

# An Approach to the Performance and Efficiency Power Analysis on Embedded Devices Using Asterisk

Adauto Cavalcante Menezes, Toniclay Andrade Nogueira,  
Edward David Moreno Ordonez and Admilson de Ribamar Lima Ribeiro

Department of Computing, UFS - Segipe Federal University, Aracaju, Brazil

## Article history

Received: 21-05-2018

Revised: 02-06-2018

Accepted: 01-08-2018

## Corresponding Author:

Adauto Cavalcante Menezes

Department of Computing, UFS -

Segipe Federal University,

Aracaju, Brazil

Email: [adauto.cavalcant@gmail.com](mailto:adauto.cavalcant@gmail.com)

**Abstract:** Voice over IP (VoIP) communication will dominate the computing world for years to come. In order to perform VoIP communication, it is necessary to encode and decode the voice. This process consumes the main computational resources, as an example, it is possible to mention the processor and memory. The telecommunication industries provide equipment with high purchasing prices, which makes the access to this technology still very restricted. Embedded devices are purposely constructed for certain applications, they execute systems with high criticality complexity. Asterisk is a free software for voice over IP communication and its main function is to implement the functions of a telephone exchange. These technologies promise to reduce costs and maximize results. This work describes a performance analysis on three modern embedded devices (Raspberry Pi 3, Orange Pi Plus 2 and Banana Pi M3) using the Asterisk voice over IP communication system. The performance analysis consists of evaluating the jitter, delay and bandwidth, as well as the number of concurrent calls supported in each device with SIP and IAX2 protocols with CODEC's G.711a, G.711u, Gsm, Speex, Ilbc, G.722 and in parallel, monitor the RAM memory consumption, processing and energy. The results show that the Raspberry Pi 3 and the Banana Pi M3 devices support in a satisfactory manner a high number of simultaneous calls with moderate memory, processing and energy consumption. However, the Orange Pi Plus 2 device showed high processing consumption.

**Keywords:** VoIP, Asterisk, Embedded Devices, Performance Analysis, Embedded Systems

## Introduction

The term Voice over Internet Protocol (VoIP) is conceptualized as the voice communication in networks that use the Internet Protocol (IP), which was developed with the emergence of IP Telephony, which consists of the provision of telephony services using the IP network for the establishment of calls and voice communication (Bernal, 2007). In the middle of 1990, the definition of VoIP was consolidated, when emerged the Internet Phone from VocalTec Communications, the first commercial software that enabled the communication of voice over IP, but with poor communication quality (Colcher *et al.*, 2005).

For Androulidakis (2016), private telephone exchanges serve to communication between internal telephones and communication with the public telephone

network. There are IP communication systems or also called Private Branch Exchange (PBX) IP and Time Division Multiplexing (TDM) communication systems or conventional TDM PBXs. Their software can be offered through proprietary or open source. Communication with an external medium depends on the interface for connecting analog or digital lines, usually provided by telecommunication operators; and the communication with internal telephones depends exclusively on the central office itself. Initially, the main advantage of the use of PBX communication systems was the cost of calling internal lines, as there is an internal switching of circuits, which makes the call free.

Also according to Androulidakis (2016), voice communication systems have gained popularity and now have functionalities, services that were not available from telecom operators such as hunt groups, call

forwarding and dial-by-extension. According to Sulkin (2002), the trend is to migrate to the IP telephony.

In 1999, the software for communication of the voice over IP Asterisk emerged. According to Bryant *et al.* (2013), Asterisk is a free software that performs all the functions of a conventional telephone exchange, developed by Digium. Currently, it receives several contributions from developers around the world, as it is a promising area of application that is constantly in development.

The telecommunication industries provide equipment with high values and mostly proprietary equipment. This makes it difficult to access VoIP technology.

Thus, it is necessary to provide a mechanism to reduce the expenses in the area of telephony, preferably with characteristics that are similar to that of a conventional telephone exchange; it is also necessary to expand and encourage the knowledge of students and researchers in the area of information technology and telecommunications. In this context, the importance of efficiency and cost reduction of the new generation of revolutionary personalized systems for voice over IP communication is justified. Thus, it is reinforced the idea that any quality advance in this technology can propitiate a considerable increase in the quality of the voice communication, both in the academic environment, as in the telecommunications industry, thus, we have the motivation for the aim that is to accomplish this work.

From Asterisk, it is possible to implement a voice communication system on embedded, low-cost devices that provides the necessary mechanisms of a conventional telephone exchange. For this, a performance analysis should be run on three modern embedded devices (Raspberry Pi 3, Orange Pi Plus 2 and Banana Pi M3), using the Asterisk voice over IP communication system. For this, we must perform an analysis of the behavior of the IP phone calls on the devices through research jitter, delay and bandwidth, as well as measuring the number of concurrent calls supported on each device with the SIP and IAX2 protocols with different CODECs and in parallel monitoring the RAM, processing and power consumption. At the end, we would be able see which implementation has performed best to support the Asterisk voice over IP communication system.

## Related Work

In this section, all works the classified as significant and related to the present study are discussed. A systematic analysis was carried out in order to find relevant works in bases considered important in the area of computing (IEEE Xplore, Science Direct and the Brazilian Digital Library of Computation). The amount of work that Asterisk addresses in embedded devices is greatly reduced. In this way, the work on the performance analysis with the use of the SIP and IAX2 protocol was also researched, thus, it was possible to know the measurement techniques currently used.

In a paper entitled "Performance Analysis of VoIP Services over WiFi-based systems", the authors Villac'is *et al.* (2013) presented an Alix hardware performance analysis for usage in VoIP systems under Wireless Fidelity (Wifi) networks, a server with Embedded Asterisk software is adopted. This analysis consists of tests of the Central Processing Unit (CPU) and RAM memory with a fixed number of simultaneous calls with the SIP and IAX2 protocols and with the CODECs GSM, G.711 u/a, SPEEX and G.726. Still in Villac'is *et al.* (2013), the authors could perform an energy efficiency analysis at the time of a high number of simultaneous calls, as well as conduct simultaneous calls behavior tests in an interconnection between Asterisk servers, this way making a more robust research.

Edan *et al.* (2016) present in their article a performance evaluation and Quality of Service (QoS) multimedia transmission (voice and video), using the SIP and IAX2 protocols based on an Asterisk server. The quality of the service evaluated used some parameters of the QoS, such as bandwidth, jitter and delay, in order to investigate the performance of different CODECs of voice and video. In the work of Edan *et al.* (2016), the authors were able to conduct concurrent call behavior tests with both softphones and servers. They were also able to design and develop an application for video transmission with support for the IAX2 protocol, in order to allow the completion of one of the tests listed in this work.

In the article "Implementation and Evaluation of Open Source Unified Communications for SMBs", Tesfamicael *et al.* (2014) presented the implementation and evaluation of a unified communication system composed of instant messaging and sharing (voice and video), voice messaging, VoIP communication and mobility. The evaluation covered only quantitative measurements of instantaneously supported telephone calls. However, Tesfamicael *et al.* (2014) started their work with a unified communication system approach, nevertheless, in their experiments they performed only voice and video communication tests, thus, they did not demonstrate the efficiency of the system when they were used with several modules of the proposed unified communication system. Tesfamicael *et al.* (2014) also carried out a work with great contribution, as the accomplishment of these same ones for the industry, researchers and academic community, however, there is still much to be done; experiments in embedded hardware devices, as well as the measurement of energy efficiency at times of high system consumption, are not addressed in this research.

