*Actas de las XIII Jornadas*
*de Ingeniería Telemática*
*(JITEL 2017),*
*Valencia (España), 27-29 de*
*Septiembre de 2017.*

ISBN: 978-84-9048-595-8
DOI: http://dx.doi.org/10.4995/JITEL2017.2017.6492

# Improving the energy efficiency of VoIP applications in IEEE 802.11 networks through control of the packetization period

Rafael Estepa, Antonio Estepa and German Madinabeitia.
Departmento de Ingeniería Telemática
Universidad de Sevilla
C/ Camino de los descubrimientos s/n. 41092 Sevilla.
{rafaestepa,aestepa,german}@us.es
Mark Davis
School of Electronic and Communications Engineering.
Dublin Institute of Technology.
Kevin Street, Dublin. Ireland.
mark.davis@dit.ie

***Abstract*—This paper presents an adaptive algorithm that improves the energy efficiency of VoIP applications over IEEE 802.11 networks. The algorithm seeks to achieve the largest energy savings subject to reaching a minimum speech quality under the prevailing network conditions. The control mechanism used is the dynamic selection of the packet size during the communication.**

**This algorithm has been implemented in an experimental testbed and the results demonstrate that our packetization period control algorithm can provide energy savings in uncongested IEEE 802.11 networks (up to 30%). Furthermore, under poor network conditions the algorithm can prolong the duration of the call before it is dropped at the expense of a higher energy consumption.**

***Palabras Clave*—Energy efficiency, VoIP, IEEE 802.11 jitel, telemática**

## I. Introduction

Increasingly powerful smartphones and wider deployment of free Wi-Fi coverage have boosted the use of VoIP applications over IEEE 802.11 networks (VoWiFi). Preserving VoWiFi quality of service (QoS), along with saving battery in the portable device are two research fields with abundant literature predominantly focused on the link layer [1], [2], [3].

VoIP is a real-time application and as such, small end-to-end delays (typically less than 300 ms) and low packet loss (typically between 1 and 5% depending on the codec) must be maintained during communication in order to ensure acceptable speech quality [4]. The challenge of preserving quality in VoIP over Wi-Fi has been extensively addressed in literature. Research efforts to support quality in VoWiFi have been focused on two main approaches: (a) dimensioning works aimed at finding the maximum number of simultaneous VoIP flows that IEEE 802.11 networks can accommodate while satisfying QoS constraints (e.g. network delay, packet loss ratio) [5], [6], [7], [8], [1], [2]; and (b) link-layer proposals aimed at meeting QoS constraints in the IEEE 802.11 network by finding optimum values for MAC-layer variables such as contention window size, maximum retry limits, etc. [9], [10], [11], [12]. As shown in a recent survey [13], VoIP software can also deal with QoS by dynamically handling application-layer variables such as codec choice, or number of speech frames encapsulated into an IP datagram ($N_f$), which depends on the packetization period and the codec inter-frame period.

Since VoIP software frequently runs on battery-powered terminals, an emergent research topic concerned with VoIP energy efficiency in IEEE 802.11 networks has been developed over recent years. A number of proposals are based on minimizing the time spent in active states (i.e.TX,RX) through MAC-layer based strategies [3]. However, as found in [14], [3], a VoWiFi device typically spends less than 2% of its time in these active states, which justifies that most research efforts are focused on maximizing the sleep time during idle periods whilst avoiding quality impairments through fine tuning of the Power Saving Mode (PSM) parameters. In [12] the authors propose a sleep strategy that dynamically adjusts the sleep time and packetization interval according to the collision probability to achieve a trade-off between energy saving and VoIP

EDITORIAL UNIVERSITAT POLITÈCNICA DE VALÈNCIA

capacity in ideals IEEE 802.11 channels; in [15] the authors schedule sleep and wake-up intervals to save energy based on end-to-end network delay and packet loss; in [16] an adaptive U-APSD (unscheduled automatic power save delivery) is proposed to achieve a certain delay constrain for each access category in IEEE 802.11 PSM.

