

## Article

# PROFEE: A Probabilistic-Feedback Based Speed Rate Adaption for IEEE 802.11bc

Javier Gomez <sup>1,†</sup> , Jose Jaime Camacho-Escoto <sup>2,\*,†</sup> , Luis Orozco-Barbosa <sup>3</sup>  and Diego Rodriguez-Torres <sup>4</sup>

<sup>1</sup> Telecommunications Department, National Autonomous University of Mexico, Mexico City 04510, Mexico; javiergo@comunidad.unam.mx

<sup>2</sup> Computer Department, National Autonomous University of Mexico, Mexico City 04510, Mexico

<sup>3</sup> Albacete Research Institute of Informatics, Universidad de Castilla-La Mancha, 02006 Albacete, Spain; luis.orocho@uclm.es

<sup>4</sup> Graduate Department, National Autonomous University of Mexico, Mexico City 04510, Mexico; diegeddy@gmail.com

\* Correspondence: jcamachoe@comunidad.unam.mx; Tel.: +52-5556-22-0866

† Current address: Engineering Faculty, Ciudad Universitaria, Coyoacán, Mexico City 04510, Mexico.

**Abstract:** WiFi is a widely used wireless technology for data transmission. WiFi can also play a crucial role in simultaneously broadcasting content to multiple devices in multimedia transmission for venues such as classrooms, theaters, and stadiums, etc. Broadcasting allows for the efficient dissemination of information to all devices connected to the network, and it becomes crucial to ensure that the WiFi network has sufficient capacity to transmit broadcast multimedia content without interruptions or delays. However, using WiFi for broadcasting presents challenges that can impact user experience, specifically the difficulty of obtaining real-time feedback from potentially hundreds or thousands of users due to potential collisions of feedback messages. This work focuses on providing accurate feedback to the Access Point about the percentage of users not receiving broadcast traffic correctly so it can adjust its Modulation and Coding Scheme (MCS) while transmitting broadcast multimedia content to many users. The proposed method is comprised of two sequential algorithms. In order to reduce the probability of a collision after transmitting each message, an algorithm searches for the best probability value for users to transmit ACK/NACK messages, depending on whether messages are received correctly or not. This feedback allows the Access Point to estimate the number of STAs correctly/incorrectly receiving the messages being transmitted. A second algorithm uses this estimation so the Access Point can select the best MCS while maintaining the percentage of users not receiving broadcast content correctly within acceptable margins, thus providing users with the best possible content quality. We implemented the proposed method in the ns-3 simulator, and the results show it yields quick, reliable feedback to the Access Point that was then able to adjust to the best possible MCS in only a few seconds, regardless of the user density and dimensions of the scenario.

**Keywords:** 802.11bc; wireless multimedia; rate adaption; broadcast



**Citation:** Gomez, J.; Camacho-Escoto, J.J.; Orozco-Barbosa, L.; Rodriguez-Torres, D. PROFEE: A Probabilistic-Feedback Based Speed Rate Adaption for IEEE 802.11bc. *Future Internet* **2023**, *15*, 396. <https://doi.org/10.3390/fi15120396>

Academic Editor: Giovanni Pau

Received: 27 October 2023

Revised: 1 December 2023

Accepted: 6 December 2023

Published: 9 December 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

A series of requirements are being established for the 802.11bc standard to provide efficient broadcast transmission without interfering with the existing 802.11 family standard. This new standard defines the parameters and functionalities necessary for optimal data dissemination within WiFi networks. The 802.11bc standard seeks to establish the foundations to optimize the transmission of information to multiple recipients simultaneously, such as in scenarios where it is necessary to send entertainment content to many users, for example, in stadiums or concert halls. The ultimate goal is to improve the efficiency and quality of broadcast transmission, maximizing the performance and capacity of the WiFi network without compromising compatibility with the existing 802.11 standards.

Various requirements have been placed on this new standard, described in [1]. First, the new 802.11bc must comply with the Project Authorization Request (PAR) and Criteria for Standards Development (CSD) established for it. It should provide authenticity protection for broadcast data messages. It must support modes of operation in a BSS (Basic Service System) infrastructure with association between the transmitter and receivers and cases without association. It must also include a mechanism to facilitate efficient diffusion in terms of energy consumption. Finally, it should provide a mechanism for the discovery of 11bc-compatible STAs and for the discovery of STAs that consume a specific broadcasting service.

The single most problematic aspect of WiFi broadcast is the inability of the Access Point (AP) to obtain real-time feedback from stations (STAs), as it becomes troublesome for users to send ACKs after receiving each message without causing a collision. Developing efficient and scalable feedback mechanisms from STAs in WiFi broadcast applications has historically been addressed through four distinct approaches. Each of these approaches offers advantages but also entails inevitable trade-offs. The first approach tries to decrease the number of transmitted feedback messages [2,3] but requires previous management among STAs (along with its signaling costs) so they all agree on which STAs can send feedback. The second approach makes the feedback messages collision-resistant [4,5]. However, implementing the scheme under this approach requires complex coding or sending the messages through a different communication channel. The third approach decreases the size of the feedback messages [6]; nevertheless, the problem remains as the number of STAs increases, even for small feedback messages. Finally, some approaches transmit broadcast messages into unicast [7–10]. However, such solutions must be properly scaled since it may take a long time to complete the feedback transmission process, depending on the number of receivers. One aspect they all have in common is that they seek to ensure that the ACKs do not collide, thus allowing the AP to obtain feedback from the STAs.

In this work, we propose a method comprised of two sequential algorithms to reduce the collision probability of the feedback messages. Our method is based on the foundation that STAs should not send feedback messages every time after receiving a message correctly/incorrectly from the AP but should transmit ACK/NACK only with a given probability. The first algorithm allows the AP to search for the best probability value for STAs to transmit ACK/NACK messages, depending on the percentage of successfully received feedback messages, collided feedback messages, or no feedback (silences). This feedback allows the AP to estimate the number of STAs receiving messages correctly/incorrectly. In the second algorithm, the AP uses this feedback to adjust the MCS to increase the data rate at the physical layer while keeping the percentage of STAs that do not receive messages correctly within a predefined range. This data rate adaption allows the AP to send better-quality content to a high proportion of the STAs in the venue, thus improving user experience. We tested the proposed method in different scenarios, varying the number of STAs and the venue's dimensions and showing that it provides fast-reliable feedback, allowing the AP to select the best MCS in only a few seconds. The method proposed in this work goes around most of the trade-offs in previous works. Our proposal does not require previous management among STAs or a different channel to send feedback messages, it has no limitations on the number of STAs, and it takes only a few seconds to complete, independent of the size and density of the scenario. An essential distinguishing feature of our proposal is its ability to monitor collisions related to feedback messages, allowing for the estimation of the number of STAs receiving messages accurately or encountering errors.

There are many challenges for next-generation networks to improve communications capabilities. As stated in [11], these shortcomings include spectral efficiency, coverage, interference, security, and privacy. These concerns and those presented in this work contribute to developing next-generation networks that enable high-throughput, low-delay communications.

This work makes the following specific contributions:

- We propose a method that quickly finds the best probability value for STAs to send ACK/NACK messages back to the AP. This allows a fast and accurate estimation of the number of STAs receiving messages correctly/incorrectly.
- Apart from using the number of ACK/NACK messages received correctly to estimate the number of STAs receiving messages correctly/incorrectly, the algorithm can also perform the very same estimation by counting the number of collisions of feedback messages and even the number of no feedback transmissions (silences) occurring after the transmission of each broadcast message. This technique allows redundancy as it provides the AP with three simultaneous and independent ways to perform the same estimation of STAs receiving messages correctly/incorrectly.
- We propose a second algorithm that uses the previous estimation to allow for the AP to adapt the MCS to increase the data rate at the physical layer while maintaining the percentage of STAs that do not receive broadcast messages correctly within acceptable limits, thus allowing for higher-quality content to be sent to most STAs, improving user experience.
- We implemented the complete method in ns-3 under diverse network conditions. We found that it provides quick (only a few seconds on average) and reliable (less than 5% estimation error) feedback, allowing the AP to rapidly adjust the MCS to maximize the users' experience.

The rest of the paper is organized as follows: Section 2 shows the most relevant works in the literature related to WiFi broadcast and how the proposed method sets apart from them. Section 3 presents an algorithm that searches for the best probability of transmitting ACK/NACK messages to estimate the number of STAs receiving messages correctly/incorrectly. Section 4 presents an algorithm that uses this estimation to select the best MCS to increase the speed rate at the physical layer, thus allowing for better-quality content to be sent to users. Section 5 details the implementation of the two algorithms, the conducted experiments, and the obtained results. Next, Section 6 presents a discussion of the findings and drawbacks of the proposed method. Finally, Section 7 presents our conclusions.

## 2. Related Work

Inherently, broadcast in the 802.11 standards is defined to be implemented without any acknowledgment (ACK) [12]. As mentioned, if more than one STA receives the broadcast message correctly, their corresponding ACKs will likely collide in the AP. However, sending no acknowledgment brings up many problems related to the uncertainty of the broadcast mechanism because it is impossible for the AP to know whether broadcast messages are being received correctly. This problem makes it difficult for any rate-adaptation algorithms to work properly. A trivial solution for this problem is continuously broadcasting messages with the lowest MCS available. Although this measure ensures that the maximum number of STAs receive broadcast messages correctly, the data rate will be the lowest, which might lead to a degradation in the quality of the broadcast content and the associated poor user experience.

The problem of adapting the data rate has been addressed from four different perspectives in the past. In general terms, the first approach tries to decrease the number of feedback messages, the second approach makes the feedback messages collision-resistant, the third approach decreases the size of the feedback messages, and the last approach turns the broadcast messages into unicast (thus avoiding collisions since unicast ACKs are transmitted at different times). However, one aspect they all have in common is that they seek that the ACKs do not collide, thus allowing the AP to obtain feedback from the receiving STAs.

In the first category, we have solutions to minimize the number of feedback messages sent by receiving STAs. The LBP [2] (Leader Based Protocol) uses the Automatic Repeat Request (ARQ) mechanism with ACK combined with leader-based techniques to minimize the number of sent ACKs, thus increasing the probability of ACK reception. Later, Ref. [3] used probabilistic NACK suppression to decrease the number of sent NACKs.

