

UNIVERSIDADE FEDERAL DO ABC

FELIPPE ZANIN

ANALISE DE DESEMPENHO DE SISTEMAS VOIP BASEADOS EM REDES SEM
FIO

SANTO ANDRÉ, SP

2017

FELIPPE ZANIN

ANALISE DE DESEMPENHO DE SISTEMAS VOIP BASEADOS EM REDES SEM
FIO

Projeto de trabalho de Graduação
da Universidade Federal do ABC
sob orientação do Prof.Dr. Luiz
Henrique Bonani do Nascimento

SANTO ANDRÉ, SP

2017

RESUMO

Este trabalho abordará o desempenho de uma central VoIP em redes sem fio. Com o crescimento do uso do VoIP e a expansão das redes sem fio, um estudo sobre o desempenho dos dois juntos seria interessante devido às vantagens que ambos os conceitos fornecem. Este estudo se baseia em um sistema VoIP construído em um hardware embarcado e utilizando redes sem fio. Os resultados obtidos nesse estudo apontam que dependendo das condições é possível a utilização dos dois conceitos juntos.

Palavras-chave: VoIP. Wireless. Desempenho. Embarcado. Asterisk.

Sumário

1 Introdução	01
2 Fundamentos de voz sobre IP	03
3 Hardware	09
3.1 Hardwares Embarcados	09
3.1.1 Raspberry Pi	10
3.2 Hardware e Protocolos para Redes sem Fio	10
4 Softwares	15
4.1 Asterisk	15
4.1.1 FreePBX	16
4.2 Softphones	17
4.3 SIPp	17
4.4 Wireshark	18
5 Construindo uma central VoIP baseada em Wi-Fi	21
5.1 Especificações dos Hardwares e Softwares utilizados	21
5.1.1 Raspberry	21
5.1.2 Computador 1	22
5.1.3 Computador 2	22
5.1.4 Roteador	23
5.2 Instalação do ambiente de teste	23
5.2.1 Sistema Operacional e Softwares na Raspberry	23
5.2.2 Codec G729 na Raspberry	23
5.2.3 Sistema Operacional e Softwares na máquina virtual	24

5.3 Utilização do ambiente de teste	24
5.4 Metodologia	26
5.4.1 Métricas	28
5.4.1.1 Cálculo de MOS	29
6 Resultados	31
6.1 Teste da capacidade da Raspberry utilizando uma rede cabeada	31
6.2 Potência do Sinal	33
6.3 Perda de Pacotes	34
6.4 Jitter	37
6.5 Latência	40
6.6 MOS	43
7 Conclusão	45
8 Referências Bibliográficas	47
ANEXO A – Cenário 1 do SIPp utilizando o codec G711	49
ANEXO B – Cenário 2 do SIPp utilizando o codec G711	54
ANEXO C – Cenário 1 do SIPp utilizando o codec G729	59
ANEXO D – Cenário 2 do SIPp utilizando o codec G729	64
ANEXO E – Cenário 1 do SIPp utilizando o codec GSM	69
ANEXO F – Cenário 2 do SIPp utilizando o codec GSM	74

1 Introdução

O início das telecomunicações é dado pela primeira transmissão de voz via cabo realizada por Graham Bell em 1876. Desde então, as telecomunicações estão em constante evolução, para cada vez mais se adaptar às novas necessidades da sociedade [1].

As redes de telefonia convencionais são denominadas de Rede Pública de Telefonia Comutada (*Public Switched Telephone Network* – PSTN), que permitem as transmissões telefônicas comutadas por circuitos. Esse conceito de rede começou a ser desenvolvida no começo da telecomunicação, inicialmente utilizando tecnologias analógicas e posteriormente migrando para tecnologias digitais, que são utilizadas até os dias de hoje [1].

Com a evolução da computação uma nova necessidade surgiu: a transmissão de dados. No início a PSTN foi utilizada para essa finalidade. Porém, como ela não havia sido desenvolvida para esse propósito seu desempenho não foi satisfatório, então uma nova rede específica para a transmissão de dados teve que ser criada para suprir de forma eficiente essa nova necessidade. Utilizando essa nova rede, agora específica para a transmissão de pacotes, um novo conceito de telefonia foi criado, de forma a transmitir voz através dela, além dos dados. Esse novo conceito foi chamado de Voz sobre IP (*Voice over IP* – VoIP) [2].

O VoIP pode ser considerado o mais novo avanço na área de telefonia, teve seu surgimento em meados dos anos 1990, porém por muito tempo não foi possível utilizá-lo com eficiência devido a limitações nas redes de internet. Conforme as redes foram melhorando, essa nova tecnologia pode evoluir e ser mais utilizada. Podendo futuramente substituir as redes de telefonia convencional [2].

As transmissões de dados também evoluíram com o tempo conforme novas necessidades foram surgindo. Inicialmente, as transmissões eram feitas apenas através de cabos. Porém, utilizar cabos faz com que se tenha problemas com mobilidade. Então, uma nova forma de transmissão teve de ser desenvolvida para suprir esse problema: as redes sem fio. O desenvolvimento começou, também, nos anos 1990, e estão evoluindo até os dias de hoje para melhorar a taxa de transmissão dos dados.

As tecnologias que utilizam o conceito de VoIP, são dependentes da qualidade da rede em que ela está sendo transmitida. As redes sem fio, apresentam grandes variações na qualidade da transmissão, que em geral não atrapalha para a transmissão de dados simples. Dessa forma, seria contra intuitivo utilizar esses dois conceitos juntos. Porém, se os dois conceitos pudessem ser unidos, novas tecnologias poderiam ser criadas utilizando os benefícios que cada um oferece.

O estudo a ser apresentado visa unir esses dois conceitos, VoIP e redes sem fio, com um hardware embarcado com a finalidade de criar uma central VoIP móvel. E realizar uma análise do desempenho em diferentes condições, para assim

poder concluir se é possível a utilização desses dois conceitos juntos de forma satisfatória, e em que condições.

A apresentação deste estudo está organizada da maneira como se segue. O primeiro capítulo mostrará o panorama geral da tecnologia VoIP, e em seguida aprofundará nos conceitos fundamentais para seu funcionamento, como os protocolos utilizados. Em seguida apresentará as desvantagens que limitam seu desempenho e vantagens da utilização dos sistemas VoIP.

O segundo capítulo elucidará sobre os conceitos dos hardwares a serem utilizados para a confecção deste estudo. Inicialmente, abordando sobre hardwares embarcados e concluindo sobre um dos principais conceitos deste estudo, que são as redes sem fio. No capítulo de redes sem fio, será apresentado um breve histórico, em seguida alguns conceitos físicos do seu funcionamento, juntamente com a capacidade que ela pode fornecer.

O terceiro capítulo apresentará os softwares que serão utilizados para a confecção do estudo e o motivo da escolha de cada um deles. Inicialmente, será apresentado qual o sistema VoIP a ser utilizado, seguido pelo software que funcionará como telefone, e por fim o software utilizado para realizar os testes de desempenho.

O quarto capítulo mostrará a confecção do projeto, desde as especificações de cada equipamento utilizado, até a implementação e configuração dos diferentes softwares. Finalizando, ele apresenta com uma explicação sobre a metodologia que será utilizada.

O quinto capítulo mostrará os resultados obtidos nos experimentos, junto com uma análise sobre o desempenho em cada condição, unindo com os conceitos descritos nos capítulos teóricos.

O sexto capítulo apresentará uma conclusão acerca dos resultados, concluindo se é possível utilizar sistemas VoIP com redes sem fio, e em quais condições é aplicável.

Além desses, há seis anexos, de A a F, com os scripts XML dos cenários utilizados nos experimentos deste estudo.

2 Fundamentos de voz sobre IP

Com o uso crescente do VoIP e o desuso dos meios de comunicações de voz convencionais, como a comunicação através de par-trançado E1¹, abriu-se espaço para os meios de comunicação de voz baseados em pacotes, como o VoIP. Este vem crescendo cada vez mais nos últimos anos, isso devido à simplicidade de implementação e da melhoria da qualidade da infraestrutura de redes e da internet. Com a necessidade da redução de custos, a maioria das empresas e entidades começou a optar por implementar tecnologias VoIP, por ser mais barata, pois normalmente utiliza a rede de dados já implementada, oferecendo recursos adicionais em relação às centrais analógicas, como a possibilidade da criação de novos módulos de forma simplificada.

Os módulos em centrais telefônicas são aplicações criadas com a finalidade de suprir alguma necessidade específica, como atendedores automáticos e sistema de gravação de chamadas. Estes são facilmente desenvolvidos em sistemas e VoIP, diferentemente dos sistemas telefônicos convencionais, que exigiam equipamentos específicos mais caros para realizar tais tarefas.

Os sistemas VoIP baseiam-se na transmissão da voz usando protocolos de internet. Foi necessário o desenvolvimento de protocolos específicos para que transmissões de voz pela internet fossem feitas, como o protocolo SIP, que é utilizado para a sinalização. Os principais protocolos utilizados pelos sistemas VoIP serão explicados a seguir:

Protocolo de Iniciação de Sessão (SIP – Session Initiation Protocol) é responsável por estabelecer a sessão entre dois terminais de comunicação, também fazendo a negociação da capacidade de transmissão entre estes [1].

