Enhancing IEEE 802.11 Energy Efficiency for Continuous Media Applications

Vitor Bernardo[1], Marilia Curado[1], and Torsten Braun[2]

[1] Center for Informatics and Systems,
University of Coimbra, Coimbra, Portugal
{vmbern,marilia}@dei.uc.pt

[2] Institute for Computer Science and Applied Mathematics,
University of Bern, Bern, Switzerland
braun@iam.unibe.ch

Abstract. This paper proposes the Optimized Power save Algorithm for continuous Media Applications (OPAMA) to improve end-user device energy efficiency. OPAMA enhances the standard legacy Power Save Mode (PSM) of IEEE 802.11 by taking into consideration application specific requirements combined with data aggregation techniques. By establishing a balanced cost/benefit tradeoff between performance and energy consumption, OPAMA is able to improve energy efficiency, while keeping the end-user experience at a desired level. OPAMA was assessed in the OMNeT++ simulator using real traces of variable bitrate video streaming applications. The results showed the capability to enhance energy efficiency, achieving savings up to 44% when compared with the IEEE 802.11 legacy PSM.

Keywords: IEEE 802.11, Energy Efficiency, Power Save Mode

1 Introduction

The opportunity to connect mobile equipment, sensors, actuators and other devices to the Internet, usually referred as Internet of Things (IoT) [1], raises new challenges in the deployment of those equipments. The battery lifetime is still one of the most relevant challenges, since it is directly affected by the device communication capabilities. Despite numerous efforts to create alternative low power radio technologies, IEEE 802.11 seems to be the *de facto* standard for wireless communications in most common scenarios. Therefore, it is crucial to investigate and propose mechanisms aimed at saving energy while providing Internet access through an IEEE 802.11 ready interface.

Furthermore, the massive deployment of high demand continuous media application, namely Video on Demand (VoD) or Internet Protocol Television (IPTV), also enforces new requirements with respect to the equilibrium between energy efficiency and application performance. Besides specific application constraints other aspects may be considered, such as end-user guidelines about whether or not energy saving is mandatory. For instance, the end-user configuration can be related with daily mobility or traveling patterns. As the end-user battery lifetime expectations are extremely hard to predict, the inclusion of end-user feedback in the optimization process will bring some benefits.

This work proposes the Optimized Power save Algorithm for continuous Media Applications (OPAMA), aiming to improve devices' energy consumption using both end-user and application specific requirements, together with an optimized IEEE 802.11 power saving scheme and frame aggregation techniques.

The remaining sections of this paper are organized as follows. Section 2 discusses the related work, followed by the OPAMA proposal presentation in Section 3. The assessment of OPAMA performance, in the OMNeT++ simulator, is described in Section 4. Lastly, Section 5 presents the conclusions.

2 Related Work

This section introduces the background of the proposed algorithm, and presents the most relevant related work concerning IEEE 802.11 energy efficiency improvements for continuous media applications employing power saving techniques.

An IEEE 802.11 station (STA) under Power Save Mode (PSM) [2] (also known as Legacy-PSM) is able to switch off the radio during a certain period, aiming at saving energy during that time. A STA must inform the Access Point (AP) about the current power management mode by defining the corresponding power management fields in the control frames. When the power saving mode is enabled for a STA, the AP buffers all the packets to that station. If the AP has packets buffered to a certain STA, it will send a notification using the Traffic Indication Map (TIM) field of the *Beacon* frames. In the PSM, a STA must wake-up regularly to receive the *Beacon* frames. By performing this action, a STA that does not have any data buffered on the AP will be required to wake up recurrently, resulting in unnecessary energy consumption. To overcome this limitation, IEEE 802.11e [3] introduced the Unscheduled Automatic Power Save Delivery (U-APSD) algorithm. The main difference between the PSM and the U-APSD is related to the proactivity implemented in the U-APSD scheme. Unlike PSM, where only the Access Point (AP) is able to inform the station about pending packets, in U-APSD, the STA can itself ask the AP for new downlink messages pending in the queue. More recently, IEEE 802.11n [4] also announces two contributions to the power saving schemes, namely the Spatial Multiplexing (SM) Power Save and the Power Save Multi-Poll (PSMP) techniques.

Energy saving mechanisms for IEEE 802.11 can consider cooperation between the energy aware mechanisms at the lower (e.g. MAC layer aggregation) and upper layers. Camps-Mur et al. [5] have studied the impact of IEEE 802.11 MAC layer aggregation on both PSM and U-APSD schemes. The authors proposed a Congestion Aware-Delayed Frame Aggregation (CA-DFA) algorithm, which is divided into two logical parts: congestion estimation and dynamic aggregator. Congestion estimation is responsible for assessing the network capabilities and uses these values as near real-time input for dynamic aggregation. Being able to measure accurately network congestion, it allows the algorithm to dynamically adapt the maximum frame aggregation size when the network congestion goes below a certain limit. When compared with the IEEE 802.11 standard aggregation schemes, the CA-DFA performance is superior, particularly in terms of energy consumption. However, the CA-DFA algorithm does not support any end-user feedback.

