SDN@Play: A Multicast Rate Adaptation Mechanism for IEEE 802.11 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*—Live multimedia applications have experienced a dramatic growth due to the development of High Definition (HD) contents. Most of these streams are targeted at groups of people, who usually use Wi–Fi networks to access the platforms. In this context, a renewed interest in multicast applications has arisen, which usually suffer low data rates and reliability issues. Software Defined Networking (SDN) has revolutionized the traditional network architecture management, hence allowing for new ways to address the most challenging network constraints. In this paper, *SDN@Play* is presented as an SDN–based algorithm to dynamically adapt the multicast data rate. The implementation over a real–world testbed has proved that this solution is able to outperform the channel occupancy rate of the IEEE 802.11 standard, while keeping the the reliability of the transmission. We release the entire implementation including the controller and the data–path under a permissive license for academic use.**

***Keywords*—SDN, WLANs, 802.11, multicast, rate adaptation.**

I. INTRODUCTION

The success of platforms such as Youtube or Netflix shows the increasing demand for multimedia applications. Many of these services are related to multicast transmissions, where the same information is delivered to a group of users. Live broadcasting events and multiplayer games are good examples of this new trend. Also, wireless access to these contents is becoming more common, the 802.11 WLANs [1] being the most wide spread option. Therefore, ensuring a high performance level is of vital importance. Nevertheless, some multicast issues still remain, such as routing and group management, and are considered even more relevant in the wireless domain.

In WLANs information is transmitted using different rates depending on the modulation and coding scheme (MCS). This process, called rate adaptation, is restricted to unicast transmissions given that it requires to use frame acknowledgments and retransmissions. Conversely, in multicast mode the usage of the lowest MCS aims to guarantee the transmissions reliability and increase the coverage area. This, in turn, involves several implications: i) it considerably impairs the applications performance, ii) the wireless channel is used for long periods, hence limiting the time available for other transmissions, and iii) the reliability of the multicast frames is highly compromised.

Software Defining Networking (SDN) has changed the traditional network management by decoupling the network logic into the control–plane and the data–plane. The introduction of high level programming abstractions allows the shift of the network intelligence to a logically centralized controller. Nevertheless, SDN features are mainly targeted at wired networks and only a few solutions for the wireless domain have recently emerged [2][3].

The contribution of this paper is twofold. Firstly, we introduce a new programming abstraction for multicast communications. Secondly, this abstraction is used to design *SDN@Play*, an SDN–based multicast rate adaptation scheme for 802.11 WLANs. This approach allows the usage of higher multicast data rates while maintaining the same reliability. Moreover, preliminary tests run over a real–world testbed have proved an improvement in the channel utilization with regard to the IEEE 802.11 standard. The controller and the data–path implementation is released under a permissive APACHE 2.0 license[1] for academic use.

The paper is organized as follows. Sec. II provides the technical background and describes some related work. In Sec. III we discuss the design of *SDN@Play*, while in Sec. IV the implementation details are presented. Finally, Sec. V concludes the paper and presents ideas for future work.

II. RELATED WORK AND TECHNICAL BACKGROUND

Multicast transmissions are an efficient way to send the same data to many clients. However, in 802.11 WLANs the lack of acknowledgments and retransmissions results in a decrease in the reliability that force the frames to be sent at the basic rate. Among the efforts made to address these issues, special attention must be given to the IEEE 802.11aa [4] amendment that intends to improve the reliability of multicast transmissions by introducing the Group Addressed Transmission Service (GATS). An indepth analysis of the performance of this service can be found in [5].

GATS is composed of two mechanisms: Direct Multicast Service (DMS) and Groupcast with Retries (GCR). DMS replicates each frame into as many unicast frames as the number of Multicast Receptors (MRs) in a group. In this way, each frame is retransmitted as many times as required until the Access Point (AP) receives the ACK or the retransmission counter reaches its limit. Despite this approach guarantees a high reliability, it presents serious scalability issues. Meanwhile, GCR defines three retransmission methods: Legacy multicast,

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

Unsolicited Retries (UR) and Block ACK (BACK). Legacy multicast is the multicast mode defined in the original IEEE 802.11 standard. UR specifies a number of retry attempts, N, so that a frame is transmitted $N+1$ times. In spite of increasing the successful delivery probability, UR reduces the network performance due to the retransmission of unnecessary frames. In BACK, the AP agrees with the MRs the number of consecutive unacknowledged frames. After that, the AP sends a burst of multicast frames up to that number, and requests the Block ACK to each MR. Both the request and the ACKs are sent in unicast mode. Although the control traffic is reduced, the scalability degree of this scheme is also limited.