Although some effort has been made to achieve energy efficiency subject to a minimum QoS level through setting IEEE 802.11 link-layer parameters (e.g. delay in [17]), to the best of our knowledge no attempts have been made to simultaneously deal with energy efficiency and QoS in VoIP by exerting control of application-layer parameters (i.e. codec and/or packetization period). This work aims to be a first approach to address the problem of saving energy associated to the execution of VoIP software subject to QoS restrictions by controlling the packetization period of VoIP during the communication. We believe that our solution is compatible with and complementary to other approaches such as dynamic codec setting [18] or optimization of IEEE 802.11 MAC parameters [17].

The remainder of the paper is as follows. Section II overviews the dependencies between QoS and energy efficiency in VoWiFi. Section III states the problem addressed and the scope of our study. Section IV details the proposed control algorithm. Section V addresses the test-bed used and implementation aspects. Results are presented in Section VI. Finally, Section VII concludes the paper.

## II. Relationship between Packetization Period, Quality and Energy Efficiency in VoWiFi

The foundation of our proposal is based on the interdependence of QoS and energy efficiency in VoWiFi. Thus, it is worth analyzing the relationships between the number of speech frames encapsulated on each packet (i.e. *packetization period* from now on), VoWiFi quality and IEEE 802.11 energy efficiency before stating our problem in more detail.

QoS in VoIP can be assessed during conversation time through the widely accepted E-Model [13]. The outcome of such model is a score from 0 to 100 termed the *R* factor which, in its simplest form, can be expressed as:

$$R = 92.8 - I_d(codec, delay) - I_{e,eff}(codec, loss, PLB) \tag{1}$$

where the factor $I_d$ accounts for the effect of delay, and $I_{e,eff}$ is associated with codec compression, packet loss rate and packet loss behavior (PLB)(i.e. burst ratio as defined in[13]). The delay factor can be broken down into network delay, and terminal delay (e.g. codec look-ahead, packetization period or jitter buffer). The packet lost impairment is attributable to lost packets in the network, as well as discarded packets due to delays greater than the jitter buffer of the receiver.

Figure 1 illustrates basic relationships between the packetization period ($N_f$) and QoS ($R$ factor) in VoIP. High values of $N_f$ reduce the packet generation frequency (i.e. transmission opportunities in the IEEE 802.11 network) and hence the traffic load. However, it also increases the overall delay (due to the frame generation period at the terminal) and, as packet size is increased, the probability of frame error increases. Large packets also have a negative effect in IEEE 802.11 networks if low Signal to Noise Ratio (SNR) or hidden node problems are present. Therefore, we can conclude that when the Wi-Fi network operates under poor performance conditions (due to MAC layer congestion or transmission problems - i.e. low SNR or hidden node problem), there is no clear predictable outcome from increasing or decreasing $N_f$ as high values can help reduce the MAC traffic load, but may result in lower performance. Observe that both impairments (congestion and transmission problems) can often exhibit high temporal variability.

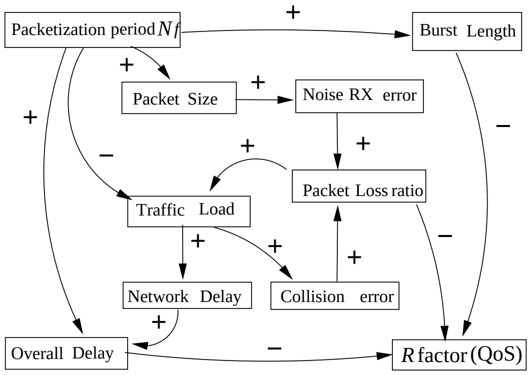

Fig. 1. Relationship between $N_f$ and QoS in VoWiFi.

The energy consumption at the IEEE 802.11 interface can be expressed as:

$$E = P_{tx} \cdot t_{TX} + P_{rx} \cdot t_{RX} + P_{idle} \cdot t_{idle} + P_{sleep} \cdot t_{sleep} \tag{2}$$

Where $P_{tx} > P_{rx} > P_{idle} > P_{sleep}$ represent the power coefficients at the radio interface for transmission, reception, idle and sleep respectively; and $t_{TX}, t_{RX}, t_{idle}, t_{sleep}$ the respective time spent in each state. Nominal values of $P_{tx}$ and $P_{rx}$ are similar in most Wi-Fi cards, and $P_{idle}$ is about 75% of $P_{tx}$ [3]. VoWiFi terminals typically spend less than 2% of their time [14] in active states (i.e. TX or RX). Consequently, the best strategy to achieve significant energy savings is to use PSM and go into the sleep state for as long as possible (as $P_{sleep}$ is significantly smaller than all the others).

