the DURATION field in its MCTS by decreasing the duration value included in the MRTS by the quantity: $i(SIFS + T_{MCTS})$.

The RM^3 protocol requires node S to send an MDATA frame including a next-hop bitmap field, NHBITMAP, which is a vector of length equal to N bits (i.e., the number of next-hop neighbors from which S expected to receive an MCTS frame). A bit 1 in the bitmap in position i, $i = 1, \ldots, N$, corresponds to the NHID of the i-th next-hop neighbor from which S received the MCTS frame. A zero in position i, $i = 1, \ldots, N$, of the bitmap means that the next-hop neighbor which was expected to transmit an MCTS after a time equal to: $i \cdot SIFS + (i-1)T_{MCTS}$, has failed to send it.

Note that the bitmap can be accommodated in one or more of the unused address fields in the MAC header, thus no overhead is added. Specifically, we choose to overwrite the fourth address field, which carries the Source Address information in case of communication between Access Points, to accommodate the 4-bit NHBITMAP field, as shown in Figure 4.

The DURATION field in the MDATA frame is set to $N_r(SIFS+T_{MACK})$, where N_r is the number of receivers from which the sender has successfully received the MCTS frames. This value can be obviously shorter than the duration originally advertised in the MRTS frame if N_r is smaller than N.

On receiving the MDATA frame, all next-hop neighbors read the bitmap field. If the bit corresponding to the NHID of a node is set to one, the node is required to transmit a MACK frame. If some of the nodes detect missing MCTSs (i.e., bits in the bitmap set to zero) from other next-hop neighbors, then they reschedule their own MACK transmission accordingly, in order to move up their transmission and replace the missing next-hop neighbors. In more detail, each receiver counts the number of zeros in the bitmap positions preceding the index position corresponding to its own NHID. If the number of the missing neighbors replies, denoted by N_m, is greater than zero, each node computes the transmission time of its own MACK frame according to the following expression:

$$(i - N_m)SIFS + (i - 1 - N_m)T_{MACK} \tag{2}$$

where i is the NHID of the node. Otherwise, the node transmits its MACK after $iSIFS + (i-1)T_{MACK}$ seconds from the reception of the MDATA frame.

A node, which has successfully received an MRTS and has sent an MCTS back to the source, by reading a 0 in the bitmap field included in the MDATA frame, realizes that its MCTS has not been received by the sender (this can be due to channel loss or collision). In this case, even if the receiver correctly decoded the bitmap field in the frame header (transmitted at the basic rate), the correct reception of the MDATA frame could be hindered by the choice of a data rate higher than the one allowed by the current link conditions between the source and the receiver. Furthermore, even if the receiver successfully detects the data frame, in order to avoid collisions with transmissions by other nodes, it cannot acknowledge its reception. It follows that the receiver needs to wait for a new MRTS/MCTS frame exchange.

The efficiency of the RM^3 transmission procedure can be increased by exploiting the DURATION field of the MACK frame. In legacy IEEE 802.11, the DURATION field in the ACK frame is only set to a value different from 0 when fragmentation is enabled, i.e., when the MORE FRAGMENT bit is set. We assume, according to the 802.11 standard, that fragmentation is not implemented for multicast transmissions, however we enable next-hop neighbors to set the DURATION field in their MACK frame so as to indicate the actual duration of the whole frame exchange. This value is copied by the relevant MDATA frame and updated by each next-hop receiver by subtracting the time already spent for preceding MACK transmissions. A node with NHID equal to i sets the DURATION field in the MACK frame according to the following expression:

$$DURATION_{MDATA} - (N_{c,i} + 1)(SIFS + T_{MACK}) \tag{3}$$

where $N_{c,i}$ is the number of nodes transmitting the MACK frame before node i. This number corresponds to the number of 1s before index i that appear in the bitmap in the header of the MDATA frame.

By doing so, the hidden terminals (i.e., the nodes that are outside the radio range of S but in radio proximity of some next hop nodes, and thus cannot hear the MDATA but can hear the MACK) can be informed of the instant of release of the radio resources. This duration can be shorter than the duration originally advertised in the MCTS frames. In this way, hidden terminals can update their NAV, and channel utilization can be increased.

5.2. Packet Retransmission

The sender S keeps track of the nodes that did not respond with an MCTS or an MACK frame. If there are any of them, S first carries out a new channel quality measurement through an MRTS/MCTS exchange with such nodes; then, it reschedules the MDATA transmission towards both types of nodes, following the standard backoff mechanism. Note that a missing MCTS or MACK trigger the same behaviour by S because a missing MACK could be the sign of an erroneous reception of the MDATA frame due to the channel conditions degradation between the transmitter and the receiver.

However, unlike the case of the first packet transmission, the receivers now have acquired a unique NHID. The MRTS frame size can therefore be reduced by including the next-hop bitmap, NHBITMAP, of few bits, instead of several MAC addresses, as shown in Figure 3(b). Clearly, by keeping the MRTS frame small in size, it is less subject to collisions due to hidden terminals. We recall that the bare-bone MRTS frame, with the NHBITMAP field in place of the MAC addresses, can be sent only when all the next-hops have retrieved their own NHID.

