Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7922273, 18 pages
https://doi.org/10.1155/2022/7922273

Research Article

Intelli-GATS: Dynamic Selection of the Wi-Fi Multicast Transmission Policy Using Interpretable-AI

Blas Gómez,[1] **Estefanía Coronado**,[1,2] **José Villalón**,[1] **and Antonio Garrido**[1]

[1] *High-Performance Networks and Architectures, Universidad de Castilla-La Mancha, Albacete, Spain*
[2] *i2CAT Foundation, Barcelona, Spain*

Correspondence should be addressed to Blas Gómez; blas.gomez@uclm.es

Received 18 July 2022; Revised 19 October 2022; Accepted 9 November 2022; Published 30 November 2022

Academic Editor: Suhua Tang

COVID-19 has changed the way we use networks, as multimedia content now represents an even more significant portion of the traffic due to the rise in remote education and telecommuting. In this context, in which Wi-Fi is the predominant radio access technology (RAT), multicast transmissions have become a way to reduce overhead in the network when many users access the same content. However, Wi-Fi lacks a versatile multicast transmission method for ensuring efficiency, scalability, and reliability. Although the IEEE 802.11aa amendment defines different multicast operation modes, these perform well only in particular situations and do not adapt to different channel conditions. Moreover, methods for dynamically adapting them to the situation do not exist. In view of these shortcomings, artificial intelligence (AI) and machine learning (ML) have emerged as solutions to automating network management. However, the most accurate models usually operate as black boxes, triggering mistrust among human experts. Accordingly, research efforts have moved towards using Interpretable-AI models that humans can easily track. Thus, this work presents an Interpretable-AI solution designed to dynamically select the best multicast operation mode to improve the scalability and efficiency of this kind of transmission. The evaluation shows that our approach outperforms the standard by up to 38%.

1. Introduction

Over the last few years, the industry has witnessed a surge in demands for real-time multimedia traffic such as video conferencing, VoIP, or IPTV. The COVID-19 pandemic has only made this increase more pronounced, especially in schools' and universities' wireless communication infrastructure, which now offers different teaching options from in-person to remote learning. Some hybrid models include mirror lectures, where students are distributed in different classrooms despite the lecturer being present in only one of them. This new way of teaching makes it possible to respect the social distance and room capacity limitations while still preserving access to educational material and equipment. Moreover, many students have had to quarantine in their halls of residence, leading to many stations connecting to the same stream. Multicast transmissions play a fundamental role in this setting as they prevent multiple identical unicast streams from saturating the network.

The most popular access network in the scenarios mentioned above is Wi-Fi. However, the contentious nature of this radio access technology (RAT) results in challenging conditions for this kind of transmission, especially when it comes to video or audio due to their stringent performance requirements. For instance, packet loss, collisions, and delay have a direct impact on the quality of the transmission. Moreover, multicast transmissions add an extra layer of complexity. In unicast transmissions, rate adaption algorithms such as Minstrel [1] determine the modulation and coding scheme (MCS) that maximizes throughput and minimizes the loss of frames. However, to avoid a severe traffic overhead due to the feedback implosion effect, multicast transmissions eliminate the use of acknowledgments, giving rise to two issues: (i) the retransmission of the frames lost due to collisions or interference is not possible as there is no way to know whether a frame was received correctly and (ii) neither is it possible to adapt the datarate to the channel conditions using the rate adaption algorithms mentioned above. This has traditionally forced these transmissions to use the most robust MCS available to guarantee the delivery of the frames regardless of the position and channel conditions of each station. Thus, multicast frames keep the channel busy for longer than unicast frames, and there is no guarantee that the frames are being delivered.

In view of this, the IEEE 802.11aa amendment [2] introduced the Group Addressed Transmission Service (GATS) to overcome these problems. The primary purpose of GATS is to ensure robust multicast communications in wireless local area networks (WLANs) while maintaining backwards compatibility with commercial devices. With this in mind, the amendment introduces a set of multicast transmission policies to improve reliability, preserving the correct operation of these services. Each one of these policies accommodates better to different network conditions. However, the standard only provides recommendations to help network administrators choose between them, but there is no dynamic method to adapt the GATS policy used to the network conditions. Within this framework, it seems evident that dynamically adapting the GATS policy to the channel conditions would improve the reliability and the performance of multicast transmissions, which is especially critical for video and voice transmissions.

As software-defined networks (SDN) and multiaccess edge computing (MEC) become widely available in last-generation networks, there is a general trend in the industry towards using artificial intelligence (AI) to exploit the new range of possibilities for the automation of network management offered by these technologies. MEC systems leverage SDN, network slicing, and virtualization to enable cloud capabilities closer to end users and offer low latency and high bandwidth for deploying innovative third-party applications. MEC provides radio network information APIs [3], which allow these applications to obtain contextual information from the radio access network. AI can take advantage of this information to automate network management and control. In particular, machine learning (ML) is used to predict the behavior of wireless networks because of its ability to approximate complex network optimization functions.

However, the ML models that obtain the most accurate approximations, such as neural networks, usually operate as black boxes. This prevents human experts from understanding the output, making troubleshooting the network difficult or even preventing the use of zero-touch management approaches. Moreover, the lack of transparency in the decisions of the ML models may trigger mistrust in them. Consequently, many research efforts have moved towards the use of Interpretable-AI techniques. Interpretable-AI predictions can be easily understood and tracked by humans. Moreover, interpretable models stand out for their low complexity, making them perfect for running on resource-constrained devices such as access points (APs) or other devices at the edge of the network.

The present work brings the following contributions:

(i) We present Intelli-GATS, an Interpretable-AI approach for the dynamic selection of the GATS policy in Wi-Fi networks that best suits the channel conditions. To the best of our knowledge, no prior approach for the dynamic selection of the GATS policies has been proposed. Two ML models are used and compared. The first one is k-nearest neighbors (kNN), which is chosen because of its good behavior when there are many data points but few dimensions. Moreover, it is inherently interpretable, requires zero training, and does not assume any underlying statistical model [4–6]. The second one is random forests (RF), an ensemble method based on decision trees that pushes the limits of interpretability. We choose this model as an eager-learning counterpart to kNN. It has been widely used for throughput prediction in the literature [7–9]

(ii) The GATS policies are implemented on the open-source network simulator NS-3 [10]. NS3-AI [11] is used to emulate SDN architecture and to connect the simulator with the most widely used ML libraries. This setting is used to train and test the performance of both models in adapting the GATS policy to the network conditions. We have released the implementation of the GATS policies in NS-3 and the dataset used for training

(iii) We provide an extensive performance evaluation of the different GATS policies and the two proposed ML models and show how adaptive approaches overperform the standard policies

The rest of the paper is organized as follows. Section 2 provides the technical background, and Section 3 gives an overview of the related work. Section 4 provides the details of our contributions, and Section 5 presents the performance evaluation. Finally, Section 6 contains our conclusions.

2. The IEEE 802.11aa Amendment

The IEEE 802.11aa amendment [2] addresses the limitations of multimedia streaming services. For that, the main additions of the IEEE 802.11aa amendment to the standard are Stream Classification Service (SCS), overlapping basic service set (OBSS), and GATS. The SCS is aimed at differentiating between separate streams within the same access category and allowing their graceful degradation in case of bandwidth shortage. The OBSS is aimed at providing a decentralized mechanism for neighboring APs to exchange information about their traffic load. This allows for more efficient channel selection and even for APs to cooperate.

Finally, another aim of the IEEE 802.11aa amendment is to improve the reliability and efficiency of multicast traffic delivery while ensuring the performance of other streams. For this purpose, the amendment introduces GATS, which is a new mechanism intended to (i) overcome the poor reliability of multicast services and (ii) increase the efficiency of the transmissions affected by the low datarates. GATS incorporates two different policies named directed multicast service (DMS) and group cast with retries (GCR). The aim of this work is to improve the multicast transmissions through the dynamic selection of the different GATS policies provided by the standard in execution time. Consequently, in the following subsection, the GATS policies are explained in detail.

2.1. Group Addressed Transmission Services. The IEEE 802.11v amendment [12] allows stations to exchange information about the network's radio status or topology. It enables for assisted roaming, allowing the network to send information to the stations about possible better APs to associate with. Moreover, the IEEE 802.11v amendment introduces DMS to help devices save battery by reducing the need for stations to wake up to receive multicast traffic. The IEEE 802.11aa applies the capabilities of DMS to the improvement of multimedia streaming services in multicast. It transforms multicast frames into individual unicast frames; i.e., for a multicast group with n multicast receivers (MRs), DMS creates n copies of the multicast frame and addresses each copy as a unicast frame to one of the MRs, as shown in Figure 1(a). Thus, as with any other unicast frame, frames in DMS are retransmitted until the source receives an acknowledgment (ACK) or the retransmission counter reaches its limit. Doing this provides multicast streams with the same reliability as that expected from unicast streams. However, bandwidth requirements are considerably higher than when using the Legacy multicast policy, and its scalability is limited since the resources needed grow linearly with the number of MRs. It is worth mentioning that unicast transmissions can use rate adaption algorithms such as Minstrel. Thus, transforming multicast into unicast allows the APs to send at much faster datarates, which eases channel occupation and makes DMS perfect for small groups.

Conversely, GCR is composed of three policies aimed at improving the reliability of multicast streams: no retry/no ACK, unsolicited retries (UR), and block ACK (BACK). The first refers to the legacy multicast mechanism used by default in the original IEEE 802.11 standard. Thus, we will hereafter refer to this policy as Legacy. When using this policy, frames are neither acknowledged nor retransmitted; i.e., the transmitter presumes that the MRs will receive the frame. However, it has no feedback on whether this is true or not, as shown in Figure 1(b). It is essential to consider that since more robust MCSs have a wider range and Legacy has no feedback on the reliability of the transmissions, frames are generally transmitted at the lowest rate, i.e., the basic rate. In this way, the reliability of multicast transmissions is maximized. However, some vendors allow the configuration of the rate used in Legacy transmissions.

UR reduces the probability of a frame not reaching its destination correctly by retransmitting it. Since there is no feedback on the reliability of the transmissions, there is no way of knowing whether a frame was received correctly. Thus, UR retransmits all frames a configurable number of times, as shown in Figure 1(c). The standard does not specify the number of times a frame can be retransmitted. Consequently, this policy is less reliable than DMS, but it solves the scalability problem since it is independent of the number of MRs. However, many frames will be unnecessarily retransmitted, which results in unwanted overhead.

The block ACK mechanism extends the capabilities introduced in the IEEE 802.11 standard for unicast frames to cope with the particular requirements of multicast transmissions. In unicast transmissions, the sender and the receiver agree on transmitting a certain number of frames before requesting a single ACK that confirms all of them. On the other hand, in multicast transmissions, this agreement is established with all the members in the multicast group. After that, the transmitter delivers a burst with the agreed number of frames addressed to the multicast group. Then, the transmitter requests the BACK from each receiver, as shown in Figure 1(d). This allows the sender to adapt the transmission rate to the channel conditions perceived by each receiver and to know which frames must be retransmitted. This introduces a considerable computing overhead in the process of achieving robustness. The number of BACK requests and BACKs increases linearly with the number of transmitters and receivers and requires more processing power to compute CRC and bitmaps. Consequently, its implementation on devices on the market is limited due to performance constraints [13]. Therefore, this work does not consider this policy.

