Macrothink Institute™

Network Protocols and Algorithms
ISSN 1943-3581
2017, Vol.9, No.3-4

Adaptative control of packetization to improve the energy efficiency of VoIP applications in IEEE 802.11 networks

Antonio Estepa, Rafael Estepa, Vicente Mayor, Germán Madinabeitia

Department of Telematics Engineering, Universidad de Sevilla, Spain

Tel: +1-954-487-384 E-mail: aestepa@us.es, rafaestepa@us.es, german@us.es, vmayor93@gmail.com

Mark Davis

School of Electronic and Communications Engineering.

Dublin Institute of Technology, Dublin

Tel: +35 -314-023-000 E-mail: mark.davis@dit.ie

Received: Oct. 15, 2017 Accepted: Dec. 20, 2017 Published: December 31, 2017

DOI:10.5296/npa.v9i3-4.12308 URL:https://doi.org/10.5296/npa.v9i3-4.12308

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. Three variants of the control algorithm are presented and compared.

Our proposal has been implemented in a VoIP application and experimentally validated in a 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 higher energy consumption during such conditions. Each variant of the algorithm provides a unique balance between keeping quality and saving energy.

Keywords: Energy efficiency, VoIP, IEEE 802.11, codec, multimedia.

1 Introduction

The use of VoIP applications in IEEE 802.11 networks (VoWiFi) has recently experienced a tremendous growth in both domestic and commercial environments. This can be partially traced back to increasingly powerful smartphones and communication apps, as well as a wider deployment of hotspots extending free Wi-Fi coverage to more and more locations.

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 [1]. In addition, 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. Both challenges: preserving VoWiFi quality of service (QoS) and saving battery in the portable device, are two mature research fields that count with abundant literature predominantly focused on the link layer [2, 3, 4].

From our perspective, most proposals from either the QoS or energy efficiency research fields focused on link-level solutions share common weaknesses. Firstly, the majority of these are based on theoretical models which are valid only under specific and ideal conditions (e.g. known external noise or traffic patterns). Secondly, most works only consider the optimization of MAC-layer variables for either QoS or energy efficiency, ignoring the relationships and trade-off between both. Thirdly, validation is performed in many cases by means of inaccurate simulations [5] under ideal network conditions that fail to consider real IEEE 802.11 operating scenarios with dynamic noise and background traffic. Finally, a number of link-layer proposals require modifications to the IEEE 802.11 drivers which may be difficult to implement in practice and may damage interoperability between STAs and APs.

VoIP software offers application-level variables such as codec choice or the number of speech frames encapsulated per datagram (i.e. the packetization interval) which have a critical impact on the traffic pattern generated by VoIP applications. Consequently, dynamic control over such variables can also be an alternative way to cope with both the QoS and energy efficiency of the VoIP application. Moreover, employing application-level variables exhibits some advantages over link-level solutions: (a) it is compatible and can be used in conjunction with existing link-layer approaches, (b) it is easier to implement as some VoIP applications already allow for the configuration of the codec and/or packetization interval, (c) the assessment of QoS can include application-level factors such as playout buffer delay or speech codec.

This work aims to be a first approach in dealing with both QoS and energy efficiency problems through the dynamic control of application-level variables. Since codec switching solutions exhibit limitations (e.g. some are proprietary, it might be noticeable by the user), in this first step we propose the use of the variable *packetization interval* as a general solution independent of (and compatible with) the particular codecs implemented by each VoIP application. We do test various codecs and analyze its implications, but we do not base our solution on dynamic codec selection. In particular, we propose a simple but efficient algorithm to dynamically increase or decrease the number of speech frames encapsulated on each packet as the main mechanism to maximize energy efficiency at the Wi-Fi interface as long as a minimum target speech quality is guaranteed. Our proposal has been implemented in a VoIP application and experimentally validated in a testbed. We believe that our solution is compatible with and complementary to other approaches such as dynamic codec setting [6] or optimization of

IEEE 802.11 MAC parameters [7]. This paper presents an extended version of our conference paper [8].

The remainder of the paper is as follows. Section 2 presents related works. Section 3 overviews the dependencies between the packet size, QoS, and energy efficiency in VoWiFi. Section 4 states the problem addressed and the scope of our study. Section 5 details the proposed control algorithm. Section 6 addresses the testbed used and implementation aspects. Results are presented in Section 7. Finally, Section 8 concludes the paper.

2 Related Work

The challenge of preserving quality in VoIP over Wi-Fi has been extensively addressed in the 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) [9, 10, 5, 11, 2, 3]; 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. [12, 13, 14, 15]. As shown in a recent survey [16], VoIP software can also deal with QoS by dynamically handling application-layer variables such as codec choice, or the number of speech frames encapsulated into an IP datagram (N_f), which depends on the packetization period and the codec inter-frame period. Our proposal will set the focus on this variable N_f.

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 [4]. However, as found in [17, 4], 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 [15] 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 capacity in ideal IEEE 802.11 channels; in [18] the authors schedule sleep and wake-up intervals to save energy based on end-to-end network delay and packet loss; in [19] an adaptive U-APSD (unscheduled automatic power save delivery) is proposed to achieve a certain delay constraint 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 [7]), 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).

3 Preliminar analysis on how the packetization period impacts on VoIP Quality and Energy efficiency

The foundation of our proposal is based on the interdependence of QoS and energy efficiency in VoWiFi. Thus, it is worth exploring the relationship between the number of speech frames encapsulated in each packet (N_f), VoWiFi quality and energy efficiency at the IEEE 802.11 NIC before stating our problem in more detail.

3.1 Relation between N_f and VoWiFi Quality

QoS in VoIP can be assessed in real time by using single-ended perceptual methods such as ITU-T P.563, or computationally, by using network and codec-dependent parameters. The E-Model [20] is the most known representative method of the latter, being suitable for an accurate real-time monitoring assessment of speech quality in VoWiFi [16]. The outcome of the E-Model is a score ranging from 0 to 100 termed the *R* factor, where *R* values greater than 80 are commonly considered acceptable. In its simplest form, the E-Model can be expressed as the addition of the following additive factors:

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

where the factor I_d accounts for the negative effect of delay in VoIP interactivity, and $I_{e,eff}$ is associated with codec compression, packet loss rate and packet loss behavior (PLB). The delay impairment can be further broken down into network delay, and terminal delay (e.g. codec look-ahead, packetization, jitter buffer) [21]. The packet loss impairment can be attributable to lost packets in the network or IEEE 802.11 sub-network, as well as discarded packets as a result of delays greater than the jitter buffer .

A VoIP terminal generates constant size (L bits) speech frames at regular intervals (T sec.). Both L and T are codec-dependent, ranging between 10-38 bytes and 10-30ms respectively in most popular codecs [17]. Since L is usually quite small, it makes sense to encapsulate multiple speech frames into one IP packet to reduce overhead and hence increase the communication efficiency. RFC 5761 recommends a default inter-packet time of 20 ms for most codecs, which leads to the encapsulation of two VoIP frames per packet in codecs such as G.729 or G.711, or just one VoIP frame for codecs such as G.723, AMR or iLBC.