As in legacy 802.11, the data frame will be retransmitted till a maximum number of attempts has been reached, after which it is dropped. If S is unable to deliver the frame to one of its next-hop neighbors, it notifies the network layer, which will react accordingly.

In conclusion, we stress that, in order to maintain the control overhead low, RM^3 limits the use of the MRTS/MCTS frames to the first packet of a multicast data flow and to the case of retransmissions. All other packets are instead sent by S without control handshaking, as shown in Figure 5. Indeed, once the first data frame has been successfully sent, the set of 'active' next-hop nodes has been determined and the handshaking overhead can be avoided. Therefore, the sender instructs next-hop nodes to transmit feedbacks through the bitmap carried in the MDATA packet.

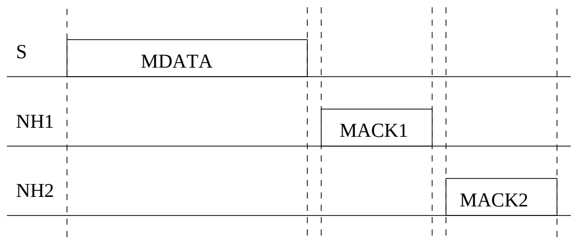

Figure 5: Transmission procedure used in RM^3 for multicast packets (except for the first one)

6. The RM^3 Rate Adaptation Mechanism

In order to increase the throughput over the multicast tree, RM^3 exploits MCTS frames so as to implement a rate adaptation mechanism at each multicast sender.

More specifically, each receiver notifies the sender of the highest data rate at which it can receive correctly, by using the 4-bit RATE subfield in the MCTS frame (see Figure 3(c)). The sender reads the RATE field in the MCTSs sent by the receivers, and computes the minimum of the advertised data rates. It then chooses this value as data rate for the next data transmission.

The rate adaptation procedure is carried out before sending the first packet of a multicast flow and it is triggered again every time a MACK frame is not received, in order to adapt to possible changes in the channel conditions during the multicast session. Indeed, a missing MACK could be due to the impossibility of correctly decoding the data packet due to channel state degradation. In this case, a new channel quality measurement phase must be started through an MRTS/MCTS exchange, in order to adapt the data rate accordingly.

We consider that each receiver selects the desired data rate based on the Signal-to-Noise Ratio (SNR) measurements taken while receiving the MRTS frame. The technique adopted to determine the desired data rate is out of scope of this work, and RM^3 is independent of the particular algorithm that is used. However, to derive the results in Section 8, we employed a simple and widely popular method that consists in comparing the SNR value with a set of predefined SNR thresholds. Such thresholds are computed assuming an a priori knowledge of the wireless channel model and represent the channel conditions required for a successful data reception at a given data rate [16], [18]. A receiver stores these threshold values and easily determines the highest acceptable data rate. Table 1 reports the set of data rates to be used according to the experienced value of SNR [18].

Table 1: SNR thresholds and data rates

Data rate [Mbps]	6	9	12	18	24	36	48	54
SNR threshold [dB]	21	22	23	26	30	34	38	40

7. NAV Updating

We now detail the NAV updating procedure followed by the nodes overhearing a transmission in the proposed RM^3 scheme.

The goal of the procedure we introduce is twofold: we aim at improving the channel utilization and at ensuring full protection against the hidden terminal problem. For the sake of clarity, we separately describe the NAV updating procedure at sender's neighbors and at receivers' neighbors. The former can hear the sender but are hidden to the receivers, the latter can hear at least one of the receivers but are hidden to the sender.

7.1. Procedure at Sender's Neighbors

Upon receiving an MRTS, S's neighbors that are not the intended receivers, will set their NAV to the following value (see Figure 6):

$$N \cdot (SIFS + T_{MCTS}) + SIFS \tag{4}$$

where N is the number of the expected receivers of the MRTS message. Note that N can be retrieved either through the number of MAC addresses carried in the MRTS frame, which is sent for the first packet of the multicast data flow, or through the number of '1s' in the bitmap carried in the MRTS frame that are sent in case of retransmissions. The procedure is also reported in Algorithm 1.

We point out that this technique avoids useless channel reservation when none of the intended next hops replies with an MCTS. It has been firstly introduced for unicast transmissions in [19], with the name of RTS Validation, to avoid the so-called false blocking problem. False blocking is due to the fact that a node receiving an RTS frame refrains from transmitting, even though the intended destination of the RTS frame does not send back a CTS and, thus, no data transmission will take place. In our case, upon overhearing an MRTS, a node sets its NAV value to the time instant corresponding to the MDATA transmission and then assesses the state of the channel, as shown in

Figure 6 (fifth row). If the channel is idle, then it defers no longer, otherwise it updates its NAV upon hearing the transmission of the MDATA frame.

However, beside false blocking, we observed a new problem, which is unique to multicast traffic and we named *false reservation*.

Firstly, when the number of next-hop neighbors from which the sender receives an MCTS (N_r) is smaller than the number of expected next-hop neighbors (N), the DURATION field in the MDATA frame advertises a shorter duration of the ongoing transmission with respect to what stated in the MCTS DURATION field, i.e., the reservation made by the sender through the MRTS frame turns out to be incorrect. The proposed NAV updating policy addresses this problem, since it gives sender's neighbors the possibility to read the DURATION field in the MDATA frame and to update their NAV accordingly, as shown in Figure 7 (seventh row), thus allowing a better channel utilization.

