Programming Abstractions for Wireless Multicasting in Software–Defined Enterprise WLANs

Estefanía Coronado*, Roberto Riggio‡, José Villalón* and Antonio Garrido*
*High-Performance Networks and Architectures (RAAP). University of Castilla–La Mancha, Albacete, Spain
Email: {Estefania.Coronado, JoseMiguel.Villalon, Antonio.Garrido}@uclm.es
‡CREATE-NET, Trento, Italy
Email: Roberto.Riggio@create-net.org

***Abstract*—The increasing demand for multimedia content and for live broadcasting is bringing renewed interest to multicast applications. In many cases, users access such streams using Wi–Fi networks. However, multicast over Wi–Fi poses several challenges including low–data rates and reliability issues with regard to other unicast streams. Software Defined Networking (SDN) has recently emerged as a novel approach to network control and management. In this paper we present *SDN@Play*, a novel SDN–based solution for multicast rate–adaptation in Wi–Fi networks. The solution builds upon a new abstraction, named *Radio Port* which allows the SDN controller to reconfigure or replace a certain rate control policy if its optimal operating conditions are not met. An experimental evaluation carried out over a real–world testbed shows that this approach can deliver an improvement of up to 80% in terms of channel utilization compared to legacy 802.11 multicast. We release the entire implementation including the controller and the data–path under a permissive license for academic use.**

***Keywords*—WLANs, IEEE 802.11, multicast, rate adaptation, software defined networking.**

I. Introduction

Multimedia content delivery has witnessed a dramatic increase in popularity in the last decades. The growth in usage of platforms like YouTube and Netflix is a clear statement in support of this trend. Multicast transmissions are a particular use case within the generic multimedia content delivery domain where the same content is to be delivered to multiple destinations or receptors. Common examples of multicast applications are live broadcasting, online courses and tutorials, and multiplayer gaming. Due to the popularity of such applications in both the home and the enterprise networking domains, it is of capital importance to properly support them on 802.11–based WLANs (which is the most popular Wireless LAN technology). However, in addition to the challenges raised by multicast transmissions in the wired domain, such as routing and group management, WLANs pose a completely new set of difficulties.

WLANs based on the 802.11 family of standards dynamically choose among differed modulation and coding schemes (MCS) for frame transmission. For example, in the case of 802.11a/g networks, devices can choose bit–rates varying from 1 to 54 Mb/s, while in the case of the 802.11n/ac networks higher throughput MCSes are also available. This process, known as rate adaptation, is however restricted to unicast frame transmissions. For this reason, 802.11 uses a two–way handshake protocol where each data transmission must be acknowledged by the receiver. However, in the case of multicast transmissions, acknowledgments cannot be used as they would inevitably collide at the transmitter. As a result, multicast transmissions are usually performed at the lowest MCS (in order to increase both the range and reliability of the transmission) and do not use any form of transmission feedback mechanism. This has several drawbacks: (i) it severally limits the throughput for multicast transmissions, (ii) it consumes a significant portion of the available airtime affecting also the capacity available to other (unicast) flows, and (iii) given that multicast frames cannot be retransmitted, the reliability of the multicast streams can be adversely impacted.

Software Defined Networking (SDN) has recently emerged as a new way of refactoring network functions. By clearly separating data–plane from control–plane and by providing high–level programming abstractions, SDN allows to implement traditional network control and management tasks on top of a *logically* centralized controller. However, albeit SDN is already an established technology in the wired domain, with OpenFlow playing the role of de–facto standard [1], equivalent solutions for wireless and mobile networks have only recently started to appear [2], [3].

The contribution of this paper is twofold. First, we introduce a new programming abstraction for multicast communications. Second, we use such an abstraction to implement *SDN@Play*, an SDN multicast rate adaptation scheme for 802.11–based WLANs. The proposed solution allows utilizing higher bitrates for multicast transmissions while preserving the reliability of the communication. Based on a real–world testbed evaluation we have been able to demonstrate an improvement of up to 80% in terms of channel utilization compared to standardized multicast schemes for 802.11 networks. We release the entire implementation including the controller and the data–path under a permissive APACHE 2.0 license[1] for academic use.

The rest of this paper is structured as follows. In Sec. II we discuss the related work. We delve into the *SDN@Play* design in Sec. III, whereas in Sec. IV the implementation details are presented. Section V describes the evaluation methodology and discusses the results of the measurements. Finally, Sec. VI draws the conclusions pointing out future work.

[1] http://empower.create-net.org/

II. Related Work

In this section we shall firt provide a short background on multicast communications in IEEE 802.11–based WLANs, then we will review the most relevant related work highlighting our technical contributions.

Multicast communications are an efficient way to send the same information to many clients. In fact, by leveraging on the broadcast nature of the wireless medium, it is possible to deliver the same frame to multiple wireless terminals instead of transmitting it individually to each of them. Nevertheless, in IEEE 802.11 WLANs multicast frames are never retransmitted nor acknowledged. As a consequence, transmission reliability is highly reduced. Moreover, the lack of feedback information makes it impossible to adapt the transmission data rate, hence being the basic rate used instead.

The IEEE 802.11aa amendment has been introduced to improve multicast communications performance while keeping the compatibility with current devices. The amendment improves the multicast frame transmission reliability by introducing the Group Addressed Transmission Service. This service specifies several retransmission policies and is composed of two different mechanisms: Direct Multicast Service (DMS) and Groupcast with Retries (GCR). In DMS mode each multicast frame is converted into as many unicast frames as the number of receptors in the multicast group. Each unicast frame may be retransmitted as often as necessary until the Access Point (AP) receives the ACK or the retransmission counter reaches its limit. In spite of ensuring high communication reliability, the DMS mode does not scale with the number of receptors in the multicast group.