The relation between the packetization interval and the speech quality perceived by a VoWiFi user is sketched in Figure 1. Assuming a non-saturated ideal channel, high values of N_f produce fewer packets of a higher size to be transmitted over the network which can reduce the traffic load in ideally. However, increasing N_f also has a negative impact on the terminal delay through an increased delay of T sec. per every frame encapsulated. It can also increase the burst size by increasing the number of speech frames lost in the case of a corrupted packet. As a consequence, increasing N_f can have a negative impact on those terminal-related aspects of QoS but can have a positive impact on network-related QoS aspects under good IEEE 802.11 channel conditions.

However, under adverse network conditions, there is no clear predictable outcome from increasing or decreasing N_f. As shown in Figure 1, in saturated IEEE 802.11 channels, increasing N_f can help in reducing congestion as it reduces the traffic load. However, it also creates larger sized packets which can have a negative impact when transmission errors are high due to low SNR or hidden nodes.

3.2 Relation between N_f and Energy

In an IEEE 802.11 interface the energy consumption can be expressed as:

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

Where $P_{tx} > P_{rx} > P_{idle} > P_{sleep}$ represent the power coefficients at the radio interface

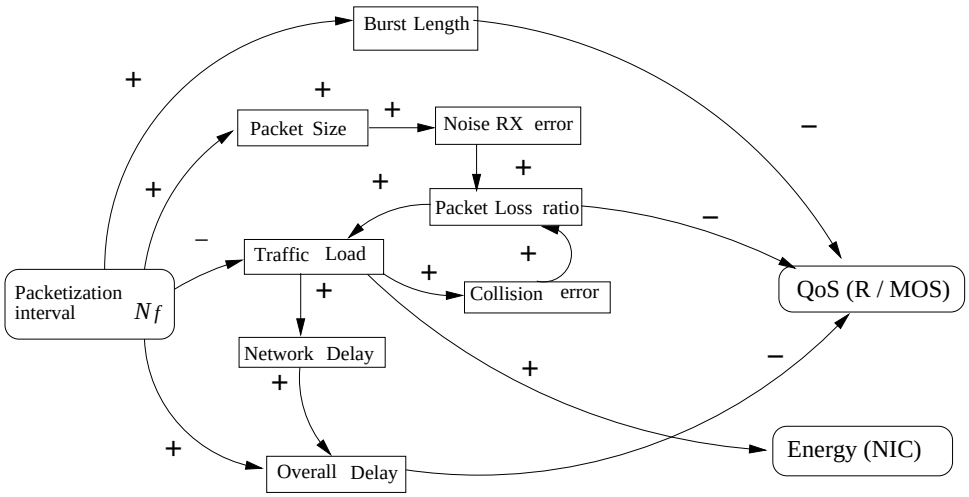

Figure 1: General relations between N_f, QoS and energy in VoWiFi.

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 current conventional Wi-Fi cards, and P_{idle} is somewhat high (at about 75% of P_{tx}) [4]. VoWiFi terminals typically spend less than 1% of their time [17] in the transmission state. Consequently, the best strategy to achieve significant energy savings is to use PSM and get terminals to go into the sleep state for as long as possible (as P_{sleep} is significantly smaller than all the others).

In general, in non-saturated and low loss IEEE 802.11 networks, increasing N_f will reduce the packet generation frequency, which will result in less time spent in the active modes (i.e. TX or RX), reducing the energy consumption. However, in lossy or saturated IEEE 802.11 channels, there exists no such straightforward relationship since higher size packets are more prone to suffer transmission errors which will lead to retransmission attempts.

4 Problem Statement and scenario

Our target scenario can be described as follows. VoWiFi terminals are connected through an IEEE 802.11 Access Point to infrastructure (i.e. wired Ethernet). Each VoWiFi station has a communication partner operating in the wired environment. In this context, VoIP clients are assumed to periodically receive QoS-related information from their counterparts via RTCP reports as defined in RFC 3550. Upon reception of these reports, each VoIP client triggers a procedure to determine the optimal value of the packetization interval (N_f) for the next period. In our implementation, we assume that the QoS perceived on one side applies to both sides and such side can act as a master instructing her communication partner to set the new value of N_f. It is assumed that SIP signaling is being used on both sides. Thus, if the packetization interval carried out differs from that of the current period, a SIP RE-INVITE message is sent to instruct the other edge of the new packetization interval. This is illustrated in Figure 2.

In the scenario previously described, the problem addressed is to dynamically set the best N_f that help VoWiFi clients to save energy constrained by reaching a minimum speech quality. More specifically, our goal is to allow VoWiFi users minimize their energy consumption at the

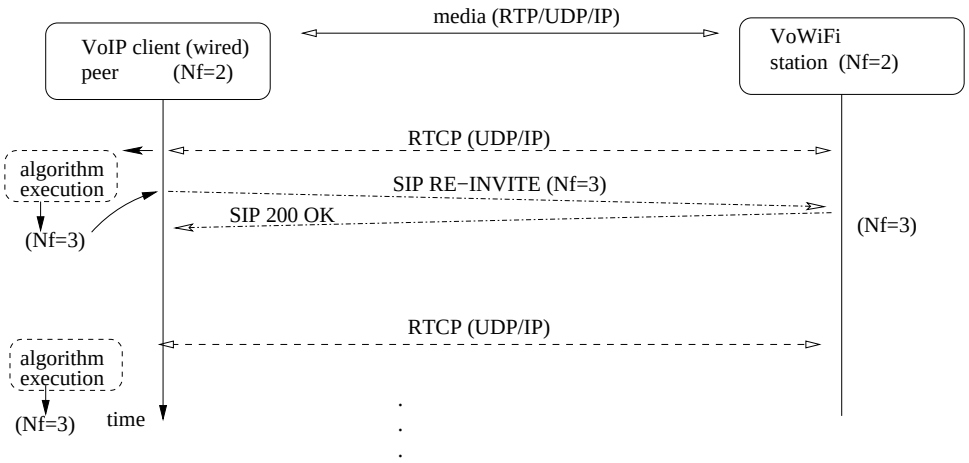

Figure 2: Example of operation of the optimization procedure.

IEEE 802.11 interface as long as a minimum target speech quality (R_{min}) is reached on the VoIP communication. Otherwise, the main goal is to achieve R_{min}.

5 Three-step Optimization Procedure

Upon the reception of periodical QoS reports, each VoIP client executes the following procedure to determine the optimal value of N_f.

- **Step 1: Assessing the QoS perceived during the last period**.

  Counters included in RTCP various reports (e.g. sender report, receiver report and extended report) plus locally-obtained information such as the jitter buffer average delay, allow one to estimate E-Model factors such as the network delay, the end-to-end delay, the packet loss rate and loss burst size. We will elaborate on this in the next section. Then, a VoIP client estimates the QoS experienced in the past period using (1) such as done in [22]. The output of this step is the value of R .