In brief, DMS is particularly suitable for small-sized multicast groups. It provides a high level of reliability, but it suffers from low scalability, and the delay increases proportionally to the number of group members. Legacy exhibits low reliability, although it shows no scalability problems, and UR falls between the two with better reliability than Legacy and with the same scalability but at the cost of more overhead. DMS offers the best reliability, but its overhead increases as the multicast group grows; i.e., it provides low scalability. This is summarized in Table 1. It is worth noting that even though the IEEE 802.11aa amendment provides these mechanisms, and it is out of its scope to provide a selection mechanism. It only provides a set of guidelines to be used by the system administrator to configure the service. This work is focused on the improvement of the multicast transmissions through the adaptive selection of the different GATS policies provided by the standard. To the best of our knowledge, no such methods have appeared in the literature either. Despite the importance of multicast transmissions, the IEEE 802.11aa amendment is usually not implemented in retail devices [13]. On the opposite side, the IEEE 802.11v amendment is usually implemented by vendors as it is used for battery-saving purposes. Consequently, off-the-shelf devices only implement Legacy and DMS. Thus, these two are considered in this work along with GCR-UR, as its simplicity makes the implementation straightforward for vendors.

3. Related Work

The GATS policies introduced in the IEEE 802.11aa amendment have been deeply studied in the literature, although, to the best of our knowledge, until now, no previous work has presented a dynamic GATS policy selection approach. However, we can still identify relevant works that evaluate the performance of the GATS policies provided by the IEEE 802.11aa amendment.

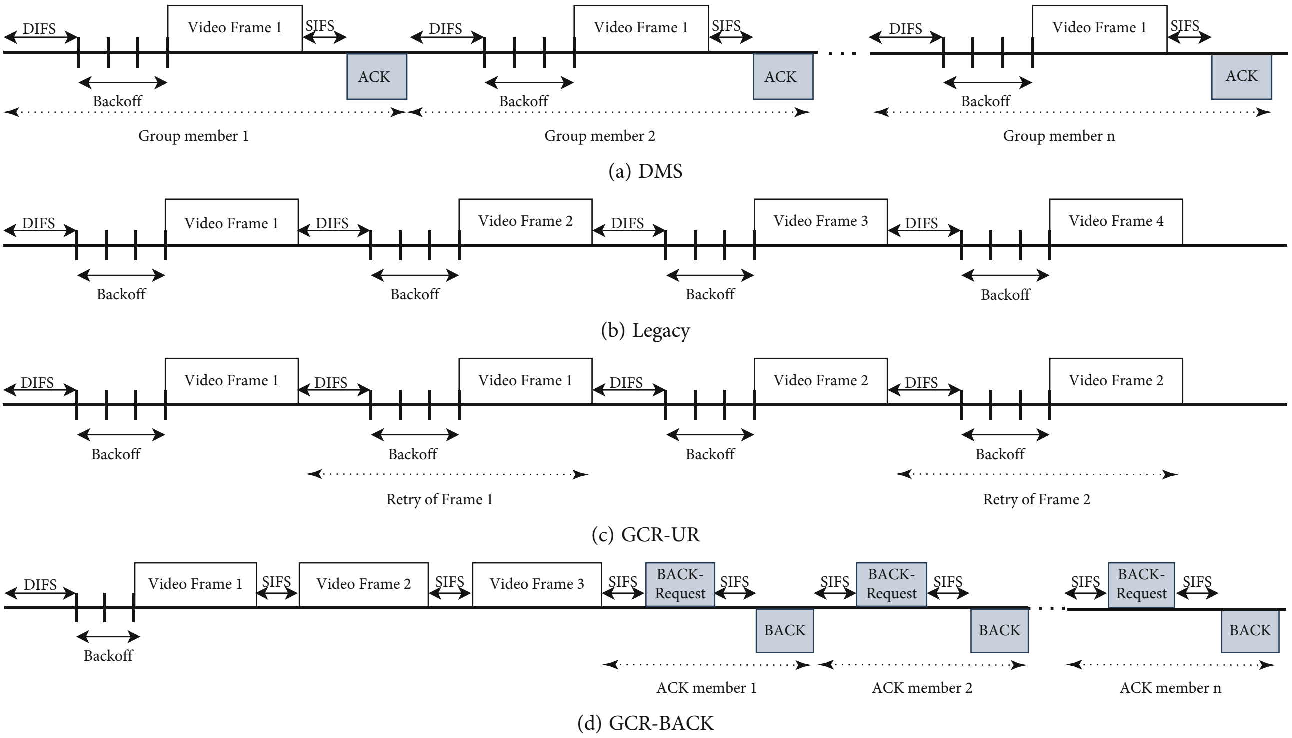


Figure 1: Group Addressed Transmission Services.

Table 1: Performance classification of the GATS.

Feature	DMS	Legacy	UR
Scalability	Low	High	High
Delay	Variable	Low	Moderate
Efficiency	Variable	Low	Moderate
Reliability	High	Low	Moderate

The authors in [14–16] use simulations to evaluate the GATS policies. On top of the evaluation, the authors in [14] also provide an offline GATS selection algorithm from which they extract usage recommendations. However, their method is not dynamic and is only used offline. They do not provide any method for run-time GATS policy selection. The authors in [15] also provide similar findings using simulations. In [16], the authors provide a mathematical model which they use to evaluate the GATS policies. The authors in [17] provide an experimental evaluation that provides insights into the trade-offs of each policy. Another experimental evaluation focused on QoE is presented in [18]. All these evaluations seem to converge on one conclusion: there is not a single GATS policy that is good for everything, and its selection should be based on network conditions. However, the standard only provides the mechanisms. Providing methods to choose one is out of the scope of the IEEE 802.11aa amendment. It is at the discretion of the network manager to choose the most appropriate one. However, there is no possible one-off choice, as the channel conditions change over time. Yet, to the best of our knowledge, no algorithms for run-time GATS policy selection have been proposed.

It is worth noting that the physical layers used in these studies (IEEE802.11a/g/n) are already outdated. More recent physical layers can provide higher throughput, and consequently, the behavior of the different GATS policies is very different. This is shown by the authors in [19], who provide an evaluation of the GATS policies using the IEEE 802.11ac very high throughput (VHT) physical layer. Increasing the available bandwidth delays the moment when the policies that introduce more overhead start to struggle. However, none of the above-mentioned evaluations use rate adaption algorithms. Note that this is one of the main advantages of DMS, which, thanks to the feedback provided to the ACKs, can use these algorithms to transmit at the most appropriate rate for channel conditions. Even if this work does not aim to be an evaluation of the standard but an ML-aided dynamic selection method instead, we provide data on the behavior of the GATS policies using an IEEE802.11ac physical layer and adaptive MCS.

All the evaluations assessed in this section reach the same conclusion; each GATS policy is better for a different scenario. The summary of their contributions can be found in Table 2. However, to this day, there are no dynamic approaches to choosing the best one in run-time. Interpretable-AI offers a way to automate network management while making transparent decisions that avoid the misgivings of stakeholders. Moreover, COVID-19 has brought a wave of multimedia transmissions to wireless networks and a new era for multicast transmissions in Wi-Fi that can be used to transmit part of this content more efficiently. Thus, the main contributions of this work are as follows:

Table 2: Related work contributions.

Work	Goal	MCS adaption	VHT PHY	Offline GATS selection	Dynamic GATS selection
[14]	Evaluation and off-line GATS selection	✗	✗	✓	✗
[15]	Evaluation	✗	✗	✗	✗
[16]	Evaluation	✗	✗	✗	✗
[17]	Evaluation	✗	✗	✗	✗
[18]	Evaluation	✗	✗	✗	✗
[19]	Evaluation	✗	✓	✗	✗
Intelli-GATS	Dynamic GATS selection	✓	✓	✓	✓

(i) We introduce an Interpretable-AI method for the dynamic selection of the GATS policies defined in the IEEE 802.11aa amendment. To the best of our knowledge, this is the first dynamic GATS selection approach. The behavior of two different ML models is assessed as an inherently interpretable one, such as K-Nearest Neighbors, and an ensemble model, such as random forests which is on the limits of interpretability

(ii) We provide the implementation of the GATS policies evaluated on the open-source network simulator NS-3 and emulate an SDN architecture using NS3-AI. The implementation of the GATS policies is made publicly available. We also provide the dataset use for training

(iii) An extensive performance evaluation of the dynamic approach using both ML models is provided using the standard as a benchmark

4. Intelli-GATS: Interpretable-AI for Dynamic GATS Policy Selection

As explained in the previous section, selecting the optimal GATS policy is heavily dependent on the channel conditions and the size of the multicast group. However, the IEEE 802.11aa standard does not define any method for dynamically adapting the GATS policy to network conditions. ML models can be useful in this situation by learning from experience to approximate complex network functions. Thus, in this work, we capitalize on these valuable attributes of ML in order to build a training dataset that represents the performance of the selected GATS policies in a wide range of scenarios and implement an adaptive solution that leverages Interpretable-AI to select the GATS policy that maximizes the average goodput of multicast transmissions.

4.1. Dataset Construction. The first step in the design process is the generation of a training dataset that will be used to train the models. The large quantity of data and scenarios needed makes network simulators a useful tool for generating such an amount of data, and in this work, we use NS-3 [10], version 3.33. The original simulator has been extended to implement the selected IEEE 802.11aa GATS policies, i.e., DMS, Legacy, and GCR-UR. The AP Wi-Fi MAC layer of NS-3 is modified to change the way that multicast frames are enqueued, replicating them as many times as retries are configured in the case of GCR-UR and setting the retry flag for the replicas. In the case of DMS, the frames are replicated as many times as members in the multicast group, and the destination MAC addresses are changed to address each one of them instead of the group. The remote station manager is modified to set the correct datarates in Legacy and GCR-UR frames and to use Minstrel in DMS frames. Moreover, the MAC layer is modified to keep the consistency of the sequence numbers of the frames. Details on the new flow are provided in Figure 2. This extension is publicly available (available at https://github.com/blasf1/ns-3-dev-git/tree/wifi_multicast_3.33).

The structure of the simulation is aimed at simulating an SDN architecture. The simulations consist of a multicast group with a configurable number of MRs and a set of unicast stations (STAs), which are both transmitters and receivers. The simulations also have an AP connected to a server that transmits both the multicast and unicast traffic and receives the uplink traffic generated by the STAs. The AP, the STAs, and the MRs are simulated in NS-3. The AP connects to another application that has been developed to simulate an SD-RAN controller, and NS3-AI [11] is used to connect the SD-RAN controller developed to the AP simulated in NS-3. NS3-AI uses shared memory to connect NS-3 to external apps and libraries in Python. This connection lets developers connect the NS-3 simulator to the most extended ML libraries implemented in this programming language. The AP regularly sends a set of key performance indicators (KPIs) to the script that simulates the SD-RAN controller in a configurable period, which in this case is set to 0.25 seconds. This interval is chosen to ensure that decisions are taken rapidly while not causing too much overhead over the network or on the CPU that needs to compute the predictions. The SD-RAN controller then communicates new settings or actions back to the AP. The scenario described is illustrated in Figure 3.