In general, increasing $N_f$ (i.e. reducing the packet generation frequency) results in less time spent in active states, reducing the energy consumption. However, in lossy or saturated IEEE 802.11 channels, there exists no such straightforward relationship since larger packets are more prone to suffer transmission errors which will lead to retransmission attempts.

## III. Scope and Problem Statement

Our target scenario includes VoWiFi terminals connected through a IEEE 802.11 Access Point to the Internet. The respective communication partners are VoIP terminals operating in a wired environment such as POTS or switched Ethernet (e.g. a call center). This is the scope of the study which has also been reproduced in the test-bed. In this context, the VoWiFi terminals are assumed to periodically receive QoS-related information from their wired counterparts through various RTCP reports as defined in RFC 3550: Extended(XR), Sender(SR) and Receiver(RR). Upon reception of a report VoWiFi terminals trigger a procedure to determine the value of $N_f$ to be set on both sides of the call for the next period through a SIP RE-INVITE message.

In such a scenario, our goal is to allow VoWiFi users minimize their energy consumption at the IEEE 802.11 interface as long as a minimum target speech quality ($R_{\text{min}}$) is reached in the VoIP communication. Otherwise, the main goal is to reach $R_{\text{min}}$.

## IV. Three-step Optimization Procedure

Upon the reception of each RTCP report VoWiFi terminals execute the following procedure.

- **Step 1: Assessing VoIP quality**.
  Counters from RTCP reports along with some locally-obtained information (e.g. packetization and jitter buffer delay) allow the estimation of the network delay, end-to-end delay, packet lost rate and loss burst size. Using this information, each VoWiFi station estimates the QoS of the past period using Eq. 1 as done in [19] from the data obtained from RTCP reports. The output of this stage is the value of $R$ .
- **Step 2: Estimation of Wi-Fi network conditions**
  We assume that in our scenario packet loss and delay come mainly from the IEEE 802.11 network [7], so when the network performance exceeds certain thresholds (e.g. a packet loss rate $> 5\%$ or a delay $> 300ms$) it can be inferred that stations are saturated due to lack of transmission opportunities, which will be indicated by setting flag SAT (stations saturated). The network delay can be obtained by measuring the time difference between Sender and the corresponding Receiver RTCP reports. Network losses can be estimated from the Ratio Loss counter used in RTCP XR reports.
- **Step 3: Finding $N_f$ for the next interval**
  The results from steps 1 (R) and 2 (SAT) are used as input for an algorithm that determines the optimal value of $N_f$ for the next period. A stateful behaviour is needed to keep the values of R, SAT and $N_f$ between two consecutive executions $i-1$ and $i$.
  Figure 2 shows the proposed algorithm[1]. The flow diagram follows the discussion in Section II. If $R \geq R_{min}$ we increase $N_f$ in order to get higher energy saving at the cost of reducing the QoS ($R$ factor) due to the overall delay increment. However, when $R < R_{min}$ the priority will be to restore $R$ to acceptable values. If flag SAT was off then we could conjecture that the poor quality is caused by excessive packetization delay, so we will decrease $N_f$. Conversely, if we observe poor network performance (flag SAT is on) then we perform a heuristic search to find out the $N_f$ value that improves the R factor (recall that no a-priori outcome can be foreseen as discussed in Section II). In this case, if increasing (or decreasing) $N_f$ improves the $R$ factor, the algorithm will continue increasing (or decreasing) $N_f$ until $R_{min}$ is achieved.

