

Performance Analysis and Algorithm Selection for Reliable Multicast in IEEE 802.11aa Wireless LAN

Albert Banchs, Antonio de la Oliva, Lucas Eznarriaga, Dariusz R. Kowalski, Pablo Serrano

Abstract—Legacy IEEE 802.11 does not efficiently support multicast transmissions. In order to cope with the increasing demand for multicast, mainly required to deliver multimedia traffic, the IEEE 802.11aa Task Group has recently standardized new mechanisms to allow efficient and robust transmission of multicast flows in Wireless LAN. However, the standard allows the use of different mechanisms for this purpose, and leaves open the choice of which one to use for a given scenario. In this paper, we present an analytical model to evaluate the performance of the mechanisms included in the 802.11aa standard and compare their performance. Our analysis shows that there is no absolute winner out of these mechanisms, and performance strongly depends on the scenario. Building on our model, we then propose a novel algorithm that selects the best multicast mechanism to use as a function of the scenario conditions. Our results are validated by extensive simulations.

I. INTRODUCTION

The IEEE 802.11 standard for Wireless LAN (WLAN) [1] is one of the most used technologies to provide broadband connectivity to the Internet. Nowadays, it is common to transport multimedia flows with relative large bandwidth requirements over 802.11 WLANs (e.g., YouTube, VideoLAN). However, the original 802.11 standard is poorly suited for the efficient support of such flows, because of the following reasons: (i) the transmission rates firstly available imposed a severe bottleneck on the maximum achievable rate, regardless of the efficiency of the MAC protocol; (ii) only “best-effort” service was supported, thus preventing any traffic differentiation to prioritize multimedia; and (iii) multicast transmissions were very inefficient and unreliable [2].

The subsequent amendments to the 802.11 standard have addressed the first two limitations. On the one hand, the introduction of PHY-amendments have boosted the maximum achievable rates, starting with the 802.11b [3] that increased the maximum rate up to 11 Mbps, continuing with the 802.11a and 802.11g amendments that reach up to 54 Mbps, and finally with the 802.11n [4], which introduces the enhancements for higher throughput. On the other hand, the 802.11e amendment [5] has introduced traffic differentiation through the setting of the contention parameters, enabling both the ability to prioritize one type of traffic over other types as well as a more efficient operation of WLANs by proper tuning of the MAC parameters [6]. The remaining challenge, therefore, is to efficiently support multicast over 802.11 WLANs.

The IEEE 802.11aa Task Group has recently addressed this last limitation, with the definition of the mechanisms to support “Robust streaming of Audio Video Transport Streams” [7]. Its focus is to extend the base 802.11 standard with mechanisms that improve performance of multimedia streaming over WLAN. In particular, the new mechanisms target at significantly improving the transport of multimedia streams by introducing (i) a ‘*stream classification service*’, which aims at providing intra-flow prioritization to allow for a graceful degradation of video quality; (ii) *interworking with IEEE 802.1AVB*, for end-to-end reservations; (iii) *Overlapping Basic Service Set (OBSS) management*, for the coordination between multiple APs; and (iv) a ‘*group addressed transmission service*’, which provides an effective and efficient way to transmit multicast traffic to a group of stations. The focus of this paper is on the latter.

The IEEE 802.11aa standard defines different mechanisms for multicast transmission, each of which provides a different behavior in terms of efficiency and reliability. However, the standard does not give any insight into the performance of each mechanism, or provide any guideline on which mechanism to use for a given scenario. In this paper, we address this issue; in particular, the two key contributions of this paper are:

- 1) We present an analytical model to evaluate the performance of the mechanisms defined in the 802.11aa standard in a mixed scenario with multicast streams and unicast traffic. We quantify both the reliability that each mechanism provides to a multicast stream (for different numbers of receivers) and its efficiency, in terms of the throughput obtained by multicast and unicast users. Our results show that there is no “winner” mechanism, as each of them offers different trade-offs between efficiency, reliability and complexity, depending on the considered scenario.
- 2) We propose a novel algorithm to select the best mechanism for a given scenario. Our algorithm takes as input the number of unicast and multicast receivers in the WLAN, and based on this input it determines which of the mechanisms defined in the standard provides the best performance. In particular, the proposed algorithm chooses the mechanism that provides the best overall throughput performance while ensuring that a sufficient level of reliability is provided to multicast streams.

Due to the poor performance of legacy IEEE 802.11 multicast, in the past a substantial amount of effort has been devoted to the design of novel mechanisms to improve multicast performance. One of the first works on reliable multicast was

A. Banchs is with Univ. Carlos III of Madrid and Institute IMDEA Networks. A. de la Oliva and P. Serrano are with Univ. Carlos III of Madrid. D. R. Kowalski is with Univ. of Liverpool. This work was performed during the visit of D. R. Kowalski to Institute IMDEA Networks and the postgraduate studies of L. Eznarriaga at the Univ. Carlos III of Madrid.

the approach proposed in [8]. This has been followed by many others, including leader-based protocols [9]–[12], a proposal based on batch mode multicast MAC [13], a broadcast medium window protocol [14] and a network coding based approach [15], among many others. However, such proposals require changes in the physical layer or the MAC, which makes their deployment difficult and prevents the coexistence with legacy IEEE 802.11.

Few works have addressed the performance of 802.11aa, which is the focus of this paper. In [16] authors provide a qualitative description of the new mechanisms, and in [17], a numerical assessment based on simulations is presented; however, neither of these approaches evaluates 802.11aa performance analytically. The work of [18] studies the problem of routing when multiple of the multicast schemes defined in 802.11aa are available; however, while it considers some metrics for each scheme, the analytical part is limited to very simple metrics. Finally, [19] proposes the first analytical model of Block Ack methods of 802.11aa. While some parts of the analysis of [19] are similar to ours, their work focuses specifically on the enhanced leader based protocol (ELBP), while ours aims at analyzing all the methods defined in 802.11aa instead of only one; furthermore, we address a more generic scenario, that includes both multicast and unicast stations. To the best of our knowledge, this is the first paper that presents a comprehensive analytical model for the performance of the mechanisms defined in the IEEE 802.11aa amendment.

The rest of the paper is organized as follows. In Section II, we provide a short summary of the different mechanisms that can be used to transport multicast traffic over 802.11 WLANs, including both the existing mechanisms as defined in the legacy standard and the new mechanisms proposed within TGaa. In Section III, we present an analytical model for the performance of these mechanisms in a WLAN with unicast and multicast stations under different conditions. In Section IV, we present a selection algorithm that allows choosing the best multicast mechanism given a reliability threshold. In Section V, we evaluate the proposed models as well as the selection algorithm via extensive simulations. Finally, in Section VI, we summarize the main results of this paper.

II. MULTICAST MECHANISMS IN IEEE 802.11

In the following, we present the different mechanisms that can be used in 802.11 WLANs for multicast delivery, including both the legacy mechanism from the original 802.11 standard as well as the new mechanisms proposed in the 802.11aa amendment.

A. Multicast service with legacy 802.11

The IEEE 802.11 standard includes a specific mechanisms to transmit multicast frames, which are the data frames with a multicast address as the Destination Address. With this mechanism, no ACK shall be transmitted by any of the recipients of the frame, and hence the frames that suffer errors due to interference or collisions are not retransmitted. The lack of MAC-level recovery on multicast frames results

in a reduced reliability for this kind of traffic. In addition, all multicast frames must be transmitted at one of the rates included in the Basic Rate Set. This set is defined by the Access Point (AP) and includes the minimum set of rates that a station must support in order to join the AP. Although it is not a requirement, usually the Basic Rate Set includes only rates with lower order modulations, and hence the transmission of multicast frames is performed at a reduced speed, which decreases the overall performance of the WLAN (i.e., the so-called *performance anomaly* [20]).

The other option for transmitting audio/video frames to multiple receivers with the legacy 802.11 standard is by using unicast transmissions. Unlike multicast, unicast traffic can be transmitted at any rate and it is acknowledged, so its reliability is higher than standard multicast traffic. The main drawback, however, is that the bandwidth required to transmit the same flow to multiple receivers (and the delay) grow with the number of receivers. Hence, this option is only feasible for low sending rates and a reduced number of receiving stations.

B. Multicast mechanisms with IEEE 802.11aa: GATS

The recently standardized IEEE 802.11aa amendment has been designed to address the transmission of multimedia, implementing a set of new functionalities over the base specification. The extensions to the IEEE 802.11 standard have been defined in the IEEE Std. 802.11aa-2012 (amendment to IEEE Std. 802.11-2012 as amended by IEEE Std. 802.11ae-2012) [7]. Their objective is to efficiently improve the reliability of audio and video streaming, while maintaining the service received by the other streams. The amendment also specifies related functionality (e.g., OBSS management, interworking with IEEE 802.1AVB and Stream Classification Service). Here, we focus on the mechanisms defined to transport multicast traffic, which are referred to by the standard as *Group Addressed Transmission Service* (GATS).

GATS addresses two of the main weaknesses of the use of multicast in the legacy 802.11 standard described in the previous section, namely, (i) the poor reliability of the service, caused by the fact that multicast frames are not acknowledged; and (ii) the high inefficiency resulting from the use of a low modulation coding scheme. In order to overcome these limitations, GATS includes the Directed Multicast Service (DMS), which was first introduced in IEEE 802.11v [21] and is extended to target group addressed frames, and also defines a Groupcast with Retries (GCR) service. The GCR service improves the performance and the reliability of the delivery of frames addressed to a group of stations by defining new *group addressed* retransmissions, a *group* being a set of stations listening to the same (non-multicast) address, called *group concealment address*.

In the following, we describe the different mechanisms that we analyze in this paper, including those specified in the 802.11aa standard as well as a variation of one of the 802.11aa mechanisms. The operation of these mechanisms is illustrated in Fig. 1, which also includes the *legacy* multicast service described in the previous section (which we refer to “No-Ack/No-Retry”).

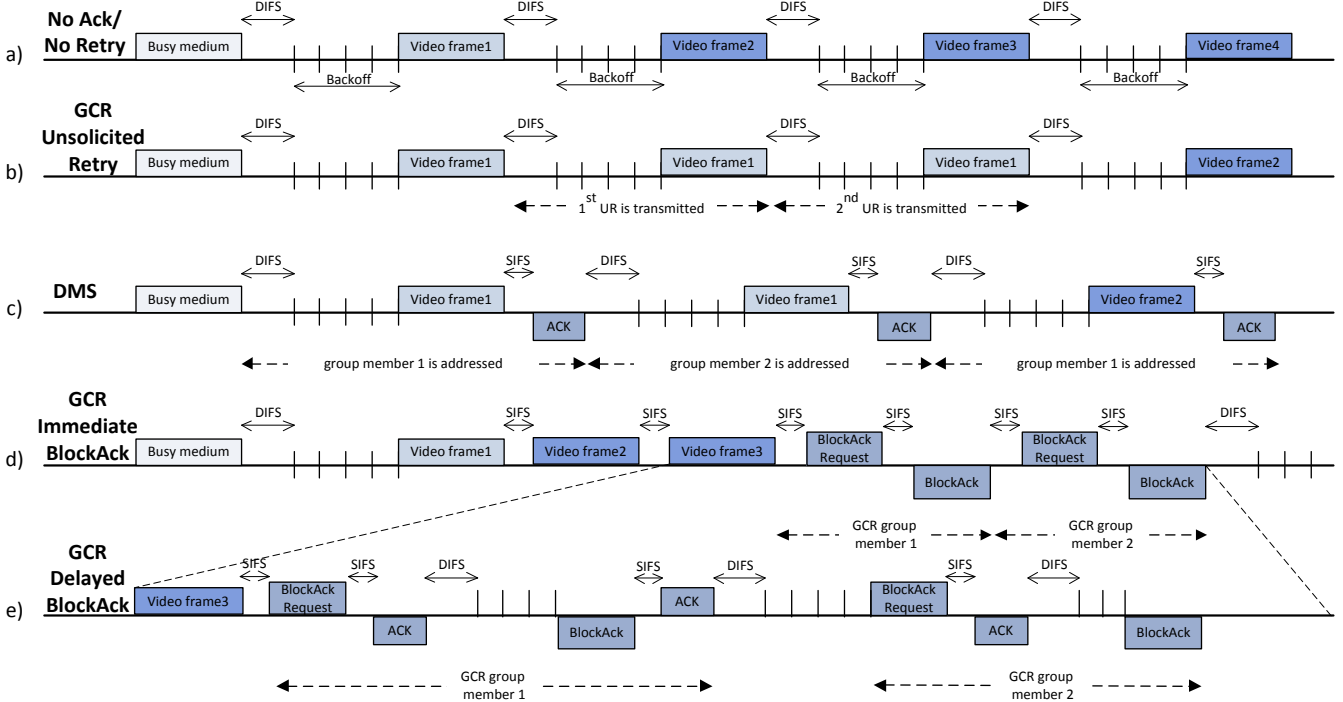


Fig. 1: Legacy IEEE 802.11 multicast and Group Addressed Transmission Service (GATS) mechanisms.