It is essential to provide the model with the most significant number of scenarios possible. The dataset presented in this work includes scenarios that combine different numbers of STAs, MRs, and datarates. In this way, the system is trained under different load levels. In addition, both the STAs and MRs are randomly distributed inside a disc-shaped area with a 40-meter radius, with the AP in the middle. This distance is the maximum that STAs can handle without disconnections. The random positioning of the STAs is intended to emulate a real network with devices

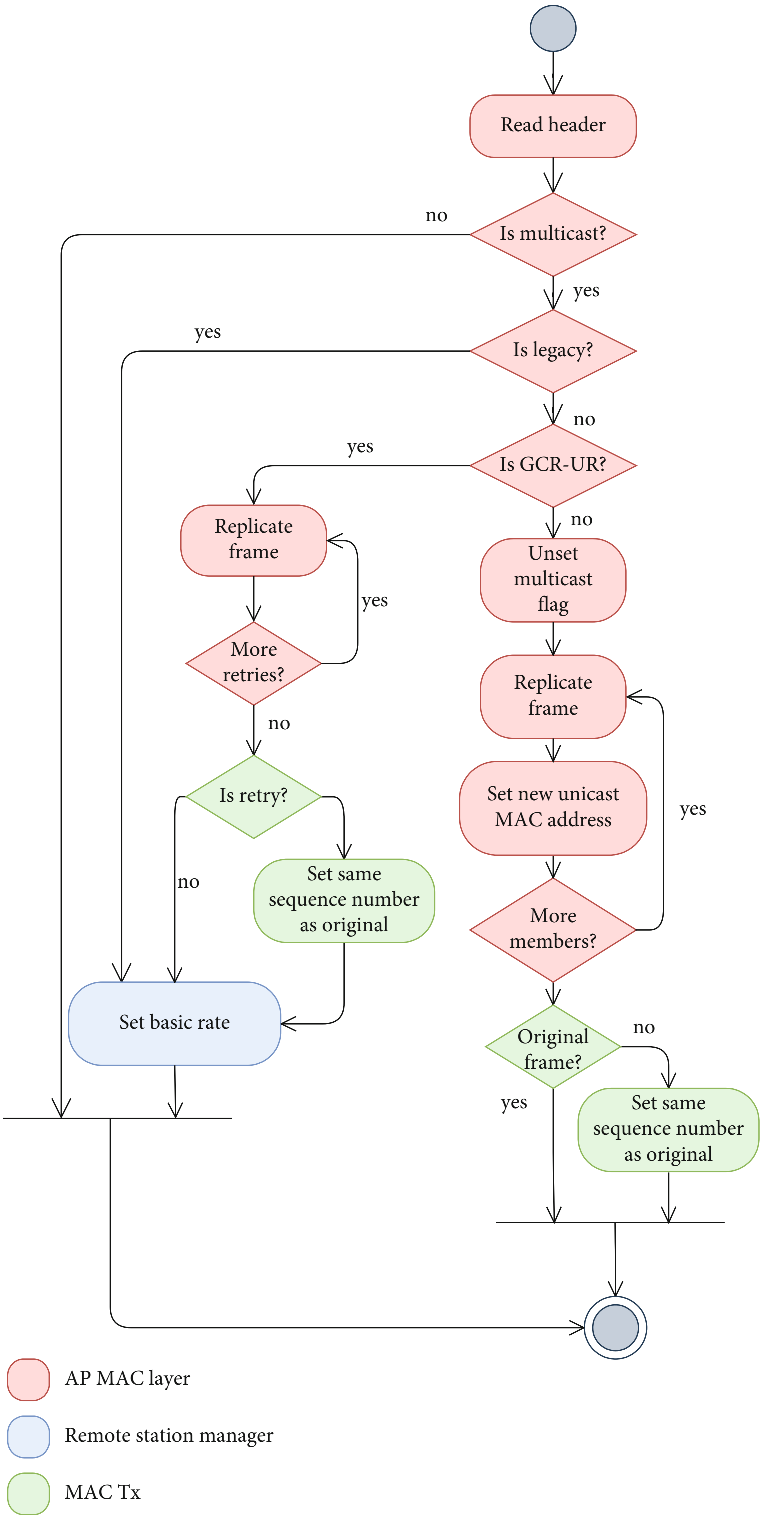


Figure 2: New activity flow of NS-3 for GATS.

experimenting with different received signal strength indicators (RSSIs). The simulations that make up the dataset can be regarded as two subsets, with each subset having a different datarate. In subset 1, each unicast STA transmits and receives 0.5 Mbps, while in subset 2, each one transmits and receives 1 Mbps. This is done to create equivalent load levels in both subsets but with a different number of STAs. This is relevant because more STAs cause more congestion even if they are injecting the same amount of traffic when combined. This is due to Wi-Fi's channel contention method, as more STAs contending to gain channel access make collisions more likely to happen. The multicast

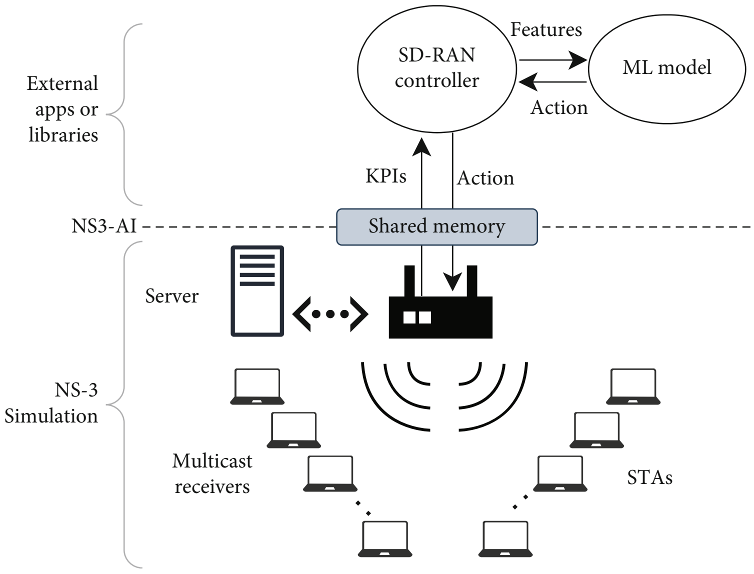


Figure 3: Network topology and simulation setting.

datarate is set to 1.5 Mbps to match the maximum datarate of group and video call applications such as Microsoft Teams [20]. In all the simulations, EDCA access categories are used. As the multicast group tries to simulate a group video call, it downloads traffic in the video access category (VI), and the STAs transmit and download best effort (BE) traffic to emulate all other types of traffic that usually occupy the network. The IEEE 802.11ac physical layer is used, and the Minstrel rate adaption algorithm adjusts the datarate of the unicast and DMS transmissions. Minstrel is used as one of the most used rate adaption algorithms in commercial devices.

The goal is to obtain different load levels by adding STAs step by step and also including different numbers of MRs. Different combinations of MRs and STAs are simulated as shown in Table 3. The set of levels to be tested in subset 1, i.e., the number of STAs in each simulation, is given by $3n : n \in [\![1, 9]\!]$. In the case of subset 2, it is given by $2n : n \in [\![1, 9]\!]$. These levels are established to maintain similar load levels in each step for both subsets while using a different number of STAs. Different multicast group sizes are included in the dataset for each load level, as this feature is decisive when choosing the GATS. The set of group sizes (the number of MRs) to be tested is given by $3n : n \in [\![1, 8]\!]$. Each of these combinations is repeated ten times with a different seed to include more variation in the dataset. In order to determine the load levels used, test simulations are run to identify when frames start to exceed the deadline and are consequently dropped, which indicates the maximum load level. More traffic in the queue will not result in more traffic injected into the channel as this has reached its maximum capacity. Each combination of STAs and MRs is tested with the three GATS policies, i.e., Legacy, DMS, and UR. The dataset generated is publicly available (available at https://www.openml.org/d/43256) and includes a wide range of KPIs that are used for evaluation purposes, although not all of them are used as features by the models. Only the channel occupancy, the share of retransmitted frames, the number of multicast stations, the injected unicast traffic, the injected multicast traffic, and the GATS policy used in the scenario are used to train the model. These features characterize the state of the channel and are the most relevant ones in the decision-making process of selecting the GATS policy. A summary of them is shown in Table 4. Training on a single cell is enough because the SD-RAN controller produces an independent output for each AP and such output already considers channel occupancy. This covers the case of a very improbable overlap (in an overlap, both cells add to the occupancy of the channel). This is

Table 3: Parameters of the dataset-building simulations.

Subset	Input	Values
Subset 1	# multicast receivers	$3n : n \in [\![1, 8]\!]$
	# unicast STAs	$3n : n \in [\![1, 9]\!]$
	Unicast STA datarate	0.5 Mbps
	Multicast datarate	1.5 Mbps
Subset 2	# multicast receivers	$3n : n \in [\![1, 8]\!]$
	# unicast STAs	$2n : n \in [\![1, 9]\!]$
	Unicast STA datarate	1 Mbps
	Multicast datarate	1.5 Mbps

Table 4: Input of the model.

Feature	Values
Channel occupancy	[0,100]
GATS policy	{GCR, DMS, Legacy}
# multicast stations	[0,255]
Share of retransmissions	[0,100]
Normalized injected multicast traffic	[0,1]
Normalized injected unicast traffic	[0,1]

realistic because IEEE 802.11ac with a 40 MHz channel bandwidth (as recommended by industry planning guidelines in campus and enterprise WLANs [21, 22]) provides enough nonoverlapping to reduce overlapping basic service sets (BSSs) and also minimize the adjacent channel interference (ACI) to a minimum using channel allocation algorithms. Moreover, the IEEE 802.11aa amendment itself specifies overlapping BSS (OBSS) management mechanisms. Thus, the normal operation of the models is a single cell with no overlapping, and the presence of overlapping BSSs that may affect the performance renders marginal even in dense enterprise deployments.

4.2. Dynamic GATS Policy Selection. Once the dataset is ready and the models are trained, the data that is generated by the AP and sent to the SD-RAN controller is used to make predictions that allow the appropriate adjustment of the network's behavior. In this respect, given an input instance vector x with n features that represent the current state of the network, the goal is to obtain the label y (the GATS to be used) that maximizes the multicast goodput. The n components of x (the selected features) are shown in Table 4. These features are the channel occupancy, the share of retransmitted frames, the number of multicast stations, the injected unicast traffic, the injected multicast traffic, and the GATS policy used in the scenario. This set of features was the one yielding a higher accuracy in the validation process of the models. The numerical input values shown in the table are then normalized in a [0,1] interval before being fed into the model. On the other side, the categorical values (the GATS policy) are converted to numerical values using a one-hot encoder; i.e., this feature is split into one feature per category, and while the category in x takes value 1, the rest takes 0.

As mentioned above, the goal is to obtain the label y (the GATS to be used) that maximizes the multicast goodput, and therefore, this is a regression problem in which the value to predict is the multicast goodput. Consequently, given x, an input matrix $X^{-1} = [x_l, x_g, x_d]$ is obtained, where x_l, x_g, and x_d are vectors such that $\forall i \in \{1, 2, \cdots, n-1\}$ and $x_{l_i} = x_{g_i} = x_{d_i}$, i.e.; all the components of the vectors are the same except for the last one ($x_{l_n} \neq x_{g_n} \neq x_{d_n}$). This last component represents the GATS policy to be used at time $t+1$ such that x_{l_n} = Legacy, x_{g_n} = UR, and x_{d_n} = DMS.