GCR is a flexible service composed of three retransmissions methods: Legacy Multicast, Unsolicited Retries (UR) and Block ACK (BACK). The Legacy Multicast mode is the one defined in the original IEEE 802.11 standard. The UR policy specifies a number of retry attempts, N, in a manner that a frame is transmitted $N + 1$ times. In this way, the probability of a successful transmission is increased. However, UR may unnecessarily retransmit frames, hence increasing the overall network utilization. In BACK mode the AP reaches an agreement with the multicast receptors about the number of consecutive unacknowledged frames. After that, the AP sends a burst of multicast packets up to that number and requests a Block ACK from each receptor. Both this request and the corresponding ACKs are sent in unicast mode. Despite the control traffic overhead is reduced, also this approach does not scale with the number of receptors in the group. A comprehensive description of the various multicast schemes supported by the 802.11 standard can be found in [4].

Multicast rate selection may be achieved by defining feedback gathering mechanisms allowing the transmitter to gain a better knowledge of the wireless medium status. Leader-based schemes are the most common proposals in the literature. LBP [5] aims at improving multicast communications by enabling ACKs. For this purpose, the receptor exhibiting the worst signal quality is selected as a leader and is in charge of sending ACKs. However, a procedure for the leader selection is not provided. Meanwhile, the operation mode of ARSM [6] is divided into two phases. During the first one, the group leader is selected or updated, whereas in the second step the Signal-to-Noise Ratio (SNR) derived from the leader ACKs is used to adapt the transmission rate. H-ARSM [7] is an evolution of ARSM for hierarchical video transmissions over WLANs. This approach ensures a minimum quality of the video sequence for all the receptors, while those with better channel conditions can receive also the enhancement video layers. The rate adaptation based on the SNR perceived by the worst receiver is also used in SARM [8]. In this scheme the AP identifies the client working under the worst channel conditions by sending beacon frames to which the clients must reply indicating their own SNR. After that, the APs must let the remaining stations know about the new situation, so that if any receptor exhibits a worse channel quality, it must reply to the request. Changes at the client side are needed in order to implement this scheme.

Quality of Experience (QoE) has often been used as basis for rate adaptation in multimedia applications. In [9] a neural network is designed to build a model that maps QoE measurements into MCSes. PSQA [10] is developed as a hybrid objective–subjective metric that simulates how humans perceive impairments to video transmissions. Similar consideration can be made for [11]. In [12] the authors make an effort to address the multicast video delivery using a real–life testbed. In the proposed solution the time is split into a transmission and a polling period. During the transmission period, stations collect the sequence numbers of the received frames. After that, APs gathers that information to calculate the link delivery probabilities. The transmission rate is then selected by comparing this values with the values obtained from the two previous rounds. Changes at the client side are needed in order to implement this scheme. MultiFlow [13] aims to improve multicast communications using SDN principles. However, results are only presented as a numerical analysis and the channel usage of the proposed scheme may exceed the Legacy multicast one when the size of group is greater than a certain threshold.

Multicast is not the only strategy to improve video delivery over wireless networks. For example in [14] dynamic channel switching is used in order to ensure that wireless video streaming takes place over the channel whose condition is most likely to provide good received video quality. Equally important are the evaluation methodology–focused works. For example in [15] the authors evaluate the effectiveness of streaming video over wireless LANs using the H.264 codec. The study concludes that streaming video content over 802.11n is a viable option and that perceptual quality of video is affected by the amount of background traffic and the presence of interfering nodes.

In spite of the improvements made, most of the aforementioned works have either only been tested via simulations or require significant modifications to the wireless client's stack making them incompatible with the IEEE 802.11 standard.

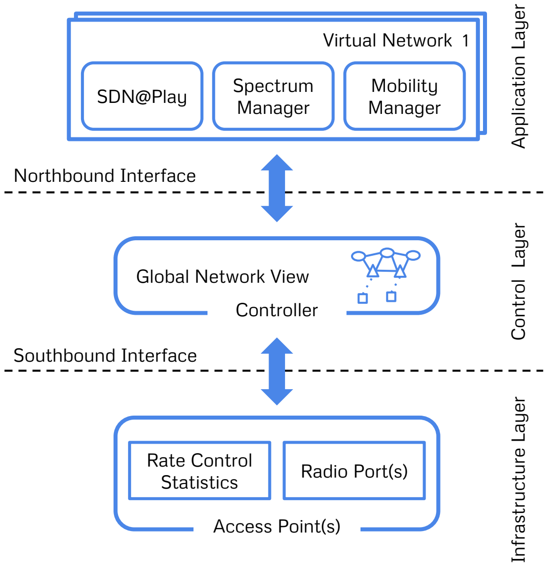


Fig. 1: *SDN@Play* System Architecture.

Conversely, in this work we aim at providing a practical *programmable* multicast rate–adaptation solution that is fully compatible with the IEEE 802.11 standard and that, by being fully software–defined, can be customized to the requirement of the particular multimedia application.

III. *SDN@Play* DESIGN

Current networking technologies have several problems whose solutions often require substantial changes to the network stack. SDN has emerged as a new paradigm capable of addressing such limitations by introducing a fully programmable and modular network, making it possible to implement control and management tasks on top of a (logically) centralized control plane instead of implementing them as distributed applications. Figure 1 depicts the high–level reference system architecture used in this work. As can be seen, it consists of three layers: infrastructure, control and application. The infrastructure layer includes the data–plane network elements (i.e. the 802.11 APs) which are in constant communication with the (logically) centralized controller situated at the control layer. Applications run at the application layer leveraging on the global network view exposed by the controller in order to implement the network intelligence.

