A Proposal for Dynamic Frequency Sharing in Wireless Networks

Luis Lopes[ϕ], Rute Sofia[ϕ], Huseyin Haci[λ], Huiling Zhu[Ω]
[ϕ] COPELABS, University Lusófona, Lisboa, Portugal
$\{luis.amaral, rute.sofia\}$ $@ulusofona.pt$
[λ] Department of Electrical and Electronic Engineering, Near East University, Nicosia, Cyprus
$huseyin.haci@neu.edu.tr$
[Ω] University of Kent,
$h.zhu@kent.ac.uk$

***Abstract*—Wireless networks are today employed as complementary access technology, implemented on the last hop towards the Internet end-user. The shared media that wireless deployments provide and which is relevant to interconnect multiple users, has a limited technical design, as only one device can be served per unit of time, design aspect which limits the potential applicability of wireless in dense environments. This paper proposes and evaluates a novel MAC Layer mechanism that extends current wireless networks with the possibility to perform downstream transmission to multiple devices within a single transmission time-frame, resulting in improved fairness for all devices. The mechanism, which is software defined, is backward compatible with current wireless standards and does not require any hardware changes. The solution has been validated in a realistic testbed and the paper provides details concerning the computational aspects of our solution; a description of the implementation; results extracted under different realistic scenarios in terms of throughput, packet loss, as well as jitter.**

I. Introduction

The new myriad of Internet services where the Internet end-user shares information associated with software defined networking introduced the need and the power for autonomous wireless architectures to grow based on the user willingness to share any form of service, including networking services such as Internet access. These new architectural paradigms, which have been addressed under the scope of user-centric networking [1], [2], introduced an additional need to re-think the MAC Layer design, not necessarily requiring changes to hardware or to existing standards.

For instance, *Wireless Fidelity (Wi-Fi)* as a technology that complements the Internet access worldwide, faces limitations in terms of fair sharing of the available spectrum. In Wi-Fi when one device (a *station*) triggers communication, it prevents the others of communicating during a specific time-frame – *contention* is the process applied to deal with collisions within the context of the shared medium.

To circumvent such problems, several techniques applied to OSI Layer 1 have been devised. Redesigning aspects of OSI Layer 1 imply hardware changes. In contrast, the *Dynamic Frequency Sharing (DFS)* approach which is the object of this publication is software-defined and operates solely on OSI Layer 2 - therefore, it does not require any hardware changes, or standard changes. DFS has been designed to adjust the MAC Layer operation only *downstream,* i.e. from a controller (antenna) to multiple stations.

Allowing downstream transmission to multiple stations provides the means to improve the performance of current solutions threefold. Firstly, by allowing data to be transmitted in the same time-frame to multiple stations, the control overhead is reduced in comparison to the current standards, as the same control information is used to transmit data to multiple stations. Secondly, for real-time traffic there is an upper bound on usable data rates. Thirdly, instead of transmitting to stations one by one, thus wasting time in particular, if the first station that captures the medium is what is known as “slow” station (e.g. away from the antenna or attaining severe interference around), our solution provides a way to transmit “simultaneously” data to several stations, where simultaneously refers to transmission within a same time frame, thus decreasing round-trip time delay and the latency of the transmission.

The paper is organized as follows. Section II goes over related literature explaining how our vision relates with such work, and our contributions. Section III describes our solution, namely, the conceptual and computational aspects; the specification; details concerning implementation of DFS. The DFS performance evaluation is then described in section IV, based on a local realistic testbed. In order to further understand the behavior of our approach with an increasing number of stations, extrapolation has been applied to analyze the performance trend. The paper is concluded in section V, where a few guidelines concerning future research are also provided.

II. Related Work

Resource management in wireless networks is a topic that has been one of the most relevant aspects of to *Quality of Service (QoS)* research in wireless networks during the last decade. Some QoS schemes consider ways to ensure fairness in a number of aspects, for instance, the capability of the network to serve more users at an instant in time [3], or the application of static or dynamic threshold models and priorities to provide fairness in terms of network utility, e.g. throughput [4].

Pong et al. provide an analysis of the trade-off between fairness and capacity in the context of *Wireless Local Area*

Networks (WLANs), for scenarios with interference [5]. Their work explores fairness in terms of throughput as a measure of network utility, and allowed transmission time, explaining how different fairness parameters impact on the capacity of the link. Pricing model approaches [6], [7] are applied to ensure fairness, again in terms of network utility, but considering all of the potential network stakeholders. Game theory is also applied as a way to assist a better notion of fairness in wireless networks [8].

A second line of work focuses on the notion of simultaneous transmission in WLANs [9], [10], [11], [12]. Such line of work assumes that an underlying system is already established and provides some techniques for performance improvement. This category of work requires hardware changes, while ours is software-based.

From a pure OSI Layer 1 perspective, some architectures have emerged, attempting to solve the identified gaps. The *Fine-grained Channel Access in Wireless (FICA)* [13] has been proposed as a new physical layer architecture, based on *Orthogonal Frequecy-Division Multiplexing (OFDM)*. FICA relies on *Distributed Inter-Frame Space (DIFS)* to provide dynamic sub-frequency allocation. In contrast, DFS models the sub-frequency based on a parameter and does not require specific coordination mechanisms other than the ones embedded currently in the MAC Layer. The parameter we consider as basis for fairness is *trust* [14], even though any other Quality of Experience (QoE) or QoS parameter can be considered instead.

The *Frequency-Aware Rate Adaptation (FARA)* approach [15], considers a centralized control approach based also on the physical layer and therefore, requiring modifications to current hardware.

Still focusing on OSI Layer 1, coding and diversity techniques further assist in addressing the optimization of the shared medium. For instance, Tan et al. propose a fine-grained channel access method [16] which recurs to the coordination mechanisms and carrier-sensing of OSI Layer 2, coupled with a frequency-domain contention and back-off to efficiently coordinate sub-channel allocation. The authors show significant improvements towards current standards. This work, albeit relevant, requires a new design of the physical layer, as well as changes to OSI Layer 2. In contrast, DFS simply integrates a new way to interpret coding received from OSI Layer 1, also applying MAC frame aggregation and reinterpretation techniques, a solution usually successfully applied to keep backward compatibility.

A third line of work that we believe is relevant to cite is adaptive rate modulation as this technique has held the best results in the face of diversity [17].

Frame aggregation [18] and frame reinterpretation techniques are already integrated into IEEE 802.11n and 802.11e standards. This solution aims at improving throughput by sending two or more clustered/aggregated data frames in a single downstream transmission period, directed still to a single station/user. Increasing the volume of useful data with respect to the overhead, at a transmission period, makes the communication more efficient. Frame aggregation has been applied to allow the MAC Layer to cope with new designs of the physical layer [19], [20]. Our work recurs to frame aggregation techniques to further assist the multi-user coding. In contrast to previous work, our proposal is fully backward compatible with current standards.

In the context of software defined MAC Layer frequency sharing approaches, it is worth to mention the work of Zarakovitis et al. [21] that proposes a scheme to adjust power and data transmission rates across subcarriers with resilience to channel errors. Such scheme, coupled to a statistical queuing model that express delay for each station, result in a joint power and subcarrier allocation policy with confirmed superior performance in comparison to cross-layer approaches. A second work by the same authors [22] describes a scheduling framework for joint channel and power allocation in orthogonal frequency division multiple access cognitive radio (CR) systems which, based on cooperative scheduling approaches (Nash bargaining) results in improved performance in terms of max-min fairness, and optimal capability.

Overall, the differentiators concerning our work are: i) no hardware changes are required to support DFS; ii) DFS targets any narrow-band wireless technology which considers OFDM, and it supports downstream transmission to multiple stations within the same time-frame; DFS is backward compatible with current IEEE 802.11 standards that recur to OFDM.

III. Dynamic Frequency Sharing

A. Background and Terminology

This sub-section introduces background notions and pointers to assist the reader in understanding the design of our solution, and its backward compatibility aspects.

802.11b uses a direct sequence spread spectrum technique, *Complementary Coded Keying (CCK)*, where the bit stream is processed with a special coding and then modulated using *Quadrature Phase Shift Keying (QPSK)*. IEEE 802.11a/g employs OFDM instead (52 channels). Out of the 52 OFDM sub-carriers, 48 are for data and 4 are "pilot subcarriers".

In terms of non-overlapping channels, and relying on the example of a 2.4 GHz wireless network, if one assumes the usual 20MHz channel width, then this means that the main frequency can be split into 4 different non-overlapping channels. Then, countries apply specific regulation to define how much power, and which channels are allowed. Due to these constrains, and to the way OFDM works, when a device grabs the medium, it will only stop once transmission has ended, independently of the signal conditions. For instance, if one station already transmitting has several neighbor stations around that are actually in better conditions to transmit to an antenna, it will keep on transmitting, and the others will have to wait. Therefore, even though there are theoretically 48 sub-frequencies to carry data, currently those carriers can only be applied at a time between a station and an antenna.