Abid *et al.* (2012), in their article entitled "Embedded Implementation of an IP-PBX/VoIP Gateway", propose the idea of designing and implementing an embedded PBX-IP gateway, which uses low cost and open source solutions. The system integrates the FPGA hardware and incorporates the software into an external memory. Their experiment consists of an ML501 FPGA hardware, with embedded asterisk software and two pex86 computers, one connected to a serial port and the other connected to

the ethernet network. Abid *et al.* (2012) have performed only one test of Asterisk software support in the embedded FPGA hardware . The authors performed only one network connectivity test. However, they did not perform tests to gauge the number of concurrent calls supported, measurement of power consumption, processing and memory.

The studies presented above are of relevant content, as they address an investigative practice for the concept of performance evaluation of a VoIP communication system that uses the SIP and IAX protocols.

Table 1 shows the hardwares, softwares, protocols and CODECs used in the experiments of the related works, as well as in our work. Table 2 shows a comparison between the activities carried out in our work and those presented in the related works. It is observed that ours is more complete. In the following sections, the implementation details of the performance analysis and energy efficiency in embedded devices using Asterisk, as well as the prototyping and the measurements taken to extract the results will be described.

It is worth noticing the use of different versions of the Linux operating system, as well as the use of the Asterisk software in embedded devices and in a different hardware. In addition, all the works performed experiments in real environment, which makes the research more productive. Tesfamicael *et al.* (2014) and Villac'is *et al.* (2013) addressed the use of the SIPP tool, which makes it possible to make several simultaneous calls. The SIP protocol, predominant in the three studies, was used more frequently, as well as CODEC G.711.

A good research work requires parameters for validating the results. However, all the above mentioned authors did not address any metrics for validation, which leads us to believe in the occurrence of possible errors in the results demonstrated.

## Methodology and Approach

When defining a performance evaluation methodology, care must be taken in order not to make common mistakes, such as lack of objectives, tendentious proposals, incorrect methods of evaluation, among others. To avoid such errors, the ideal is to adopt a systematic approach such as the one proposed by Jain (1991), which was applied in this work. To employ this methodology, it is necessary to follow a sequence of steps.

The first step is the definition of the objectives and of the system. The second step is the preparation of the list of expected services and results. The third step is the selection of metrics, which establish the criteria for the performance comparison. The fourth step is to compile the parameter list, in fact, it is the list of parameters that affects the performance. The fifth step is about the choice of factors for studying, these factors are parameters that will suffer variations during the research. The sixth step is the selection of the evaluation technique; there are three techniques, which are simulation, analytical modeling and measurement. As a seventh step, there is the load choice , which consists of a list of service requests to the system. It is important to portray the current use.

**Table 1:** Data of Related Works

Authors	Hardware	Software	Protocol	Codec
Villac'is <i>et al.</i> (2013)	PC X86 i7 8gb RAM Switch	AsteriskNow X-lite	SIP IAX2	G.711 u G711a Gsm G.722 Speex H.263 H.264 H.261 H.263p
Edan <i>et al.</i> (2016)	Iphone PC x86 i7 3.40Ghz 8GB RAM	Zoiper Wireshark CentOS Ubuntu FreePBX	SIP	G.711 u G.711a G.729 Gsm
Tesfamicael <i>et al.</i> (2014)	Alix 2d2	Sipp Openfire Voyage Asterisk Sipp	SIP IAX2	G.711 u G.711a G.722 Gsm Speex G.726
Abid <i>et al.</i> (2012)	ML501 FPGA	Linux Asterisk	-	-
This work	Raspberry Pi 3, Banana Pi M3, Orange Pi Plus 2, Mikrotik 951g.	Raspbian Armbian Wireshark Zabbix Zoiper X-lite	SIP IAX2	G.711a G.711u G.722 Gsm Ilbc Speex

**Table 2:** Activities carried out

Qualities	Villac'is <i>et al.</i> (2013)	Edan <i>et al.</i> (2016)	Tesfamicael <i>et al.</i> (2014)	Abid <i>et al.</i> (2012)	This work
Embedded device	x	-	-	x	x
Simultaneous calls sip	x	-	x	-	x
Simultaneous calls iax2	x	-	-	-	x
Memory Consumption	x	-	-	-	x
CPU consumption	x	-	x	-	x
Energy consumption	-	-	-	-	x
Validation metric	-	-	-	-	x

Then the eighth step is the planning of the experiments. As a ninth step, we have to analyze and interpret the data, in this step we must use adequate statistical techniques in order to consolidate the results obtained, in order to allow conclusions about the performance of the system. Already tenth and final step is the presentation of the results, in this step we must pay attention to the final presentation of the evaluation.

### *Application of the Methodology*

When applying this methodology, it is possible to notice its importance, given the organized form in which the work was conducted. Initially, it is necessary to define the objectives, then the scope of the system, the services offered, as well as the evaluation technique, which are shown below.

#### Goals:

- Carrying out a performance analysis on three low-cost embedded devices.
- Determining relevant factors in the performance of these equipments.

#### System:

- The system corresponds to a voice communication software over IP called Asterisk, it interacts with the medium through the reception and realization of telephone calls through the IP Protocol.

#### Service:

- Voice over IP communication.

#### Assessment Technique:

- Measurement, therefore, it is a useful technique for analyzing the performance of computer systems.

The activity was divided into five stages:

- Step 1 - Design and test scenario
- Step 2 - Specification of metrics
- Step 3 - Definition of parameters, factors and load
- Step 4 - Planning and conducting the experiments
- Step 5 - Statistical analysis of the results obtained

The next subsection describes the first three steps, while the fourth and fifth are discussed in the Experiments and Results section.

### *Design and Test Scenario*

To carry out the prototyping, it is necessary to assume the existence of a computational model identical to the

real production environment. In the literature, good descriptions of the performance analysis in voice over IP communication systems have been found, for example, the work of Villac'is *et al.* (2013) and Edan *et al.* (2016). The first work presents a performance analysis of the Alix 2D2 hardware for use in VoIP systems. The second one presents a performance assessment and Quality of Service (QoS) for multimedia transmission (voice and video).

Based on these works and technical specifications of commercially available equipment, the present work was carried out with three modern embedded devices and with the proposal of complementing the research already done, in this way, it effectively contributes to the telecommunications industry, for small and large companies, as well as for the academic area, therefore, extending the conduction of new researches.

This work analyses the possibility of verifying the use of modern embedded devices as servers of a voice over IP communication system called Asterisk using the SIP and IAX2 protocols and with the following CODECs : G.711a (Alaw), G.711u (Ulaw), Gsm, Speex, Ilbc and G.722, no other CODECs were used because of the limitations found in the work performed, such as free softphones used only to support these CODECs. The embedded devices discussed in this work have an average price of 50 dollars, thus, it is possible to consider as a low cost solution.

The performance analysis consists in verifying and evaluating the jitter, delay and bandwidth, as well as the number of concurrent calls supported in each embedded device with the protocols and CODECs mentioned above , and in parallel to carry out the monitoring of the of RAM, processing and energy consumption. For this, the tool used for traffic analysis was the Wireshark, a software specialized in traffic analysis in IP networks, widely used by researchers in the academic environment, for example, by Edan *et al.*, 2016.

In order to perform this work, it was necessary to elaborate a test scenario, so the scenario included three modern and low cost embedded devices (Raspberry Pi 3, Banana Pi M3 and Orange Pi Plus 2), a Mikrotik RouterBoard 951g that made the switch , a Core i7 8GB RAM 500GB HD laptop. Figure 1 illustrates the architecture of the scenario designed to perform the experiments.

The Mikrotik RouterBoard that performs the switch is responsible for interconnecting all the equipment in the network. Each embedded device supports the Asterisk voice over IP communication system. The laptop has been allocated to perform the data collection with the Wireshark software.

The devices were submitted to moments of increased system consumption. For this, a virtual machine was installed on the laptop, which made it possible to make several calls in order to reach the maximum level of calls supported on each device.

