

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

Albert Banchs, *Senior Member, IEEE*, Antonio de la Oliva, *Member, IEEE*, Lucas Eznarriaga, Dariusz R. Kowalski, *Member, IEEE*, and Pablo Serrano, *Member, IEEE*

Abstract—Legacy IEEE 802.11 does not efficiently support multicast transmissions. To cope with the increasing demand for multicast, which is mainly required to deliver multimedia traffic, the IEEE 802.11aa Task Group has recently standardized new mechanisms for allowing efficient and robust transmission of multicast flows in wireless local area networks (WLAN). 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 for evaluating the performance of the mechanisms included in the 802.11aa standard and then 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.

Index Terms—IEEE 802.11 standards, multicast, multimedia communication, reliability, 802.11aa.

I. INTRODUCTION

THE IEEE 802.11 standard for wireless local area networks (WLANs) [1] is one of the most used technologies for providing 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: 1) the transmission rates first available imposed a severe bottleneck on the maximum achievable rate, regardless

Manuscript received May 1, 2013; revised September 2, 2013 and December 23, 2013; accepted January 28, 2014. Date of publication February 12, 2014; date of current version October 14, 2014. This work was supported in part by the European Community through the CROWD Project under Grant FP7-ICT-318115. The work of A. Banchs and A. de la Oliva was supported in part by the Spanish Ministry of Science and Innovation under Research Grant TIN2010-20136-C03. The review of this paper was coordinated by Dr. D. Zhao.

A. Banchs is with the University Carlos III of Madrid, Madrid 28911, Spain, and also with the Institute IMDEA Networks, Leganes 28918, Spain (e-mail: banchs@it.uc3m.es).

A. de la Oliva and P. Serrano are with University Carlos III of Madrid, Madrid 28911, Spain (e-mail: aoliva@it.uc3m.es; pablo@it.uc3m.es).

L. Eznarriaga was with the University Carlos III of Madrid, Madrid 28911, Spain. He is now with Midokura, Barcelona 08006, Spain (e-mail: lucas.eznarriaga@imdea.org).

D. R. Kowalski was with Institute IMDEA Networks, Leganes 28918, Spain. He is now with the University of Liverpool, Liverpool L69 3BX, U.K (e-mail: D.Kowalski@liverpool.ac.uk).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TVT.2014.2306014

of the efficiency of the medium-access-control (MAC) protocol; 2) only “best effort” service was supported, thus preventing any traffic differentiation to prioritize multimedia; and 3) 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 physical amendments has boosted the maximum achievable rates, starting with the 802.11b [3] that increased the maximum rate up to 11 Mb/s, continuing with the 802.11a and 802.11g amendments that reach up to 54 Mb/s, and finally, with the 802.11n [4], which introduces enhancements for higher throughput. On the other hand, the 802.11e amendment [5] has introduced traffic differentiation through the setting of contention parameters, enabling both the ability to prioritize one type of traffic over other types and 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 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 the performance of multimedia streaming over WLANs. In particular, the new mechanisms target significant improvement of the transport of multimedia streams by introducing: 1) a “*stream classification service*,” which aims at providing intraflow prioritization to allow for a graceful degradation of video quality; 2) *interworking with IEEE 802.11AVB*, for end-to-end reservations; 3) *overlapping basic service set (OBSS) management*, for coordination between multiple access points (APs); and 4) a “*group addressed transmission service*,” which provides an effective and efficient way for transmitting multicast traffic to a group of stations (STAs). 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 the following.

- 1) We present an analytical model for evaluating 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 tradeoffs between efficiency, reliability, and complexity, depending on the considered scenario.

- 2) We propose a novel algorithm for selecting the best mechanism for a given scenario. Our algorithm takes as input the number of unicast and multicast receivers in the WLANs, 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 for improving multicast performance. One of the first works on reliable multicast was 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 coexistence with legacy IEEE 802.11.

Few works have addressed the performance of 802.11aa, which is the focus of this paper. In [16], a qualitative description of the new mechanisms is provided, 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 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, in [19], the first analytical model of block acknowledgment (Block Ack) methods of 802.11aa is proposed. While some parts of the analysis of [19] are similar to ours, their work focuses specifically on the enhanced leader-based protocol, whereas ours is aimed at analyzing all the methods defined in 802.11aa instead of only one; furthermore, we address a more generic scenario, which includes both multicast and unicast STAs. 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 remainder of this 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 STAs 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 and 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 and the new mechanisms proposed in the 802.11aa amendment.

A. Multicast Service With Legacy 802.11

The IEEE 802.11 standard includes specific mechanisms for transmitting 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; 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 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 AP and includes the minimum set of rates that a STA must support to join the AP. Although it is not a requirement, usually the basic rate set includes only rates with lower order modulations; 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; therefore, 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 STAs.

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 802.11AVB, and stream classification service). Here, we focus on the mechanisms defined for transporting 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 earlier: 1) the poor reliability of the service, which is caused by the fact that multicast frames are not acknowledged; and 2) the high