One of the chosen models then returns a vector $p = (p_l, p_g, p_d)$ containing 3 multicast goodput predictions, one for each vector in X. Next, given a function $f : Y \longrightarrow P$, where Y is the set of labels and P is the set of predictions, the arg $\max_y f(y)$ provides the label that maximizes the multicast goodput. Thus, the model receives the current state of the network and predicts the multicast goodput that each one of the GATS policies would obtain for such network conditions. Then, the function f obtains the labels for each of those predictions. The label corresponding to the highest predicted multicast goodput is configured by the SD-RAN controller on the AP, which will then be used at $t+1$. The following subsections explain the models used to make the predictions in the vector p. This methodology does not add extra complexity to the models used, i.e., the complexity to obtain the prediction is that of the model used. The complexity of each model is analyzed in the following subsections.

4.3. k-Nearest Neighbors. k-nearest neighbors (kNN) [4] is an interpretable [5] supervised learning algorithm that can be used for both classification and regression. It is also defined as a lazy learning and nonparametric algorithm. It is a lazy learning algorithm because it does not learn an approximation function from the training data. Instead, the generalization of the training data takes place when there is a query to the model. This model is chosen for this study since it is good at predicting channel status and classifying behaviors that can be grouped together (in our case, behaviors that should use the same GATS policy) [23]. Furthermore, due to its nonparametric nature, it does not assume any underlying statistical model, which allows it to behave well when there are many data points but few dimensions. This means that kNN can make accurate predictions from only the five selected features shown in Table 4. Moreover, it is inherently interpretable and requires zero training.

kNN finds the distance from every query to all the entries in the training dataset and generalizes by solving an aggregation problem (e.g., mean in regression or mode in classification) with the target values of the k closest instances. Thus, its complexity is $\mathcal{O}(n)$, where n is the number of instances in the dataset. There are different ways to compute the distance, such as Euclidean, Manhattan, or Minkowski. In this work, the Euclidean distance is used. Note that k is a hyperparameter (to be optimized). Hyperparameters are values used to control the learning process, in contrast to parameters usually calculated via training.

4.4. Random Forests. Random forest (RF) [24] is an ensemble nonparametric supervised ML method that can be used for classification and regression. First, it builds multiple decision trees at training time using samples of the original training dataset. Then, each of the trees provides its prediction. In regression problems, the final prediction is the average prediction of the trees. In contrast, each tree provides a vote in classification problems, and the final prediction is the label with the highest number of votes. Consequently, the complexity of RF depends on the number of trees and the depth of those trees. Since the complexity of a single decision tree is $\mathcal{O}(n)$ where n is the depth of the tree, it can be inferred that the complexity of RF is $\mathcal{O}(m \times n)$, where m is the number of trees and n is the maximum depth. The features used by RF are the same that were chosen for kNN. An example of this is shown in Figure 4. RF pushes the limits of interpretability to achieve better accuracy and more resiliency to outliers than decision trees [25]. RF reduces the tendency of decision trees to overfit the training dataset. Decision trees achieve good accuracy in classification problems, but in a more complex regression problem like this one, RF can help improve performance. Thus, both kNN and RF are evaluated. An important hyperparameter in decision trees and RF is the function used to measure the quality

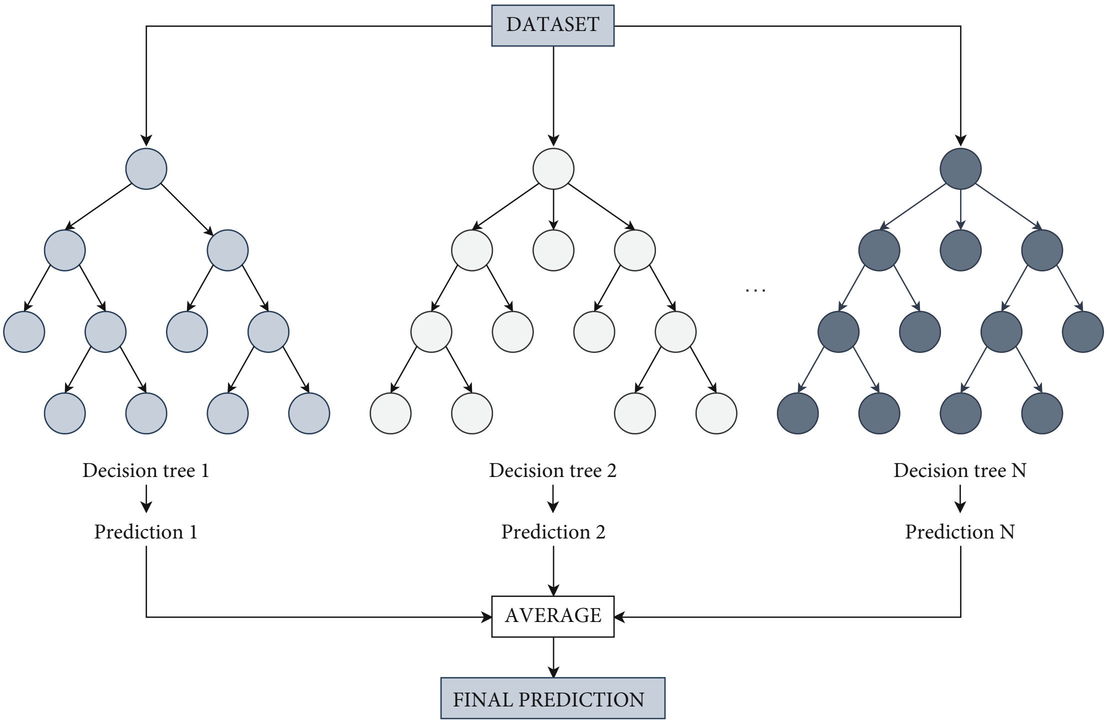


Figure 4: Generic random forest example for regression problems.

of a split, also known as the criterion. Its optimization is critical to improving the accuracy of the model. We choose this model as an eager-learning counterpart to kNN. RF has been widely used for throughput prediction in the literature [7–9] and has the same advantages of a nonparametric model that made kNN suitable for this problem.

4.5. Hyperparameter Optimization and Model Validation. The hyperparameters of an ML model affect the accuracy of its predictions, so choosing the correct values is of vital importance. For the purpose of optimizing these values, we employ grid search which is a technique that performs an exhaustive search to find the hyperparameters that maximize the accuracy of the predictions. Accuracy refers to the share of times that the model predicts the correct label. Such accuracy is calculated via 5-fold cross-validation (CV). The training dataset is divided into 5 splits so that 4 of them are used for training while the other group is kept aside and used to test the model; i.e., check whether the model's predictions are correct. The process is repeated until all the splits have been used for testing. The final accuracy is the average of the accuracy obtained by each test group. Note that grid search is a brute force algorithm, so a specific interval of hyperparameter values in which the search is carried out must be specified. In this particular case, the interval [1,10] is tested for k, which is the only hyperparameter in kNN. The search returns 2 as the optimal value for k. The 2NN is then validated using the same 5-fold CV procedure, obtaining an accuracy of 95.58%. Similarly, the criterion is optimized in RF. The search returned Friedman MSE [26] as the optimal one. After optimizing the hyperparameters, the model is validated using the same 5-fold CV, obtaining an accuracy of 94.89%.

Table 5: Simulator parameters.

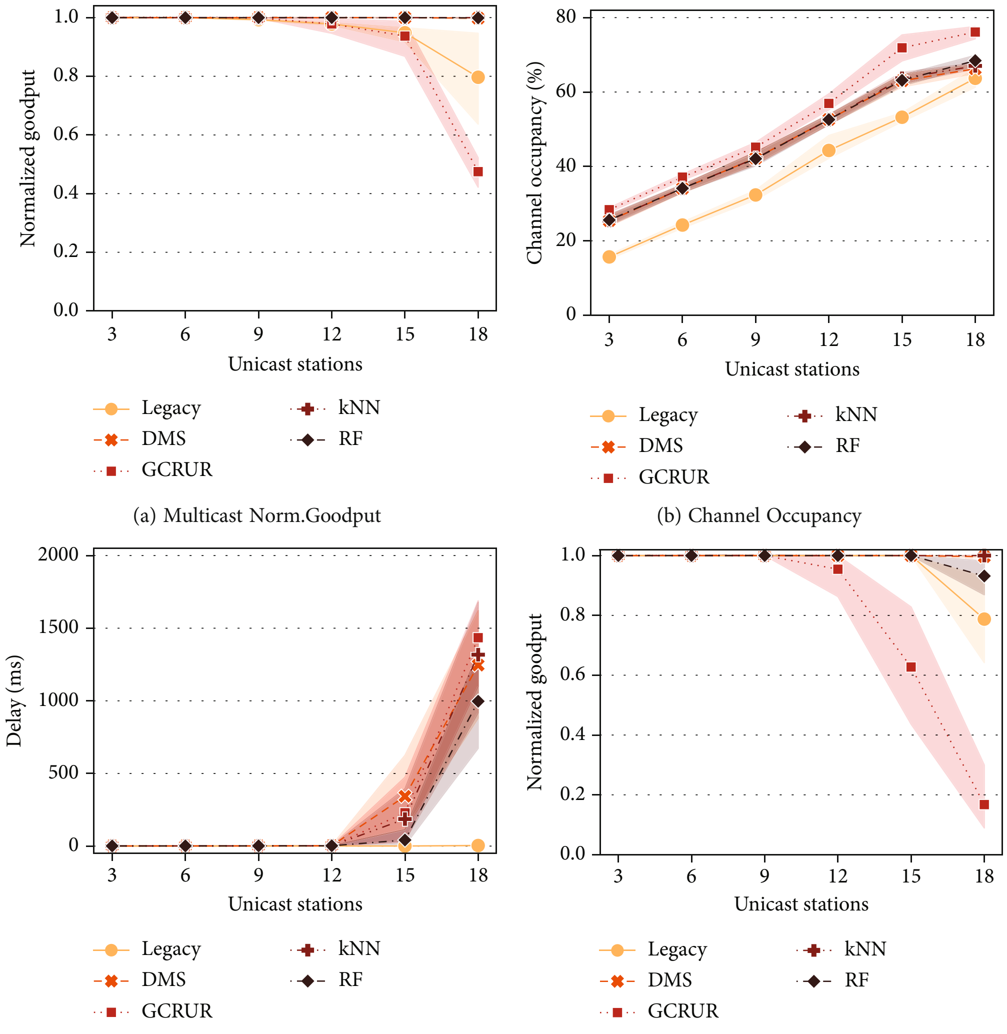
Setting	Value
PHY layer	802.11ac
Rate adaption	Minstrel
Channel width	40 MHz
Tx power	15 dbm
Error rate model	Table-based
Max distance	30 m

5. Performance Evaluation

In this section, the proposed dynamic GATS selection mechanism is evaluated. Both kNN and RF are compared against the static GATS defined in the IEEE 802.11aa amendment: Legacy, GCR-UR, and DMS. Since the literature does not provide any other dynamic method, no other benchmarks other than the static GATS policies can be shown for comparison. The proposed solution was tested using the same setting shown in Figure 3. The evaluation is carried out on a single cell, as the SD-RAN controller produces independent outputs considering the specific conditions of each AP. Moreover, the input already accounts for channel occupancy, which considers occupancy caused by OBBSs. Furthermore, the IEEE 802.11aa already provides mechanisms to deal with OBSSs, which are out of the scope of this study. In fact, in this evaluation, the channel width is limited to 40 MHz to simulate an enterprise WLAN, where the use of 80 MHz and 160 MHz channel bandwidths is discouraged by the industry guidelines [21, 22] due to the limited number

Table 6: Evaluation parameters.

