Impact of VoIP codecs on the energy consumption of portable devices

Article · October 2011

DOI: 10.1145/2069087.2069104

4 authors, including:

Antonio Estepa
Universidad de Sevilla

Melany Vozmediano
Universidad de las Fuerzas Armadas-ESPE

Rafael Estepa
Universidad de Sevilla

Impact of VoIP codecs on the Energy Consumption of portable devices

Antonio Estepa, Juan M. Vozmediano, Jorge López, Rafael M. Estepa
Department of Telematics Engineering
Universidad de Sevilla (Spain)
C/ Camino de los Descubrimientos s/n
{aestepa,jvt}@trajano.us.es, jorge.lopez@descalzos.es, rafa@trajano.us.es

***Abstract*—In this paper we investigate the influence of the codec into the energy consumption of VoIP applications. These applications are increasingly extended among users of battery-dependent devices such as laptops, smartphones and tablets.**

VoIP applications usually offer a number of different speech codecs to the choice of users, who can select the codec (or application) that is better suited to their needs. Although the absolute energy savings will be device dependent, we provide a methodology to compare the energy efficiency of different VoIP codecs for a given device. This allows users and developers to minimize the energy consumption by choosing the most adequate codec, and introduces a new variable into the QoS-bandwidth balance that has traditionally lead the codec selection in VoIP applications. Our results show that the codec can have a significant impact on the energy consumption attributable to the VoIP software, specially if Power Saving Mode (PSM) is used in the Wifi card of the portable device.

***Index Terms*—Energy, speech codecs, VoIP, 802.11**

I. Introduction

Mobile devices such as smartphones, netbooks or tablets have recently experienced dramatic growth in both business and residential markets. The evolution of technology with increasing processing power, memory capacity and screen resolution at low cost is allowing the use of current mobile devices as multi-purpose machines running productivity, social and multimedia applications. However, the battery life still persists as a key concern experienced by users of portable devices.

The software under execution makes use of several hardware components such as CPU, network interface card (NIC) or the screen, that contribute to the overall energy expenditure of a battery-operated device. The particular weight of each component in the overall energy budget is not only device-dependent, but also configuration-dependent (e.g. screen brightness) and software-dependent [1].

Multimedia distributed applications such as streaming, videoconferencing or VoIP are gaining momentum in today's portable devices. Mobile users have massively adopted multi-platform VoIP programs such as Skype, Nimbuzz or Viber which allow Wifi-or-3G VoIP communication using the free Internet [5]. Unfortunately, this kind of applications exhibit high power consumption due to the intensive use of the processor and NIC compared to other type of applications such as web browsing [2], [3].

There has been some related work dealing the energy consumption of portable WiFi devices running VoIP software. Most works provide techniques for energy-savings at the NIC compoment, either by optimizing the MAC-sublayer operational values for specific network conditions [7], using alternative NICs when possible [2], [5], [8] and most notably, by using the Power Saving Mode (PSM) of WiFi cards with cross-layer approaches [9]–[12]. However, the proposed techniques prevent the use of standard VoIP software or default NIC operational behavior since there is a need to be aware of the VoIP application traffic pattern.

Codecs play a central role in VoIP applications. The codec and its associated features such as mode or silence suppression can be set dynamically by the VoIP application or manually forced by user-preferences (e.g. Skype [24]) or by SIP/H.323 negotiation. Since the selected codec has impact on the computational requirements and traffic pattern, it makes sense to inquire about the potential impact on the energy consumption of a battery-operated device. However, to the best of our knowledge, scarce efforts have been done in this field. In [6], codec switching is proposed as power-saving mechanism according to the battery level and QoS thresholds. The authors fetch the battery level of a device during the execution of a traffic generator that emulates the traffic from different VoIP codecs. However, actual VoIP codecs are not used and, consequently, the codec processing power in both encoding and decoding processes is ignored. Moreover, the authors do not specify how their results are extensible to other portable devices.

In this work we investigate the influence of popular VoIP codec on the energy consumption of a device. The specific contributions of this work are as follows:

- We investigate how codecs and its associated features, use those hardware components that are expected to be codec-sensitive such as CPU and NIC. We also define a simple software-based methodology to obtain an estimation of the energy consumed by these components on a particular device that is easily extensible to any other device.
- We present the VoIP-attributable energy consumption of popular VoIP codecs such as G.711, G.723.1, G.729, AMR and iLBC, including its optional features (e.g. multi-rate, Silence Suppression). This can be useful for for codec selection by both users and VoIP software

developers. Although our results are device-specific, we are convinced that our resultant ranking still holds for most portable devices.

As suggested in [6], [12], energy optimization might also negatively impact on the user-perceived VoIP quality, which also affects to codec selection. Consequently, we believe that the final decision on the codec to be used should be a balance between energy-efficiency and MOS. However, the scope this paper is the study of the energy consumption of codecs, leaving the inclusion of QoS for further work.

The remainder of this paper is as follows. Section II overviews the codecs under study. Section III will analyze the relationship between the energy and the VoIP application parameters. Section IV will present the methodology followed in our experiment. Section V shows the experimental results obtained. Finally, Section VI concludes the paper.

II. VoIP APPLICATIONS AND CODECS

Figure 1 represents the main components involved in the media pipelining of one side of a VoIP application. The device's sound card generates 8 KHz digital samples (PCM 16 bits) of the original sound which is a mix of both background noise and speech. These samples are processed by the codec, which is a key component of a VoIP application. In its basic mode of operation, codecs operate on a group of PCM samples to generate code-words termed *frames* at periodic intervals. The frame size and inter-frame period will depend on the particular compression algorithm applied by each codec. The frames generated by the codec are packetized to be delivered to the right codec in the destination. This process includes the use of the real-time protocol (RTP) [13], UDP and IP protocols. Overhead savings can be obtained by packing multiple frames into a single IP datagram. However, most VoIP applications apply when feasible the default inter-packet value of 20 ms as suggested by the RTP profile [14]. IP packets are sent and received by the NIC of the device which, for our study, will be a WiFi card.