As noticed before, OpenFlow is one of the most widely adopted options to implement the link between the data–plane and the control–plane (the so–called southbound interface). Nevertheless, its features are mostly targeted at wired packet switched networks and are poorly suited for controlling wireless networks [2]. As a consequence, in the last years several SDN solutions for wireless and mobile networks have emerged, examples include EmPOWER [2] and Odin [3].

A. The Transmission Policy Abstraction

The fundamentals of SDN call for a clear separation between control–plane and data–plane this in time requires identifying how network resources are exposed (and represented) to software modules written by developers and how those can affect the network state. Due to the stochastic nature of the wireless medium, the physical layer parameters that characterize the radio link between a Wi–Fi AP and a wireless client, such as transmission power, modulation and coding schemes, and MIMO configuration must be adapted in real–time to the actual channel conditions. As a consequence, any programming abstraction for rate–adaptation in Wi–Fi networks must clearly separate fast–control operations that must happen very close to the air interface, such as rate adaptation, from operations with looser latency constrains, such as mobility management.

In this work we proposed the *Transmission Policy* abstraction which allows an SDN controller to reconfigure or replace a certain rate control policy if its optimal operating conditions are not met. More specifically, the *Transmission Policy* specifies the range of parameters the AP can use for its communication with a wireless client. Such parameters include:

- *MCSes*. The set of MCSes that can be used by the rate selection algorithm.
- *RTS/CTS Threshold*. The frame length above which the RTS/CTS handshake must be used.
- *No ACK*. The AP shall not wait for ACKs if true.
- *Multicast policy*. Specifies the multicast policy, which can be Legacy, DMS, or UR.
- *UR Count*. Specifies the number of UR retransmissions.

Transmission Policy configurations can be specified on a L2 unicast destination address basis. As a result, for each destination address and for each AP in the network a specific *Transmission Policy* configuration can be created. Notice how the *Transmission Policy* abstraction allows the controller to specify which set of MCSes can be used by the rate control algorithm implemented by the AP, however the frame–by–frame selection of the MCS is implemented at the AP and not at the controller. We

Table I lists four *Transmission Policy* configurations examples, two for unicast addresses and two for multicast addresses. The first multicast entry (*01:00:5e:b4:21:90*) specifies *Legacy* as multicast mode. This instructs the AP to send every multicast frame with the specified destination address to use 24 Mb/s as transmission rate. We remind the reader that in Legacy mode multicast frames are sent only once and no acknowledgment is generated by receptors. The second multicast entry (*01:00:5e:40:a4:b4*) specifies DMS as multicast mode. In this case for every multicast frame with this destination address, the AP will generate as many unicast frames as the number of receptors in the multicast group. The transmission rate for such unicast frame will be selected by the AP using the list of available MCSes specified by the corresponding unicast *Transmission Policy* configuration.

TABLE I: *Radio Port* Configuration Examples.

Destination	**Type**	**MCS**	**RTS/CTS**	**No ACK**	**Multicast**	**UR Count**
20:47:47:ac:61:5f	unicast	6, 12, 18, 24, 36, 48, 54	2436	False	n.a.	n.a.
5c:e0:c5:ac:b4:a3	unicast	6, 12, 18, 24, 36, 48, 54	2436	False	n.a.	n.a.
01:00:5e:b4:21:90	multicast	24	n.a.	n.a.	Legacy	n.a.
01:00:5e:40:a4:b4	multicast	n.a.	n.a.	n.a.	DMS	n.a.

The content of Table I is manipulated by the controller via the southbound interface using a CRUD (Create, Retrieve, Update, Delete) model. The details of the signaling protocol are omitted due to space constrains.

B. Multicast Rate Adaption

In this section we illustrate of the *Transmission Policy* abstraction can be used to implement the *SDN@Play* multicast rate adaptation mechanism. This algorithm has the goal of intelligently steering the data rate selection for multicast applications toward a more efficient operating point.

The idea behind *SDN@Play* is to use the link delivery statistics collected by the rate adaptation algorithm implemented at the AP and available at the controller to dynamically adapt the MCS used for multicast transmissions in Legacy mode. However, as noticed before, the rate adaptation algorithm is used only for unicast transmissions. As a result, if there are no ongoing unicast transmissions between an AP and a wireless client, no link delivery statistics will be computed. In order to circumvent this issue we introduce a two phases scheme.

During the *first phase*, which represents the smallest percentage of the algorithm time, the controller sets *DMS* as multicast policy for the multicast address M. We remind the reader that in DMS mode multicast transmissions are replaced by as many unicast transmissions as the number of receptors in a group[2]. This allows the rate adaptation algorithm to kick–in and to gather the link delivery statistics for all the receptors in the group. In the *second phase* the controller uses the link delivery statistics collected during the first phase to compute the MCS with the highest successful delivery probability for every receptor in the group. Based on this information, a worst receptor approach is used to compute the optimal transmission MCS R. The controller then sets *Legacy* as multicast policy for the multicast address M and specifies R as single entry in the list of available MCSes for that destination.

The whole process, sketched in Fig. 2, is repeated periodically with a configurable ratio between DMS and Legacy periods. This allows the programmer to trade accuracy for airtime utilization. Specifically, by increasing the fraction of time of the DMS mode it is possible to improve the link delivery ratio at the price of higher channel utilization. Conversely, by increasing the fraction of time of the Legacy mode, the airtime utilization is improved at the price of a possible lower frame delivery ratio (especially if channel conditions are fluctuating).