Secondly, a false reservation can occur due to the rate adaptation mechanism. Indeed, through the MRTS the sender advertises the duration of the upcoming data exchange by considering that the MDATA frame will be transmitted at the basic rate. When the data rate selected for the MDATA frame is higher than the basic rate, the channel is reserved for an unnecessarily longer time. Again, we solve this problem by letting the nodes set their NAV accordingly to the expression in (4): thus, sender's neighbors have the chance to update their NAV based on the actual duration advertised in the MDATA frame.

Algorithm 1 Sender's neighbors: NAV updating procedure

let *Nav* $\Rightarrow$ the current value of NAV
let *newNav* $\Rightarrow$ the new computed value of NAV
let *now* $\Rightarrow$ the current time
let $L \Rightarrow$ the MDATA time duration
let $f \Rightarrow$ the overheard frame
switch SUBTYPE in f
    case MRTS:
        $newNav \leftarrow now + N(\text{SIFS} + T_{MCTS}) + \text{SIFS}$
    case MDATA:
        $newNav \leftarrow now + \text{MDATA DURATION} + L$
    end case
end switch
if $newNav \geq Nav$ **then**
    $Nav \leftarrow newNav$
end if

7.2. Procedure at Receivers' neighbors

Similarly to [16], in RM^3 every node maintains, beside the usual NAV variable, a list of the tentative end times of each ongoing multicast transmission from the generic sender i to receiver j. The generic entry in this list is denoted by $NAV_{i,j}$ and is indexed

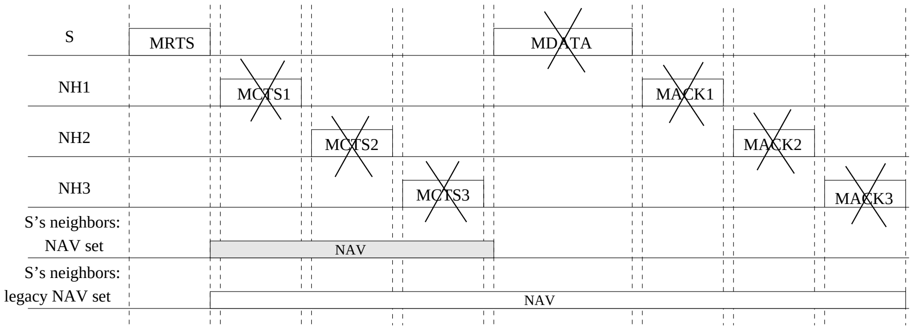

Figure 6: NAV updating procedure: S does not receive MCTSs and does not transmit MDATA

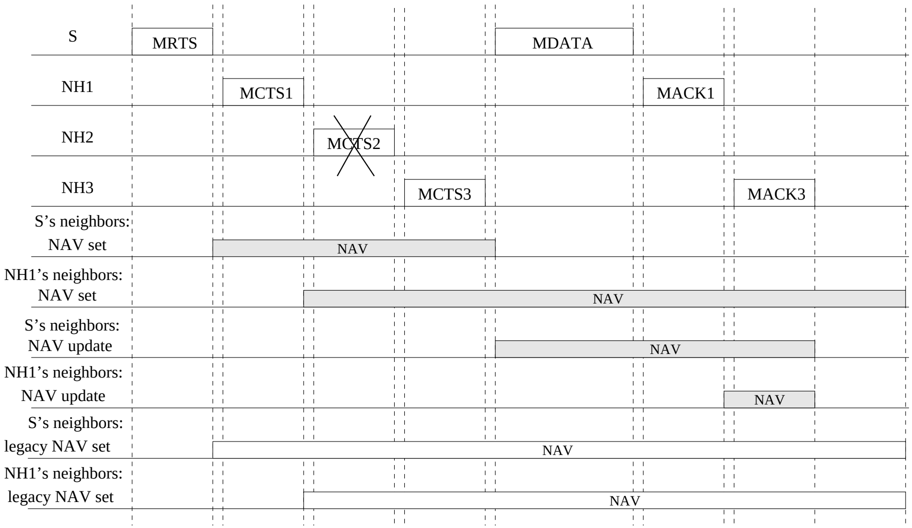

Figure 7: NAV updating procedure: NH2 does not answer with an MCTS

with the addresses of node i (which is the data sender that receives the MCTS frame) and of node j (i.e., the node with NHID $= j$ that transmits the MCTS frame).

As reported in Algorithm 2, when a node overhears an MCTS frame from node j it updates its NAV if the current value is lower than the newly advertised duration. Also, it stores the advertised duration value in $NAV_{i,j}$. However, the MCTS frame advertises the duration of the ongoing transmission in the worst case scenario, i.e., by assuming that the basic rate will be used for the transmission of the MDATA frame. It follows that nodes overhearing an MCTS and setting their NAV accordingly likely incur a false reservation problem. To avoid such inefficiency, in RM^3 nodes can update their NAV according to the information included in the MACK frames (see Figure 7).