Tan et al. [6] proposed a cross-layer mechanism based on the standard PSM, but using information provided by the upper layers. The algorithm, named PSM-throttling, aims at minimizing energy consumption for bulk data communications over IEEE 802.11. The PSM-throttling concept is based on the idea that there are already many Internet based applications performing bandwidth throttling and, as a result, there is an opportunity to improve energy efficiency at the client side. PSM-throttling uses the under-utilized bandwidth to improve the energy consumption of bandwidth throttling applications, such as video streaming. Nonetheless, it does neither consider the inclusion of dynamic aggregation, nor the possibility that the end-user controls himself the maximum allowed delay.

An adaptive-buffer power save mechanism (AB-PSM) for mobile multimedia streaming was proposed by Adams and Muntean [7] to maximize the STA sleep period. The proposal includes an application buffer, able to hide the frames from the Access Point and, consequently, to avoid the TIM reports with pending traffic indication. The authors argue that the amount of packets to store in that buffer could be dynamic, but they do not explain how to overcome this issue. Moreover, AB-PSM aims to be an application-based approach, but the mechanism to be used by the STA to provide feedback to the AP was not defined. Additionally, aggregation mechanisms were not employed and the testbed study is very limited, since only battery lifetime was analyzed. This is an important parameter, but it should always be correlated with the drawbacks introduced in the end-user application (e.g., extra delay or jitter).

According to Palit et al. [8] the feasibility of employing aggregation is strongly related with the scenario and/or application. In order to understand the typical packet distribution in a smartphone data communication, the authors have analyzed mobile device traffic. The main observations are that around 50% of the packets have a size less than 100 bytes and 40% have an inter-arrival time of 0.5ms or less. These conditions enable a good opportunity to perform aggregation. Using this motivation, the authors have studied the aggregation impact in the smartphones' energy consumption. The proposed aggregation scheme uses a buffering/queuing system in the AP together with PSM in the client side. The proposed packet aggregation mechanism, named Low Energy Data-packet Aggregation Scheme (LEDAS), receives packets from the different applications through the Logical Link Control sub-layer and performs the aggregation. This approach showed some good results, but application requirements, such as the maximum tolerable delay, were not taken into account. With the native support for frame aggregation in IEEE 802.11n [9], which includes two distinct approaches to perform MAC frame aggregation, named Aggregated MAC Service

Data Unit (A-MSDU) and Aggregated Mac Protocol Data Unit (A-MPDU), various studies concerning aggregation performance have been done [10]. Kennedy et. al studied the adaptive energy optimization mechanism for multimedia centric wireless devices [11] and concluded that significant energy saving could be achieved when performing application-aware optimization. Pathak et al. [13] have proposed an application level energy consumption profiling tool for mobile phones and reported issues concerning high energy usage in I/O operations. The software-based energy methodologies were early surveyed by Kshirasagar [12].

Although others in the literature [7][14] have also proposed energy optimization for continuous media applications none takes advantage of all the key optimization parameters proposed in this work. To the best of our knowledge, this paper proposes an original optimized power saving algorithm for continuous media applications, which combines the usage of buffering techniques and frame aggregation mechanisms, while using the end-user feedback to keep the application quality within the defined limits. Additionally, although the novel power saving modes and aggregation schemes are available in more recent IEEE 802.11 standards, the Legacy PSM still is the *de facto* standard algorithm concerning PSM in IEEE 802.11, while the implementation of other algorithms is mainly optional. As a result, the proposed algorithm is based on Legacy PSM and uses A-MSDU aggregation, which is already mandatory in the reception side of the IEEE 802.11n standard.

3 Optimized Power save Algorithm for continuous Media Applications (OPAMA)

This section introduces the proposed Optimized Power save Algorithm for continuous Media Applications (OPAMA).

3.1 Motivation

Mobile end-user energy constraints are still one of the critical issues to be addressed in wireless communication protocols, particularly at the MAC Layer. IEEE 802.11, the most popular in real world equipment wireless technology uses the Power Save Mode (PSM), usually referred in the literature as Legacy Power Save Mode (Legacy-PSM), to limit energy consumption. However, the Legacy-PSM utilization in the presence of continuous media applications (e.g., video or voice) does not bring considerable energy savings, due to protocol design limitations, as explained next.