Based on the aforementioned link delivery statistics, the optimal transmission rate for a given multicast group is calculated by the Wi–Fi AP as follows. Let M be the set of $n = |M|$ multicast receptors in a multicast group and let R_i be set MCS supported by the multicast receptor $i \in M$. If $p_i(r_j)$ is the delivery probability of the MCS index j at the multicast receptor i, we can define the valid multicast group transmission rates R_{valid} as follows:

$$R_{valid} = \bigcap_{i=1}^{n} \{r \in R_i | p_i(r) > r_{th}\} \quad (1)$$

This is the list of MCS indexes with a delivery probability higher than an input threshold r_{th} for all the receptors in the multicast group, i.e. any of those rates would results in a delivery probability of *at least* r_{th}. The optimal multicast transmission rate r_opt is then computed as follows:

$$r_{opt} = \begin{cases} max\,(R_{valid}) & if\ R_{valid} \neq \emptyset \\ min\Big(\bigcap_{i=1}^{n} \underset{r}{\operatorname{argmax}}(p_i(r))\Big) & otherwise \end{cases} \quad (2)$$

This approach ensures that the selected multicast rate has a high delivery probability even for the multicast receptors experiencing bad channel conditions. Notice how if for a receptor there are not MCS indexes whose delivery probability is higher than the input threshold, then our algorithm selects for each receptor the MCS index with the highest delivery probability and from this set picks the lowest MCS index. This is done in order to ensure that the transmission can be decoded by the receptor with the weakest link to the AP.

IV. Implementation details

To demonstrate the usefulness of *SDN@Play* in real-world settings, we implemented it over the EmPOWER platform. In particular: (i) we extended the southbound interface allowing it to collect link delivery ratio statistics; (ii) we extended the data–path implementation in order to properly handle multicast frames; and (iii) we added support for the new *Radio Port* primitive in the EmPOWER SDK.

A. Statistics gathering

The EmPOWER platform, on which *SDN@Play* is based, provides a rich set of programming primitives made available to the programmers trough a Python–based SDK. The list of primitives can be found in [2]. Primitives can operate in either *polling* or *trigger* mode. In the former mode (*polling*) the controller periodically polls one or more APs for a specific information, e.g. the number of packets received by a client. In the latter mode (*trigger*) a thread is created at one or more APs. Such thread is identified by a firing condition, e.g. the

[2]Notice how creation and maintenance of the multicast group is out of the scope of this work.

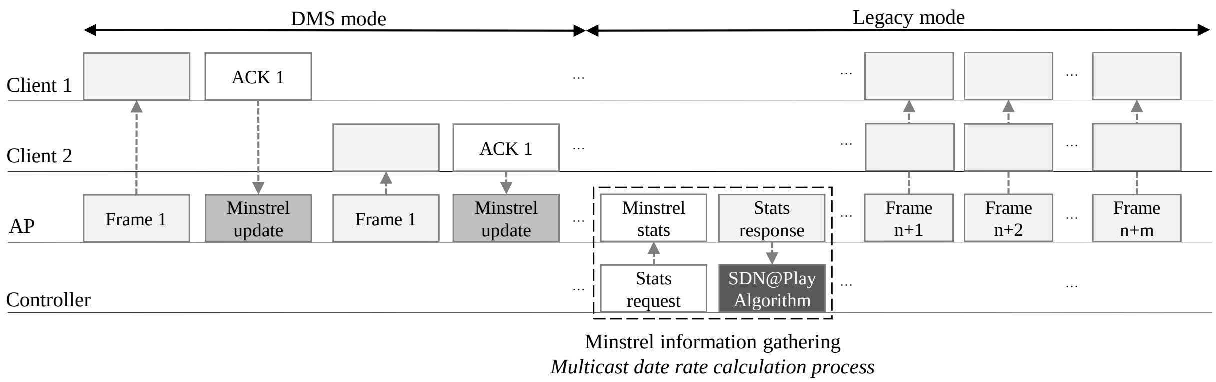


Fig. 2: *SDN@Play*'s two phases scheme. In the *first phase DMS* is used as multicast policy allowing the rate adaptation algorithm to gather link delivery statistics. In the *second phase* the multicast policy is switched to *Legacy* and the collected link delivery statistics are used to compute the optimal multicast MCS.

RSSI of one client going below a certain threshold. When such condition is verified a message is generated by the AP.

In this work we added support for a new *polling*–based primitive allowing the access to the rate adaptation algorithm statistics for a given client. For each supported MCS, the Exponentially Weighted Moving Average (EWMA) probability and the expected throughput in the last observation window are reported. Moreover, the total number of successful and failed transmissions are also reported. This primitive is used by the *SDN@Play* application to gather the link delivery statistics for all the wireless clients involved in multicast transmissions. We remind the reader that this information is maintained by the rate adaptation algorithm implemented by the AP. Therefore, no extra computation is added to the APs logic. More information on the particular rate adaptation algorithm used in our prototype is provided in the next subsection.

B. Data–path Implementation

Each AP consists of two components: one OpenvSwitch [16] instance managing the communication over the wired backhaul; and one Click modular router [17] instance implementing the 802.11 data–path. Click is a framework for writing multi–purpose packet processing engines and is being used to implement just the wireless client/AP frame exchange, while all the network intelligence is implemented at the centralized controller. Communications between Click and the controller take place over a persistent TCP connection (i.e. the southbound interface).