O protocolo H.323 tem a mesma função que o SIP, porém ele é menos utilizado que o anterior, mesmo tendo uma quantidade de recursos maior, devido principalmente à sua complexidade [1].

O protocolo de transmissão em tempo real (*Real Time Protocol* - RTP) é um protocolo utilizado para transmissões multimídia, como a voz sobre IP. Ele trafega

¹ E1 é um padrão de linha telefônica europeu utilizado no Brasil que consiste em uma conexão com aproximadamente 2Mbps de taxa de transferência divididos em 32 canais, sendo 30 destinados a voz. [1]

junto com o protocolo UDP (*User Datagram Protocol*) e é responsável por criar o fluxo de pacotes de uma transmissão multimídia. Isso é feito adicionando ao cabeçalho do pacote informações como o número da sequência e o *timestamp*. O RTP é implementado junto com outro protocolo chamado RTCP (*Real Time Control Protocol*), que fornece informações da qualidade da conexão, dando suporte a dispositivos garantirem a qualidade da conexão [1].

O protocolo UDP está implementado na camada de transporte, é um protocolo de transporte rápido não orientado a conexão. O RTP é encapsulado neste protocolo para a transmissão de dados multimídia e é usado para aplicações que não necessitam de retransmissão, ou seja, se um pacote não for entregue ele não irá verificar isso para realizar uma eventual retransmissão. No caso de chamadas de voz, ele torna-se a melhor escolha, devido a não realizar retransmissões de pacotes, uma vez que retransmissões em transmissões em tempo real poderiam gerar confusão no fluxo de uma conversa, pois os pacotes poderiam chegar em uma ordem diferente da transmitida, podendo tornar a conversa sem nexos ou aumentando o atraso para conseguir reorganizar os pacotes [1].

Devido a sistemas VoIP serem baseados na estrutura da internet, ele apresenta algumas vantagens e desvantagens em relação à telefonia convencional. As principais desvantagens se dão por problemas relacionados à qualidade das conexões com a internet. Os parâmetros que influenciam na qualidade das ligações do VoIP são:

- **Latência:** Esse parâmetro em VoIP refere-se ao atraso que a voz demora para chegar ao ouvinte. Diversos fatores influenciam nesse atraso, entre os quais: a distância entre as partes, em que quanto maior a distância maior será o atraso de propagação; o atraso de processamento, que devido à utilização da rede de computadores é gerado por exemplo, na conversão da voz em pacotes; por fim, o atraso de fila, em redes de dados ocorre um problema que são as filas, as filas são causadas devido ao grande fluxo de dados que trafegam nas redes, que ultrapassam a capacidade dos roteadores, fazendo com que pacotes passem um período esperando em filas gerando um atraso[1].
- **Jitter:** Este parâmetro refere-se às diferenças de tempo entre a chegada dos pacotes no destino, ou seja, à variação de tempo com que pacotes sucessivos conseguem chegar no receptor em um intervalo regular [1].

- Perda de pacotes: Este parâmetro refere-se à perda de informação até chegar ao destino. Isso pode impossibilitar a comunicação, pois partes importantes dos dados podem ser perdidos, fazendo que não seja possível recuperar o que está sendo transmitido [1].

Apesar dos problemas com a qualidade de voz que os sistemas VoIP apresentam em decorrência da utilização da estrutura da internet, os sistemas VoIP também apresentam vantagens atrativas para empresas. A principal vantagem é a redução de custo em relação aos métodos convencionais de telefonia. Essa economia pode ser vista, por exemplo, na interligação entre duas sedes de uma mesma empresa em diferentes países, que é feita utilizando apenas dois servidores, fazendo com que não tenha custos para realizar uma ligação entre elas. Outra vantagem interessante, como citado anteriormente, é a criação de diferentes módulos, sendo que com uma equipe pequena uma empresa pode desenvolver aplicações específicas conforme suas necessidades.

Outro requisito importante para comunicação de voz via pacotes são os codecs. Os codecs são a forma com a qual a voz é codificada e decodificada, cada um usando um algoritmo diferente para realizar a codificação. Dessa forma alguns tornam-se mais eficientes que outros dependendo da aplicação. Na Tabela 1 é possível ver a diferença nas características de alguns dos principais codecs existentes.

Na tabela é possível notar a diferença na taxa de transmissão de cada um dos codecs. Dessa forma, um conceito importante para o bom funcionamento é o de qualidade de serviço (QoS - *Quality of Service*), em que técnicas são aplicadas com a finalidade de tentar garantir uma determinada largura de banda e nível de latência para uma aplicação específica, dessa forma podendo garantir a taxa de transmissão necessária para a transmissão da voz com o codec desejado [1].

Tabela 1 – Características dos principais codecs.

Codec	Algoritmo	Tamanho do Frame/Atraso de algoritmo	Taxa Usual	Comentários
G.711	PCM	0.125ms/0	64Kb/s	Uso universal
G.722		0.125ms/1.5ms	48, 56 ou 64Kb/s	Codificador de banda larga
G.726	ADPCM	0.125ms/0	32Kb/s	Grande qualidade, baixa complexidade
G.728	LD-CELP	0.625ms/0	16Kb/s	Grande qualidade em tandem; Recomendado para cabos;
G.729(A)	CS-ACELP	10ms/5ms	8Kb/s	Mais utilizado
G.729e	CELP Híbrido	10ms/5ms	11.8Kb/s	Grande qualidade e complexidade; Recomendado para cabos
G.723.1(6.3)	MPC-MLQ	30ms/7.5ms	6.3Kb/s	Utilizado em multimídia
G.723.1(5.3)	ACELP	30ms/7.5ms	5.3Kb/s	Utilizado em multimídia
GSM	ACELP	20ms/7.5ms	13Kb/s	Utilizado em redes celular
iLBC		20, 30ms	15.2, 13.33Kb/s	Desenvolvido para a utilização em redes IP

Fonte: [3] p.1497

Outro dado importante sobre os codecs é o tamanho do payload de voz, o payload é o tamanho em bytes da voz que está sendo alocada em cada pacote. Com esse dado é possível calcular a taxa de transmissão que está presente na Tabela 1. O payload dos codecs que serão utilizados nesse estudo são:

- G711 apresenta um payload de 160 bytes.
- G729 apresenta um payload de 20 bytes.
- GSM apresenta um payload de 33 bytes.

Para medir a qualidade das ligações foi criada uma metodologia de testes chamada *Mean Opinion Score* (MOS). Ela é uma metodologia subjetiva, pois a qualidade de áudio pode ser diferente para cada pessoa, tornando assim algo subjetivo. Essa metodologia consiste em entregar a um grupo de pessoas um áudio em determinada circunstância com o codec ao qual deseja-se testar, e assim as pessoas avaliam em uma escala de 1(ruim) até 5(excelente) [1].

A metodologia dos testes de MOS está padronizada na recomendação P.800 da ITU-T, que é o setor de normatização das telecomunicações da União Internacional de Telecomunicações. Os experimentos devem ser feitos sobre determinadas circunstâncias, como por exemplo o ambiente de teste deve ser uma sala silenciosa, com tempo de reverberação menor que 500ms e o ruído interno ser menor que 30dBA. Os áudios a serem tocados devem ser frases aleatórias curtas de fácil entendimento, sem conexão uma com a outra e o número de frases é decidido por quem for realizar o experimento. O número de pessoas em cada experimento, não é especificado pela norma, porém deve se ter um número estatisticamente considerável [4].

Após os experimentos é realizada uma média das opiniões e com o resultado final é possível identificar a eficiência de cada codec em diversas circunstâncias. Na Tabela 2 temos um exemplo dos testes realizados para alguns dos principais codecs utilizados.

Tabela 2 – Pontuação MOS de alguns codecs.

Método de Compressão	Bit rate (Kbps)	Tamanho do Frame (ms)	Pontuação MOS
G.711 PCM	64	0.125	4.1
G.726 ADPCM	32	0.125	3.85
G.728 LD-CELP	15	0.625	3.61
G.729 CS-ACELP	8	10	3.92
G.729a CS-ACELP	8	10	3.7
G.723.1 MP-MLQ	6.3	30	3.9
G.723.1 ACELP	5.3	30	3.65
iLBC Freeware	15.2, 13.3	20, 30	3.9

Fonte: [3] p.124

Os sistemas VoIP, devido à dependência da qualidade da rede que ele irá operar, é sustentado hoje pela infraestrutura cabeada, mas as redes sem fio estão a cada dia sendo mais utilizadas em comparação com a infraestrutura cabeada, e estão melhorando seu desempenho. Com isso para a realização do estudo de como uma central

móvel, usando rede *wireless*² poderia se comportar em termos de desempenho, será necessária a apresentação do hardware usado para a implementação, e as principais características das redes sem fio.

² Termo em inglês usual para definir uma rede que não utiliza cabos nas transmissões.

3 Hardware

Os sistemas VoIP podem ser implementados em diversos tipos de hardware devido a sua estrutura, em geral, estar presente em um sistema operacional convencional, como o Linux e o Windows. Dessa forma, é fácil de ser implementado, por exemplo em hardwares embarcados, como a Raspberry PI.

As redes de comunicação utilizadas em sistemas VoIP, em geral, são redes com fio que são uma forma mais estável para a comunicação. Porém as redes sem fio podem se tornar uma opção interessante em decorrência da sua mobilidade. Contudo, essas redes sofrem com o problema de serem instáveis.