1) *GCR Unsolicited Retry*: This delivery method *preemptively* retransmits a frame one or more times (up to a certain retry limit), to increase the probability of successful delivery at the STAs (see Fig. 1b). The retransmission of these frames is implementation-dependent. In this way, the mechanism aims to improve the reliability of the legacy multicast without introducing the overhead of an acknowledgment mechanism.

2) *Directed Multicast Service*: This mechanism basically converts multicast to unicast (as illustrated in Fig. 1c for two groupcast members). In this way, the frames transmitted to a multicast address are transmitted individually to each of the associated STAs that belong to the multicast group. The individually addressed frames will therefore be retransmitted until an ACK is received by the AP, or the retransmission count limit is exceeded (in which case the frame is discarded). Even though this mechanism provides very high reliability, it is also very inefficient as the resources consumed increase linearly with the number of group members.

3) *GCR Block Ack*: This mechanism extends the Block Ack mechanism (already defined in the current version of the 802.11 standard) to account for group addressed frames as follows. In the original mechanism, the sender transmitted a burst of data frames to one destination, and then explicitly requested an ACK to the receiving station. In this case, the AP transmits a number of data frames to the GCR group address, and then polls each or some of the destination STAs using a *BlockAckRequest* frame, while each recipient of a *BlockAckRequest* replies immediately (after a SIFS time). In the rest of the paper, we will assume that the AP polls each destination before sending a new burst of frames.

4) *GCR Delayed Block Ack*: In addition to the above mechanisms, included in the 802.11aa standard, we will also analyze the following *delayed* version of the GCR Block Ack scheme. While this scheme is not explicitly included as part of the 802.11aa standard, it builds on the standardized Delayed Block Ack operation for unicast, extending it to multicast. Similarly to the previous scheme (which we hereafter will refer to with GCR Immediate Block Ack to avoid confusion), the AP first transmits a burst of frames and then polls each intended destination. However, in this case, both the *BlockAckRequest* and the *BlockAck* frames are acknowledged with an ACK frame, and after receiving a *BlockAckRequest* the recipient starts a backoff process before sending the *BlockAck* frame, c.f., Fig. 1e.

Despite this variety of mechanisms, the amendment provides no guidelines to choose the most efficient for a given scenario. This is the main motivation of this paper, which we tackle in the next two sections: we first provide a performance analysis of each of the mechanisms to understand their benefits and limitations, and then propose an algorithm that selects the best performing algorithm depending on the WLAN scenario.

III. PERFORMANCE ANALYSIS

We consider the WLAN scenario illustrated in Fig. 2, with one AP sending multicast traffic to a group of N_{rx} stations, using mechanism \mathcal{M} , where $\mathcal{M} \in \{No-Ack/No-Retry, GCR UR, DMS, GCR I.BlockAck, GCR D.BlockAck\}$. In the same WLAN, there is another independent set of N_u unicast senders, transmitting data to the AP. The performance metric that we focus on is the guaranteed capacity available for multicast and unicast traffic, which corresponds to the case when

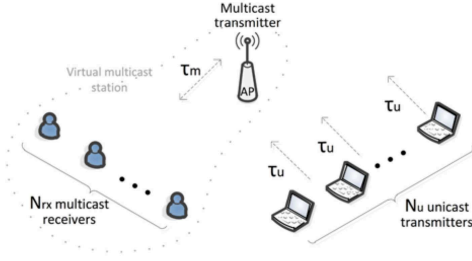


Fig. 2: Considered scenario for the performance analysis

all stations are saturated (i.e., constantly backlogged). This metric gives the throughput guarantees for each traffic type, and thus can be used to ensure a good level of performance to multicast and unicast traffic; indeed, as long as the sending rate is below the saturation throughput, performance will be good.¹

For simplicity, we assume a constant packet payload size for unicast and multicast traffic of L_u and L_m bits, and that both unicast and multicast are in Access Categories that use the Arbitration inter-frame spacing (AIFS) such that $AIFS = DIFS$ (but can use different CW parameters). We further assume that all stations use the same transmission rate.² It is worthwhile noting that, following our previous work of [6], the assumptions of saturation, fixed packet lengths, $AIFS = DIFS$ and equal transmission rates schemes could be easily relaxed, by combining the analysis of the multicast schemes of this paper with (i) the computation in [6] of the slot time durations for variable packet lengths and transmission rates; (ii) the algorithm proposed in [6] to compute the transmission probability of non-saturated stations; and (iii) the analysis of [6] for the $AIFS$ parameter. However, this would require a more complex derivation and would not add much insight to the performance trade-offs of the different multicast schemes, indeed the main focus of this paper.

We let p_{e_i} denote the bit error rate (BER) of receiver e_i for multicast traffic, for $i = 1, \dots, N_{rx}$, and p_{e_u} the BER for unicast (for simplicity, we assume the same BER for all unicast stations). From this, the frame error rate (FER) for multicast receiver e_i and for a unicast station can be computed, respectively, as follows:

$$p_{f_{e_i}} = 1 - (1 - p_{e_i})^{L_m}, \quad (1)$$

$$p_{f_u} = 1 - (1 - p_{e_u})^{L_u}. \quad (2)$$

In addition to channel errors, collisions are the other source of frame losses. Note that, as there is only one multicast sender, collisions can only happen either between unicast stations, or between one multicast station and one or more unicast

stations. As collisions cannot occur between multicast stations, we assume that all multicast activity can be modeled with a single “virtual” station, which captures the aggregate behavior of the multicast sender and the receivers. This “virtual” station attempts transmission with a constant probability τ_m upon a backoff counter decrement in the WLAN (following [24], hereafter we refer to the period between two backoff counter decrements as *slot time*). For unicast traffic, we let τ_u denote the probability that a station transmits in a slot time.

The performance figures of interest are the following. For unicast traffic, as it is saturated, we are only interested in its throughput experienced, denoted as S_u^M . For multicast traffic, we are interested in two performance figures, namely, the *reliability* η_m^M and the *throughput* experienced S_m^M . Note that, in some cases, there may be a trade-off between these two metrics: for instance, with the GCR UR scheme, the larger the number of retransmissions, the higher reliability; however, this decreases throughput as more time is devoted to retransmit old frames rather than transmitting new ones.

While the definition of reliability is rather clear for unicast traffic (a frame is either received or not by its receiver), the definition of this metric for multicast is less clear (as a frame may be received by some receivers but not by others). In this paper, we define reliability as the number of successful receptions, counting all intended destinations, over the total number of receptions if all deliveries had been successful (which is given by the number of frames sent times the number of receivers). For instance, if we consider one frame addressed to N receivers, out of which N_s receive the frame successfully, reliability is given by N_s/N . Following this definition, we can express reliability as $\eta_m^M = \text{\#successful frames}/\text{\#all frames}$, and we can compute it as

$$\eta_m^M = \frac{1}{N_{rx}} \sum_{i=1}^{N_{rx}} P_m^M(i), \quad (3)$$

where $P_m^M(i)$ is the probability that a multicast frame is successfully received by station i .

The throughput experienced by a unicast flow follows the standard definition of the average number of successfully received bits per time unit. For a multicast flow, we define the throughput experienced as the average of the throughput successfully received by each of the stations of the multicast flow, i.e.,

$$S_m^M = \frac{1}{N_{rx}} \sum_{i=1}^{N_{rx}} S_{m,i}^M \quad (4)$$

where $S_{m,i}^M$ is the throughput of the multicast flow successfully received by station i .

In the following, we present the analysis of the above metrics for each of the schemes introduced in Section II. For the case of the GCR schemes, we first analyze their performance for the basic access mode and then explain how the analysis can be extended for the RTS/CTS access mode.³

¹As shown by [22], as long as the sending rate is below the saturation throughput, not only the required throughput will be satisfied, but also the delays experienced by a station will be low. Therefore, the saturation throughput metric serves to guarantee good performance both in terms of throughput and in terms of delay.

²Dynamic rate adaptation in a multicast scenario is a very relevant topic that has attracted substantial attention, like e.g. the recent paper of [23]. This issue, however, it is out of the scope of this paper, which assumes that the rate has already been selected and focuses on the performance taking into account the given transmission rates and the resulting BERs.

³RTS/CTS is mentioned in the standard as a possible protective mechanism to reduce the probability that other transmissions collide with a multicast transmission).

A. No-Ack/No-Retry

With this scheme, each frame is sent only once. Therefore, a multicast frame is received correctly by each of the N_{rx} receivers when it does not collide with other frames and does not suffer from channel errors. Thus, the average multicast reliability can be computed as:

$$\eta_m^{NoAck} = \frac{1}{N_{rx}} \sum_{i=1}^{N_{rx}} (1 - p_{c_m})(1 - p_{f_{e_i}}) = (1 - p_{c_m})(1 - p_f), \quad (5)$$

where $p_f = \sum p_{f_{e_i}}/N_{rx}$ is the average frame error probability across multicast receivers, and p_{c_m} is the conditional probability that a transmission attempt collides, i.e., the probability that when a multicast transmission is taking place, at least one data transmission happens thus resulting in a collision, which is given by

$$p_{c_m} = 1 - (1 - \tau_u)^{N_u}. \quad (6)$$

The saturation throughput achieved by multicast (unicast) traffic using No-Ack/No-Retry S_m^{NoAck} (S_u^{NoAck}) can be computed as the average successfully transmitted payload in a slot time over the average slot time:

$$S_m^{NoAck} = \frac{P_{s_m} L_m (1 - p_f)}{T_{slot}}, \quad (7)$$

$$S_u^{NoAck} = \frac{P_{s_u} L_u}{T_{slot}} (1 - p_{f_u}), \quad (8)$$

where T_{slot} is the average slot time duration, and P_{s_m} (P_{s_u}) is the probability that a randomly chosen slot time contains a multicast (unicast) transmission that did not collide. These probabilities can be expressed in terms of the pair $\{\tau_m, \tau_u\}$ as

$$P_{s_m} = \tau_m (1 - p_{c_m}), \quad (9)$$

$$P_{s_u} = N_u \tau_u (1 - \tau_u)^{N_u - 1} (1 - \tau_m). \quad (10)$$

while the average slot duration T_{slot} in (7), (8) can be computed as,

$$T_{slot} = P_{s_u} T_{s_u} + P_{c_u} T_{c_u} + P_{s_m} T_{s_m} + P_{c_m} T_{c_m} + P_e T_e. \quad (11)$$

where P_e , P_{c_u} and P_{c_m} are the probabilities that a random slot time is empty, contains a collision of only unicast frames or a collision involving a multicast transmission, respectively, T_e , T_{c_u} and T_{c_m} are the slot time durations in each of these cases, and T_{s_u} (T_{s_m}) is the length of a successful unicast (multicast) transmission. For simplicity, we assume that whenever there is a multicast transmission involved in a collision, the transmission length of the multicast frame determines the total time the medium is sensed as busy. Following this, we account in P_{c_u} for the collisions exclusively between unicast stations, which can be computed as:

$$P_{c_u} = (1 - \tau_m) \left[1 - (1 - \tau_u)^{N_u} - N_u \tau_u (1 - \tau_u)^{N_u - 1} \right]. \quad (12)$$

An empty slot occurs when there are no unicast nor multicast transmissions:

$$P_e = (1 - \tau_u)^{N_u} (1 - \tau_m), \quad (13)$$

while the probability of a collision involving multicast traffic can be derived from the previous probabilities:

$$P_{c_m} = 1 - P_{s_u} - P_{c_u} - P_{s_m} - P_e. \quad (14)$$

The time slot durations in (11) can be computed as:

$$T_{s_u} = T_{PLCP} + \frac{H}{R_u} + \frac{L_u}{R_u} + SIFS + T_{PLCP} + \frac{ACK}{R_c} + DIFS \quad (15)$$

$$T_{c_u} = T_{PLCP} + \frac{H}{R_u} + \frac{L_u}{R_u} + DIFS \quad (16)$$

$$T_{s_m} = T_{c_m} = T_{PLCP} + \frac{H}{R_m} + \frac{L_m}{R_m} + DIFS \quad (17)$$

where T_{PLCP} is the Physical Layer Convergence Protocol (PLCP) preamble, H is the MAC header plus the Frame Check Sequence (FCS) length, ACK is the length of the Ack frame, and R_u , R_m and R_c are the channel bit rates for unicast, multicast and control frames, respectively. We note that No-Ack/No-Retry multicast uses for R_m one of the Basic Service Set rates.