From an OSI Layer 1 perspective, data is assigned to specific OFDM *symbols*. Then, between a transmitter and a receiver, only one symbol can be transported per unit of time.

In terms of terminology, throughout the next sections we refer to the device that takes care of controlling the communication between multiple stations as *antenna* or *controller*. In Wi-Fi, the antenna is therefore the *Access Point (AP)*.

A *station* is a device served by a controller. Stations communicate via a specific channel, controlled and served by the controller.

Data transmission *upstream* refers to transmission performed from stations to a controller. While data transmission *downstream* refers to transmission performed from a controller to stations.

B. Overview

DFS [23] follows the recent trend concerning frequency assignment and sub-division which argues that the channel width of nodes should be adaptive. After reviewing state-of-the-art [24], [25], we have identified two major persistent drawbacks in adaptive channel approaches: the coordination complexity intrinsic to the per-node channel width adaptation, and the periodic computation of NP-hard problems. During prior work we have designed DFS to employ an alternative way of arranging wireless channel assignments based on OFDM and yet applicable in real systems. DFS considers adaptive multi-user access, modulation, error coding and power allocation techniques to balance the tradeoff of cost for performance gain. Therefore, DFS is capable of assigning a specific subset (*frequency chunk*) of sub-carriers to each station during a controller duty time frame. In other words, in the presence of multiple stations requesting data transmission, the channel will adjust to the number of stations, based on specific network or user policies.

In regards to data transmission and from a frequency perspective, the resources to be managed are: the frequency spectrum (i.e. sub-channels); number of bits to be transmitted (i.e. modulation level); transmit power. The joint management of these three resources can be seen as a mixed integer optimization problem as it integrates both integer (e.g. assignment of a sub-channel to a station or not) and real valued parameters (e.g. allocated transmit power). Solving it in a realistic way, interoperable with the current MAC Layer design, is a highly novel feature that DFS brings into the MAC Layer of 802.11 OFDM based standards.

In order to allow several stations to profit with greater fairness of the wireless medium access the frequency sub-carriers are grouped into several sets (*frequency chunks*) which are assigned to stations requesting the medium. Frequency chunking in DFS is performed via two stages. At the first stage DFS does frequency spectrum allocation and in the second stage it performs rate adaptation (bit allocation), which can significantly reduce the computational complexity.

The medium access control is performed in DFS by recurring to MAC virtualization techniques in the controller, as shall be explained in section IV-A. Hence, each station is assigned by its controller with a virtual MAC interface which in reality provides only a frequency chunk to that specific station. The downstream transmission to multiple stations is performed by recurring to frame aggregation, where DFS creates a *superframe* that integrates pieces of MAC data frames directed to different stations, as shall be explained in the next section.

In the next sub-section we shall explain into detail the three functional blocks of DFS: rate adaptation; frequency chunk assignment; frame aggregation.

C. DFS Functional Blocks

Rate Adaptation: Rate adaptation provides adaptive modulation per atomic transmission period for the frequency chunking process described next in this sub-section. Resource allocation depends on channel quality as well as on the data rate that can be provided to each station. This rate is dependent on the number of stations being served at a specific instant in time by a controller as well as on the processing cost of such computation. In each atomic transmission period, a controller relies on regular rate adaptation scheduling to decide on which data rate to apply to a specific station. Based on such data rate, a specific encoding scheme is applied accordingly with OFDM rules that are presented via Table I, to derive the modulation level m that the controller shall employ.

Hence, the first step in rate adaptation is to obtain e.g. from a call admission control mechanism, the possible data rate level for the atomic transmission period. Then, the second step is to select a possible modulation level m, and the third step is to provide that m level to the frequency chunk assignment process.

Frequency Chunk Assignment: The main aim of Frequency Chunk Assignment is to enable simultaneous transmission of nodes data over the frequency carrier. This is achieved by dividing the broadband carrier into narrower frequency-chunks and allocating these chunks to stations being served "simultaneously".

It has been shown that by allocating a single sub-channel to the station which has the best channel conditions, the best performance of the system can be achieved [26]. However, when the number of sub-carriers becomes high, sub-channel based allocation may result in significant overhead. One way of reducing the complexity and overhead derived from the application of sub-channel allocation schemes, is to explore the correlation between neighboring sub-channels in OFDM. A simplified resource allocation scheme, chunk based, can be adopted by properly grouping set of adjacent sub channels into a chunk as explained in our prior work [24], where it has been demonstrated that the performance of chunk-based allocation can approach to the performance of sub-channel based allocation when the chunk size is set properly.

Parameter I: index of the next station to serve: The first aspect to tackle in terms of frequency chunk allocation is how and which stations to match to specific frequency chunks, from the perspective of a controller. In the controller, we assume that there are N virtual interfaces and k stations to be served. Eq. 1 provides us with index J_n for the next station i to be served via interface n of the controller.

$$J_n = \arg\max\{t_i\}, 0 \leq i \leq k \quad (1)$$

Table I: Encoding details, different data rates and OFDM.

Speed (Mbps)	Modulation and coding rate (R)	Coded bits per carrier	Coded bits per symbol	Data bits per symbol
6	BPSK, R=1/2	1	48	24
9	BPSK, R=3/4	1	48	36
12	QPSK, R=1/2	2	96	48
18	QPSK, R=3/4	2	96	72
24	16-QAM, R=1/2	4	192	96
36	16-QAM, R=3/4	4	192	144
48	64-QAM, R=2/3	6	288	192
54	64-QAM, R=3/4	6	288	216
72	64-QAM	6	288	288

In DFS[1] is presented in section IV-A, we apply a novel approach where the station is entitled to resources based on the reputation that its user acquires due to sharing resources. This cost is measured in tokens t_i: the serving priority of station i increases in comparison to others with an increase in the reputation level (due to e.g. more sharing of resources) [14].

Parameter 2: Width of each chunk to assign: The second parameter concerning frequency chunk assignment relates with the size of the frequency chunk to assign to station i. This parameter affects the number of bits that will be carried within each transmission symbol for corresponding stations. The more tokens t_i a station i provides to access a specific service, the higher the priority it has in getting such service and the greater the size $fc_{n,i}$ of the frequency chunk n to be allocated to station i. We compute such width via Eq. 2, where $IFFT_{size}$ corresponds to 48 symbols (IEEE 802.11 standards, OFDM based), and where m correspond to the coded bits per carrier, dependent upon the selected modulation and coding rate, as shown in Table I.

$$fc_{n,i} = \frac{t_i}{\sum_{j=0}^{N} t_j} * IFFT_{size} * m \tag{2}$$

Selecting the Size of the Superframe to be sent to multiple stations: The frequency chunk assigned has then to be materialized into the multiple station MAC frame which a controller transmits to multiple stations. This frame, which we name *superframe*, is therefore composed of chunks $fc_{n,i}$ to be delivered to a station i.

The constraint for the superframe size relates with backward compatibility, from the definition of 802.11 MAC standards, where maximum length of a MAC frame, *max_payload*, is set up to be 2312 bytes.

The number of chunks to be transmitted to station i, $n_{b,i}$, within a single superframe is provided via Eq. 3.

$$n_{b,i} = \frac{maxpayload}{fc_{n,i}} \tag{3}$$

Frame Aggregation: The two previous DFS blocks, rate adaptation and frequency chunk assignment, are blocks that belong, from a protocolar perspective, to the lower level of the OSI MAC Layer. Frame aggregation is performed on the upper level of the OSI MAC Layer. This function block of DFS arranges data to be transmitted to a specific station to be placed in a virtual queue, based on MAC Layer virtualization techniques. Such queue is associated with the identifier i of a specific station, and contains several MAC frames, for which pieces of size $fc_{n,i}$ shall be added to a MAC frame built to allow simultaneous transmission to multiple stations. When building such frame, the controller relies on a specific scheduling mechanism to serve the different queues assigned to different stations, e.g. based on a specific priority scheme that the controller integrates.

Assuming that a queue holds more data than a MAC frame can carry then regular IEEE 802.11 fragmentation schemes are applied to generate a new frame. Thus, several MAC frames may be created in order to transmit the whole data set in a virtual queue for a station.

Frame aggregation starts with the assignment of stations to virtual MAC interfaces (*vif*), as the MAC layer requires such step to take place before the physical layer can allocate frequency chunks of size $fc_{n,i}$ to stations. Such assignment is a result of an applied scheduling mechanism which takes into consideration the utility function that generates index J_n, as explained in the beginning of this sub-section. The next step in frame aggregation is the usual MAC frame creation process. The payload of the frame is built based on common MAC concatenation techniques. A key difference in DFS, as explained next, is that each data block to be delivered to a specific station i has 3 initial bits reserved to the DFS identification of the station.

A fourth step in DFS frame aggregation takes care of creating the MAC header. This header is fully compatible with 802.11 formats. We rely on the “Type” and “Subtype” sub-fields of the *Frame Control* field, i.e. *TYPE_DATA+CF-Poll* to allow stations to understand, when they receive superframes, that they need to look for interpolated blocks of data.

Step five is dedicated to the creation of a special *Frame Check Sequence (FCS)*. The final and sixth step is to send the created frame to the PHY layer for transmission.