A second approach tries to decrease the size of the feedback messages to decrease their probability of collision. In [6], the authors explored using ARQ messages combined with virtual ACK bitmaps significantly shorter than the actual ACK/NACK messages. The AP assigns small time slots to each STA in which they will answer with ACK pulses. Although this technique prevents collisions, it assumes that STAs previously received the time distribution correctly.

A third solution makes the ACK/NACK collision-resistant, as in [4]. Although this proposal needs extra hardware, it increases the probability that the feedback is received correctly, even if obtained from collisions. Later, HIMAC [5] proposes using Unary Channels to transmit the feedback to overcome collisions.

The last approach is to turn the broadcast messages into individual unicast messages as introduced in the BMW protocol proposed in [7], which was further improved in the BMMM protocol in [8] by grouping the transmissions into a single one and introducing Request for ACK (RAK) messages. Then, the author of [9] proposed an improvement that reduced the number of RAK transmissions. Later, a similar approach in which clients can request the AP to send a multicast stream as unicast was incorporated in the Directed Multicast Service (DMS) of the IEEE 802.11v amendment [10].

Apart from these four general categories, other particular approaches deal with similar problems. One of these approaches overcomes reception errors of multicast messages using the Groupcast with Retries (GCR), adopted in the IEEE 802.11aa amendment [13]. Further, Ref. [14] combined GCR with the Adaptive Auto Rate Fallback (AARF) method to change the modulation to decrease the Packet Loss Ratio.

More recent solutions, such as the one presented by T. Kanda et al. [15], use the RSSI of unicast ACKs coming from non-eBCS APs and deep reinforcement learning to estimate the eBCS STAs. Their work assumes that the distribution on the eBCS and non-eBCS STAs is the same, which might only sometimes occur. Further, the authors use distributional reinforcement learning in [16] and evaluated their algorithm regarding the data rate of the broadcast messages and the reception success rate at STAs.

In [17], the authors proposed MuDRA, a rate adaption algorithm for multicast flows. MuDRA relies on the information of the  $K$  STAs with the worst channel conditions, which the AP selects. Once selected, these STAs send their packet delivery ratio after a short random interval to avoid collisions. In order to reach the target rate, the proposed algorithm monitors the number of devices switching from PDR of 100% to 0% to satisfy the maximum permitted number of abnormal devices. Authors in [18] designed and tested AMuSe, a mechanism to broadcast packets to a large group of STAs. AMuSe relies on the cluster formation of adjacent neighbors and feedback from cluster heads. Although this might be a good technique in static scenarios, cluster formation might lead to long delays and lots of overhead in the presence of mobility. AMuSe provides rate adaption provided by MuDRA [17] to transmit at the highest possible bitrate while meeting some constraints. AMuSe is the only proposal verified on a testbed with over 150 STAs [19].

Further, the authors of [20] proposed ATRaCt, a cross-layer rate control for wireless multicast that takes information from the transport layer to modify the data rate of the link layer. The transport layer information is taken using the NORM protocol [3]. Also, ATRaCt periodically sends MSRq control messages using the base transmission rate to get statistics that are obtained via MSRp messages.

Table 1 shows a summary of the above literature review, which denotes that most efforts related to feedback acquisition in broadcast services consider feedback from ACKs and/or NACKs in various ways. Other works use RTS/CTS messages to avoid collisions, and many seek the reception of every message from all STAs. Different from all these works, our proposal not only introduces a probability in the transmission of ACK/NACK messages that allows for the estimation of the number of STAs receiving messages correctly/incorrectly but also, from the percentage of both, collisions and absent periods (silence). The latter refers to the scenario in which the AP does not receive an ACK/NACK message in response to the transmission of a broadcast message. In other words, our pro-

positional can estimate the number of STAs receiving messages correctly/incorrectly from three different events simultaneously and independently of each other (i.e., successfully received ACK/NACK messages, ACK/NACK collisions, and ACK/NACK silences). The proposed method relies on traditional ACK/NACK transmissions. It requires no additional signaling or hardware to quickly and reliably estimate the percentage of STAs receiving messages correctly/incorrectly at the present MCS, which only takes a few seconds to converge to the actual number of STAs. On top of that, our method can use this estimation to select the best possible MCS while keeping the percentage of STAs that do not receive broadcast transmission under the desired levels.

**Table 1.** Comparison of the state-of-the-art algorithms (fb: feedback).

	[2]	[3]	[6]	[4]	[5]	[7]	[8]	[9]	[15]	[16]	[20]	[18]	[13]	[14]	PROFEE
Avoid collisions of fb	X	X	X	X	X	X	X	X	X	X	X	X	X	X	-
Decrease number of fb messages	X	X	-	-	-	-	-	-	-	-	-	X	X	X	X
Make the fb collision resistant	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-
Decrease the size of the fb	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
Turns the broadcast into unicast	-	-	-	-	-	X	X	X	-	-	-	-	-	-	-
Indirect fb from unicast packets	-	-	-	-	-	-	-	-	X	X	X	-	-	-	-
Based on clusters	X	X	-	-	-	-	-	-	-	-	-	X	-	-	-
Consider positive fb	-	-	X	-	X	X	X	X	X	X	X	-	-	X	X
Consider negative fb	X	X	-	X	X	-	-	-	-	-	-	-	-	-	X
Considers silence fb	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-
Seeks 100% receptions	X	-	X	X	X	X	X	-	-	-	-	-	-	-	-
Random-delayed fb	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-
Time slotted pre-assigned fb	-	-	X	-	-	-	X	X	-	-	-	-	-	-	-
Use RTS/CTS	X	-	-	X	X	X	X	-	X	X	-	-	-	-	-

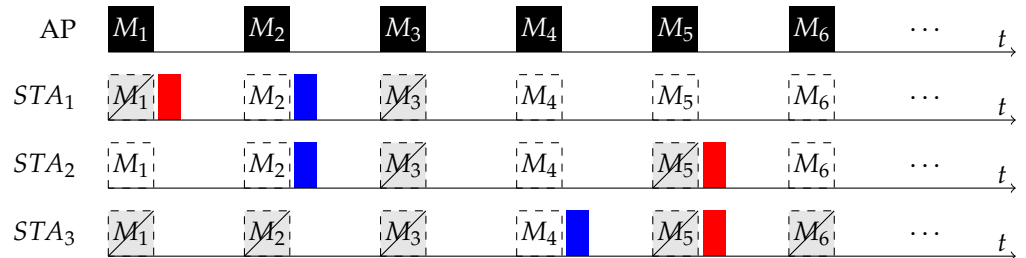
### 3. Estimation of the Number of STAs Receiving Broadcast Messages Correctly/Incorrectly

This section presents an algorithm that allows for the AP to estimate the number of STAs that are receiving broadcast messages correctly ( $n_{ACK}$ ) or incorrectly ( $n_{NACK}$ ). The solution uses probabilistic ACKs/NACKs as feedback from STAs. When an STA correctly decodes the preamble and the corresponding message payload, it sends an ACK to the AP with probability  $p_{ACK}$ . Similarly, when an STA correctly decodes the preamble of a message but incorrectly decodes the rest of the payload, it sends a NACK to the AP with probability  $p_{NACK}$ . This scheme is aided by the preambles being always sent using BPSK with a 1/2 coding rate [12] (the lowest MCS) to maximize the probability of preamble reception. STAs know precisely which values of  $p_{NACK}$  and  $p_{ACK}$  to use because the AP includes those probabilities within each message it transmits. In this algorithm, the AP performs an active role in selecting the best values of  $p_{NACK}$  and  $p_{ACK}$  that will result in the best estimation of  $n_{ACK}$  and  $n_{NACK}$ . On the other hand, STAs play a passive role as they simply react to sending feedback messages according to the probabilities included in each message.

To avoid ACK and NACK colliding in the AP after each transmitted message, NACKs are only sent in response to the odd-sequenced AP messages, and ACKs are sent only in response to even-sequenced AP messages. This work assumes no previous association of STAs with the AP, as this is one requirement of the new 802.11bc standard. This condition makes the problem harder, as the AP does not know the total number of listening STAs and their distribution over the network area. However, in those scenarios in which the number of STAs is known in advance, it will simplify the protocol’s operation as it is only necessary to estimate either  $n_{ACK}$  or  $n_{NACK}$  to calculate the other.

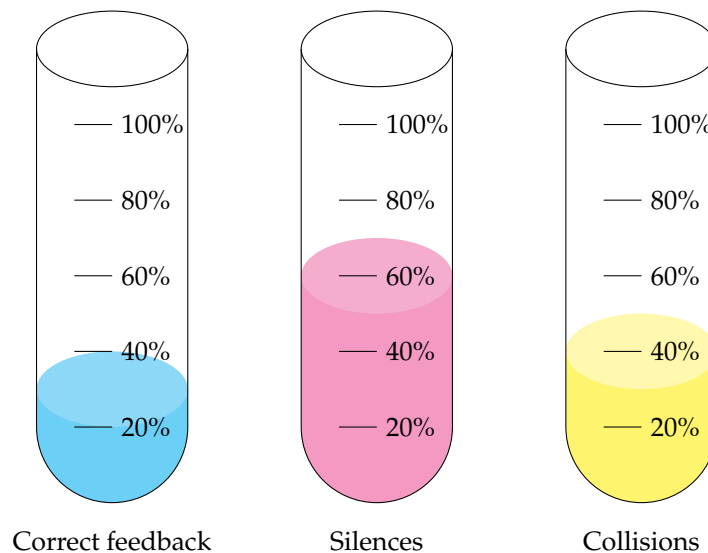
Figure 1 illustrates the operation of the proposed algorithm with three STAs, where transmitted NACKs and ACKs are represented as red and blue rectangles, respectively. After sending message  $M_1$ ,  $STA_1$  and  $STA_3$  received it incorrectly, while  $STA_2$  received it correctly.  $STA_1$  and  $STA_3$  will transmit a NACK with probability  $p_{NACK}$ . In this example,

only  $STA_1$  sends the NACK, which is received in the AP. Message  $M_2$  is received correctly by  $STA_1$  and  $STA_2$ , and both transmit an ACK with probability  $p_{ACK}$ . In the example, both STAs transmitted an ACK, resulting in a collision in the AP. For message  $M_3$ , none of the three STAs received it correctly, and none transmitted a NACK, resulting in a silence event. Similarly, message  $M_4$  resulted in a correct reception of one ACK,  $M_5$  resulted in a collision of NACKs, and  $M_6$  resulted in a silence event.