3.1. Hardwares Embarcados

Hardwares embarcados vêm se popularizando a cada dia, pela facilidade de uso, baixo preço e por serem capazes de realizar as mesmas funções de um computador convencional, como um desktop ou um notebook. Porém, eles apresentam outra aplicação que também está se popularizando, que é o uso deles como servidores. Contudo, muitos apresentam um desempenho modesto, pois as suas especificações, como tipo de processador, memória, placa gráfica, capacidade de armazenamento e placas de rede, tanto a sem fio quanto a cabeada, são normalmente inferiores à dos computadores convencionais. Portanto, sua aplicação como servidor normalmente é limitada, ou seja, é adequado apenas para aplicações que não vão exigir uma grande capacidade de processamento ou armazenamento.

Existem diversos modelos de hardwares embarcados, cada um com especificações diferentes para aplicações diferentes, como os hardwares de roteadores, que são projetados para serem eficientes para a aplicação de redes. Neste estudo será utilizada a Raspberry Pi, que é um hardware embarcado para uso geral, ou seja, é possível utilizá-la para diversas aplicações. Existem outros modelos de hardwares embarcados de uso geral, como a Raspberry, Cubieboard, Cubietruck, dentre outros. Porém a escolha da Raspberry Pi neste estudo se dá por sua popularidade e pelo fato de existirem mais softwares otimizados.

3.1.1. Raspberry Pi

A Raspberry Pi se apresenta em diversos modelos. Em todos é utilizado um processador de arquitetura ARM, que é um processador específico para hardwares embarcados. A Raspberry Pi 3, por exemplo, apresenta as seguintes especificações: um processador 1,2GHz quad-core ARM Cortex-A53, com 1GB de memória, com uma porta ethernet, em que o sistema operacional é executado em um Micro SD. Dessa forma, pode-se notar que a aplicação dela é apenas para servidores pequenos em que não irá apresentar um alto nível de tráfego de dados [5]. Na Figura 1 é apresentada a imagem de como é a Raspberry Pi 3.

Devido à Raspberry utilizar um processador ARM, as versões de sistemas operacionais a serem instalados na mesma devem sofrer algumas modificações para funcionarem. Isso ocorre devido ao núcleo do sistema operacional ser diferente. Um exemplo de sistema operacional desenvolvido para Raspberry é o Raspbian, uma versão modificada do sistema operacional Debian, com o núcleo desenvolvido para a arquitetura ARM, que é a arquitetura mais utilizada em sistemas embarcados, devido ao seu baixo custo.



Figura 1 – Raspberry Pi 3 Modelo B

Fonte: [6]

3.2. Hardware e Protocolos para Redes sem Fio

Redes sem fio são uma infraestrutura de comunicação entre equipamentos com o envio dos dados por meio de ondas eletromagnéticas não guiadas. Esse conceito torna a forma de comunicação algo mais interessante, pois fornece uma maior mobilidade para os equipamentos, não ficando limitados apenas a posições onde

está disponível uma conexão cabeada. Devido a essa forma de transmissão, as redes sem fio sofrem com problemas de instabilidade, pois diversos fatores do ambiente podem interferir no sinal, impactando em parâmetros de desempenho. Existem diversos padrões de comunicação sem fio, dependendo da aplicação que será utilizada. Este estudo abordará as redes locais sem fio.

O surgimento das redes locais sem fio ocorreu na época do aparecimento dos notebooks, em que uma tecnologia de acesso à internet, que fosse móvel, era necessária para acompanhar a mobilidade que os notebooks forneciam. Então em 1990 iniciou-se o desenvolvimento dessa tecnologia com a criação de um novo protocolo. Porém para ser possível todas as redes se comunicarem, um novo padrão foi criado. Em 1997 o comitê do IEEE (Instituto de Engenheiros Eletricistas e Eletrônicos) criou o padrão denominado de 802.11, que ficou posteriormente conhecido como Wi-Fi³ [7].

As redes sem fio utilizam ondas eletromagnéticas que se propagam no espaço livre, em determinadas bandas de frequências, e com os dados modulados. As bandas de frequência utilizadas no 802.11 foram um problema inicial para serem escolhidas, pois é difícil encontrar uma banda de frequência que esteja disponível. Então acabou sendo escolhida uma banda de frequência livre ao invés de uma banda licenciada, que poderia tornar a tecnologia cara. Foi então escolhida a banda ISM (*Industrial, Scientific and Medical*) definida pelo ITU-R, que é a divisão de comunicação à rádio da União Internacional de Telecomunicações (ITU). Assim, as bandas escolhidas foram o 2.4GHz e 5GHz [8].

A principal topologia de rede sem fio utilizada é chamada infraestrutura, em que é utilizado um ponto de acesso principal, com vários clientes conectados nele, em que toda a informação transmitida por cada cliente passa por ele, como pode ser visto na Figura 2. Na figura é possível verificar três computadores conectados em um ponto de acesso. E esse ponto de acesso está conectado a uma rede cabeada.

³ Wi-Fi é uma marca de uma organização chamada Wi-Fi Alliance que certifica produtos de rede local sem fio que utilize o padrão 802.11, e passe por um teste de interoperabilidade.

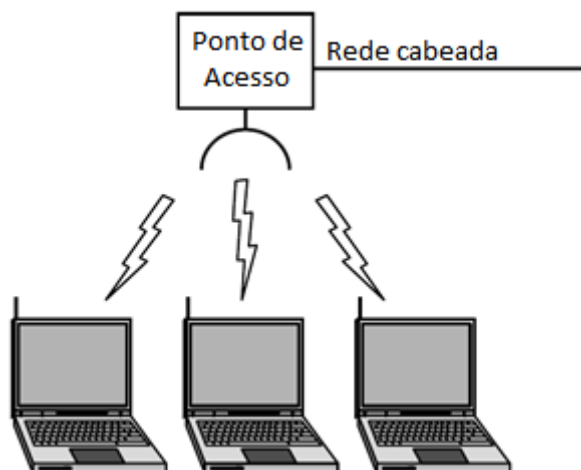


Figura 2 – Topologia de rede sem fio com ponto de acesso.

Fonte: [8], p.43

Como as ondas se propagam no ambiente, diversos fatores podem influenciar no desempenho desta tecnologia. As frequências que foram escolhidas no 802.11 sofrem com o problema de reflexão em objetos sólidos. Dessa forma, são gerados ecos de transmissão, o que faz com que o sinal atinja o receptor por vários caminhos. Assim, o sinal que é recebido sempre flutua bastante, e esse fenômeno é denominado de desvanescimento por múltiplos caminhos, comumente chamado de *fading*. Esse efeito pode ser visto na Figura 3. No lado esquerdo da figura é mostrado o percurso que o sinal está percorrendo sofrendo reflexões chegando no receptor por dois caminhos diferentes. Enquanto na direita é mostrado como sinais sem sofrer com o efeito de *fading* e sofrendo o efeito de *fading* chegariam ao receptor.

Para contornar o problema de *fading* foi criada uma técnica para a transmissão, que é a transmissão por vários caminhos independentes. Assim, a informação sempre chega ao receptor por algum caminho. Esse método normalmente é implementado por meio de modulações digitais na camada física, que consiste em transmitir esses caminhos em diferentes frequências da banda, repetir bits por diferentes períodos, ou seguir diferentes caminhos espaciais entre diferentes pares de antenas [8].

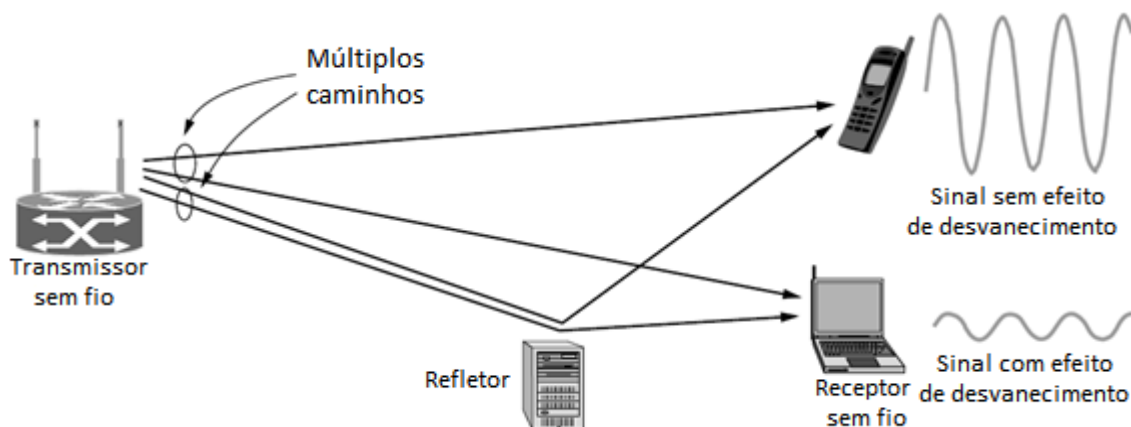


Figura 3 – Efeito desvanecimento por muito caminhos.