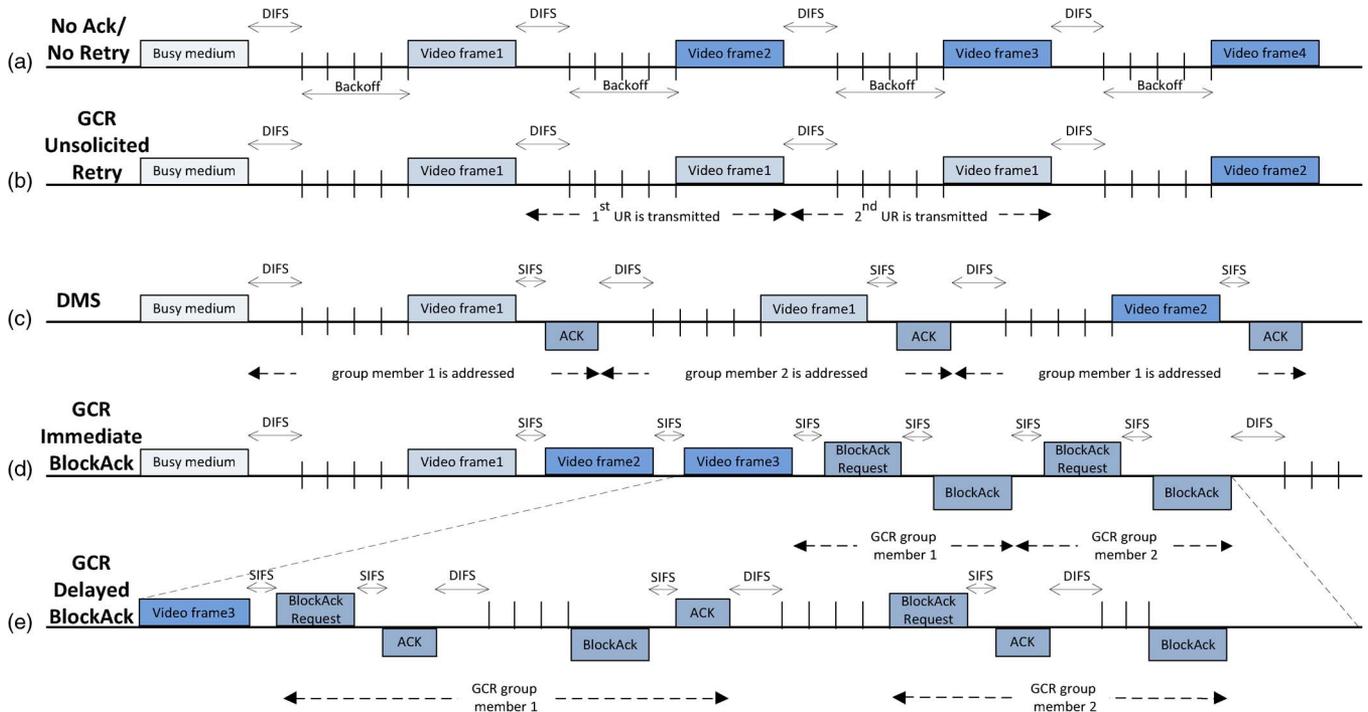


Fig. 1. Legacy IEEE 802.11 multicast and GATS mechanisms.

inefficiency resulting from the use of a low modulation coding scheme. 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 STAs by defining new *group addressed* retransmissions, a *group* being a set of STAs listening to the same (nonmulticast) address, which is 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 and a variation of one of the 802.11aa mechanisms. The operation of these mechanisms is shown in Fig. 1, which also includes the *legacy* multicast service described earlier (which we refer to “No-Ack/No-Retry”).

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. 1(b)]. 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 shown Fig. 1(c) 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 until the retransmission count limit is exceeded. (In which case, the frame is discarded.) Although 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 STA. 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, whereas each recipient of a *BlockAckRequest* replies immediately (after a short interframe space (SIFS) time). In the remainder of this 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 given mechanisms, which are 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. Similar to the previous scheme (which we hereafter will refer to as 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, cf., Fig. 1(e).

Despite this variety of mechanisms, the amendment provides no guidelines to choosing the most efficient for a given scenario. This is the main motivation of this paper, which we tackle in Sections III and IV. 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.

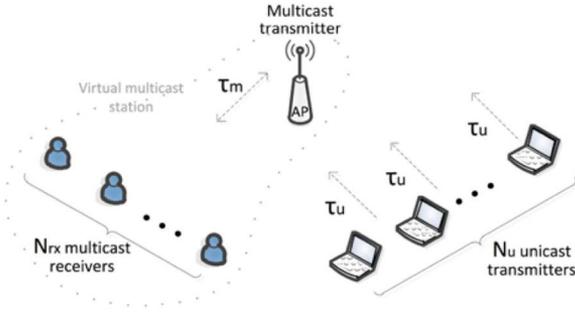


Fig. 2. Considered scenario for the performance analysis.

III. PERFORMANCE ANALYSIS

We consider the WLAN scenario shown in Fig. 2, with one AP sending multicast traffic to a group of N_{rx} STAs, using mechanism \mathcal{M} , where $\mathcal{M} \in \{\text{No-Ack/No-Retry}, \text{GCR UR}, \text{DMS}, \text{GCR I.BlockAck}, \text{GCR D.BlockAck}\}$. 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 all STAs are saturated (i.e., constantly backlogged). This metric gives the throughput guarantees for each traffic type and can be thus 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 interframe spacing (AIFS) such that $\text{AIFS} = \text{DIFS}$ (but can use different contention window (CW) parameters). We further assume that all STAs use the same transmission rate.² It is worthwhile noting that, following our previous work [6], the assumptions of saturation, fixed packet lengths, $\text{AIFS} = \text{DIFS}$, and equal transmission rates schemes could be easily relaxed, by combining the analysis of the multicast schemes of this paper with: 1) the computation in [6] of the slot time duration for variable packet lengths and transmission rates; 2) the algorithm proposed in [6] to compute the transmission probability of nonsaturated STAs; and 3) 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 tradeoffs of the different multicast schemes, which is 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 STAs). From this, the frame error rate (FER) for multicast

receiver e_i and for a unicast STA can be computed, respectively, as follows:

$$\begin{aligned} p_{fe_i} &= 1 - (1 - p_{e_i})^{L_m} \\ p_{f_u} &= 1 - (1 - p_{e_u})^{L_u}. \end{aligned} \quad (1)$$

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 STAs, or between one multicast STA and one or more unicast STAs. As collisions cannot occur between multicast STAs, we assume that all multicast activity can be modeled with a single “virtual” STA, which captures the aggregate behavior of the multicast sender and the receivers. This “virtual” STA 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 STA 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 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 tradeoff between these two metrics. For instance, with the GCR unsolicited retry (GCR UR) scheme, the larger the number of retransmissions, the higher reliability; however, this decreases throughput as more time is devoted to retransmitting 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 STA 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 STAs of the multicast flow, i.e.,

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

¹As shown in [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 STA 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, such as, for example, the recent paper of [23]. This issue, however, is outside 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.

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

In the following, we present the analysis of the given 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 Request-to-Send/Clear-to-Send (RTS/CTS) access mode.³

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

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

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, i.e.,

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

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

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

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

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

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

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

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 contains a collision involving a multicast transmission, respectively; T_e , T_{c_u} , and T_{c_m} are the slot time duration in each of these cases, respectively; and T_{s_u} (T_{s_m}) is the length of a successful unicast (multicast) transmission. For simplicity,

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

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 STAs, which can be computed as

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

An empty slot occurs when there are no unicast nor multicast transmissions, i.e.,

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

whereas the probability of a collision involving multicast traffic can be derived from the previous probabilities as follows:

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

The time slot duration in (11) can be computed as

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

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

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

where T_{PLCP} is the physical-layer convergence protocol (PLCP) preamble, H is the MAC header plus the frame check sequence 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 . 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 the distributed coordination function (DCF) with no backoff. Under these conditions, τ_m can be computed as

$$\tau_m = \frac{2}{\text{CW}_{\min_m} + 1} \quad (18)$$

where CW_{\min_m} is the minimum CW of multicast traffic.