D. Specification

This section is dedicated to the specification of DFS based on IEEE 802.11g in infrastructure mode. DFS has been specified as an extension of IEEE 802.11 which resides both on a controller, and on a station. On the controller side, DFS performs the operation to allow downstream data transmission to multiple stations “simultaneously”. While on the station side, DFS is engineered to receive and interpret a superframe adequately.

[1]DFS has been conceived, specified, implemented and validated in the context of the European project ULOOP, which considered trust as a QoE parameter relevant to assist resource management in wireless networks.

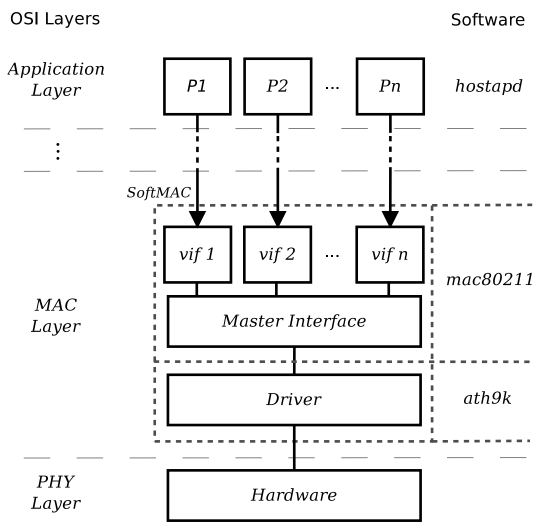


Figure 1: High Level scheme of DFS, controller side.

The Controller Side: Fig. 1 illustrates a high level implementation view for DFS operating on the controller within the context of Wi-Fi infrastructure mode. The figure holds on the left-hand side a high level representation of the OSI stack layers involved in the DFS mechanism. On the right hand side, the terms correspond to the software packages that we have considered in the implementation.

One controller is expected to serve multiple stations, here referred to as *s1, ..., sn*. Each time a new station requests resources from a controller, if entitled to resources, then the controller shall map the station DFS identifier to a specific virtual queue, represented by the virtual interface vif_n. The master interface is a MAC interface that manages the virtual queues and sends the MAC frames to the driver for transmission. The stations may then be served by their controller according to their priority in the system – the highest priority stations get served first and may get greater amount of resources than other low priority stations.

The controller periodically serves the queues assigned to each station, and creates the MAC frames (superframes) to be sent to all active stations *s1, ..., sn*. This is not a new type of MAC frame; instead, we simply consider a new way to interpret the MAC frame payload. Hence, stations that implement DFS interpret the modified payload by extracting only the bits that are assigned to them; regular IEEE 802.11 stations will simply discard that content, following the regular MAC Layer procedure.

Fig. 2 provides the flow-chart of the controller side, where we highlight in dark grey the DFS part, in contrast to the regular MAC Layer architecture (boxes in white).

Station Authorization: After scanning, station A realizes that AP is available and therefore starts the MAC Layer authorization and authentication process. On the controller side, during this process, the controller obtains feedback from the local call admission control process (1). In case there are available resources, the controller creates an identifier for the station (2) and provides the identifier via the usual MAC Layer procedures (3). The controller also creates a virtual interface and assign it to the station (4).

Handling of MAC Frames: When the controller gets a MAC frame to a specific destination, the controller checks the MAC destination address of the station (5) to understand whether or not there is a virtual interface associated to that particular station. If there is yet no such association (which means that the station is not DFS-enabled) then the frame is sent as usual (6).

Periodically and in background DFS checks the queues of the virtual interfaces (7). Assuming a controller is only serving one station (1 virtual queue active) then the frames are sent as usual (6). However, if the controller has more than one active queue, the process of creating a superframe is triggered (8). Sending of superframes is processed as usual by the MAC Layer, recurring to fragmentation if required (9).

Superframe Creation: The superframe process is illustrated in Fig. 3. As explained in section III-C, a superframe is a concatenated MAC frame, containing different blocks of data, *v_packets*. V_packets first 3 bits identify the station in the context of DFS. Moreover, the usual *End of Data (EOD)* flag is used to indicate to the station that the next piece of data is the last one, i.e. EOD = 1, or otherwise EOD = 0, and the length field is updated with the length of this piece of data. If it turns out that the sum of all of the v_packets size is higher than the maximum MAC frame payload length allowed, fragmentation will be applied, accordingly with IEEE 802.11 fragmentation rules for broadcast frames.

Station Side: Fig. 4 provides the flow-chart for the station-side operation.

During its regular MAC Layer operation, stations periodically check frames received. Superframes are of type broadcast and therefore, stations simply analyze corresponding flag inside the type field of the MAC header (1). If the received frame is a superframe (2), the station analyzes the payload sequentially, looking for data chunks that have its own identifier, computed during DFS initialization (3), as illustrated in Fig. 5. Therefore, chunks that have as identifier another station are discarded once the station reaches the end of the superframe. Each time a station finds a block assigned to itself it also checks if such block is the last one or not, as well as the length of data to read. If this is not the last piece of data, it continues reading its controls bits fields and data until finding the EOD flag set or by reaching the end of the payload.

All blocks are placed in a new, internal MAC frame. Once such frame payload is built, the station adds the MAC header and passes it up to the kernel.

If a station reaches the end of a payload without finding the flag EOD set, the station must wait for a next fragment which should arrive in a new superframe. In this case, DFS applies 802.11 de-fragmentation to re-create the original frame. To achieve this, instead of creating the complete frame, it creates a fragment of the original frame. This is done by setting on the flag "More frags" and also by setting an index of the fragment, in the sequence control field at MAC header. When the station receives a superframe fragment that carries the last block sent to the station, the same process done on the others fragments is done as well; however, the flag "More frags" is set off, since this is the

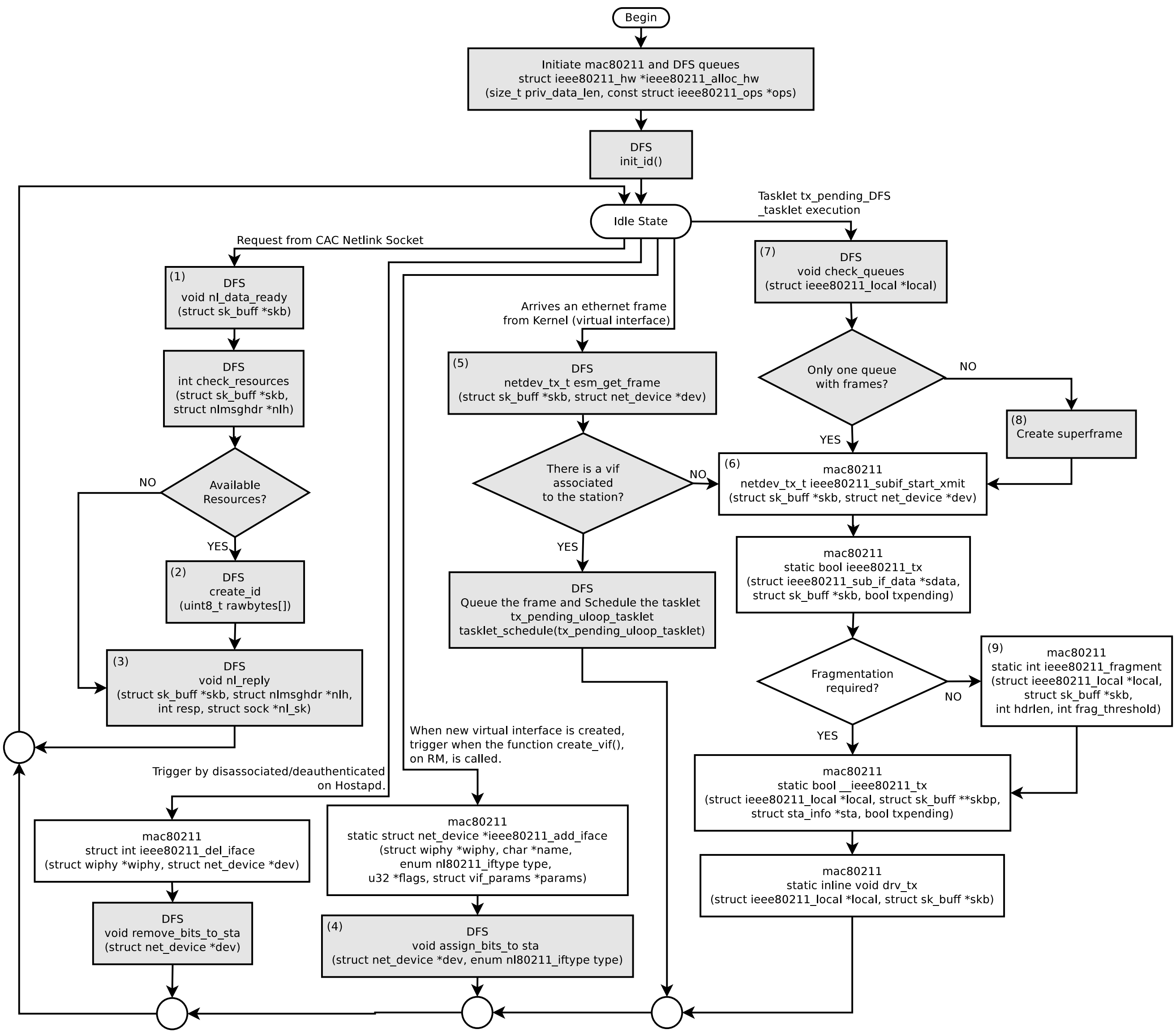


Figure 2: Controller flow-chart, MAC layer.

last fragment. Queuing and treatment of the MAC frames is then process by regular IEEE 802.11 (4).