Fonte: [8], p.44

Utilizando esses métodos de caminhos diferentes, foi lançada a primeira versão do 802.11 em 1997. Ela atingia taxas de transmissão de 1 ou 2Mbps, utilizando o método de transmitir os caminhos em diferentes frequências da banda. Porém, essa primeira versão não foi bem aceita devido à taxa de transmissão ser muito baixa. Então foi desenvolvida uma nova tecnologia: a 802.11b, em 1999, que também utiliza o espalhamento na banda. Porém foi melhorada de forma a conseguir atingir taxas de até 11Mbps [8].

Mesmo assim, ainda eram necessários avanços para melhorar a taxa de transmissão, o que levou ao desenvolvimento de uma nova técnica de modulação chamada de OFDM (Multiplexação ortogonal por divisão de frequência). Essa técnica consiste na divisão da banda total do espectro em muitas fatias estreitas, permitindo assim que vários bits sejam enviados de forma paralela.

Com a utilização da técnica de OFDM foi criada uma nova versão do protocolo 802.11. A nova versão criada foi o 802.11g, que trabalha na banda de 2.4GHz, e atinge 54Mbps de taxa de transmissão. A versão 802.11a, que opera em 5GHz, e havia começado a ser desenvolvido junto com o padrão 802.11b, e não havia sido finalizado, foi modificado também para utilizar o OFDM. Com essa modificação, o padrão 802.11a conseguiu também atingir uma taxa de transmissão de até 54Mbps.

Mesmo 54Mbps sendo uma taxa da transferência razoável, a necessidade de taxas de transmissão maiores devido à alta demanda de dispositivos

móveis, implicou no desenvolvimento de uma nova tecnologia. O 802.11n, de 2009, utiliza uma técnica de múltiplas antenas e uma banda de frequências maior, gerando assim taxas de transmissão que vão de 54Mbps até 600Mbps, dependendo do número de antenas. Este último padrão é o mais utilizado atualmente [8].

Como descrito nesse capítulo, as redes sem fio estão cada vez mais presentes nas diferentes aplicações por fornecer uma mobilidade aos dispositivos e por estarem melhorando suas as taxas de transmissão. Os hardwares embarcados são vistos como alternativa interessante para se aliar às redes sem fio para criar novos aplicações móveis, como centrais VoIP de pequena escala.

Devido ao ambiente integrado de redes sem fio e hardwares embarcados serem diferentes, os softwares utilizados devem ser específicos, com a finalidade de contornar alguns problemas. Dessa forma torna-se importante descrever neste estudo os softwares necessários para a criação da central VoIP de pequena escala, que serão abordados no capítulo seguinte.

4 Softwares

Devido aos sistemas VoIP serem implementados, em geral, em hardwares mais comuns como servidores Linux, a facilidade de criação de softwares é maior. Existem diversos softwares que implementam essas centrais VoIP, como o Asterisk e o FreeSWITCH.

Para este estudo a plataforma escolhida foi o Asterisk, devido algumas vantagens que ele apresenta em relação ao FreeSWITCH. Uma vantagem é a quantidade de documentação disponível, que para o Asterisk é maior devido a ele ser predecessor do FreeSWITCH. Outro ponto que pesou na escolha foi que o Asterisk se adapta melhor em centrais menores, enquanto o FreeSWITCH é melhor aplicado em centrais grandes. Por conta da escolha, o software Asterisk será discutido em maiores detalhes.

4.1.Asterisk

O Asterisk começou a ser desenvolvido em 1999, quando a empresa Linux Support Service, precisou adquirir um sistema telefônico convencional, e os preços ultrapassavam os 50 mil dólares. Assim, ela resolveu desenvolver seu próprio sistema telefônico baseado na plataforma Linux, com a finalidade de baratear seus custos. Alguns meses depois começaram a sair as primeiras versões do Asterisk. Como o foco da empresa era em trabalhos com softwares livres, o Asterisk também foi disponibilizado como um software livre. Assim, seu desenvolvimento tornou-se melhor ao longo dos anos, pois a comunidade ajudou nas suas melhorias. Em 2001 a Linux Support Service mudou de nome para Digium e até hoje continua o desenvolvimento do Asterisk com a ajuda da comunidade [9].

O Asterisk atualmente é uma das principais plataformas utilizadas para a criação de centrais VoIP, isso devido ao grande engajamento da comunidade e da estabilidade que a plataforma obteve ao longo dos anos. Ele é implementado em uma distribuição Linux e tem a capacidade de criar centrais que utilizam protocolos VoIP, por exemplo o SIP, interligando com a rede convencional PSTN [10].

Outro ponto interessante do Asterisk é a possibilidade da criação de módulos que aumentam suas funcionalidades, por exemplo o modulo de Unidade de Resposta Audível (URA), que é um atendedor automático muito utilizado em centrais de atendimento. Tal tecnologia já existia nas centrais antigas, porém eram caras para serem

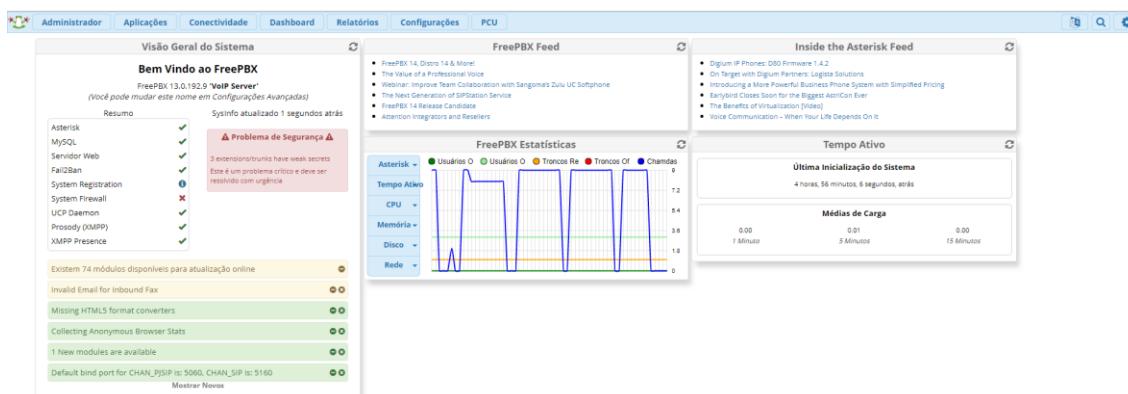
compradas. Com o Asterisk, isso é feito de forma simples e com um custo muito baixo [10].

Apesar de todas as facilidades, o Asterisk é uma plataforma em que grande parte da configuração tem que ser realizada via linhas de comando. Com a finalidade de facilitar isso, algumas empresas desenvolveram interfaces gráficas para facilitar a configuração das centrais [10].

4.1.1. FreePBX

O FreePBX é uma interface gráfica que segue o modelo de software livre, em que diversas pessoas da comunidade ajudam no desenvolvimento dela, e de módulos adicionais. Ela tem a função de facilitar a configuração de ambientes Asterisk, por meio de interfaces gráficas. Essas interfaces fazem com que as pessoas que estão configurando a central não necessitem de um vasto conhecimento de como funciona a plataforma Asterisk para realizar funções simples, como a criação de troncos e ramais. [9]

Outra vantagem da utilização de interfaces gráficas é o auxílio no monitoramento da central. Na Figura 4 é mostrada a interface do FreePBX 13 em que é possível ver as opções para configuração e gráficos de monitoramento.



FreePBX is a registered trademark of Sangoma Technologies Inc. FreePBX 13.0.192.0 is licensed under the GPL. Copyright © 2001-2017



Figura 4 – Interface do FreePBX 13.

Essas interfaces gráficas em conjunto com o software base Asterisk, criam a base de uma central VoIP, porém isso é apenas o servidor. Para se comunicar com esse servidor são necessários clientes, que podem ser de diversos tipos. Um tipo de cliente

são os telefones IP, que em geral apresentam algumas características diferentes de telefones convencionais, como a presença de uma placa de rede para se comunicar através da rede com a central VoIP. Nesse estudo, porém, será utilizado outro tipo de cliente, chamados de softphones.

4.2.Softphones

Para o recebimento e realização de chamadas multimídia em computadores foi necessária a criação de softwares com essa capacidade. Eles são chamados de softphones, e executam as mesmas funções de um telefone convencional. Porém, são executados em computadores, e alguns apresentam algumas funcionalidades extras, como a realização de chamadas de vídeo.

Para este estudo foi utilizado um softphone chamado de MicroSIP. A escolha vem do fato de ele ser gratuito e apresentar o suporte aos codecs, G711 e G729, uma vez que outros softphones apresentam essa funcionalidade apenas nas suas versões pagas. A Figura 5 apresenta a interface do MicroSIP. A interface dele é simples sendo similar à de um telefone convencional, com um teclado numérico, controles de áudio e botões para pausar e desligar a ligação.

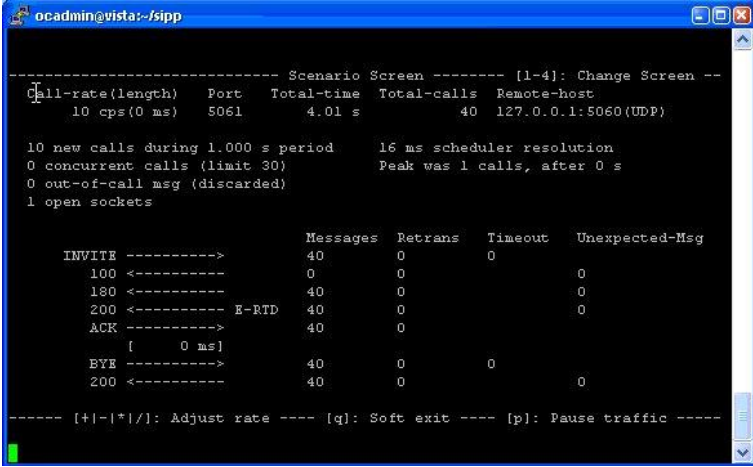


Figura 5 – Interface do MicroSIP.