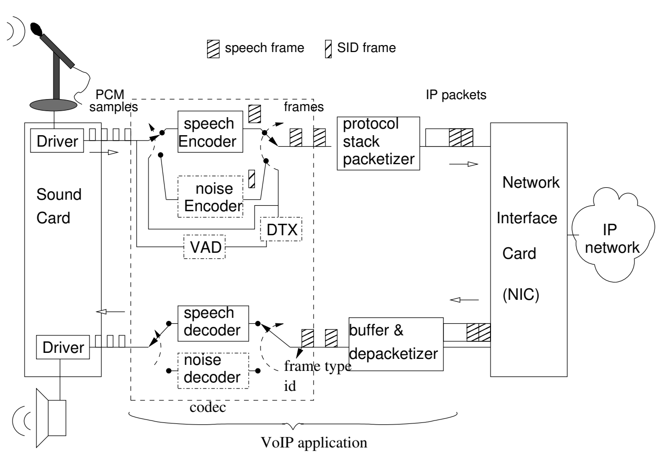

Fig. 1. Media pipelining in VoIP applications

In the reverse path, the frames carried by the IP packets are received by the NIC and placed in a buffer managed by the VoIP application to compensate the network jitter [15]. Every new inter-frame period, a new frame is read from the buffer and decoded. Some codecs define Packet Loss Concealment (PLC) mechanism [16] to mitigate the sound effect of missing frames. The output of the decoding process is, again, a set of 16-bit PCM samples that reconstruct the original signal with some distortion. Speech codecs vary in the sound quality, the bandwidth required and the computational requirements.

The codec operation periodically generates fixed-sized packets or, equivalently, Constant Bit Rate (CBR) traffic. However, in order to reduce their bandwidth requirement, some codecs generate VBR traffic by featuring the following techniques:

- *Silence Suppression*. This feature prevents the generation of frames during inactive voice periods, obtaining bandwidth savings over 50%. It relies on a Voice Activity Detector (VAD) algorithm which distinguishes whether the voice is active or not. A Discontinuous Transmission Algorithm (DTX) determines, for each inactive voice frame, the need to send a background noise update from the encoder to the receiver. This background noise update is encoded differently from speech frames in smaller-sized frames termed Silence Insertion Descriptor (SID). SID frames account for 3-7% of the total frames generated in a conversation [17] and are decoded by the Comfort Noise Generator (CNG) at the receiver codec [18] thanks to the frame-type field included in the frames. G.729b, G.723.1 and AMR are examples of codecs that feature Silence Suppression.
- *Multi-rate Codecs*. Some codecs offer multiple encoding algorithms which results in multiple bit-rate by generating frames of different sizes or/and changing the inter-frame period. Each mode of operation in these multi-rate codecs exhibits also a different sound quality, and can be dynamically selected and correspondingly indicated in the frames. G.723.1, AMR and iLBC are examples of multi-rate codecs.

Table I summarizes the main characteristics of the ITU-T, 3GPP and IETF narrowband [1] codecs that are widely adopted in today's VoIP applications. For each codec it provides its sampling and bit-rate, the size of speech and SID frames, the encoding algorithm family, its capacity for DTX and the intrinsic listening speech quality resulting from the ITU-T P.862 PESQ measurement method. Finally, we show the URL of the reference source code from the corresponding standards.

III. VoIP ENERGY BREAKDOWN

As stated in Section I, the battery-life of a mobile device running a VoIP application will be both hardware-dependent and VoIP-application dependent. A VoIP program typically uses the following hardware components:

- Processor (CPU). Used by the codec's encoding and decoding processes, including the VAD algorithm when

[1]Narrowband codecs take 8.000 samples/sec while Wideband codecs take the double for higher fidelity at the cost of higher bandwidth usage. However, the use of wideband codecs is out of the scope of this paper.

Codec	Sampl.-rate (kHz)	Bitrate (Kbit/s)	Frame Size (ms)	speech frame size (bit)	SID frame size (bit)	Encoding algo-rithm	DTX	MOS	C reference source code
G.711	8	64	-	640	-	PCM	No	4.39	http://www.itu.int/rec/T-REC-G.711/es [19]
G.723.1	8	6.3 5.3	30	192 160	32	MP-MLQ ACELP	Yes	3.69 3.49	http://www.itu.int/rec/T-REC-G.723.1/es [20]
G.729	8	8	10	80	-	CS-ACELP	No	3.75	http://www.itu.int/rec/T-REC-G.729/es [21]
G.729A	8	8	10	80	-	CS-ACELP	No	3.67	http://www.itu.int/rec/T-REC-G.729/es [21]
G.729B	8	8	10	80	10	CS-ACELP	Yes	3.51	http://www.itu.int/rec/T-REC-G.729/es [21]
G.729AB	8	8	10	80	10	CS-ACELP	Yes	3.55	http://www.itu.int/rec/T-REC-G.729/es [21]
AMR	8	12.2 10.2 7.95 7.4 6.7 5.9 5.15 4.75	20	244 204 159 148 134 118 103 95	39	MR-ACELP	Yes	3.97 3.93 3.69 3.71 3.64 3.55 3.44 3.39	http://www.3gpp.org/ftp/Specs/html-info/26073.htm [22]
iLBC	8	15.2 13.3	20 30	303 399	-	iLBC	No	3.86 3.82	http://www.ietf.org/rfc/rfc3951.txt [23]

TABLE I
MAIN CHARACTERISTICS OF VOIP CODECS UNDER STUDY.

applicable. The CPU time required to generate a new frame will depend on the codec's encoding algorithm and should be significantly shorter than the codec inter-frame period.