Finally, we address the computation of τ_m and τ_u . In order to compute τ_m , recall that multicast traffic is sent in saturation, which in terms of channel access can be modeled as a particular case of DCF with no backoff. Under these conditions, τ_m can be computed as:

$$\tau_m = \frac{2}{CW_{min_m} + 1} \quad (18)$$

where CW_{min_m} is the minimum contention window of multicast traffic.

To compute τ_u , we follow the expression provided in [25]

$$\begin{aligned} \tau_u &= \frac{2(1 - 2p_{e_u})(1 - p_{e_u}^{R+1})}{A + B + C} \\ A &= W_u (1 - (2p_{e_u})^{m_u + 1})(1 - p_{e_u}) \\ B &= (1 - 2p_{e_u})(1 - p_{e_u}^{R+1}) \\ C &= W_u 2^{m_u} p_{e_u}^{m_u + 1} (1 - p_{e_u}^{R - m_u}) \end{aligned} \quad (19)$$

where R is the retry limit, $W_u = CW_{min_u}$ is the minimum contention window, m_u is the maximum backoff stage (note that we assume $R > m_u$), and p_{e_u} is given by

$$p_{e_u} = 1 - (1 - \tau_u)^{N_u - 1} (1 - \tau_m) (1 - p_{f_u}). \quad (20)$$

The expressions (19) and (20) form a system of equations that can be solved numerically, which terminates the analysis of this scheme.

B. GCR Unsolicited Retry

With GCR UR, a frame is lost if all the $R + 1$ transmission attempts suffer from either collisions or channel errors. Accordingly, a multicast frame is correctly received if any of the transmission attempts is successfully delivered. Therefore, the average multicast reliability for GCR UR can be computed as:

$$\eta_m^{UR} = \frac{1}{N_{rx}} \sum_{i=1}^{N_{rx}} \left(1 - (1 - (1 - p_{c_m})(1 - p_{f_{e_i}}))^{R+1} \right) \quad (21)$$

To compute the throughput achieved by the multicast flow when using GCR UR, we proceed as follows. The throughput obtained by receiver i is given by

$$S_{m,i}^{UR} = \frac{\tau_m L_m}{T_{slot}^{UR}} \frac{1}{R+1} \left(1 - (1 - (1 - p_{c_m})(1 - p_{f_{e_i}}))^{R+1} \right) \quad (22)$$

which takes into account that each frame is transmitted $R+1$ times and that its content is successfully received with a certain probability. Combining the above with (4) and (21) yields

$$S_m^{UR} = \frac{1}{R+1} \frac{\tau_m L_m \eta_m^{UR}}{T_{slot}^{UR}} \quad (23)$$

The unicast throughput has a similar expression as in (8),

$$S_u^{UR} = \frac{P_{s_u} L_u}{T_{slot}^{UR}} (1 - p_{f_u}), \quad (24)$$

although the average slot time is different in this case. Indeed, for the computation of T_{slot}^{UR} we have to update in (11) the value of T_{s_m} and T_{c_m} with the modulation and coding scheme used in this case.⁴

Finally, given that unicast stations do not distinguish between the first multicast transmission or any of the retries that follow, τ_u and τ_m are computed as in the previous case, i.e., following (18) and (19).

The above analysis can be easily adapted to account for the use of the RTS/CTS protective mechanism, as suggested by the standard. The required modifications are the following: (i) to set $p_{c_m} = 0$ in (21), as now video frames are lost only due to channel conditions; (ii) to multiply (23) by $(1 - p_{c_m})$, given that only those channel access that do not collide constitute an actual frame transmission; (iii) to update the duration of a successful multicast transmission given by (17) to include the exchange of the RTS/CTS frames; and (iv) to compute τ_m by taking into account the collision probability for RTS frames, p_{c_m} , and their retries limited by the short retry limit different from the unsolicited retry limit R . Note that there is no need to update the duration of a collision (16) as this is still determined by the (longer) unicast transmissions. The accuracy of this extension is confirmed by the results in Section V-F.

C. Directed Multicast

We next analyze the case when the AP transmits a multicast flow to N_{rx} stations using the DMS for each destination. In this case, for each multicast receiver we have a unicast flow under saturation that is transmitted with the DCF access scheme.

To compute the multicast reliability we use the same expression as the obtained for the GCR UR mechanism:

$$\eta_m^{DMS} = \frac{1}{N_{rx}} \sum_{i=1}^{N_{rx}} \left(1 - (1 - (1 - p_{c_m})(1 - p_{f_{e_i}}))^{R+1} \right) \quad (25)$$

⁴In contrast to the No-Ack/No-Retry scheme, with GCR Unsolicited Retry a frame is transmitted more than once. As a consequence, using the transmission rate of No-Ack/No-Retry for GCR Unsolicited Retry would be overly conservative. Indeed, since with the GCR Unsolicited Retry scheme a failed transmission is retransmitted, it is reasonable to assume that higher failure rates can be afforded and hence higher transmission rates are used.

The multicast throughput with DMS can be computed as

$$S_m^{DMS} = \frac{1}{N_{rx}} \frac{\sum_{i=1}^{N_{rx}} \tau_{m,i} L_m (1 - p_{c_m})(1 - p_{f_{e_i}})}{T_{slot}^{DMS}}, \quad (26)$$

where the variable $\tau_{m,i}$ denotes the probability that a randomly chosen slot contains a multicast transmission addressed to station i ,⁵ which can be computed as

$$\tau_{m,i} = \frac{N_{tx_i}}{\sum_{j=1}^{N_{rx}} B_j + N_{tx_j}}, \quad (27)$$

with B_i being the average number of slots counted during the backoff process when transmitting to station i (without counting transmissions), and N_{tx_i} being the average number of attempts for a frame addressed to station i . The latter term is given by

$$N_{tx_i} = \sum_{j=1}^R j (1 - p_{c_m})(1 - p_{f_{e_i}}) (1 - (1 - p_{c_m})(1 - p_{f_{e_i}}))^{j-1} + (R+1) (1 - (1 - p_{c_m})(1 - p_{f_{e_i}}))^R, \quad (28)$$

which can be re-arranged as,

$$N_{tx_i} = \frac{1 - (1 - (1 - p_{c_m})(1 - p_{f_{e_i}}))^{(R+1)}}{(1 - p_{c_m})(1 - p_{f_{e_i}})} \quad (29)$$

and the former by

$$B_i = \sum_{j=1}^R \sum_{k=1}^j \frac{2^{\min(k, m_m)} CW_{min_m} - 1}{2} (1 - p_{c_m})(1 - p_{f_{e_i}}) \cdot (1 - (1 - p_{c_m})(1 - p_{f_{e_i}}))^{j-1} + \sum_{k=1}^{R+1} \frac{2^{\min(k, m_m)} CW_{min_m} - 1}{2} (1 - (1 - p_{c_m})(1 - p_{f_{e_i}})) \quad (30)$$

where m_m is the maximum backoff stage for video traffic. The throughput for unicast traffic when multicast is transmitted using DMS is given by the same expression as in (8)

$$S_u^{DMS} = \frac{P_{s_u} L_u}{T_{slot}^{DMS}} (1 - p_{f_u}) \quad (31)$$

In the throughput expressions of (26) and (31), the average slot length T_{slot}^{DMS} is computed from (11) but, similarly to the previous case, updating T_{c_m} and T_{s_m} to account for channel bit rate resulting from the modulation and coding scheme used, i.e., $R_m = R_u$.

Finally, to compute τ_m and τ_u , we proceed as follows. τ_u is computed as in (19), while τ_m is given by

$$\tau_m = \sum_i \tau_{m,i} \quad (32)$$

⁵This accounts for the fact that those destinations with poor channel conditions will require more retransmissions than those with good channel conditions.

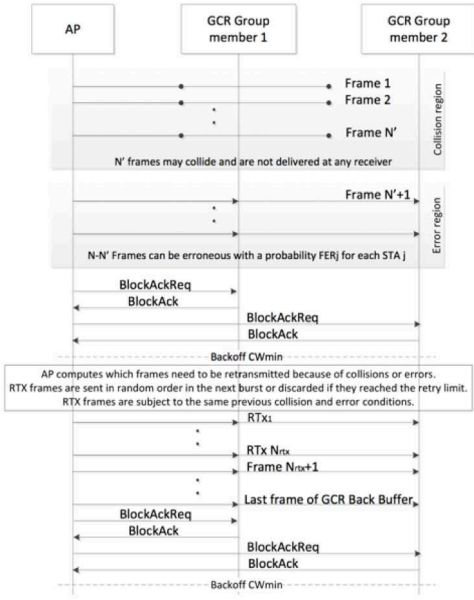


Fig. 3: GCR Immediate Block Ack example scenario

D. GCR Immediate Block Ack

In the following, we analyze the performance of the *GCR Immediate Block Ack* mechanism. As explained in Section II, with this mechanism multicast frames are transmitted in groups of N frames that are sent to all the stations of the multicast group. In what follows, we refer to each of these transmissions of N frames as a multicast *burst*. We represent with τ_m the probability that a multicast burst of N frames is transmitted in a randomly chosen slot, and with p_{cm} the probability of such a multicast burst transmission collides.

In Fig. 3 we depict an example of an AP transmitting multicast to two receivers using GCR Immediate Block Ack. The example assumes that the first multicast burst transmission attempt collides with a unicast transmission (which is not shown), and illustrates that:

- As the unicast transmission duration only covers part of the multicast burst, it only affects the first N' frames, which are lost, while the remaining $N - N'$ frames are not affected. N' can be computed as

$$N' = \left\lceil \frac{T_{PLCP} + (H + L_u)/R_u}{T_{PLCP} + (H + L_m)/R_m + SIFS} \right\rceil \quad (33)$$

- The $N - N'$ frames that *survive* the collision can still be lost due to channel conditions, with an independent per-receiver error probability, p_{fei} .
- Immediately after the end of the burst, each receiver informs the AP of the frames successfully received by means of a *BlockAckRequest*.
- The next multicast burst transmission follows after a backoff. In this transmission, the AP sends N additional multicast frames, a number of which are retransmissions of frames of the previous burst, while the rest are new frames.