4.3.SIPp

Com a finalidade de testar centrais VoIP, um software chamado SIPp foi desenvolvido pela HP para gerar cargas e verificar a capacidade delas, de forma aberta e que inicialmente as atualizações eram feitas com auxílio da comunidade. Porém, a HP parou de atualizar o software na versão 3.4, mas a comunidade desenvolveu novas versões com correção de bugs, estando atualmente na versão 3.5.1.

O SIPp é capaz de criar transmissões SIP por meio de cenários que são criados via um arquivo XML. Nesse cenário é possível criar diversos casos de teste, bem como configurar com qual codec será feita a ligação, ou o tempo da ligação. No momento da execução é possível definir quantas ligações serão feitas, bem como a taxa de chegada e o limite de ligações simultâneas [11]. Na Figura 6 está apresentada a interface do SIPp durante um teste, em que é possível verificar as informações de como estão as ligações, o número de ligações e alguns comandos para pausar o teste ou aumentar a carga de teste.



```

ocadmin@vista:~/sipp
----- Scenario Screen ----- [1-4]: Change Screen --
Call-rate(length)  Port  Total-time  Total-calls  Remote-host
10 cps(0 ms)      5061  4.01 s      40           127.0.0.1:5060(UDP)

10 new calls during 1.000 s period  16 ms scheduler resolution
0 concurrent calls (limit 30)       Peak was 1 calls, after 0 s
0 out-of-call msg (discarded)
1 open sockets

Messages  Retrans  Timeout  Unexpected-Msg
INVITE ----->  40      0        0           0
100 <-----  0        0           0
180 <-----  40      0           0
200 <----- E-RTD  40      0           0
ACK ----->  40      0
[ 0 ms]
EYE ----->  40      0        0           0
200 <-----  40      0           0

----- [+|-|*|/]: Adjust rate --- [q]: Soft exit ---- [p]: Pause traffic -----

```

Figura 6 – Execução de um cenário no SIPp.

Fonte: [11]

4.4. Wireshark

O Wireshark é um software analisador de redes. Analisadores de rede são softwares que capturam todos os pacotes que trafegam em uma placa de rede de um dispositivo. Com os pacotes capturados ele é capaz de decodificá-los caso eles estejam definidos em uma RFC⁴ (*Request for Comments*) e não estejam criptografados. Na Figura 7 é mostrada a interface do Wireshark.

⁴ RFC são documentos técnicos criados e mantidos pela IETF (*Internet Engineering Task Force*), que é uma instituição que especifica padrões que serão implementados e utilizados na internet.

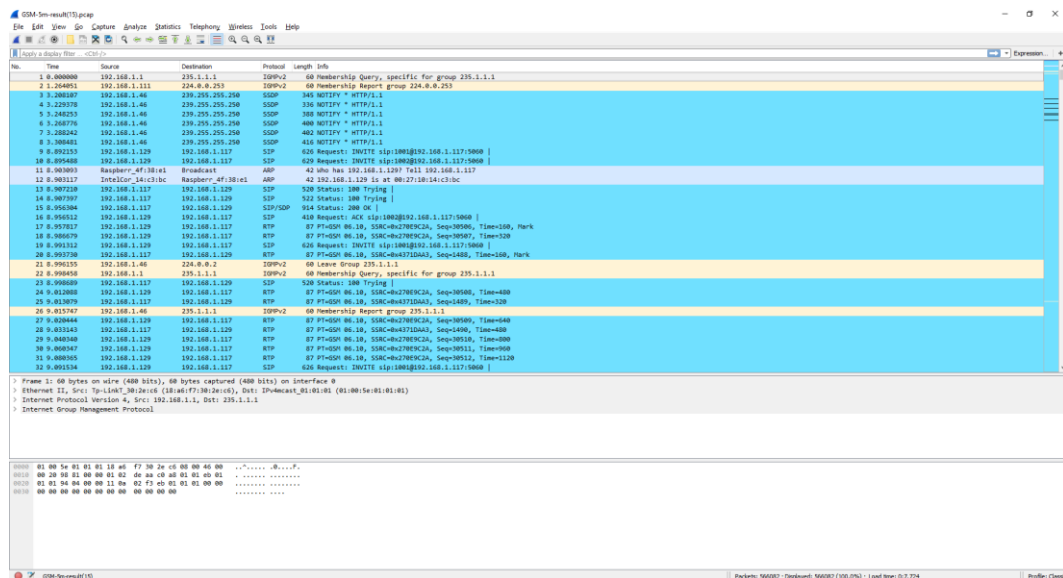


Figura 7 – Interface do Wireshark.

O Wireshark, além realizar a captura dos pacotes, também realiza a análise desses pacotes utilizando filtros e realizando cálculos baseado nas informações do cabeçalho dos pacotes para obter alguns parâmetros, como jitter.

O jitter é calculado no Wireshark seguindo a RFC3550. No cabeçalho do protocolo RTP existe um campo de 32 bits que com informações temporais do pacote. Os relógios do emissor e do receptor não são sincronizados, então é difícil obter a diferença absoluta de tempo utilizando os dados do cabeçalho do RTP, logo é necessário a utilização de formulas para a obtenção da diferença relativa. O cálculo é feito pela diferença entre a marca de relógio temporal do pacote RTP e o clock RTP do receptor no tempo da chegada. O jitter J é definido como a diferença D , que corresponde ao período de tempo entre a chegada de um par de pacotes, comparando com o *timestamp* neste par de pacotes no momento do envio [12].

Considerando um pacote i , em que o tempo estampado seja S_i e R_i seja o tempo de chegada. Para dois pacotes i e j , o D é expresso como:

$$D(i, j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i) \quad (1)$$

O jitter deve ser calculado continuamente para cada novo pacote recebido. Inicialmente utilizando a equação 1 para obter o valor de D , e em seguida a equação 2 para obter o valor do jitter.

$$J(i) = J(i - 1) + \frac{(|D(i-1,i)| - J(i-1))}{16} \quad (2)$$

Após a apresentação dos principais softwares, todo o embasamento teórico para a confecção da central VoIP a ser utilizada nesse estudo está apresentado. Passando inicialmente pelos princípios do VoIP, em seguida pelo conceito de hardwares embarcados, redes sem fio e por fim os principais softwares. No próximo capítulo serão apresentadas as especificações e implementações do estudo, bem como a metodologia de testes a ser seguida.

5 Construindo uma central VoIP baseada em Wi-Fi

O projeto visa realizar testes de desempenho de uma central VoIP, implementada em um hardware embarcado, utilizando a rede Wi-Fi para se comunicar com os ramais, que neste caso serão computadores utilizando softphones.

Esse experimento torna-se interessante em decorrência dos pontos apresentados nos capítulos anteriores, pois o Wi-Fi garante uma grande mobilidade, porém apresenta problemas de desempenho e como descrito no Capítulo 2. Uma vez que sistemas VoIP são sensíveis a variações na qualidade da conexão, então seria contra intuitivo a utilização dos dois juntos. Contudo, em caso de um desempenho aceitável, novas possibilidades de tecnologias poderiam ser criadas integrando os dois conceitos.

Na Figura 8 é apresentada a topologia que será empregada nesse estudo, em que uma central VoIP está conectada com os clientes, por meio de um roteador sem fio.

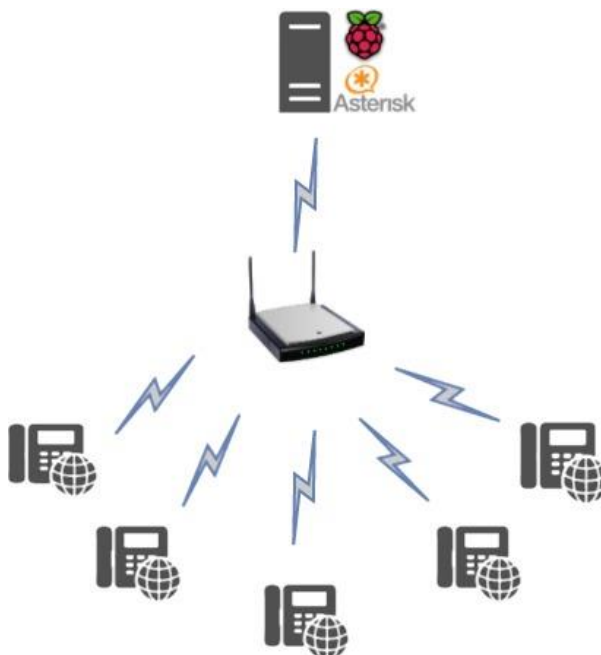


Figura 8 – Topologia do projeto.

O projeto foi implementado utilizando dois computadores funcionando como clientes, e a Raspberry sendo utilizada como central telefônica e um roteador sem fio, utilizando o protocolo 802.11g.