- Memory. Used by the codec encoding and decoding algorithms. Codecs do not require access to large amounts of memory [1]. Moreover, we have experimentally checked that there are not significant differences in amount of memory used by different codecs, which justifies that memory will not be accounted as sensitive to the codec selected for the remainder of this paper.
- NIC. The packet size and inter-packet period will have a decisive impact on the NIC energy expenditure which is expected to be significant in the overall energy budget.
- Others: components such as the screen and audio card will also be used by the VoIP application and have impact on the overall energy consumption. However, we can assume that these components (i.e. screen and audio-card), plus other non-VoIP related- will be insensitive to the particular VoIP application or codec utilized

Therefore, we can classify the hardware components as codec-sensitive (i.e. CPU and NIC) or codec-insensitive. Consequently, we can breakdown the overall energy consumption of a device as:

$$\text{Energy} = \text{Energy}_{\text{VoIP(codec)}} + \text{Energy}_{\text{nonVoIP(codec)}} \quad (1)$$

Where $\text{Energy}_{\text{VoIP(codec)}}$ stands for the energy expenditure of the CPU and NIC, which are the two main hardware components whose energy that can be minimized by choosing codec and their related features, and $\text{Energy}_{\text{nonVoIP(codec)}}$ which includes the rest of components such as screen, audio card or memory insensitive to the VoIP codec. Figure 2 shows the relationship between the VoIP codec features and the NIC and CPU hardware components of $\text{Energy}_{VoIP(codec)}$. As shown, the final weight of these two components in the overall energy expenditure of a device will be device-dependent and should be experimentally measured (e.g. [1], [4]).

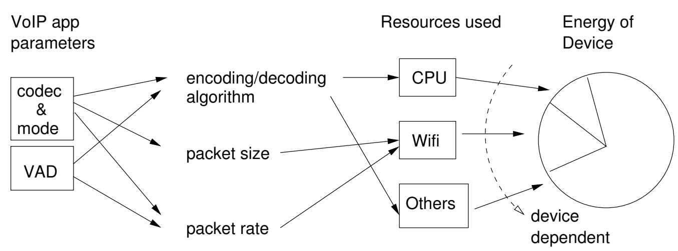

Fig. 2. Energy budget in VoIP applications

Our objective is to study the influence of the codecs and its associated features in $\text{Energy}_{\text{VoIP(codec)}}$, or, equivalently, how the NIC and CPU usage can be minimized by the selection of different VoIP codecs. Next, we develop simple models to obtain the energy consumption of both hardware components based on the usage time of each component.

A. CPU Energy Model

To measure the energy consumption of CPU attributable to the VoIP application, we propose the use of a simple linear model expressed by:

$$\text{Energy}_{CPU} = P_{\text{HW}} \times t \quad (2)$$

where P_{HW} is a power constant that will depend on the particular hardware (processor) and t is the CPU execution time of a task that consist of encoding/decoding a reference speech. t will clearly depend on the number of operations required by a particular codec in order to complete the task, and the CPU performance (i.e. Gigaflops). Different codecs can then, be easily compared by measuring t. Although more complex and comprehensive energy models for CPU can be found in literature [1], [25], [26], we believe that our simple lineal model suffices for the purposes of codec comparison.

B. Wifi Energy Model

In a similar fashion, the energy consumption of a wifi card can be broken down into the time spent on each possible state (i.e. transmission, reception, idle - and sleep for PSM) of the wifi card as:

$$\text{Energy}_{\text{wifi}} = \sum_{state} P_{\text{state}} \times t_{\text{state}} \quad (3)$$

where P_{state} is the card's power on each state and t_{state} is the time spend on each state. P_{state} is hardware-dependent and can be readily determined from the card's specifications. However, t_{state} will depend on a number of physical and MAC parameters, the number of stations sharing the radio link, and the size and frequency of the packets generated and received by the VoIP application [7], [9].

IV. Measurement Methodology

Taking physical measurements while running a target software seems to be the most accurate way to obtain the power consumption of a particular device or its hardware components [2], [3]. However, electronics testbeds are device-specific and can be tough to measure specific hardware components.

We have designed a simple software-based experiment to estimate the influence of the VoIP codecs from Table I on $\text{Energy}_{\text{CPU}}$ and $\text{Energy}_{\text{wifi}}$, that is easily extensible to any device. The idea is to measure the usage of those hardware components (i.e. the time factors in Eq. 2 and Eq. 3 respectively) that each codec and mode listed in Table I make during the encoding/decoding of a set of reference telephone conversations. The former times are later multiplied by power coefficients that are hardware dependent.

We have used the standard implementation for each codec, obtained from the URL indicated in Table I. We are aware that other proprietary optimized hardware and software implementations can be found. However, for the sake of generality, we will assume that the standard implementations are more likely to be found in VoIP soft-phones and that all codecs exhibit the same capacity for improvement. The conversations used in our experiment have been obtained from the Linguistic Data Consortium [29]. In particular we have used the CD-ROM LDC 97S42 that includes a set of 120 telephone conversations in American-English.

A. CPU energy measurement

To analyze the codecs' impact on $\text{Energy}_{\text{CPU}}$ (Eq. 2), we gauge the CPU time t that different codecs require to complete the encoding/decoding of a set of reference speeches from our conversation's bank. The codecs' source code have been modified to call the GNU C library function `getrusage` each time a new frame (of either type) is going to be processed for encoding or decoding and get the results after the completion of the frame. `getrusage` measures the CPU time and memory used by a user process.

In addition to t, Equation 2 requires the estimation of the power coefficient P_{HW}. This has been experimentally determined as in [9], by periodically fetching the battery level before and after the encoding/decoding processes with the `acpi` library. The obtained power constant has been averaged from 5 different conversations 3-min long each, corrected by subtracting the baseline power (i.e. the power when the system is idle).

The results obtained can be readily translated to other CPUs by using the ratio between their power coefficients and CPU performance [2]. The latter can be found in the processor technical specifications. A comprehensive list for Intel© CPUs can be found at http://www.intel.com/support/processors/sb/cs-017346.htm.