Legacy-PSM buffers traffic at the Access Point (AP) to all the stations (STA) operating in PSM mode, which indicate that they are in a *doze* state. A STA must wake-up to receive the *Beacon* sent by the AP at the beginning of each *Beacon Interval*. When broadcasting a *Beacon*, an AP supporting PSM must look for pending packets for each STA in a *doze* state that is currently associated with the AP. If there is data pending for a certain STA, the AP reveals it through the Traffic Information Map (TIM) field present in the *Beacon*.

When receiving a *Beacon*, a STA analyzes the TIM to verify the pending information existing in the AP buffer. Once there is pending data, the STA sends back a *PS-Poll* message to the AP asking for the data. The AP may reply with a single acknowledgement (ACK) or directly with the pending data frames. Then, the STA must stay awake while the *MoreData* flag is set. The AP will set this flag, while there is data to be delivered, while the STA should send back a *PS-Poll* for each pending frame. Therefore, when receiving data from a continuous media application, the STA will not be able to stay in a *doze* state for long, since there will be almost always some data to be received. As a result, even if the device battery is near a critical threshold, it will not be possible to save energy by employing the Legacy-PSM. A detailed discussion concerning PSM operation and buffer-related issues at the AP was performed by Zhu et al. [15].

OPAMA addresses these issues by introducing the end-user expected performance feedback in the process, allowing higher control opportunities at the AP. The next subsection presents OPAMA design and architecture.

3.2 Architecture

The main goal of OPAMA is to allow the end-user to save energy while keeping a desired quality at the application level. For instance, when the device battery is low, the end-user might like to have the possibility to slow down the transmission performance up to a certain level in order to save energy. To accomplish this goal, the STA sleep periods must be maximized. Consequently, OPAMA will manage the AP buffer differently compared to Legacy-PSM. While the Legacy-PSM will always inform the STA about any pending data to the STA, OPAMA will employ an algorithm based on the end-user expectations for the application performance to decide when pending data information should be sent to the STA. As on Legacy-PSM, OPAMA pending packets will stay in the AP queue. As a result, this operation will not affect the Legacy-PSM standard protocol [7].

Figure 1 depicts a simplified operation scenario of OPAMA. *STA-1* is operating in a doze state, and it is being served by *AP-1*, which is then connected to the core network (not represented here).

OPAMA operates as follows: *STA-1* left the *doze* state to receive *Beacon-1*. As there are no pending frames to be delivered, it just goes back into sleep mode. The first data for *STA-1* arrives at the *AP-1* when the STA is sleeping, then it is buffered. Again, *STA-1* becomes awake to receive *Beacon-2*. At this moment, there is already pending data for the STA. However, OPAMA will employ a specific algorithm (Algorithm 1) to determine whether *STA-1* should be informed about pending data. In the example of Figure 1, the algorithm returned false and the TIM of *Beacon-2* does not include information about pending data for *STA-1*. The pending data information is only sent within *Beacon-3*, followed by the data transmission start. Later, in *Beacon-5* OPAMA decides again to queue the frames for a longer time, allowing *STA-1* to return into the *doze* state with pending data available.

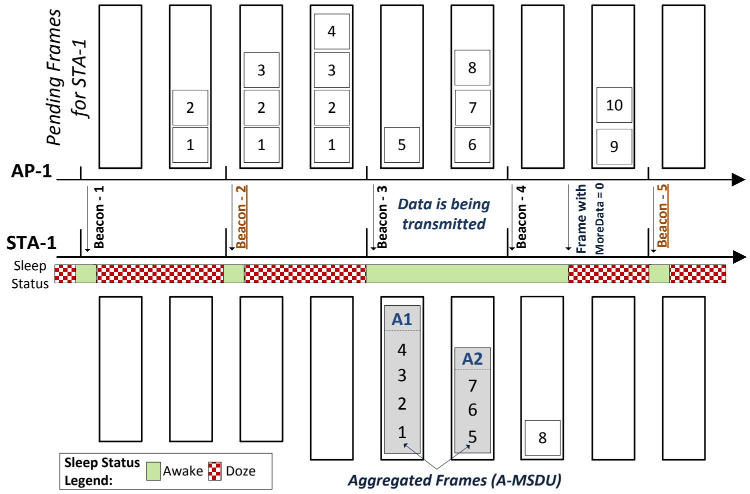

Fig. 1. OPAMA algorithm simplified operation example.

When the frames stay longer in the AP queue there are more opportunities to perform aggregation, as represented in Figure 1. In this case, *Frame-1*, *Frame-2*, *Frame-3* and *Frame-4* were aggregated using the A-MSDU scheme into *Frame-A1* and *Frame-5*, *Frame-6* and *Frame-7* into *Frame-A2*. The number of frames present in each A-MSDU is dynamic and depends on the total amount of bytes to be sent. As a result, *Frame-A2* carries fewer frames than *Frame-A1*. *Frame-8* was sent without aggregation, since there is only a single frame to be sent.