- **Step 2: Estimation of Wi-Fi network conditions**

  We assume that in our scenario packet loss and delay can be attributed mainly to the IEEE 802.11 network [5], so when the network performance exceeds a certain threshold (e.g. packet loss rate $> 5\%$ or 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).

  Then, the SAT flag will be evaluated periodically according to the network conditions derived from the RTCP reports. The network delay is 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 of an algorithm which determines the optimal value of N_f for the next period. A stateful behavior is needed to keep the values of R, SAT and N_f between two consecutive executions $i-1$ and i.

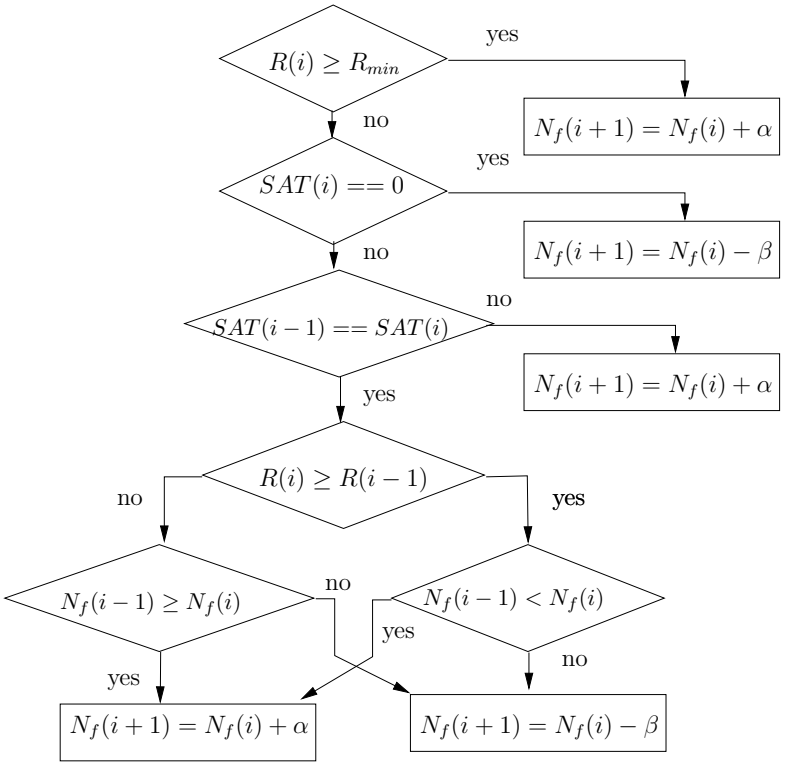

Figure 3: Flow diagram of step 3

Figure 3 shows the proposed algorithm. The flow diagram follows the discussion presented in Section 3. It can be summarized as follows:

- *If* $R \geq R_{min}$ our goal is to reduce energy consumption by decreasing the offered load. Note that increasing N_f produces a delay increment (end-to-end delay) as a result of a longer packetization period, reducing the QoS (R factor).
- If $R < R_{min}$ 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. However, if SAT was on (i.e. poor network conditions) 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 3). 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

Note that increments and decrements of N_f are achieved by adding α or subtracting β to the current value. These variables α, β let the creation of different variants of the algorithm. For example, $\alpha = \beta = 1$ will implement a smoother version while $\alpha = 1$, $\beta = 2$ is more conservative when it comes to increasing the packetization delay, but exhibits more aggressive adaptation to sudden network degradation.

6 Testbed and implementation

The target scenario described in Section 4 has been set up in a testbed. This section describes the experiment performed and elaborates on how steps from Section 5 have been im-

plemented.

6.1 Scenario set-up

The test set-up is shown in Figure 4. Stations 1 and 2 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 are associated with an Access Point (Airlink DWL-3500AP) configured to operate in IEEE 802.11b with a PHY rate set at 1 Mbps to facilitate experimenting with saturation conditions. A PC acting as the router between the Access Point and a 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). The router is connected to an Ethernet 100Mbps switch, which in turns connects the wired stations 3 and 4.

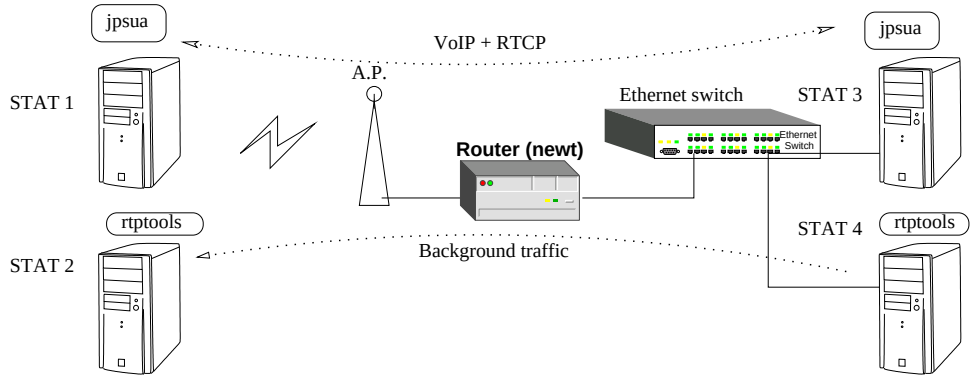

Figure 4: Testbed implemented.

Station 1 and its communication peer, station 3 are running the VoIP software *pjsip* (the *pjsua* command-line client) [23] modified to accommodate our packetization period control algorithm from Section 5. Each communication pair uses a 30 min conversation playback from recorded telephony conversations (LDC data bank, CallHome recordset) [24], each station playing its respective side of the conversation. In our tests, we use the codecs G.711 (speech frames each 10 ms) and iLBC30 (speech frames each 30 ms) using voice activity detection (VAD) to prevent the generation of traffic during unvoiced periods. Stations 4 is dedicated to sending background traffic to station 2 in order to saturate the wireless channel. Both use *rtptools* [25] to generate/receive traffic.

6.2 Integration of the packetization control algorithm in pjsua

An optimization module has been written in C language to implement the algorithm from Section 5. As illustrated in Figure 5, 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 the value received from the other side of the call.

6.2.1 QoS assesment

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

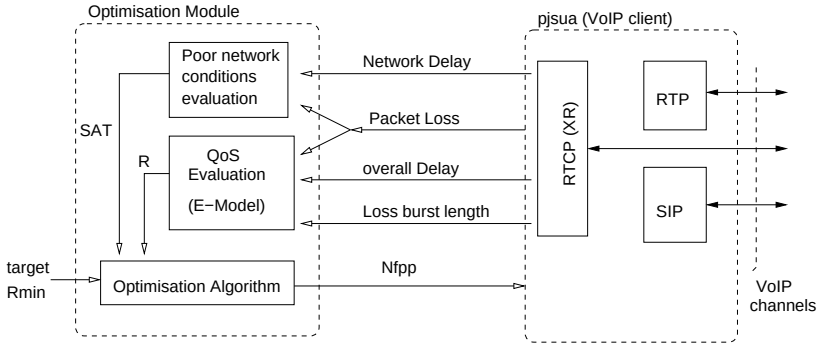

Figure 5: Module architecture implemented.