As it follows from the above example, in general out of the N frames in a given burst, some will be retransmissions of frames that have not reached successfully all stations in previous attempts, either due to collisions or channel errors, while the others will be new. As the 802.11aa standard does not specify in which position of the burst the retransmitted frames are placed, in the analysis that follows we assume they are placed randomly. While an alternative ordering for the frames of the burst could be based on their sequence number, this would reduce the resulting reliability as a frame that reaches the head of line position would be vulnerable to collisions in all transmission attempts, and thus the probability that such a frame reaches the retry limit would be higher.⁶

With the above assumption, the probability that a randomly chosen frame transmission within a burst is successfully received by station i is given by

$$p(s_i) = \frac{N'}{N}(1 - p_{cm})(1 - p_{fei}) + \left(1 - \frac{N'}{N}\right)(1 - p_{fei}) \quad (34)$$

where the first term corresponds to the case that the frame falls within the first N' frames of the burst, and hence is vulnerable to a collision, and the second term corresponds to the case that the frame is placed with the last $N - N'$ frames. From this,

$$p(s_i) = (1 - p_{fei}) \left(1 - \frac{N'}{N}p_{cm}\right) \quad (35)$$

According to the standard, the number of retransmissions with the Block Ack is not subject to a retry limit but to a MSDU lifetime. For analytical tractability, we model this with a parameter R , which mimics a retry limit and is obtained by dividing the MSDU lifetime by the average time spent between two retries. This is computed as $(1/\tau_m) \cdot T_{slot,m}$ plus the time required to transmit one burst, where $T_{slot,m}$ is the average duration of a slot in which the multicast sender does not transmit (it is computed similarly to (11), but without taking into account the multicast transmissions).

Based on the above, and given that a frame is retransmitted until it has been successfully received by all stations or it reaches R , the reliability of the GCR Immediate Block Ack mechanism can be computed as

$$\eta_m^{I.BlockAck} = \frac{1}{N_{rx}} \sum_{i=1}^{N_{rx}} \left(1 - (1 - p(s_i))^{R+1}\right) \quad (36)$$

We next compute the multicast and unicast throughputs. To calculate the former, we let N_{tx} denote the average number of transmission attempts per frame. With this definition, the average number of frames successfully transmitted per multicast burst is $\left(\frac{N}{N_{tx}}\right) \cdot \eta_m^{I.BlockAck}$, which leads to the following expression for the multicast throughput (already derived in [19]):

$$S_m^{I.BlockAck} = \frac{\tau_m L_m}{T_{slot}^{I.BlockAck}} \cdot \left(\frac{N}{N_{tx}}\right) \cdot \eta_m^{I.BlockAck} \quad (37)$$

⁶Note that the analysis presented here could be adjusted to the case in which frames are ordered based on their sequence number, by taking into account that a frame reaches a position in which it is vulnerable to collisions with a certain probability. From this point on, it suffers from collisions in all attempts until reaching the retry limit.

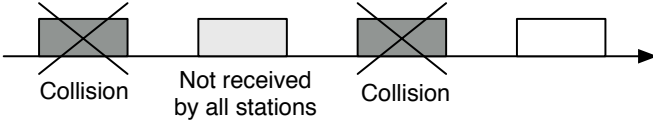


Fig. 4: A frame that is transmitted a total number of $j = 4$ times, because it suffers $k = 2$ collisions (with probability $P_{tx,col}(4, 2)$), and $j - k = 2$ collision-free transmissions were required to be successfully received by all stations (with probability $P_{cf_attempts}(2)$).

The throughput of unicast traffic is given by the same expression as the No-Ack/No-Retry and GCR UR mechanisms (8),

$$S_u^{I.BlockAck} = \frac{P_{su} L_u}{T_{slot}^{I.BlockAck}} (1 - p_{fu}). \quad (38)$$

For the two throughput expressions provided above, we need to compute the average slot durations $T_{slot}^{I.BlockAck}$. This can be computed using the expression of (11) with the following average duration of a multicast successful transmission:

$$T_{sm} = T_{cm} = N \left(T_{PLCP} + \frac{H}{R_m} + \frac{L_m}{R_m} + SIFS \right) + 2N_{rx} \left(T_{PLCP} + \frac{BACK}{R_c} \right) + (2N_{rx} - 1) SIFS + DIFS \quad (39)$$

where $BACK$ is the length of a BlockAckRequest or a BlockAck.

The remaining challenge to compute the above throughputs is to obtain the average number of required transmissions per frame N_{tx} in (37). This number can be computed from

$$N_{tx} = \sum_{j=1}^R j P_{tx}(j), \quad (40)$$

where $P_{tx}(j)$ denotes the probability that the AP transmits a frame exactly j times, due to collisions or channel errors. To obtain this probability, we sweep for each value j along the possible numbers of collisions k . Note that, unless we reach the retry limit R , k can be at most $j - 1$ since at least one collision-free attempt is required. For each k value, we compute the following two probabilities:

- $P_{tx,col}(j, k)$: This is the probability that out of the j transmission attempts, k result in collisions.
- $P_{cf_attempts}(j - k)$: This is the probability that exactly $j - k$ collision-free attempts are required for all nodes to successfully receive the frame (i.e., given that k out of the j attempts collide, the remaining $j - k$ are sufficient to successfully reach all receivers).

We depict an example illustrating these two probabilities in Fig. 4. Based on the above, $P_{tx}(j)$ can be computed as

$$P_{tx}(j) = \begin{cases} \sum_{k=0}^{j-1} P_{tx,col}(j, k) P_{cf_attempts}(j - k), & j < R \\ 1 - \sum_{l=1}^R \sum_{k=0}^{l-1} P_{tx,col}(j, k) P_{cf_attempts}(j - k), & j = R \end{cases} \quad (41)$$

To compute $P_{tx,col}(j, k)$, we note that the number of collided attempts can be modeled after a binomial distribution, in which the probability of a collision is $p_{cm} N' / N$ (i.e., the probability that a collision occurs multiplied by the probability that the frame is affected by the collision). Based on this, the probability of k collisions is given by

$$P_{tx,col}(j, k) = \binom{j}{k} \left(\frac{N'}{N} p_c \right)^k \left[1 - \left(\frac{N'}{N} p_c \right) \right]^{j-k}. \quad (42)$$

To compute $P_{cf_attempts}(j - k)$, we first compute the probability that $j - k$ transmissions or less are required for a frame to be received by all stations, which is denoted as $\sum_{n=1}^{j-k} P_{cf_attempts}(n)$. For a given station i , the probability that $j - k$ transmission attempts are not successful is $p_{fe_i}^{j-k}$. Base on this, the former can be computed as

$$\sum_{n=1}^{j-k} P_{cf_attempts}(n) = \prod_{i=1}^{N_{rx}} \left(1 - p_{fe_i}^{j-k} \right), \quad (43)$$

and $P_{cf_attempts}(j - k)$ can be computed as (already derived in [19])

$$P_{cf_attempts}(j - k) = \begin{cases} \prod_{i=1}^{N_{rx}} \left(1 - p_{fe_i}^{j-k} \right) - \prod_{i=1}^{N_{rx}} \left(1 - p_{fe_i}^{j-k-1} \right), & j < R \\ 1 - \prod_{i=1}^{N_{rx}} \left(1 - p_{fe_i}^{j-k-1} \right), & j = R \end{cases} \quad (44)$$

Finally, we need to compute τ_m and τ_u . Since with GCR Immediate Block Ack the CW is always CW_{min_m} , the probability of transmitting a multicast burst τ_m can be computed from (18). Similarly, the transmission probability for unicast frames τ_u can be computed by using (19), which terminates the analysis for this mechanism.

Similarly to GCR UR, in case RTS/CTS is used the analysis has to be adapted by (i) setting $N' = 0$ in (35), as a burst of frames cannot suffer a collision; (ii) multiplying (37) by $(1 - p_{cm})$; (iii) updating the duration of a successful multicast transmission T_{sm} in (39) to include the exchange of the RTS/CTS frames; (iv) instead of using T_{cm} as given by (39), we set the length of a collision to that of a unicast transmission (15); and (v) we compute τ_m by taking into account the collision probability for multicast frames, p_{cm} , and the retry limit for RTS/CTS.

E. GCR Delayed Block Ack

In contrast to the previous scheme, with Delayed Block Ack a backoff process is executed for each BlockAckRequest frame sent by the AP and the corresponding BlockAck sent by a station as a reply. However, given that at a given point in time there is only one backoff process being executed (either by the AP or by a station), we can model (like in the previous sections) all multicast-related activity with a single virtual station. This station is constantly backlogged, i.e., it is always executing a backoff process, and transmits sequentially multicast bursts, BlockAckRequests and BlockAcks.

According to the above, the virtual station first transmits a multicast burst of N frames and a BlockAckRequest (sent a *SIFS* after the multicast burst) and then transmits $2N_{rx} - 1$ control frames (BlockAcks and BlockAckRequests). In contrast to the initial multicast burst, the control frames are retransmitted in case of collision.⁷ Given that transmissions from the multicast station collide with probability p_{cm} , on average control frames are transmitted $1/(1-p_{cm})$ times. Based on this, the probability that a randomly chosen transmission attempt from the virtual multicast station is a multicast burst (p_b) can be computed as:

$$p_b = \frac{1}{1 + \frac{1}{1-p_{cm}}(2N_{rx} - 1)}. \quad (45)$$

The expression for the reliability is the same as in the previous case, i.e., (36), while the expression for the multicast throughput has to be updated to account for the fact that only a fraction of the multicast transmissions (given by p_b) corresponds to actual multicast data, which leads to

$$S_m^{D.BlockAck} = \frac{\tau_m p_b L_m}{T_{slot}^{D.BlockAck}} \cdot \left(\frac{N}{N_{tx}}\right) \cdot \eta_m^{I.BlockAck} \quad (46)$$

On the other hand, the throughput for unicast traffic is given by the same expression as (8) but updating the average slot duration:

$$S_u^{D.BlockAck} = \frac{P_{su} L_u}{T_{slot}^{D.BlockAck}} (1 - p_{fu}) \quad (47)$$

In both throughput expressions, $T_{slot}^{D.BlockAck}$ needs to be updated to account for the two types of multicast transmissions, i.e., bursts of N frames and control frames:

$$\begin{aligned} T_{slot}^{D.BlockAck} &= P_{su} T_{su} + P_{cu} T_{cu} \\ &+ \tau_m p_b T_{Nframes} \\ &+ \tau_m (1 - p_b) (1 - p_{cm}) T_{BlockAcks} \\ &+ P_e T_e + \tau_m (1 - p_b) p_{cm} T_{cu} \end{aligned} \quad (48)$$

where the last term of the above equation accounts for a collision between a multicast control frame and unicast, whose duration is given by the multicast frame. The terms $T_{Nframes}$ and $T_{BlockAcks}$ are computed as

$$T_{Nframes} = N \left(T_{PLCP} + \frac{H}{R_m} + \frac{L_m}{R_m} + SIFS \right) + DIFS - SIFS \quad (49)$$

$$\begin{aligned} T_{BlockAcks} &= T_{PLCP} + \frac{BACK}{R_c} + SIFS + T_{PLCP} \\ &+ \frac{ACK}{R_c} + DIFS \end{aligned} \quad (50)$$

We next address the computation of the transmission probabilities τ_u and τ_m . For the case of unicast stations, the transmission probability τ_u is the same as the one given by (19). To compute the transmission probability of the virtual station, we need to take into account that even though a backoff process is executed both for multicast bursts and

⁷As the control BlockAck frames are sent at the Basic Rate Set, the transmission error probability is very low and neglected in our analysis.

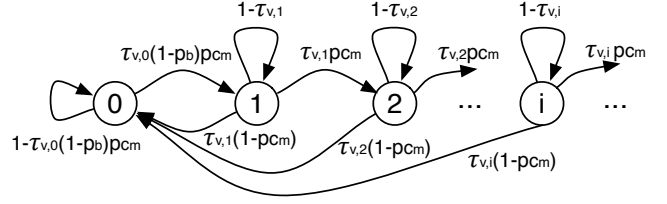


Fig. 5: Markov chain model of the virtual multicast station.

control frames, the CW is doubled *only* after a control frame collision.