The end-user feedback will be transmitted to the access point using two distinct messages, *PS-Poll* and *NullFunction*. The first message is used to request data from the AP, while the latter is an empty message used to inform the AP about shifts between two distinct power modes (e.g., going to sleep). Therefore, these message types are only transmitted from the STA to the AP and they do not carry payload data. OPAMA will add one extra byte field to these messages, allowing the STAs to inform the AP about two different performance parameters: the average delay for the last received frames and the maximum allowed delay.

The decision to determine whether or not pending data information should be sent is performed by the OPAMA core algorithm, defined in Algorithm 1. First of all, OPAMA gets all the reference values needed to execute the algorithm, such as the maximum delay allowed by the STA or the aggregation limit support. Later, OPAMA analyzes the pending frames for the current STA, starting by verifying the delay related constraints (*lines 11 to 18*). When analyzing each frame, OPAMA also updates the total pending bytes to be sent (*line 24*) and performs an application dependent assessment (*lines 19-23*). Actually, OPAMA provides specific mechanisms for video applications, where the main goal is to ensure that no more than a defined number of video key frames (α parameter in *line 20*) will be queued. The video key frames parameter is specific to video

Algorithm 1 Determine whether or not pending data information should be sent to a certain STA

```
1: function Send_Pending_Data_To_STA_Decision(STA_MacAddress)
2:     ▷ Update the STA maximum allowed delay with information received from the STA in a previous PS-Poll or NullFunction message.
3:     refresh_STA_Maximum_Allow_Delay(STA_MacAddress)
4:     STA_MaxDelay ← STAList[STA_MacAddress].maxDelay
5:     Aggregation_Threshold ← getAggregationThreshold(STA_MacAddress)
6:     TimeUBeacon ← getTimeUntilNextBeacon() ▷ Gets the time until sending next beacon
7:     ...
8:     PendingFramesList ← getPendingFrames(STA_MacAddress)
9:     TempPendingBytes ← NULL
10:    for each PFrame in PendingFramesList do
11:        ▷ Check if the actual frame delay is greater or equal than the maximum delay defined by the STA
12:        if getActualDelay(PFrame) > STA_MaxDelay then
13:            return TRUE
14:        end if
15:        ▷ Check if the sum of actual frame delay with the time until next beacon is greater or equal than the STA_MaxDelay
16:        if (getActualDelay(PFrame) + TimeUBeacon) ≥ STA_MaxDelay then
17:            return TRUE
18:        end if
19:        if PFrame_MediaType == "video" and PFrame_FrameType == "I" then
20:            if get_Total_Video_KeyFrames_Pending_To_STA(STA_MacAddress) > α then
21:                return TRUE
22:            end if
23:        end if
24:        TempPendingBytes ← TempPendingBytes + PFrame_SizeBytes
25:    end for
26:    if (TempPendingBytes/Aggregation_Threshold) ≥ β then
27:        return TRUE
28:    end if
29:    return FALSE                    ▷ Pending data information will not be sent
30: end function
```

applications, but all the others mechanisms can be used with mixed traffic scenarios. The performance when handling combined application scenarios might depend on end-user preferences. For instance, the STA maximum allowed delay, defined by the end-user, can be defined using an algorithm designed to select the best parameter according to the end-user high level preferences for each application type. Additionally, the algorithm analyzes the maximum allowed number of aggregated frames to be sent using the STA aggregation limit information ($Aggregation_{Threshold}$) and the total size of current pending data. The parameter β (*line 26*) controls the maximum number of aggregated frames, which can be queued to a certain STA. The configuration of this parameter might also be performed using dynamic approaches where, for instance, the network conditions or frames queuing time in lower layers (e.g. physical) are considered. The aggregation threshold information is associated with each STA (*lines 3-5*), since the maximum feasible aggregation size is related to the STA Maximum Transmission Unit (MTU).

The following section presents detailed information concerning OPAMA performance when compared with the Legacy-PSM and when no PSM is used.

4 Performance Evaluation

This section shows the OPAMA evaluation performed in OMNeT++. First, the simulation details and configuration parameters are given, followed by OPAMA detailed performance analysis. The analysis includes OPAMA performance against Legacy-PSM and no PSM case, and a study concerning OPAMA key configurable parameters.

4.1 Simulation Scenario and Setup

The assessment of OPAMA was performed with two objectives. First, it aims to evaluate the impact of the proposed mechanism on both energy consumption and delay, when compared to Legacy-PSM and no PSM scenarios. Second, to assess how the aggregation threshold influences the behavior of OPAMA.

The tests were conducted in the OMNeT++ 4.2.2 [16] simulator together with the INET Framework 2.0.0. As one of the main goals of this work is to study energy consumption in the IEEE 802.11 interfaces, a multimeter like module, based on the existing INET Framework battery model, was created. This module can measure energy consumed in a IEEE 802.11 interface, by computing the time spent in each state. The simulation scenario used is illustrated in Figure 2.