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 allow 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. 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 *R* factor. This is done using (1) where its components are calculated as follows:

I_d is calculated from the end-to-end delay as shown in (3) where $H(x)$ represents the step function (Heavyside function).

$$I_d = 0.0024 \cdot d + 0.11 \cdot (d - 177.3) \cdot H(d - 177.3) \quad (3)$$

And $I_{e,eff}$ is calculated as:

$$I_{e,eff} = I_e + (95 - I_e) \cdot P_{pl}/(P_{pl}/\text{BurstR} + B_{pl}) \quad (4)$$

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 [22].

6.2.2 Saturation (SAT) flag

The network conditions flag (SAT) is set to 1 when either the network delay or the packet loss rate exceeds a certain threshold. The threshold used in our tests was 300 ms and 5% respectively.

6.2.3 Estimation of N_f

After evaluating SAT and R the rest of the algorithm as described in Section 5 is executed. We have created the following three variants of the algorithm:

- slow: $\alpha = \beta = 1$.(i.e. $N_f(i+1) = N_f(i) + / - 1$)
- fast: $\alpha = \beta = 2$.(i.e. $N_f(i+1) = N_f(i) + / - 2$)
- tcp-like: $\alpha = 1$, $\beta = \lceil N_f(i)/2 \rceil$). (i.e. $N_f(i+1) = N_f(i) + 1$ or $N_f(i+1) = \lceil N_f(i)/2 \rceil$)

This will let us compare the performance of each version.

We have set an upper limit for N_f of 16 and a minimum value that follows the recommended interval of 20 ms (e.g. that means 2 for G.711 and 1 for the iLBC codec). 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.

6.3 Energy Measurement procedure

In order to measure the energy consumption of the WiFi station during the VoIP communication 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 with *awk* to calculate for each station the time spent in every energy state (i.e. TX, RX, IDLE, SLEEP) as follows:

- Time spent in TX and RX states can be immediately identified from the Wireshark capture just by checking the source, destination and physical rate of captured packets. Those states also include broadcast traffic, acknowledges and PS-POLL frames.
- Time spent in IDLE and SLEEP states can be estimated assuming that stations follow a power saving mode (PSM) traffic pattern. In this mode, a station only wakes up to listen to beacon frames in order to check for buffered packets at the AP. If so, the station sends a PS-POLL frame to the AP to retrieve the data. Once the station finishes fetching data, it goes back to sleep.

The energy consumption over intervals of 10 secs. is then calculated using (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 (see [26]).

7 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 3 repetitions of each experiment. In all experiments the voice activity detector was active and the jitter buffer was managed by pjsua dynamically according to network conditions.

7.1 Scenario 1: unsaturated channel

This scenario includes one VoIP communication between stations 1 and 3. The network emulator adds a constant network delay of 100 ms to simulate a long-distance call. No background traffic is inserted. Two codecs have been tested: ITU-T G.711 and iLBC (encoding frame length of 30 ms) from Global IP sound.

Our first test aims to verify whether our packetization control algorithm allows one to obtain energy savings at the IEEE 802.11 NIC when the target minimum quality (R_{min}) decreases. Thus, we reduce R_{min} during the conversation from 90 to 70 in steps of 10 (step duration about 240 secs). Figure 6 plots the evolution of QoS (R factor) during the call (G.711), comparing the default case (baseline) $N_f = 2$ (constant, without packetization control), with the three variants of our algorithm described in Section 6.2.3: slow (i.e. $\alpha = \beta = 1$), fast (i.e. $\alpha = \beta = 2$) and tcp-like (i.e. $\alpha = 1$, $\beta = \lceil N_f(i)/2 \rceil$).

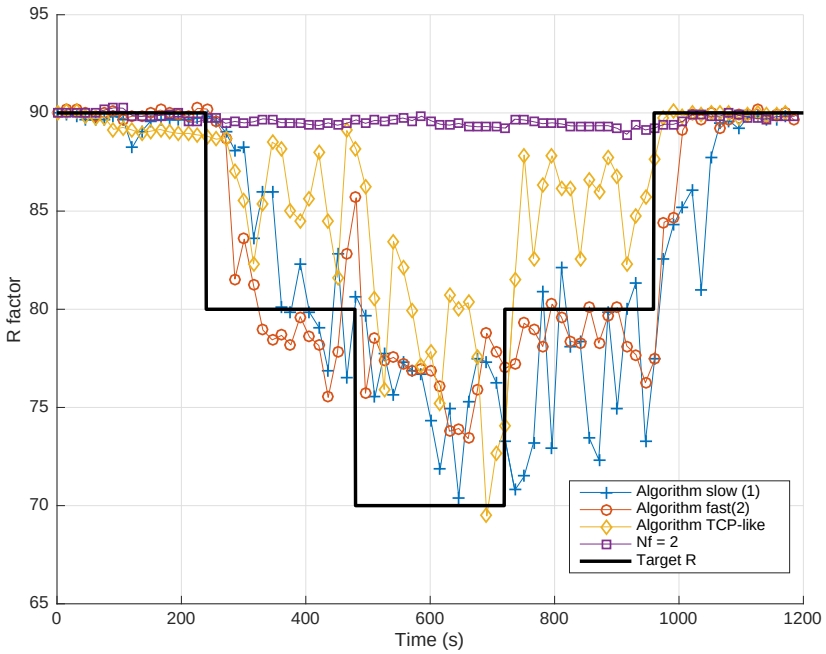

Figure 6: R factor evolution during conversation (codec G.711).

Figure 6 shows how our algorithm adapts successfully to changes in the target quality, although each version exhibit logical variations. For example, the fast version reacts faster than the rest to changes in R_{min}. The tcp-like version reacts more aggressively to reduce N_f, thus increasing the quality as a result of reduced packetization delay. Then, tcp-like is more conservative in terms of overprovisioning quality. Logically, when the lowest packetization interval is used (constant $N_f = 2$) the quality is the highest possible.

The differences found among algorithm variants can be traced back to the packetization delay exhibited by each version of the algorithm. Figure 7 shows the evolution of the overall delay during the conversation. Note that the overall delay includes both network delay (constant around 100 ms) and the application delay which in turn includes the packetization delay and the jitter buffering delay. Differences observed in Figure 7 can be mainly attributed to the packetization delay of each version of the algorithm. However, the adaptive buffering done by pjsua also has some impact on fluctuations, specially when used in conjunction with VAD.