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

$$\tau_u = \frac{2(1 - 2p_{e_u})(1 - p_{e_u}^{R+1})}{A + B + C} \quad (19)$$

$$A = W_u (1 - (2p_{e_u})^{m_u + 1})(1 - p_{e_u})$$

$$B = (1 - 2p_{e_u})(1 - p_{e_u}^{R+1})$$

$$C = W_u 2^{m_u} p_{e_u}^{m_u + 1} (1 - p_{e_u}^{R - m_u})$$

where R is the retry limit, $W_u = \text{CW}_{\min_u}$ is the minimum CW, 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

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

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 as follows:

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

which takes into account that each frame is transmitted $R + 1$ times and that its content is successfully received with a certain probability. Combining (22) with (4) and (21) yields

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

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

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

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

Finally, given that unicast STAs 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 earlier 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: 1) to set $p_{c_m} = 0$ in (21), as video frames are now lost only due to channel conditions; 2) to multiply (23) by $(1 - p_{c_m})$, given that only those channel access that do not collide constitute an actual frame transmission; 3) to update the duration of a successful multicast transmission given by (17) to include the exchange of the RTS/CTS frames; and 4) 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 UR 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.

⁴In contrast to the No-Ack/No-Retry scheme, with GCR UR, a frame is transmitted more than once. As a consequence, using the transmission rate of No-Ack/No-Retry for GCR UR would be overly conservative. Indeed, since with the GCR UR 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.

C. Directed Multicast

We next analyze the case when the AP transmits a multicast flow to N_{rx} STAs 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 that obtained for the GCR UR mechanism, i.e.,

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

The multicast throughput with DMS can be computed as

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

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

$$\tau_{m,i} = \frac{N_{\text{tx}_i}}{\sum_{j=1}^{N_{\text{rx}}} B_j + N_{\text{tx}_j}} \quad (27)$$

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

$$N_{\text{tx}_i} = \sum_{j=1}^R j (1 - p_{c_m})(1 - p_{fe_i}) (1 - (1 - p_{c_m})(1 - p_{fe_i}))^{j-1} + (R+1) (1 - (1 - p_{c_m})(1 - p_{fe_i}))^R \quad (28)$$

which can be rearranged as

$$N_{\text{tx}_i} = \frac{1 - (1 - (1 - p_{c_m})(1 - p_{fe_i}))^{R+1}}{(1 - p_{c_m})(1 - p_{fe_i})} \quad (29)$$

and the former by

$$B_i = \sum_{j=1}^R \sum_{k=1}^j \frac{2^{\min(k, m_m)} \text{CW}_{\min_m} - 1}{2} (1 - p_{c_m})(1 - p_{fe_i}) \cdot (1 - (1 - p_{c_m})(1 - p_{fe_i}))^{j-1} + \sum_{k=1}^{R+1} \frac{2^{\min(k, m_m)} \text{CW}_{\min_m} - 1}{2} \times (1 - (1 - p_{c_m})(1 - p_{fe_i})) \quad (30)$$

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), i.e.,

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

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

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

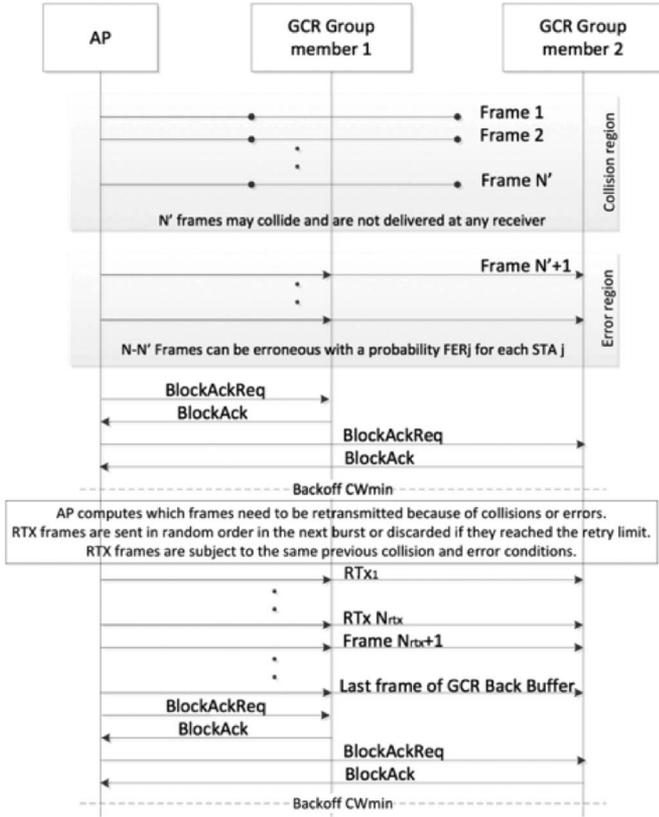


Fig. 3. GCR Immediate Block Ack example scenario.

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

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

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 STAs 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 show 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 the following.

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

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

- The $N - N'$ frames that *survive* the collision can still be lost due to channel conditions, with an independent per-receiver error probability p_{fe_i} .
- 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, whereas the rest are new frames.

As it follows from the given example, in general, out of the N frames in a given burst, some will be retransmissions of frames that have not reached successfully all STAs in previous attempts, either due to collisions or channel errors, whereas 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; thus, the probability that such a frame reaches the retry limit would be higher.⁶

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

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

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

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

According to the standard, the number of retransmissions with the Block Ack is not subject to a retry limit but to a MAC service data unit (MSDU) lifetime. For analytical tractability, we model this with 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 $(1/\tau_m) \cdot T_{\text{slot}, m}$ plus the time required to transmit one burst, where $T_{\text{slot}, m}$ 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 this and given that a frame is retransmitted until it has been successfully received by all STAs or it reaches R , the

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.

reliability of the GCR Immediate Block Ack mechanism can be computed as

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

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 $(N/N_{\text{tx}}) \cdot \eta_m^{\text{I.BlockAck}}$, which leads to the following expression for the multicast throughput (already derived in [19]):

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

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

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

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

$$\begin{aligned} T_{s_m} = T_{c_m} = N & \left(T_{\text{PLCP}} + \frac{H}{R_m} + \frac{L_m}{R_m} + \text{SIFS} \right) \\ & + 2N_{\text{rx}} \left(T_{\text{PLCP}} + \frac{\text{BACK}}{R_c} \right) + (2N_{\text{rx}} - 1)\text{SIFS} + \text{DIFS} \end{aligned} \quad (39)$$

where BACK is the length of a BlockAckRequest or a Block Ack.

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

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

where $P_{\text{tx}}(j)$ 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.

- $P_{\text{tx,col}}(j, k)$: This is the probability that out of the j transmission attempts, k result in collisions.
- $P_{\text{cf_attempts}}(j - k)$: 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).

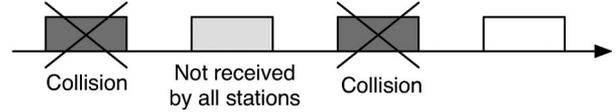


Fig. 4. Frame that is transmitted a total number of $j = 4$ times because it suffers $k = 2$ collisions (with probability $P_{\text{tx,col}}(4, 2)$). $j - k = 2$ collision-free transmissions were required to be successfully received by all STAs (with probability $P_{\text{cf_attempts}}(2)$).