IV. Performance Evaluation

A. Implementation Details

DFS has been implemented in the context of the European project ULOOP [2]. DFS software is available as a mac80211 patch, and the DFS software available under LGPLv3.0 [27]. The provided implementation is an extension of the mac80211 module. We have tested the implementation on two different UNIX flavours: OpenWRT, and Ubuntu. mac80211 is a UNIX module that implements the IEEE OSI MAC Layer and which is today widely used across wireless devices.

The virtual interfaces on the controller side are created by using a shell script that relies on the UNIX *iw* utility, a command line configuration utility for wireless devices. The *AP* functionality is based on the UNIX daemon *hostapd*. Hostapd is an IEEE 802.11 controller process and IEEE 802.1X/WPA/WPA2/EAP/RADIUS authenticator. Hence, DFS is a full software-defined networking approach which can be applied to any wireless OFDM based device as long as it considers hostapd, and mac80211.

The performance evaluation described in this section has the intention to understand potential performance gains under realistic settings that DFS can bring to IEEE 802.11. The performance evaluation considers the following performance parameters: network throughput; packet loss; jitter.

We define *network throughput* as the average load of data bytes being transmitted on the network. Throughput has been computed recurring to the *Network Traffic Analyser (nload)* tool for each station, and then averaged across all stations. Jitter corresponds to the variation in time between packets arriving to stations. Packet loss has been defined as the percentage of packets that fail to reach their

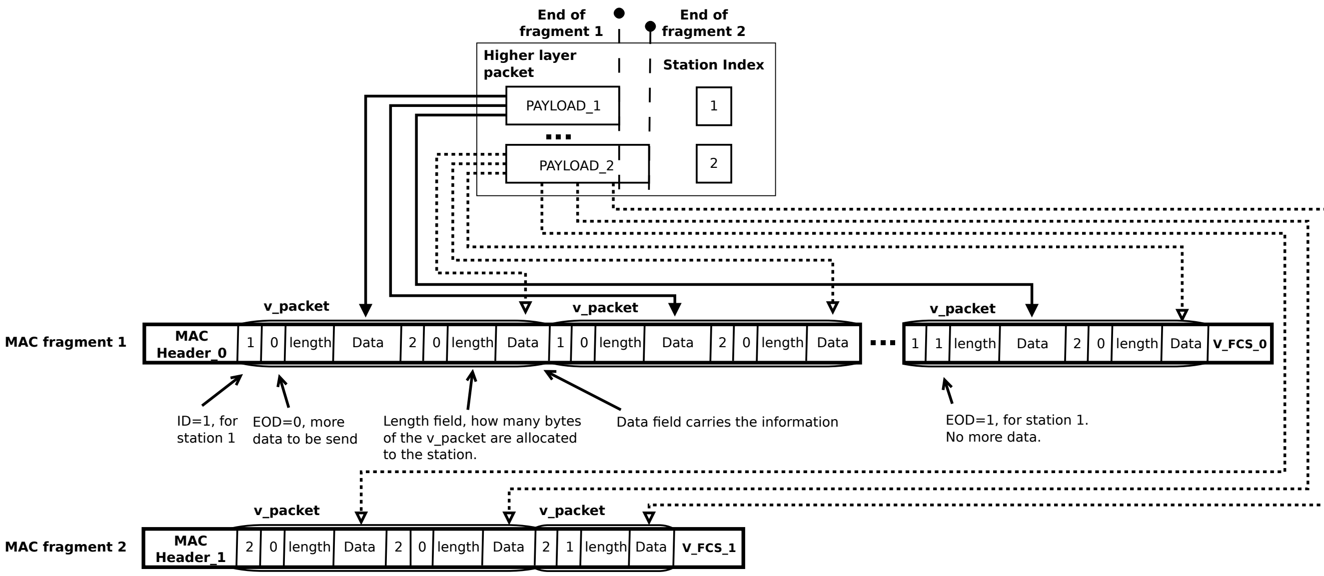


Figure 3: Superframe illustration.

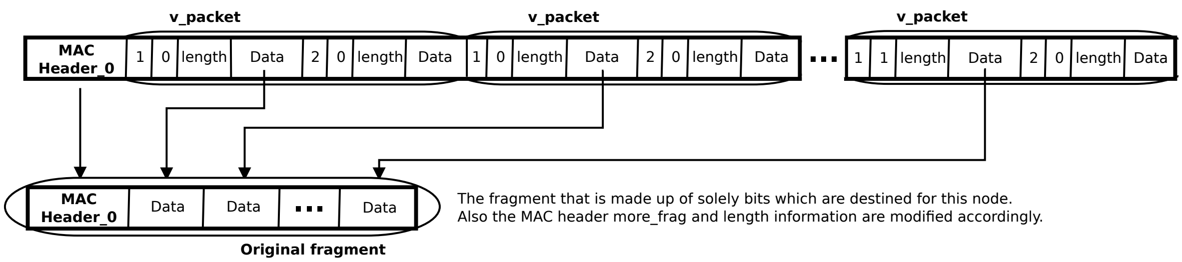


Figure 5: Station-side, re-creating a fragment of the original frame based on a received superframe.

destination. Both these parameters have been computed with iperf v2.0.5[2].

B. Experimental Settings

The local testbed used in our experiments is illustrated in Fig. 6. It consists of three Toshiba laptops, similar processing and CPU features, equipped with IEEE 802.11b/g/n cards, namely, two with chipset Atheros AR9485 (ath9k driver) and one with chipset Realtek rtl8192se (rtlwifi driver). The laptops, which serve as stations, rely on the operating system Ubuntu 12.04, kernel version 3.5.0-23-generic. The two represented controllers are IEEE 802.11g wireless Access Points, namely, Ubiquiti NanoStation M2 and airRouter HP, both holding the Atheros chip-set AR7241, 32Mb of RAM, with OpenWRT backfire 10.03.1, revision 29638, kernel version 2.6.32.27. Both APs consider hostapd version dated of 2011-11-03 and mac80211, included on compat-wireless package dated of 2011-11-15. Both modules are included with the OpenWRT system. One of the APs has been set to integrate DFS. Both controllers were set to work only in IEEE 802.11g mode. For the stations we considered mac80211, included on *compat-wireless version 3.6.8*.

The controllers have been placed in a room of 120 square meters.

Since the superframes are broadcast frames, the data rate has been limited to 1 Mbit per second (Mbps) in all scenarios. The upstream data rate is set to 54Mbps.

Concerning traffic, we have considered two different settings, one of which is based on real-time video streaming, and one which is based on the traffic generator tool iperf. Jitter, packet loss and bandwidth have been measured using iperf, while throughput have been measured using nload running on the stations. Our benchmark is plain IEEE 802.11g.

All experiments have been repeated ten times over different days, and at different times of the day. Results have been computed within a 95% confidence interval.

C. Performance Results

Scenario I, Basic Realistic Settings: On the first experiment, we have considered live video streaming via YouTube[3] at a data rate of 54Mbps, for 5 minutes and 58 seconds.

[2] Iperf - Tool for performing network throughput measurements. Available at Ubuntu repository and http://packages.ubuntu.com/source/precise/iperf

[3] Youtube video "Maestroo - an immersive sensing tool". Available at http://www.youtube.com/user/SITIulht/.