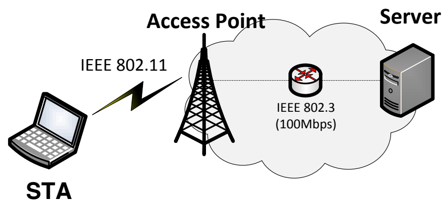

Fig. 2. OMNeT++ IEEE 802.11 simulation scenario.

Table 1 illustrates the power values [5] used for each considered state in the IEEE 802.11 physical layer implementation and the key parameters defined for the simulation. Both Legacy-PSM and OPAMA were implemented using the OMNeT++ INET framework. The IEEE 802.11 radio Bit Error Rate (BER) used in this simulation study results from values obtained for various IEEE 802.11g physical modes, using a dedicated Orthogonal Frequency-Division Multiplexing (OFDM) physical layer simulator. The OPAMA related values are the configuration of both α and β parameters used in Algorithm 1.

The assessment of OPAMA was performed using publicly available real traces from a video application [17]. The selected video was the “Sony Demo”. This

Table 1. OMNeT++ simulation parameters.

Parameter	**Value**
Total simulation time	660 seconds
Number of Runs	20
IEEE 802.11 - Operation mode	G
IEEE 802.11 - Beacon interval	100ms
IEEE 802.11 - Aggregation type	A-MSDU
Radio - Attenuation threshold	-110dBm
Radio - Maximum sending power	2.0mW
Radio - SNIR threshold	4dB
Radio - BER table	"per_table_80211g_Trivellato.dat"
Power while transmitting	2000mW
Power while receiving	1500mW
Power while idle	300mW
Power while sleep	20mW
OPAMA α parameter	10
OPAMA β parameter	3

sequence was encoded with MPEG-4 using a Variable Bit Rate (VBR), and has a resolution of 352x288, containing 17000 frames. The video is played for 10 minutes. Additionally, three distinct video qualities were selected for the tests, as summarized in Table 2.

Table 2. Video traces details.

Name	**Quantizer**	**Mean Frame Bitrate**	**Peak Frame Bitrate**	**Mean Frame Size**
Video-Q1	20	199.91 KBit/sec	2410.56 KBit/sec	832.99 Bytes
Video-Q2	12	319.55 KBit/sec	4139.04 KBit/sec	1331.45 Bytes
Video-Q3	04	1164.22 KBit/sec	10989.84 KBit/sec	4850.90 Bytes

All the results presented in the following sections include 20 runs using distinct random seed numbers with a confidence interval of 95%.

4.2 Results

This subsection presents the attained results regarding OPAMA performance assessment, compared with Legacy-PSM and no PSM scenarios.

OPAMA with No End-user Feedback (OPAMA-NEF): In order to compare the proposed algorithm base implementation against both Legacy-PSM and no PSM scenarios, in this study OPAMA was used without considering the STA maximum allowed delay information (OPAMA-NEF). Nevertheless, OPAMA-NEF still uses aggregation to send multiple packets arriving within a small interval ($\leq$ *5ms*). The maximum aggregation size was defined as 2272 Bytes, which is the IEEE 802.11g MTU. This configuration will allow a proper validation against the Legacy-PSM.

Figure 3 depicts a *boxplot* representing the end-to-end delay (in milliseconds) obtained for all the packets needed to stream each of the three distinct VBR videos already presented (Table 2).

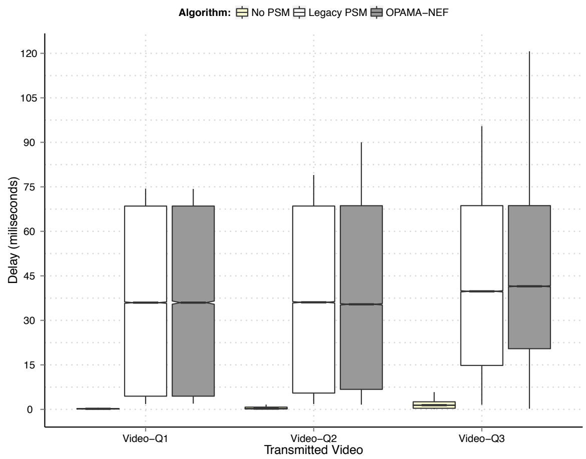

Fig. 3. No PSM, Legacy PSM and OPAMA-NEF end-to-end delay.

As expected, the scenario where no PSM is used shows a lower delay compared with both Legacy-PSM and OPAMA-NEF. When assessing Legacy-PSM and OPAMA-NEF performance it is noticeable that the delay is similar in both cases. The total energy consumed (in Joule) during the video transmission is illustrated in Figure 4.