Test	# MRs	Multicast datarate	# unicast STAs	Unicast STA datarate
Scalability	$2m : m \in [\![1, 11]\!]$	1.5 Mbps	Scenario 1: $u = 9$ Scenario 2: $u = 18$	0.5 Mbps stations in $A = \{3n : n \in [\![1, u/3]\!]\}$ 0.75 Mbps stations in $B = \{3n - 1 : n \in [\![1, u/3]\!]\}$ 1 Mbps stations in $C = \{3n - 2 : n \in [\![1, u/3]\!]\}$
Efficiency	Scenario 1: $m = 3$ Scenario 2: $m = 12$		$u : u = 3n ; n \in [\![1, 6]\!]$	
Adaptation	$m = 5$		$u : u = 3n ; n \in [\![2, 6]\!]$	



(a) Multicast Norm.Goodput

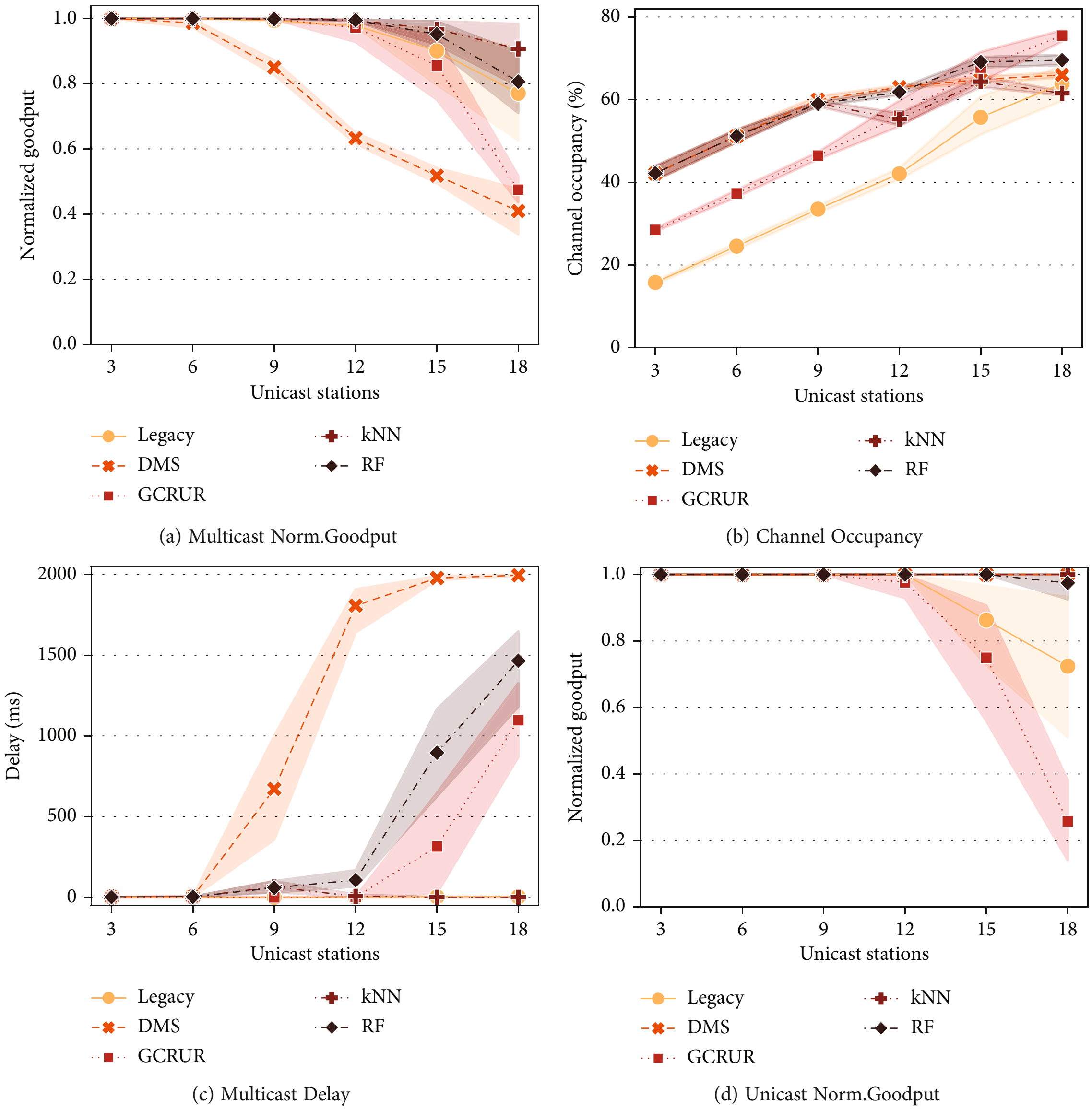
(b) Channel Occupancy

(c) Multicast Delay

(d) Unicast Norm.Goodput

Figure 5: Scenario 1 of the efficiency evaluation ($m = 3$).

of nonoverlapping channels. Considering this, the normal operation of the approach presented in this paper is a single independent cell. Despite evaluating using the same topology, changes to the datarates, the distance of the stations, and the number of these were made to introduce variability with respect to the training scenarios. These changes are described in detail in the following subsections. The simulations were carried out using the IEEE 802.11ac model of the NS-3 network simulator, and Minstrel was used for rate adaption of the unicast and DMS transmissions as one of the most implemented by vendors and the one included in Linux kernels. The Tx power is set to 15 dbm following European Communications Office recommendations [27]; although the maximum allowed Tx power by ETSI is set at 23 dbm, this is hardly ever reached by vendors. The rate error model used is the table-based provided by NS-3, as is the only one that recreates the bit error rate (BER) for the VHT physical layer in NS-3. The rest of the parameters are left unchanged. A summary on the physical layer settings chosen in the simulator is given in Table 5.



(a) Multicast Norm.Goodput

(b) Channel Occupancy

(c) Multicast Delay

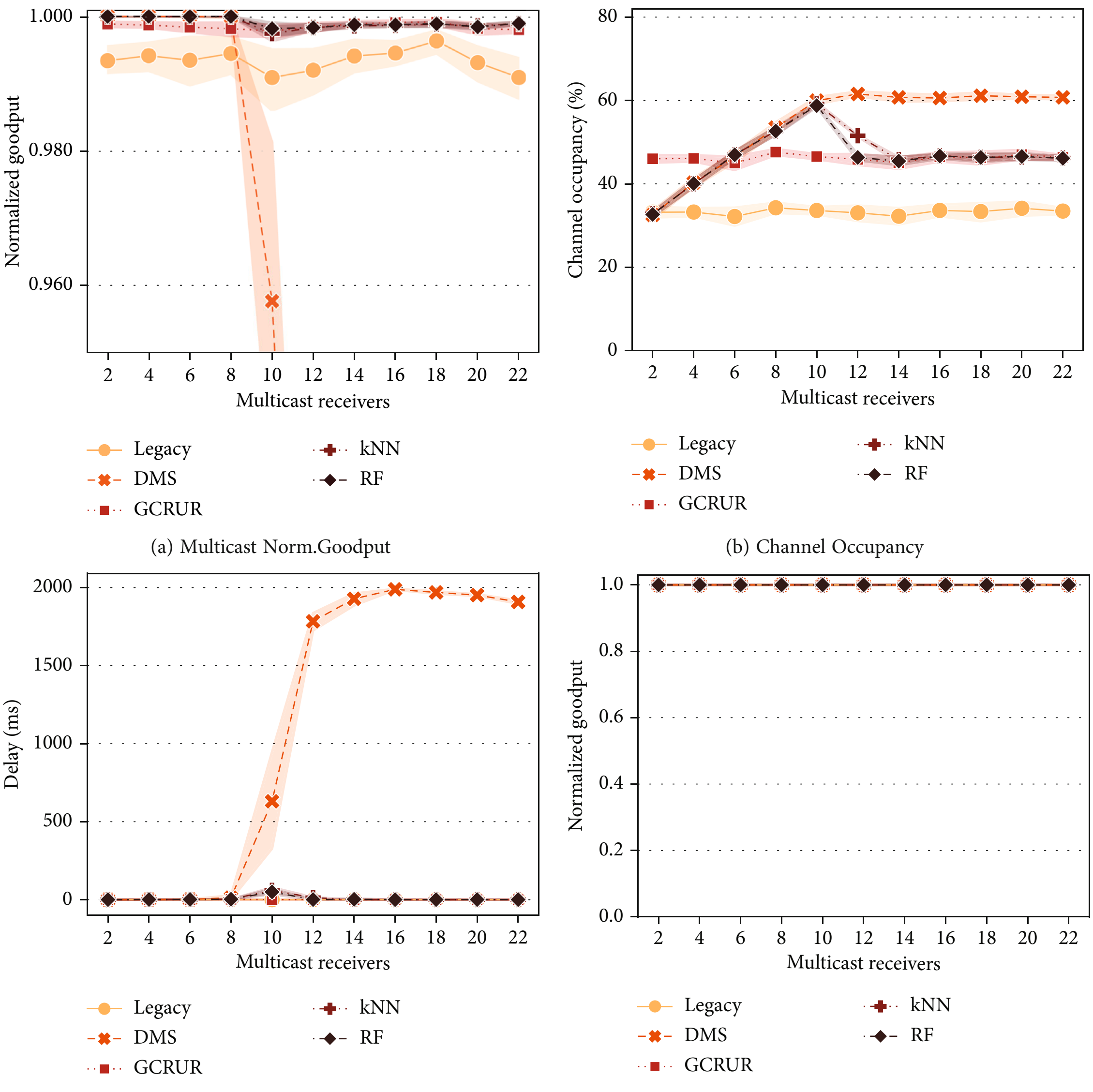
(d) Unicast Norm.Goodput

Figure 6: Scenario 2 of the efficiency evaluation ($m = 12$).

In all the simulations, EDCA access categories were used. The multicast group downloaded VI traffic, and the STAs downloaded and transmitted BE traffic. Thus, multicast (VI) and unicast (BE) traffic were in different queues. All the stations were randomly positioned inside a disc with a 30-meter radius, and the AP is in the center of the disc. Note that the distance is reduced from 40 to 30 meters only to introduce variability on quality channel with respect to the training scenarios. Each simulation represented 30 seconds, during which and the average of the last 20 seconds was computed and used for the evaluation. This was repeated 10 times for each case. A multicast datarate of 1.5 Mbps was chosen in order to represent a high-definition video call. The evaluation focused on the scalability, efficiency, and adaptability of Intelli-GATS. In the following subsections, the scenarios designed to test these properties are explained.

The main KPIs that reflect the status of the network and the quality of the multicast transmissions are the average normalized multicast goodput, which shows the quantity of traffic being correctly delivered to the multicast stations in relation to that which was injected by the source; the unicast goodput, which shows how the approach affects the unicast transmissions; the channel occupancy, which shows how efficiently the wireless resources, i.e., airtime, are being used; and the delay, which shows how long a frame has been waiting in the queue to be transmitted, or whether the frames are exceeding the deadline.