Upon hearing a MACK frame from node j, a neighboring node checks the current value of its NAV and updates it accordingly, even if, unlike the standard, the DURATION field in the MACK frame advertises a shorter duration than the one advertised in the

Algorithm 2 Receivers' neighbors: NAV updating procedure

1: **let** *Nav* $\Rightarrow$ the current value of NAV
2: **let** *newNav* $\Rightarrow$ the new computed value of NAV
3: **let** $l(i, j)$ $\Rightarrow$ the generic entry of the list $NAV_{i,j}$
4: **let** *now* $\Rightarrow$ the current time
5: **let** f $\Rightarrow$ the overheard frame sent from node i to node j
6: **switch** SUBTYPE in f
7:     **case** MCTS:
8:         *newNav* $\leftarrow$ *now* $+$ MCTS DURATION
9:         store *newNav* in $l(i, j)$
10:         **if** *newNav* $\geq$ *Nav*
11:             *Nav* $\leftarrow$ *newNav*
12:         **end if**
13:     **end case**
14:     **case** MACK:
15:         *newNav* $\leftarrow$ *now* $+$ MACK DURATION
16:         **if** (*newNav* $\leq$ *Nav* **and** *newNav* $\geq l(z, k))\ \ \forall z \neq i,\ \forall k \neq j$
17:             *Nav* $\leftarrow$ *newNav*
18:             store *newNav* in $l(i, j)$
19:         **end if**
20:     **end case**
21: **end switch**

previously received MCTS frames (lines 14–20 in Algorithm 2). In particular, the second condition in line 16 ensures that a NAV update does not affect other ongoing transmissions whose end is scheduled after the one from i to j.

Again, we stress that enabling neighboring nodes to shorten their NAV helps counteract the false reservation problem arising when the duration of the ongoing transmission is shorter than what stated in the MCTSs DURATION field. This can happen when either the number of MACK frames is lower than expected (i.e., $N_r < N$), or when the data transmission is sped up by using a higher data rate than the basic one, originally advertised in the MRTS/MCTS handshake.

Finally, we remark that the NAV updating procedure holds for both transmissions of first packets and retransmissions. For the following packets, the MRTS/MCTSs handshake is avoided, thus, nodes overhearing both MDATA and MACK frames update their NAV as suggested in the legacy 802.11. However, an important difference exists: in RM^3 the MACK DURATION field informs the nodes, which overhear this frame and are hidden to the sender, about the residual duration of the ongoing transmission[3] (see Figure 8), while in the case of unicast 802.11 traffic the channel is released after the ACK frame sent by the receiver. Thus, thanks to the fact that in RM^3 the receivers set the MACK DURATION field to a value other than 0 and the overhearing nodes update

[3] Notice that, given a MACK transmissions, other MACK frames may follow.

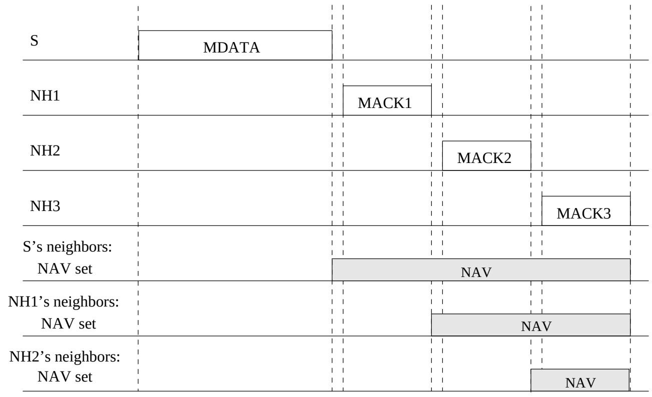

Figure 8: NAV updating procedure: MACK DURATION field not zero

their NAV accordingly, the multicast transmission is not threatened by collisions due to hidden nodes.

8. Performance Evaluation

We implemented RM^3 in ns2 [20], and compared its performance against results derived using either the MMP scheme [9], the standard IEEE 802.11, and the early version of our protocol called RAMP [8].

8.1. Simulation Settings and Benchmark Schemes

Table 2 summarizes the main simulation settings. We consider a network composed of 50 nodes and simulate several instances of network topologies where nodes are randomly deployed over a 1 km $\times$1 km geographical area, according to a uniform distribution. Mobility of nodes is not considered, since our aim is to focus on the validation of the proposed MAC scheme, regardless of how the routing layer addresses lack of connectivity and mobility issues.

To analyze the performance of RM^3, we collect two kinds of results. The first set of results is derived by considering a source node generating Constant Bit Rate (CBR) traffic at the rate of 64 Kbps, each packet has size equal to 512 bytes. Both the well-known Two-ray Ground Model and the Ricean fading model [21], which accounts for time-varying channel conditions, are used as propagation models.

The second set of results are computed for a streaming session issued by a source node under a harsh Ricean fading propagation model. Specifically, we used the well known benchmark "akyio" CIF video sequence (10 sec), available at [22], coded with MPEG4 standard codec at 30 frames/sec (fps), with an intra-period of 300 frames. The generated traffic, both CBR and video traffic, has to be delivered to a group of multicast users. The size of the multicast group is a varying parameter in our simulations.