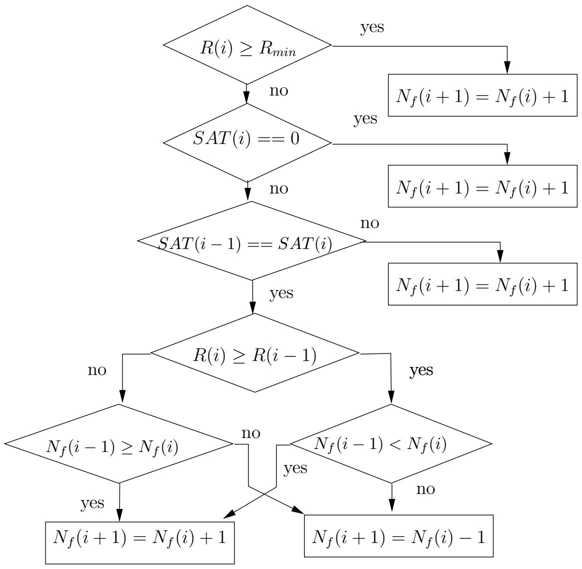

Fig. 2. Flow diagram of step 3

## V. Testbed and Implementation

### A. Scenario set-up

The test scenario is shown in Figure 3. Stations 1 to 4 are equipped with a D-link DWA-131 Wi-Fi card managed by the *ndiswrapper* driver v1.59-6 under Linux Ubuntu distribution v14.04.2. These wireless stations communicate with their wired peers connected to an Ethernet 100Mbps switch. The wireless stations are associated with an Access Point (Airlink DWL-3500AP) set to operate in the IEEE 802.11b with a PHY rate set at 1 Mbps so we can easily experiment with saturation conditions. Finally, a PC acting as router between the Access Point and the Switch is running the *Network Emulator for Windows Toolkit* (NEWT) software to emulate adverse network conditions (i.e. to insert delay or packet losses into the network).

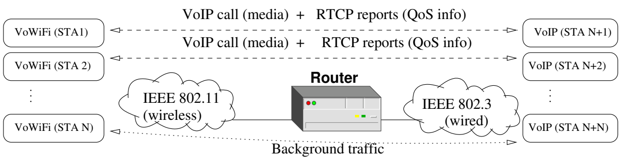

Fig. 3. Test scenario implemented.

Stations 1 to 3 and its peers 5 to 7 are running the VoIP software *pjsip* (the *pjsua* command-line client) [20] containing the modifications necessary to accommodate our packetization period control algorithm from Section IV. Each communication pair uses a 30 min conversation

[1] A fixed step size has been used here for simplicity. However, adjustable step size could be used for faster adaptation.

playback from recorded telephony conversations (LDC data bank, CallHome recordset) [21], each station playing its respective side of the conversation. In our tests, we use the codec G.711 with VAD. Stations 4 and 8 are dedicated to inserting background traffic. Both use *rtptools* [22] as traffic generator to send and receive UDP datagrams to each other.

### B. Algorithm implementation

An optimization module that implements the algorithm from Section IV has been written in C language. As illustrated in Figure 4, this module communicates with the VoIP client *pjsua* rather than being integrated into it. Thus, changes in the original *pjsua* VoIP client consist of reporting QoS-related information to our module, reading the algorithm output ($N_f$) and sending a SIP RE-INVITE message to its peer so that the new packetization period can take effect. VoIP clients can be configured to locally apply the new value of $N_f$ or to apply the value received from the other side of the call.

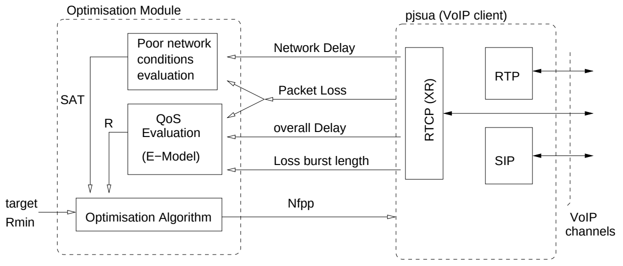

Fig. 4. Module architecture implemented.

QoS-related information is periodically evaluated by *pjsua* and transported in various RTCP reports such as Extended (XR), Sender (SR) and Receiver (RR). By taking advantage of this functionality already implemented, we can simply use these counters and complement them with some additional locally-obtained information to get the input variables for our algorithm. In contrast to the information included in the RTCP reports, these counters are reset between successive reports which allows for an evaluation of the quality of service for during an interval. The QoS evaluation period configured in our tests is 7.64 secs. Small intervals (i.e. $< 5$ secs.) provide uneven and coarse-grained loss rates due to insufficient number of packets generated (especially with large $N_f$ values), whereas large intervals (e.g. $\geq$ 10secs.) provide a poor reaction time. Therefore, we experimentally found this value to be a good balance.