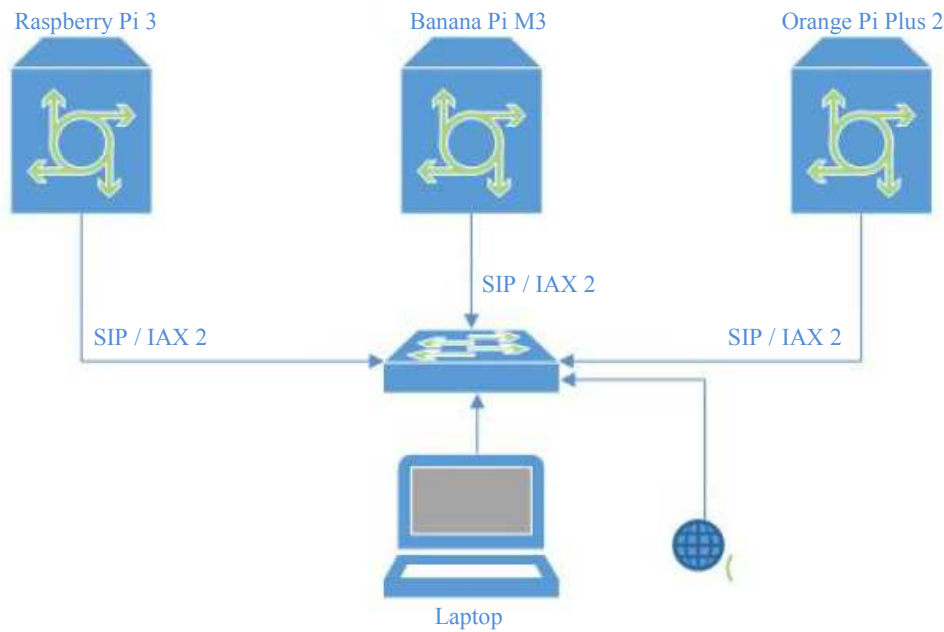


Fig. 1: Test scenario

Table 3: Embedded Devices

Manufacturer	Raspberry Pi	Banana Pi	Orange Pi	Odroid
Model	3 Model B	M3	Plus 2	XU4
CPU Cores	4	8	4	4+4
CPU Freq.	1.2 GHz	2 GHz	1.5 GHz	2.1 GHz
Memory	1 GB DDR2	2 GB DDR3	2 GB DDR3	2 GB DDR3
Storage	MicroSD	MicroSD USB Sata 2.0	MicroSD USB Sata 2.0	MicroSD eMMC
Linux	Yes	Yes	Yes	Yes
Price	\$40	\$59,99	\$59,99	\$59

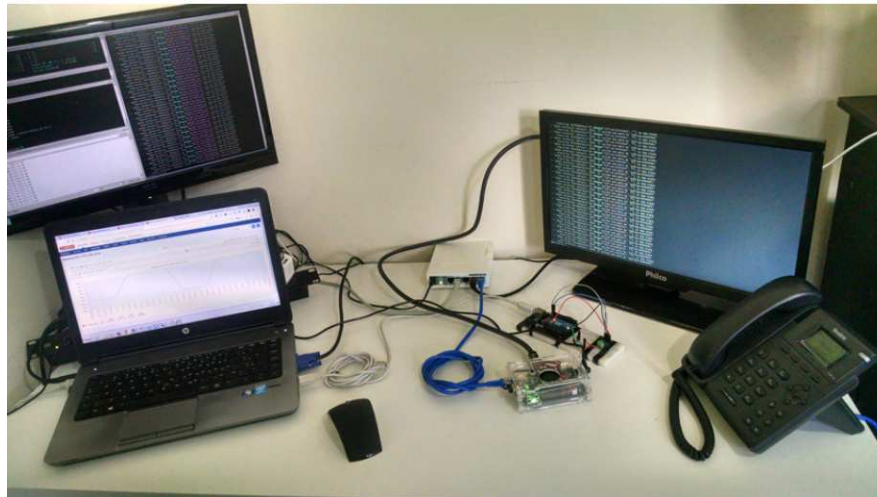
Currently, there is a great offer of embedded devices in the market, also called Single-Board Computers (SBC), with the most diverse configurations, for example, it is possible to mention the Raspberry Pi Zero, Raspberry 2 Model B, Raspberry Pi 3 Model B, among others. Table 3 illustrates four models of the last generation embedded devices, their configurations and price.

According to Digium (2017), it is not easy to size a minimum hardware for the installation of the voice over IP communication system Asterisk, however, for this difficult question, there is no precise answer. It is recommended to always use a hardware slightly beyond the needs, it is also suggested that if the system needs to support more than 20 simultaneous calls, a dedicated server should be used. Thus, for the accomplishment of this work, three devices of low cost and with good configurations were chosen, which are: Raspberry Pi 3, Banana Pi M3 and Orange Pi Plus 2.

## Experiments and Results

This section presents the implementation of the experiment, which consists of: assembling the test scenario with the embedded devices individually; accomplishment of the software approach in the devices for the elaboration of the experiment; proceeding with the process of collecting the jitter, delay and bandwidth with the Wireshark software; implementing a code in a Shell Script in order to generate loads of SIP and IAX calls with CODEC's G.711u (Ulaw), G.711a (Alaw); Gsm, Speex, Ilbc and G.722; running high call flows and in parallel to carry out the monitoring of the consumption of processing; RAM and energy; and finally, to gauge the results.

Figure 2 illustrates the current testing scenario with the Banana Pi M3 device, INA219 power meter circuit, two monitors to debug the occurrence of possible errors in Asterisk, the MikroTik switch, Yalink t19p IP phone, as well as the laptop to generate loads and gauge the processing and memory consumption.



**Fig. 2:** Experiment running

The tests were performed in order to reach the maximum number of calls supported with no occurrence of errors. Thus, several attempts were made until the expected result was found. Thereafter, 10 replicates were performed to validate the occurrence of no errors. For it is a static result, there is no variation of the mean and thus a greater number of repetitions can be done with it.

The collection was performed with the SIP protocol and IAX2 and with the CODECs supported by the free softphones addressed (Zoiper and X-lite). However, 33 collections (6 SIP and 5 IAX2 in each embedded device) were performed, each of the calls lasting approximately 2 minutes, so it was possible to obtain a good quantitative of SIP and IAX2 packets to carry out the analysis, approximately 5,000 datagrams IP.

The Wireshark tool has filters that automatically perform jitter, bandwidth and delay analysis. Though, the mean was used as a parameter for the measurements. Notwithstanding, the mean bandwidth and delay were calculated manually by exporting the captured data to LibreOffice Calc, since Wireshark does not report the average, only the maximum value reached.

Thus, the calls were made between the softphones and a self-service set up in the Asterisk of the embedded device.

The tests were performed with the SIP and IAX2 protocols, as well as with the CODEC's mentioned above, in order to achieve the objective of this work with the measurements.

In order to carry out the generation of simultaneous SIP call load, Asterisk was used in a virtual machine installed on the laptop and a dialer was implemented in Shell Script. For calls with the IAX2 protocol, it was held the exchange of the channel on the dialer, from SIP to IAX2. The dialer code is shown in Fig. 3, as well as the comments of each command executed. Figure 4 shows the flow chart of the code, which makes it is possible to better understand its operation.

Thus, two files were generated to perform a high number of simultaneous calls, it is called CallFilePlay.sh and CallFileCodec.sh. The first is used for normal calls, the second for calls with transcoding. The only difference between the two files is that in CallFileCodec.sh the extension number on the dialer is different. In this way, these files were inserted in the Asterisk/etc/Asterisk directory on the virtual machine dialer. Then, it was possible to start the tests.