We show an example illustrating these two probabilities in Fig. 4. Based on the given examples, $P_{\text{tx}}(j)$ can be computed as

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

To compute $P_{\text{tx,col}}(j, k)$, we note that the number of collided attempts can be modeled after a binomial distribution, in which the probability of a collision is $p_{c_m} N'/N$ (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

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

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

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

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

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

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.

Similar to GCR UR, in the case that RTS/CTS is used, the analysis has to be adapted by: 1) setting $N' = 0$ in (35) as a burst of frames cannot suffer a collision; 2) multiplying (37) by $(1 - p_{c_m})$; 3) updating the duration of a successful multicast transmission T_{s_m} in (39) to include the exchange of

the RTS/CTS frames; 4) setting the length of a collision to that of a unicast transmission (15) instead of using T_{c_m} as given by (39); and 5) computing τ_m by taking into account the collision probability for multicast frames p_{c_m} 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 STA 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 an STA), we can model (as previously mentioned) all multicast-related activity with a single *virtual STA*. This STA is constantly backlogged, i.e., it is always executing a backoff process, and transmits sequentially multicast bursts, BlockAckRequests and BlockAcks.

Thus, the virtual STA 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 STA collide with probability p_{c_m} , on average control frames are transmitted $1/(1 - p_{c_m})$ times. Based on this, the probability that a randomly chosen transmission attempt from the virtual multicast STA is a multicast burst (p_b) can be computed as

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

The expression for the reliability is the same as in the previous case, i.e., (36), whereas 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, i.e.,

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

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, as shown in the following:

$$\begin{aligned} T_{slot}^{D.BlockAck} &= P_{s_u} T_{s_u} + P_{c_u} T_{c_u} + \tau_m p_b T_{N_{frames}} \\ &\quad + \tau_m (1 - p_b) (1 - p_{c_m}) T_{BlockAcks} \\ &\quad + P_e T_e + \tau_m (1 - p_b) p_{c_m} T_{c_u} \end{aligned} \quad (48)$$

⁷As the control BlockAck frames are sent at the basic rate set, the transmission error probability is very low and neglected in our analysis.

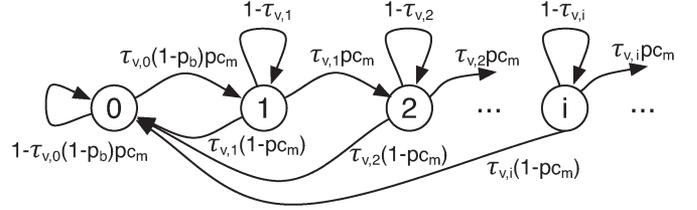


Fig. 5. Markov chain model of the virtual multicast STA.

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

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

$$T_{BlockAcks} = T_{PLCP} + \frac{BACK}{R_c} + SIFS + T_{PLCP} + \frac{ACK}{R_c} + DIFS. \quad (50)$$

We next address the computation of the transmission probabilities τ_u and τ_m . For the case of unicast STAs, the transmission probability τ_u is the same as the one given by (19). To compute the transmission probability of the virtual STA, we need to take into account that, although a backoff process is executed both for multicast bursts and control frames, the CW is doubled *only* after a control frame collision.

We model this behavior using the Markov chain shown in Fig. 5, in which each state represents the backoff stage of the virtual multicast STA. Thus, at state 0, the CW value of the virtual STA is CW_{min_m} ; 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 STA transmits (which happens with probability $\tau_{v,0}$), and the transmission does not correspond to a multicast burst ($1 - p_b$) and collides (p_{c_m}), we move to state 1; otherwise, we stay in state 0. For all other states, there are three possible outcomes per backoff counter decrement: 1) the STA does not transmit (which happens with probability $1 - \tau_{m,i}$) and remains in that state; 2) it performs a successful transmission ($\tau_{m,i}(1 - p_{c_m})$), and we move to stage 0; or 3) there is a transmission that collides ($\tau_{m,i}p_{c_m}$), 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

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

and p_i for $i > 0$ as

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

From this, we have

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

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

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

from which

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

From (53) and (56), we have

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

By substituting $\tau_{v,0}$ and p_0 , we get

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

Finally, numerical techniques can be used to solve the non-linear system given by (6), (19), (20), and (60), terminating 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 Section III allows for the computation of the performance of each of the multicast mechanisms as a function of the number of multicast and unicast STAs 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 be also 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., $\eta_m^M > \eta_{\min}$.
- As long as the minimum reliability requirement is satisfied, we would like to maximize the throughput per-STA guarantees provided to multicast and unicast, i.e., S_m^M and S_u^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* criterion gives the same weight to unicast and multicast flows, and has the drawback that multicast is not given a higher weight even if it benefits a larger number of users and hence contributes more to the overall utility of the system.
- 2) The *linear-receiver-dependent* 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 toward 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 tradeoff between prioritizing multicast and avoiding the starvation of unicast.

Following these criteria, 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

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

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

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

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

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

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

which terminates the selection of the mechanism.

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)	-		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 $N \in \{8, \dots, 12\}$.

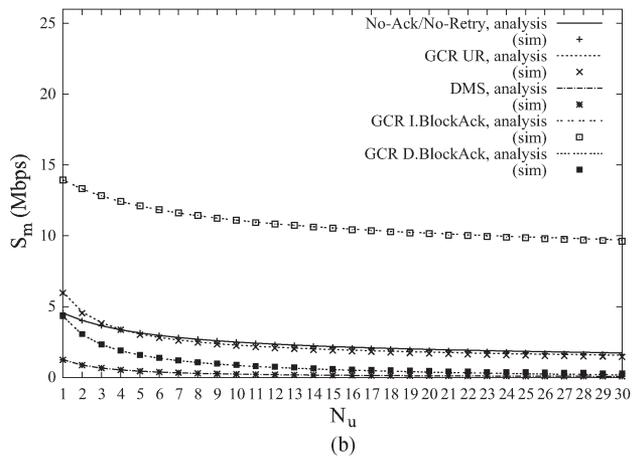
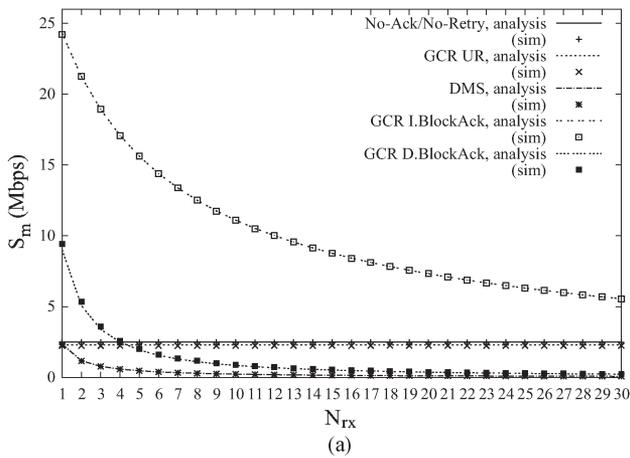


Fig. 6. Multicast throughput. (a) Throughput versus number of receivers, $N_{rx} = 10$. (b) Throughput versus number of unicast STAs, $N_{rx} = 10$.