At the network layer, we consider that the Multicast Ad-hoc On demand Distance Vector (MAODV) [23] is implemented. MAODV is a tree-based routing protocol,

Table 2: Simulation settings

Parameter	Value
Area Size	1000 x 1000 m^2
Number of Nodes	50
Routing Protocol	MAODV
Basic Transmission Rate	6 Mbps
Transmission Range	250 m

which follows directly from the unicast AODV and discovers multicast routes on demand. The route discovery mechanism employs the same Route Request and Route Reply messages used in the AODV protocol.

At the MAC layer, we set the 802.11 parameters to their default value [2]. The basic transmission rate is set to 6 Mb/s. All nodes have a common transmission range of 250 m.

The main features of the benchmarked protocols are shown in Table 3.

Note that, when RM^3, RAMP, and MMP are applied, we modified MAODV in order to enable any upstream node's routing layer, at each step, to pass information about the downstream nodes to the MAC layer. Also, according to the multicast tree created through MAODV, a given node may have one or more next-hops and, thus, either unicast or multicast transmissions may have to be used. In the former case, we point out that when the RM^3 is used, the same mechanisms for rate adaptation and NAV updating described in Sections 6 and 7 are applied.

Table 3: Main Features of the Compared Protocols

Feature	802.11	MMP	RAMP	RM^3
Acknowledgements	No	Yes	Yes	Yes
Virtual Carrier Sensing	No	retransmissions	1st packet/retransmissions	1st packet/retransmissions
Additional Address Fields	No	Always	1st packet	1st packet
Enhanced NAV Updating	No	No	No	Yes
MACK Duration	0	0	not zero	not zero
Transmission Rate	Basic	Basic	Basic	Rate-adaptive

8.2. Performance Metrics

In order to assess the protocol performance, we evaluate the *packet delivery ratio* and the average *end-to-end packet delay*. The former is computed as the the number of packets successfully received at the application layer by a given multicast final receiver over the total number of packets generated by the application layer at the source node. The latter is the average time elapsed from the time instant when the packet is generated at the source node to the time instant when the packet is received by a given member of the multicast group.

Furthermore, to evaluate the efficiency of the reliable schemes under study, with respect to the IEEE 802.11 standard, we compute the *multicast overhead*, i.e., the ratio between the number of bytes transmitted at the MAC layer by a reliable scheme and the number of bytes transmitted at the MAC layer when 802.11 is used. In particular, for RM^3 and RAMP, such metric accounts for the bytes included in the transmitted MRTS, MCTS, MACK frames, and required for retransmitted packets, while in the case of MMP, it accounts also for the extra bytes added to the data packet for carrying the MAC addresses of the intended next-hops. Clearly, such an overhead implies both a waste of bandwidth and an additional energy consumption for the whole network, with respect to when the IEEE 802.11 MAC scheme is applied.

In order to account for the channel utilization efficiency, we also evaluate the *collisions percentage* as the ratio of the number of occurred collision events to the number of data packets generated by the application layer at the multicast source. The computation of this metric allows a fair comparison among the deployed schemes, since we measure the overhead and the collision events regardless of the number of packets sent at the MAC layer, which may vary according to the considered protocol.

Finally, to emphasize the gain in performance obtained through RM^3 with respect to the IEEE 802.11 standard, we consider the *average rate satisfaction*. It is computed for every downstream node among a set of multicast MAC-layer intended receivers, and is given by the data rate used by a given original/intermediate sender for the transmission of an MDATA over the allowed data rate advertised by the node in the MCTS frame. This metric can be used as an indicator of the data rate increase that we enable through our rate adaptation mechanism.

With regard to the second scenario under study, we evaluate as additional performance metric the perceptual quality experienced by the users, in terms of Peak Signal-to-Noise Ratio (PSNR). We evaluate the PSNR at all members of the multicast group, and we report the best, worst and average PSNR values, along with the related standard deviation.

We point out that all of the above metrics are computed by averaging the values experienced by each multicast group member, and then averaging such intermediate results over ten different instances of the network topology. As for the average rate satisfaction, this is first averaged over the number of multicast MAC-layer receivers having the same upstream node and, then, over all one-to-many transmissions that occur along the multicast tree built through MAODV. Finally, we average the results over ten topology instances.

8.3. Results with CBR Traffic

In the remainder of this section, we present the behavior of the above performance metrics when the multicast group size varies.

Figure 9 presents the packet delivery ratio obtained through RM^3 and 802.11, both when the Two-ray Ground and the Ricean propagation models are used. We observe that the retransmission policy adopted in RM^3 ensures a very high packet delivery ratio, i.e., approximately 100% under the two-ray ground propagation model and slightly lower values when the time-varying harsh Ricean channel is considered. In order to increase of the readability of the plot, the performance of MMP and RAMP, similar to the one exhibited by RM^3, is not reported.

Conversely, the standard IEEE 802.11 suffers from unrecovered packet losses. Specifically, when the Two-ray Ground model is considered, just few losses due to collisions are experienced, causing a slight performance degradation. When, instead, the Ricean model is used and losses due to bad channel conditions become more likely, the packet delivery ratio significantly decreases. These results therefore show that a reliable scheme is highly needed in order to successfully tackle transmission failures due to both collisions and channel errors, hence to ensure a good level of quality of service.