B. Wifi energy measurement

Measuring t_{state} (see Eq. 3) along a VoIP conversation seems to be the most accurate way to obtain $\text{Energy}_{\text{wifi}}$. However, for one station under ideal channel conditions, its is possible to take advantage of the periodic-nature of the VoIP traffic and determine analytically t_{state} for an inter-packet period of a VoIP communication between a single station and an Access Point (AP). This is done in [9], where the authors assume a bi-directional packet exchange between the station and the AP every inter-packet period. Figure 3 show this analysis for one inter-packet period when the PSM mode is used.

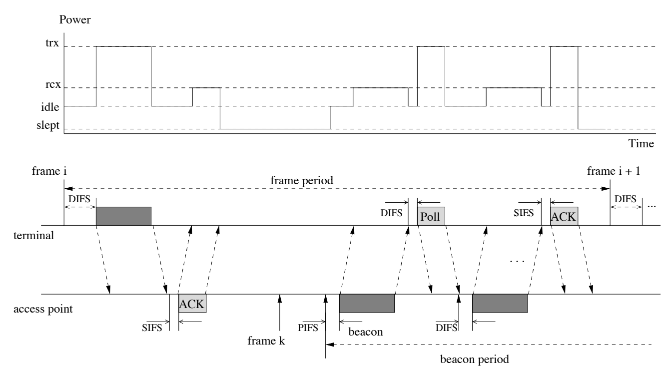

Fig. 3. WiFi operation during a VoIP inter-packet period with PSM

As in [9], we will also adopt a theoretical analysis of one inter-packet period with and without PSM[3], where the following can happen:

- *A speech packet is sent and received*. In this case, if PSM is not used, the time spent in TX state (t_{TX}) will

[2]Portable devices are likely to have single-core processors. However, in case of multiple-core processors and for the purposes of comparison, the standard performance metric (i.e. Gigaflops) could be divided by the number of cores since the encoding/decoding of a speech is done by a single thread in one processor.

[3]In PSM we assume that the period of the beacon frames is coincident with the inter-packet time, and the transmission time of a Poll frame and a Beacon frame are approximately the same. In addition, we include in our calculus the sleep-to-idle transition time [12].

include the transmission time of a speech packet size (which is codec-dependent and mode-dependent) plus the transmission time of one ACK. The time spent in RX is similar (t_{TX}=t_{RX}). Finally the time spent in IDLE state is the the rest of the inter-packet period. When PSM is used, t_{TX} will also include the transmission time of a Poll frame and its respective ACK (see Fig. 3); t_{RX} will also include the reception of a beacon frame; t_{IDLE} will only include $2 \times \mathrm{SIDFS} + \mathrm{PIFS} + 3 \times \mathrm{DIFS}$; and t_{SLEEP} will account for the rest of the inter-packet period.
- *A SID packet is sent and received*. It applies the same times previously determined with the only difference that the SID packet only includes one SID frame which size depends only on the codec.
- *No packet is sent nor received*. In this case, if PSM is used, the card only stays in RX state to receive the beacon frame and in TX state to answer with a Poll frame. Otherwise, the card stays in RX.

Conversely to [9], we do not generate emulated VoIP traffic (e.g. alternated voiced and unvoiced periods exponentially distributed) but account for the actual frames generated by each codec in the encoding process of reference telephone conversations. This will reduce the errors attributable to voice behavior models [27] and simulators [28] for those codecs performing Silence Suppression such as as G.729b, G.723.1 and AMR, that will send speech packets, SID packets or no packets according to its specific VAD/DTX algorithms.

Thus, we have modified the codecs' source code to write an auxiliary file with the sequence of frame-type identifiers generated (i.e. speech, SID or null frame). We will use these auxiliary files to obtain the number of cases of each kind for each inter-packet period of a conversation (i.e. #ACT, #SID, #NULL respectively), which in turns, yields t_{state} as follows (PSM case):

$$
\begin{aligned}
t_{TX} &= (\alpha + \frac{L_{ACT} + H}{R}) \cdot \#\mathrm{ACT} + (\alpha + \frac{L_{SID} + H}{R}) \cdot \#\mathrm{SID} \\
t_{RX} &= t_{TX} + t_{\text{Beacon}} \cdot \#\mathrm{NULL} \\
t_{IDLE} &= (3 \cdot DIFS + PIFS + 2 \cdot SIFS)(\#\mathrm{ACT} + \#\mathrm{SID}) \\
&\quad + (\#\mathrm{ACT} + \#\mathrm{SID} + \#\mathrm{NULL}) \cdot t_{\text{sleep-to-idle}} \\
t_{SLEEP} &= (\#\mathrm{ACT} + \#\mathrm{SID} + \#\mathrm{NULL}) \cdot t_{\text{inter-packet}} + \\
&\quad (t_{TX} + t_{RX} + t_{IDLE})
\end{aligned}
$$

where α is the time required to transmit the preamble, a Poll frame and a ACK frame, R is the physical bit-rate, H accounts for the headers from all layers, L_{ACT} and L_{SID} stands for the size of the speech frames carried into a single IP packet and the size of SID frames respectively, and $t_{\text{sleep-to-idle}}$ would be the transition time between the sleep and idle states. If PSM is not used, then α would only account for the transmission time of a preamble and ACK, whereas t_{IDLE} would account for all the time excluding t_{TX} and t_{RX}.

Finally, P_{state} will be read from the card's technical specifications. Therefore, our results will be readily translated to other WiFi cards by simply using the card's specifications.

V. RESULTS

The results presented in this section have been run in a DELL LATITUDE D820 laptop equipped with 2GB of RAM, Intel CoreDuo 2.33Ghz processor, and the Intel Pro 3945ABG WiFi card. The results presented have been obtained from the processing of 25 speech files of the LCD reference speech data bank. Every graph shows average values and the corresponding confidence intervals for a 95% confidence level, though in some figures these are hardly noticeable.