V. PERFORMANCE EVALUATION

Here, 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 shown in Fig. 2, in which the AP transmits a multicast flow to N_{rx} receivers, and N_u STAs transmit unicast data to the AP. The length of multicast and unicast frames is set to 1500 B. Each simulation value corresponds to the average of ten 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 and 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, i.e., S_m . To understand the impact on performance of the number

of receivers and the number of unicast STAs, 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 Fig. 6(a) and (b), 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. 6(a) show the same qualitative behavior. Indeed, their behavior does not depend on any feedback from the receivers; 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 using 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 that the AP has to transmit increases with N_{rx} ; therefore, although the aggregated throughput in saturation may be large, the per-STA 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

⁸<http://www.omnetpp.org>

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

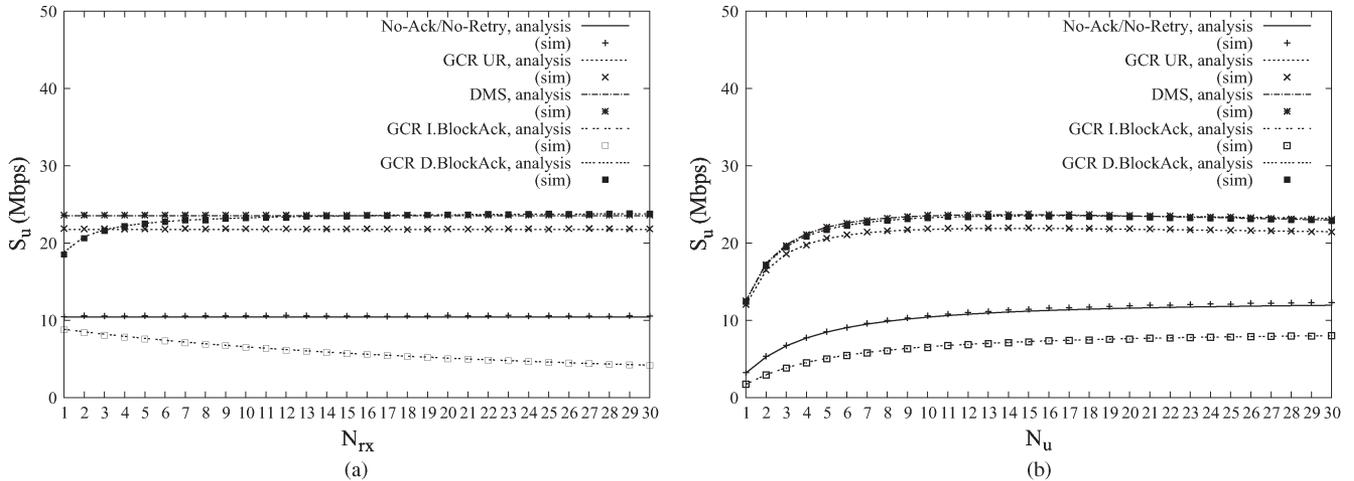


Fig. 7. Unicast throughput. (a) Throughput versus number of receivers, $N_u = 10$. (b) Throughput versus number of unicast STAs, $N_{rx} = 10$.

potentially suitable for scenarios with a very low number of STAs (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, whereas with the Delayed version, S_m is reduced by approximately two thirds. The reasons for the good performance of the Immediate scheme are twofold. 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 STAs. In contrast, with the Delayed version, not only more channel time is spent on control messages due to the acknowledgments but also STAs have to defer channel access and compete with unicast STAs to transmit these frames, which may result in collisions and retransmissions.

B. Unicast Throughput

Next, we analyze the throughput performance of unicast traffic, i.e., S_u . To this end, we proceed as in Section IV. 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 shown in Fig. 7(a) and (b), 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. 7(a) confirms that the performance does not depend on the number of receivers for both schemes, but the throughput obtained by unicast STAs 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; therefore, due to the *performance anomaly*, the throughput for all STAs in the WLAN is severely degraded. Similar results are obtained in Fig. 7(b); in that figure, we observe that the difference between the two mechanisms decreases with the number of unicast STAs as the impact of the performance anomaly is reduced.

For the DMS mechanism, we observe that, when the number of unicast STAs is fixed to $N_u = 10$, their throughput S_u is not affected by the number of video receivers N_{rx} [see Fig. 7(a)]. This is because, with DMS, the AP behaves like a single saturated STA, regardless of N_{rx} . On the other hand, when N_{rx} is fixed to 10 and N_u varies [see Fig. 7(b)], the throughput S_u first increases (with the activity on the WLAN) and then decreases (as the number of STAs 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 (see 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 occurs 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. Similar to that conducted earlier, we fix N_{rx} (N_u) and perform a sweep on N_u (N_{rx}). The corresponding analytical and simulation results are provided in Fig. 8(a) and (b).

We first analyze the impact of the number of receivers on the reliability. According to the results of Fig. 8(a), 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 we have shown in Section IV. Additionally, for the “feedback-based”

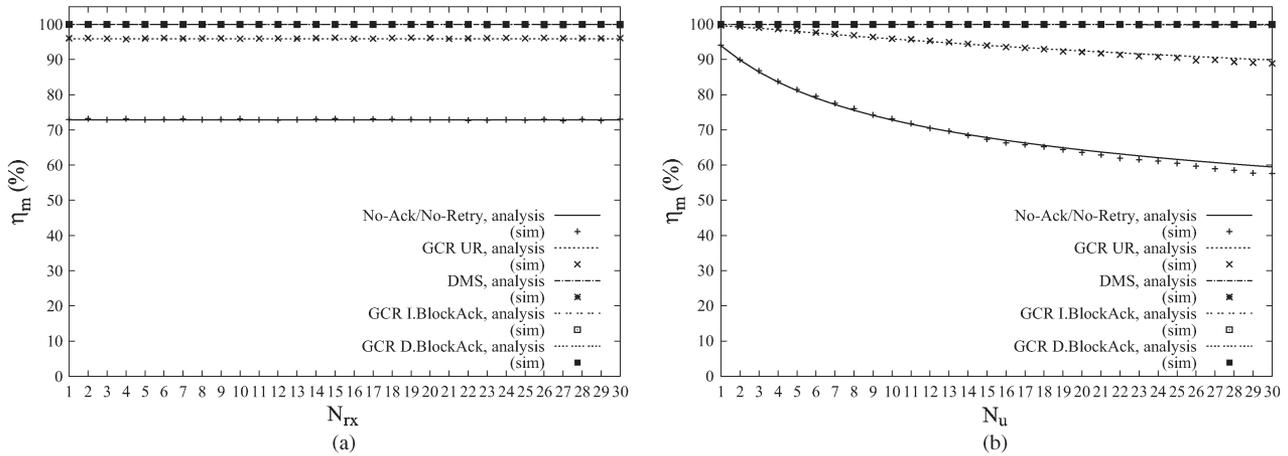


Fig. 8. Multicast reliability. (a) Reliability versus number of receivers, $N_u = 10$. (b) Reliability versus number of unicast STAs, $N_{rx} = 10$.