If the reliable schemes under study all provide similar performance in terms of packet delivery ratio, RM^3 is no match for MMP and RAMP in terms of average end-to-end packet delay, as shown in Figure 10 in the case of the Two-ray Ground propagation model. Indeed, unlike MMP, RM^3 does not require the use of additional control packets or packet fields with respect to the standard 802.11 DCF, unless retransmissions have to take place. Also, the rate adaptation mechanism in RM^3 leads to a higher throughput, hence lower delay, than RAMP. Finally, the standard IEEE 802.11 exhibits a slightly lower latency than RM^3 but at the expense of reliability (since no packets are retransmitted upon collision).

This performance is confirmed by the results on multicast overhead, which are presented in Figure 11. As expected, RM^3 and RAMP incur the same overhead, since the rate adaptation mechanism in RM^3 does not require the transmission of additional packet fields or control frames. Relatively to MMP, instead, we notice a significant overhead reduction due to the bare-bone frame exchange implemented in RM^3 and RAMP.

Figure 12 presents the collisions occurrence for all the schemes under study, again under two-ray ground propagation conditions. As expected, thanks to the higher reliability level and the NAV updating procedures, in RM^3 collisions are less likely than in the standard 802.11 MAC. Interestingly, however, both RM^3 and RAMP outperform also MMP. This behavior is due to the fact that, in RAMP, MACK frames carry the remaining duration of the whole frame exchange thus informing hidden terminals about the channel status. Furthermore, barebone MRTS frames, less prone to collisions are employed in case of retransmissions. As a result, the collision probability is greatly reduced.

When the Ricean propagation model is used, the same differences among the compared schemes are noticed, with higher values of overhead and delay resulting from the higher number of retransmissions incurred in by the reliable schemes, as compared to the ones obtained with the Two-ray Ground model.

At last, by looking at Table 4, we can observe that, thanks to the RM^3 rate adaptation mechanism, nodes involved in the multicast forwarding experience high rate satisfaction values, with respect to the case when the basic rate is used as transmission rate, as in IEEE 802.11. Specifically, values reported by Table 4 show that three-fold data rate improvement can be experienced with respect to the basic rate used by the other protocols.

8.4. *Results with Video Traffic*

We now consider a video traffic source and present the PSNR performance for the IEEE 802.11 standard and the RM^3 scheme, in Table 5. We observe that the perceived video quality is significantly improved when RM^3 is used: compared to 802.11, a gain in the PSNR of at least 5 dB is obtained for the average and worst cases, while up to 20 dB gain can be achieved in the case of the best PSNR experienced among the multicast group members. Such benefits are due to the joint implementation of rate adaptation and retransmission procedures in RM^3, which allow a better use of channel resources and the recovery of lost frames.

Looking at the standard deviation values, we notice a high variability in the measured PSNR values. Such behavior is mainly due to the fact that results are averaged over the final receivers, i.e., the multicast group members, which are differently located with respect to the source node. The multicast final receivers that are located nearby the source node achieve excellent performance, close to the ideal PSNR that can be achieved in case of no losses (namely, 45.48 dB). Instead, others, which are further away from the source, experience an increasing delay, since retransmissions cumulate, as the number of hops between source and receiver grows. This could strongly penalize the digital video quality which typically relies on frames to be displayed at a constant rate.

In video transmission systems, the packet loss is not the only important metric for the perceived video quality, but the delay of packets and the variation of the delay, usually referred to as jitter, must also be recorded. Table 6 shows the average packet jitter computed as the time difference between the instants of reception of a packet and the previous one. We can notice that the average jitter is higher for RM^3, as compared to the 802.11 protocol. The reason for such a behavior is again due to the enforcement of retransmission procedures for lost packets, which unavoidably increases the delay.

Although the effects of high jitter can be mitigated using a properly modified playout buffer, it is worth pointing out that, in our performance evaluation, no playout buffer was introduced at the multicast final receivers; its presence might have further boosted the PSNR values, by enhancing the chances of packet reordering.

9. Conclusion

We proposed a reliable and efficient MAC scheme to support high-throughput multicast traffic in wireless multi-hop networks. Reliability is provided by integrating the standard features of the legacy 802.11 MAC protocol with low-overhead mechanisms for error recovery. Throughput efficiency is instead obtained through an efficient NAV-updating technique and a receiver-based rate adaptation scheme, which are implemented at every downstream node involved in the forwarding procedure over the multicast tree. Our solution has several advantages: it is fully distributed, easy to deploy in 802.11-based devices, and has low complexity. Simulation results showed that the proposed scheme greatly outperforms the standard 802.11 protocol in terms of transmission reliability, and provides lower delay and higher channel utilization than other state-of-the-art reliable MAC protocols.

References

[1] L. Junhai, Y. Danxia, X. Liu, and F. Mingyu. “A Survey of Multicast Routing Protocols for Mobile Ad-Hoc Networks,” *IEEE Communications Surveys and Tutorials*, vol. 11, no. 1, First Quarter 2009.

[2] “IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, June 2007, Std. 802.11-2007.

[3] K. Tang, and M. Gerla, “MAC Layer Broadcast Support in 802.11 Wireless Networks,” *IEEE MILCOM, 2000*, vol. 1, pp. 544–548, Oct. 2000.

[4] K. Tang, and M. Gerla, “Random Access MAC for Efficient Broadcast Support in Ad Hoc Networks,” *IEEE Wireless Communications and Networking Conference (WCNC 2000)*, pp. 454–459, Sept. 2000.