5.1. Especificações dos Hardwares e Softwares utilizados

5.1.1. Raspberry

A Raspberry Pi 3 modelo B apresenta as seguintes especificações:

- Processador Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB de memória
- Chipset BCM43438 para a rede wireless, que utiliza o protocolo 802.11n.

Para armazenamento, foi utilizado um cartão mini SD classe 10 de 16GB. Ela estava utilizando o sistema operacional Raspbian, que é uma adaptação do sistema operacional Debian para processadores ARM, com o Asterisk 13, juntamente com o FreePBX 13 e com os codecs G711 e G729.

5.1.2. Computador 1

Este computador tem como função servir de atendedor tanto para a ligação de teste quanto para a carga de ligações. A ligação é atendida pelo softphone, e a carga é atendida pela máquina que está funcionando como um atendedor.

Este computador possui um processador AMD Phenom II x4, com 8GB de memória, utilizando o sistema operacional Windows 10, com o MircoSIP e o VirtualBox 5.1, que é um software para a criação de máquinas virtuais. Máquinas Virtuais são softwares que emulam um computador virtual, ou seja, é possível ter praticamente todas as funções de um computador sendo executado de forma virtual, como a instalação de um novo sistema operacional e a obtenção de um IP diferente do hospedeiro.

A máquina virtual que estava funcionando como atendedor, está utilizando o processador do hospedeiro, porém com 3GB de memória, nela foi utilizada a distribuição Linux CentOS, rodando o Asterisk 13 e o FreePBX 13, apenas com o codec G711 instalado. Os outros codecs não foram instalados nessa central, pois a mesma é uma central separada que só serve como atendedor, então o áudio vindo dela tinha que passar pela central principal, que realizava a conversão para o codec correto antes enviá-lo para o cliente.

5.1.3. Computador 2

Este computador tem como função gerar as ligações dos experimentos, tanto a ligação de teste quanto a carga de ligações.

Este computador possui um processador Intel Core I5, com 4GB de memória, rodando o sistema operacional Ubuntu 15.04, com o software SIPp 3.5.1 e o software tshark para realizar a coleta dos dados e armazená-los em um arquivo pcap.

5.1.4. Roteador

O roteador utilizado foi um TP-Link TL-WR1043ND, que é um roteador Wireless de 3 antenas com suporte ao 802.11n,g,b. Nele o firmware padrão foi alterado para o dd-wrt, que é um firmware alternativo para roteadores baseados em Linux. O firmware foi alterado devido a funções adicionais que ele fornece em relação ao original, como a intensidade do sinal de cada cliente, que será um parâmetro utilizado neste estudo.

5.2.Instalação do ambiente de teste

5.2.1. Sistema Operacional e Softwares na Raspberry

A Raspberry estava utilizando a distribuição Linux Raspbian, com o Asterisk instalado e com a interface gráfica FreePBX. A instalação desse ambiente foi feita utilizando os arquivos da referência [13], que é um projeto chamado Raspbx. Este projeto disponibiliza imagens prontas e atualizadas dos softwares citados acima para a instalação em diferentes modelos de Raspberry. A instalação dessa imagem é feita em um cartão micro SD e para realizar a instalação no cartão, foi utilizado um software chamado Etcher, que é software para distribuições Linux, em que ele realiza a cópia de uma imagem para um dispositivo de armazenamento, nesse caso o cartão micro SD.

5.2.2. Codec G729 na Raspberry

Na instalação inicial a central apresentava apenas alguns codecs instalados, sendo que o codec G729 não estava instalado. Para a instalação deste codec foram utilizados os procedimentos da referência [14]. Devido a diferenças que processadores ARM apresentam foi necessária a modificação de scripts de instalação para que o codec fosse instalado com sucesso. No modelo utilizado neste projeto, Raspberry Pi 3 modelo B, foi necessário a modificação, no script configure.ac na linha 31 e 32, mudando:

Quadro 1 – Mudança no script para a instalação do codec G729.

De:
<pre>cflags="-mcpu=arm1176jzf-s -mfloat-abi=hard -mfpv=vfp -O3 -Wall" march = "armv6zk"</pre>
Para:
<pre>cflags="-mcpu=cortex-a53 -mfloat-abi=hard -mfpv=neon-vfpv4 -O3 -Wall" march = "armv8-a+crc"</pre>

Os trechos alterados, são necessários para modificar as flags que indicam qual o modelo de processador está sendo usado, essas flags são utilizadas pelo compilador gcc, para compilar os arquivos. E por fim é necessário modificar os parâmetros do comando `./configure` descrito na referência [14]. O comando modificado fica:

Quadro 2 – Comando modificado para a instalação do codec G729.

<pre>./configure CFLAGS='-march=armv8-a+crc' --with-bcg729 --with-asterisk130 -- with-asterisk-includes=../asterisk-13.17.1/include</pre>

5.2.3. Sistema Operacional e Softwares na máquina virtual

Na máquina virtual foi instalada uma imagem pronta da distribuição do FreePBX. A instalação é feita de forma automática assim que é executada a imagem no hardware em que ela será instalada, sendo necessário apenas configurar parâmetros básicos, como usuário e senha [15].

5.3. Utilização do ambiente de teste

Em um dos computadores estava executando o SIPp, para gerar a carga e uma ligação para coleta de dados. No SIPp um cenário foi adaptado [16] para transmitir um áudio no formato pcap por meio de uma ligação usando o codec G711, G729 e por fim GSM. No Anexos A, B e C estão os cenários utilizados.

Para obter o arquivo pcap referente a cada codec, foi feita uma ligação utilizando cada codec em um ambiente sem perdas, ou seja, foi utilizada uma rede com fio. A transmissão da ligação foi gravada com o software Wireshark, e após gravação

foram filtrados os pacotes da transmissão para sobrar apenas a ligação, e salvar em um arquivo no formato pcap.

Para conseguir gerar de forma separada a ligação de teste e a carga, o SIPp foi executado em dois processos simultâneos. Para a executar os dois cenários foram utilizados os seguintes comandos com os parâmetros:

Quadro 3 – Comando para a execução do primeiro cenário.

```
./sipp -I 192.168.1.129 192.168.1.117 -sn uac -s 1001 -sf sipp_uac_g711a.xml  
-m 9 -l 9
```

Os parâmetros utilizados nesse primeiro comando são:

- -i que indica o IP de origem, que neste caso foi o 192.168.1.129, e o IP da central que nesse caso foi 192.168.1.117.
- O parâmetro -sn uac indica que o software irá trabalhar como um cliente.
- O parâmetro -s indica para qual ramal as ligações serão destinadas
- -sf executa o arquivo XML do cenário.
- -m indica a taxa de chegada das ligações, neste caso 9.
- -l indica o limite de ligações simultâneas que também é 9.

Para o cenário 2 o comando executado é:

Quadro 4 – Comando para a execução do segundo cenário.

```
./sipp -I 192.168.1.129 192.168.1.117 -sn uac -s 1002 -sf  
sipp_uac_g711a2.xml -m 1 -l 1
```

Os parâmetros do segundo cenário em sua maioria são iguais aos do primeiro cenário. Os parâmetros que mudam são: -s que dessa vez será destinado para o ramal 1002; -sf que irá executar o segundo cenário XML. E como essa é a ligação que será monitorada, a taxa de chegada e o limite de ligações é apenas 1.

Por último o outro computador estava executando o MicroSIP e uma máquina virtual com outro servidor Asterisk. A ligação para coletar os dados estava sendo direcionado para o ramal 1002 que estava configurado no softphone para receber a ligação, enquanto a máquina virtual com o segundo servidor Asterisk estava servindo como um atendedor. Ou seja, as ligações para o ramal 1001 eram direcionadas para este servidor, que as colocava na URA (Unidade de Resposta Audível). No caso de centrais

VoIP, URA é um módulo que atende uma ligação automaticamente e em seguida toca uma gravação.

As duas centrais foram interligadas através de um tronco utilizando o protocolo de comunicação IAX (Inter Asterisk eXchange), que é um protocolo desenvolvido pela Digium como uma alternativa para o SIP. Assim, todas as ligações que chegam no ramal 1001 são direcionadas para o tronco de interligação. Essa topologia foi baseada na referência [17]. Na Figura 9 é apresentada a topologia utilizada nesse estudo e nela é possível verificar como está a configuração de cada servidor e cliente.

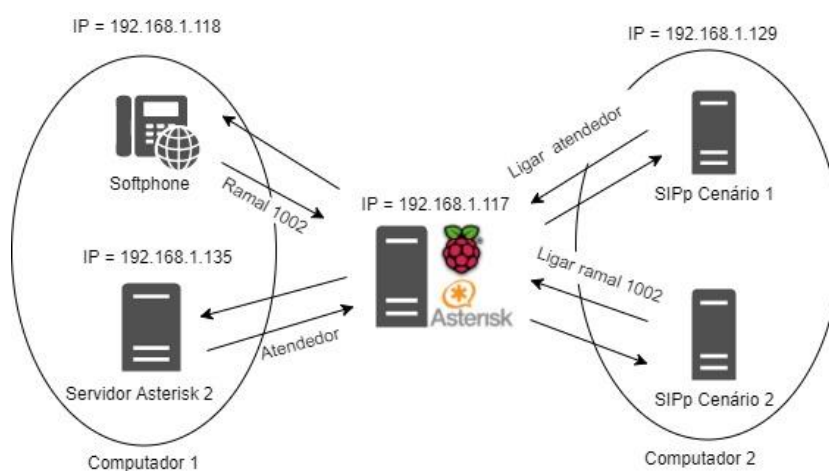


Figura 9 – Topologia implementada.