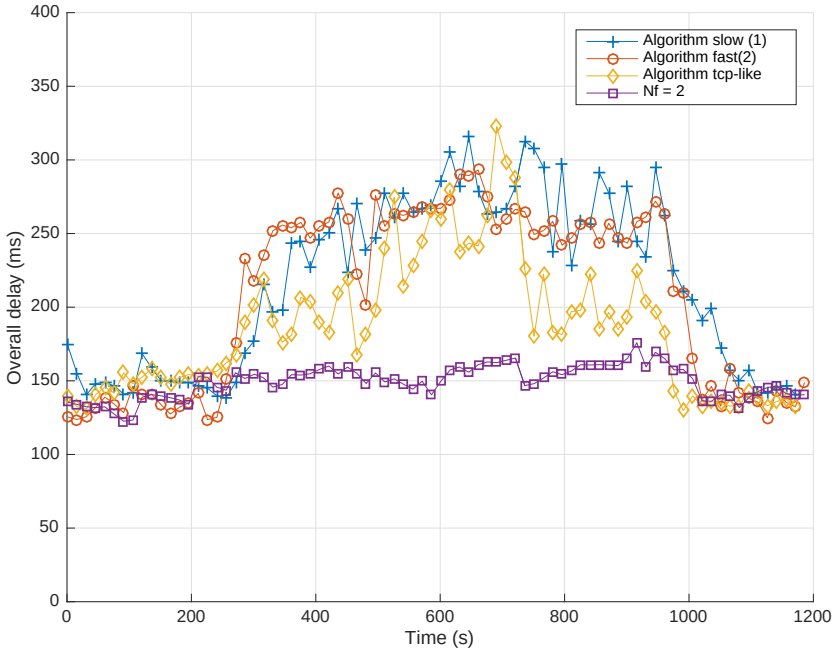

Figure 7: Overall delay (network + packetization + jitter buffer) during conversations.

Figure 8 shows the energy consumption associated with this first experiment assuming the use of power saving mode. One can observe that any version of the packetization control algorithm provides significant savings with respect to the baseline ($N_f = 2$). The lower the target quality is, the greater savings are obtained. This confirms that adaptive packetization control allows one to better take advantage of the trade-off between quality and energy expenditure. In regard to the three algorithm variations, it can be observed that tcp-like exhibits higher energy consumption than the rest which is consistent with the fact that it also achieves a higher quality score.

The evolution of N_f during the communication is plotted in Figure 9, where it can be observed that tcp-like exhibits less variability than the rest, and that the variant with $\alpha = \beta = 2$

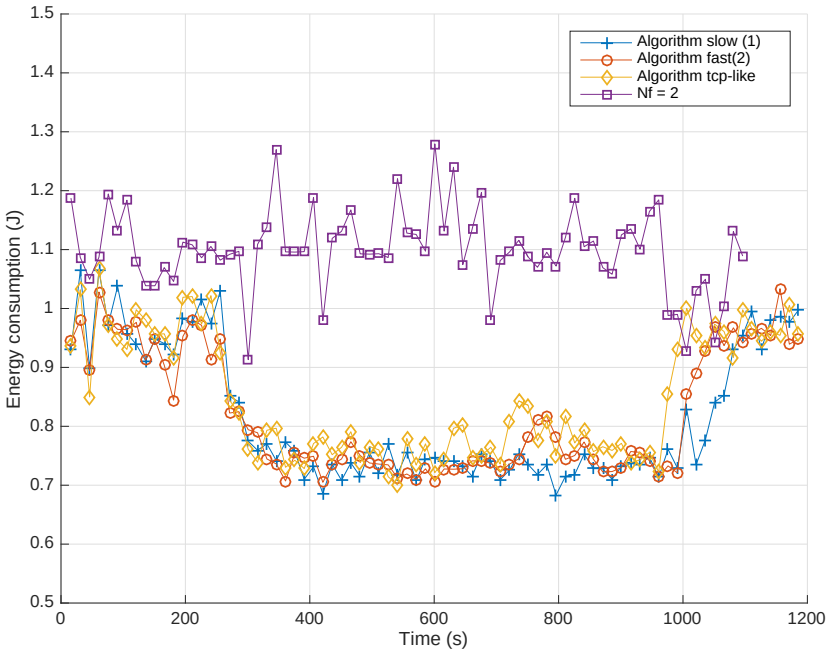

Figure 8: Energy consumption of the IEEE 802.11 NIC in non saturated scenario.

reacts faster than the rest to changes in R_{min}, resulting in greater energy savings.

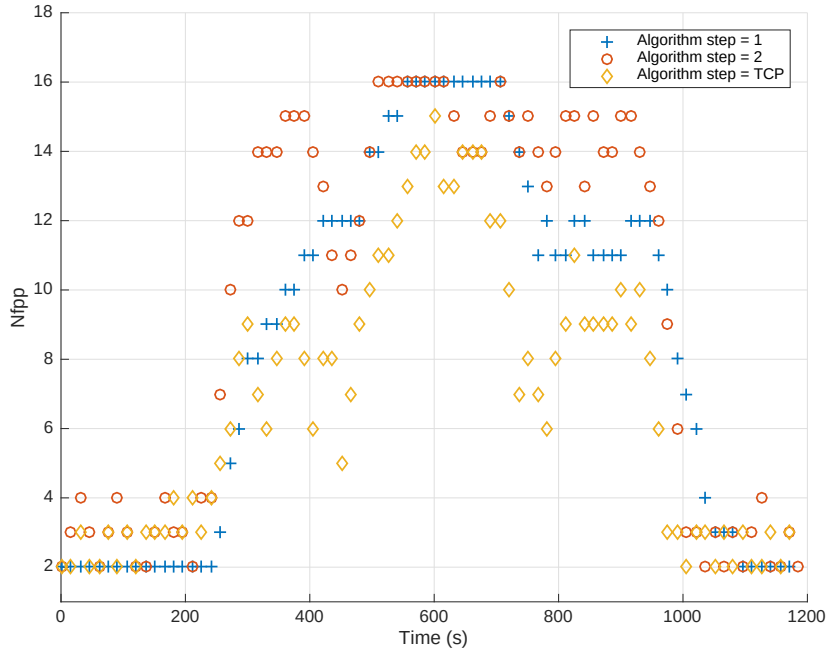

Figure 9: evolution of N_f during conversations for each variant of the algorithm.

Figure 10 introduces the energy savings obtained by our algorithm with respect to the baseline (i.e. non-adaptive) for different codecs and values of R_{min}. In the case of G.711 (Figure 10(a)), savings up to 30% can be obtained if one is willing to accept $R_{min} = 80$ in the conversation. However, a lower target quality of $R_{min} = 70$ does not compensate for the extra energy saved, which indicates that $R_{min} = 80$ would be a good target for this codec in practice. It is also noticeable that each variant of the algorithm provides a unique balance