schemes such as DMS and GCR BlockAck, performance does depend on N_{rx} . 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 STAs on η_m , as shown in Fig. 8(b). 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 URs are able to limit the performance degradation to some extent. For instance, with $N_u = 30$ STAs, reliability is approximately 55% with No-Ack/No-Retry, whereas 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 (particularly when the number of receivers or unicast STAs is large), their reliability in terms of performance is poor. We also observe that, although all feedback-based schemes provide the same reliability, their performance in terms of multicast throughput is very different [see Fig. 6(a)]. 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 affected by the 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: 1) the trends observed earlier for the different schemes hold independent of the FER value; 2) even for a FER as high as 20%, feedback-based schemes keep a 100% reliability; and 3) although 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.

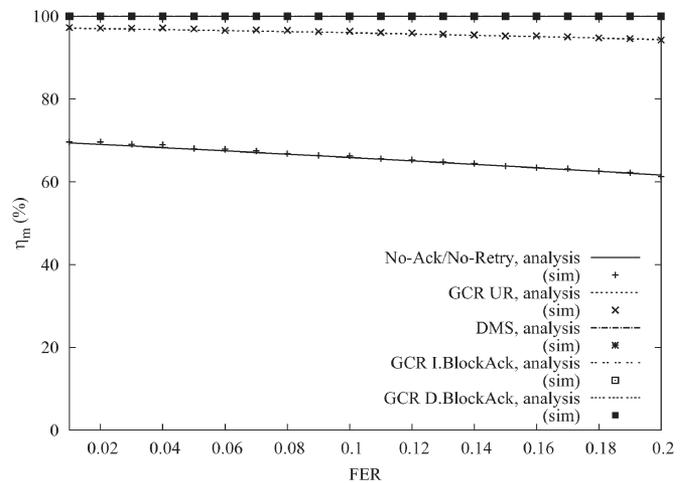


Fig. 9. Impact of the FER on the reliability of multicast.

E. Impact of TXOPlimit

The configuration of the TXOPlimit parameter is critical for the performance of the Block Ack schemes. Indeed, the higher the setting of this parameter, the larger priority is given to multicast traffic over unicast. To assess the impact of this parameter, Fig. 10 shows the throughput of a multicast and a unicast STA for the two Block Ack mechanisms (Immediate and Delayed) with $N_{rx} = N_u = 10$. We observe from the figure that: 1) this parameter serves to set the level of priority for unicast and multicast, in particular for the Immediate scheme; and 2) the sum of the unicast and multicast throughputs increases slightly with this parameter, e.g., from 13.5 Mb/s with a burst size of 2 to 17.7 Mb/s 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 worthwhile.

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). 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

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

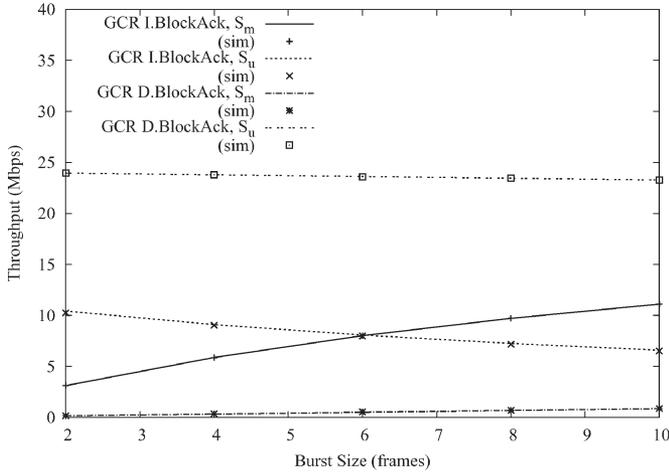


Fig. 10. Impact of the burst size on unicast and multicast throughput with GCR Block Ack schemes.

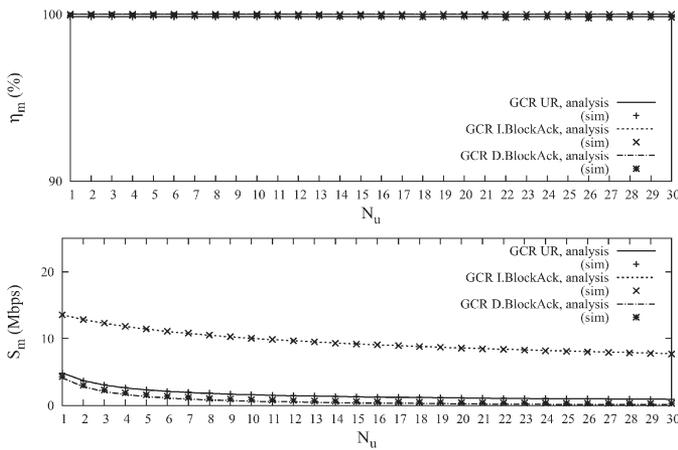


Fig. 11. Performance of GCR schemes with RTS/CTS.

figure that, as compared with the basic access mechanism, reliability is maximized for a 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 slight decrease in throughput, due to the longer frame exchange required for successful transmissions. Other than that, the trends observed for the different mechanisms given earlier do not change significantly. Results for a varying N_{rx} (not reported here for space reasons) lead to a similar conclusion.

G. Nonsaturation 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 STA can use in the worst case (when all other STAs are saturated). Hence, this metric serves to optimize the throughput guarantees provided to multicast and unicast STAs.

To show this, we set up a scenario in which the AP generates multicast traffic at rate R_m , and unicast STAs generate 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 STAs are

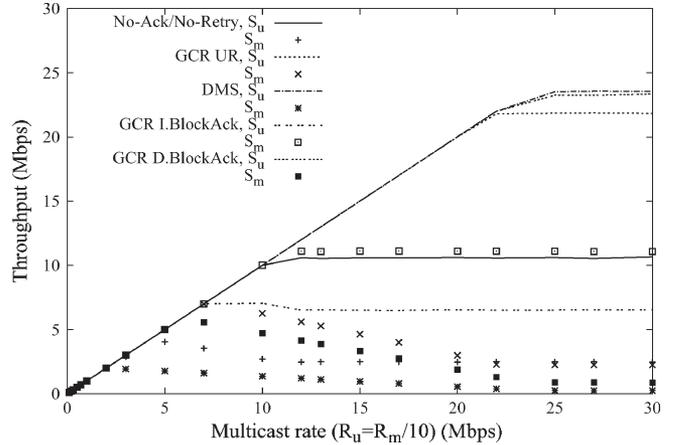


Fig. 12. Throughput with nonsaturation conditions.