The callfile works as a script to generate functionality in Asterisk, a user or application writes a calling file to the/var/spool/asterisk/outgoing/directory, and Asterisk processes immediately.

It was also necessary to configure an audio file called test.gsm, with 3 minutes, and to include it in Asterisk's "sounds/var/lib/asterisk/sounds" directory, whose purpose is for Asterisk to receive calls from the load generator, forward them to self-service and then play the audio. In this way, the media flow limited time occurs, enough to run the concurrent call test.

In order to start the tests, the virtual machine was used on the laptop to run the dialer load generator, and in parallel the monitoring of the RAM memory and processing consumption was carried out with the Zabbix software; the monitoring of the energy consumption was carried out with the INA219 circuit, as well as the verification of errors occurring in the processing of calls, through the debug in Asterisk.

A Yalink T19p E2 IP telephone was also used in order to validate the quality of the call at a high load consumption. However, it was not possible to validate the quality with the IAX2 protocol and neither with the Speex and Gsm CODEC's with the SIP protocol, due to the license limitations of the telephone set.

As the tests were carried out, the CODECs were modified, as well as the protocol. The experiments of the simultaneous calls were divided into 4 stages, which will be described in next subsections, as well as their results.

```

1  #!/bin/bash
2  aux1=$1: "Variable defined the number of calls"
3  aux2=$2: "Variable defined the number of repetitions"
4  CONTADOR=0 'Counter for number of calls'
5  CONTADOR2=0 'Counter for number of repetitions'
6  echo "Enviando $1 chamadas $2 vezes" 'Print the number of calls and replays'
7  while [ $CONTADOR2 -lt $aux2 ]; do 'Command to execute the number of repetitions'
8  echo "$CONTADOR2 x"
9  while [ $CONTADOR -lt $aux1 ]; do 'Command to execute the number of calls'
10 echo "Channel: IAX2/vm/5000" >>
11 /etc/asterisk/callfiles/teste$CONTADOR_$CONTADOR2.call 'Asterisk channel to send the call'
12 echo "CallerID: vm" >>
13 /etc/asterisk/callfiles/teste$CONTADOR_$CONTADOR2.call 'Set Callerid'
14 echo "MaxRetries: 0" >>
15 /etc/asterisk/callfiles/teste$CONTADOR_$CONTADOR2.call 'Maximum number of dial attempts (in case of failure)'
16 echo "RetryTime: 60" >>
17 /etc/asterisk/callfiles/teste$CONTADOR_$CONTADOR2.call 'Number of seconds to wait until the next dialing attempt (in case of failure)'
18 echo "WaitTime: 30" >>
19 /etc/asterisk/callfiles/teste$CONTADOR_$CONTADOR2.call 'Number of seconds the system waits for the call to be answered'
20 echo "Context: default" >>
21 /etc/asterisk/callfiles/teste$CONTADOR_$CONTADOR2.call 'When the call is answered, the call flow enters this Asterisk context
22 (/etc/asterisk/extensions.conf)'
23 echo "Extension: 3000" >>
24 /etc/asterisk/callfiles/teste$CONTADOR_$CONTADOR2.call 'When the call is answered, the call flow enters the Asterisk context above and
25 in that extension (/etc/asterisk/extensions.conf)'
26 'The above commands create the test_file_repeat_number_call_number.call'
27 /var/spool/asterisk/outgoing/
28 'The command below moves to Asterisk to process'
29 mv /etc/asterisk/callfiles/teste$CONTADOR_$CONTADOR2.call
30 /var/spool/asterisk/outgoing/
31 let CONTADOR=CONTADOR+1; 'Adding the number of calls counter'
32 done
33 sleep 10s; 'Wait time to execute the next replay'
34 CONTADOR=0
35 let CONTADOR2=CONTADOR2+1; 'Adding the number of repetitions counter'
36 done 'Finish shell script'
    
```

Fig. 3: Dialer code

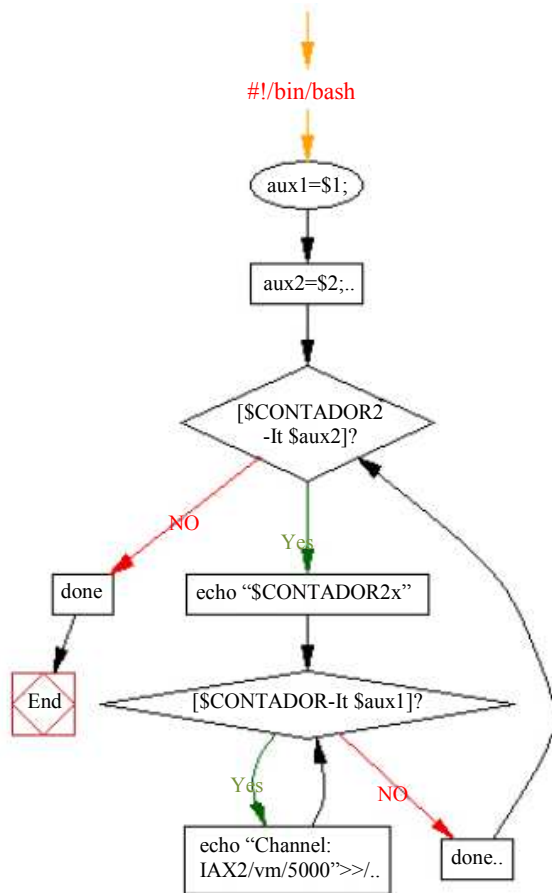


Fig. 4: Code Flowchart

The following subsections demonstrate the results of the jitter, delay and bandwidth collected with the SIP and IAX2 protocols on the embedded devices Raspberry Pi 3, Banana Pi M3 and Orange Pi Plus 2, with mean, standard deviation and confidence interval of 95%, as well as it holds brief discussions.

*Bandwidth*

Tables 4 and 5 illustrate the bandwidth of the collected packets. It is possible to observe that the CODECs G.711a, G.711u, and G.722 have higher bandwidth rates and identical values, since the CODECs Gsm, Speex, and Ilbc stand out because they have a lower bandwidth, both with SIP protocol and with IAX2. The results found on the three embedded devices are similar. However, the distinction is clear in the approach of the SIP and IAX2 protocols, since it is observed that the IAX2 protocol consumes a smaller band in front of the SIP, as shown in Fig. 5. It was not possible to collect the CODEC G.722 with the protocol IAX2 for not finding a free softphone with CODEC support. Moreover, it is possible to state that the standard deviations and the confidence interval are minimal due to the high number of samples to carry out the approach.

In order to facilitate the identification of the devices in Fig. 5, a code predecessor to the CODECs name has been inserted, so that, (R.) for the Raspberry Pi device 3, (B.) for the Banana Pi device M3 and (O.) , respectively, for the Orange Pi Plus 2 device.