**Figure 1.** Example of ACK and NACK events for three receiving STAs. Black squares: transmitted messages, red squares: NACKs, blue squares: ACKs, white squares: correctly received messages, grey-crossed squares: incorrectly received messages.

To estimate the number of STAs receiving messages correctly/incorrectly, the AP waits for  $2f$  messages ( $f$  messages for NACKs and  $f$  messages for ACKs)—which we call a *frame*—and counts how many of these messages resulted in receiving ACK/NACK messages from only one STA (no collision), received no ACK/NACK feedback (silence), and received multiple ACK/NACK messages from more than one STA (collisions). The AP then counts the number of messages that fall into each of the previous three categories, as shown in Figure 2, where it is possible to see in this example that, for 20% of transmitted messages, the AP could decode a feedback message (i.e., correct feedback), for 50% of the messages no feedback response was received (i.e., silence), and for 30% of the messages more than one response was received causing a collision. In most cases, the distribution of the three events will be different depending on whether it corresponds to ACK or NACK messages, the probabilities of sending ACK/NACK feedback, the density of STAs, and the dimensions of the scenario.



**Figure 2.** Example of the distribution of the three events at the end of a frame.

Not all possible values for  $p_{ACK}$  and  $p_{NACK}$  allow the AP to accurately estimate the number of STAs receiving messages correctly/incorrectly for a finite number of messages ( $2f$ ). However, when  $p_{ACK}$  and  $p_{NACK}$  are tuned correctly, the probabilistic estimation of either number of STAs becomes quite accurate, as shown later. With the transmitted

messages associated with one of the three possible events, the AP can estimate the number of STAs receiving messages correctly/incorrectly using standard probability functions.

The expected number of correct ACK/NACK receptions after  $2f$  messages are transmitted is shown in Equations (1) and (2), respectively.

$$E[ACKCorrect] = f \cdot n_{ACK} \cdot p_{ACK}(1 - p_{ACK})^{n_{ACK}-1} \tag{1}$$

$$E[NACKCorrect] = f \cdot n_{NACK} \cdot p_{NACK}(1 - p_{NACK})^{n_{NACK}-1}. \tag{2}$$

$E[ACKCorrect]$  and  $E[NACKCorrect]$  are the number of ACKs and NACKs received correctly by the AP after transmitting  $2f$  messages, respectively.  $p_{ACK}$  and  $p_{NACK}$  are the probabilities that a STA transmits an ACK or NACK message. Finally,  $n_{ACK}$  and  $n_{NACK}$  are the number of STAs that received messages correctly/incorrectly after  $2f$  messages were transmitted, respectively.

Similarly, the expected number of silences related to ACK and NACK messages after  $2f$  messages were transmitted become

$$E[ACKSilence] = f \cdot (1 - p_{ACK})^{n_{ACK}} \tag{3}$$

$$E[NACKSilence] = f \cdot (1 - p_{NACK})^{n_{NACK}}. \tag{4}$$

Finally, the expected number of collisions related to ACK and NACK messages after  $2f$  messages were transmitted become

$$E[ACKCollision] = f - E[ACKCorrect] - E[ACKSilence] \tag{5}$$

$$E[NACKCollision] = f - E[NACKCorrect] - E[NACKSilence]. \tag{6}$$

From these equations, the AP can compute  $n_{ACK}$  using either Equations (1), (3), or (5). Similarly, the AP can compute  $n_{NACK}$  using either Equations (2), (4), or (6). Thus, the AP has three different, independent, and simultaneous ways of computing the number of STAs receiving messages correctly/incorrectly. In theory, the three ways should provide the same estimation, which adds robustness to the system; however, as we will see soon, depending on the chosen probabilities, not all three alternatives provide an accurate estimation for a finite number of messages. Similarly, in some scenarios, the capture effect might change a collision into a correct reception in the AP, thus introducing additional errors while using the correct feedback and collision estimators.

The only equation with a closed form when solving for  $n_{ACK}$  is Equation (3),

$$n_{ACK} = \frac{\log\left(\frac{E[ACKSilence]}{f}\right)}{\log(1 - p_{ACK})}, \tag{7}$$

and the only equation with a closed form when solving for  $n_{NACK}$  is Equation (4),

$$n_{NACK} = \frac{\log\left(\frac{E[NACKSilence]}{f}\right)}{\log(1 - p_{NACK})}. \tag{8}$$

Equation (7) is valid while  $0 < p_{ACK} < 1$  and  $E[ACKSilence] > 0$ , while Equation (8) is valid when  $0 < p_{NACK} < 1$  and  $E[NACKSilence] > 0$ . Although there is no closed form for the other equations, they can be solved via numerical methods.

In order to illustrate the error and estimation convergence time, we evaluated the expected error by comparing the number of STAs obtained from Equation (8), and the number of STAs obtained via simulations for estimating  $n_{NACK}$  using the *silence* estima-

tor. The experiment was performed in Python for  $f = 1000$  messages and 10, 100, and 1000 STAs. Each STA transmitted with probability  $p_{NACK}$  in each slot and did not transmit with probability  $1 - p_{NACK}$ . When no STA transmitted a NACK, a silence event was counted. A collision event was counted when more than one STA transmitted a NACK, and when exactly one STA transmitted a NACK, a successful NACK reception was counted. After 1000 messages,  $n_{NACK}$  was estimated using Equation (8), and the estimation error was computed. This process was repeated 10,000 times to obtain the expected error for such probability. Figure 3 shows the results of the simulations for evaluating the expected error for estimating  $n_{NACK}$  for different values of  $p_{NACK}$  using the expected number of silences  $E[NACKSilence]$ . The plots also show the analytic plot of Equation (4) for  $n_{NACK} = 10$ ,  $n_{NACK} = 100$ , and  $n_{NACK} = 1000$ . These plots show an area (red rectangle) where the percentage of silences minimizes the expected error in estimating  $n_{NACK}$  after 1000 messages. Notice that, in these figures, this is between 15% and 45% of silence events, which we will call  $prop_{NACK}^{inf}$  and  $prop_{NACK}^{sup}$ , respectively. Notice that, although the simulations are performed for estimating  $n_{NACK}$ , the same process coincides with estimating  $n_{ACK}$ .

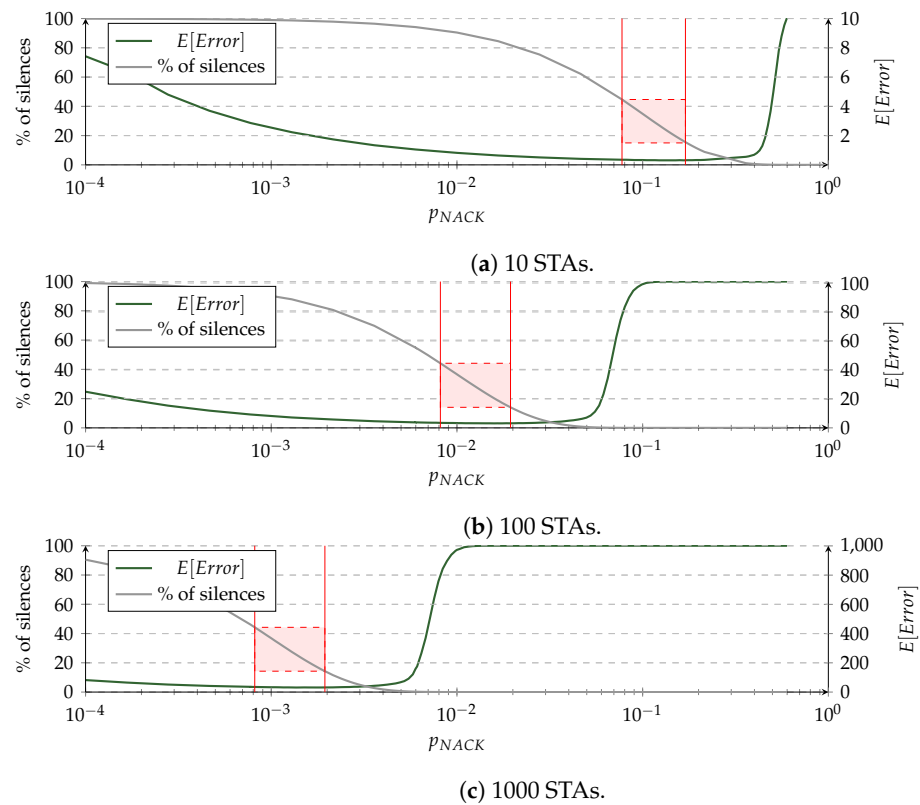


Figure 3. Expected error obtained with the silence estimator after 1000 messages.

To achieve the percentage of silences that minimizes the expected error in the estimation of  $n_{ACK}$  and  $n_{NACK}$ , we propose a simple algorithm that iterates over different values of  $p_{ACK}$  and  $p_{NACK}$ , respectively, until it finds one that satisfies that the target percentage of silence events, in this case, is between  $prop_{ACK}^{inf}$  and  $prop_{ACK}^{sup}$  (or  $prop_{NACK}^{inf}$  and  $prop_{NACK}^{sup}$ ). The proposed algorithm is similar to a binary search algorithm as shown in Algorithm 1. On each iteration, the value of  $p_{ACK}$  or  $p_{NACK}$  will logarithmically approach a value that satisfies  $prop_{ACK}^{inf} \leq p_{ACK} \leq prop_{ACK}^{sup}$  (or  $prop_{NACK}^{inf} \leq p_{NACK} \leq prop_{NACK}^{sup}$ ). Notice that the maximum value for  $p_{ACK}$  and  $p_{NACK}$  is 0.1, as shown in lines 13–15 of the algorithm, which works well for more than 10 STAs.