[5] S. Gupta, V. Shankar, and S. Lalwani, “Reliable Multicast MAC Protocol for Wireless LANs,” *IEEE ICC’03*, vol. 1, pp. 93–97, 2007.

[6] W. Si, and C. Li, “RMAC: A Reliable Multicast MAC Protocol for Wireless Ad Hoc Networks,” *IEEE International Conference on Parallel Processing (ICPP’04)*, vol. 1, pp. 494–501, Aug. 2004.

[7] C.Y. Chiu, E.H.-K. Wu, Y. Wang, and G.-H. Chen, “A Reliable and Efficient MAC Layer Broadcast Protocol for Mobile Ad Hoc Networks,” *IEEE Transactions on Vehicular Technology*, vol. 56, no. 4, pp. 2296–2305, July 2007.

[8] C. Campolo, C. Casetti, C.F.Chiasserini, and A. Molinaro, “An 802.11-based MAC Protocol for Reliable Multicasting in Multihop Networks,” *IEEE VTC-Spring*, 2009.

[9] H. Gossain, N. Nandiraju, K. Anand, and D. P. Agrawal, “Supporting MAC Layer Multicast in IEEE 802.11 based MANETs: Issues and Solutions,” *IEEE International Conference on Local Computer Networks (LCN’04)*, pp. 172–179, 2004.

[10] K. Tang, and M. Gerla, “MAC Reliable Broadcast in Ad Hoc Networks,” *IEEE MILCOM 2001*, pp. 1008–1013, Oct. 2001.

[11] M.-T. Sun, L. Huang, A. Arora, and T.-H. Lai, “Reliable MAC Layer Multicast in IEEE 802.11 Wireless Networks,” *International Conference on Parallel Processing*, pp. 527–536, Aug. 2002.

[12] J. Kuri, and S.K. Kasera, “Reliable Multicast in Multi-access Wireless LANs,” *Wireless Networks*, vol. 7, no. 4, pp. 359–369, July 2001.

[13] B.S. Kim, and S.W. Kim, “OFDMA-Based Reliable Multicasting MAC Protocol for WLANs,” *IEEE Transactions on Vehicular Technology*, vol. 57, no. 5, pp. 172–179, September 2008.

[14] S. Jain, and S. R. Das, "MAC layer Multicast in Wireless Multihop Networks," *1st International Conference on Communication System Software and Middleware (Comsware '06)*, pp. 1–10, 2006.

[15] A. Kamerman, and L. Monteban, "WaveLAN II: A high-performance wireless LAN for the unlicensed band," *Bell Labs Technical Journal*, pp. 118–133, Summer 1997.

[16] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," *Mobile Computing and Networking*, 2001.

[17] A. Basalamah, H. Sugimoto, and T. Sato, "Rate Adaptive Reliable Multicast MAC Protocol for WLANs," *IEEE VTC-Spring*, pp. 1216–1220, Melbourne, Australia, May 2006.

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

[19] S. Ray, J.B. Carruthers, and D. Starobinski, "RTS/CTS-induced congestion in ad hoc wireless LANs," *IEEE WCNC*, 2003.

[20] The Network Simulator, ns2. http://www.isi.edu/nsnam/ns/

[21] R.J. Punnoose, P.V. Nikitin, and D.D. Stancil, "Efficient Simulation of Ricean Fading within a Packet Simulator," in *Proc. of IEEE Vehicular Technology Conference*, pp. 764-767, 2000.

[22] "Yuv video sequences," Available at http://trace.eas.asu.edu/yuv.

[23] E. M. Royer, and C. E. Perkins, "Multicast Operation of the Ad Hoc On Demand Distance Vector Routing Protocol," *ACM MOBICOM*, pp. 207–218, Aug. 1999.

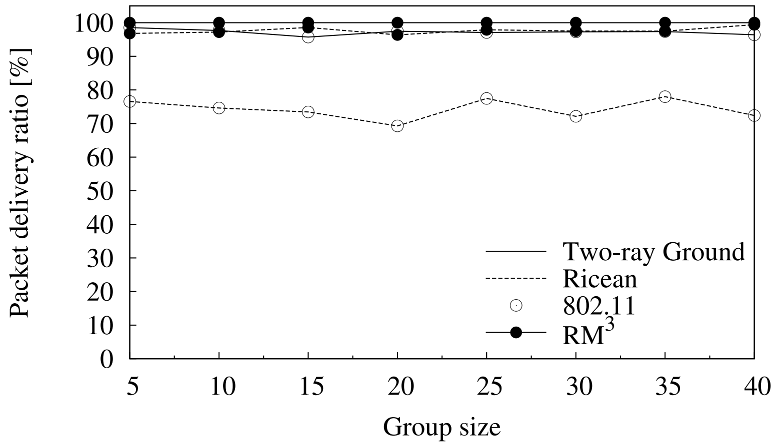

Figure 9: Average packet delivery ratio as the multicast group size varies. RM^3 is compared against standard 802.11 DCF