5.1. Efficiency. In heavily loaded networks, the dynamic selection of the GATS policies must improve network efficiency, not necessarily by decreasing the load on the network but by being able to fit more traffic into the channel and making the most of the available airtime. Thus, to test efficiency, two other scenarios were defined: scenario 1, which has a small multicast group ($m = 3$ MRs), and scenario 2, which has a big multicast group ($m = 12$ MRs). Group sizes



(a) Multicast Norm.Goodput

(b) Channel Occupancy

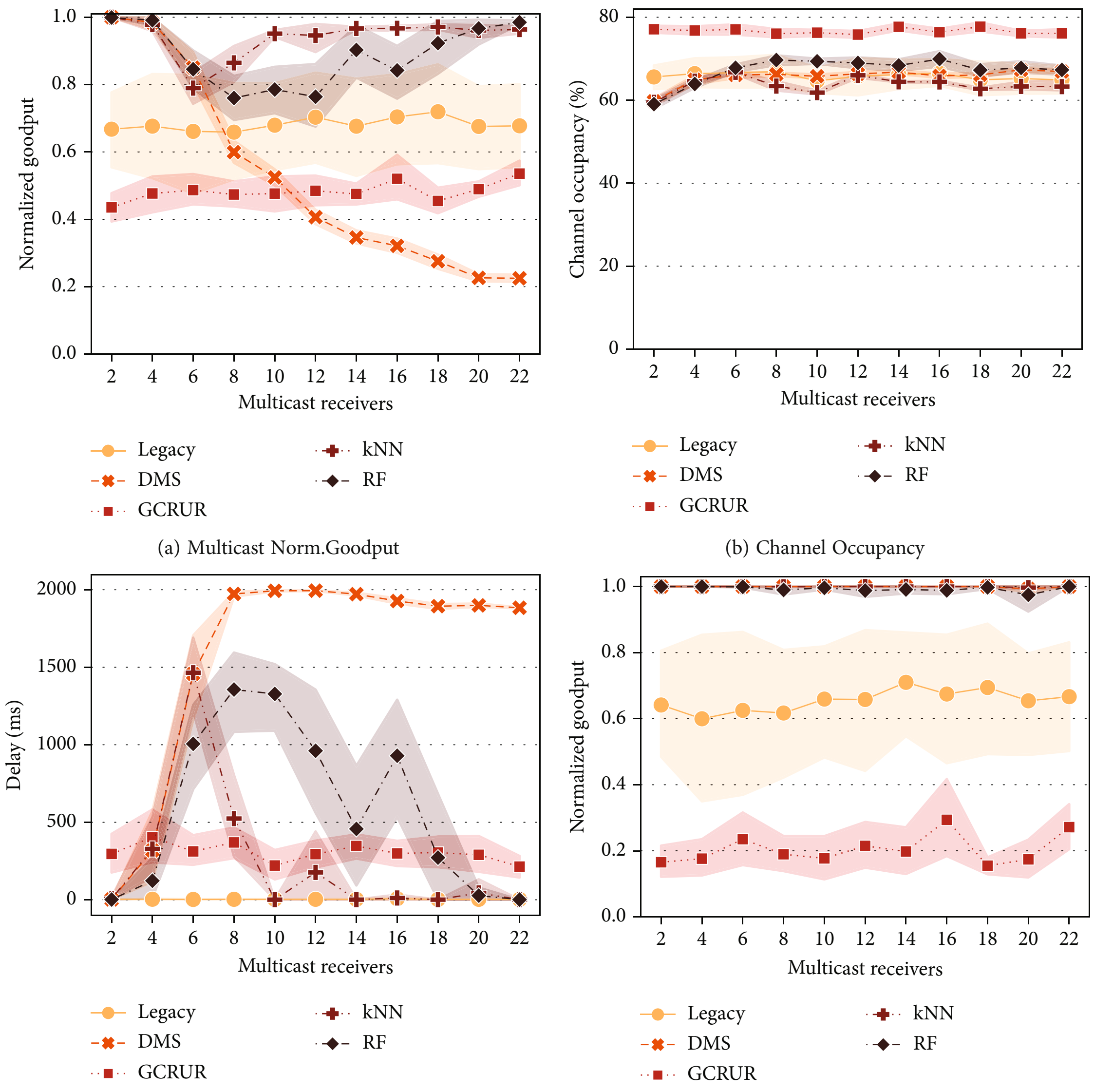
(c) Multicast Delay

(d) Unicast Norm.Goodput

Figure 7: Scenario 1 of the scalability evaluation ($u = 9$).

are determined by doing preliminary simulations. These simulations show that groups bigger than 12 do not report changes in the behavior, as DMS cannot handle that many stations, and Legacy and GCR-UR are unaffected by the group size. This is assessed in detail in Result Analysis. To evaluate the efficiency of these scenarios, the load was gradually incremented by increasing the number of STAs. Thus, the number of STAs for each test case was given by $u : u = 3n\,; n \in [\![1, 6]\!]$; i.e., tests with all the number of STAs being multiples of 3 between 3 and 18 were carried out. To determine where to stop, preliminary tests were carried out. With 18 STAs, the network reaches saturation, and adding more STAs or more load does not change conditions, as the network is already saturated. For all of these tests, the STAs were divided into three groups following the same pattern as in the scalability scenarios, as indicated in Table 6. Each station transmitted or received at 0.5 Mbps, 0.75 Mbps, or 1 Mbps, depending on the group to which it belonged. Note that these scenarios and the scalability ones were chosen to be different from the training scenarios.

5.2. Scalability. The size of a multicast group affects the selection of the GATS policy, so Intelli-GATS should be able to adapt to a growing number of MRs. To test this, two scenarios were defined with different network loads: scenario 1, which had only 9 STAs ($u = 9$) to simulate a low load in the network, and scenario 2, which had 18 STAs ($u = 18$) to simulate a high load on the network. These load levels are determined by looking at preliminary simulations and the results of the efficiency part simulations. With 18 STAs, the network reaches saturation, and adding more load does not change conditions, as the network is already saturated. For each of these scenarios, a whole range of multicast group sizes given by $2m : m \in [\![1, 11]\!]$ were tested, where m is the number of MRs. Thus, all the even group sizes between 2 and 22 were tested. The load on the network was introduced



(a) Multicast Norm.Goodput

(b) Channel Occupancy

(c) Multicast Delay

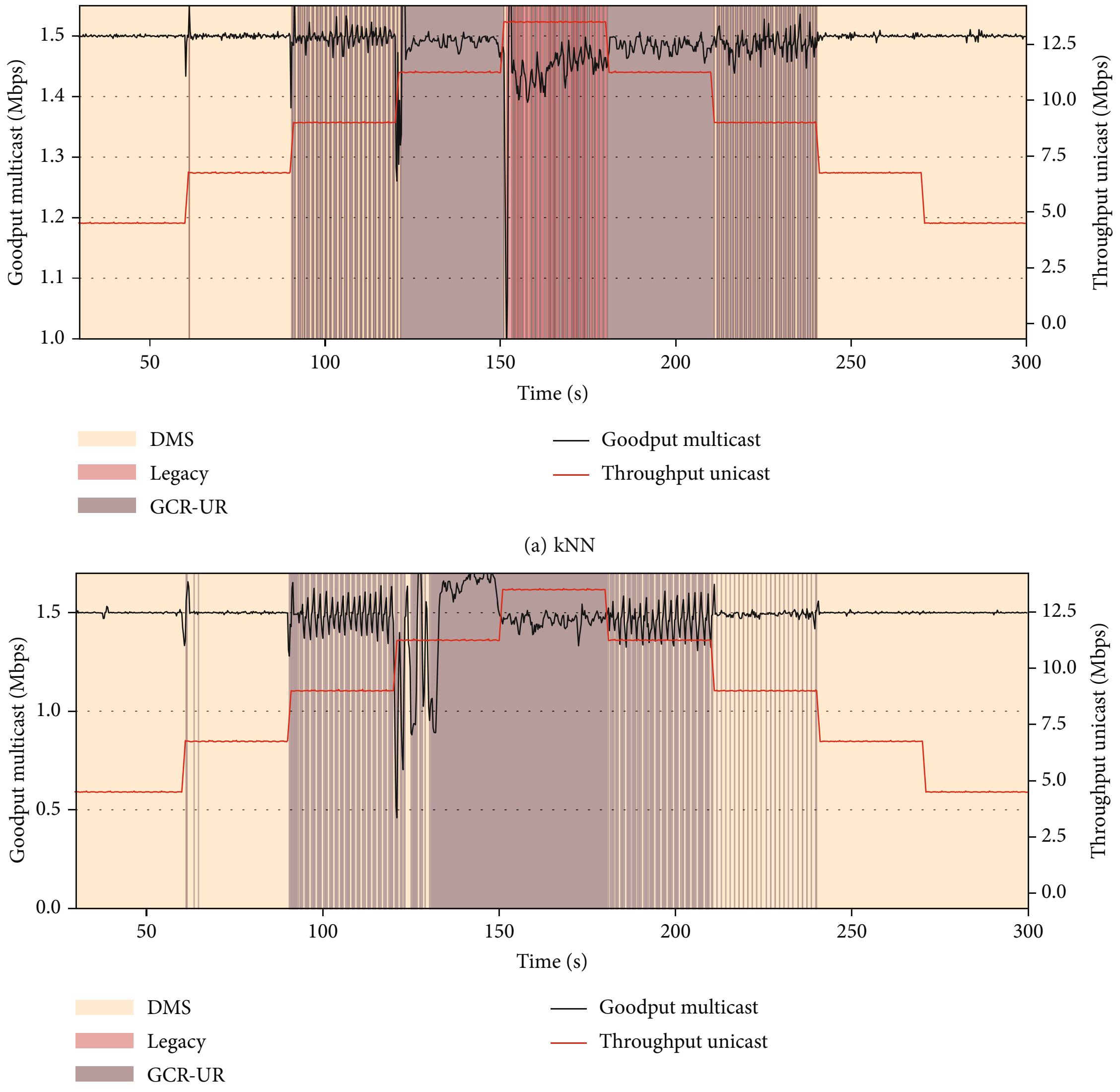
(d) Unicast Norm.Goodput

Figure 8: Scenario 2 of the scalability evaluation ($u = 18$).

through unicast traffic, and the STAs received and transmitted the same amount of traffic all the time, and consequently, different load levels were achieved by changing the number of STAs. All the STAs in the group given by $A = \{3n : n \in [\![1, u/3]\!]\}$, where u is the number of STAs, transmitted and received 0.5 Mbps each. In the same way, all the STAs in the group $B = \{3n - 1 : n \in [\![1, u/3]\!]\}$ transmitted 0.75 Mbps. Finally, all the STAs in the group $C = \{3n - 2 : n \in [\![1, u/3]\!]\}$ transmitted 1 Mbps each. That is, the STAs were divided into 3 groups, each of which transmitted and received traffic at a different datarate. Therefore, the number of STAs was always a multiple of 3. This was done to create more variability with respect to the scenarios in the training dataset.

5.3. Adaptability. An additional evaluation scenario was designed to test how fast the models are able to adapt to changing network conditions. This does not affect the static GATS selection as one of their main drawbacks is that they do not adapt at all. With a fixed number of MRs, the whole scenario simulated 5 minutes of network operation. Once again, the load was defined by the number of STAs that constantly transmitted at a defined rate, depending on the group to which they belonged (see Table 6). For the first minute, the network conditions remained static, with only 6 STAs transmitting and receiving ($n_t = 2$). The first 30 seconds was taken as network warm-up and was not considered in the evaluation. After the first minute, 3 more STAs started transmitting, i.e., $n_t = n_{t-1} + 1$. This was repeated every 30 seconds until $n = 6$. This load level was kept for 30 more seconds until $t = 270$, when the load started to decrease by stopping 3 STAs ($n_t = n_{t-1} - 1$). Again, this was repeated in the same period until $n_t = 2$.