We model the above behavior using the Markov Chain depicted in Fig. 5, in which each state represents the backoff stage of the virtual multicast station. Thus, at state 0 the CW value of the virtual station is CW_{min_m} and hence it transmits with probability:

$$\tau_{m,0} = \frac{2}{CW_{min_m} + 1}, \quad (51)$$

and at state i the CW has been doubled i times and the transmission probability can be approximated as [26]:

$$\tau_{m,i} \approx \frac{\tau_{m,0}}{2^i} \quad (52)$$

For state 0, there are two possible outcomes: if the station transmits (which happens with probability $\tau_{v,0}$), the transmission does not correspond to a multicast burst ($1 - p_b$) and it collides (p_{cm}), we move to state 1; otherwise, we stay in state 0. For all other states, there are three possible outcomes per backoff counter decrement: (i) the station does not transmit (which happens with probability $1 - \tau_{m,i}$) and remains in that state; (ii) it performs a successful transmission ($\tau_{m,i}(1-p_{cm})$) and we move to stage 0; or (iii) there is a transmission that collides ($\tau_{m,i}p_{cm}$) and we move to the next stage.

If we denote by p_i the probability of being at stage i , τ_m can be computed as:

$$\tau_m = \sum_{i=0}^{\infty} \tau_{m,i} p_i \quad (53)$$

From the Markov chain, we can compute p_1 as

$$p_1 = p_0 \tau_{m,0} (1 - p_b) p_{cm} + p_1 (1 - \tau_{m,1}) \quad (54)$$

and p_i for $i > 0$ as:

$$p_i = p_{i-1} \tau_{m,i-1} p_{cm} + p_i (1 - \tau_{m,i}) \quad (55)$$

From the above,

$$p_i = (2p_{cm})^i (1 - p_b) p_0 \quad (56)$$

Combining this with $\sum_{i=0}^{\infty} p_i = 1$ yields

$$p_0 + \sum_{i=1}^{\infty} (2p_{cm})^i (1 - p_b) p_0 = 1 \quad (57)$$

from which

$$p_0 = \frac{1 - 2p_{cm}}{1 - 2p_{cm}p_b} \quad (58)$$

From equations (53) and (56):

$$\begin{aligned}\tau_m &= \tau_{v,0} p_0 + \sum_{i=1}^{\infty} \frac{\tau_{v,0}}{2^i} (2p_{c_m})^i (1-p_b) p_0 \\ &= \tau_{v,0} p_0 \left(\frac{1-p_{c_m} p_b}{1-p_{c_m}} \right)\end{aligned}\quad (59)$$

Substituting $\tau_{v,0}$ and p_0 we get:

$$\tau_m = \left(\frac{2}{CW_{min_m} + 1} \right) \left(\frac{1-2p_{c_m}}{1-2p_{c_m} p_b} \right) \left(\frac{1-p_{c_m} p_b}{1-p_{c_m}} \right) \quad (60)$$

Finally, numerical techniques can be used to solve the non-linear system given by equations (60), (19), (6) and (20), which terminates the analysis. We note that the analysis can be extended to account for the use of RTS/CTS by performing the same extensions as described with the Immediate scheme.

IV. SELECTION ALGORITHM

The analysis conducted in the previous section allows to compute the performance of each of the multicast mechanisms as a function of the number of multicast and unicast stations in the WLAN. In the following, we propose an algorithm that exploits this analysis to select the best multicast mechanism for a given scenario. While this algorithm is one particular example of how our analysis can be used to select a multicast scheme based on a specific metric, it is worthwhile noting that the analysis could also be used to devise alternative algorithms that optimize other performance metrics and/or guarantee different constraints.

For the design of our selection algorithm, we set the following goals:

- The reliability provided to multicast traffic by the selected mechanisms needs to be above a certain minimum threshold, i.e., $\eta_m^{\mathcal{M}} > \eta_{min}$.
- As long as the minimum reliability requirement is satisfied, we would like to maximize the throughput per-station guarantees provided to multicast and unicast, i.e., $S_m^{\mathcal{M}}$ and $S_u^{\mathcal{M}}/N_u$.

The rationale behind the goal on reliability is that if the drop rate of multicast traffic is higher than that corresponding to this threshold, then the resulting quality will be unacceptable. The setting of η_{min} depends on the type of content transported by the multicast flow. For instance, for video traffic a reliability of $\eta_{min} = 0.9$ (which means a drop rate of 10%) is typically considered sufficient [27], [28].

As for the goal on throughput, if one mechanism provides a larger throughput both to multicast and unicast than another one, the former is clearly more desirable (provided that both satisfy the reliability requirement). However, if one mechanism provides more throughput to one traffic type (multicast or unicast) but less to the other type, then we need a criterion to weight them and perform a comparison. A similar issue is addressed when allocating bandwidth to unicast and multicast flows in [29], which suggests the following possible criteria:

- 1) The *Receiver Independent (RI)* criterion gives the same weight to unicast and multicast flows, and has the drawback that multicast is not given a higher weight

even though it benefits a larger number of users and hence contributes more to the overall utility of the system.

- 2) The *Linear Receiver Dependent (LinRD)* criterion gives a weight to each flow proportional to the number of receivers; while it prioritizes those flows with more receivers, it is very unfair towards unicast flows, which may suffer from starvation if there are many multicast receivers.
- 3) The *Logarithmic Receiver Dependent (LogRD)* criterion gives a weight to each flow that depends logarithmically on the number of receivers; as shown in [29], this criterion provides a good trade-off between prioritizing multicast and avoiding the starvation of unicast.

Following the above, in this paper we adopt the *LogRD* criterion to weight unicast against multicast. In particular, we consider that the fair allocation in the system is the one that maximizes

$$\min \left(\frac{S_m^{\mathcal{M}}}{w_m}, \frac{S_u^{\mathcal{M}}/N_u}{w_u} \right) \quad (61)$$

where the weights for multicast and unicast are $w_m = 1 + \ln N_{rx}$ and $w_u = 1$, respectively.

From the above, we consider that the utility provided by a multicast scheme, $U^{\mathcal{M}}$ is given by the one that provides the best allocation according to the above metric while satisfying the requirement on reliability. Thus,

$$U^{\mathcal{M}} = \begin{cases} \min (S_m^{\mathcal{M}}/w_m, (S_u^{\mathcal{M}}/N_u)/w_u), & \eta_m^{\mathcal{M}} \geq \eta_{min} \\ 0, & \text{otherwise} \end{cases} \quad (62)$$

The proposed *selection algorithm* then works as follows. Given a scenario with a certain number of multicast and unicast stations and their frame lengths, we compute for each multicast scheme \mathcal{M} the reliability $\eta_m^{\mathcal{M}}$ and the multicast and unicast saturation throughputs, $S_m^{\mathcal{M}}$ and $S_u^{\mathcal{M}}$. Based on these metrics, we then compute that utility $U^{\mathcal{M}}$ provided by each mechanism following (62), and select the mechanism that provides the largest utility, i.e.,

$$\text{Selected mechanism} = \max_{\mathcal{M}} \{U^{\mathcal{M}}\} \quad (63)$$

which terminates the selection of the mechanism.

V. PERFORMANCE EVALUATION

In this section, we validate the accuracy of the model by comparing the analytical values against those obtained using simulations. Our simulator is built on the OMNeT++⁸ framework and extends the tool used in our previous works [6], [30],⁹ which accurately models the behavior of the 802.11 MAC protocol.

Our simulations focus on the scenario depicted in Fig. 2, in which the AP transmits a multicast flow to N_{rx} receivers, and N_u stations transmit unicast data to the AP. The length of multicast and unicast frames is set to 1500 B. Each simulation

⁸<http://www.omnetpp.org>

⁹The simulation tool is available at <http://labs.netcom.it.uc3m.es/~ppatras/owsim/>

value corresponds to the average of 10 runs (we confirmed that with this number of runs, 95%-confidence intervals are well below 1%). Table I summarizes the parameters used for the different mechanisms evaluated (their choice has been inspired in the default configuration of the standard as well as by the findings of other works).

A. Multicast Throughput

We start our performance evaluation with the analysis and validation of the throughput obtained by multicast traffic, S_m . In order to understand the impact on performance of the number of receivers and the number of unicast stations, we perform a sweep on both variables: first we set $N_u = 10$ and vary N_{rx} from 1 to 30 and then we set $N_{rx} = 10$ and perform the sweep on N_u . The results are given in Figs. 6a and 6b, respectively. We use lines to represent analytical values and points to depict the simulation ones; we observe that in all cases simulation results follow very closely the analytical ones, which confirms the accuracy of our analysis.

We first analyze the results with the No-ACK/No-Retry and GCR UR mechanisms, which according to Fig. 6a show the same qualitative behavior. Indeed, their behavior does not depend on any feedback from the receivers, and therefore their performance does not vary with N_{rx} . Furthermore, their performance is also quantitatively very similar. This is caused by the configuration of their parameters: despite the GCR UR mechanism uses a higher MCS, it has to retransmit every frame $R = 2$ times, which significantly reduces its throughput.

The performance of the DMS mechanism is the lowest in all considered scenarios. This is caused by its poor scalability, since the number of unicast flows the AP has to transmit increases with N_{rx} and therefore, even though the aggregated throughput in saturation may be large, the per-station throughput S_m decreases with the number of receivers. The presence of unicast traffic further chokes throughput, as unacknowledged frames are retransmitted until they are received or the retry limit is reached. Based on these results, we conclude that DMS is only potentially suitable for scenarios with very low number of stations (this will be confirmed in Section V-H).

Finally, the two GCR BlockAck mechanisms show a very different behavior despite their similar design. The Immediate scheme provides the largest throughput of all schemes (by far) in all considered scenarios, while with the Delayed version, S_m is reduced by approximately two thirds. The reasons for the good performance of the Immediate scheme are two-fold: first, the ‘‘Burst Size’’ configuration guarantees that whenever the AP accesses the channel, it obtains a large share of the resources; and second, the exchange of control frames cannot be interrupted by unicast stations. In contrast, with the Delayed version not only more channel time is spent on control messages due to the acknowledgments, but also stations have to defer channel access and compete with unicast stations to transmit these frames, which may result in collisions and retransmissions.

B. Unicast Throughput

Next, we analyze the throughput performance of unicast traffic, S_u . To this end, we proceed as in the previous section:

first, we fix N_u and perform a sweep over N_{rx} , and then we fix N_{rx} and perform a sweep over N_u . The results are depicted in Figs. 7a and 7b, respectively. Again, simulation results match analytical ones very closely, which further confirms the accuracy of the analysis.

In contrast to the multicast throughput, for the case of the unicast throughput the No-Ack/No-Retry and GCR UR mechanisms show the same qualitative behavior but different quantitative results. Indeed, Fig. 7a confirms that the performance does not depend on the number of receivers for both schemes, but the throughput obtained by unicast stations with No-Ack/No-Retry is approximately half of that obtained by GCR UR. The reason is that with the No-Ack/No-Retry mechanism, data frames are sent with the lowest MCS and therefore, due to the *performance anomaly*, the throughput for all stations in the WLAN is severely degraded. Similar results are obtained in Fig. 7b; in that figure, we observe that the difference between the two mechanisms decreases with the number of unicast stations, as the impact of the performance anomaly is reduced.

For the DMS mechanism, we observe that when the number of unicast stations is fixed to $N_u = 10$, their throughput S_u is not affected by the number of video receivers N_{rx} (see Fig. 7a). This is because with DMS, the AP behaves like a single saturated station, regardless of N_{rx} . On the other hand, when N_{rx} is fixed to 10 and N_u varies (Fig. 7b), the throughput S_u first increases (with the activity on the WLAN) and then decreases (as the number of stations becomes too large leading to many collisions).