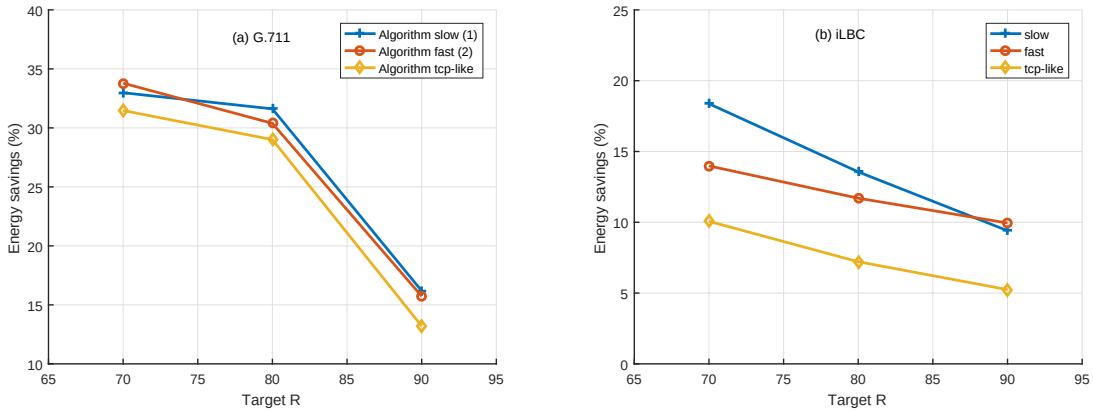

Figure 10: energy savings versus constant N_f obtained for different R_{min} setpoints for the codecs (a) G.711 and (b) iLBC.

between energy expenditure and quality. For example, tcp-like exhibits lower energy savings (but higher quality) than the rest, which can be useful when R_{min} is very low. The fast variant offers results similar to the slow in terms of energy savings but allows faster adaptation amid changing network conditions. Finally, Figure 10(b) shows the energy savings obtained with the iLBC codec (in this case compared with constant $N_f = 1$). As with G.711, the savings also increase when the target quality decreases. However, in this case, the energy savings are lower in absolute terms. It is also noticeable a greater difference between variants. This can be attributed to the fact that the codec frame speech length is 30 ms and incrementing or halving N_f is amplified by this fact.

7.2 Scenario 2: saturated channel

In this second experiment, the network emulator has not been used. However, background traffic has been generated (udp packets sized 100 bytes) using *rptools*, and the period has been set to create a specific bitrate. We have increased the bitrate of the background traffic in steps of 20kbps every 90 seconds to saturate the wireless channel. As a result of saturation, high delay and loss are expected to impact on the quality. Notice that VoIP needs a minimum acceptable quality (e.g. $R_{min} = 70$) so that communication can be viable. It is expected that our control algorithm prolongs the viability of the call under sudden adverse network conditions. As with the previous experiment, we have used the codecs G.711 and iLBC with VAD and compare the results of applying our algorithm to using a non-adaptative client (i.e. constant packetization interval).

Figure 11 shows the quality obtained with the G.711 codec for R_{min} 90. It can be observed that the non adaptative client ($N_f = 2$) can only keep up the conversation until the background traffic reaches 470 kbps, then the quality drops below 60, making a conversation unfeasible.

The variants slow and fast of the algorithm are able to keep up an acceptable quality score until the background traffic reaches 530 kbps. The tcp-like variant achieves the best performance as it prioritizes quality versus energy savings, being able to keep up the conversation until the background traffic reaches 550 kbps.

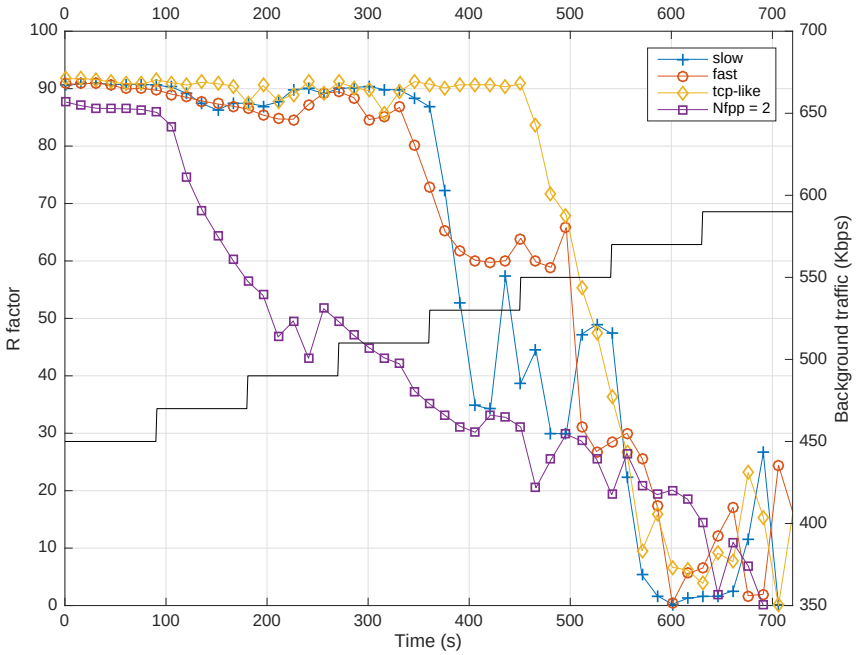

Figure 11: R variation with background traffic for the G.711 codec and $R_{min} = 90$

Observing the overall delay in the previous experiment (see Figure 12), one can conclude that once the channel is saturated, the delay increases exponentially. The algorithm tries to deal with this by changing the packetization interval, which explains that a constant $N_f = 2$ saturates faster than the rest. This suggests that higher packetization intervals (such as those provided by the algorithm whenever is possible) contribute to alleviating network congestion. Finally, although we hypothesized that the bigger packets created by longer packetization intervals are more prone to suffer errors and cause retransmissions, this was not the case in our experiment.

When the target quality is lowered to $R_{min} = 75$ we obtain the results shown in Figure 13. In this case, the conversation is extended until the background traffic reaches 570 kpbs. This can be attributed to the fact that lower quality targets provide room for increasing the packetization interval near the upper limit of 16. This happens with the three variants of the algorithm as shown in Figure 14. It is interesting to point out that the quality offered is still clearly above 75, which can be traced back to the lack of extra delay with respect to our previous experiment (i.e. the network emulator was not used).

Although the algorithm is aimed to prolong the feasibility of a conversation under channel saturation, the energy consumption also benefits from using the algorithm. In Figure 15 we represent the energy consumption for the G.711 and $R_{min} = 75$ and it can be observed a significant saving (around 30%) when any variant of the algorithm is used with respect to a