Research community has also proposed various works to tackle the multicast rate adaptation problem from numerous points of view. They include feedback gathering mechanisms approaches and Quality of Experience (QoE) based schemes.

In [6] the transmission reliability is improved by enabling ACKs for the group leader, which is selected as the receptor exhibiting the worst signal quality. However, a procedure for the leader selection is not provided. A similar approach using the Signal-to-Noise Ratio (SNR) derived from the leader ACKs to adapt the data rate is performed in [7] and [8]. However, some of these approaches require to modify some packet structures defined in 802.11. Furthermore, QoE has been also taken as the basis for the rate adaptation in multimedia applications. In [9] QoE measurements are mapped into transmission rates using neural networks, while in [10] a hybrid objective-subjective metric is presented. The problem has been also undertaken in real–world testbeds [11] by comparing the sequence numbers of the received frames. Nevertheless, changes in the Linux kernel are needed.

Despite the progresses made, most of the works are not validated in real–world environments or are not compatible with the IEEE 802.11 standard. Therefore, the lack of practical approaches to address the multicast data rate adaptation in Wi–Fi networks becomes highly noticeable.

III. *SDN@Play* Design

SDNs introduce programming abstractions that allow for a fully programmable network and a substantial simplification in the management tasks. Specifically, *SDN@Play* is presented as an SDN–based algorithm that is run on top of a logically centralized controller for the multicast rate adaptation.

A. EmPOWER SDK

SDN@Play has been designed in a manner that facilitates its implementation in any SDN architecture. In particular, in this work the deployment of *SDN@Play* takes as a reference point the EmPOWER architecture already presented in [2].

EmPOWER is an open toolkit for SDN/NFV research in wireless networks, which provides a high–level reference system divided into three layers, as sketched in Fig. 1. The *infrastructure layer* consists on a programmable 802.11 data-path and includes the network devices. The southbound interface enables the communication between these elements and the central controller of the network situated in the *control layer*.

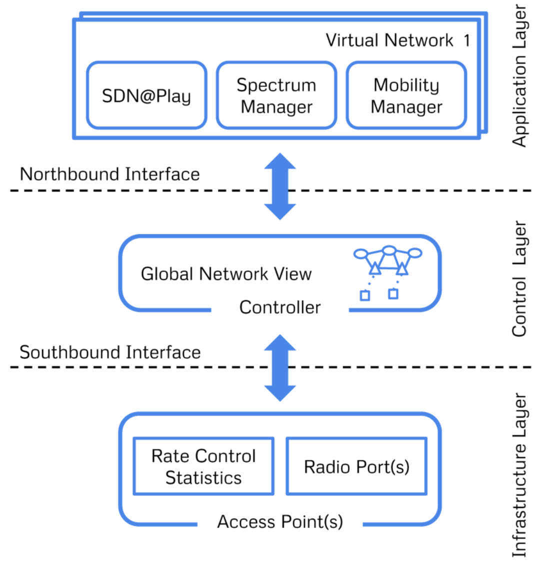

Fig. 1: *SDN@Play* System Architecture

The client state information is called Light Virtual Access Point (LVAP), where details about association, authentication and resource scheduling are stored. Lastly, the *application layer* introduces a Python-based SDK for the development of Networks Apps. These apps are situated on top of the control–plane, therefore having a global view of the network resources that allows it to simplify the network management.

B. The Multicast Radio Port Abstraction

The EmPOWER platform is originally addressed at unicast wireless communications. In this work, the capabilities are extended through the introduction of new abstractions for providing multicast support. Rate control operations are handled in EmPOWER by means of the *Radio Port* abstraction, which allows it to specify a group of parameters for the communication with a given client at a L2 level. In this way, a *Radio Port* is defined for each AP and each client in the network. The parameters already supported are listed below:

- *MCSes*. The set of MCSes that can be selected by the rate adaptation 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.