Note that since we measure the time-usage of hardware components, these results can be readily extended to any portable device by measuring the new value of P_{HW} and taking the power values of its NIC and CPU performance from the technical specifications.

A. CPU energy consumption

The CPU is used in both encoding and decoding processes by all codecs while also by the VAD algorithm in those codecs performing Silence Suppression[4]. Fig. 4 shows the evolution of the accumulated CPU time used by different codecs[5] in the encoding process, over a background image of the first 8 sec of one speech file. It can be noticed that those codecs performing Silence Suppression (VAD) reduce their slope during voice inactive periods since the codec only executes the VAD algorithm and the occasional encoding of SID frames. Those codecs that exhibit a sharper slope require higher computation time and consequently, demand higher energy consumption.

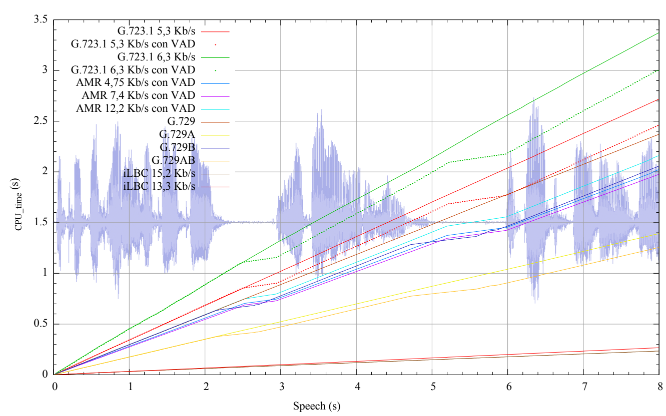

Fig. 4. Accumulated CPU time in the encoding process for different codecs.

Fig. 5 shows the CPU processing time required by each codec per second of speech, in both the encoding and decoding processes. The first noticeable result is that the decoding process requires significantly less processing power than the encoding process. The codecs demanding higher CPU time are the G.723.1 and the AMR. On the other side are the G.711, iLBC and G.729 codecs (the G.729a is a low complexity version of G.729). As expected, the use of VAD reduces

[4]We assume that no PLC mechanisms are used.

[5]For the sake of clarity, we will only show the results relative to three representative modes out of the eight possible in the AMR codec

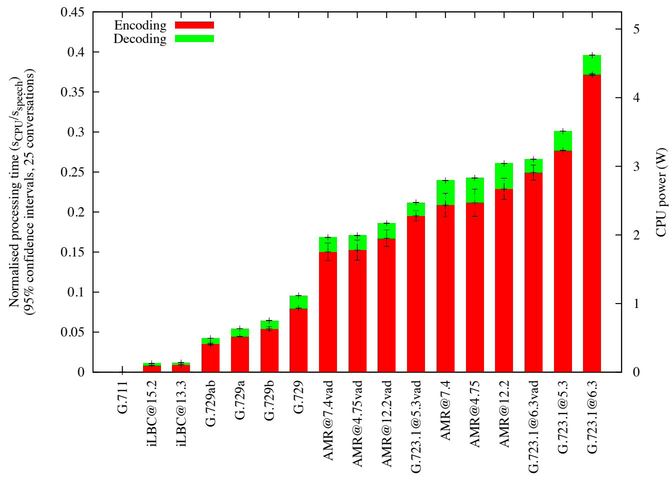

Fig. 5. Rate of CPU usage for each codec.

the average CPU power demand. The same figure shows the power on the right vertical axis. Our measurements with the `acpi` library resulted in a P_{HW} value of 11.66 J/s during the encoding/decoding process.

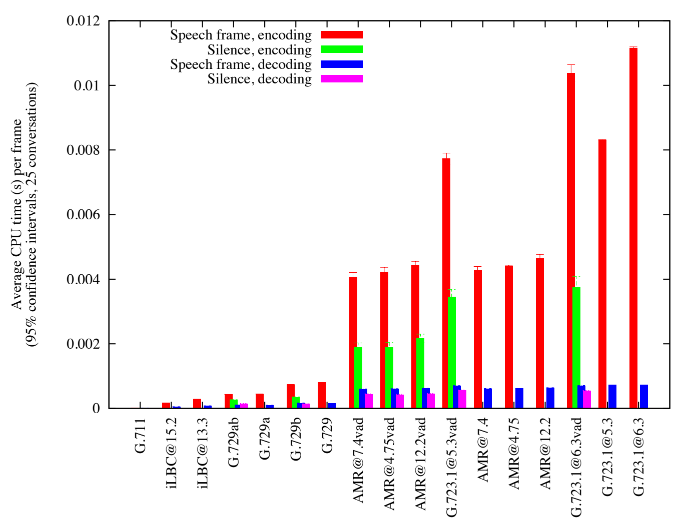

Fig. 6. Rate of CPU usage for each codec as per frame type.

Fig. 6 shows the average CPU time consumed in the encoding/decoding processes of one speech frame and SID frame for each codec under study. As it may be expected, it confirms that SID frames ocasionally generated by the DTX algorithm of some codecs, require less processing power than speech frames.

B. WiFi energy consumption

Table II shows the average time spent on each state (t_{state}) per second of our speech files. For the estimation of these times we have proceeded as indicated in Section IV-B (using the values of 802.11g), counting each packet generated in the conversation. The most significant result is that the WiFi card spends 98%-99% of the time on sleep state (when PSM is used) or idle state (when PSM is not used). This can be traced back to the small packet transmission time compared with the physical bit-rate and inter-packet times.