The QoS variables used by the algorithm are as follows:

- Network Delay. When a SR is transmitted we measure the time until we get the corresponding RR. This round-trip time is corrected by subtracting the processing time at the far end (as indicated in the RR). The network delay is obtained by halving this corrected round-trip time.
- End-to-End delay ($d$). This is the delay resulting from adding network delay, jitter buffer and packetization period. The packetization period and jitter buffer delay are locally measured by the *pjsua* application every time it reads or writes a new speech frame to the buffer.
- Packet loss rate. This is the overall packet loss resulting from network losses and discarded packets at the jitter buffer. Network losses are calculated from the *Ratio Loss* counter used in the RTCP XR reports. Discarded packets at the jitter buffer are already computed by *pjsua*.
- Loss burst size. This is measured by default in *pjsua* and is included in RTCP XR reports.

Using the previous variables, the optimization module evaluates the following:

- Network conditions flag ($SAT$) which is set to 1 when either the Network Delay or the Packet loss rate exceeds a certain threshold (300 ms or 5% respectively in our tests).
- *R* factor. Calculated using Eq. 1 where $I_d$ is calculated from the end-to-end delay $d$ as $0.0024 \cdot d + 0.11(d - 177.3)H(d - 177.3)$ being $H(x)$ the step function (Heavyside function); $I_{e,eff}$ is calculated as $I_e + (95 - I_e) \cdot P_{pl}/(P_{pl}/\text{BurstR} + B_{pl})$, where $I_e$ is a measure of the codec intrinsic speech quality degradation, $P_{pl}$ is the loss rate percentage, BurstR is the ratio of the registered packet loss burst size over random losses, and $B_{pl}$ measures the loss concealment of each codec. Tables with values for the constants $B_{pl}$ and $I_d$ can be found in [19].

After evaluating $SAT$ and $R$ the rest of the algorithm as described in Section IV is executed. In our results we have used a maximum value of 16 and a minimum value of 2 (i.e. 20 ms) for the packetization period. The output of the algorithm is the new value of $N_f$. The wired VoIP client executes the algorithm, applies the resulting $N_f$ and sends this same value to its wireless counterpart which also applies it.

### C. Energy Measurement procedure

In order to measure the energy consumption of each WiFi station 1-3 during the VoIP calls we have used a dedicated device running Wireshark to capture all the physical frames (including management frames) over the air interface during each experiment. Captured frames have been processed afterwards with *awk* to calculate for each station the time spent in every energy state (i.e. TX, RX, IDLE, SLEEP) over intervals of 10 secs. The energy consumption over such interval is then calculated using Eq. 2 with the Wi-Fi adapter power coefficients $1.65, 1.2, 0.9$ and $0.1$ Watts for the respective states. Although Wi-Fi adapter power coefficients exhibit high variability, those values can be found in Wireless Adapter from vendors like Intel or D-Link ( [23])

## VI. Results

Two experiments have been carried out to test the goodness of the proposed algorithm in both uncongested and congested IEEE 802.11 network conditions. Results represent average values from stations 1 to 3.

### A. *Unloaded Network Scenario (no background traffic)*

Figure 5 shows the variation of QoS, delay and energy over the first 700 secs of the conversation for different quality targets ($R_{min}$). Fig. 5(a) shows the value of $R$ and $N_f$ (as calculated by the algorithm) during the test. Note that we have reduced $R_{min}$ from 90 to 85 and 80 at 300 secs and 550 secs respectively, which resulted in increments of the packetization period up to its maximum value of 16. As observed in Fig 5(c) the increment in $N_f$ results in energy savings at the IEEE 802.11 interface with respect to the default constant value of 2 (20ms) recommended in RFC5761. The bottom part of Fig 5(c) shows savings as large as 30% when PSM is used (approximately 1 Joule of difference), whereas without PSM only 1% is saved which can be attributed to the fact that for 98% of the time Wi-Fi cards are in idle mode as observed in [14]. Fig. 5(b) shows the end-to-end and network delays. The fluctuations in the former can be mostly attributed to the playout buffer management performed by *pjsip*, which dynamically sets the buffer size according to network conditions or discarded packets. For large values of $N_f$ (e.g. 16) a small increment in the buffer size (e.g. 18 speech frames) causes waiting for various packets to arrive (i.e. a packetization delay of 16x2 speech frames) while the next time only one packet is necessary.