The *Radio Port* abstraction is extended to allow the configuration for multicast and broadcast addresses. To this end, an additional group of parameters is introduced:

- *Multicast policy*. Specifies the multicast policy, which can be Legacy, DMS, or UR.
- *UR Count*. Specifies the number of UR retransmissions.

Table I presents three *Radio Port* configuration examples for unicast and multicast destination addresses. The first multicast entry (*01:00:5e:b4:21:90*) specifies the usage of *Legacy* as multicast mode, and 24 Mb/s as transmission rate. By contrast, in the second multicast entry (*01:00:5e:40:a4:b4*), the *DMS* mode is selected. Given that *DMS* transmits each frame in

TABLE I: *Radio Port* Configuration Examples

Destination	**Type**	**MCS**	**RTS/CTS**	**No ACK**	**Multicast**	**UR Count**
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.

unicast mode as many times as the number of receptors in the group, the transmission rate is selected from the list of MCSes specified in the *Radio Port* configuration of each receptor.

The *Radio Port* configurations are manipulated by the controller via the southbound interface using a CRUD (Create, Retrieve, Update, Delete) model. Notice that the details of the signaling protocol are omitted due to space constrains.

C. Multicast Rate Adaption

The *SDN@Play* tool aims to use the link delivery statistics estimated by the rate adaptation algorithm implemented in the APs to calculate the most convenient MCS that must be used in the *Legacy* mode of the multicast transmissions.

Rate adaptation algorithms are only available for unicast transmissions. Therefore, the statistical information cannot be gathered if there are no ongoing unicast transmissions between the clients and the AP. Considering these constraints, the working mode of *SDN@Play* is divided into two phases, as depicted in Fig. 2. In the *first phase*, the controller sets *DMS* as multicast policy for a given address M. This allows the APs to gather the statistical information of all the MRs. In the *second phase* the previous statistics are used to compute the MCS with the highest delivery probability, R, for all the MR in the group. Then, the *Legacy* mode is set as retransmission policy for the multicast address M, and R is configured as single entry in the set of MCSes for that destination.

This process is repeated periodically with a configurable ratio between the duration of the *DMS* and *Legacy* periods. This makes it possible to trade accuracy for airtime utilization. Specifically, increasing the amount of time in which *DMS* is used leads to an improvement in the link delivery ratio at the price of increasing the channel utilization ratio. Conversely, increasing the fraction of time of the *Legacy* mode leads to an enhancement in the airtime utilization at the price of a possible lower frame delivery ratio.

IV. Implementation details

To show the features of *SDN@Play* in real–world environments, it has been implemented over the EmPOWER platform. The challenges addressed are the following: (i) we extended the southbound interface to allow the link delivery statistics gathering; (ii) we improve the data–path implementation in order to properly handle multicast frames; and (iii) we implement the new *Radio Port* abstraction in the EmPOWER SDK.

A. Statistics gathering

The EmPOWER platform provides a full set of programming primitives for the network management trough a Python–based SDK [2]. These primitives can be used in *polling* or *trigger* mode. The *polling* mode allows the controller to periodically poll the APs for specific information, while in the *trigger* mode this information is sent by the APs to the controller when the firing condition is verified.

In this work, a new *polling*–based primitive is presented for the collection of the rate adaptation algorithm statistics for a given client. This information includes for each supported MCS, the average delivery probability, the expected throughput, and the number of successful and failed transmissions. In particular, *SDN@Play* uses this primitive periodically to gather and update the statistics related to all the MRs in a group.

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

B. Data–path Implementation

APs are composed of one OpenvSwitch [12] instance for the wired backhaul and one Click modular router instance for the 802.11 data–path implementation. In this work, Click is used to handle the clients/APs frame exchange, while the remaining network intelligence is managed by the controller. The controller communicates with Click via the southbound interface through a persistent TCP connection.

Rate adaptation mechanism is implemented in Click using the Minstrel [13] algorithm. Minstrel follows a multi–rate retry chain model in which four rate–count pairs, $r0/c0$, $r1/c1$, $r2/c2$ and $r3/c3$ are defined, as shown in Table II. They specify the rate that must be used to transmit a given number of retry attempts. If a frame is successfully transmitted, the remaining part of the retry chain is ignored. Otherwise, the next pair is used until the frame is properly transmitted or is finally dropped. In order to adapt to changing channel conditions, the statistics are recomputed every 100ms. Moreover, Minstrel divides the time into two parts: it spends the 90% of the time using the collected link delivery statistics to configure the retry chain, while in the remaining 10% of the time, other MCSes are randomly selected to gather new statistics.