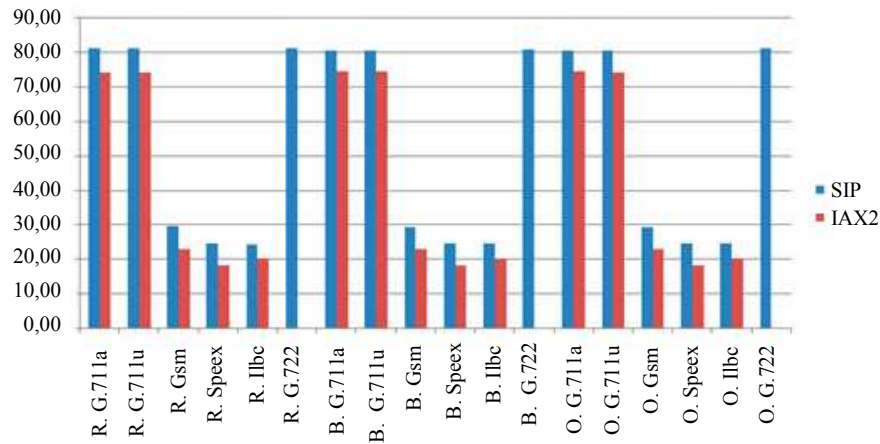


Fig. 5: Bandwidth SIP e IAX2. Unit kilobits per second

Table 4: Bandwidth SIP. Unit kilobits per second (Kbps)

Raspberry Pi 3				
Codecs	Average	Standard Deviation	Inferior Limit	Upper Limit
G.711a	81.13	0.76	81.08	81.18
G.711u	81.11	0.77	81.06	81.15
Gsm	29.57	0.28	29.56	29.58
Speex	24.52	0.20	24.51	24.52
Ilbc	24.36	0.38	24.35	24.38
G.722	81.09	0.75	81.06	81.11
Banana Pi M3				
G.711a	80.50	0.73	80.48	80.53
G.711u	80.48	0.72	80.46	80.50
Gsm	29.38	0.27	29.37	29.39
Speex	24.58	0.24	24.58	24.59
Ilbc	24.49	0.00	24.49	24.49
G.722	80.94	0.79	80.92	80.97
Orange Pi Plus 2				
G.711a	80.51	0.74	80.49	80.53
G.711u	80.52	0.74	80.50	80.54
Gsm	29.40	0.27	29.39	29.41
Speex	24.58	0.23	24.57	24.58
Ilbc	24.49	0.00	24.49	24.49
G.722	81.06	0.76	81.04	81.08

Table 5: Bandwidth IAX2. Unit kilobits per second

Raspberry Pi 3				
Codecs	Average	Standard deviation	Inferior limit	Upper limit
G.711a	74.21	0.72	74.19	74.23
G.711u	74.20	0.72	74.18	74.22
Gsm	22.99	0.22	22.98	23.00
Speex	18.14	0.17	18.13	18.14
Ilbc	20.13	0.02	20.13	20.13
Banana Pi M3				
G.711a	74.37	0.78	74.35	74.40
G.711u	74.33	0.78	74.31	74.36
Gsm	23.03	0.23	23.03	23.04
Speex	18.18	0.18	18.17	18.18
Ilbc	20.03	0.32	20.02	20.04
Orange Pi Plus 2				
G.711a	74.57	0.70	74.55	74.60
G.711u	74.25	0.73	74.23	74.28
Gsm	23.01	0.23	23.00	23.02
Speex	18.16	0.18	18.15	18.16
Ilbc	20.13	0.19	20.12	20.13



**Delay**

Tables 6 and 7 demonstrate the delay of the collected packets. It is possible to observe that the Delay with the SIP protocol is in accordance with the standard Rfc1890 (2017), which establishes that the default delay of the RTP packet should be 20ms. However, there is an exception to the Ilbc CODEC, which stands out with a value that is 50% higher than the others, as shown in Fig. 6. This is due to its coding algorithm, which performs a high compression of the audio and consequently increases the transmission delay.

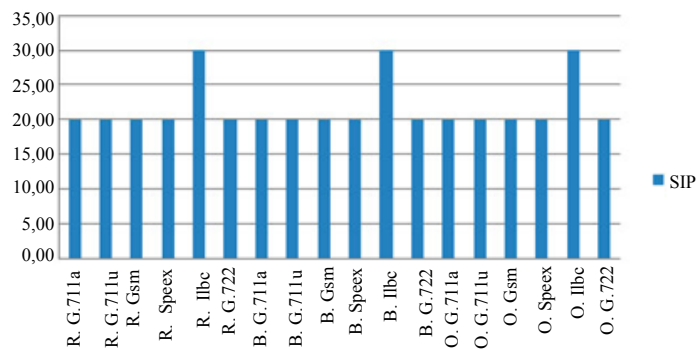
The delay with the IAX2 protocol tends to zero with all the CODECs addressed in this research, which validates its proposal as a new protocol standard. According to Rfc5456 (2016) it is an open protocol that carries the transport of signaling and the media. In addition, the IAX2 protocol also proposes to eliminate any transmission delays. It is also observed that the standard deviations and the confidence interval are minimal due to the high number of samples to carry out the approach.

**Jitter**

Tables 8 and 9 illustrate the jitter of the collected packets. It is possible to observe that, with both the SIP protocol and the IAX2 protocol, the jitter tends to zero, this is because both protocols have as strategy to keep the frames in a buffer, in order to allow the slower frames to arrive in time to be played in the correct sequence. The higher the amount of jitter, the greater the number of frames in the buffer in order to minimize the jitter in VoIP calls.

**SIP Calls**

Table 10 illustrates the data collected using the SIP protocol. It is possible to observe that the GSM CODEC supported the largest number of calls in the Raspberry Pi 3 and Banana Pi M3 devices. However, there is a high consumption of RAM. Nevertheless, given the RAM capacity of the Raspberry Pi 3 device of 1GB RAM and Banana Pi M3 of 2GB, this result becomes insignificant.



**Fig. 6:** Delay SIP. Unit milliseconds (ms)

**Table 6:** Delay SIP. Unit milliseconds (ms)

Codecs	Raspberry Pi 3			
	Average	Standard deviation	Inferior limit	Upper limit
G.711a	20.02	0.30	20.00	20.04
G.711u	20.02	0.30	20.00	20.04
Gsm	20.01	0.12	20.01	20.02
Speex	20.02	0.12	20.01	20.03
Ilbc	29.92	0.25	29.91	29.93
G.722	20.00	0.30	19.99	20.01
Banana Pi M3				
G.711a	20.00	0.06	20.00	20.01
G.711u	20.00	0.05	20.00	20.01
Gsm	20.00	0.06	20.00	20.01
Speex	20.00	0.12	20.00	20.01
Ilbc	30.01	0.41	29.99	30.02
G.722	20.00	0.27	20.00	20.01
Orange Pi Plus 2				
G.711a	20.00	0.06	20.00	20.01
G.711u	20.00	0.07	20.00	20.01
Gsm	20.00	0.11	20.00	20.01
Speex	20.00	0.12	20.00	20.01
Ilbc	30.01	0.17	30.00	30.01
G.722	20.00	0.10	20.00	20.01

**Table 7:** Delay IAX2. Unit milliseconds (ms)

Raspberry Pi 3				
Codecs	Average	Standard Deviation	Low Limit	Upper Limit
G.711a	0.02	0.00	0.02	0.02
G.711u	0.02	0.00	0.02	0.02
Gsm	0.01	0.02	0.02	0.02
Speex	0.02	0.00	0.02	0.02
Ilbc	0.03	0.00	0.03	0.03
Banana Pi M3				
G.711a	0.02	0.01	0.02	0.02
G.711u	0.02	0.01	0.02	0.02
Gsm	0.02	0.01	0.02	0.02
Speex	0.02	0.01	0.02	0.02
Ilbc	0.03	0.01	0.03	0.03
Orange Pi Plus 2				
G.711a	0.02	0.00	0.02	0.02
G.711u	0.02	0.00	0.02	0.02
Gsm	0.02	0.00	0.02	0.02
Speex	0.02	0.00	0.02	0.02
Ilbc	0.03	0.00	0.03	0.03