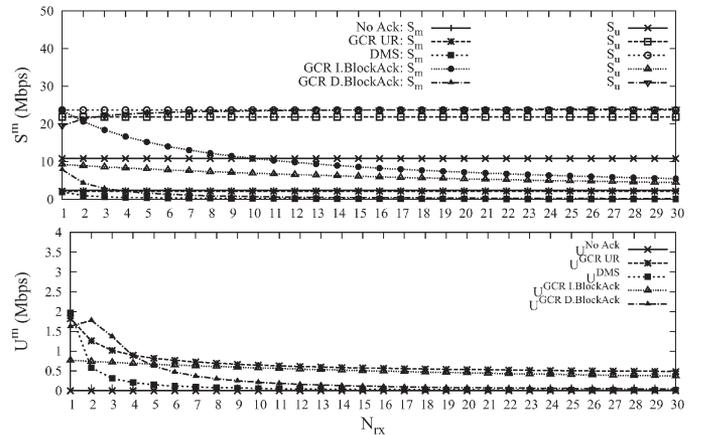


Fig. 13. (top) Multicast and unicast throughput and (bottom) overall utility versus N_{rx} for $N_u = 12$.

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 Mb/s for the No-Ack/No-Retry and 22 Mb/s for the case of GCR UR). Additionally, in some cases, STAs may receive a throughput larger than the saturation throughput, but this only happens when other STAs are not using all their saturation throughput [33]. Thus, this figure shows the usefulness of the saturation throughput as a performance metric for the purpose of this paper.

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 STAs¹¹ 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 are shown 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 .)

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

The results show the same qualitative performance that we have observed earlier. For instance, S_m is maximized using GCR I.BlockAck, whereas 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, to select the best mechanism, we need to use a criterion that provides a good balance between the multicast and unicast throughputs S_m and S_u , respectively, 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^M as defined in (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.

- The No-Ack/No-Retry mechanism does not guarantee the minimum threshold for the reliability; therefore, its utility is zero for all considered cases.
- The LogRD criterion is able to provide a tradeoff between S_u and S_m as, for the I.BlockAck mechanism, for example, performance is determined by S_u , whereas 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 these 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, whereas unicast throughput increases. Thus, for a small number of receivers, utility is dominated by unicast traffic whose throughput increases, whereas for a larger number of receivers, it is dominated by multicast traffic whose throughput decreases.

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

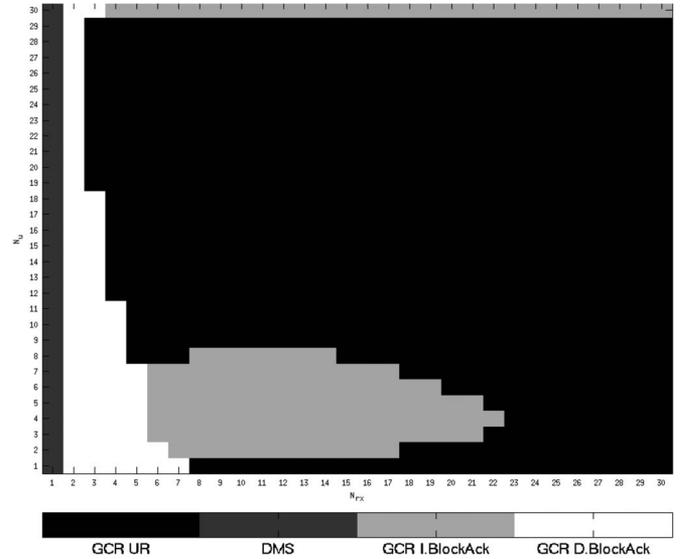


Fig. 14. Mechanism providing the best utility.

- 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: 1) when N_u is large ($N_u \geq 30$), as in this region, the GCR UR fails to provide the required η_{\min} ; and 2) 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 STAs ($N_u > 8$) 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 as the winner of the selection algorithm.

Thus, 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 such as the one proposed in this paper is required to select, for some given network conditions, the best performing scheme.

It is worthwhile noting that, although the results presented here are very illustrative of the tradeoffs 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. CONCLUSION

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 tradeoffs considering their complexity, efficiency, and reliability. We have identified the main limiting factors of each mechanism and derived an algorithm that 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 studies. Following similar techniques to, for example, [6], the analysis of the 802.11aa mechanisms provided here could be leveraged to derive the optimal configuration of the different schemes.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their valuable comments, which greatly helped improve the quality of this paper.

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. Kaser, “Reliable multicast in multi-access wireless LANs,” in *Proc. IEEE INFOCOM*, 1999, vol. 2, pp. 760–767.
- [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 Trans. Veh. Technol.*, 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 *Proc. IEEE WCNC*, 2000, pp. 454–459.
- [9] J. Kuri and S. K. Kaser, “Reliable multicast in multi-access wireless LANs,” *Wireless Netw.*, 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 *Proc. IEEE 14th ISCE*, 2010, pp. 1–6.
- [11] H.-C. Chao, S. W. Chang, and J. L. Chen, “Throughput improvements using the random leader technique for the reliable multicast wireless LANs,” in *Proc. ICN*, vol. 2093, *Lecture Notes in Computer Science*, 2001, pp. 708–719.
- [12] Y. Seok, D. Dujovne, T. Turletti, and P. Cuenca, Leader based Multicast Service Proposal (IEEE 802.11-07/0120r6) 2007. [Online]. Available: <ftp://ftp-sop.inria.fr/rodeo/turletti/11-07-2127-00-000v-normative-text-leader-based-multicast.pdf>
- [13] M. Sun, L. Huang, A. Arora, and T. Lai, “Reliable MAC layer multicast in IEEE 802.11 wireless networks,” in *Proc. ICPP*, 2002, pp. 527–536.
- [14] K. Tang and M. Gerla, “MAC reliable broadcast in ad hoc networks,” in *Proc. IEEE MILCOM*, 2001, pp. 1008–1013.
- [15] Y. Bang, D. Lee, J. K. Kevin Rhee, Efficient Error Control Using Network Coding for Multicast Transmission (IEEE 802.11-09/0277r00), Mar. 2009. [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>
- [16] K. Maraslis, P. Chatzimisios, and A. Boucouvalas, “802.11aa: Improvements on video transmission over wireless LANs,” in *Proc. IEEE ICC*, Jun. 2012, pp. 115–119.
- [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 *Proc. IEEE CIC ICC*, 2012, pp. 6879–6884.
- [18] S. A. A. Lyakhov, A. Urgenson, and O. Sokolova, “Wireless groupcast routing with palette of transmission methods,” in *Proc. Int. Workshop MACOM*, 2012, pp. 97–108.
- [19] A. Lyakhov and M. Yakimov, “Analytical study of QoS-oriented multicast in wireless networks,” *EURASIP J. Wireless Commun. Netw.*, vol. 2011, pp. 307507-1–307507-13, 2011.
- [20] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, “Performance anomaly of 802.11b,” in *Proc. IEEE INFOCOM*, 2003, vol. 2, pp. 836–843.
- [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 Commun. Lett.*, vol. 10, no. 10, pp. 701–703, Oct. 2006.
- [23] S. Paris, N. Facchi, F. Gringoli, and A. Capone, “An innovative rate adaptation algorithm for multicast transmissions in wireless LANs,” in *Proc. IEEE VTC-Spring*, Dresden, Germany, 2013, pp. 1–5.
- [24] G. Bianchi, “Performance analysis of the IEEE 802.11 distributed coordination function,” *IEEE J. Sel. Areas Commun.*, 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 *Proc. IEEE INFOCOM*, New York, NY, USA, Jun. 2002, pp. 599–607.
- [26] A. Banchs, P. Serrano, and L. Vollero, “Providing service guarantees in 802.11e EDCA WLANs with legacy stations,” *IEEE Trans. Mobile Comput.*, 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 Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 657–673, Jul. 2003.
- [28] T. Schierl, H. Schwarz, D. Marpe, and T. Wiegand, “Wireless broadcasting using the scalable extension of H.264/AVC,” in *Proc. IEEE ICME*, 2005, pp. 884–887.
- [29] A. Legout, J. Nonnenmacher, and E. W. Biersack, “Bandwidth allocation policies for unicast and multicast flows,” *IEEE/ACM Trans. Netw.*, 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 Netw. Appl.*, 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 *Proc. IEEE WoWMoM*, 2013, pp. 1–9.