Rate adaptation is also implemented in Click using the Minstrel [18] algorithm (ported to C++ from its Linux Kernel implementation). Minstrel operations follow a multi–rate retry chain model in which four rate–count pairs, $r0/c0$, $r1/c1$, $r2/c2$ and $r3/c3$, are defined. Each pair specifies the rate at which a unicast frame shall be transmitted and a fixed number of retry attempts. Once the packet is successfully transmitted, the remainder of the retry chain is ignored. Otherwise the AP will move to the next pair in the chain. When the last pair has been also tried, the frame is dropped. For each supported MCS, Minstrel tracks the link delivery ratio and the expected packet throughput given the probability of success. Statistics are recomputed every 100ms. In particular the rates with the highest throughput, second highest throughput, and highest delivery probability are maintained by Minstrel.

TABLE II: Minstrel Retry Chain Configuration.

Rate	Look–around		Normal transmission
	Random $<$ Best	Random $>$ Best	
$r0$	Best rate	Random rate	Best rate
$r1$	Random rate	Best rate	Second best rate
$r2$	Best probability	Best probability	Best probability
$r3$	Base rate	Base rate	Base rate

In order to adapt to changes in channel conditions, Minstrel spends part of its time in a so-called *look–around* mode. Specifically, 90% of the time, Minstrel configures the retry chain using the collected link delivery statistics. In the remaining 10% of the time it randomly tries other MCSes to gather statistics. Table II summarizes the criteria used by Minstrel to fill the retry chain in both normal and *look–around* mode.

We extended the Click data–path implementation in order to support generalized transmission policies for unicast, multicast, and broadcast addresses as opposed to the original transmission policies that could be specified only for unicast addresses. According to the new transmission policies, the rate adaptation algorithm (i.e., Minstrel) will use the first entry in the list of available MCSes if the multicast mode is set to Legacy. Conversely, if the multicast mode is set to DMS, the frame will be duplicated for each receptor in the group and will be fed back to the rate control algorithm which will then apply the unicast transmission policy associated to that receptor. Finally, if the multicast mode is set to UR, the frame will be transmitted N times at the specified multicast rate.

C. The Multicast Radio Port Abstraction

The *Radio Port* abstraction is exposed through an object mapping properties to operations. Such an interface allows programmers to fetch the *Radio Port* configuration for a certain address by accessing the `tx_policy` property of a *Resource Block* object. A *Resource Block* is the minimum allocation block in the network and is defined as a 2–tuple $\langle f, b\rangle$, where f and b are, respectively, the center frequency and the band type. For example, the *Resource Block* made available by an 802.11n AP tuned on channel 36 and supporting 40 MHz–wide channels is represented by the tuple $(36, HT40)$. The prefix HT is used to indicate that this band supports the High Throughput MCSes. Each AP has as many *Resource Block*s as the number of available WiFi interfaces.

The following Python listing shows how to access the *Radio Port* configuration for the *04:F0:21:09:F9:96* unicast address:

```
>>>block.tx_policy['04:F0:21:09:F9:96']
(<12,36,48,54>, 2436, False, None, None)
```

As can be seen, the object above contains a single entry mapping a unicast address with a *Radio Port* configuration. In this example, the address *04:F0:21:09:F9:96* has been assigned a configuration specifying which range of parameters the AP can use for its communication with the client (in this case adaptive rates selection will constrain on valid MCSes).

Configuring the *Radio Port* is simply a matter of assigning new values to any of the port properties, for example the following listing sets DMS as transmission policy for the *01:00:5e:00:00:fb* multicast address:

```
>>>txp = block.tx_policies['01:00:5e:00:00:fb']
>>>txp.mcast = TX_MCAST_DMS
```

Similarly, the following listing sets the multicast mode back to Legacy and specifies also a new multicast rate:

```
>>>txp = block.tx_policies["01:00:5e:00:00:fb"]
>>>txp.mcast = TX_MCAST_LEGACY
>>>txp.mcs = [24]
```

The proposed solution allows the specification of flexible transmission policies for each multicast group. As a result, each group is assigned a rate that is calculated considering the conditions of all its MR.

V. PERFORMANCE EVALUATION

The evaluation presented in this section has been carried out in a real environment with the goal of comparing *SDN@Play* with the multicast schemes currently defined in the 802.11 standard, namely Legacy and DMS. In this section we shall first describe the testing environment and the evaluation methodology, then we will discuss the outcomes of the measurements campaign.

A. Evaluation Methodology

The testbed is composed of one AP (W), five multicast receptors (MRx), two wireless background traffic generators ($U1, U2$), one controller (C), one video server (S), and one Ethernet switch (SW). The testbed layout is sketched in Fig. 3.

The AP is based on the PCEngines ALIX 2D (x86) processing board and is equipped with a single WiFi interface

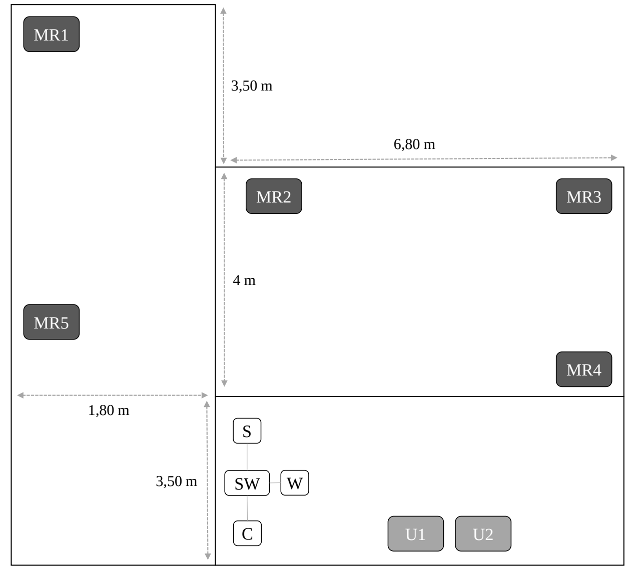


Fig. 3: Testbed deployment layout.