**Algorithm 1** Algorithm to tune  $p_{ACK}$  to achieve minimum expected error on  $n_{ACK}$  estimation

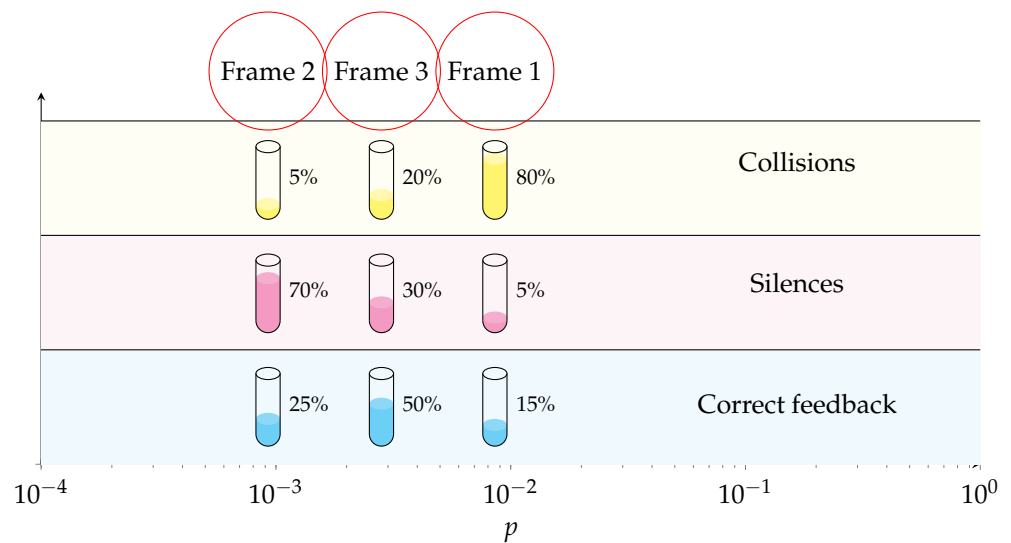
```

1:  $p_{ACK} \leftarrow 10^{-2}$ 
2:  $diff \leftarrow 1$  magnitude order
3: while  $prop_{ACK}^{sup} \leq prop \leq prop_{ACK}^{inf}$  do
4:   if  $f$  messages passed until the last check then state  $prop \leftarrow$  percentage of si-
     lence/collision/reception detected
5:     if  $prop > prop^{sup}$  then
6:        $diff \leftarrow diff/2$ 
7:        $p \leftarrow p + diff$ 
8:     else if  $prop < prop^{inf}$  then
9:        $diff \leftarrow diff/2$ 
10:       $p_{ACK} \leftarrow p_{ACK} - diff$ 
11:     end if
12:     if  $p_{ACK} > 0.1$  then
13:        $p_{ACK} \leftarrow 0.1$ 
14:       break
15:     end if
16:   end if
17: end while

```

Although the algorithm is presented for  $n_{ACK}$ , a similar one works for  $n_{NACK}$  by changing  $p_{ACK}$  for  $p_{NACK}$  and  $prop_{ACK}^{inf}$  and  $prop_{ACK}^{sup}$  for  $prop_{NACK}^{inf}$  and  $prop_{NACK}^{sup}$ . When an appropriate value of  $p_{ACK}$  or  $p_{NACK}$  is found, the expected error in the estimation will be the minimum, which means that the AP will estimate  $n_{ACK}$  or  $n_{NACK}$  as accurately as possible.

Figure 4 shows an example of how the first algorithm works for ACKs when  $prop_{ACK}^{inf} = 15\%$  and  $prop_{ACK}^{sup} = 45\%$  as defined earlier. In Frame 1, the algorithm chooses  $p_{ACK} = 10^{-2}$ . Since the percentage of silences is below 15% at the end of Frame 1,  $p_{ACK}$  switches to  $10^{-3}$  for Frame 2. Now the percentage of silence events is over  $prop_{ACK}^{sup} = 45\%$ , and  $p_{ACK}$  switches to the logarithmic halfway between  $10^{-2}$  and  $10^{-3}$  ( $p_{ACK} = 10^{-2.5}$ ) for Frame 3, that is  $10^{-2.5}$ . Finally, at the end of Frame 3, the percentage of silences is 30%, which is within  $prop_{ACK}^{inf}$  and  $prop_{ACK}^{sup}$ ; thus, the first algorithm ends and the AP proceeds to estimate  $n_{ACK}$ .



**Figure 4.** Algorithm for finding the  $p_{ACK}$  or  $p_{NACK}$  that minimizes the error.

### 4. Adaptive Modulation Algorithm

The general purpose of an adaptive modulation algorithm is to find the best MCS that maximizes the physical transmission speed at the physical layer with a low Bit Error Rate (BER). Historically, these algorithms have been used in unicast scenarios, where the AP adapts the MCS independently for each user. Each MCS changes the number of bits encoded in each symbol and the coding rate, resulting in a different transmission speed at the physical layer. Figure 5 shows an example of the network coverage for MCS(0)-MCS(8). A disk with a different color represents the area covered by a different MCS. Notice that MCS(0) covers 53.41% of the total area, while MCS(8) covers only 0.01%. Table 2 shows the different MCSs and their characteristics for the IEEE 802.11ax standard. The same table shows (in the right-hand part) the various H.265 video levels that can be supported by the corresponding MCS on the left. Each H.265 level provides a different video resolution and refresh rate, generating a different transmission bandwidth. This table does not show an exhaustive list of all the different H.265 levels but lists some to illustrate how a different MCS will support all H.265 levels up to a particular level. For example, MCS(5) can support all H.265 levels up to level 5.1. In summary, the proposed method in this work will look for the best MCS(i) that supports the best video quality to users yet keeps the percentage of STAs receiving no video content correctly within the desired tolerated limits.

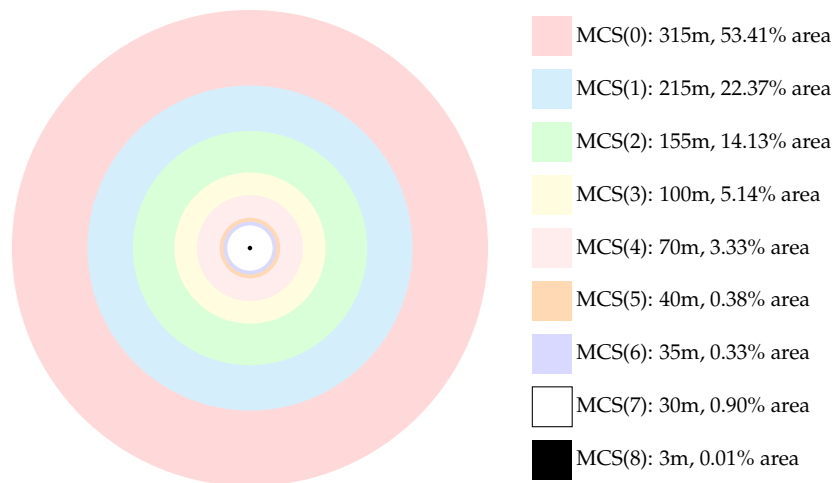


Figure 5. Coverage of each MCS.

Table 2. OFDM MCS for 802.11ax for one spatial stream, 20 MHz channel, and 3.2 μs GI (first three columns). H.265 video resolution and approximate bandwidth required for different levels (last three columns).

MCS(i) Index	Speed [Mbps]	Modulation and Coding Rate (R)	Video Resolution	~Bandwidth Required	Level
MCS(0)	7.3	BPSK, R = 1/2	1280 × 720 at 60 fps	4 Mbps	4
MCS(1)	14.6	QPSK, R = 1/2	1920 × 1080 at 60 fps	9 Mbps	4.1
MCS(2)	21.9	QPSK, R = 3/4			
MCS(3)	29.3	16-QAM, R = 1/2	3840 × 2160 at 30 fps	19 Mbps	5
MCS(4)	43.9	16-QAM, R = 3/4			
MCS(5)	58.5	64-QAM, R = 2/3	3840 × 2160 at 60 fps	40 Mbps	5.1
MCS(6)	65.8	64-QAM, R = 3/4			
MCS(7)	73.1	64-QAM, R = 5/6			
MCS(8)	87.8	256-QAM, R = 3/4	3840 × 2160 at 120 fps	80 Mbps	5.2
MCS(9)	97.5	256-QAM, R = 5/6			
MCS(10)	109.7	1024-QAM, R = 3/4			
MCS(11)	121.9	1024-QAM, R = 5/6			

In the case of WiFi broadcast, one transmission performed by the AP is expected to reach hundreds, if not thousands, of receiving STAs simultaneously. Thus, the concept of an adaptive

modulation algorithm takes a slightly new turn, and now it seeks to find the best MCS that maximizes the transmission speed with a low BER for a given percentage of STAs.

In the previous section, we proposed an algorithm that estimates, for a current MCS( $i$ ),  $n_{ACK}$  and  $n_{NACK}$ . Thus, we can compute the percentage of STAs receiving messages incorrectly ( $\%n_{NACK}$ ) as

$$\%n_{NACK} = \frac{n_{NACK}}{n_{ACK} + n_{NACK}} \times 100. \quad (9)$$

We now define  $\%n_{NACK}^{range} = [\%n_{NACK}^{min}, \%n_{NACK}^{max}]$  as a range of values of  $\%n_{NACK}$  that is considered acceptable, and then the Algorithm 2 modifies (or not) the current MCS( $i$ ) at the end of each frame of  $2f$  messages according to

$$MCS(i) = \begin{cases} MCS(i+1), & \%n_{NACK} < \%n_{NACK}^{min} \\ MCS(i-1), & \%n_{NACK} > \%n_{NACK}^{max} \\ MCS(i), & \%n_{NACK}^{min} \leq \%n_{NACK} \leq \%n_{NACK}^{max} \end{cases} \quad (10)$$

From Equation (10), it results that the MCS( $i$ ) will be increased by one index if  $\%n_{NACK}$  is lower than the minimum tolerated percentage ( $\%n_{NACK}^{min}$ ) to test a higher MSC that provides higher data speed, decreased by one if  $\%n_{NACK}$  is above the maximum tolerated percentage ( $\%n_{NACK}^{max}$ ) to increase the number of STAs that received messages correctly. The MCS will remain the same index for the following  $2f$  messages as long as  $\%n_{NACK}$  is within  $\%n_{NACK}^{range}$ . The purpose of the adaptive modulation algorithm in the broadcast scenario will be to reach the third case, where the MCS( $i$ ) in the current frame remains the same in the next one (as long as the distribution of users does not change).