### B. *Congested Network Scenario (background traffic)*

In this test background traffic is inserted by sending 100B packets with an increasing frequency resulting in a bit rate from 450 kbps to 650 kbps in steps of 20 kbps and back to its initial value. Packet loss and network delay peak at 20% and 600 ms respectively. Such conditions would prevent any form of acceptable communication and in practice such a VoIP call would be torn down after a few seconds.

We have run the test three times using our algorithm with $R_{min} = 90$, $R_{min} = 75$, and the RFC5761 default constant packetization ($N_f$=2). Figure 6 shows the results. As expected, while the network conditions allow us to achieve $R_{min}$, the packetization period increases gradually to save energy. However, as traffic increases and the network conditions deteriorate (the SAT flag is on), the algorithm adjusts $N_f$ to preserve the target quality. But as the background traffic continues to increase, there is a point when it is no longer possible to maintain the quality and it quickly drops to zero. This point is first reached for the case of the default packetization period at around t = 450 secs and 490 kbps of background traffic, next for the algorithm-controlled with $R_{min} = 90$ at around t = 600 secs and 510 kbps of background traffic, and finally for the algorithm-controlled with $R_{min} = 75$ at around t = 1,150 secs and 570 kbps of background traffic. When the background traffic returns to lower values and channel conditions improve, we can observe $R$ returning to acceptable values. Therefore, the use of our proposed algorithm allows longer periods of acceptable quality, typically 48% of the time for the more relaxed target quality and 16% for the more demanding target. The packetization period explains the difference in the quality obtained for each $R_{min}$. To achieve a higher *R* value, $N_f$ is reduced to its minimum value in order to realize the lowest packetization delay possible. This increases the traffic inserted by the VoIP application and consequently causes stations to saturate faster. This is even worse with the default packetization value of 2.

Regarding the energy consumption, we can observe in Figure 6(b) that the algorithm-controlled packetization schemes achieve higher energy efficiency than the default packetization period. Moreover, the lower the minimum target quality that one is prepared to accept the greater the energy savings that can be realized. From Figure 6(b) it can be observed that there is a difference of 11% in the $R$ value between 600 and 1,150 secs. This difference is greater with respect to the default packetization period as expected (i.e. around 20%). During prolonged saturated conditions there is no significant differences in the energy consumption between both targets as both seek to maximize the packetization period. At any rate, the energy consumption during this period is not relevant as in practice the call would be torn down by users after few seconds due to the lack of a minimum quality.

## VII. Conclusions and further work

From our experiments we have demonstrated that the application of our packetization period control algorithm can provide energy savings in VoWiFi terminals at the cost of reducing the VoIP quality down to a minimum acceptable level. In uncongested IEEE 802.11 networks with low delay and loss conditions, we continue to experience acceptable QoS levels and also obtain significant energy savings of up to 30% using PSM depending of the terminal characteristics.

Under poor network conditions, our algorithm allows the extension of experiencing the target quality at the cost of a higher energy consumption when compare with the uncongested scenario. In the worse case we can get no energy saving, but on these network conditions the VoIP call should be torn down. Therefore by adaptatively controlling the packetization period we can tolerate saturated channel conditions for longer periods.

Further works includes dynamic codec selection and variable $N_f$ increments as part of the optimization strategy.

## References

[1] S. Harsha, A. Kumar, and V. Sharma, “An analytical model for performance evaluation of multimedia applications over edca in an ieee 802.11e wlan,” *Wirel. Netw.*, vol. 16, no. 2, pp. 367–385, Feb. 2010. [Online]. Available: http://dx.doi.org/10.1007/s11276-008-0137-y

[2] K.-W. Chin, “On maximizing voip capacity and energy conservation in multi-rate wlans,” *Communications Letters, IEEE*, vol. 14, no. 7, pp. 611–613, 2010.

[3] T. Shiao-Li and H. Chung-Huei, “A survey of energy efficient mac protocols for ieee 802.11 wlan,” *Computer Communications*, vol. 34, no. 1, pp. 54 – 67, 2011.