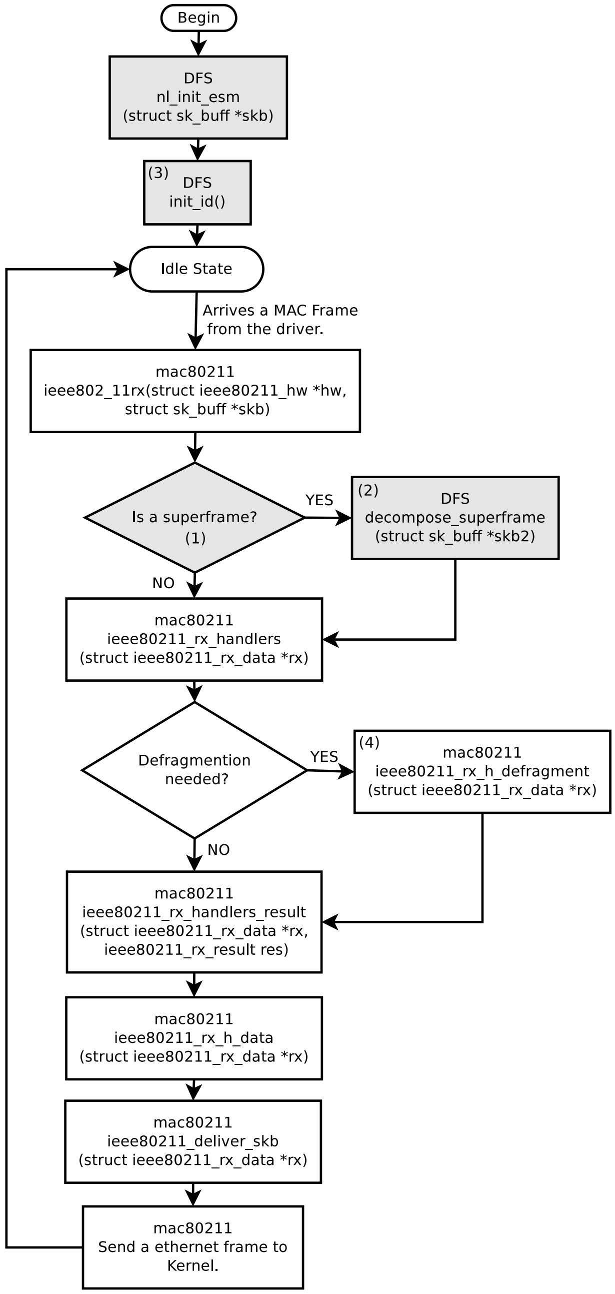


Figure 4: Station-side flow-chart.

The motivation for this scenario is to understand how DFS operates under realistic operational conditions, in particular for the controller side. Only two stations were used in this scenario and the distance to the controller AP was one meter.

Fig. 8 provides throughput results for Scenario I. The x-axis represents the two stations, while the y-axis represents the average throughput.

Under realistic conditions, what is observable is that DFS results in a fairer use of the channel across both stations, in average. While for plain IEEE 802.11g, what happens is that the first station that has the opportunity to transmit

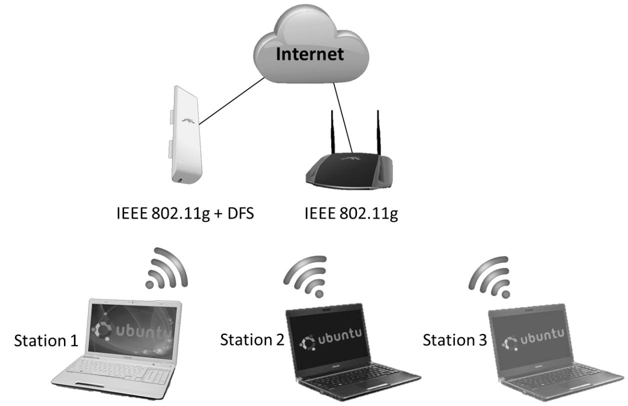


Figure 6: Testbed illustration.

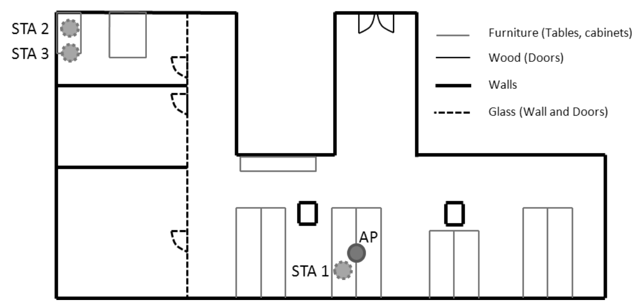


Figure 7: Scenario II, stations STA1, STA2, STA3 positioning for LoS and nLoS settings.

(in this case STA2) profits from such early opportunity, independently of the channel conditions.

The results show that, despite a decrease of 30.71% of throughput for STA2 by using DFS, compared with IEEE 802.11g, STA1 has its throughput increased 21.95%. In terms of fairness, the throughput difference between the two stations with DFS is only 3.59% against 82.32%, with IEEE 802.11g.

Scenario II, Line of Sight, Live Streaming Results: On the second experiment, we have relied on TCP and UDP streaming via the traffic generator tool iperf. A first ex-

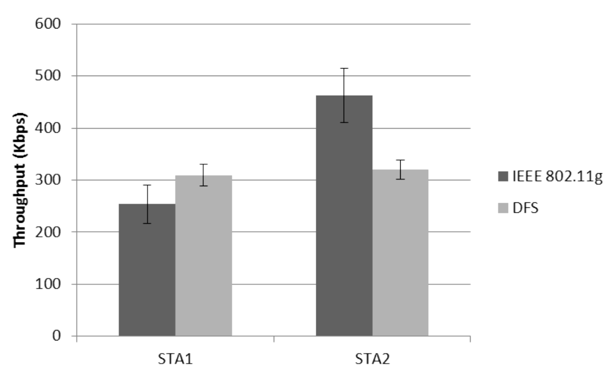


Figure 8: Scenario I, throughput results.

periment was performed again in *Line of Sight (LoS).* Two stations, STA1 and STA2, were placed one meter away from the controller to have a clear channel. Results are provided in Fig. 9.

In comparison to Scenario I, DFS again results in more fairness than plain IEEE 802.11g, even though the fairness ratio is not as high as in Scenario I. Still, fairness improves as with IEEE 802.11g the difference between the average throughput of the stations was of 68.11%. While for DFS that difference is of 27.04%. Hence, these results show that STA1 did not exclusively grab the medium until the end of the transmission. This is visible for the jitter and packet loss results (cf. Fig. 9 (b) and (c), respectively) for the case of STA2, as the jitter decreased 33.11% while the packet loss decreased 89.93%. The difference in jitter is a consequence of the frame concatenation process applied to superframes, which provides a way to transmit to multiple stations within a same duty cycle of a device.

Scenario II has been repeated for TCP traffic, being results illustrated in Fig. 10. There is again a fairness improvement in terms of throughput distribution of 44.4% for IEEE 802.11g against 16.65% with DFS. We would like to stress that STA1 is not penalized with this behavior: as shown, STA1 saw a decrease of circa 8% in throughput, while STA2 obtained an increase of circa 38%.

Scenario II, nLoS, Impact of Location: As interference is one of the main limitations of an adequate resource channel management, we have set another scenario, *non Line of Sight (nLoS),* where the two stations STA1 and STA2 were placed in different rooms as illustrated in Fig. 7. STA2 was placed at a distance of 14 meters away from the AP. The experiment aims at understanding the performance that DFS may attain for cases that are similar to the ones resulting from the "hidden station" problem.

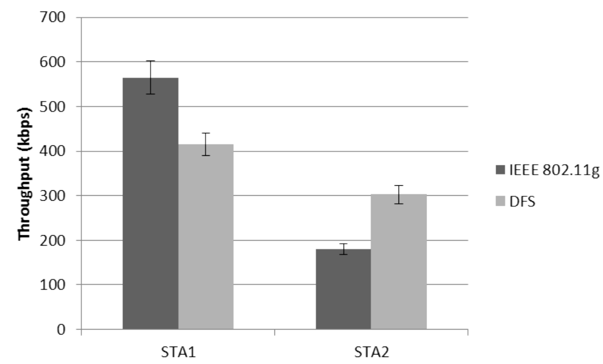
Fig. 11 provides results for the case of the nLoS test in Scenario II, with only STA1 and STA2, when generating UDP traffic. STA2 is the station with the worse conditions, as it is located further away from the controller.

By applying DFS, STA2 sees a throughput improvement of 22.43%, while STA1 obtains an improvement of 0.48%. As for jitter, results show a decrease in jitter of 31.94% for STA1 and of 39.62% for STA2, when applying DFS.

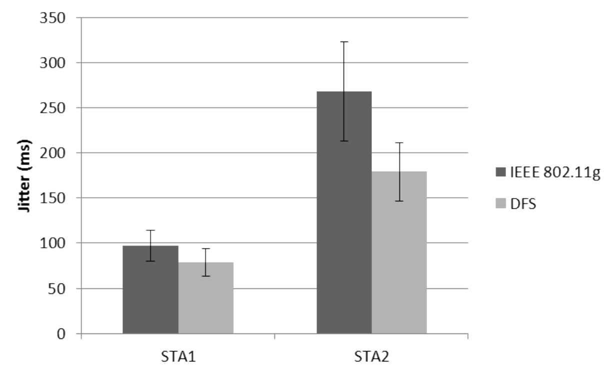
Packet loss decreases for STA1 and increases for STA2, as a consequence of applying DFS and its superframes. With UDP traffic, and considering plain IEEE 802.11, corrupted packets that arrive to a destination are simply discarded. As superframes carry data to multiple stations, and as they are treated in all destinations, their corruption implies that packet loss increases in all destinations.

Scenario II nLoS was repeated with TCP traffic and results obtained are illustrated in Fig. 12.