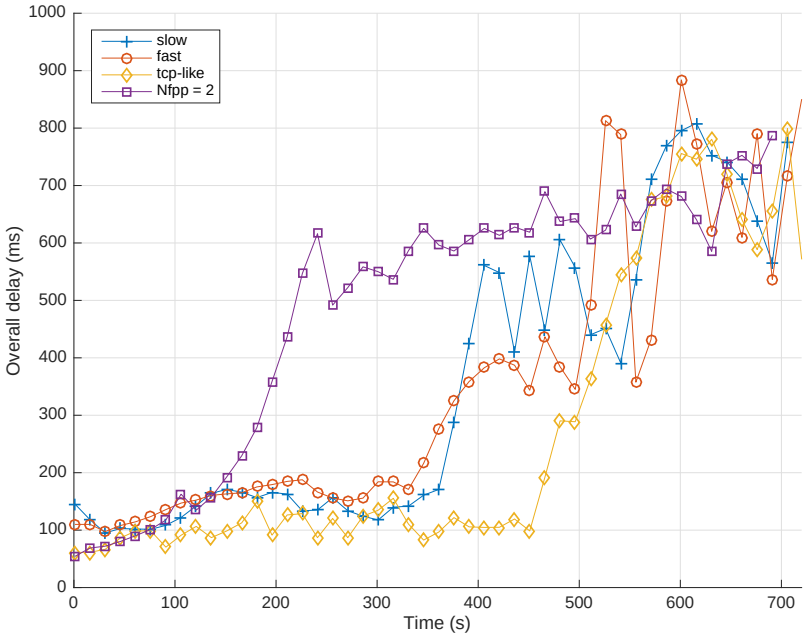

Figure 12: Overall delay with G.711

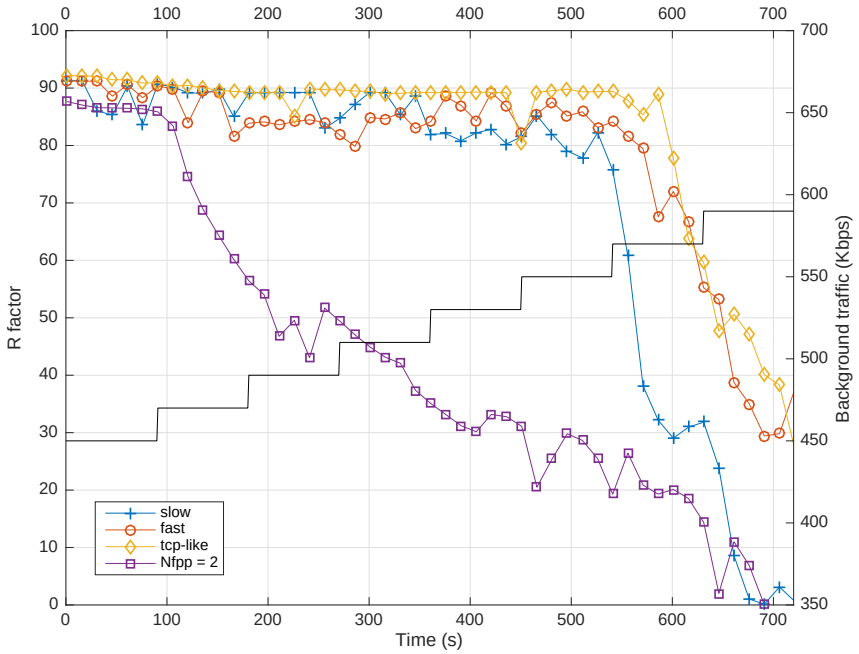

Figure 13: R variation with background traffic for the G.711 codec and $R_{min} = 75$

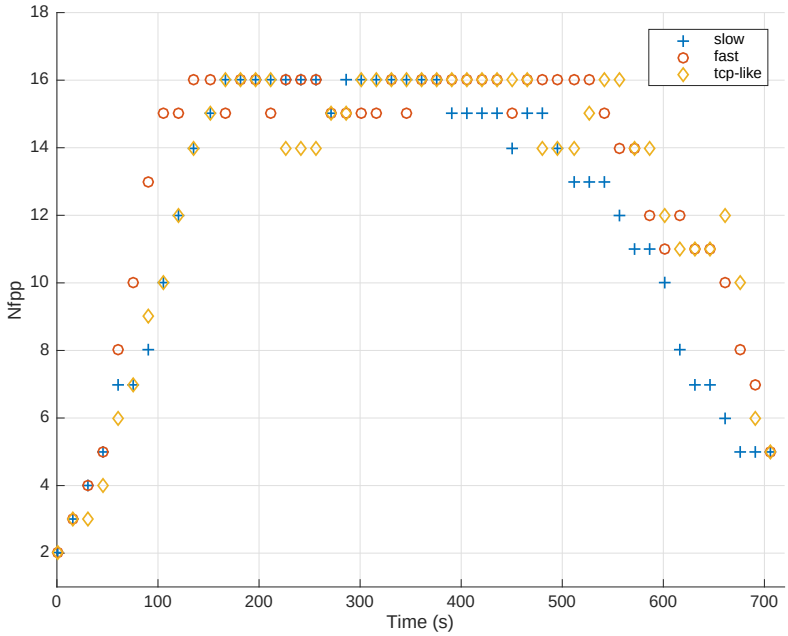

Figure 14: Evolution of the packetization interval (N_f) during the experiment (G.711 and $R_{min} = 75$)

non-adaptive client (i.e. constant packetization $N_f = 2$). The high similarity between the algorithm variants is attributable to the fact that N_f is near the upper limit most of the time independently of the variant (see Figure 14).

In the case of the codec iLBC, the results obtained do not differ much. In this case, the lower limit applied (baseline) was $N_f = 1$ as this codec exhibits a minimum packetization delay of 30ms. Another difference with the previous experiment is the need to increase the background traffic to reach the saturation point which can be traced back to its lower bandwidth (13,3kbps). Figure 16 shows the perceived quality during the conversation for a target $R_{min} = 75$. Note that $R_{min} = 90$ is not reachable with this codec.

Finally, Figure 17 shows the energy consumption for the iLBC codec in the saturated scenario. One can observe that energy savings of about 25% can be achieved with respect to a non-adaptive baseline client.

We can finally conclude that in a saturated scenario our algorithm allows one to prolong the duration of the call under adverse circumstances. In addition, energy savings around 25% are also noticeable in our experiments.

8 Conclusion

From our experiments, we have demonstrated that the application of our packetization rate control algorithm can provide significant energy savings at the IEEE802.11 NIC 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, our algorithm is able to obtain significant energy savings at the NIC (up to 30% when using PSM) while QoS levels continue being acceptable (*R* between 75 and 80).

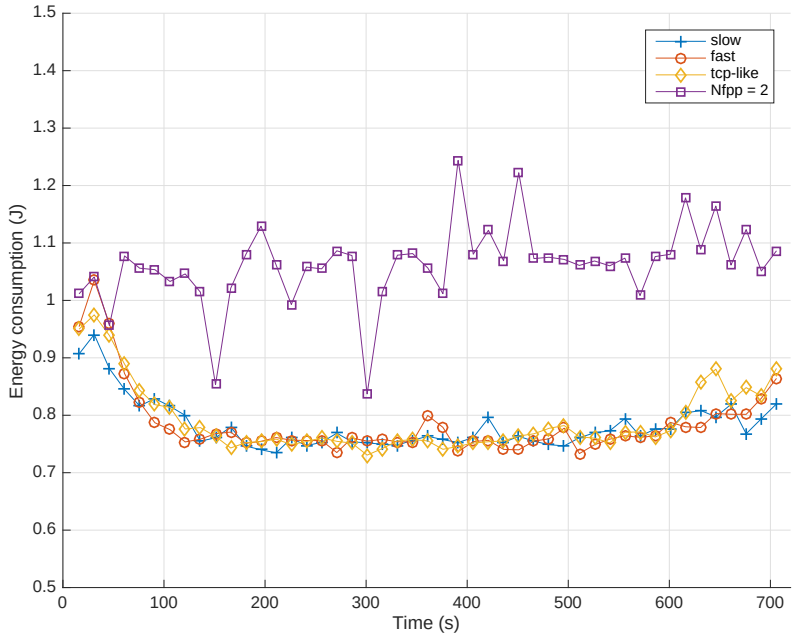

Figure 15: Energy consumption in the saturated scenario (G.711 and $R_{min} = 75$)