Finally, for the case of the GCR BlockAck schemes, we observe that the Immediate mechanism provides the lowest S_u values. This result is expected, as this mechanism provided the largest values for S_m (Fig. 6). We further observe that the throughput decreases with N_{rx} , which is caused by the fact that the channel time required by the exchange of control messages grows with the number of receivers. In contrast, the throughput obtained by unicast traffic with the Delayed mechanism increases with N_{rx} . This is caused because, with the Delayed scheme, the larger the number of receivers, the higher the probability that a multicast access corresponds to a (short) Block Ack exchange rather than a (long) burst of multicast frames (the frequency of multicast accesses does not increase). As a result, when the number of receiver grows, the average duration of multicast transmissions decreases, reducing the total channel time devoted to multicast, which in its turn increases the time available for unicast.

C. Multicast Reliability

We next analyze the remaining metric of interest, which is the reliability η_m obtained by multicast traffic. Like in the previous sections, we fix N_{rx} (N_u) and perform a sweep on N_u (N_{rx}). The corresponding analytical and simulation results are provided in Figs. 8a and 8b.

We first analyze the impact of the number of receivers on the reliability. According to the results of Fig. 8a, this impact is practically negligible. The reasons for this behavior are as follows. For the ‘‘open loop’’ schemes, namely No-Ack/No-Retry and GCR UR, performance is not affected by N_{rx} as

TABLE I: Parameters used in the simulations.

Mechanism	No-Ack/No-Retry	GCR UR	DMS	GCR BlockAck	Unicast
Data Rate (Mbps)	6	54			54
Bit Error Rate	10^{-6}		10^{-5}		0
Control Rate (Mbps)	-		8	6	6
Retries/MSDU lifetime ^a	0	2 ^b	8	400 ms ^c	8
Burst Size	-			10 ^d	-
$\{CW_{min}, m\}$	16, 0				32, 5

^a MSDU lifetime for the Block Ack mechanisms / Number of unsolicited retries configured for the GCR Unsolicited Retry mechanism / Maximum number of retries for the other mechanisms.

^b According to our previous work [31], larger values of R only increases reliability marginally while degrading efficiency severely.

^c We set up the MSDU lifetime to the maximum delay recommended by ITU-T G.114 for interactive real-time multimedia. This is the most conservative choice as it ensures that a frame that could possibly be useful at its destination is never discarded.

^d Following [32], the saturation throughput does not significantly increase for burst sizes longer than $N \in \{8, \dots, 12\}$.

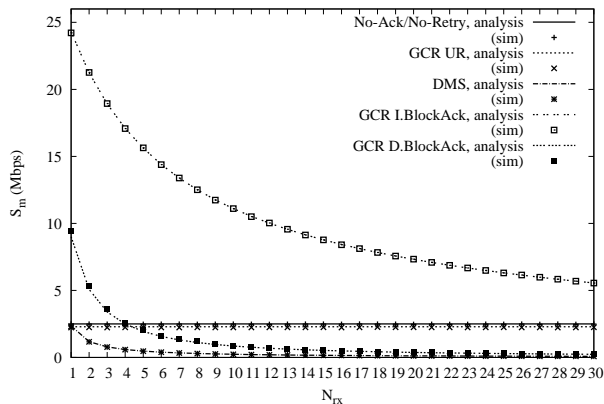
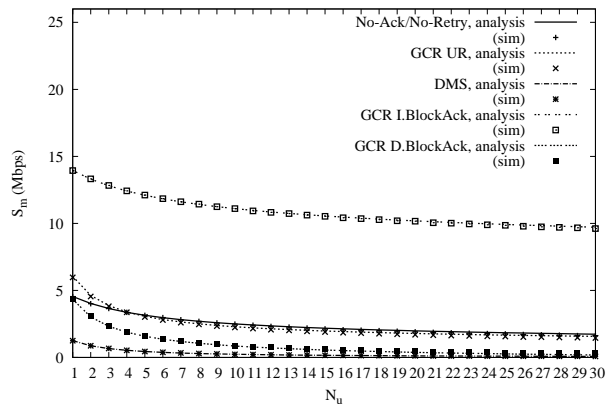
(a) Throughput vs. number of receivers, $N_u = 10$.(b) Throughput vs. number of unicast stations, $N_{rx} = 10$

Fig. 6: Multicast throughput.

we have seen in the previous section. Additionally, for the “feedback-based” schemes such as DMS and GCR BlockAck, performance does depend on N_{rx} either: given the ability of these schemes to recover from packet losses, reliability is 100% in all cases.¹⁰

We now focus on the impact of the number of contending unicast stations on η_m , shown in Fig. 8b. Here, we observe that the two GCR BlockAck based schemes provide a good performance, as in all cases the reliability is 100%. In contrast, the open loop schemes see their performance degraded as N_u grows as a result of the increased channel contention. Comparing GCR UR against No-Ack/No-Retry, we observe that the impact of the increased unicast activity is more severe on the latter, as with the former the unsolicited retries are able to limit the performance degradation to some extent. For instance, with $N_u = 30$ stations, reliability is approximately 55% with No-Ack/No-Retry while with GCR UR it is well above 80%.

On a more general basis, we observe from the above two figures that, while the open-loop schemes perform relatively well in terms of throughput (specially when the number of receivers or unicast stations is large), their reliability in terms

of performance is poor. We also observe that, even though all feedback-based schemes provide the same reliability, their performance in terms of multicast throughput is very different (as shown by Fig. 6a). This confirms the need to account for both metrics when analyzing the performance.

D. Impact of Bit Error Rate

The reliability metric for the different multicast schemes is clearly impacted by the frame error rate (FER) of the channel, as precisely the various algorithms implemented by those mechanisms aim at recovering from such errors. Fig. 9 shows the impact of the FER on the reliability of multicast. We observe from the results that (i) the trends observed in the previous sections for the different schemes hold independent of the FER value; (ii) even for a FER as high as 20%, feedback-based schemes keep a 100% reliability; and (iii) even though for open-loop mechanisms reliability decreases with the FER, degradation is rather soft, which shows that collisions have a higher impact on reliability than transmission errors.

E. Impact of TXOPlimit

The configuration of the TXOPlimit parameter is critical for the performance of the Block Ack schemes. Indeed, the

¹⁰It is worthwhile noting that for the BlockAck schemes, reliability may be lower for a less conservative choice of the MSDU lifetime.

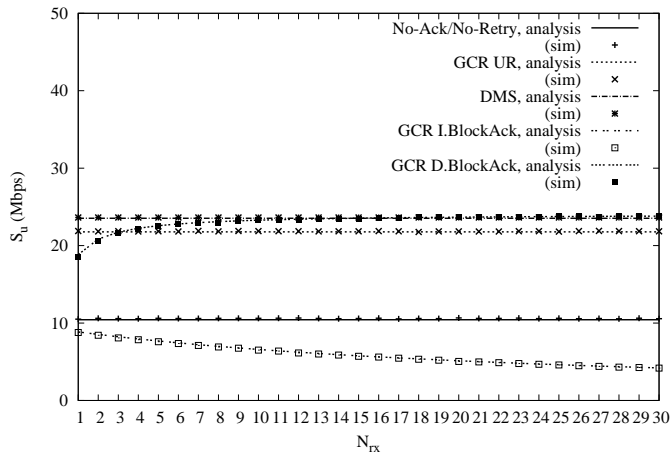
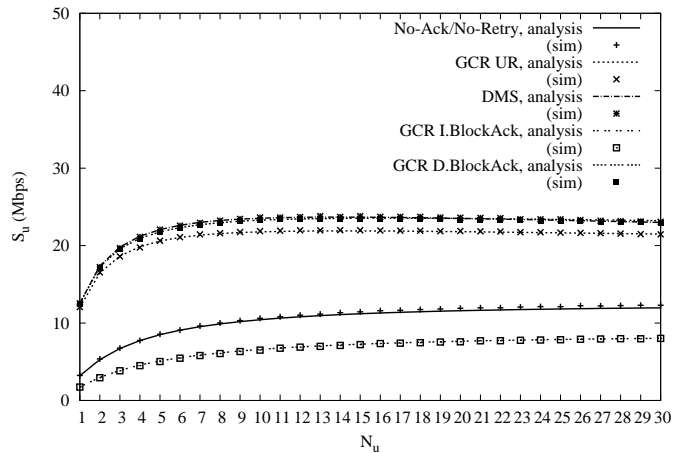
(a) Throughput vs. number of receivers, $N_u = 10$.(b) Throughput vs. number of unicast stations, $N_{rx} = 10$

Fig. 7: Unicast throughput.

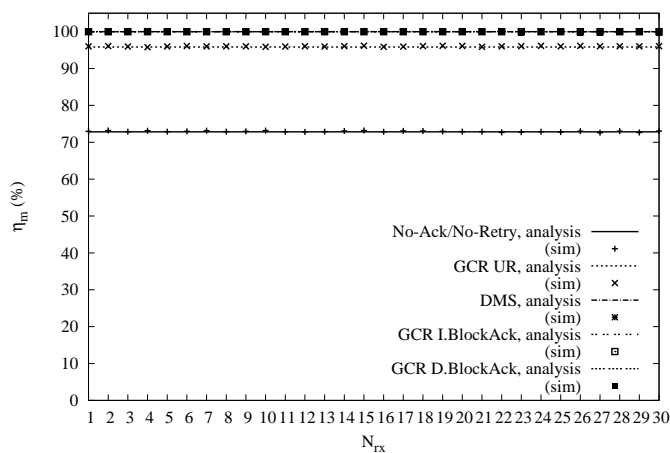
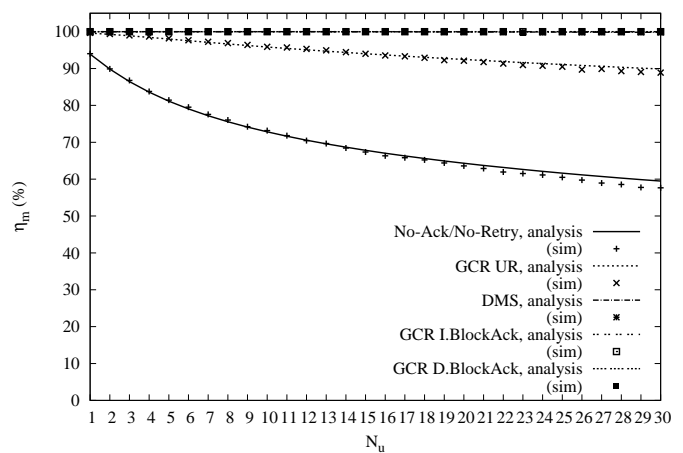
(a) Reliability vs. number of receivers, $N_u = 10$.(b) Reliability vs. number of unicast stations, $N_{rx} = 10$

Fig. 8: Multicast reliability.

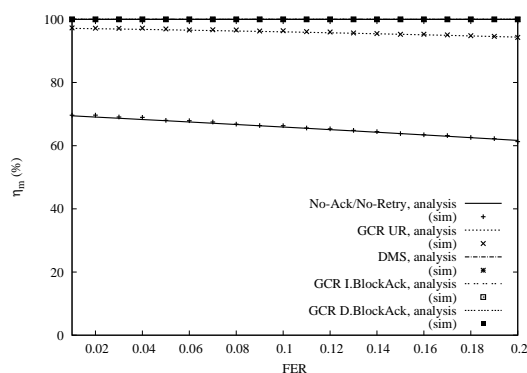


Fig. 9: Impact of the Frame Error Rate on the reliability of multicast.

higher the setting of this parameter, the larger priority is given to multicast traffic over unicast. In order to assess the impact of this parameter, Fig. 10 shows the throughput of

a multicast and a unicast station for the two Block Ack mechanisms (Immediate and Delayed) with $N_{rx} = N_u = 10$. We observe from the figure that (i) this parameter serves to set the level of priority for unicast and multicast, in particular for the Immediate scheme, and (ii) the sum of the unicast and multicast throughputs increases slightly with this parameter, e.g., from 13.5 Mbps with a burst size of 2, to 17.7 Mbps with a burst size of 10 (GCR I.BlockAck). These results suggest that a closer analysis to derive the optimal setting of this parameter may be worth.