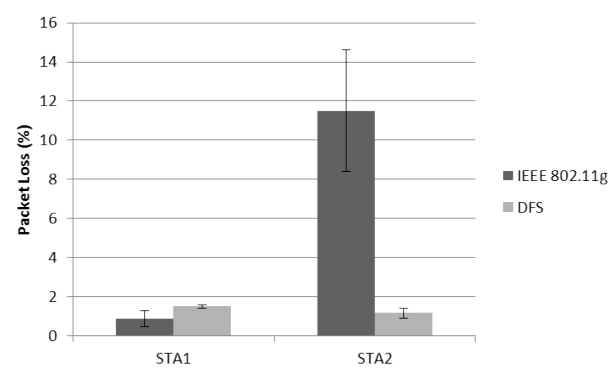
For this scenario, STA2, which had the worse channel conditions, had a significant improvement in throughput by applying DFS. This corroborates the justification which we have provided for packet loss in UDP, namely, that the packet loss increased in the previous scenario is due to an increase in packets being sent to STA2, due to the unreliable nature of UDP. In terms of fairness for both stations, the difference in average throughput for both stations is of circa 83% with IEEE 802.11g, against 42% for DFS.



(a) Throughput.



(b) Jitter.



(c) Packet Loss.

Figure 9: Scenario II, LoS, UDP results.

The relative difference for results obtained across all scenarios, from applying DFS to IEEE 802.11g are provided in Table II.

Fairness when Increasing the Number of Stations: To understand potential scalability aspects, we have repeated the nLoS scenario including 3 stations and UDP traffic as represented in Fig. 13. Both STA2 and STA3 are the stations with the worse conditions, as they are located further away from the controller (refer to Fig. 7).

By applying DFS, STA1 sees a throughput improvement of 39.71% compared to IEEE 802.11g, STA2 sees an improvement of 52.52%, while STA3 gets an improvement of 26.70%. As for jitter, results show a decrease in jitter of

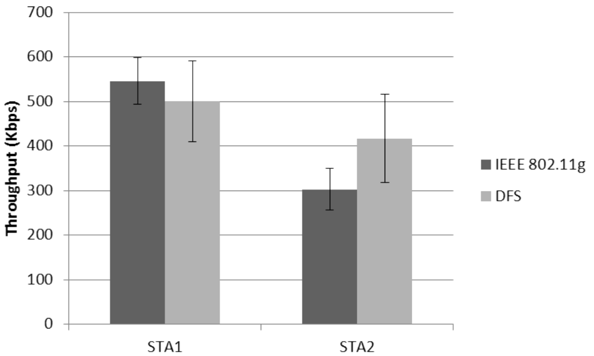


Figure 10: Scenario II, LoS, TCP throughput.

Table II: Summary of relative differences between DFS and IEEE 802.11g across all scenarios, with two stations.

	Avg Throughput Difference between the two stations (%)	
	DFS	IEEE 802.11g
Scenario I	3.59	82.32
Scenario II, LoS, UDP	27.04	68.11
Scenario II, LoS, TCP	16.65	44.44
Scenario II, nLoS, UDP	81.02	48.57
Scenario II, nLoS, TCP	41.97	82.95

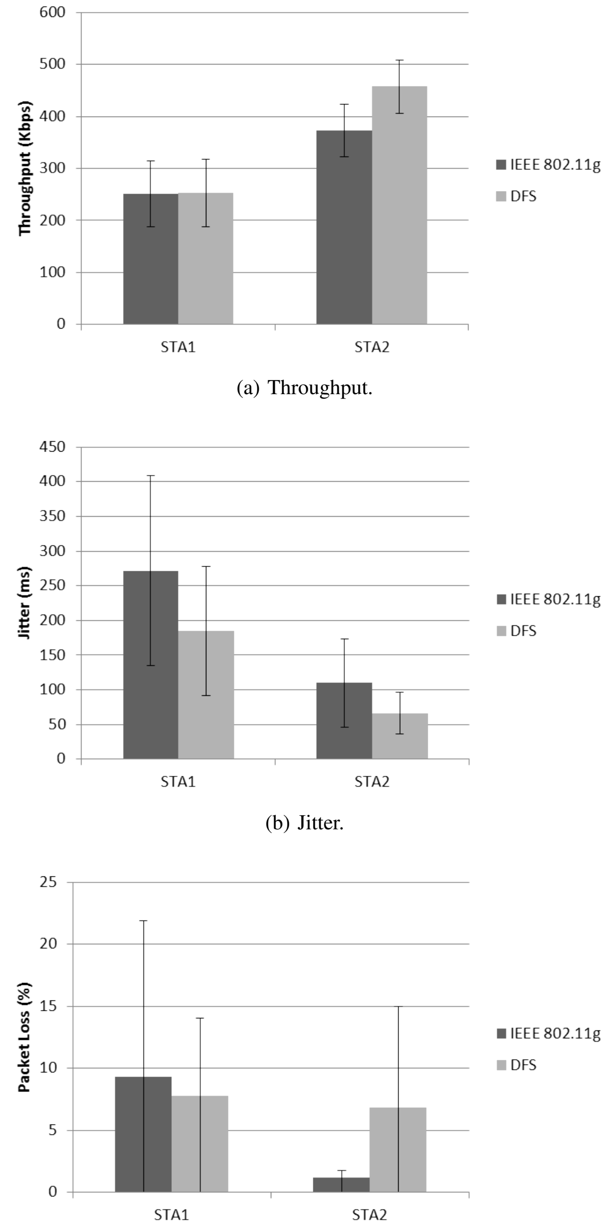
54.41% for STA1, 80.90% for STA2 and 17.88% for STA3, when DFS is used.

Packet loss has also significantly decreased for all stations, when running DFS instead of plain IEEE 802.11: 87.97% for STA1; 87.80% for STA2; 78.17% for STA3, when DFS is applied. Such decrease relates with the advantage of considering superframes when the controller transmits to multiple stations, as the controller does not need to consider fragmentation and additional buffering, when transmitting to each station. In plain IEEE 802.11g, the controller has to manage the requests of each station to capture the medium.We highlight that with plain IEEE 802.11g all stations experience a packet loss above 60%. While with DFS the maximum packet loss observed has been of 14.4%, for STA3.

Fig. 14 provides results when TCP traffic is considered.

For the case of TCP, and addressing throughput (cf. Fig. 14a), STA1 and STA2 have a higher throughput compared with STA3 when using IEEE 802.11g, as STA2 can only transmit when STA1 releases the medium. This explanation is confirmed in Fig. 14b, which illustrates the total amount of time that each station uses in active transmission (medium occupancy). STA1 used the medium less time when compared with the other two stations, therefore implying that it had a shorter communication period. Moreover, STA3 and STA2 had to contend for the medium thus resulting in lower throughput achieved by STA3 when compared with STA2.

Furthermore, in terms of throughput, with DFS, STA1 experiences lower throughput (decrease of 45.45%) and STA2 experiences a decrease of 9.59%. While STA3 throughput increased 8.37%.



(a) Throughput.

(b) Jitter.

(c) Packet Loss.

Figure 11: Scenario II, nLoS, UDP results.

The throughput variation relates also with the medium capacity. However, we highlight that even with the throughput reductions, DFS still results in better end-to-end delay - STA1 has an insignificant increase in delay, 1.69%; STA2 experiences lesser delay, 25.49%; STA3 also experiences lesser delay, 22.46%.

Scaling with DFS, Performance Prediction Aspects: As mentioned, DFS has been tested via a proof-of-concept implementation which is available to the community as a patch for mac80211. Our local testbed consisted of three physical devices. In order to further understand the performance of DFS when more stations are involved, and due to the usual limitations of local testbeds, in this section we forecast the DFS behavior in terms of average throughput, packet loss,

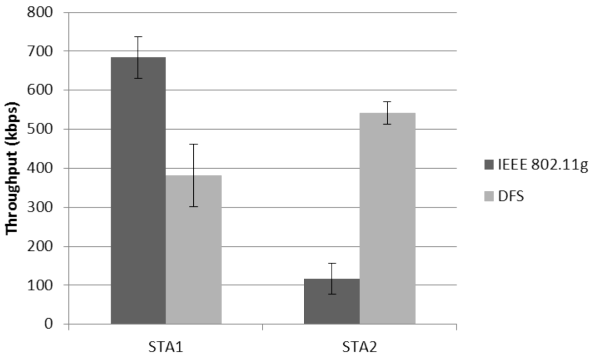


Figure 12: Scenario II, nLoS, TCP throughput.

as well as end-to-end delay, by considering a least squares estimation approach.

$$a = \bar{y} - b\bar{x} \tag{4}$$

$$b = \frac{\sum (x - \bar{x})(u - \bar{y})}{\sum (x - \bar{x})^2} \tag{5}$$

To assist in understanding DFS behaviour trend, we used a linear and a non-linear extrapolation. To get the linear extrapolation, a forecast equation was used.