The confidence interval limits are represented by the lines in the top of each bar. Although both Legacy-PSM and OPAMA-NEF introduce extra delay, the energy saving is not significant (only 6% to 8%), which does not configure a good tradeoff between the extra delay introduced and the energy consumed. As discussed previously, this behavior is mainly caused by the limitations of these power save mode protocols in the case of continuous media applications. Since those applications have almost always data pending to be transmitted, the possibilities for the STA to sleep are very limited. It must be highlighted that unlike OPAMA-NEF, OPAMA will be able to control whether or not the pending data information should be broadcasted to the STA, allowing a better sleep period optimization.

Impact of STA Maximum Allowed Delay on OPAMA performance: This subsection studies the impact of the maximum allowed delay defined by

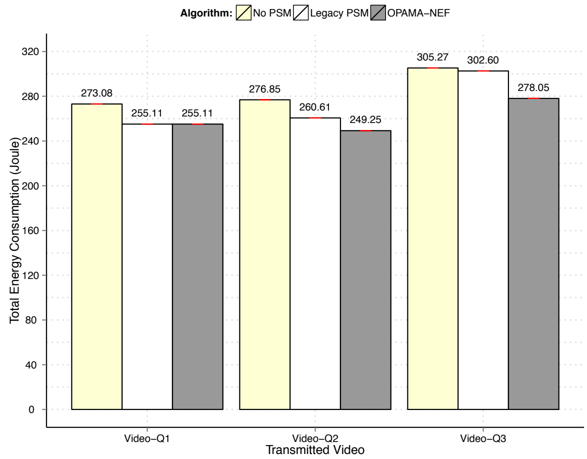

Fig. 4. No PSM, Legacy PSM and OPAMA-NEF energy consumption.

the STA on OPAMA performance. From now on, as the obtained results with the three distinct videos (*Video-Q1*, *Video-Q2* and *Video-Q3*) are similar, and due to lack of space, only Video-Q2 will be used in the analysis. Figure 5 depicts a *boxplot* with the end-to-end delay (in milliseconds) in the y-axis. The x-axis represents the STA maximum allowed delay (in milliseconds). To allow a proper performance comparison, the maximum allowed delay defined by the STA was always kept constant in each test set.

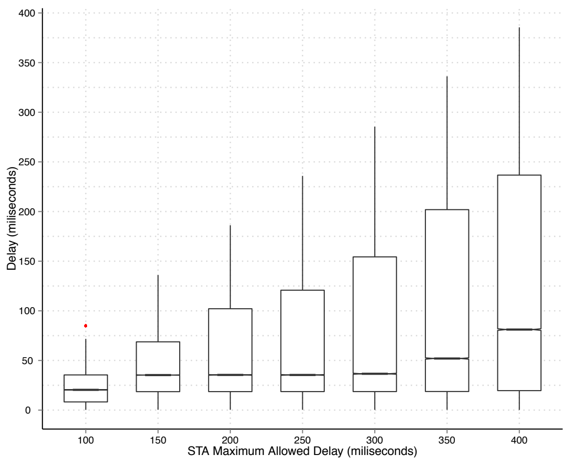

Fig. 5. End-to-end delay for OPAMA with Maximum Allowed Delay defined by the STA.

The STA maximum allowed delay was never exceeded for all the test cases. By observing the *boxplots* mean values, it is possible to conclude that end-to-end delay is below 100ms in all the tested scenarios. This behavior can be explained by the strict delay control performed in conjunction with frame aggregation, as OPAMA always tries to maximize the number of frames sent in each A-MSDU frame. The first quartile analysis also shows that for 25% of the packets, the delay is roughly the same as in the no PSM scenario (see Figure 3). This fact is directly related with the proper aggregation opportunities created by OPAMA. Additionally, it is also possible to observe that 75% (third quartile) of the delivered packets have only around a half of the maximum delay tolerated by the STA.

A comparison of the obtained energy savings regarding the employment of OPAMA compared with both Legacy-PSM and no PSM scenario is shown in Figure 6. The y-axis represents the energy saved in percentage, while the maximum allowed delay defined by the STA is depicted in the x-axis.

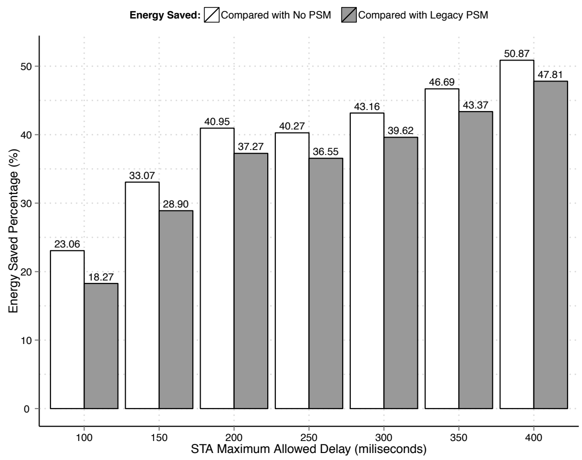

Fig. 6. Energy savings with OPAMA, compared with Legacy-PSM and no PSM scenarios.