[4] R. G. Cole and J. H. Rosenbluth, “Voice over ip performance monitoring,” *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 2, pp. 9–24, Apr. 2001.

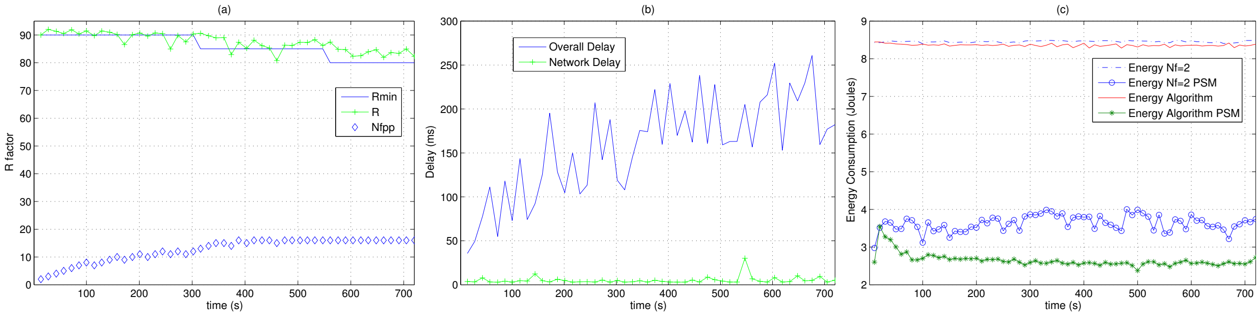

Fig. 5. Scenario without background traffic: evolution of (a) $R$ and $N_f$, (b) delay and (c) energy over the first 700 secs.

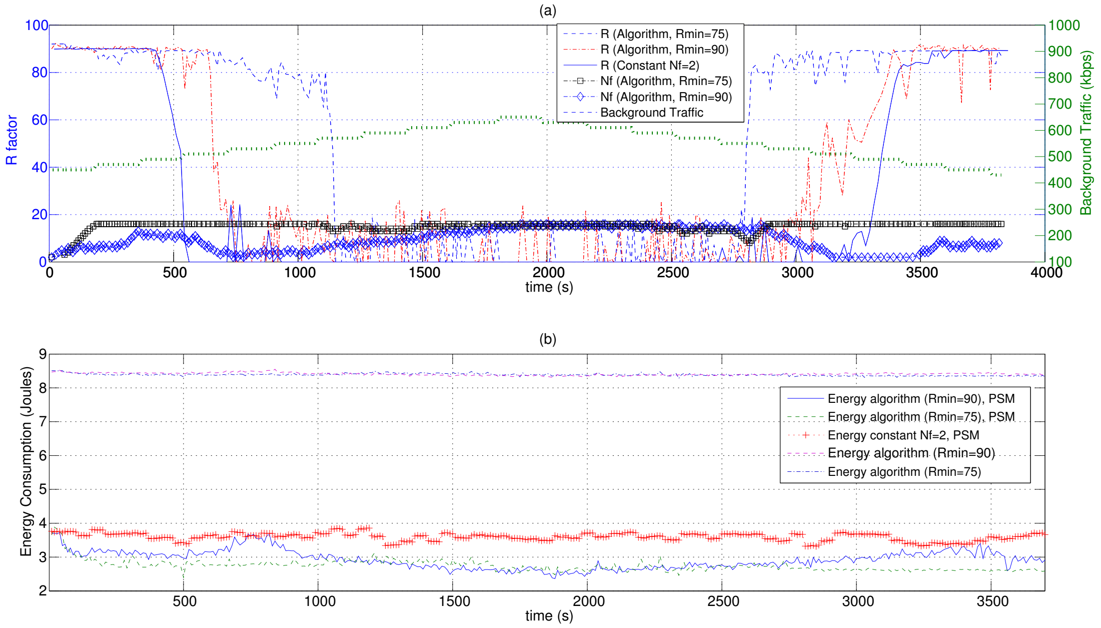

Fig. 6. Scenario with background traffic (a) quality and packetization period for different packetization schemes; (b) energy consumption with different packetization schemes.

[5] A. Trad, F. Munir, and H. Afifi, “Capacity evaluation of voip in ieee 802.11e wlan environment,” in *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*, vol. 2, 2006, pp. 828–832.