The reason why we defined a range to determine an acceptable percentage of STAs not receiving messages correctly ( $\%n_{NACK}^{range}$ ) is to force the AP to find an MCS( $i$ ) that keeps  $\%n_{NACK}$  under the desired maximum tolerated level ( $\%n_{NACK}^{max}$ ), but at the same time allows the AP to tolerate a minimum percentage of STAs with bad reception by setting  $\%n_{NACK}^{min} \geq 0$ . This strategy allows the AP to select a higher MCS( $i$ ), likely providing most users better-quality broadcast content. Ultimately, it is up to the network administrator and the specific media content to set the  $\%n_{NACK}^{range}$ . In the experiments section and for performance testing purposes, we used a  $\%n_{NACK}^{range} = [10, 20]\%$ .

---

#### Algorithm 2 Algorithm to select the MCS for the next cycle

---

```

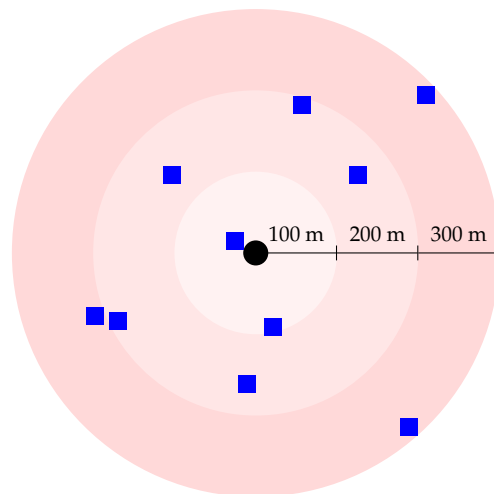
1: MCS ← MCS( $i$ )
2: if  $\%n_{NACK} \geq \%n_{NACK}^{max}$  then
3:   MCS ← MCS( $i-1$ )
4: else if  $\%n_{NACK} \leq \%n_{NACK}^{min}$  then
5:   MCS ← MCS( $i+1$ )
6: else if  $\%n_{NACK}^{min} \leq \%n_{NACK} \leq \%n_{NACK}^{max}$  then
7:   MCS ← MCS( $i$ )
8: end if

```

---

## 5. Experiments

We used ns-3 to implement and test the probability and modulation selector algorithms. We consider a scenario where  $n$  STAs are homogeneously distributed in a disk of radius  $R$  meters with one AP at its center, as shown in Figure 6. Table 3 shows the configuration used in the simulations.



**Figure 6.** Scenario for the experiments. Randomly distributed STAs at 100 m, 200 m, and 300 m.

**Table 3.** Simulation parameters.

Parameter	Value
Number of APs	One AP only located in the center of a disk
Number of STAs ( $n$ )	10, 100, and 1000 STAs random uniformly distributed within a disk
Disk Radius	100, 200 and 300 m
Propagation model	2.4 GHz frequency, Friis propagation model, Nist error model
Initial transmit power:	1 dBm
Final transmit power:	1 dBm
Spectrum Channel:	Standard: 802.11ax, Width 20 MHz
Mobility:	Both the AP and the STAs remain static
Network layer	Only a broadcast address is required.

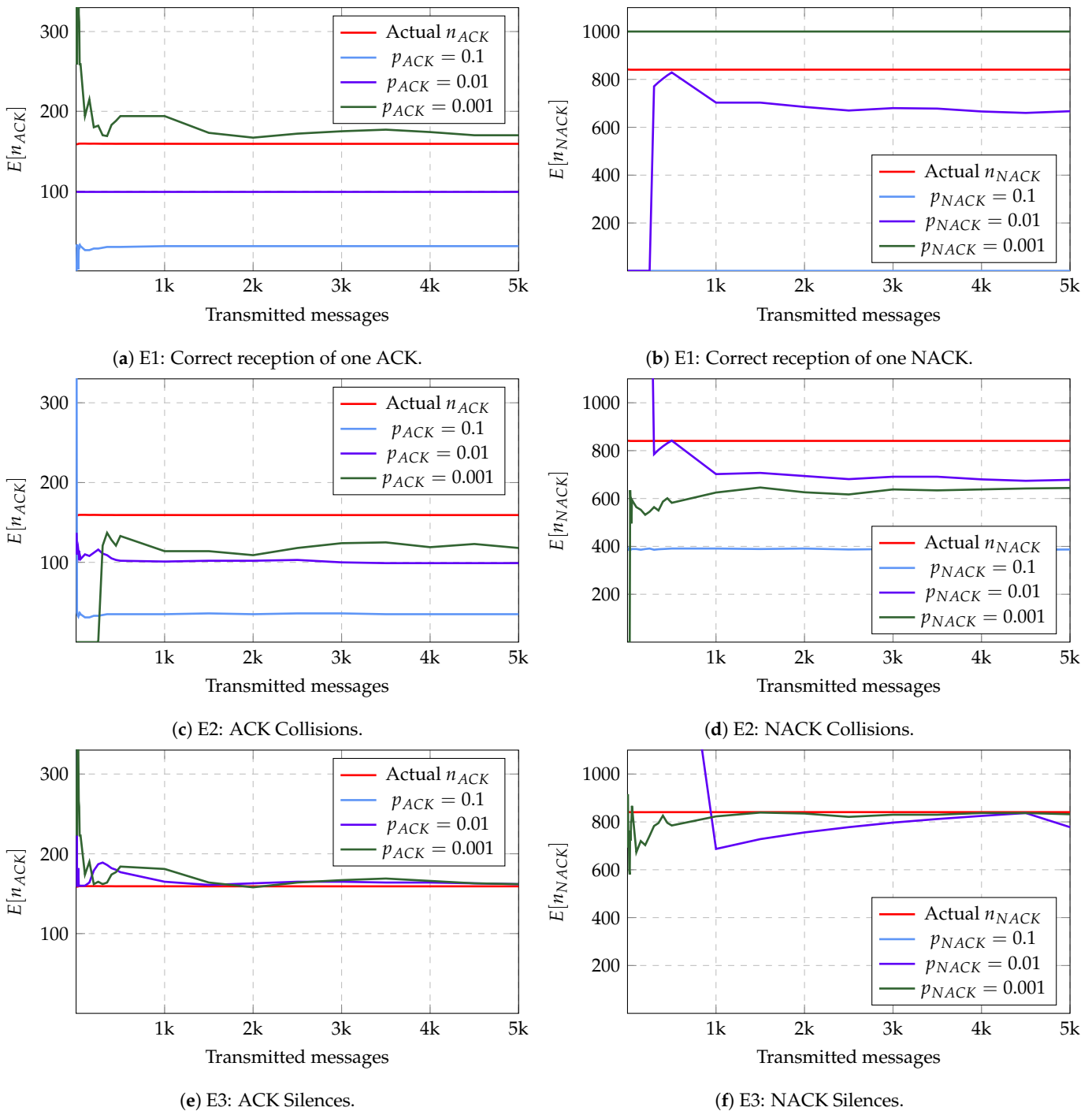
Finally, this work assumes that only the AP transmits broadcast messages, and STAs do not transmit any messages independently. However, the proposed method should work even if STAs are allowed to send unicast traffic to the AP. In this case, STAs will only reply with ACK and NACK messages to broadcast messages transmitted by the AP using the probabilities included in those messages. However, mixing downlink broadcast content with uplink traffic is not recommended since the available bandwidth for broadcast will shrink considerably, potentially compromising the quality of the transmitted video stream. This assumption can be relaxed by applying a protocol as the one described in [21].

### 5.1. Performance of the Three Estimators

The first part of the experiments is to analyze the performance of each of the three estimators for positive feedback (one ACK, ACK collision, and ACK silence) and negative feedback (one NACK, NACK collision, and NACK silence). As mentioned before in Section 3. Ideally, the three estimators should provide a similar estimation of either  $n_{ACK}$  or  $n_{NACK}$ , provided the AP transmits a vast number of messages (in theory is an infinite number). However, the AP is interested in obtaining an accurate estimation as soon as possible using a few messages. As we will see, for a small finite number of messages, each estimator will show a different accuracy and convergence time depending on the particular scenario and the probabilities being used ( $p_{ACK}$  and  $p_{NACK}$ ).

Figure 7 shows the results of the experiments. The AP computes the number of STAs receiving messages correctly/incorrectly using the three estimators and for three different probabilities (0.1, 0.01, and 0.001). There are 1000 STAs distributed in a disk of 100 m radius. Receiving STAs used even-numbered messages to transmit ACKs and odd-numbered messages for transmitting NACK messages to the AP. The highest MCS (MCS(5)) was used

in all these figures. The green horizontal line shows the real number of STAs receiving messages correctly/incorrectly in each experiment. As can be observed, each estimator has a different accuracy and convergence time depending on the used probability. However, each estimator presents some general behavior. The ACK/NACK estimator (see Figure 7a,b) presents significant errors and takes longer to converge to a steady state. The collision estimator (see Figure 7c,d) presents even more significant error variations, and it takes even longer to reach a steady state. Finally, the silence estimator (see Figure 7e,f) has good accuracy and reaches a steady state after a few hundred messages.



**Figure 7.** Performance of three estimators for 1000 STAs randomly positioned at 100 m. In this scenario, 841 STAs receive messages correctly, and 159 STAs receive them incorrectly.

Table 4 shows the absolute estimation error (measured in the number of STAs) in the steady state for 10, 100, and 1000 STAs distributed in the same 100 m radius disk. The table shows results after 2000 and 5000 messages. The estimation that presents the smaller error in each cell for a given number of messages is highlighted for each probability. As verified in the table, the *silence* estimator offers the most accurate estimation in 14 out of the 18 cells shown for 2000 messages and in 15 out of 18 cells for 5000 messages, the ACK/NACK correct feedback estimator in 4 out of 18 cells for 2000 messages and in 3 out of 18 cells for 5000 messages, and the collision estimator in 4 out of 18 cells for 2000 messages and in 5 out of 18 cells for 5000 messages. Cells where the silence estimator did not achieve the lowest error occurred when  $p_{ACK}$  or  $p_{NACK}$  were far from their optimal value. Figure 7 shows that the silence estimator also reaches the steady state faster than the other two.