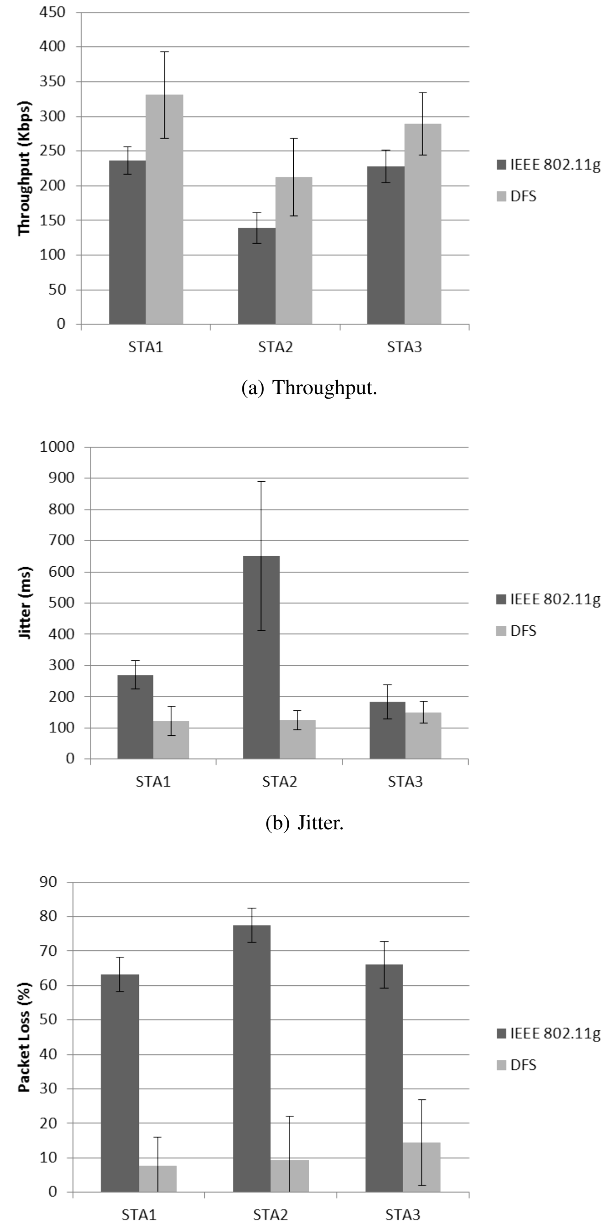
The equation for forecast is $y = a + bx$, where a is provided by equation 4, and b is the slope, provided by equation 5. The predicted value is y for a given x value.

From 4 and 5, $\bar{x}$ means the average of number of stations and $\bar{y}$, the average of the results obtained in our local testbed, in a nLoS scenario with UDP traffic.

The results for the linear forecast extrapolation are represented in Fig. 15. As shown, DFS results in greater fairness even when the number of stations increases, thus showing a similar trend to the one observed in our testbed under realistic conditions. The greater fairness obtained when DFS is applied is observed in terms of jitter and packet loss. We believe that the better performance relates with the aggregation technique for superframes. Our hypothesis for the observed behavior is related with the number of frames sent, which with the DFS aggregation technique is lower than when considering plain IEEE 802.11: several stations are served and hence medium competition is lower. While regular IEEE 802.11g stations are subject to the regular medium access protocols in which medium access competition has greater impact as observable on the jitter performance.

The next set of results considers a non-linear extrapolation based on the growth equation represented in Eq. 6. The growth function calculates the exponential growth curve that has the best fit for the provided known x-values, number of stations, and y-values, results obtained in our testbed (e.g. throughput, delay, packet loss). The parameter b is a constant which is the value of y when x is 0, and m is the growth factor. Results are provided in Fig.16.

$$y = b \times m^x \tag{6}$$



(a) Throughput.

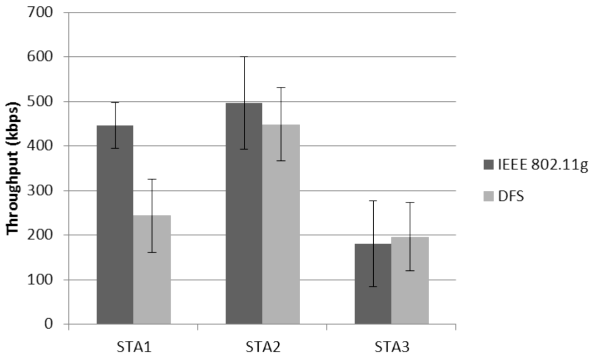
(b) Jitter.

(c) Packet Loss.

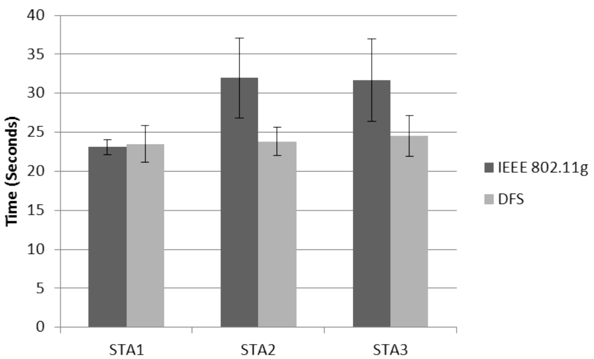
Figure 13: Scenario II, nLoS, UDP results with three stations.

The growth trend for packet loss and jitter is similar to the one previously observed when considering a linear extrapolation. DFS improves performance in terms of significantly lower jitter and packet loss. As for the throughtput growth trend, the results show that the application of DFS results in close behavior to IEEE 802.11g as expected, in particular when the channel is congested, as the data rate has been limited to 1Mbps, to simulate the IEEE 802.11 signaling conditions.

Therefore, IEEE 802.11 can benefit from integrating DFS, as the results extrapolation show trends with a significant improvement in terms of fairness. This improvement is



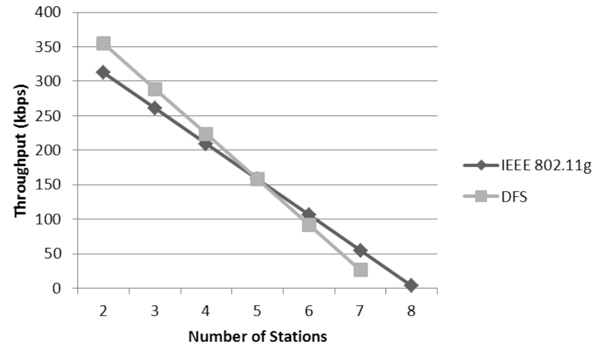
(a) Throughput.



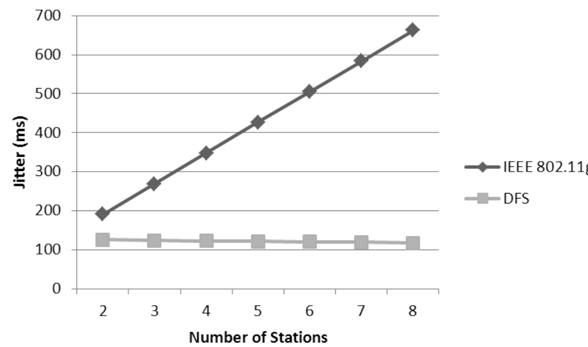
(b) Time.

Figure 14: Scenario II, nLoS, TCP result with three stations.

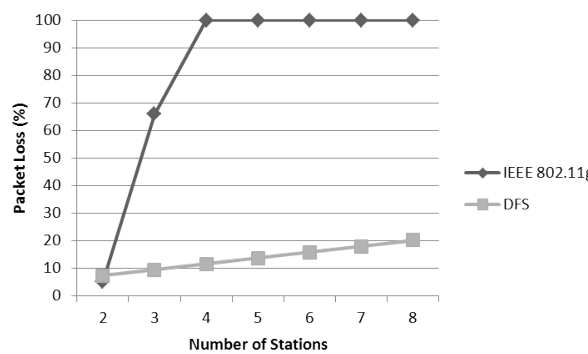
visible in a consistently lower and nearly constant delay, and of a low packet loss growth. With four stations, the communication is practically impossible in IEEE 802.11g under the features tested, while if DFS is applied, such communication is still feasible when considering eight stations.



(a) Throughput



(b) Jitter



(c) Packet Loss

Figure 15: Forecast extrapolation.

V. CONCLUSIONS

This paper describes DFS, a MAC Layer software-defined solution that has as motivation to improve fairness in wireless networks, in particular in environments that heavily depend on shared networking resources, such as UCNs. DFS is backward compatible with IEEE 802.11 standards, and provides a way to perform downstream transmission to multiple stations in the time frame that is in IEEE 802.11 only applicable to a single station. DFS shows relevant results in terms of fairness and DFS software is available as open-source, under LGPLv3.0.

As follow-up we are testing DFS in more complex environments and in particular analyzing whether or not DFS can solve additional problems of the MAC Layer, e.g. the hidden station problem. A second relevant aspect to address relates with extending the mechanism upstream, i.e., allowing stations to immediately transmit, without observing MAC contention.