(Atheros AR9220 chipset). The AP runs the OpenWRT Operating System (15.05.01). All experiments are carried out on the 5 GHz band. The controller, the background traffic generators, and the multicast receptors are all Dell laptops equipped with an Intel i7 CPU, 8GB of RAM, and running Ubuntu 16.04.

A variable number of multicast receptors, ranging from 1 to 5, has been used in our measurements. The deployment is depicted in Fig. 3. In order to present a more realistic scenario we have also introduced some artificial background traffic in the network. For this purpose, two stations, defined as *U1* and *U2*, generate a saturated UDP connection addressed at the AP. A multicast video stream is generated by the video server S and delivered to an increasing number of receptors. The video stream consists of a one minute sequence encoded using the High Efficiency Video Coding Standard (HEVC) and transmitted at 1.2 Mbps using FFmpeg [19].

Five different scenarios have been defined in this study: Legacy, DMS, and *SDN@Play*. In the case of *SDN@Play* we considered three configurations, namely: $100/900$, $500/2500$ and $500/4500$. The first number refers to the duration (in ms) of the DMS period while the second one refers to the duration of the Legacy period. As evaluation metrics we considered delivery ratio and wireless channel utilization. Between each measurement the rate adaptation statistics have been cleared. Moreover, apart from the multicast video stream, no downlink traffic exists between AP and receptors. Therefore the only opportunity for the rate adaptation algorithm to be executed is during the DMS periods. Every measurement has been repeated 5 times.

B. Results

Figure 4 plots the average delivery ratio using different multicast strategies. As can be seen, the Legacy multicast strategy provides the highest frame delivery ratio. This is

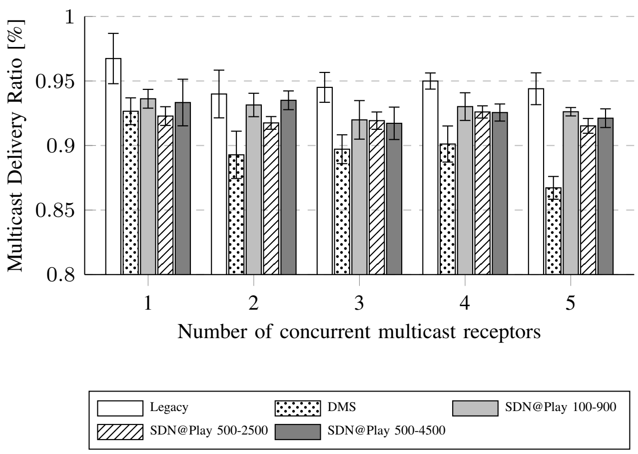


Fig. 4: Delivery ratio for the multicast video transmission Vs an increasing number of multicast receptors.

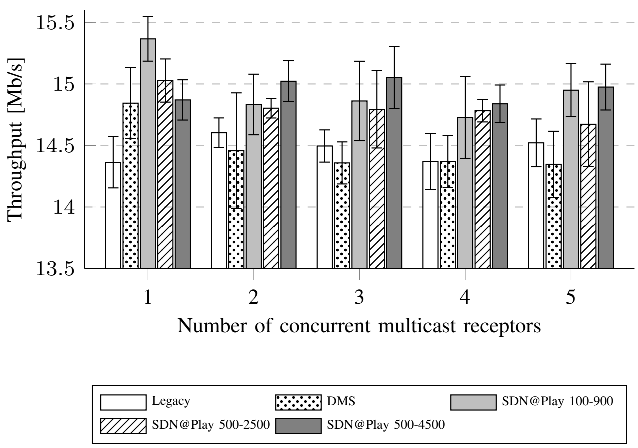


Fig. 5: Background unicast throughput Vs an increasing number of multicast receptors.