As Table 4 shows, it becomes clear why we cannot use all three estimators and why we selected the *silence* estimator for the first algorithm of the proposed method that searches for a probability that guarantees low estimation error in a short time. It is worth noticing from this table that the estimation obtained after 2000 messages is very similar to the one obtained for 5000 messages, which shows that it is unnecessary to wait for 5000 messages to get an accurate estimate on  $n_{ACK}$  and  $n_{NACK}$ .

**Table 4.** Absolute error of three estimators after 2 k and 5 k messages for 10, 100, and 1000 STAs, radius = 100 m.

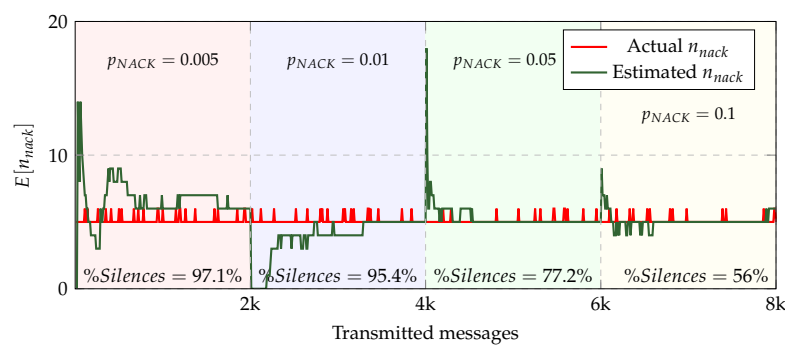
STAs	$p$ ( $p_{ACK}$ or $p_{NACK}$ )	Estimator	$n_{ACK}$ Error [STAs]		$n_{NACK}$ Error [STAs]	
			$2f = 2k$	$2f = 5k$	$2f = 2k$	$2f = 5k$
n = 10	0.001	Correct feedback	0	0	1	0
		Collisions	3	3	7	7
		Silences	0	0	1	0
$n_{NACK}^{real} = 7$	0.01	Correct feedback	510	526	394	392
		Collisions	3	0	3	2
		Silences	0	0	0	1
$n_{ACK}^{real} = 3$	0.1	Correct feedback	17	17	2	2
		Collisions	0	0	3	3
		Silences	0	0	0	0
n = 100	0.001	Correct feedback	0	1	9	5
		Collisions	18	18	82	46
		Silences	0	1	12	2
$n_{NACK}^{real} = 82$	0.01	Correct feedback	267	269	17	17
		Collisions	4	7	20	21
		Silences	1	0	2	0
$n_{ACK}^{real} = 18$	0.1	Correct feedback	9	9	22	22
		Collisions	7	7	19	19
		Silences	0	0	1918	1918
n = 1000	0.001	Correct feedback	8	11	159	159
		Collisions	50	41	215	197
		Silences	1	3	6	9
$n_{NACK}^{real} = 841$	0.01	Correct feedback	60	60	156	174
		Collisions	57	60	147	163
		Silences	4	3	85	63
$n_{ACK}^{real} = 159$	0.1	Correct feedback	128	128	841	841
		Collisions	124	124	450	454
		Silences	1841	1841	1159	1159

### 5.2. Probability Selection Algorithm

Something to extract from Figure 7 is that each of the three estimators performs better in accuracy and convergence for a particular probability, which is the basis of the probability selector algorithm. Now, we will show the algorithm’s operation that searches for a probability that provides the best estimation of the number of STAs receiving messages correctly/incorrectly, employing the silence estimator only.

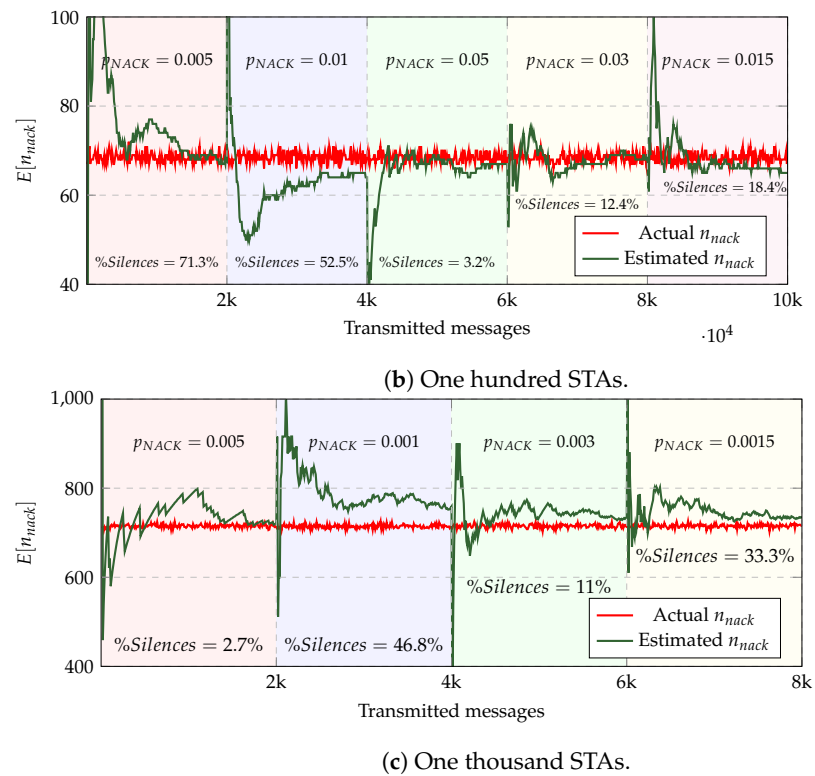
Figure 8 shows three plots of simulations in ns-3 of the probability selector algorithm ( $p_{NACK}$ ) for 10, 100, and 1000 STAs positioned within a 100 m disk and the AP uses MCS(5) to transmit messages. The silence percentage (%Silences) bounds  $[prop_{NACK}^{inf}, prop_{NACK}^{sup}]$  for these experiments were set between 15% and 45% as discussed before in Section 3. The red line represents the actual number of STAs that did not receive the messages correctly (which may vary over time depending on channel variations at the physical layer). The green line represents the estimation of the number of STAs that did not receive the messages correctly up to the given message (that is, using the number of silences up to that time) using the silence estimator. Every 2000 messages, the algorithm changes  $p_{NACK}$  according to Algorithm 1, and at this time, the counter of silence events restarts. This operation generates significant estimation errors after the counter restarts since it could involve divisions by a small number, as defined in Equation (7) and (8).

Figure 8a shows that for 10 STAs, the probability selector algorithm increased the value of  $p_{NACK}$  until it reached 0.1 at the end of Frame 4, which provides a 56% of silence events. Notice that even if this percentage is higher than the upper limit (45%), 0.1 is the maximum probability allowed by the algorithm. Due to the reduced number of STAs in this simulation, the error is of, at most, one STA in a steady state. On the other hand, Figure 8b shows the results for 100 STAs (the number of STAs that did not receive the messages correctly is around 70). For 100 STAs,  $p_{NACK}$  started at 0.005 in Frame 1 and ended at 0.015 in Frame 5, where the percentage of silences reached 18.4%, within the desired range. Finally, Figure 8c shows the results for 1000 STAs. In this case, the value of  $p_{NACK}$  went up and down in various Frames until it reached 0.0015 at the end of Frame 4, achieving a 33% of silence events, which is between the 15–45% bounds. From these plots, it is possible to notice that the final value of  $p_{NACK}$  selected by the algorithm corresponds to approximately  $\frac{1}{n_{NACK}}$ , which occurred without the AP having any previous knowledge about the number of STAs it is serving.



(a) Ten STAs.

Figure 8. Cont.



**Figure 8.** Algorithm for obtaining the  $p_{NACK}$  that minimizes the error for  $n_{NACK}$  using the silence estimator.

### 5.3. Modulation Selection Algorithm

Now that the probability selector algorithm has found the best probabilities for transmitting ACKs and NACKs that accurately estimate the number of STAs receiving messages correctly/incorrectly, we can proceed with the second algorithm that selects the best MCS. As Section 4 mentions, the AP seeks to choose the highest MCS possible while keeping the percentage of STAs that do not receive messages correctly within the desired range.

Figures 9 (for 100 STAs) and 10 (for 1000 STAs) show the simultaneous operation of the probability selector algorithm (left column) and the modulation selector algorithm (right column) for a network area of radius 100 m (first row), 200 m (second row), and 300 m (third row). In each row, the left plot shows the behavior of the probability selector algorithm for transmitting ACKs and NACKs after every  $2f = 2k$  messages (1000 odd-numbered messages used for NACK and 1000 even-numbered messages used for ACK). As mentioned in Section 3, the desired percentage of silence events that guarantees low error estimation is set between  $[15, 45]\%$ . Once the percentage of silences of both  $n_{NACK}$  and  $n_{ACK}$  falls within this range, the AP moves to the second algorithm and decides to increase, decrease, or keep the current MCS depending on whether the percentage of STAs receiving messages incorrectly is below, above, or within the desired range (set to  $[10, 20]\%$  of STAs in all experiments), respectively. In all plots, the initial MCS is MCS(5). However, any other initial MCS strategy should work (i.e., the lowest or the middle MCS.).

As we can see in the figures, the probability selector algorithm takes about 4000–8000 messages to reliably estimate the number of STAs receiving messages correctly/incorrectly every time the second algorithm changes the MCS. Similarly, as the scenario increases the network radius, more MCS changes are needed as more STAs become further away from the AP, thus requiring a lower MCS to receive messages correctly. As for the number of STAs, we can see that it takes fewer messages for the probability selector algorithm to converge for 1000 STAs (see Figure 10) compared to 100 STAs (see Figure 9).