codec (mode)	PSM	t_{TX}(s)	t_{RX}(s)	t_{IDLE} (s)	t_{SLEEP} (s)
iLBC(13.3)	No	1.063e-03	4.491e-03	9.944e-01	-
	Yes	1.935e-03	4.491e-03	4.168e-03	9.894e-01
AMR(4.75)	No	1.309e-03	4.737e-03	9.939e-01	-
	Yes	2.616e-03	4.737e-03	6.252e-03	9.863e-01
AMR(7.4)	No	1.359e-03	4.787e-03	9.938e-01	-
	Yes	2.666e-03	4.787e-03	6.252e-03	9.862e-01
G.729	No	1.370e-03	4.798e-03	9.938e-01	-
	Yes	2.677e-03	4.798e-03	6.252e-03	9.862e-01
G.729a	No	1.370e-03	4.798e-03	9.938e-01	-
	Yes	2.677e-03	4.798e-03	6.252e-03	9.862e-01
AMR(12.2)	No	1.449e-03	4.877e-03	9.936e-01	-
	Yes	2.756e-03	4.877e-03	6.252e-03	9.861e-01
iLBC(15.2)	No	1.505e-03	4.933e-03	9.935e-01	-
	Yes	2.812e-03	4.933e-03	6.252e-03	9.860e-01
G.723.1(5.3)vad	No	6.721e-04	4.100e-03	9.952e-01	-
	Yes	1.327e-03	4.100e-03	3.370e-03	9.912e-01
G.723.1(6.3)vad	No	6.834e-04	4.111e-03	9.952e-01	-
	Yes	1.338e-03	4.111e-03	3.370e-03	9.911e-01
AMR(4.75)vad	No	8.131e-04	4.241e-03	9.949e-01	-
	Yes	1.628e-03	4.241e-03	4.435e-03	9.896e-01
AMR(7.4)vad	No	8.407e-04	4.268e-03	9.948e-01	-
	Yes	1.656e-03	4.268e-03	4.435e-03	9.896e-01
AMR(12.2)vad	No	8.907e-04	4.318e-03	9.947e-01	-
	Yes	1.706e-03	4.318e-03	4.435e-03	9.895e-01
G.711	No	9.131e-04	4.341e-03	9.947e-01	-
	Yes	1.730e-03	4.341e-03	3.908e-03	9.900e-01
G.723.1(5.3)	No	9.136e-04	4.341e-03	9.947e-01	-
	Yes	1.784e-03	4.341e-03	4.167e-03	9.897e-01
G.723.1(6.3)	No	9.337e-04	4.361e-03	9.947e-01	-
	Yes	1.804e-03	4.361e-03	4.167e-03	9.896e-01
G.729ab	No	9.912e-04	4.419e-03	9.945e-01	-
	Yes	1.965e-03	4.419e-03	5.690e-03	9.879e-01
G.729b	No	9.912e-04	4.419e-03	9.945e-01	-
	Yes	1.965e-03	4.419e-03	5.690e-03	9.879e-01

TABLE II
AVERAGE TIME SPENT ON EACH STATE PER SECOND OF CONVERSATION.

The WiFi card's specifications for P_{state} are: transmission (1.8 W), reception (1.4 W), idle (150 mW), and sleep (30 mW). Figure 7 shows the WiFi average power (i.e. energy consumed per second of speech) when PSM is used. Differences between codecs are noticeable, although small. These differences can not be attributed to the codecs' frame size but to the fact of using VAD and to the inter-packet period. When the VAD is used fewer packets are sent and received by the WiFi card, and, consequently less energy is required. However, when VAD is not used the main impact factor is the inter-packet default period which is 30 ms for the G.723.1 and iLBC 13.3 codecs while 20 ms for the rest. In any case, as stated earlier, the WiFi card spends 98%-99% of the time in sleep mode, which is clearly dominant in the results.

Figure 8 adds to Fig. 7 the results obtained when PSM mode is not used. It can be noticed that the overall WiFi power is significantly reduced with PSM, thanks to the low-power sleep state. Differences between codecs are not noticeable without PSM due to the fact that most of the time the card is on idle state and the overall WiFi power is at least three times

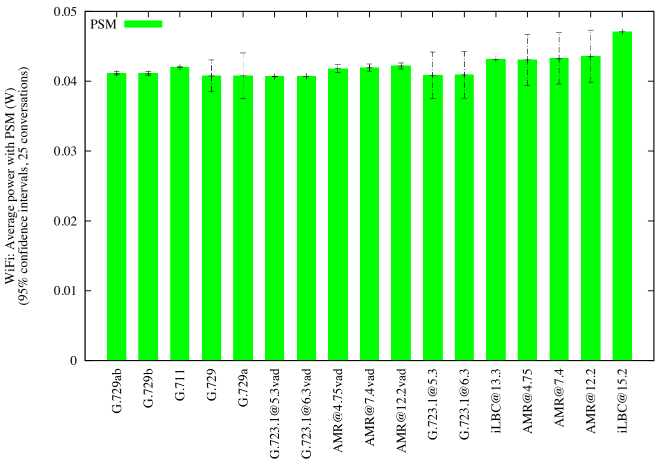

Fig. 7. WiFi power with PSM.

greater than with PSM. Then, we could state that WiFi power is codec-insensitive when PSM is not used.

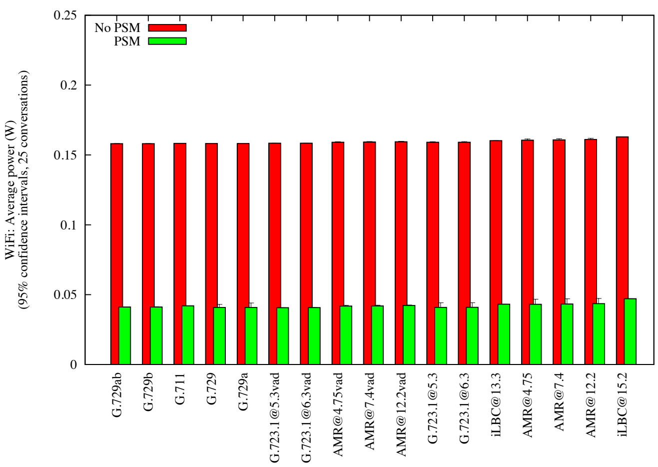

Fig. 8. WiFi power with and without PSM.