5.4. Result Analysis. In this subsection, the results of the simulations for the properties mentioned above, including the



(a) kNN

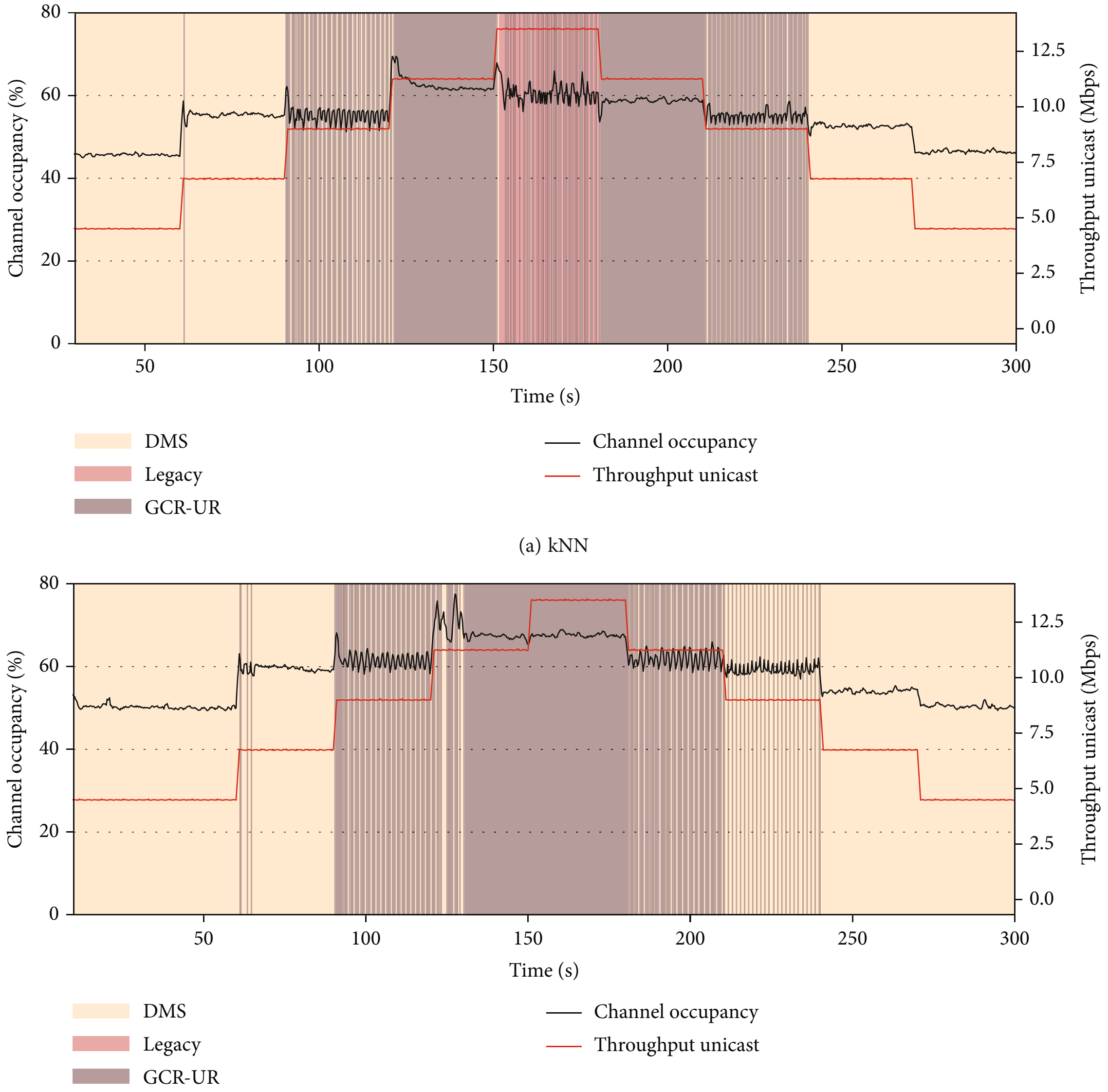
(b) Random Forest

Figure 9: Multicast goodput in the adaptability test.

scenarios defined for each of them, are presented. First, we look at efficiency.

5.4.1. Efficiency. Figure 5 shows the efficiency results in scenario 1, which uses a small multicast group ($m = 3$ MRs). In particular, Figure 5(a) shows how DMS performs best with smaller multicast groups, and it is able to deliver nearly 100% of the frames at all the load levels represented in the x-axis. This is thanks to the use of a single unicast frame for each MR instead of a multicast frame that has to be received by all of them. This enables DMS to use the ACKs, which provides the feedback necessary for the use of the Minstrel rate adaption algorithm. Moreover, the reliability of DMS is better than that achieved by Legacy and GCR-UR as each unicast frame is acknowledged by an ACK frame. Consequently, in this scenario, both kNN and RF choose DMS to equal its performance. Conversely, Legacy and GCR-UR have to use the basic rate. This causes an increment in the channel occupancy, especially in GCR-UR, as shown in Figure 5(b). Consequently, it is the logical decision for the models to constantly choose DMS in this scenario, as both GCR-UR and Legacy are underperforming. This bigger occupancy also results an increase in the delay, as shown in Figure 5(c). Since kNN and RF use DMS, they are also able to deliver nearly 100% of the frames. Another important factor to take into account is how this affects unicast stations. Multicast frames, which are transmitted on the VI AC, take priority over unicast frames as these are transmitted over the BE AC. In Legacy and GCR-UR, VI frames are transmitted at a basic rate, and therefore, they keep the channel busy for longer. This affects the performance of unicast stations, as shown in Figure 5(d), which displays the evolution of the average normalized unicast goodput. RF and kNN do not suffer from this, as they select DMS, which uses Minstrel to transmit frames at higher rates.

Things are different in scenario 2, when a big multicast group is used ($m = 12$ MRs). DMS does not scale; as for each multicast frame, it needs to transmit a unicast frame for each



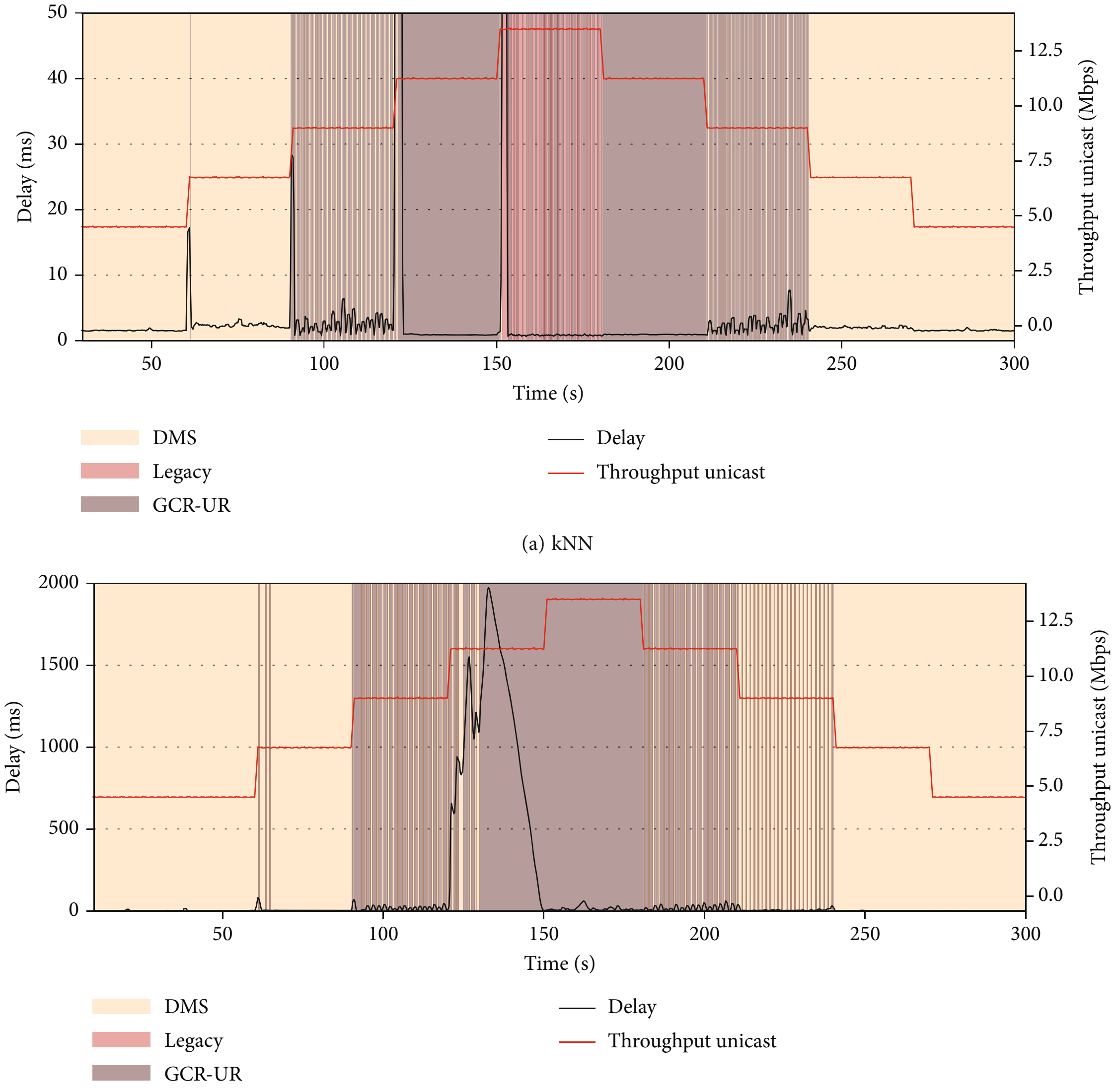
(a) kNN

(b) Random Forest

Figure 10: Occupancy in the adaptability test.

MR. This causes a lot of overhead in bigger multicast groups even if the frames are transmitted at higher rates. Figure 6 shows the results for scenario 2. In particular, Figure 6(a) shows how DMS performance starts to drop fast, and with only 6 STAs it already underperforms Legacy and GCR-UR. This happens because the channel cannot cope with the overhead caused by the bigger multicast group. The size of the multicast group does not affect GCR-UR or Legacy, whose performance is similar to that in scenario 1. However, kNN and RF cannot longer rely on DMS alone, as the performance of this one is no longer good across the whole *x*-axis, and it only behaves well for the smallest groups. When the network is not congested, the models rely on GCR-UR to achieve better reliability, as shown by Figure 6(b). However, when the network load increases more, the performance of GCR-UR drops due to the higher congestion caused by the retries sent at the basic rate, and the models start combining DMS periods and Legacy periods to achieve better reliability with less occupation. This happens, because during the DMS periods, frames are delivered with high reliability while during Legacy periods, the queue, which was flooded by DMS, is emptied and congestion eases. kNN achieves a more efficient combination than RF, which also uses periods of GCR-UR that are causing more congestion on the network. This bigger congestion is clearly shown by the delay in Figure 6(c).