The data–path is extended to provide unicast, multicast and broadcast support. For a multicast address, Minstrel will use the first MCS in the list if the retransmission mode is set to *Legacy*. If the policy is set to *DMS*, the entry is ignored and the policy associated to each MR is used instead. Finally, if UR is selected, the frame is sent N times at the specified rate.

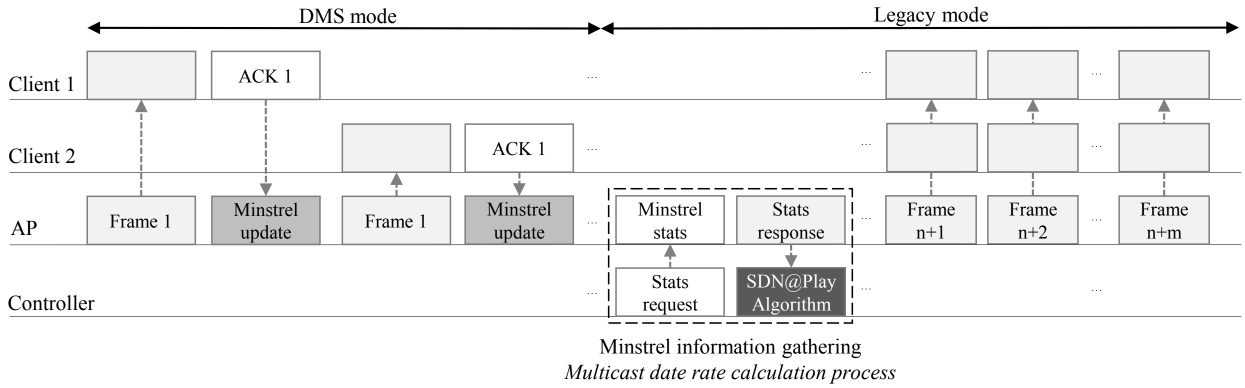

Fig. 2: *SDN@Play*'s scheme. In the *first phase DMS* is used as multicast policy allowing the link delivery statistics gathering. In the *second phase* the policy is switch to *Legacy* and the collected statistics are used to compute the optimal multicast MCS

C. The Multicast Radio Port Abstraction

The *Radio Port* configuration is set through 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. Therefore, the controller handles as many *Resource Block*s as the number of Wi–Fi interfaces for each AP.

The Multicast *Radio Port* configuration only requires to specify the information for the MCS and multicast policy. The following example shows the configuration needed to set the *DMS* retransmission policy for the *01:00:5e:00:00:fb* address:

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

In a similar manner, the `tx_policy` can be reset to the *Legacy* mode, where the new multicast rate is also defined:

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

V. Conclusions and Future work

This paper proposes a new SDN–based scheme to adapt the multicast data rate in 802.11 WLANs. *SDN@Play* is presented as a two–phase algorithm that coordinates the use of different multicast retransmission policies to calculate the optimum data rate for each AP at any moment. Preliminary tests have proved that the algorithm outperforms the IEEE 802.11 standard in terms of channel utilization while preserving the performance. Moreover, this approach has been implemented over a real–world testbed and is fully compatible with 802.11 devices.

As future work we aim to extend the *SDN@Play* capabilities in order to support multiple multicast groups and multiple APs. Additional works also include rate adaptation and mobility management for both unicast and multicast flows.

Acknowledgment

This work has been jointly supported by the Spanish Ministry of Economy under project TIN2015-66972-C5-2-R and 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] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, ANSI/IEEE Std 802.11, LAN/MAN Standards Committee of the IEEE Computer Society Std., 2012.

[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] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Amendment 3: MAC Enchancements for Robust Audio Video Streaming*, ANSI/IEEE Std 802.11aa, LAN/MAN Standards Committee of the IEEE Computer Society Std., 2011.

[5] 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.

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

[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] 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.

[12] 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.

[13] 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.