C. *Overall* $\text{Energy}_{\text{VoIP(codec)}}$ *estimation*

Figure 9 shows the addition of both CPU and WiFi power for our target device when PSM is used. Clearly, $\text{Energy}_{\text{VoIP(codec)}}$ is dramatically dependent on the CPU power used by the codecs in the speech encoding. As suggested earlier, this can be traced back to the fact that the WiFi card spends in the low-power sleep state 98%-99% of the time. Consequently, the selected codec and its associated features have a decisive impact on the overall VoIP-energy of the device, holding the raking sorted by the CPU usage shown in Figure 5.

Figure 10 shows $\text{Energy}_{\text{VoIP(codec)}}$ when PSM is not used. In this case, the WiFi power grows by a factor of three. However, it is still greatly outweighed by the CPU processing power, which, again, makes the codec processing load a decisive factor for energy saving. Note that we have assumed

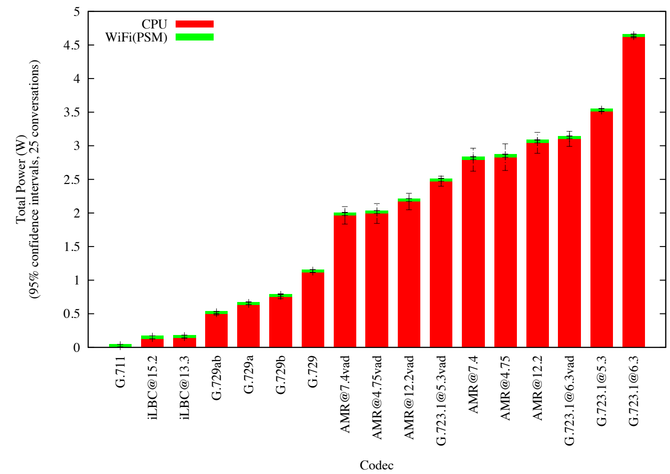

Fig. 9. $\text{Energy}_{\text{VoIP(codec)}}$ per second of speech with PSM.

an ideal scenario and channel conditions in our WiFi model. The WiFi power could grow with the number of terminals sharing the radio link in a different scenario (e.g. the time in RX state would be increased lineally with the number of users). However, even in the case of 15 simultaneous VoIP users, the WiFi card would spend the 91% of the inter-packet time in idle state and differences between codecs would be still neglectable, which still makes the CPU factor still clearly dominant and decisive to sort $\text{Energy}_{\text{VoIP(codec)}}$.

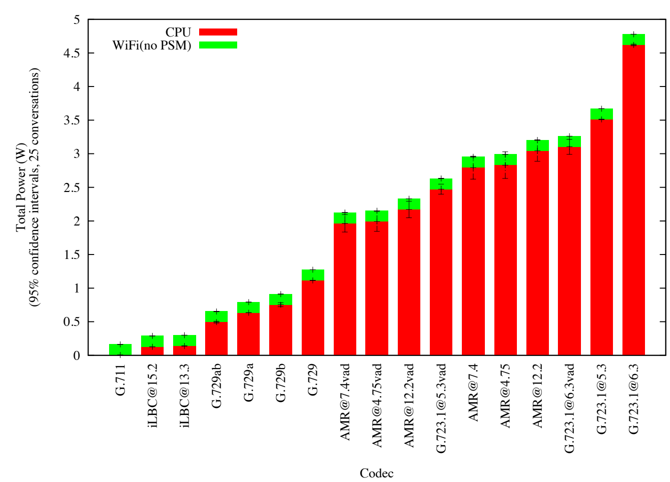

Fig. 10. $\text{Energy}_{\text{VoIP(codec)}}$ per second of speech without PSM.

Two main results can summarize this Section. First, the codec processing power is decisive in the final energy consumption attributable to the codec and its associated features. Consequently, the G.711, iLBC and G.729{ab} codecs consume less energy than the rest. Second, $\text{Energy}_{\text{wifi}}$ is almost codec-insensitive, having a significant impact on $\text{Energy}_{\text{VoIP(codec)}}$ for the aforementioned codecs when PSM is not used. When PSM is used, $\text{Energy}_{\text{wifi}}$ has almost no impact on the overall VoIP energy since the NIC is in sleep state most of the time. These results would still hold for a lighter processor that halved the processing power.

VI. Conclusions and further work

We have designed a simple methodology to gauge the impact of the codec in the VoIP-energy of a portable device. Our results show that G.711, iLBC and G.729{ab} codecs are more power-efficient than the rest under study (i.e. AMR and G.723), constituting a feasible application-level choice for energy-savings. We also found unexpectedly, that WiFi power consumption is codec-insensitive when PSM mode is not used and almost neglectable when PSM mode is used. Thus, our results are clearly dominated by the processing power that each codec needs. Nevertheless, using inter-packet times other than the RTP profile (RFC 3551) and different scenario and channel conditions could potentially bring some changes in our WiFi-related result.

We are presently working on two continuation lines. The first is the inclusion of factors that impact on the QoS such as the number of frames included in each IP packet, wide-band codecs, and the codecs' sensitivity to packet loss to obtain a energy/QoS balance that can lead to an optimized codec selection that also depended on the network conditions. Our second research line is the inclusion of comprehensive WiFi energy models that take into account other factors ignored in our ideal scenario such as number of users and background traffic.

References

[1] A. Mahesri and V. Vardhan, "Power consumption breakdown on a modern laptop," in *Power-Aware Computer Systems*, ser. Lecture Notes in Computer Science, B. Falsafi and T. Vijaykumar, Eds. Springer Berlin / Heidelberg, 2005, vol. 3471, pp. 165–180.

[2] G. R. Pering T., Agarwal Y., "Coolspots: redugin the power consumption of wireless mobile devices with multiple radio interfaces," in *4th International Conference on Mobile systems, applications and services*, june 2006, pp. 220 –232.