- [32] T. Li, Q. Ni, T. Turetli, and Y. Xiao, "Performance analysis of the IEEE 802.11e block ACK scheme in a noisy channel," in *Proc. Int. Conf. BroadNets*, 2005, pp. 511–517.
- [33] P. Serrano, A. Banchs, T. Melia, and L. Vollero, "Performance anomalies of nonoptimally configured wireless lans," in *Proc. IEEE WCNC*, 2006, pp. 920–925.



Albert Banchs (M'04–SM'12) received the Telecommunications Engineering and Ph.D. degrees from the Polytechnic University of Catalonia, Barcelona, Spain, in 1997 and 2002.

In 1997, he was a Visiting Researcher with the International Computer Science Institute, Berkeley, CA, USA. In 1998, he was with Telefonica I+D. From 1998 to 2003, he was with NEC Europe Ltd., Germany. Since 2003, he has been with the University Carlos III of Madrid, Madrid, Spain. Since 2009, he has also been a Deputy Director with Institute

IMDEA Networks, Leganes, Spain. He is the author of over 80 papers in peer-reviewed journals and conferences and is the holder of six patents.

Dr. Banchs has served on the technical program committees (TPCs) of a number of conferences and workshops, including the IEEE Conference on Computer Communications; the IEEE International Conference on Communications; and the IEEE Conference on Global Communications. He has served as the TPC Chair for the 2010 European Wireless Conference; the 2010 IEEE Workshop on Hot Topics in Mesh Networking; and the 2012 IEEE International Symposium on a World of Wireless, Mobile, and Multimedia Networks. He has served as a Senior Editor and as an Associate Editor for IEEE COMMUNICATIONS LETTERS and as a Guest Editor for IEEE WIRELESS COMMUNICATIONS, *Computer Networks*, and *Computer Communications*. He currently serves as an Area Editor for *Computer Communications*. He received a national award for the best Ph.D. thesis on broadband networks.



Antonio de la Oliva (M'09) received the Telecommunication Engineering and Ph.D. degrees in telematics from the University Carlos III of Madrid, Madrid, Spain.

From 2005 to 2008, he was a Research and Teaching Assistant with the University Carlos III of Madrid. Since then, he has been a Visiting Professor with the same university. He is the author of over 30 scientific papers in prestigious international journals and conferences. He is currently involved in the Seventh Framework Programme CROWD Project.

His research interests include mobility in heterogeneous networks and wireless systems.

Dr. de la Oliva is an active contributor and a voting member of the IEEE 802.21, where he has served as a Vice Chair of IEEE 802.21b and as a Technical Editor of IEEE 802.21d. He received a national award *ex aequo* for the best Ph.D. thesis in IPTV services.



Lucas Eznarriaga received the Telecommunications Engineering and M.Sc. degrees in telematics engineering from the University Carlos III of Madrid, Madrid, Spain, in 2010 and 2011, respectively.

In 2010, he took an internship funded by the European Erasmus Placement Program with Deutsche Telekom Laboratories, Berlin, Germany, where he worked on an investigation team of the SoftToken protocol prototype implementation. He was a Research Assistant with Institute IMDEA Networks, Leganes, Spain, investigating wireless networks and Internet protocols. He was then a Consultant Engineer with Altran Spain. He is currently a Quality Assurance Engineer with Midokura, Barcelona, Spain.



Dariusz R. Kowalski (M'11) received the M.Sc. degree in mathematics and the Ph.D. degree in computer Science from the University of Warsaw, Warszawa, Poland, in 1996 and 2001, respectively.

He is currently a Professor with the University of Liverpool, Liverpool, U.K. He is the author of over 115 peer-reviewed research papers, mainly on distributed and parallel computing, network protocols, and fault tolerance.

Dr. Kowalski has served on several program committees of more than 30 conferences, including the Association for Computing Machinery (ACM) Symposium on Principles of Distributed Computing; the ACM Symposium on Parallelism in Algorithms and Architectures; the International Colloquium on Automata, Languages, and Programming; the International Symposium on Distributed Computing; the IEEE International Conference on Distributed Computing Systems Workshops; and the IEEE International Parallel and Distributed Processing Symposium.



Pablo Serrano (M'09) received the Telecommunication Engineering and Ph.D. degrees from the Universidad Carlos III de Madrid (UC3M), Madrid, Spain, in 2002 and 2006, respectively.

Since 2002, he has been with the Department of Telematics, UC3M, where he is currently an Associate Professor. In 2007, he was a Visiting Researcher with the Computer Network Research Group, University of Massachusetts, Amherst, MA, USA, where his research was partially supported by the Spanish Ministry of Education under a José

Castillejo grant. In 2013, he was a Visiting Researcher with Telefonica Research Center, Barcelona, Spain. He is the author of 50 scientific papers in peer-reviewed international journal and conferences. His research interests include performance evaluation of wireless networks.

Dr. Serrano serves on the Editorial Board of the IEEE COMMUNICATIONS LETTERS and the technical program committees of a number of conferences and workshops, including the IEEE Conference on Global Communications and the IEEE Conference on Computer Communications.