F. RTS/CTS access mechanism

The standard mentions RTS/CTS as a possible protective mechanism for the GCR service (i.e., UR and Block Ack). In order to evaluate the performance resulting from using these access mechanisms, Fig. 11 shows two subplots with the multicast reliability (top) and throughput (bottom) for these schemes, when $N_{rx} = 10$ and $N_u \in \{1, \dots, 30\}$. We observe from the figure that, as compared to the basic access mecha-

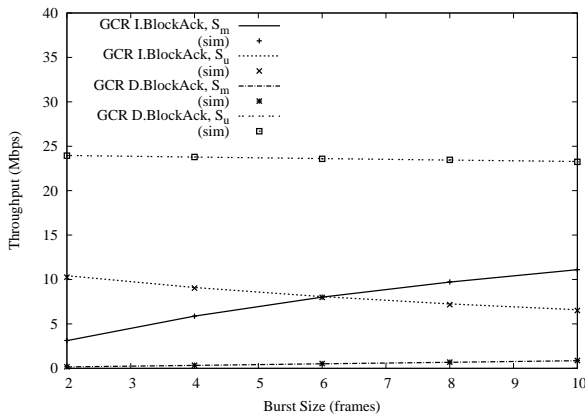


Fig. 10: Impact of the Burst Size on unicast and multicast throughput with GCR Block Ack schemes.

nism, reliability is maximized for UR as a result of avoiding the collision of video frames (the performance of Block Ack is kept at 100%). The price to pay for this improvement is a slightly decrease of throughput, due to the longer frame exchange required for successful transmissions. Other than that, the trends observed for the different mechanisms in the previous subsections do not change significantly. Results for a varying N_{rx} (not reported here for space reasons), lead to a similar conclusions.

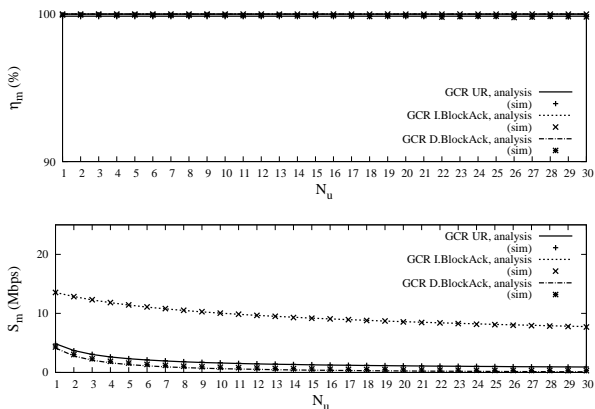


Fig. 11: Performance of GCR schemes with RTS/CTS.

G. Non-saturation conditions

One of the key performance metrics that we target in this paper is the saturation throughput. This metric gives the available throughput that a given multicast/unicast station can use in the worst case (when all other stations are saturated). Hence, this metric serves to optimize the throughput guarantees provided to multicast and unicast stations.

In order to show this, we set up a scenario in which the AP generates multicast traffic at rate R_m , and unicast stations generates traffic at a rate $R_u = R_m/10$. We report in Fig. 12 the resulting throughput allocation for unicast and multicast traffic, as a function of this sending rate. We observe from the figure that all stations are guaranteed their saturation

throughput, as they always see their needs satisfied as long as their sending rate is below the saturation throughput (e.g., approximately 12 Mbps for the No-Ack/No-Retry, 22 Mbps for the case of GCR UR). Additionally, in some cases stations may receive a throughput larger than the saturation throughput, but this only happens when other stations are not using all their saturation throughput [33]. Thus, this figure illustrates the usefulness of the saturation throughput as performance metric for the purpose of this paper.

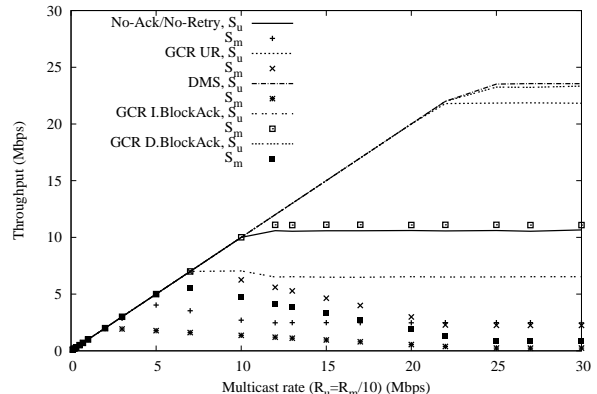


Fig. 12: Throughput with non-saturation conditions.

H. Algorithm Selection

To provide some insights on the performance of the proposed algorithm to select the *best* multicast scheme, we consider a WLAN scenario with 12 unicast stations¹¹ and perform a sweep on the number of multicast receivers. For each configuration, we compute the throughput obtained by multicast and unicast traffic. The results depicted in the top plot of Fig. 13 (the results for each mechanism \mathcal{M} are plotted using the same color, but with different line styles to distinguish between S_m and S_u).

The results show the same qualitative performance that we observed in the previous sections. For instance, S_m is maximized using GCR I.BlockAck while S_u is maximized with DMS. A main conclusion from the figure is that performance heavily varies with the different mechanisms, and a mechanism that performs better with one metric is likely to perform worse with others. This shows that in order to select the best mechanism need to use a criterion that provides a good balance between the multicast and unicast throughputs, S_m and S_u , and the reliability η_m (not shown in the figure).

In the bottom of Fig. 13 we depict the resulting values of the utility $U^{\mathcal{M}}$ as defined by (62), setting the minimum threshold for reliability equal $\eta_{\min} = 90\%$, which is a widely accepted criterion in the literature to guarantee a high quality for video [27], [28]. The main conclusions can be summarized as follows:

¹¹The number of unicast stations has been chosen after evaluating the performance of different scenarios and selecting the one that provides the most illustrative results.

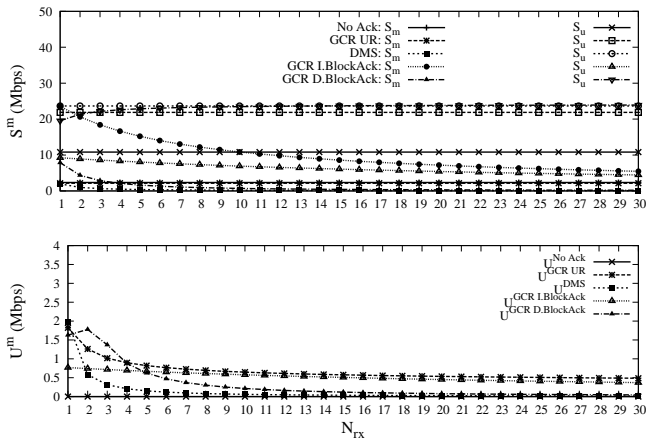


Fig. 13: Multicast and unicast throughput (top) and overall utility (bottom) vs. N_{rx} for $N_u = 12$

- The No-Ack/No-Retry mechanism does not guarantee the minimum threshold for the reliability, and therefore its utility is zero for all considered cases.
- The LogRD criterion is able to provide a trade-off between S_u and S_m , as e.g. for the I.BlockAck mechanism performance is determined by S_u while for the D.BlockAck mechanism it is determined by S_u for $N_{rx} \leq 2$ and by S_m for $N_{rx} > 2$.
- The mechanism that provides the highest U^M varies with N_{rx} : it is the GCR UR mechanism for $N_{rx} \in \{1, 4, 5, 28, 29, 30\}$, D.BlockAck for $N_{rx} \in \{2, 3\}$ and I.BlockAck for $N_{rx} \in \{6, 7, \dots, 27\}$.

In addition to the above conclusions, we also observe from Fig. 13 that the GCR D.BlockAck scheme has a peak at $N_{rx} = 2$. This is caused by the fact that, as explained in Section V-B, for this mechanism multicast throughput decreases with the number of receivers while unicast throughput increases. Thus, for a small number of receivers utility is dominated by unicast traffic whose throughput increases, while for a larger number of receivers it is dominated by multicast traffic whose throughput decreases.

In order to gain a more comprehensive understanding of the behavior of the proposed selection algorithm, we perform a sweep on both N_{rx} and N_u and evaluate, for each case, the mechanism that provides the highest value of U^M . We tag each mechanism with a different color, and plot the selected mechanism for each $\{N_{rx}, N_u\}$ pair in Fig. 14. Note that the “row” for $N_u = 12$ corresponds to the case discussed above: indeed, the selected mechanisms in this row are the ones that provide the highest U^M according to Fig. 13 for the different N_{rx} values.

The figure shows that the selection of the best mechanism very much depends on the considered scenario. Except for the No-Ack/No-Retry scheme, which is never selected due to its poor multicast reliability, all the other mechanisms are selected for some of the $\{N_{rx}, N_u\}$ values. The main conclusions can be summarized as follows:

- The DMS scheme provides the best performance when there is only one multicast receiver regardless of the

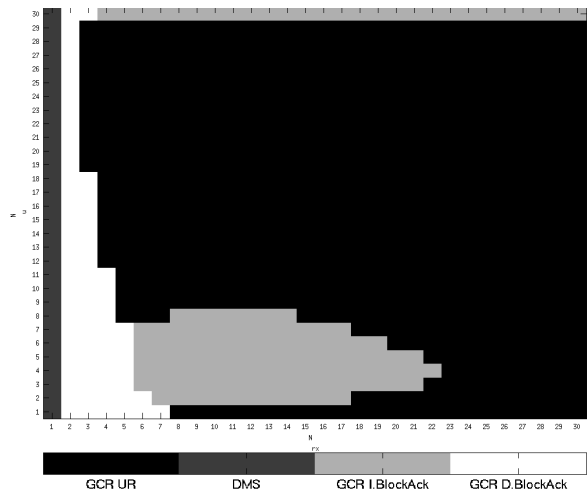


Fig. 14: Mechanism providing the best utility.

number of unicast stations, as in this case DMS achieves a good efficiency despite its simplicity and poor scalability properties.

- The GCR D.BlockAck scheme is best suited for scenarios in which the number of receivers is small ($2 \leq N_{rx} \leq 7$). This is due to the fact that this scheme is able to guarantee the required reliability through the use of acknowledgments and at the same time it is not too aggressive and thus leaves sufficient channel time for unicast traffic.
- The GCR I.BlockAck scheme provides the best performance in the following cases (i) when N_u is large ($N_u \geq 30$), as in this region the GCR UR fails to provide the required η_{min} ; and (ii) when N_u is small and N_{rx} is moderate, as in this case the large values of S_m compensate the relatively low values of S_u .
- Finally, the GCR UR scheme provides the best performance for scenarios with a large number of unicast stations ($N_u > 8$) or a large number of receivers ($N_{rx} > 22$). The reason for this performance is that, in these scenarios, with GCR I.BlockAck the performance of unicast traffic (i.e., the one not being prioritized) is worse than the performance of multicast traffic with GCR UR. With our definition of utility (the minimum of weighted throughputs), the latter results the winner of the selection algorithm.

Based on the above, we conclude that there is no “absolutely best” multicast mechanism to deliver video in 802.11 WLANs, as their performance highly varies with the network conditions. Indeed, for half of the considered cases the difference in terms of utility between the best two schemes is above 25%. As a consequence, a selection mechanism like the one proposed in this paper is required in order to select, for some given network conditions, the best performing scheme.

It is worthwhile noting that, even though the results presented in this section are very illustrative of the trade-offs involved by the different schemes and their qualitative behavior, the quantitative results obtained are specific to the considered set of network conditions, and would be different under different conditions. Therefore, Fig. 14 should not be

understood as a guide for choosing the best scheme; instead, the algorithm of Section IV should be run, for the specific current conditions, every time a scheme has to be selected.