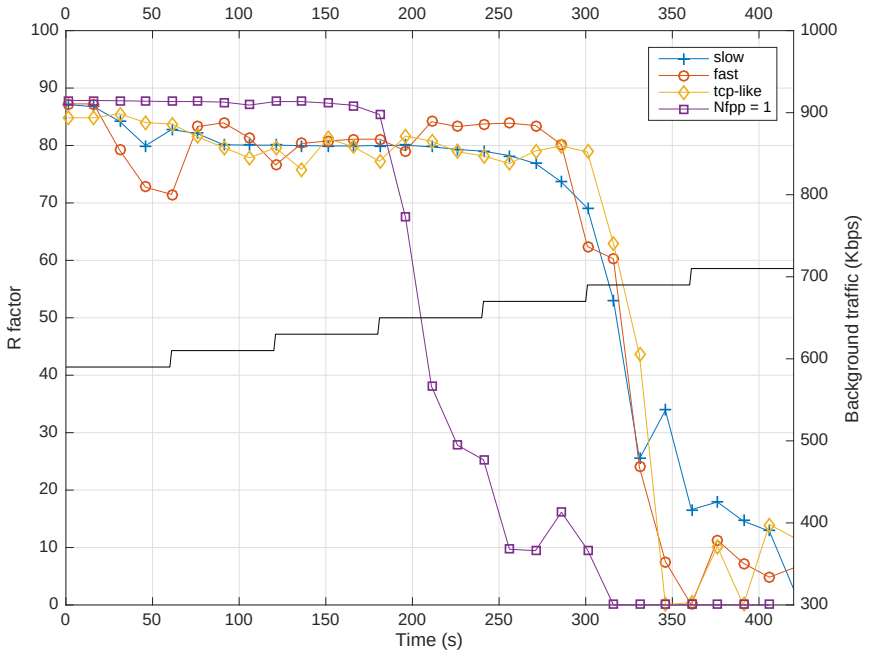

Figure 16: R factor evolution during conversation (iLBC and $R_{min} = 75$)

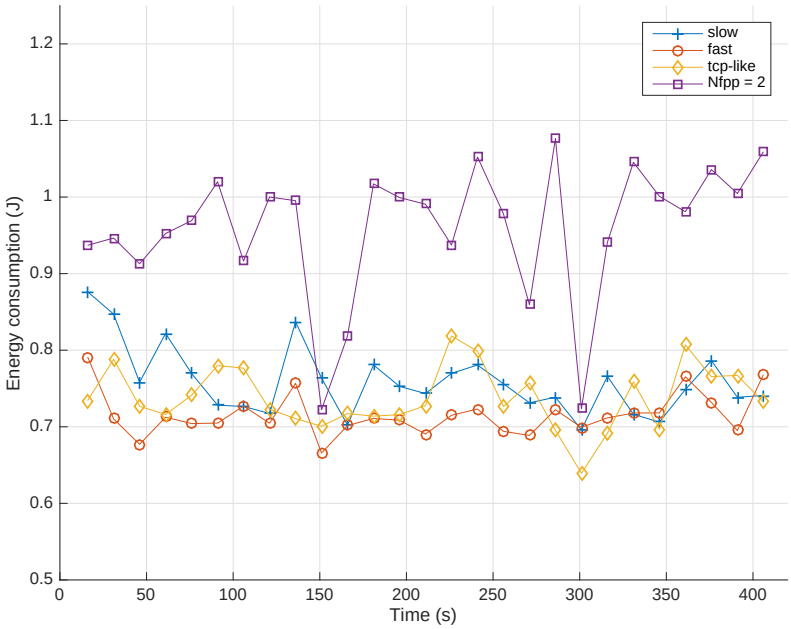

Figure 17: Energy consumption in the saturated scenario (iLBC and $R_{min} = 75$)

Under poor network conditions, our algorithm prioritizes to reach minimum quality over energy efficiency which may reduce the energy savings. Therefore by adaptively controlling the packetization process, we can tolerate saturated channel conditions for longer periods.

The balance between quality and energy savings can also be refined by using one of the three variants of the algorithm proposed in this paper.

Finally, we are presently working on including codec selection as part of the optimization strategy since different codecs will react differently to packet loss and will also exhibit different energy consumptions due to both traffic and processing power.

References

[1] 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, https://doi.org/10.1145/505666.505669

[2] 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, https://doi.org/10.1007/s11276-008-0137-y

[3] 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, https://doi.org/10.1109/LCOMM.2010.07.100468

[4] 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, https://doi.org/10.1016/j.comcom.2010.09.008

[5] 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, https://doi.org/10.1109/TMC.2009.49

[6] 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, https://doi.org/10.5296/npa.v6i2.5370

[7] 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, https://doi.org/10.1109/TVT.2010.2043274

[8] R. Estepa, A. Estepa, G. Madinabeitia, and M. Davis, "Improving the energy efficiency of voip applications in ieee 802.11 networks through control of the packetization period," in *Jornadas de Ingeniería Telemática 2017. Valencia (Spain)*, 2017, https://doi.org/10.4995/JITEL2017.2017.6492

[9] 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, https://doi.org/10.1109/CCNC.2006.1593155

[10] 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, https://doi.org/10.1109/QSHINE.2005.64

[11] 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, https://doi.org/10.1007/s11276-007-0051-8

[12] 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, https://doi.org/10.1109/TNET.2007.892881

[13] 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, https://doi.org/http://dx.doi.org/10.1016/j.conengprac.2007.04.015

[14] 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, https://doi.org/10.1109/LCN.2009.5355192

[15] 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, https://doi.org/10.1002/wcm.747

[16] 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, https://doi.org/10.1145/2480741.2480753

[17] 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, https://doi.org/10.1145/2069087.2069104

[18] V. Namboodiri and L. Gao, "Energy-efficient voip over wireless lans," *Mobile Computing, IEEE Transactions on*, vol. 9, no. 4, pp. 566–581, 2010, https://doi.org/10.1109/TMC.2009.150

[19] 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, https://doi.org/10.1016/j.comnet.2007.01.026

[20] I.-T. G.107, "The e-model, a computational model for use in transmission planning," *ITU-T Recommendation G.107*, 2009.

[21] A. Estepa, R. Estepa, and J. M. Vozmediano, "On the suitability of the e-model to voip networks," in *Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on*. IEEE, 2002, pp. 511–516, https://doi.org/10.1109/ISCC.2002.1021723

[22] 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, https://doi.org/10.15598/aeee.v9i5.542

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

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

[25] 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.

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

Copyright Disclaimer

Copyright reserved by the author(s).

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/3.0/).