While the 20,000–30,000 messages transmitted for the two combined algorithms to converge to the desired behavior could appear too many, in practice, it will take only a few

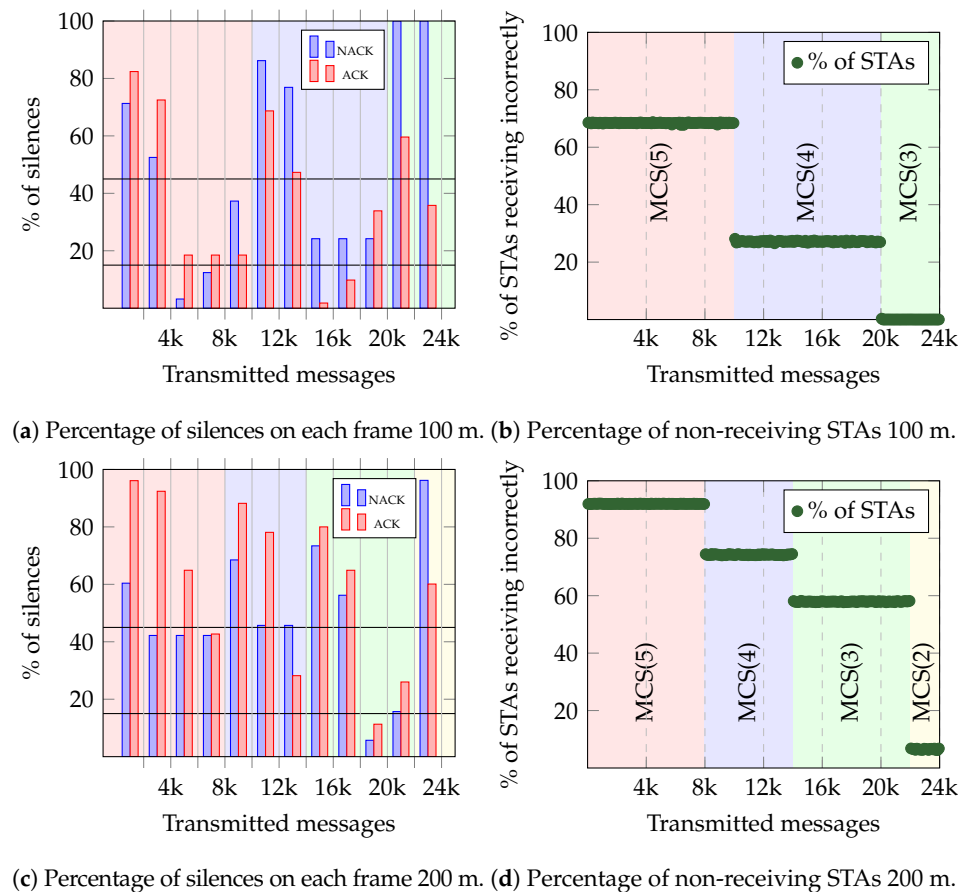


seconds, depending on the number of messages per second being transmitted by the AP. We consider the H.265 codec as an example of the AP sending video/audio content to STAs. Table 5 shows the number of seconds this particular codec takes to reach 24k messages for some of its resolution modes.

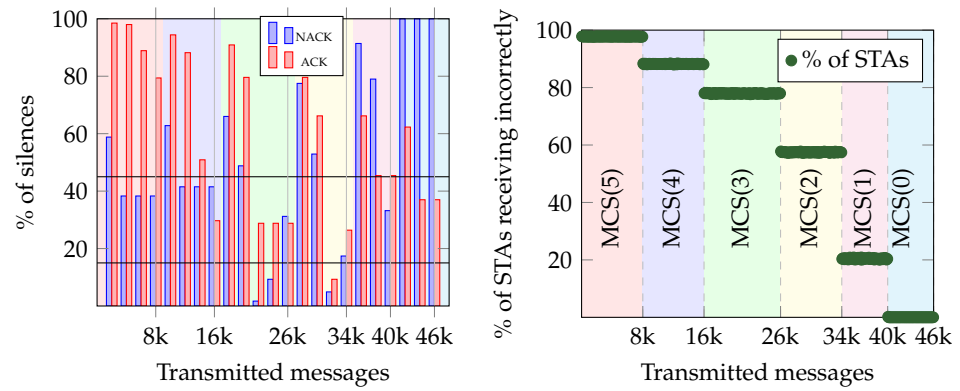
Overall, Figures 9 and 10 show that the proposed method can find the best MCS that maintains the number of STAs receiving messages incorrectly within the desired limits independently of the number of STAs and the size of the scenario. Similarly, as Table 5 illustrates, the proposed method takes only a few seconds to find the best possible MCS.

**Table 5.** Time [s] for H.265 codec to transmit 24,000 messages for various levels and 188 Bytes packet.

Level	Packets per Second	~Time to Reach 24,000 Messages [s]
4	2659	9
4.1	5984	4
5	12,632	1.9
5.1	26,595	0.9
5.2	53,191	0.45

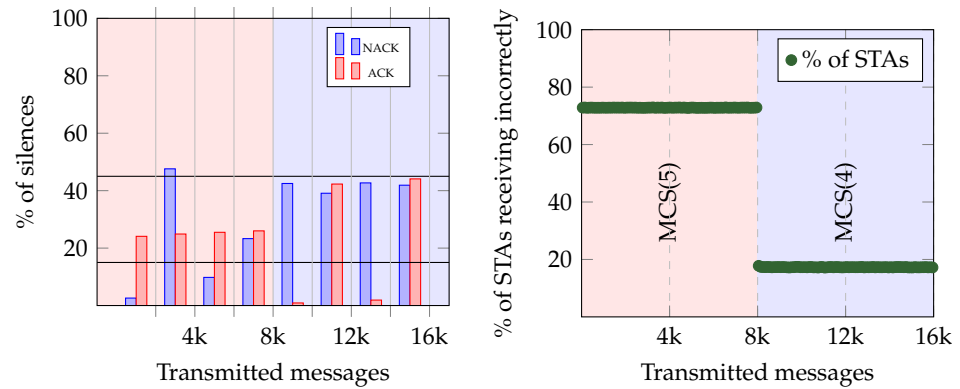


**Figure 9.** Cont.

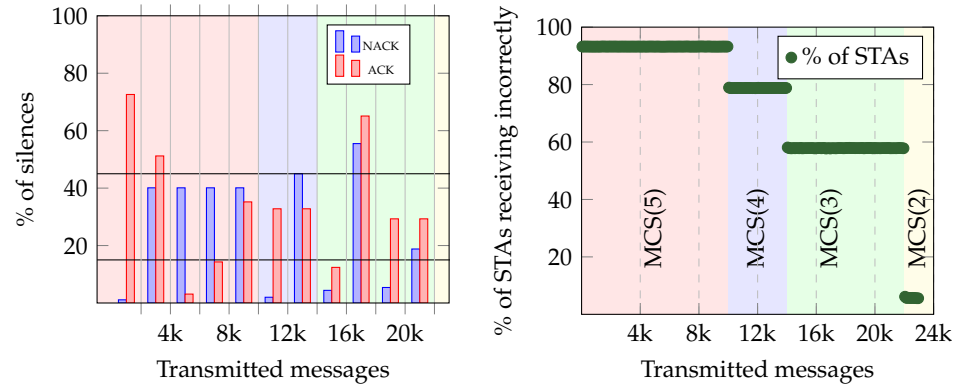


(e) Percentage of silences on each frame 300 m. (f) Percentage of non-receiving STAs 300 m.

Figure 9. Rate Adaption for 100 STAs.

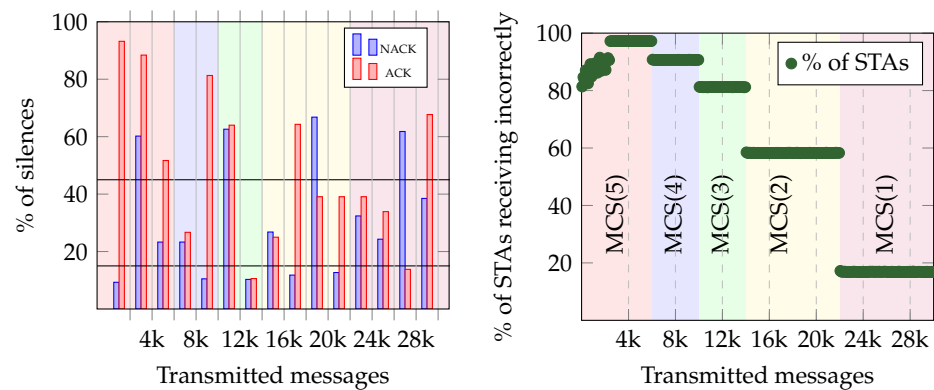


(a) Percentage of silences on each frame 100 m. (b) Percentage of non-receiving STAs 100 m.



(c) Percentage of silences on each frame 200 m. (d) Percentage of non-receiving STAs 200 m.

Figure 10. Cont.



(e) Percentage of silences on each frame 300 m. (f) Percentage of non-receiving STAs 300 m.

Figure 10. Rate Adaption for 1000 STAs.

### 6. Discussion

When evaluating the performance of the three estimators for Algorithm 1, the initial expectation was that all three would yield similar estimations, thereby enhancing the system’s robustness. However, it became evident that the silence estimator outperformed the others for a finite number of messages, providing superior results. The explanation for this outcome is the frequent occurrence of packet capture during the experiments. Packet capture enables a receiver to retrieve a single packet (typically the one with the highest signal-to-noise ratio (SNR) from a collision of packets. This mechanism transforms a collision event into a successful reception event, introducing discrepancies or errors in the collision and correct feedback estimators. Interestingly, the silence estimator remained unaffected by this issue and consistently delivered accurate estimations of the number of STAs receiving messages, demonstrating remarkable accuracy (within a 1% margin of the actual number of STAs).

In the evaluation of Algorithm 2 in Section 5.3, focusing on the adjustment of MCS, the experiments were conducted using a high value for MCS(i). However, alternative strategies, such as commencing with a middle-valued MCS(i), can be explored. Furthermore, instead of the incremental or decremental adjustment of the current MCS(i) by one step per frame, Algorithm 2 could evaluate the percentage of STAs experiencing message reception errors to execute larger jumps across multiple MCS indexes after each frame. Irrespective of the initial MCS index chosen, the results illustrate that the ultimate MCS outcome remains consistent—solely the time taken to achieve convergence varies. Yet, commencing with a higher MCS offers an advantage by expediting the transmission of broadcast messages. This facilitates a quicker estimation by the AP of the number of STAs that receive messages inaccurately, in contrast to initiating the process with a lower MCS. Due to limitations in space, we only presented the results of experiments using a NACK range set between 10% and 20%. Nevertheless, experiments were conducted with varying NACK ranges, revealing that a reduction in the NACK range (e.g., 5% to 10%) led to the AP selecting a lower MCS index. Conversely, a higher NACK range (e.g., 20% to 40%) resulted in a higher MCS index compared to the findings detailed in Section 5.3.