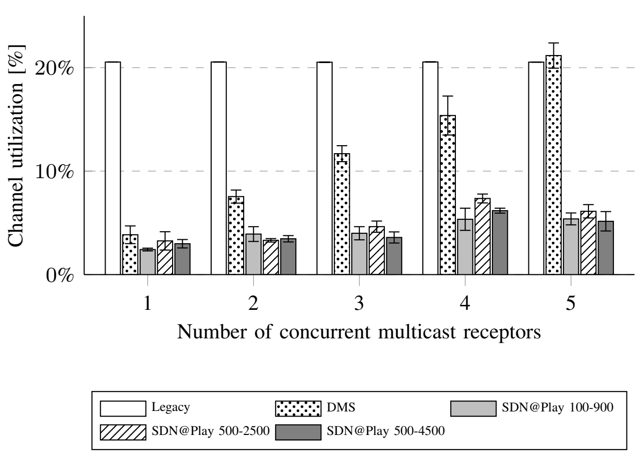


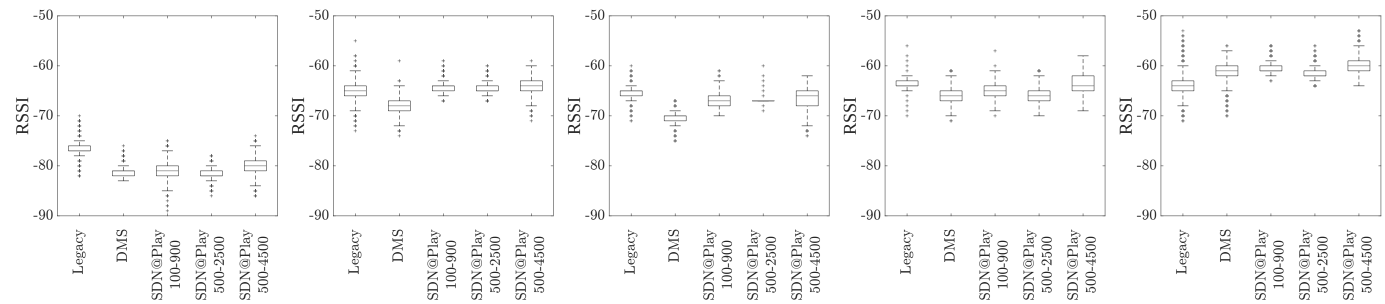
Fig. 6: Channel utilization Vs an increasing number of multicast receptors.

due to to fact that, in Legacy mode, multicast frames are transmitted at the lowest rate (which is usually also the more robust). However, DMS delivers the worst performance due to the fact that DMS converts each multicast stream into as many unicast streams as the number of MRs. Each stream uses ACKs and retransmissions leading to a higher wireless channel utilization. Conversely, the *SDN@Play* scheme delivers in general the same performance level irrespective of the number of receptors. Nevertheless, the overall performance is slightly worse than the one provided by the legacy multicast scheme. The reason for this behavior is that *SDN@Play* tries in general to use a high MCS index for multicast frames which may result in higher packet loss in case of channel quality fluctuations.

Figure 5 plots the average throughput of the two unicast streams for an increasing number of MRs. From this figure we can notice that, when the Legacy mode is used, the unicast streams have the lowest throughput. This is due to the fact that, in Legacy mode, multicast frames are transmitted at the lowest MCS which in time results in less resources being available to the unicast streams. In fact, *SDN@Play* outperforms the unicast throughput by up to 500 kbps in comparison with the standard schemes. Such behavior is more evident in Fig. 6, where the airtime utilized by the multicast stream is plotted. As can be seen, when operating in legacy mode, 20% of the channel resources are used by the multicast stream. It is also interesting to notice that, when 5 multicast receptors are active, the DMS and legacy airtime utilization are approximately the same. On the other hand, the airtime used by *SDN@Play* only marginally increases with the number of receptors. As a result, a reduction in the channel utilization up to 80% is achieved. Figure 8 shows the total traffic associated to the multicast stream in the various scenarios. It is interesting to notice that *SDN@Play* essentially generates as much traffic as the legacy scheme while using only a fraction of the resources.

Finally, Fig. 9 reports the distribution of the MCS used in the case of 5 MRs. The Legacy scheme is omitted because only the lowest MCS is used. As can be seen, in DMS mode almost all transmissions happen at the highest MCS (54 Mb/s). On the contrary, in the three *SDN@Play* scenarios the MCS distribution is significantly different. In particular it can be noticed that using long DMS periods ($500-2500$ and $500-4500$) allows the system to quickly converge on the best MCS. Conversely, with short DMS periods ($100-900$) the rate adaptation algorithm may not have enough time to converge on the optimal MCS, thus the presence in the distribution of low MCS indexes.

In order to provide additional information about the conditions in which the performance evaluation described in this section has been performed, we collected the RSSI values for all the frames decoded by the MRs during the measurements. Figure 7 plots the distribution of the RSSI samples in the case of 5 multicast receptors. As can be seen the RSSI conditions for the 5 multicast schemes are essentially constant across all the multicast receptors. Similar considerations can be made for the other measurements with a smaller number of receptors.



(a) Multicast receptor 1. (b) Multicast receptor 2. (c) Multicast receptor 3. (d) Multicast receptor 4. (e) Multicast receptor 5.

Fig. 7: Signal Strength perceived by the five multicast receptors.

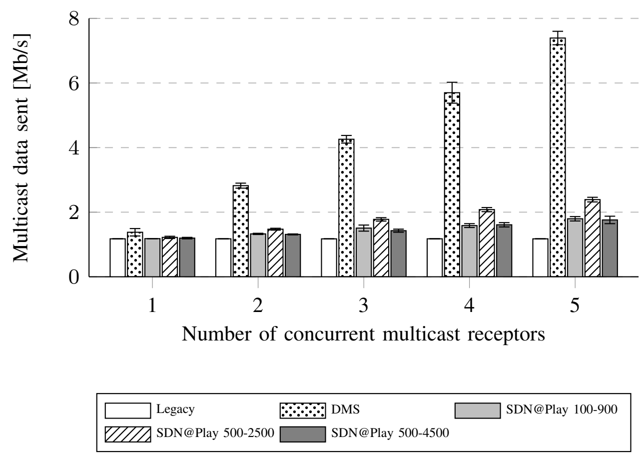


Fig. 8: Multicast traffic Vs an increasing number of receptors.