The results show benefits of using OPAMA, when the STA can accommodate some delay (e.g., by using a local buffering technique). The savings for the 100ms maximum delay when compared with the Legacy-PSM are around 18%, which in this particular case allows the end-user to play the video for almost two minutes more using the same energy. Higher maximum allowed delays, such as 300ms, boost the savings to around 40%. At a first glance, it might not seem interesting to employ such large delays. However, the STA can dynamically inform the OPAMA ready AP about the maximum expected delay to reflect the end-user behavior.

OPAMA performance with larger aggregation threshold: OPAMA uses MAC layer aggregation (A-MSDU) as one of the algorithm components. Until now, all the tests were performed using a maximum aggregation size of 2272 Bytes, which is the MTU for IEEE 802.11g. Nevertheless, in IEEE 802.11n, where aggregation at the receiver side is already mandatory, the MTU can be up to 7935 Bytes. Therefore, this section investigates the OPAMA behavior with two distinct maximum aggregation sizes.

Figure 7 shows the end-to-end delay for aggregation threshold size of 2272 and 7935 Bytes. The x-axis represents the maximum allowed delay by the STA.

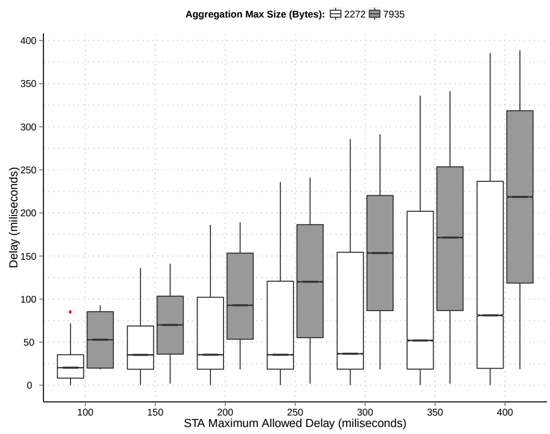

Fig. 7. Aggregation threshold impact on OPAMA delay.

For both scenarios, the maximum delay is not greater than the one defined by the STA. When comparing the two aggregation scenarios, the one with the lowest aggregation size shows a lower mean delay. If only the end-to-end delay is analyzed this might reveal a poor OPAMA performance with larger aggregation threshold. However, the information regarding energy consumption, shown in Figure 8, highlights the opposite.

When employing larger aggregation threshold (i.e., 7935 Bytes) OPAMA introduces additional delay for almost all the delivered packets. The energy consumption is significantly lower. Therefore, OPAMA clearly improves the cost/benefit tradeoff between delay and energy consumption under these conditions. The usage of larger aggregation frames also reduces the number of MAC layer acknowledgments in the network, which reduces the global network contention and maximizes the STA sleep time.

The usage of 7935 bytes as maximum aggregation threshold, when STA maximum allowed delay is defined as 100ms, is able to achieve savings of 32%. Moreover, when comparing OPAMA performance under these conditions with Legacy-PSM, the savings are up 44%. The savings for a STA allowing a maximum delay of 300ms and 400ms are 74% and 76%, respectively.

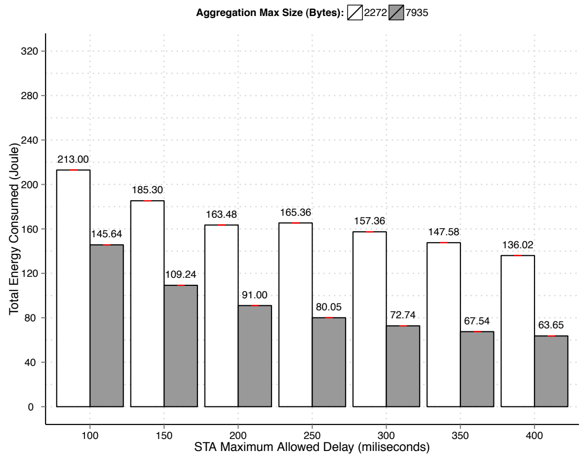

Fig. 8. Aggregation threshold impact on OPAMA energy consumption.