VI. CONCLUSIONS

In this paper, we have performed the analysis and evaluation of the novel IEEE 802.11aa multicast mechanisms, in terms of throughput and reliability, for different WLAN and channel conditions, and compared them against the legacy multicast service of IEEE 802.11. While this analysis has been (mostly) limited to the mechanisms defined in the 802.11aa standard, many of our analytical techniques may also be used to analyze other mechanisms proposed in the literature for multicast transmission. Based on the results of our analysis, we have confirmed that the new mechanisms of 802.11aa are able to substantially improve performance, and that they provide different trade-offs considering their complexity, efficiency and reliability. We have identified the main limiting factors of each mechanism, and we have derived an algorithm which decides on the best multicast mechanism for a given WLAN scenario. According to these results, there seems to be no “absolutely the best” service for video delivery, as in addition to their relative differences in terms of complexity, their relative performance also varies with the traffic conditions. For the performance evaluation conducted in this paper, we have set the various parameters of each mechanism heuristically based on the standard recommendations and/or the results of other works. Following similar techniques to e.g. [6], the analysis of the 802.11aa mechanisms provided here could be leveraged to derive the optimal configuration of the different schemes.

ACKNOWLEDGMENTS

The authors are very grateful to the anonymous referees for their valuable comments which greatly helped in improving the paper. This work has been supported by the European Community through the CROWD project (FP7-ICT-318115). The work of A. Banchs and A. de la Oliva have been partially funded by the Spanish Government, MICINN, under research grant TIN2010-20136-C03.

REFERENCES

- [1] *IEEE Standard for Information Technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, IEEE Std. 802.11-2012 (Revision of IEEE Std 802.11-2007), 2012.
- [2] J. Kuri and S. Kasera, in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'99)*, vol. 2, 1999, pp. 760–767 vol.2.
- [3] *IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Higher-speed Physical Layer (PHY) extension in the 2.4 GHz band*, IEEE Amendment 802.11b, 2001.
- [4] *IEEE Standard for Information Technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 5: Enhancements for Higher Throughput*, IEEE Std. 802.11n, 2009.
- [5] *IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements*, IEEE Amendment 802.11e, 2005.
- [6] P. Serrano, A. Banchs, P. Patras, and A. Azcorra, “Optimal Configuration of 802.11e EDCA for Real-Time and Data Traffic,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 5, pp. 2511–2528, Jun. 2010.
- [7] *IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: MAC Enhancements for Robust Audio Video Streaming*, IEEE Std. 802.11aa-2012 (Amendment to IEEE Std 802.11-2012 as amended by IEEE Std 802.11ae-2012), 2012.
- [8] K. Tang and M. Gerla, “Random Access MAC for Efficient Broadcast Support in Ad Hoc Networks,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'00)*, 2000.
- [9] J. Kuri and S. K. Kasera, “Reliable Multicast in Multi-access Wireless LANs,” *Wireless Networks*, vol. 7, no. 4, pp. 359–369, Jul. 2001.
- [10] J. Miroll, Z. Li, and T. Herfet, “Wireless Feedback Cancellation for Leader-Based MAC Layer Multicast Protocols,” in *Proceedings of the IEEE 14th International Symposium on Consumer Electronics (ISCE'10)*, 2010.
- [11] H.-C. Chao, S. W. Chang, and J. L. Chen, “Throughput Improvements Using the Random Leader Technique for the Reliable Multicast Wireless LANs,” *Lecture Notes in Computer Science*, vol. 2093, pp. 708–719, 2001.
- [12] Y. Seok, D. Dujovne, T. Turletti, and P. Cuenca, “Leader based Multicast Service Proposal (IEEE 802.11-07/0120r6),” [Online] Available: <ftp://ftp-sop.inria.fr/rodeo/turletti/11-07-2127-00-000v-normative-text-leader-based-multicast.pdf>, 2007.
- [13] M. Sun, L. Huang, A. Arora, and T. Lai, “Reliable MAC Layer Multicast in IEEE 802.11 Wireless Networks,” in *Proceedings of the International Conference on Parallel Processing (ICPP'02)*, 2002.
- [14] K. Tang and M. Gerla, “MAC Reliable Broadcast in Ad Hoc Networks,” in *Proceedings of IEEE Military Communications Conference (MIL-COM'01)*, 2001.
- [15] Y. Bang, D. Lee, and J. K. Kevin Rhee, “Efficient Error Control Using Network Coding for Multicast Transmission (IEEE 802.11-09/0277r00),” [Online]. Available: <http://mentor.ieee.org/802.11/dcn/09/11-09-0277-00-00aa-efficient-error-control-using-network-coding-for-multicast-transmission.ppt>, Mar. 2009.
- [16] K. Maraslis, P. Chatzimisios, and A. Boucouvalas, “802.11aa: Improvements on Video Transmission over Wireless LANs,” in *Proceedings of the IEEE International Conference on Communications (ICC'12)*, Jun. 2012.
- [17] M. A. Santos, J. Villalon, and L. Orozco-Barbosa, “Evaluation of the IEEE 802.11aa Group Addressed Service for Robust Audio-Video Streaming,” in *Proceedings of the IEEE & CIC International Conference on Communications in China (ICCC'12)*, 2012.
- [18] S. A., A. Lyakhov, A. Urgenson, and O. Sokolova, “Wireless Groupcast Routing with Palette of Transmission Methods,” in *Proceedings of the International Workshop on Multiple Access Communications (MACOM'12)*, 2012.
- [19] A. Lyakhov and M. Yakimov, “Analytical Study of QoS-Oriented Multicast in Wireless Networks,” *EURASIP Journal on Wireless Communications and Networking*, 2011.
- [20] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, “Performance Anomaly of 802.11b,” in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'03)*, vol. 2, 2003.
- [21] *IEEE Standard for Information Technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: IEEE 802.11 Wireless Network Management*, IEEE Std. 802.11v, 2011.
- [22] D. Malone, P. Clifford, and D. Leith, “On Buffer Sizing for Voice in 802.11 WLANs,” *IEEE Communications Letters*, vol. 10, no. 10, pp. 701–703, 2006.
- [23] S. Paris, N. Focchi, F. Gringoli, and A. Capone, “An Innovative Rate Adaptation Algorithm for Multicast Transmissions in Wireless LANs,” in *Proceedings of IEEE Vehicular Technology Conference (VTC2013-Spring)*, Dresden, Germany, 2013.

- [24] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, Mar. 2000.
- [25] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma, "Performance of Reliable Transport Protocol over IEEE 802.11 Wireless LAN: Analysis and Enhancement," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'02)*, New York City, New York, June 2002.
- [26] A. Banchs, P. Serrano, and L. Vullero, "Providing Service Guarantees in 802.11e EDCA WLANs with Legacy Stations," *IEEE Transactions on Mobile Computing*, vol. 9, no. 8, pp. 1057–1071, Aug. 2010.
- [27] T. Stockhammer, M.-M. Hannuksela and T. Wiegand, "H.264/AVC in Wireless Environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 657–673, 2003.
- [28] T. Schierl, H. Schwarz, D. Marpe, and T. Wiegand, "Wireless Broadcasting Using the Scalable Extension of H.264/AVC," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'05)*, 2005.
- [29] A. Legout, J. Nonnenmacher, and E. W. Biersack, "Bandwidth Allocation Policies for Unicast and Multicast Flows," *IEEE/ACM Transactions on Networking*, vol. 9, no. 4, pp. 464–478, Aug. 2001.
- [30] P. Patras, A. Banchs, and P. Serrano, "A Control Theoretic Approach for Throughput Optimization in IEEE 802.11e EDCA WLANs," *Mobile Networks and Applications*, vol. 14, no. 6, pp. 697–708, Dec. 2009.
- [31] A. de la Oliva, P. Serrano, P. Salvador, and A. Banchs, "Performance evaluation of the IEEE 802.11aa multicast mechanisms for video streaming," in *Proceedings of IEEE WoWMoM*, 2013.
- [32] T. Li, Q. Ni, T. Turletti, and Y. Xiao, "Performance Analysis of the IEEE 802.11e Block ACK Scheme in a Noisy Channel," in *Proceedings of the International Conference on Broadband Networks (BroadNets'05)*, 2005.
- [33] P. Serrano, A. Banchs, T. Melia, and L. Vullero, "Performance anomalies of nonoptimally configured wireless lans," in *Proceedings of IEEE Wireless Communications & Networking Conference (WCNC)*, 2006.



Albert Banchs (M'04–SM'12) received his degree in telecommunications engineering from the Polytechnic University of Catalonia in 1997, and his Ph.D. degree from the same university in 2002. He received a national award for the best Ph.D. thesis on broadband networks. He was a visiting researcher at ICSI, Berkeley, in 1997, worked for Telefonica I+D, Spain, in 1998, and for NEC Europe Ltd., Germany, from 1998 to 2003. He has been with the University Carlos III of Madrid since 2003. Since 2009, he also has a double affiliation as Deputy Director of the

institute IMDEA Networks. Albert Banchs has authored over 80 publications in peer-reviewed journals and conferences and holds six patents. He is area editor for *Computer Communications* and has been senior and associate editor for *IEEE Communications Letters* and guest editor for *IEEE Wireless Communications*, *Computer Networks* and *Computer Communications*. He has served on the TPC of a number of conferences and workshops including IEEE INFOCOM, IEEE ICC and IEEE GLOBECOM, and has been TPC chair for European Wireless 2010, IEEE HotMESH 2010 and IEEE WoWMoM 2012. He is senior member of IEEE.



Antonio de la Oliva received a Telecommunication Engineering degree in 2004, and a PhD in Telematics in 2008 (receiving a national award ex-aequo for the best Ph.D. thesis in IPTV services), both from the University Carlos III of Madrid (UC3M), where he worked as a research and teaching assistant from 2005 to 2008 and, since then, has worked as an Visiting Professor. His research is focused to mobility in heterogeneous networks and wireless systems. He has published over 30 scientific papers in prestigious international journals and conferences, and he is also

an active contributor and voting member of the IEEE 802.21 where he has served as Vice-chair of IEEE 802.21b and Technical Editor of IEEE 802.21d. Currently he is involved in the FP7 CROWD project.



Lucas Eznarriaga is a QA Engineer at Midokura. Before joining Midokura, he was a Consultant Engineer at Altran Spain. Previously, Lucas was a Research Assistant at Institute IMDEA Networks investigating in the field of wireless networks and Internet protocols. He received his degree in Telecommunications Engineering and an MSc in Telematics Engineering from Universidad Carlos III de Madrid (UC3M), Spain during the years of 2010 and 2011, respectively. In the year of 2010, he took on an internship funded by the European Erasmus Placement Program at the Deutsche Telekom Laboratories in Berlin, Germany, where he worked on an investigation team of the SoftToken protocol prototype implementation.



Dariusz R. Kowalski received his Ph.D. degree in Computer Science in 2001 and MSc degree in Mathematics in 1996, both from the University of Warsaw, Poland. He is currently a Professor at the University of Liverpool, UK. He published over 115 peer reviewed research papers, mainly on distributed and parallel computing, network protocols and fault-tolerance. He worked in program committees of more than 30 conferences, including ACM PODC, ACM SPAA, ICALP, DISC, IEEE ICDCS, IEEE IPDPS.



Pablo Serrano got his Telecommunication Engineering degree and his PhD from the Universidad Carlos III de Madrid (UC3M) in 2002 and 2006, respectively. He has been with the Telematics Department of UC3M since 2002, where he currently holds the position of Associate Professor. In 2007 he was a Visiting Researcher at the Computer Network Research Group at Univ. of Massachusetts Amherst partially supported by the Spanish Ministry of Education under a Jos Castillejo grant, and in 2013 he was a Visiting Researcher at Telefonica Research

Center in Barcelona. He has over 50 scientific papers in peer-reviewed international journal and conferences. He serves on the Editorial Board of *IEEE Communications Letters*, and has served on the TPC of a number of conferences and workshops including IEEE Globecom and IEEE INFOCOM. His current work focuses on performance evaluation of wireless networks.