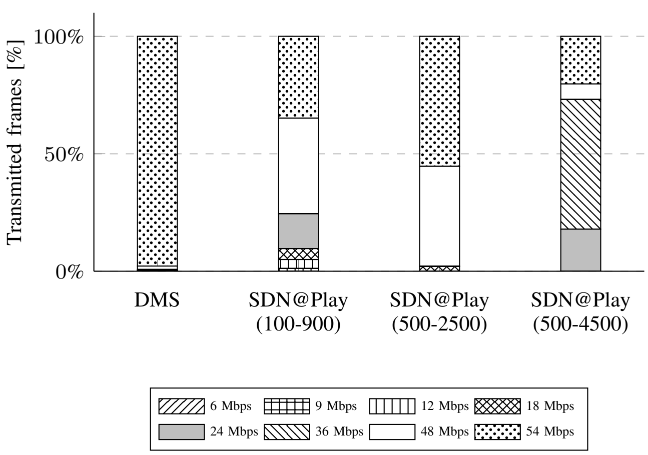


Fig. 9: Distribution of the rates used by each multicast scheme.

VI. Conclusions

In this paper a new SDN–based approach is proposed to adapt the multicast data rate in IEEE 802.11–based WLANs. For this purpose, a two–phase algorithm, named *SDN@Play*, has been designed and implemented. *SDN@Play* coordinates the usage of different retransmission policies allowing the Wi–Fi APs to always use the most efficient multicast transmission rate. An experimental evaluation carried out over a real–world testbed shows that *SDN@Play* can deliver an improvement of up to 80% in terms of channel utilization compared to legacy 802.11 multicast while maintaining full backward compatibility with standard 802.11 wireless terminals.

As future work we plan to extend *SDN@Play* in order to account for multiple multicast groups as well as multiple Wi–Fi APs. Moreover, we also plan to jointly address mobility management and rate adaptation for both unicast and multicast flows. Finally, we are also considering extending the scope of the work to encompass also the wired segment of the network.

Acknowledgment

This work has been supported by the Spanish Ministry of Economy under project TIN2015-66972-C5-2-R, by the European Union's MINECO/FEDER funds under Grant Agreement BES-2013-065457 and H2020 Research and Innovation Action under Grant Agreement H2020-ICT-671639 (COHERENT).

References

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.

[2] R. Riggio, M. K. Marina, J. Schulz-Zander, S. Kuklinski, and T. Rasheed, "Programming abstractions for software-defined wireless networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 2, pp. 146–162, 2015.

[3] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards Programmable Enterprise WLANS with Odin," in *Proc. of ACM HotNets*, New York, 2012.

[4] 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, pp. 1987–2000, 2016.

[5] J. Kuri and S. K. Kasera, "Reliable Multicast in Multi-Access Wireless LANs," in *Proc. of IEEE INFOCOM*, New York, 2001.

[6] J. Villalón, P. Cuenca, Y. S. L. Orozco-Barbosa, and T. Turletti, "ARSM: a cross-layer auto rate selection multicast mechanism for multi-rate wireless LANs," *IET Communications*, vol. 1, no. 5, pp. 893–902, 2007.

[7] J. Villalón, P. Cuenca, Y. S. L. Orozco-Barbosa, and T. Turletti, "Crosslayer Architecture for Adaptive Video Multicast Streaming over Multirate Wireless LANs," *IEEE Journal on Selected Areas in Communications*, pp. 699–711, 2007.

[8] Y. Park, Y. Seok, N. Choi, Y. Choi, and J.-M. Bonnin, "Rate-adaptive multimedia multicasting over IEEE 802.11 wireless LAN," in *Proc. of IEEE CCNC*, Las Vegas, 2006.

[9] K. Piamrat, A. Ksentini, J.-M. Bonnin, and C. Viho, "Q-DRAM: QoE-Based Dynamic Rate Adaptation Mechanism for Multicast in Wireless Networks," in *Proc. of IEEE GLOBECOM*, Honolulu, 2009.

[10] G. Rubino, "Quantifying the Quality of Audio and Video Transmissions over the Internet: the PSQA Approach," *Imperial College Press*, 2005.

[11] C. T. Calafate, M. P. Malumbres, J. Oliver, J. C. Cano, and P. Manzoni, "QoS Support in MANETs: a Modular Architecture Based on the IEEE 802.11e Technology," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 5, pp. 678–692, 2009.

[12] S. Paris, N. Facchi, F. Gringoli, and A. Capone, "An Innovative Rate Adaptation Algorithm for Multicast Transmissions in Wireless LANs," in *Proc. of IEEE VTC–Spring*, Dresden, 2013.

[13] S. Tajik and A. Rostami, "MultiFlow: Enhancing IP Multicast over IEEE 802.11 WLAN," in *Proc. of the IFIP WD*, Valencia, 2013.

[14] E. Sullivan, T. Johnson, and B. Lee, "Dynamic Channel Switching for high-definition peer-to-peer 802.11-based video streaming," in *Proc. of IEEE CCNC*, Las Vegas, 2016.

[15] K. Gatimu, T. Johnson, M. Sinky, J. Zhao, B. Lee, M. Kim, H.-S. Kim, C. gone Kim, and J.-S. Baek, "Evaluation of wireless high definition video transmission using H.264 over WLANs," in *Proc. of IEEE CCNC*, Las Vegas, 2012.

[16] B. Pfaff, J. Pettit, K. Amidon, M. Casado, T. Koponen, and S. Shenker, "Extending Networking into the Virtualization Layer," in *Proc. of ACM Workshop on Hot Topics in Networks*, New York, 2009.

[17] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The Click Modular Router," *ACM Trans. Comput. Syst.*, vol. 18, no. 3, pp. 263–297, 2000.

[18] D. Xia, J. Hart, and Q. Fu, "Evaluation of the Minstrel rate adaptation algorithm in IEEE 802.11g WLANs," in *Proc. of IEEE ICC*, Budapest, 2013.

[19] "FFmpeg project," 2016. [Online]. Available: https://ffmpeg.org/