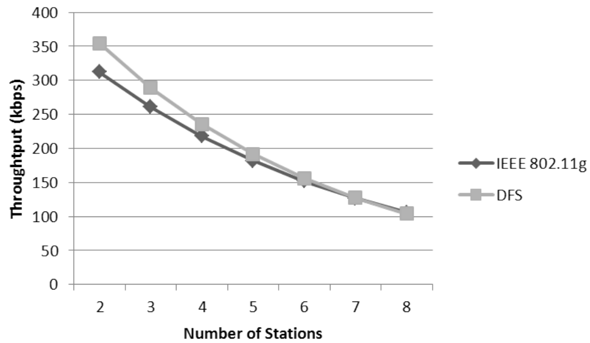
REFERENCES

[1] R. Sofia and P. Mendes. "User-provided Networks: Consumer as Provider", IEEE Communication Magazines, Feature Topic on Consumer Communications and Networking - Gaming and Entertainment.Volume: 46 Issue: 12 Pages: 86-91, December, 2008.

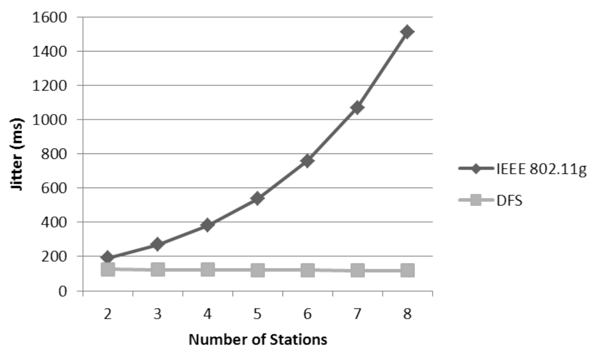
[2] EU IST FP7 ULOOP - User-centric Wireless Local Loop project, 2010-2013, grant number 257418. Available at http://copelabs.ulusofona.pt/~uloop/.

[3] M. H. Ahmed. "Call admission control in wireless networks: A comprehensive survey." IEEE Communications Surveys and Tutorials, 7.1-4 (2005): 50-69. 2005.

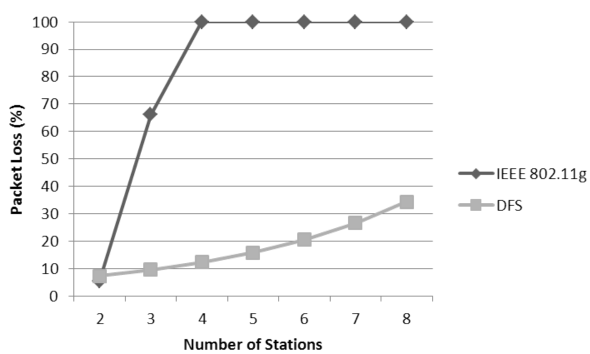
[4] T. Kwon, S. Kim, Y. Choi, and M. Naghshineh. "Threshold-type call admission control in wireless/mobile multimedia networks using prioritised adaptive framework." IEEE Electronics Letters 36, no. 9 (2000): 852-854.



(a) Throughput



(b) Jitter



(c) Packet Loss

Figure 16: Growth extrapolation.

[5] D. Pong, T. Moors. “Fairness and capacity trade-off in IEEE 802.11 WLANs”, in Proceedings Local Computer Networks, 2004.

[6] Y. Xue, L. Baochun, K. Nahrstedt. "Price-based resource allocation in wireless ad hoc networks." Quality of Service—IWQoS 2003. Springer Berlin Heidelberg, 2003. 79-96.

[7] M. J. Neely. "Optimal pricing in a free market wireless network." Wireless Networks 15.7 (2009): 901-915. 2009.

[8] Z. Fang, B. Bensaou. "Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad-hoc networks." INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies. Vol. 2. IEEE, 2004.

[9] B.-S. Kim, S. W. Kim, R. L. Ekl. "OFDMA-Based Reliable Multicasting MAC Protocol for WLANs," Vehicular Technology, IEEE Transactions on , vol.57, no.5, pp.3136,3145. 2008.

[10] K. Huang, G. Kang, H. Wang, G. Wang. "An adaptive contention window algorithm to hybrid OFDMA/CSMA mechanism in WLANs," Communications and Information Technologies (ISCIT), 2012 International Symposium on , vol., no., pp.1171,1176. 2012.

[11] G. Haile, L. Jaesung. "C-OFDMA: Improved Throughput for Next Generation WLAN Systems Based on OFDMA and CSMA/CA," Intelligent Systems Modelling & Simulation (ISMS), 2013 4th International Conference on , vol., no., pp.497,502, 29-31. 2013.

[12] S. D’Alessandro, A. M. Tonello, L. Lampe. "Improving WLAN capacity via OFDMA and cyclic prefix adaptation," Wireless Days (WD), 2009 2nd IFIP , vol., no., pp.1,5, 15-17. 2009.

[13] Kun Tan, Ji Fang, Yuanyang Zhang, Shouyuan Chen, Lixin Shi, Jiansong Zhang, and Yongguang Zhang. “Fine Grained Channel Access in Wireless LAN”, in ACM SIGCOMM 2010, Association for Computing Machinery, Inc., 2010.

[14] R. Sofia, L. Lopes. "Trust as a Fairness Parameter for Quality of Experience in Wireless Networks." In User-Centric Networking, pp. 159-169. Springer International Publishing, 2014. ISBN 978-3-319-05218-2.

[15] H. Rahul, F. Edalat, D. Katabi, C. G. Sodini. "Frequency-aware rate adaptation and MAC protocols." In Proceedings of the 15th annual international conference on Mobile computing and networking, pp. 193-204. ACM, 2009.

[16] J. Fang, K. Tan ; Y. Zhang ; S. Chen, L. Shi, J. Zhang, Y. Zhang, Z. Tan. "Fine-grained channel access in wireless LAN" ACM SIGCOMM Computer Communication Review 40.4 (2010): 147-158. 2010.

[17] V. K. Lau, Adaptive resource allocation for multiuser MIMO systems with transmit group MMSE.Lau. "Adaptive resource allocation for multiuser MIMO systems with transmit group MMSE." Wireless Communications, IEEE Transactions on 8.5 (2009): 2362-2368. 2009.

[18] D. Skordoulis, N. Qiang, C. Hsiao-Hwa Chen, A.P. Stephens, L. Changwen, A. Jamalipour. "IEEE 802.11n MAC frame aggregation mechanisms for next-generation high-throughput WLANs," Wireless Communications, IEEE , vol.15, no.1, pp.40,47. 2008.

[19] J. Dong Guen, C.-H. Choi, and W. S. Jeon. "Design and performance evaluation of a new medium access control protocol for local wireless data communications." IEEE/ACM Transactions on Networking (TON) 3.6 (1995): 742-752. 1995.

[20] T. Li, Q. Ni, D. Malone, D. Leith, Y. Xiao, T. Turletti. "Aggregation with fragment retransmission for very high-speed WLANs." IEEE/ACM Transactions on Networking (TON) 17.2 (2009): 591-604. 2009.

[21] C. Zarakovitis, Q. Ni, D. Skordoulis and M. Hadjinicolaou. "Power-Efficient Cross-Layer Design for OFDMA Systems with Heterogeneous QoS, Imperfect CSI and Outage Considerations", IEEE Transactions on Vehicular Technology, Vol. 61, No. 2, Feb 2012, pp. 781-798.

[22] Q. Ni, C. Zarakovitis. "Nash Bargaining Game Theoretic Scheduling for Joint Channel and Power Allocation in Cognitive Radio System", IEEE Journal on Selected Areas in Communications, Vol. 30, No. 1, Jan 2012, pp. 70-81.

[23] L. Lopes. R. Sofia, H. Osman, H. Haci. “A Proposal for Elastic Spectrum Management in Wireless Local Area Networks”, inProc. Infocom 2014. 2014.

[24] H. Zhu and J. Wang, “Chunk-based resource allocation in OFDMA systems - part I: chunk allocation”, Communications, IEEE Transactions on, vol. 57, no. 9, pp. 2734-2744, 2009.

[25] R. Sofia, P. Mendes, W. M. Junior, J. Saltarin, L. Lopes, Q. Zhou, J. M. Seigneurz, C. Ballester, H. Haci, H. Zhu, H. Osman, S. Peters, M. Yildiz, A. Bogliolo. “ULOOP Deliverable D3a - Specification of ULOOP functionality based on flow-chart and link to code available on the ULOOP SVN server”. Technical report number COPELABS-SITI-TR-13-20, Technical Report, 2013. Available at http://copelabs.ulusofona.pt/scicommons/index.php/publications/show/822.

[26] C.Y. Wong, R. S. Cheng, K. B. Lataief, and R. D. Murch. “Multiuser OFDM with adaptive subcarrier, bit, and power allocation,” Selected Areas in Communications, IEEE Journal on, vol. 17, no. 10, pp. 1747-1758, 1999.

[27] A. Matos, N. Martins, P. Mendes. “D3.9: ULOOP Software Suite”. Deliverable D3.9, European project User-centric Wireless Local Loop, 2013. Available at http://copelabs.ulusofona.pt/scicommons/index.php/publications/show/590.