**Table 8:** Jitter SIP. Unit milliseconds (ms)

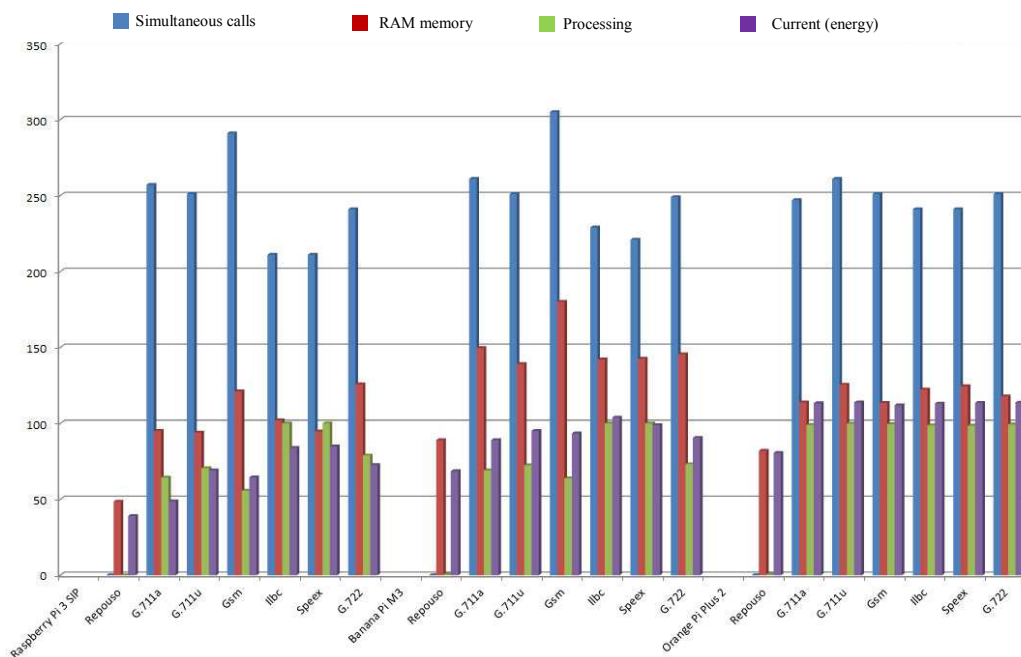
Raspberry Pi 3				
Codecs	Average	Standard Deviation	Low Limit	Upper Limit
G.711a	0.15	0.09	0.15	0.15
G.711u	0.15	0.09	0.15	0.15
Gsm	0.10	0.23	0.09	0.10
Speex	0.21	0.15	0.21	0.21
Ilbc	0.24	0.15	0.23	0.25
G.722	0.15	0.10	0.14	0.15
Banana Pi M3				
G.711a	0.03	0.01	0.03	0.03
G.711u	0.03	0.01	0.03	0.03
Gsm	0.03	0.01	0.03	0.03
Speex	0.06	0.02	0.06	0.06
Ilbc	0.24	0.08	0.24	0.25
G.722	0.15	0.09	0.14	0.15
Orange Pi Plus 2				
G.711a	0.04	0.01	0.04	0.04
G.711u	0.04	0.01	0.04	0.04
Gsm	0.06	0.03	0.06	0.06
Speex	0.06	0.02	0.06	0.06
Ilbc	0.09	0.04	0.10	0.10
G.722	0.06	0.03	0.06	0.06

**Table 9:** Jitter IAX2. Unit milliseconds (ms)

Raspberry Pi 3				
Codecs	Average	Standard Deviation	Low Limit	Upper Limit
G.711a	0.02	0.19	0.01	0.02
G.711u	0.02	0.19	0.01	0.02
Gsm	0.00	0.00	0.00	0.00
Speex	0.02	0.19	0.01	0.02
Ilbc	0.02	0.19	0.01	0.03
Banana Pi M3				
G.711a	0.02	0.19	0.01	0.03
G.711u	0.02	0.19	0.01	0.03
Gsm	0.02	0.19	0.01	0.02
Speex	0.02	0.19	0.01	0.02
Ilbc	0.03	0.19	0.01	0.02
Orange Pi Plus 2				
G.711a	0.02	0.19	0.01	0.02
G.711u	0.02	0.19	0.01	0.02
Gsm	0.02	0.19	0.01	0.02
Speex	0.02	0.19	0.01	0.02
Ilbc	0.02	0.19	0.01	0.02

**Table 10:** Data collected using the SIP protocol.

Raspberry Pi 3 SIP							
Codecs	Repose	Alaw	Ulaw	Gsm	Ilbc	Speex	G.722
Simultaneous calls	0.00	257.00	251.00	291.00	211.00	211.00	241.00
Memory (MB)	48.48	95.01	93.92	121.13	102.00	94.66	125.75
Processing (%)	0.05	64.36	70.55	55.72	100.00	100.00	78.89
Current (energy)	390.76	486.91	691.31	644.37	838.33	848.5	726.43
Banana Pi M3 SIP							
Simultaneous calls	0.00	261.00	251.00	305.00	229.00	221.00	249.00
Memory (MB)	89.04	149.62	139.13	180.09	142.13	142.68	145.57
Processing (%)	0.51	69.05	72.43	63.89	100.00	100.00	73.16
Current (energy)	686.11	890.39	950.45	934.21	1038.25	988.64	905.00
Orange Pi Plus 2 SIP							
Simultaneous calls	0.00	247.00	261.00	251.00	241.00	241.00	251.00
Memory (MB)	81.97	113.77	125.50	113.51	122.32	124.46	117.83
Processing (%)	0.56	98.97	99.82	99.48	98.67	98.53	99.34
Current (energy)	805.33	1133.69	1138.51	1119.26	1130.21	1135.64	1137.43



**Fig. 7:** SIP comparative analysis

Though the GSM CODEC has the lowest processing. The CODEC G.711a stands out because it is the CODEC that consumes less energy and supports a significant number of calls because it is an embedded device.

On the other hand, the Orange Pi Plus 2 device got a high processing in all the tests. Given this information, it is possible to state that this device is not ideal for use with Asterisk.

Figure 7 illustrates the behavior of the calls made on the 3 embedded devices, which allows better evaluation and comparison of the data. However, it is possible to observe that the Orange Pi Plus 2 device has identical results to all the CODECs addressed in this research, which reinforces the thesis that this device is not adequate to use with the Asterisk software.

### *SIP Calls with Transcoding*

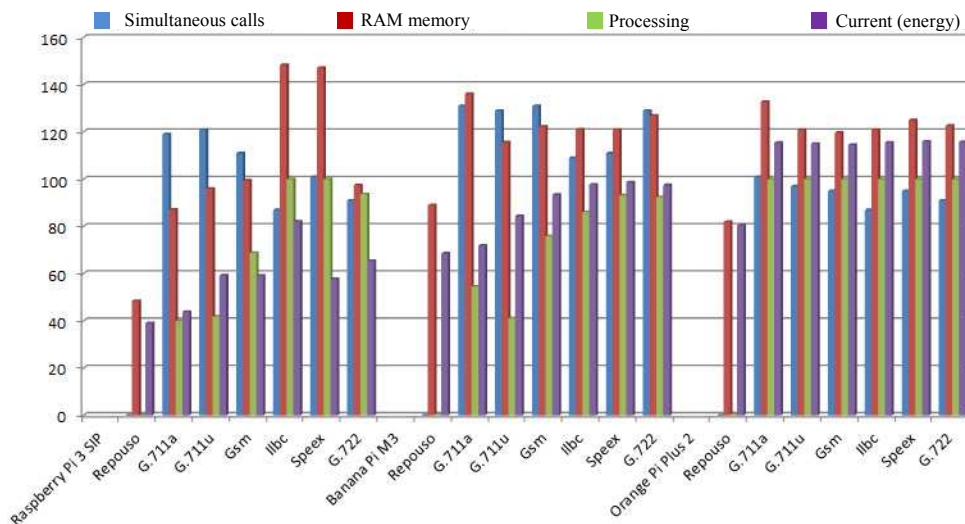
Table 11 shows the results collected on the calls made with the SIP protocol and with transcoding. It is possible to observe that with transcoding the performance of the devices is greatly reduced. Even so, the processing of the Orange Pi Plus 2 device remains high. This, once again, reinforces the claim that such a device is not suitable for Asterisk use. In the other devices, it is observed that the G.711a and G.711u CODECs have a higher number of concurrent calls supported and lower processing, memory and power consumption. In addition, the Raspberry Pi 3 device stands out with a considerable consumption of processing with CODECs G.711a and G.711u, as shown in Fig. 8.