5.4. Metodologia

Antes de começar os experimentos utilizando as redes sem fio, foram feitos alguns experimentos utilizando uma conexão cabeada com a finalidade de se obter dados de referência.

O primeiro experimento foi a capacidade de ligações que a central suportaria utilizando uma conexão cabeada e o codec G711, ou seja, a partir de quantas ligações ela começaria a apresentar perdas. O segundo experimento foi o mesmo, porém utilizando o codec G729. E o terceiro experimento foi o mesmo, utilizando o codec GSM.

O segundo experimento foi feito utilizando a rede sem fio. Este foi feito utilizando 3 distâncias diferentes em um ambiente indoor com os três codecs citados anteriormente. As distâncias escolhidas foram:

- 0m: A central estava no mesmo ambiente do roteador a pouco centímetros de distância. Para essa distância foram utilizados os codecs G711, G729 e GSM. A Figura 10 mostra o ambiente para esse caso de teste.

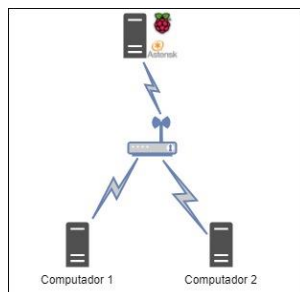


Figura 10 – Ambiente onde a central estava próxima do roteador.

- 5m: A central estava em um ambiente externo ao roteador, e com obstruções entre eles, como uma parede de alvenaria. E também foram utilizados os codecs G711, G729 e GSM. A Figura 11 mostra o ambiente de teste para este caso.

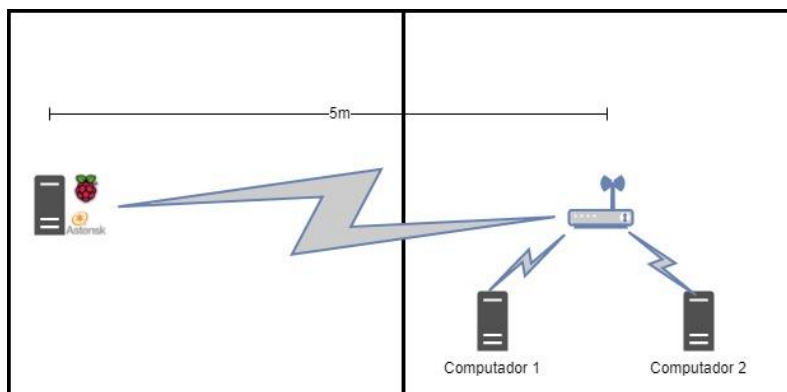


Figura 11 – Ambiente onde a central estava a 5m do roteador.

- 10m: A central estava um ambiente distante do roteador com bastante obstruções, como 4 paredes de alvenaria. E também foram utilizados os codecs G711, G729 e GSM. A Figura 12 mostra o ambiente para esse caso de teste.

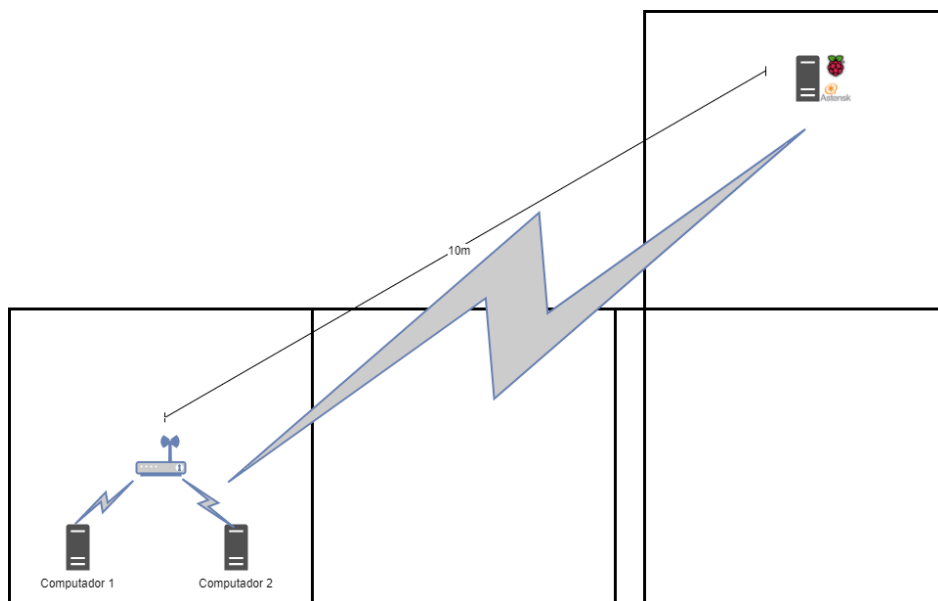


Figura 12 – Ambiente onde a central estava a 10 m do roteador.

O ambiente de teste utilizado para as três distâncias foi o mesmo, em que não havia nenhuma outra conexão no roteador, além das conexões utilizadas para realizar os experimentos. Quaisquer fontes de interferência, quando possível foram retiradas com a finalidade de não interferir na qualidade dos resultados.

A carga de ligações utilizada em todos os casos foi de 9 ligações e uma ligação de controle que estava sendo monitorada para coletar dos dados. O período de duração das ligações foi de 11 minutos para a carga de 9 ligações, e de 10 minutos para a ligação monitorada. Em todos os casos o áudio utilizado foi uma música chamada Epic [18]. Esse número de ligações foi escolhido baseado em testes realizados, em que após essa quantidade de ligações em determinadas distâncias não era possível realizar experimentos, devido a ligação ser encerrada em um período curto de tempo em decorrência da grande perda de pacotes ou perda total do sinal da rede sem fio.

Para cada caso os experimentos foram replicados 15 vezes, com a finalidade de gerar maior aleatoriedade e fazendo com que a média seja estatisticamente relevante.

5.4.1. Métricas

As métricas obtidas foram diferentes para o teste com a conexão cabeada e para a conexão sem fio.

Na conexão cabeada foi medido o uso de CPU, memória e perda de pacotes. Tais métricas foram escolhidas para verificar se as perdas seriam em decorrência da conexão ou da capacidade máxima de processamento do servidor ser atingida.

Na conexão sem fio foram medidos a perda de pacotes, o Jitter e a latência, que conforme descrito, são fundamentais para o funcionamento de sistemas VoIP. Também foi gravado na central os áudios das ligações para que futuramente possa ser comparada a qualidade dos áudios. O uso de CPU neste caso não foi medido, pois pela quantidade de ligações ser baixa esse dado não traria nenhuma informação de relevância.

Os valores obtidos de perda de pacotes, Jitter e latência são importantes para este estudo, pois com eles é possível estimar um valor de MOS para cada condição e com isso verificar qual seria a qualidade de cada ligação.

5.4.1.1. Cálculo de MOS

Apesar do MOS ser um parâmetro subjetivo, existem alguns modelos que por meio de equações, e utilizando parâmetros de redes, conseguem fornecer um valor de MOS, próximo dos obtidos em teste subjetivos. Um desses modelos é o E-Model, que leva em consideração para a obtenção do MOS todos os parâmetros de redes obtidos neste estudo. As fórmulas para a obtenção do MOS serão apresentadas a seguir. [19] [20]

Inicialmente é calculada a latência efetiva, levando em consideração a latência média e jitter em ms, através da Equação 1.

$$Latência\ Efetiva = (Latência\ média + Jitter * 2 + 10) \quad (3)$$

Se a latência efetiva for menor que 160ms:

$$R = 93,2 - \left(\frac{Latência\ Efetiva}{40} \right) \quad (4)$$

Caso contrário:

$$R' = 93,2 - \left(\frac{Latência\ Efetiva - 120}{10} \right) \quad (5)$$

Após o cálculo R' utilizando a latência efetiva obtida anteriormente, é necessário levar em consideração a perda de pacotes em porcentagem, assim obtendo um novo valor de R , segundo a Equação 3.

$$R = R' - (\text{Perda de Pacotes} * 2,5) \quad (6)$$

E por fim com o valor de R calculado, é possível obter o valor final do MOS por meio da Equação 5.

$$MOS = 1 + 0,035 * R + 0,000007 * R * (R - 60) * (100 - R) \quad (7)$$

Utilizando a metodologia descrita anteriormente e obtendo as métricas descritas, foram obtidos os resultados dos experimentos. No capítulo seguinte os resultados serão apresentados e analisados.

6 Resultados

6.1. Teste da capacidade da Raspberry utilizando uma rede cabeada.

Ao longo dos experimentos a central foi submetida a uma carga de ligações, inicialmente com 10 ligações, depois com 25, 50, 75, 90 e 100 utilizando os codecs G711, G729 e GSM. Em todos os experimentos não houve perdas de pacotes e o limitante do número de ligações ficou na capacidade de processamento da Raspberry. Nas Tabelas 3, 4 e 5 são apresentados os dados da utilização de CPU e o número de chamadas bloqueadas. Para melhor visualização, na Figura 13 é apresentado um gráfico que sintetiza o comportamento do que ocorreu.

Tabela 3 – Tabela do número de ligações, uso da CPU e ligações bloqueadas para o protocolo G711.