[6] L. Cai, X. Shen, J. W. Mark, and Y. Xiao, “Voice capacity analysis of wlan with unbalanced traffic,” in *Quality of Service in Heterogeneous Wired/Wireless Networks, 2005. Second International Conference on*, 2005, pp. 8 pp.–9.

[7] S. Shin and H. Schulzrinne, “Measurement and analysis of the voip capacity in ieee 802.11 wlan,” *Mobile Computing, IEEE Transactions on*, vol. 8, no. 9, pp. 1265–1279, 2009.

[8] G. Kuriakose, S. Harsha, A. Kumar, and V. Sharma, “Analytical models for capacity estimation of ieee 802.11 wlans using dcf for internet applications,” *Wirel. Netw.*, vol. 15, no. 2, pp. 259–277, Feb. 2009. [Online]. Available: http://dx.doi.org/10.1007/s11276-007-0051-8

[9] G. Boggia, P. Camarda, L. Grieco, and S. Mascolo, “Feedback-based control for providing real-time services with the 802.11e mac,” *Networking, IEEE/ACM Transactions on*, vol. 15, no. 2, pp. 323–333, 2007.

[10] G. Nikolakopoulos, A. Panousopoulou, and A. Tzes, “Experimental controller tuning and qos optimization of a wireless transmission scheme for real-time remote control applications,” *Control Engineering Practice*, vol. 16, no. 3, pp. 333 – 346, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066107001074

[11] K. Stoeckigt and H. Vu, “Voip capacity analysis in ieee 802.11 wlan,” in *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, 2009, pp. 116–123.

[12] S.-L. Tsao and C.-H. Huang, “An energy-efficient transmission mechanism for voip over ieee 802.11 wlan,” *Wireless Communications and Mobile Computing*, vol. 9, no. 12, pp. 1629–1644, 2009. [Online]. Available: http://dx.doi.org/10.1002/wcm.747

[13] L. S. G. D. Carvalho and E. D. S. Mota, “Survey on application-layer mechanisms for speech quality adaptation in voip,” *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, p. 36, 2013.

[14] A. J. Estepa, J. M. Vozmediano, J. López, and R. M. Estepa, “Impact of voip codecs on the energy consumption of portable devices,” in *Proceedings of the 6th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*. ACM, 2011, pp. 123–130.

[15] V. Namboodiri and L. Gao, “Energy-efficient voip over wireless lans,” *Mobile Computing, IEEE Transactions on*, vol. 9, no. 4, pp. 566–581, 2010.

[16] X. Pérez-Costa, D. Camps-Mur, and A. Vidal, “On distributed power saving mechanisms of wireless lans 802.11 e u-apsd vs 802.11 power save mode,” *Computer Networks*, vol. 51, no. 9, pp. 2326–2344, 2007.

[17] P. Serrano, A. Banchs, P. Patras, and A. Azcorra, “Optimal configuration of 802.11 e edca for real-time and data traffic,” *Vehicular*

*Technology, IEEE Transactions on*, vol. 59, no. 5, pp. 2511–2528, 2010.

[18] A. J. Estepa, R. Estepa, J. Vozmediano, and P. Carrillo, "Dynamic voip codec selection on smartphones," *Network Protocols and Algorithms*, vol. 6, no. 2, pp. 22–37, 2014.

[19] A. Kovac, M. Halas, M. Orgon, and M. Voznak, "E-model mos estimate improvement through jitter buffer packet loss modelling," *Advances in Electrical and Electronic Engineering*, vol. 9, no. 5, pp. 233–242, 2011.

[20] S. PJSIP-Open Source, "Stack and media stack for presence, im/instant messaging, and multimedia communication, 2008."

[21] C. Alexandra, D. Graff, and G. Zipperlen, "Callhome american english speech," *Linguistic Data Consortium, Philadelphia*, 1996.

[22] H. Schulzrinne, P. Pan, A. Tsukamoto, D. Sisalem, and S. Casner, "Rtp tools," *URL: ftp://ftp. cs. columbia. edu/pub/schulzrinne/rtptools/rtptools. html*, 1996.

[23] S. Chiaravalloti, F. Idzikowski, and L. Budzisz, "Power consumption of wlan network elements," no. TKN-11-002, 2011.