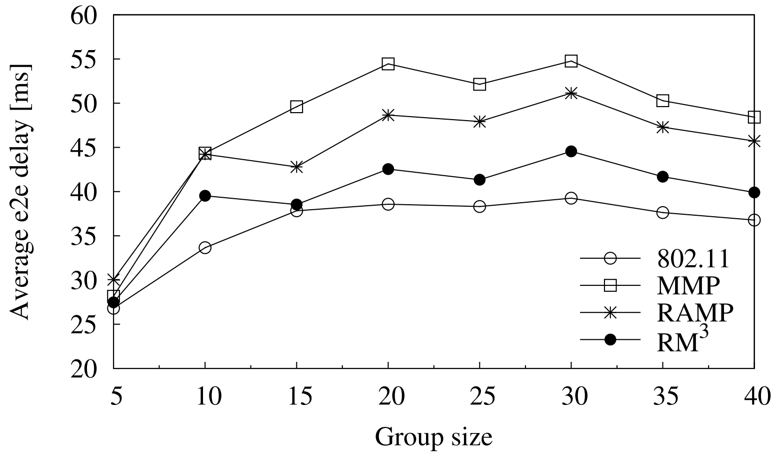

Figure 10: Average end-to-end delay as the multicast group size varies. RM^3 is compared against standard 802.11 DCF, MMP, and RAMP

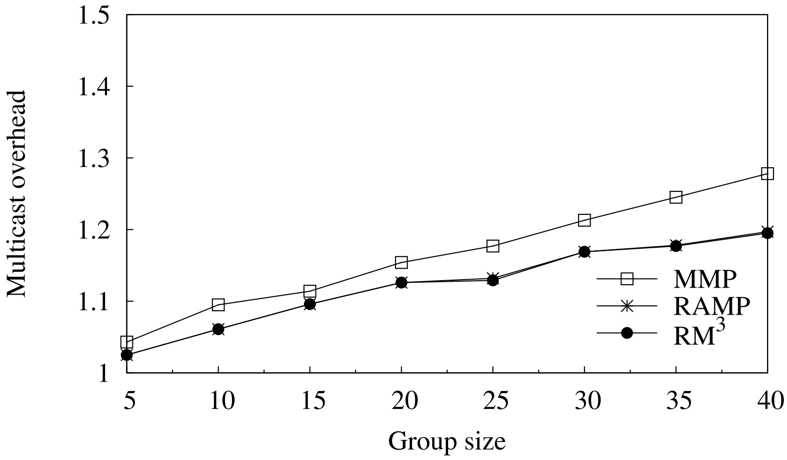

Figure 11: Multicast overhead (with respect to legacy 802.11) as the multicast group size varies. RM^3, MMP, and RAMP are compared

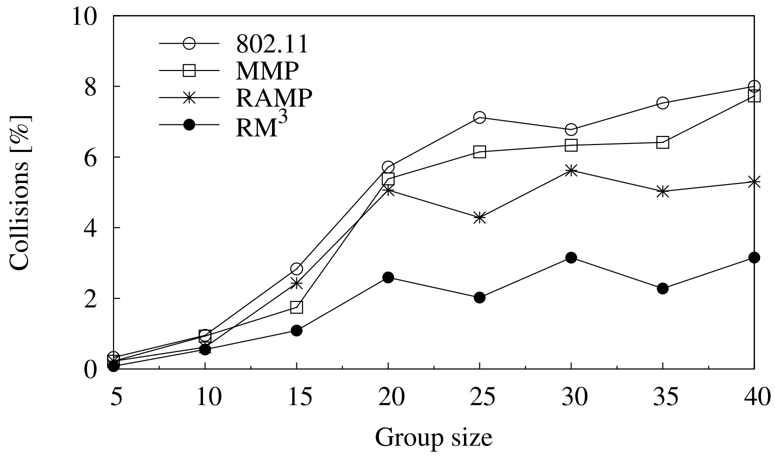

Figure 12: Percentage of collision events as a function of the multicast group size. RM^3 is compared against standard 802.11 DCF, MMP, and RAMP

Table 4: Average rate satisfaction

Group size	802.11	RM^3
5	0.331	0.997
10	0.287	0.867
15	0.287	0.866
20	0.275	0.837
25	0.282	0.866
30	0.263	0.812
35	0.257	0.789
40	0.290	0.877

Table 5: PSNR (expressed in decibel)

Group size	802.11				RM^3			
	Avg	Best	Worst	Std. dev	Avg	Best	Worst	Std. dev
5	19.63	25.63	15.14	2.58	32.46	46.45	22.19	6.59
10	15.48	17.74	14.06	0.88	23.01	38.51	18.55	3.80
15	14.44	17.06	13.67	0.83	24.82	39.60	19.05	5.87
20	17.73	24.16	16.21	1.70	26.53	39.83	20.68	6.05
25	15.21	17.93	12.97	1.26	24.24	37.69	20.31	5.13
30	15.37	18.81	13.37	1.17	21.37	32.37	15.08	4.20
35	14.82	18.09	12.95	1.02	26.12	35.98	21.19	5.88
40	14.17	16.14	12.94	0.75	23.88	37.02	19.39	4.84

Table 6: Jitter (expressed in ms)

Group size	802.11	RM^3
5	17.1	19.13
10	26.3	27.19
15	31.26	33.67
20	32.17	36.21
25	35.31	42.97
30	37.37	43.37
35	39.92	52.95
40	43.78	56.06