5 Conclusions

The energy efficiency in the end-user IEEE 802.11 ready devices is still an important factor towards a fast and global deployment of the future Internet of Things paradigm, since battery lifetime is one of the most critical factors in a daily usage. This paper investigates and proposes a mechanism aiming at saving energy while supporting continuous media applications. The proposed power save algorithm for IEEE 802.11 networks, named OPAMA, was designed to enhance the energy consumption by extending the IEEE 802.11 legacy PSM in order to accommodate the end-user feedback, and using Aggregated MAC Service Data Unit (A-MSDU) to deliver the data frames.

OPAMA performance assessment showed capabilities to improve energy efficiency, while keeping the end-user expectation at the defined level. When compared with the IEEE 802.11 Legacy-PSM, the OPAMA proposal achieved energy savings up to 76% in a higher tolerable delay scenario and 44% for a scenario where the STA can only accommodate a maximum delay of 100ms. The impact of the aggregation threshold in the proposed algorithm performance was also noticeable, depicting considerable energy savings.

Acknowledgments. This work was partially supported by the iCIS project (CENTRO-07-ST24-FEDER-002003), co-financed by QREN, in the scope of the Mais Centro Program and European Union's FEDER and by the COST framework, under Actions IC0804 and IC0906. The first author was also supported by the Portuguese National Foundation for Science and Technology (FCT) through a Doctoral Grant (SFRH/BD/66181/2009).

References

1. Tozlu, S., Senel, M., Mao, W., Keshavarzian, A.: Wi-fi enabled sensors for internet of things: A practical approach. Communications Magazine, IEEE **50**(6) (June

2012) 134 –143
2. IEEE: ANSI/IEEE Std 802.11, 1999 Edition (r2003). (2003) i –513
3. IEEE: IEEE std 802.11e-2005 (amendment to ieee std 802.11-1999. (2005) 0–189
4. IEEE: IEEE std 802.11n-2009 (amendment to ieee std 802.11-2007). (2009) 1 –565
5. Camps-Mur, D., Gomony, M.D., PéRez-Costa, X., Sallent-Ribes, S.: Leveraging 802.11n frame aggregation to enhance qos and power consumption in wi-fi networks. Comput. Netw. **56**(12) (August 2012) 2896–2911
6. Tan, E., Guo, L., Chen, S., Zhang, X.: Psm-throttling: Minimizing energy consumption for bulk data communications in wlans. In: Network Protocols, 2007. ICNP 2007. IEEE International Conference on. (October 2007) 123 –132
7. Adams, J., Muntean, G.M.: Adaptive-buffer power save mechanism for mobile multimedia streaming. In: IEEE International Conference on Communications, 2007. ICC '07. (June 2007) 4548 –4553
8. Palit, R., Naik, K., Singh, A.: Impact of packet aggregation on energy consumption in smartphones. In: Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International. (July 2011) 589 –594
9. Skordoulis, D., Ni, Q., Chen, H.H., Stephens, A., Liu, C., Jamalipour, A.: Ieee 802.11n mac frame aggregation mechanisms for next-generation high-throughput wlans. Wireless Communications, IEEE **15**(1) (February 2008) 40 –47
10. Lorchat, J., Noel, T.: Reducing power consumption in ieee 802.11 networks. In: Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), IEEE International Conference on. (August 2005) 106 – 110 Vol. 2
11. Kennedy, M., Ksentini, A., Hadjadj-Aoul, Y., Muntean, G.M.: Adaptive energy optimization in multimedia-centric wireless devices: A survey. Communications Surveys Tutorials, IEEE **PP**(99) (2012) 1 –19
12. Naik, K.: A survey of software based energy saving methodologies for handheld wireless communication devices. Tech. Report No. 2010-13, Dept. of ECE, University of Waterloo (2010)
13. Pathak, A., Hu, Y.C., Zhang, M.: Where is the energy spent inside my app?: fine grained energy accounting on smartphones with eprof. In: Proceedings of the 7th ACM european conference on Computer Systems. EuroSys '12, New York, NY, USA, ACM (2012) 29–42
14. Dogar, F.R., Steenkiste, P., Papagiannaki, K.: Catnap: exploiting high bandwidth wireless interfaces to save energy for mobile devices. In: Proceedings of the 8th international conference on Mobile systems, applications, and services. MobiSys '10, New York, NY, USA, ACM (2010) 107–122
15. Zhu, Y., Lu, H., Leung, V.: Access point buffer management for power saving in ieee 802.11 wlans. Network and Service Management, IEEE Transactions on **9**(4) (December 2012) 473 –486
16. Varga, A., Hornig, R.: An overview of the omnet++ simulation environment. In: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops. Simutools '08, ICST, Brussels, Belgium, Belgium, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) (2008) 60:1–60:10
17. Van der Auwera, G., David, P., Reisslein, M.: Traffic and quality characterization of single-layer video streams encoded with the h.264/mpeg-4 advanced video coding standard and scalable video coding extension. Broadcasting, IEEE Transactions on **54**(3) (September 2008) 698 –718