**Table 11:** SIP with Transcoding

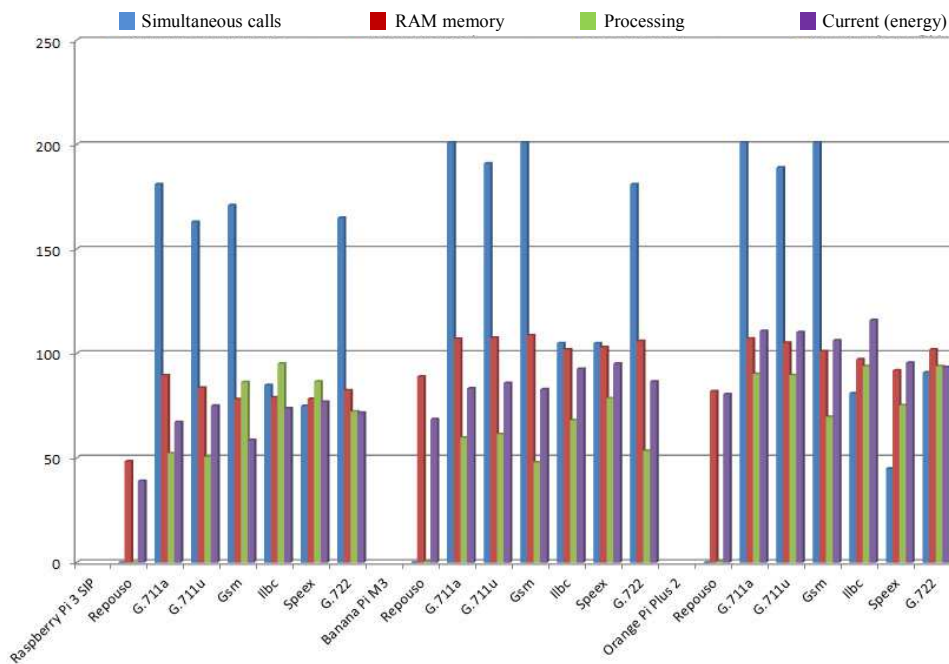
Raspberry Pi 3							
Codecs	Repose	Alaw	Ulaw	Gsm	Ilbc	Speex	G.722
Simultaneous calls	0.00	119.00	121.00	111.00	87.00	101.00	91.00
Memory (MB)	48.48	87.17	95.98	99.47	148.30	247.16	97.49
Processing (%)	0.05	3.97	41.87	68.83	100.00	100.00	93.70
Current (energy)	390.76	438.85	592.64	592.04	821.49	577.56	654.27
Banana Pi M3							
Simultaneous calls	0.00	131.00	129.00	131.00	109.00	111.00	129.00
Memory (MB)	89.04	136.14	115.63	122.30	121.01	120.95	126.86
Processing (%)	0.51	54.57	41.13	75.87	86.11	93.23	92.48
Current (energy)	686.11	718.81	844.7	934.56	977.82	987.33	975.25
Orange Pi Plus 2							
Simultaneous calls	0.00	101	97.00	95.00	87.00	95.00	91.00
Memory (MB)	81.97	132.74	120.92	119.70	120.95	124.95	122.67
Processing (%)	0.56	100.00	99.99	99.99	99.98	99.99	99.99
Current (energy)	805.33	1154.52	1150.26	1145.42	1155.57	1159.00	1157.58

**Table 12:** Data collected with the IAX2 protocol

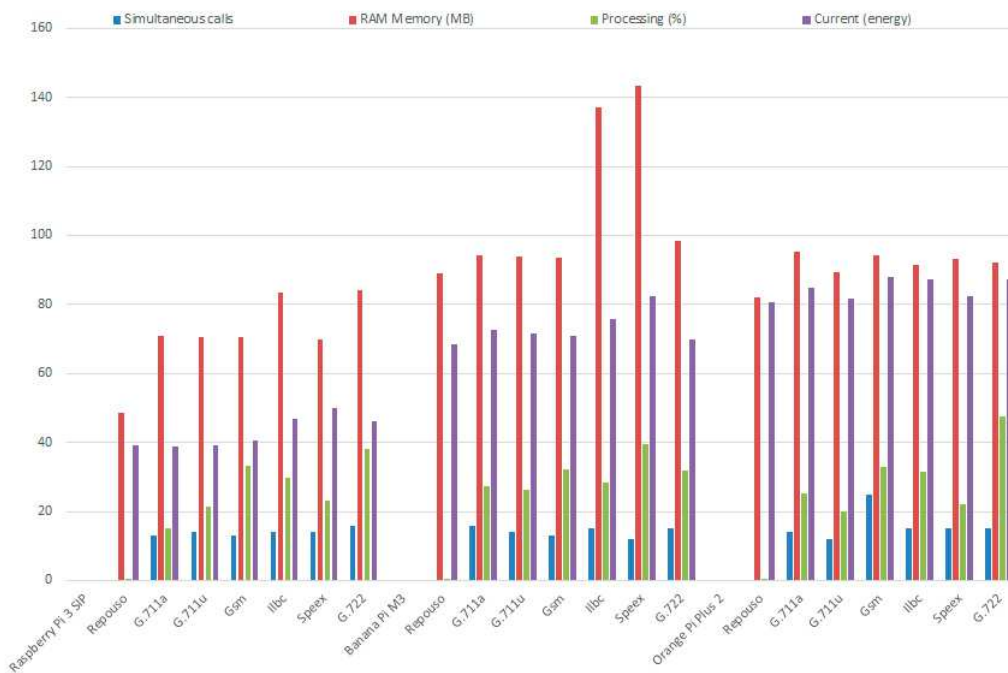
Raspberry Pi 3 IAX2							
Codecs	Repose	Alaw	Ulaw	Gsm	Ilbc	Speex	G.722
Simultaneous calls	0.00	181.00	163.00	171.00	85.00	75.00	165.00
Memory (MB)	48.48	89.54	83.66	78.16	78.99	78.38	82.49
Processing (%)	0.05	52.28	50.67	86.41	95.21	86.64	72.23
Current (energy)	39.07	67.28	75.12	58.58	73.81	76.87	71.73
Banana Pi M3 IAX2							
Simultaneous calls	0.00	201.00	191.00	201.00	105.00	105.00	181.00
Memory (MB)	89.04	107.09	107.62	108.77	101.94	103.04	106.08
Processing (%)	0.51	59.82	61.47	47.97	68.06	78.69	53.46
Current (energy)	68.61	83.39	85.96	82.92	92.7	95.29	86.69
Orange Pi Plus 2 IAX2							
Simultaneous calls	0.00	201.00	189.00	201.00	81.00	45.00	91.00
Memory (MB)	81.97	107.23	105.25	101.09	97.29	91.97	102.00
Processing (%)	0.56	90.30	89.77	69.78	94.00	75.27	94.00
Current (energy)	80.53	110.87	110.31	106.29	116.04	95.65	93.49



**Fig. 8:** SIP with transcoding comparative



**Fig. 9:** IAX2 Comparative analysis



**Fig. 10:** IAX2 with Transcoding Analysis

*IAX2 Calls*

This subsection addresses the data collected using the IAX2 protocol. They are shown in Table 12. It is possible to note that the G711.a CODEC stands out with low memory, processing and power consumption, and a high number of simultaneous calls on the Raspberry Pi 3

and Banana Pi M3 devices. The Orange Pi Plus 2 was highlighted by the GSM CODEC due to the high number of simultaneous calls, low processing and power consumption, and moderate memory consumption. However, if compared to the other devices addressed in this research, this is the one that has inferior performance as shown in Fig. 9.