[3] V. Raghunathan, T. Pering, R. Want, A. Nguyen, and P. Jensen, "Experience with a low power wireless mobile computing platform," in *ISLPED '04: Proceedings of the 2004 international symposium on Low power electronics and design*. New York, NY, USA: ACM, 2004, pp. 363–368.

[4] D. W. MA Viredaz, "Power evaluation of a handheld computer," in *IEEE Micro*, 2003, pp. 66–74.

[5] Y. Agarwal, R. Chandra, A. Wolman, P. Bahl, K. Chin, and R. Gupta, "Wireless wakeups revisited: energy management for voip over wi-fi smartphones," in *Proceedings of the 5th international conference on Mobile systems, applications and services*, ser. MobiSys '07. New York, NY, USA: ACM, 2007, pp. 179–191.

[6] M. Naeem, V. Namboodiri, and R. Pendse, "Energy implication of various voip codecs in portable devices," in *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, oct. 2010, pp. 196 –199.

[7] P. Serrano, A. Garcia-Saavedra, M. Hollick, and A. Banchs, "On the energy efficiency of ieee 802.11 wlans," in *Wireless Conference (EW), 2010 European*, 2010, pp. 932 –939.

[8] M.-R. Ra, J. Paek, A. B. Sharma, R. Govindan, M. H. Krieger, and M. J. Neely, "Energy-delay tradeoffs in smartphone applications," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, ser. MobiSys '10. ACM, 2010, pp. 255–270.

[9] S.-L. Tsao and C.-H. Huang, "An energy-efficient transmission mechanism for voip over ieee 802.11 wlan," *Wirel. Commun. Mob. Comput.*, vol. 9, pp. 1629–1644, December 2009. [Online]. Available: http://portal.acm.org/citation.cfm?id=1673018.1673024

[10] Y. Chen, N. Smavatkul, and S. Emeott, "Power management for voip over ieee 802.11 wlan," in *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, vol. 3, 2004, pp. 1648 – 1653 Vol.3.

[11] V. Namboodiri and L. Gao, "Towards energy efficient voip over wireless lans," in *MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM, 2008, pp. 169–178.

[12] J. Kotwicki, "An analysis of energy-efficient voice over ip communication in wireless networks," Ph.D. dissertation, 2004.

[13] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550 (Standard), Internet Engineering Task Force, Jul. 2003, updated by RFCs 5506, 5761, 6051. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt

[14] H. Schulzrinne and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control," RFC 3551 (Standard), Internet Engineering Task Force, Jul. 2003, updated by RFC 5761. [Online]. Available: http://www.ietf.org/rfc/rfc3551.txt

[15] C.-C. Wu, K.-T. Chen, C.-Y. Huang, and C.-L. Lei, "An empirical evaluation of VoIP playout buffer dimensioning in Skype," in *Proceedings of ACM NOSSDAV 2009*, 2009.

[16] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *Network, IEEE*, vol. 12, no. 5, pp. 40 –48, 1998.

[17] R. Estepa, J. Vozmediano, and A. Estepa, "Background noise influence on voip traffic profile," in *Interactive Multimedia and Next Generation Networks*, ser. Lecture Notes in Computer Science, V. Roca and F. Rousseau, Eds. Springer Berlin / Heidelberg, 2004, vol. 3311, pp. 13–24.

[18] A. Benyassine, E. Shlomot, H.-Y. Su, D. Massaloux, C. Lamblin, and J.-P. Petit, "Itu-t recommendation g.729 annex b: a silence compression scheme for use with g.729 optimized for v.70 digital simultaneous voice and data applications," *Communications Magazine, IEEE*, vol. 35, no. 9, pp. 64 –73, Sep. 1997.

[19] "Pulse code modulation (pcm) of voice frequencies," ITU-T, Recommendation G.711, November 1988. [Online]. Available: http://www.itu.int/rec/T-REC-G.711-198811-I/en

[20] "Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s," ITU-T, Recommendation G.723.1, May 2006. [Online]. Available: http://www.itu.int/rec/T-REC-G.723.1-200605-I/en

[21] "Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (cs-acelp)," ITU-T, Recommendation G.729, January 2007. [Online]. Available: http://www.itu.int/rec/T-REC-G.729-200701-I/en

[22] S. Andersen, H. Astrom, R. Hagen, R. Hagen, W. Kleijn, and J. Linden, "Adaptative multi-rate speech codec; c-source code," ETSI, Recommendation TS 126 073, December 2008. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/26073.htm

[23] S. Andersen, A. Duric, H. Astrom, R. Hagen, W. Kleijn, and J. Linden, "Internet low bit rate codec (ilbc)," IETF, Recommendation RFC 3951, December 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3951.txt

[24] D. Bonfiglio, M. Mellia, M. Meo, and D. Rossi, "Detailed analysis of skype traffic," *Multimedia, IEEE Transactions on*, vol. 11, no. 1, pp. 117 –127, 2009.

[25] R. Jayaseelan, T. Mitra, and X. Li, "Estimating the worst-case energy consumption of embedded software," in *Real-Time and Embedded Technology and Applications Symposium, 2006. Proceedings of the 12th IEEE*, 2006, pp. 81 – 90.

[26] G. Contreras and M. Martonosi, "Power prediction for intel xscale; processors using performance monitoring unit events," in *Proceedings of the 2005 international symposium on Low power electronics and design*, 2005, pp. 221–226.

[27] W. Jiang and H. Schulzrinne, "Analysis of on-off patterns in voip and their effect on voice traffic aggregation," in *Computer Communications and Networks, 2000. Proceedings. Ninth International Conference on*, 2000, pp. 82 –87.

[28] S. Shin and H. Schulzrinne, "Measurement and analysis of the voip capacity in ieee 802.11 wlan," *IEEE Transactions on Mobile Computing*, vol. 8, pp. 1265–1279, September 2009.

[29] A. Canavan, D. Graff, and G. Zipperlen, "Callhome american english speech," in *LDC97S42*, ser. Linguistic Data Consortium, 1997.