The improvement in the multicast goodput achieved by kNN and RF also results in an indirect improvement in the unicast goodput as shown in Figure 6(d). Even though the channel occupancy obtained by them is similar, the more efficient use of the airtime allows unicast traffic to use the channel time freed by the dynamic approach. DMS also obtains good results in terms of unicast goodput. The bigger congestion caused by DMS overhead is getting the AP's queue flooded with multicast frames. Since the airtime is limited, not all can gain medium access before the deadline and are dropped. Thus, most of them never reach the



(a) kNN

(b) Random Forest

Figure 11: Delay in the adaptability test.

channel, which leaves more airtime for unicast frames. This is causing unicast traffic to perform well when using DMS. Similarly to what happens in scenario 1, GCR-UR overloads the channel by transmitting and retransmitting multicast frames at the basic rate, and its performance drops compared to Legacy and the dynamic approaches.

5.4.2. Scalability. Figure 7 shows the results for the medium-low load scenario ($u = 9$ STAs). In particular, Figure 7(a) shows how DMS performs best with smaller multicast groups. More specifically, in this scenario, the performance falls rapidly after 8 MRs, clearly showing DMS's scalability problems. Consequently, both kNN and RF select DMS when the multicast group is smaller than 8 stations. After this, the lines clearly diverge as the models start to combine DMS with the other two policies. As mentioned in the previous subsection, with lower loads, GCR-UR performs better than Legacy as there is enough free airtime to fit the retries without causing congestion, which improves reliability. Thus, in this scenario, both kNN and RF start using GCR-UR with groups bigger than 8 MRs. Channel occupancy clearly shows this in Figure 7(b), in which the kNN and RF lines follow DMS until it starts saturating the channel, at which point they start following the GCR-UR line. Even with a low network load, Figure 7(c) shows how as the number of MRs grows; part of the big amount of frames produced by DMS is no longer able to gain medium access before exceeding the deadline and being dropped. In this scenario, the low load of the network does not cause any problem for unicast transmissions which are able to deliver all traffic correctly, as shown in Figure 7(d).

Understandably, if the load on the network is increased, DMS's performance will drop, even with fewer MRs as there is less margin in the channel for the overhead produced by this policy. This can be seen in Figure 8(a), which shows how DMS starts struggling with just 6 MRs. As shown in the efficiency evaluation, in scenarios with channel congestion, Legacy behaves better than both GCR-UR and DMS for bigger multicast groups, as it introduces less overhead on the channel. In this context, kNN and RF combine DMS, GCR, and Legacy to achieve better reliability without flooding the queue. Even if during DMS periods, the queue starts to fill, Legacy periods and, to a lesser extent, GCR-UR periods can reduce the load on the queue as they introduce less overhead on the channel. This results in higher goodput with appreciably less occupancy, as shown in Figure 8(b). Nevertheless, RF does not use Legacy, and as a consequence, it does not unload the queue and the channel as much as kNN. This can be seen by the delays shown in Figure 8(c). This scenario clearly shows how the channel congestion caused by GCR-UR harms the unicast goodput, as shown in Figure 8(d). The same happens to a lesser extent with Legacy, which, despite not having retries, the fact that frames are sent at the basic rate increases the airtime used by multicast traffic, which has a negative impact on the BE traffic. However, the good behavior of DMS in terms of unicast goodput comes at the cost of bad multicast performance. When using DMS, multicast frames are dropped because they exceed the deadline, as shown by the delay in Figure 8(c). Thus, most of the multicast frames never reach the channel, which leaves more airtime for unicast frames. Since kNN and RF reduce the overhead on the channel by combining periods of Legacy, GCR-UR, and DMS, they achieve similar performance.

5.4.3. Adaptability. Finally, this part of the evaluation shows how both models assessed in Intelli-GATS adapt to changes in network conditions. The models receive updates to the selected features every 250 ms and use these updates to predict the best GATS policy for each AP. Then, with this output, the configuration of the APs is updated accordingly. The results shown in Figure 9 display the multicast goodput achieved (black line, left y-axis) when the load (red line, right y-axis) is incremented in different steps by activating stations, and the background color indicates the GATS policy chosen by the model. In particular, Figure 9(a) shows the decisions taken by kNN. As explained above, when the load is low, DMS can deliver most frames, but as the load progressively increases, it combines GCR-UR with DMS to reduce the load on the channel in the GCR-UR periods. Nevertheless, combining DMS and GCR-UR is not enough when the load increases again, and only GCR-UR is used as it provides better reliability than Legacy. However, when the load increases again between $t = 150$ s and $t = 180$ s, it combines GCR-UR, Legacy, and shorter periods of DMS. During Legacy periods, the network load is reduced, while reliability is improved during GCR-UR and DMS periods, which improves the overall multicast goodput, as explained in the scalability and efficiency scenarios. Thanks to this, Intelli-GATS keeps the multicast goodput stable even with a saturated network as shown in Figure 10(a). Figure 11(a) also shows how Intelli-GATS reacts to changes in the network. When there is a peak in the delay, the switch in the GATS policy controls the keep and brings it back to previous levels thanks to the reduction in the load of the channel, which allows more frames to gain medium access and therefore reduce the waiting time in the queue. On the other hand, RF needs more time to adapt to the new network conditions. Because of this, the network goes through a short period with a low multicast goodput (Figure 9(b)) and a high delay (Figure 11(b)), but soon, the network adapts and the performance stabilizes. However, during the period of maximum load, RF does not alternate with Legacy, which results in a higher channel occupancy, as shown in Figure 10(b), but a similar multicast goodput.

6. Conclusion

In this paper, we have presented Intelli-GATS, which is an Interpretable-AI approach for the dynamic selection of the GATS policy in Wi-Fi networks that best suits the network conditions at any given moment. The performance evaluation carried out has shown that Intelli-GATS outperforms the static standard policies by up to 38% thanks to the rapid adaption of the models, which allows the network to rapidly switch between policies to achieve better reliability while keeping channel occupancy within an acceptable range.

Two ML models have been evaluated, namely, kNN and RF. kNN, which is inherently interpretable, outperforms RF, which pushes the limits of interpretability. Thus, kNN is able to take transparent and more accurate decisions that can be trusted by human experts, and these decisions make troubleshooting easier than those taken by RF. This shows that directing research efforts towards Interpretable-AI techniques can be beneficial for the effective use of ML in real-life deployments, as accuracy is not necessarily compromised and the stakeholders can trust the system. Models that can process time series to detect the negative evolution of delays and react even more rapidly to changes in the conditions can be assessed in the future.

Data Availability

The data used to support the findings of this study are included within the article. Data used for training of the models can be found in: https://www.openml.org/d/43256.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is part of the R&D project PID2021-123627OB-C52, funded by the MCIN and the European Regional Development Fund: “a way of making Europe”. This work is also funded by the European Union: “The European Social Fund investing in your future” (Grant 2019-PREDUCLM-10921) and the Government of Castilla-La Mancha (project SBPLY/21/180501/000195). This work has also been supported by the EU “NextGenerationEU/PRTR,” MCIN, and Agencia Estatal de Investigación (Spain) under project IJC2020-043058-I.

References

[1] X. Dong, J. Hart, and F. Qiang, "Evaluation of the Minstrel rate adaptation algorithm in IEEE 802.11g WLANs," in *Proc. of IEEE International Conference on Communications (ICC)*, Budapest, Hungary, 2013.

[2] "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, 2012, https://ieeexplore.ieee.org/document/6178212.

[3] ETSI, "MEC 028- V2.1.1 - multi-access edge computing (MEC); WLAN information API," Technical Report 028, 2020.

[4] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[5] C. J. Hazard, C. Fusting, M. Resnick, M. Auerbach, M. Meehan, and V. Korobov, "Natively interpretable machine learning and artificial intelligence: preliminary results and future directions," 2019, https://arxiv.org/abs/1901.00246.

[6] L. Hao and B. Ng, "Self-healing solutions for Wi-Fi networks to provide seamless handover," in *Proc. of IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, Arlington, VA, USA, 2019.

[7] K. Kousias, Ö. Alay, A. Argyriou, A. Lutu, and M. Riegler, "Estimating downlink throughput from end-user measurements in mobile broadband networks," in *Proc. of IEEE International Symposium on "a World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pp. 1–10, Washington, DC, USA, 2019.

[8] A. Samba, Y. Busnel, A. Blanc, P. Dooze, and G. Simon, *1ères Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication*, CORES2016, Bayonne, France, 2016.

[9] M. A. Khan, R. Hamila, N. A. Al-Emadi, S. Kiranyaz, and M. Gabbouj, "Real-time throughput prediction for cognitive Wi-Fi networks," *Journal of Network and Computer Applications*, vol. 150, article 102499, 2020.

[10] G. F. Riley and T. R. Henderson, "The *NS-3* network simulator," in *Modeling and Tools for Network Simulation*, pp. 15–34, Springer, 2010.

[11] H. Yin, P. Liu, K. Liu et al., "Ns3-Ai: fostering artificial intelligence algorithms for networking research," in *Proc. of Workshop on NS3*, Gaithersburg, MD, USA, 2020.

[12] "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 8: IEEE 802.11 wireless network management," IEEE Std 802.11v-2011, 2011.

[13] "Wi-Fi network roaming with 802.11k, 802.11r, and 802.11v on iOS, iPadOS, and macOS," 2022, https://support.apple.com/en-bh/HT202628.

[14] A. Banchs, A. de la Oliva, L. Eznarriaga, D. R. Kowalski, and P. Serrano, "Performance analysis and algorithm selection for reliable multicast in IEEE 802.11aa wireless LAN," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 8, pp. 3875–3891, 2014.

[15] 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. of IEEE International Symposium on "a World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, Madrid, Spain, 2013.

[16] Y. Daldoul, D.-E. Meddour, T. Ahmed, and R. Boutaba, "Performance and scalability evaluation of IEEE 802.11v/aa multicast transport," *Wireless Communications and Mobile Computing*, vol. 16, no. 14, 2000 pages, 2016.

[17] P. Salvador, L. Cominardi, F. Gringoli, and P. Serrano, "A first implementation and evaluation of the IEEE 802.11aa group addressed transmission service," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 1, pp. 35–41, 2013.

[18] F. Gringoli, P. Serrano, I. Ucar, N. Facchi, and A. Azcorra, "Experimental QoE evaluation of multicast video delivery over IEEE 802.11aa WLANs," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2549–2561, 2019.

[19] K. Mansour, I. Jabri, and T. Ezzedine, "IEEE 802.11 aa GATS mechanisms performance study for very high throughput WLANs," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, Barcelona, Spain, 2018.

[20] Microsoft, "Microsoft Teams - network requirements," 2022, https://docs.microsoft.com/en-us/microsoftteams/prepare-network.

[21] Cisco, "High density Wi-Fi deployments," 2020, https://nnnmeraki.com/Architectures_and_Best_Practices/Cisco_Meraki_Best_Practice_Design/Best_Practice_Design__MR_Wireless/High_Density_WiFi_Deployments.

[22] Aruba, "Campus wireless networks (6.x) validated reference design," 2017, https://www.arubanetworks.com/assets/vrd/2018_01_02_Campus_6_x_Whole_VRD.pdf.

[23] H. Yang, X. Xie, and M. Kadoch, "Machine learning techniques and a case study for intelligent wireless networks," *IEEE Network*, vol. 34, no. 3, pp. 208–215, 2020.

[24] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[25] G. Louppe, *Understanding Random Forests: From Theory to Practice, [Ph.D. thesis]*, University of Liege, 2014.

[26] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *The Annals of Statistics*, vol. 29, pp. 1189–1232, 2001.

[27] ECC CEPT, "ERC recommendation 70-03, relating to the use of short range devices (SRD)," *Electronic Communications Committee*, 2017.