**Table 13:** Data collected with the IAX2 protocol with Transcoding

Raspberry Pi 3							
Codec's	Repose	Alaw	Ulaw	Gsm	Ilbc	Speex	G.722
Simultaneous calls	0.00	13.00	14.00	13.00	14.00	14.00	16.00
Memory (MB)	48.48	70.81	70.43	70.43	83.48	69.95	84.02
Processing (%)	0.05	14.98	21.46	33.35	29.92	23.30	37.99
Current (energy)	390.76	389.51	392.62	403.92	467.55	498.12	463.00
Banana Pi M3							
Simultaneous calls	0.00	16.00	14.00	13.00	15.00	12.00	15.00
Memory (MB)	89.04	94.18	93.79	93.66	137.07	143.37	98.33
Processing (%)	0.51	27.34	26.44	32.18	28.42	39.57	31.74
Current (energy)	686.11	725.73	716.77	708.26	758.22	823.97	697.62
Orange Pi Plus 2							
Simultaneous calls	0.00	14.00	12.00	25.00	15.00	15.00	15.00
Memory (MB)	81.97	95.12	89.29	94.15	91.29	93.15	92.26
Processing (%)	0.56	25.32	20.06	32.92	31.63	22.05	47.52
Current (energy)	805.33	846.61	817.15	879.85	874.27	825.30	873.12

### *IAX2 Calls with Transcoding*

Table 13 shows the data collected using the IAX2 protocol with transcoding. Immediately, it is possible to observe that there is a low number of concurrent calls supported. Given this context, it is possible to state that the embedded devices addressed in this research do not support the process of transcoding CODECs with the IAX2 protocol. Only in this test the Orange Pi Plus 2 device did not report high processing consumption, as shown in Fig. 10.

### **Conclusion and Future Work**

In this work, an approach was performed to analyze the performance and efficiency energy in three state-of-art embedded devices using the Asterisk software of voice over IP, which measured the jitter, delay and bandwidth with SIP and IAX2 protocols with CODEC's G.711a (Alaw), G.711u (Ulaw), G.722, Ilbc, Speex and Gsm.

The measurements were performed in order to compare the three embedded devices with the use of Asterisk. However, the results showed great similarity in the data, both with the SIP protocol and with the IAX2 protocol, this in the network requirement.

We also verified the number of concurrent calls supported in each device with the SIP and IAX2 protocols, both in normal calls and in transcoded calls and, in parallel, the analysis of the RAM memory, processing and energy consumption was performed.

The prototyping was performed to compare the three embedded devices using the Asterisk. The results were surprising. The Raspberry Pi 3 and Banana Pi M3 devices satisfactorily support a high number of simultaneous calls with moderate memory, processing and power consumption through CODEC G.711a and G.711u. However, the Orange Pi Plus 2 device showed a

high processing power. Thus, it is possible assert that this device is not suitable for use with Asterisk.

All the 3 devices showed stability throughout the research. Not occurring unintentional restart of the equipment, even during high loads.

The performance of the 3 embedded devices discussed in this work were evaluated with the purpose of finding the best device to support the communication system by the Asterisk voice over IP. Nevertheless, new embedded devices, as well as new technologies, will emerge. In this way, the possibility of extending this work is certain.

As future work, it is proposed to carry out experiments with CODEC Opus, since it was not possible to perform the compilation on the embedded devices due to incompatibility. It is also possible to carry out the same approach with the IAX2 protocol, using encryption. We can to propose a comparison of the behavior of the IAX2 protocol with transcoding on computing platforms with x86 architecture. It should be performed the same bandwidth, jitter and delay tests with other tools available in the market, specific to VoIP packet analysis, such as NetQuality Voip and SIP Tester.

### **Acknowledgement**

This research work is possible through the and support of all, including family, future wife, friends and especially in recognition of gratitude for my Professors, Dr. Admilson de Ribamar Lima Ribeiro and Dr. Edward David Moreno Ordóñez, thank you for all the support and encouragement.

### **Author's Contributions**

**Adauto Cavalcante Menezes:** Participated in all experiments, coordinated the data-analysis and contributed to the writing of the manuscript.

**Toniclay Andrade Nogueira:** The author organized the study, participated in all experiments, coordinated the data-analysis.

**Edward David Moreno Ordonez:** The author designed the research plan, gave final approval of the version to be submitted, as well as supervised and coordinated the entire work.

**Admilson de Ribamar Lima Ribeiro:** The author designed the research plan, gave final approval of the version to be submitted, as well as supervised and coordinated the entire work.

## Ethics

The authors confirm that they abide to all ethical protocols and procedures while preparing this manuscript.

## References

- Abid, F., N. Izeboudjen, M. Bakiri, S. Titri and F. Louiz, 2012. Embedded implementation of an IP-PBX /VoIP gateway. Proceedings of the 24th International Conference on Microelectronics, (ICM' 12), Algeria, pp: 5-8.
- Androulidakis, I.I., 2016. VoIP and PBX Security and Forensics.
- Bernal, P.S.M., 2007. Voz sobre protocolo IP: A nova realidade da telefonia. São Paulo.
- Bryant, R., L. Madsen and J.V. Meggelen, 2013. Asterisk: The Definitive Guide. Sebastopol, 4th Edn., O'Reilly Media, Inc., Sebastopol, ISBN-10: 1449332455, pp: 846.
- Colcher, S., A.T.A. Gomes, A.O. da Silva, G.L.d.S. Filho and L.F.G. Soares, 2005. VoIP: voz sobre IP. Rio de Janeiro.
- Digium, 2017. Dimensioning, [accessed 2017 Dez 10]. Available: <http://www.digium.com/blog/2012/09/25/asteriskdimensioning-what-server-do-i-need>.
- Edan, N. M., A. Al-Sherbaz, S. Turner and S. Ajit, 2016. Performance evaluation of QoS using SIP and IAX2 VVoIP protocols with CODECS. Proceedings of the SAI Computing Conference, 13-15 Jul., IEEE Xplore press, London, UK, pp: 631-636. DOI: 10.1109/SAI.2016.7556048
- Jain, R., 1991. The Art of computer systems performance analysis: Techniques for experimental design, measurement, simulation and modeling. New York.
- Rfc1890, 2017. RTP Profile for audio and video conferences with minimal control. RFC.
- Rfc5456, 2016. Rfc5456.
- Sulkin, A., 2002. PBS systems for Ip telephony: Migrating enterprise communications. Sykesville.
- Tesfamicael, A.D., V. Liu, W. Caelli and J. Zureo, 2014. Implementation and evaluation of open source unified communications for SMBs. Proceedings of the International Conference on Computational Intelligence and Communication Networks, Jul. 14-16, IEEE Xplore press, Bhopal, India, pp: 1243-1248. DOI: 10.1109/CICN.2014.260
- Villac'is, D., F.R. Acosta and R.A. Lara Cueva, 2013. Performance analysis of VoIP services over WiFi-based systems. Proceedings of the IEEE Colombian Conference on Communications and Computing, May 22-24, IEEE Xplore press, Medellin, Colombia, pp: 1-6. DOI: 10.1109/ColComCon.2013.6564813