While the present analysis exclusively focuses on static scenarios, it is worth noting that, even in scenarios where STAs are mobile, the time it takes for an STA to traverse a single disk associated with a specific MCS(i) (under the scenario depicted in Figure 5) ranges between 15 to 50 s. Considering that the proposed method achieves convergence to the optimal MCS within 1 to 8 s, the convergence will most likely occur before an STA transitions to a different disk area due to movement. It’s also essential to emphasize that the 802.11bc standard mainly envisions static scenarios. This assumption aligns with the understanding that engaging in activities like viewing multimedia content while walking should be discouraged.

Nonetheless, the proposed method presents its limitations and tradeoffs. On one hand, the time it takes to converge is directly proportional to the rate of broadcast messages. In some high-quality video stream applications, it will take the method a very short time to converge (less than one second); however, if the content is being sent at a slow pace, convergence times to find the optimum MCS will take much longer. Also, this work assumes that only one AP operates in the venue. Having one or more APs interfering with the AP sending broadcast content might cause a collision of broadcast and feedback messages, adding errors in the estimation process. In the future, we plan on modifying the base method to make it resilient to this constraint.

Finally, we evaluated the performance of the method in a simulation environment. Thus, future work should focus on implementing the presented strategy to validate its operation in real-world conditions. However, we anticipate several discrepancies impacting the proposed method's performance in a real-world setting. On the one hand, SNR variations are similar from one instant to the next in the simulations. However, this may not happen in real-world conditions where, due to channel conditions, SNR might experience significant changes in a very short time. On the other hand, the simulation assumes symmetric channels. This condition will not always happen in real-world situations, thus introducing errors in the number of events the AP counts for each of the three estimators. Then, more experiments must be conducted in heterogeneous networks where devices have different hardware features, such as antennas or transceiver capabilities.

## 7. Conclusions

This work proposes a novel method to select the best possible MCS in a WiFi broadcast scenario where hundreds or thousands of STAs receive media content from a single AP. The proposed method is composed of two sequential algorithms. The first algorithm allows the AP to find the best probability for the STAs to send ACKs or NACKs to accurately estimate the number of STAs receiving broadcast content correctly/incorrectly. Then, a second algorithm selects the best possible MCS that maintains the number of STAs receiving broadcast content within the desired levels. We implemented both algorithms in ns-3 and found they choose the best possible MCS independently of the number of STAs and the scenario size in only a few seconds. The proposed method is a robust, lightweight, and scalable solution for the widespread adoption of WiFi broadcast, such as in the IEEE 802.11bc initiative that was recently released.

In future work, we plan to study mobility issues, ways to speed up the convergence time, and ways to allow for the protocol to operate in the presence of uplink unicast traffic and cases where other APs interfere in the vicinity. We also plan to study the behavior of the proposed method in heterogeneous scenarios. Finally, we plan to implement and test the proposed method in real conditions to learn more about potential issues missed in the simulation phase.

**Author Contributions:** Conceptualization, J.G. and J.J.C.-E.; methodology, J.G.; validation, D.R.-T., J.G. and J.J.C.-E.; formal analysis, J.J.C.-E. and L.O.-B.; investigation, L.O.-B.; resources, J.G.; software, D.R.-T.; supervision, J.G.; visualization, J.J.C.-E.; writing-original draft, J.G. and J.J.C.-E.; writing-review and editing, L.O.-B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded in part by the Consejo Nacional de Humanidades Ciencias y Tecnologías with the Grant Program "Becas Conacyt Nacionales" and with the Program "Paradigmas y Controversias de la Ciencia 2022" under grant number 0320403. Also, this work was supported in part by the Spanish Ministry of Science, Education and Universities, the European Regional Development Fund, and the State Research Agency, Grant No. PID2021-123627OB-C52. The APC was funded by the Consejo Nacional de Humanidades Ciencias y Tecnologías with the Program "Paradigmas y Controversias de la Ciencia 2022" under grant number 0320403.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

ACK	Acknowledgement
AP	Access Point
MCS	Modulation and Coding Scheme
NACK	Negative Acknowledgement
SNR	Signal to Noise Ratio
STA	Station

## References

1. Ansley, C.; Morioka, H.; Emmelmann, M.; McCann, S.; Wang, X.; Patil, A. *802.11bc Functional Requirements Document*; IEEE: Piscataway, NJ, USA, 2019.
2. Kuri, J.; Kasera, S. Reliable multicast in multi-access wireless LANs. In Proceedings of the IEEE INFOCOM '99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies: The Future Is Now, New York, NY, USA, 21–25 March 1999; Volume 2, pp. 760–767. [\[CrossRef\]](#)
3. Adamson, R.; Macker, J. Quantitative prediction of NACK-oriented reliable multicast (norm) feedback. In Proceedings of the MILCOM 2002, Anaheim, CA, USA, 7–10 October 2002; Volume 2, pp. 964–969. [\[CrossRef\]](#)
4. Gupta, S.; Shankar, V.; Lalwani, S. Reliable multicast MAC protocol for wireless LANs. In Proceedings of the IEEE International Conference on Communications, 2003. ICC '03., Anchorage, AK, USA, 11–15 May 2003; Volume 1, pp. 93–97. [\[CrossRef\]](#)
5. Chen, A.; Lee, D.; Chandrasekaran, G.; Sinha, P. HIMAC: High Throughput MAC Layer Multicasting in Wireless Networks. In Proceedings of the 2006 IEEE International Conference on Mobile Ad Hoc and Sensor Systems, Vancouver, BC, Canada, 9–12 October 2006; pp. 41–50. [\[CrossRef\]](#)
6. Peng, J. A New ARQ Scheme for Reliable Broadcasting in Wireless LANs. *IEEE Commun. Lett.* **2008**, *12*, 146–148. [\[CrossRef\]](#)
7. Tang, K.; Gerla, M. MAC reliable broadcast in ad hoc networks. In Proceedings of the 2001 MILCOM Proceedings Communications for Network-Centric Operations: Creating the Information Force, McLean, VA, USA, 28–31 October 2001; Volume 2, pp. 1008–1013. [\[CrossRef\]](#)
8. Sun, M.t.; Huang, L.; Wang, S.; Arora, A.; Lai, T.H. Reliable MAC layer multicast in IEEE 802.11 wireless networks. *Wirel. Commun. Mob. Comput.* **2003**, *3*, 439–453. [\[CrossRef\]](#)
9. Choi, W.Y. Connectivity-Based Reliable Multicast MAC Protocol for IEEE 802.11 Wireless LANs. *EURASIP J. Wirel. Commun. Netw.* **2009**, *2009*, 968408. [\[CrossRef\]](#)
10. *IEEE Std 802.11v-2011 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11w-2009, IEEE Std 802.11n-2009, IEEE Std 802.11p-2010, and IEEE Std 802.11z-2010)*; IEEE Standard for Information Technology—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: Piscataway, NJ, USA, 2011; pp. 1–433. [\[CrossRef\]](#)
11. Moussaoui, M.; Bertin, E.; Crespi, N. 5G shortcomings and Beyond-5G/6G requirements. In Proceedings of the 2022 1st International Conference on 6G Networking (6GNet), Paris, France, 6–8 July 2022; pp. 1–8. [\[CrossRef\]](#)
12. *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*; 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: Piscataway, NJ, USA, 2021; pp. 1–4379. [\[CrossRef\]](#)
13. *IEEE Std 802.11aa-2012 (Amendment to IEEE Std 802.11-2012 as amended by IEEE Std 802.11ae-2012)*; 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: Piscataway, NJ, USA, 2012; pp. 1–162. [\[CrossRef\]](#)
14. Nunome, T.; Mizutani, K. An Adaptive Modulation Method with IEEE 802.11aa GCR Block Ack for QoE Enhancement of Audio-Video Reliable Groupcast. In Proceedings of the 2019 2nd World Symposium on Communication Engineering (WSCE), Nagoya, Japan, 20–23 December 2019; pp. 42–47. [\[CrossRef\]](#)
15. Kanda, T.; Koda, Y.; Yamamoto, K.; Nishio, T. ACK-Less Rate Adaptation for IEEE 802.11bc Enhanced Broadcast Services Using Sim-to-Real Deep Reinforcement Learning. In Proceedings of the 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2022; pp. 139–143. [\[CrossRef\]](#)
16. Kanda, T.; Koda, Y.; Kihira, Y.; Yamamoto, K.; Nishio, T. ACK-Less Rate Adaptation Using Distributional Reinforcement Learning for Reliable IEEE 802.11bc Broadcast WLANs. *IEEE Access* **2022**, *10*, 58858–58868. [\[CrossRef\]](#)
17. Gupta, V.; Gutterman, C.; Bejerano, Y.; Zussman, G. Experimental evaluation of large scale WiFi multicast rate control. In Proceedings of the IEEE INFOCOM 2016—The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9. [\[CrossRef\]](#)
18. Bejerano, Y.; Gupta, V.; Gutterman, C.; Zussman, G. AMuSe: Adaptive Multicast Services to Very Large Groups—Project Overview. In Proceedings of the 2016 25th International Conference on Computer Communication and Networks (ICCCN), Waikoloa, HI, USA, 1–4 August 2016; pp. 1–9. [\[CrossRef\]](#)

19. Gupta, V.; Norwitz, R.; Petridis, S.; Gutterman, C.; Zussman, G.; Bejerano, Y. AMuSe: Large-scale WiFi video distribution—Experimentation on the ORBIT testbed. In Proceedings of the 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), San Francisco, CA, USA, 10–14 April 2016; pp. 854–855. [[CrossRef](#)]
20. Geithner, T.; Sivrikaya, F. Adaptive reliable multicast in 802.11 networks. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, Spain, 15–18 April 2018; pp. 1–6. [[CrossRef](#)]
21. Hernández, A.R.; Orozco-Barbosa, L.; Boujnoui, A.; Camacho-Escoto, J.J.; Gomez, J. WBAP: A Wireless Broadcast Access Protocol for the IEEE802.11bc Enhanced Broadcast Service. In Proceedings of the IEEE Latin-American Conference on Communications, LATINCOM 2022, Rio de Janeiro, Brazil, 30 November–2 December 2022; pp. 1–6. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.