Número de Ligações	Uso da CPU(%)	Chamadas Bloqueadas
10	11,80 ± 4,03	0
25	24,19 ± 2,11	0
50	35,64 ± 8,31	0
75	51,61 ± 13,91	0
90	67,53 ± 11,43	0
100	74,21 ± 10,47	2

Tabela 4 – Tabela do número de ligações, uso da CPU e ligações bloqueadas para o protocolo G729.

Número de Ligações	Uso da CPU(%)	Chamadas Bloqueadas
10	20,65 ± 4,27	0
25	32,56 ± 4,00	0
50	68,92 ± 6,35	0
75	96,08 ± 5,98	0
90	97,31 ± 3,75	6
100	97,75 ± 2,55	9

Tabela 5 – Tabela do número de ligações, uso da CPU e ligações bloqueadas para o protocolo GSM.

Número de Ligações	Uso da CPU(%)	Chamadas Bloqueadas
10	19,83 ± 3,93	0
25	31,93 ± 3,40	0
50	37,59 ± 14,74	0
75	65,71 ± 2,99	0
90	82,37 ± 8,34	0
100	97,69 ± 3,46	0

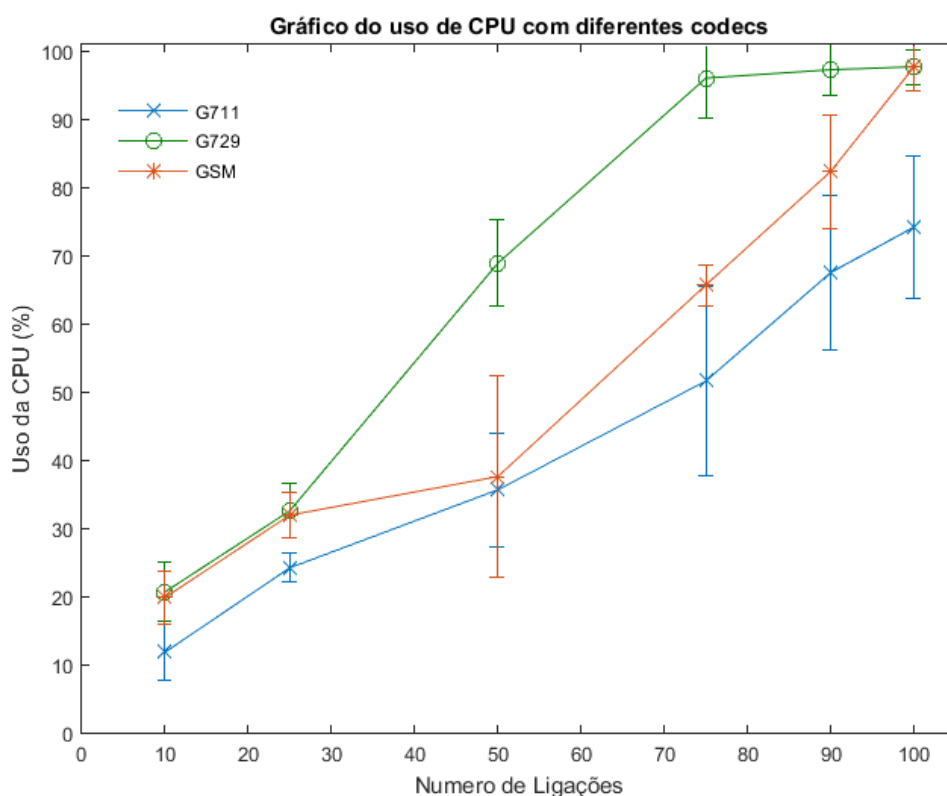


Figura 13 – Uso de CPU.

O codec G711 apresenta o melhor desempenho do uso da CPU. Tal fato ocorre devido ele não apresentar compressão, o que faz com que a central não utilize processamento para realizar a compressão dos dados. Já nos casos do G729 e GSM, é realizada uma compressão do áudio para que ele fique com taxa de bits de 8kb/s. Porém é possível verificar que o desempenho do GSM é um pouco superior ao do G729, atingindo o ápice de processamento apenas com 100 ligações.

Para os codecs acima, em todos os casos, até atingir o número de 90 ligações, nenhuma ligação foi bloqueada, como pode ser verificado nas tabelas. Para 90 ligações apenas o codec G729 teve ligações bloqueadas (6 ligações). Isso ocorreu devido à capacidade máxima da CPU ser atingida, então não foi possível estabelecer a conexão. É possível verificar pela Figura 12 que em torno de 70 ligações ele já começa a se aproximar da capacidade máxima de processamento. Quando havia 100 ligações, tanto o G711 quanto o G729 tiveram ligações bloqueadas (2 e 9, respectivamente). O codec GSM não teve nenhuma ligação recusada. Para o codec G729 já era de se esperar que o número de ligações bloqueadas aumentasse. Quanto ao G711, não era de se esperar que alguma ligação fosse recusada. Isso ocorreu devido ao início da ligação utilizar mais processamento para estabelecer a conexão, e nesse período a central ter atingido a sua capacidade máxima, bloqueando duas ligações. O GSM, apesar de utilizar compressão, não teve ligações bloqueadas, porém com 100 ligações é possível notar que o uso de CPU estava bem próximo de 100%. Então, caso o número de ligações fosse maior, provavelmente ele iria começar a ter ligações bloqueadas.

O codec G711 é o codec que apresentou a melhor eficiência no uso da CPU. Seria possível ter um número maior de ligações simultâneas utilizando o G711, caso houvesse um maior intervalo de tempo na chegada das ligações, pois daria tempo para a central estabelecer novas conexões. Já tanto para o G729 como para o GSM não é possível ter um número maior de ligações na central sem que haja bloqueio de ligações.

6.2. Potência do Sinal

Como descrito anteriormente os experimentos foram realizados utilizando 3 distâncias diferentes, sendo que em dois casos existiam obstáculos entre a central e o roteador sem fio. Isso fez com que a potência do sinal recebido pela central variasse. Na Tabela 6 são apresentados os valores da porcentagem de potência do sinal para as três diferentes distâncias.

Tabela 6 – Tabela da intensidade de sinal recebido pela Raspberry.

Distância	Potência (%)
Próximo	89,53 ± 1,56
5m com obstáculo	60,40 ± 2,56
10m com obstáculo	31,00 ± 3,80

Para as diferentes distâncias no cenário analisado, pode-se concluir que a potência do sinal está caindo cerca de 30% a cada 5m no ambiente indoor. Com isso é possível prever que para distâncias maiores que 10m as condições do experimento não tornariam possível realizar nenhuma transmissão de dados de forma satisfatória, pois a potência do sinal seria muito baixa, fazendo com que houvesse uma grande perda de pacotes, bem como um aumento significativo da latência e do jitter. Isso deixaria inviável o uso dessa conexão tanto para uso simples como o acesso a sites, quanto para usos mais complexos, como uma transmissão VoIP.

6.3.Perda de Pacotes

Nas Tabela 7, 8 e 9 são apresentados os dados de porcentagem de perda de pacotes para cada experimento. Na Figura 14 é apresentada a perda de pacotes média, referentes às diferentes distâncias e aos três codecs. A perda de pacotes foi medida levando em consideração que uma transmissão completa apresenta um total de 30000 pacotes para o período de 10 minutos. Foi medido quantos pacotes efetivamente chegaram ao cliente, e calculada a porcentagem de pacotes que foram perdidos ao longo da transmissão.

Tabela 7 – Porcentagem de perda de pacotes ao longo de cada experimento para os codecs G711, G729 e GSM na distância de 0m.

Experimento	G711	G729	GSM
1	0,63	28,08	20,32
2	0,11	12,44	23,52
3	0,39	10,14	21,23
4	1,28	7,77	23,31
5	0,26	9,98	21,27
6	0,22	6,72	14,21
7	0,02	8,87	15,19
8	0,30	14,83	12,61
9	0,41	7,78	11,04
10	0,21	9,00	14,84
11	0,40	16,00	30,70
12	0,58	12,20	25,14
13	0,18	11,02	26,93
14	0,23	8,65	33,63
15	2,24	10,33	17,47

Tabela 8 – Porcentagem de perda de pacotes ao longo de cada experimento para os codecs G711, G729 e GSM na distância de 5m.

Experimento	G711	G729	GSM
1	1,73	15,69	6,78
2	0,68	20,92	26,71
3	0,97	15,96	13,88
4	1,79	12,32	18,95
5	0,87	31,01	7,73
6	0,96	49,52	15,61
7	0,85	48,82	9,53
8	1,78	35,23	14,55
9	1,08	23,87	17,31
10	1,76	9,06	24,16
11	37,13	36,62	19,06
12	3,46	15,91	13,73
13	0,81	24,58	15,01
14	1,18	12,84	21,22
15	16,43	9,85	13,74

Tabela 9 – Porcentagem de perda de pacotes ao longo de cada experimento para os codecs G711, G729 e GSM na distância de 10m.

Experimento	G711	G729	GSM
1	5,93	31,90	29,25
2	8,32	55,52	40,22
3	17,95	96,76	48,91
4	8,36	39,65	54,49
5	0,27	49,70	65,90
6	5,07	45,56	61,57
7	9,40	16,92	43,65
8	2,51	13,06	42,65
9	19,00	31,15	61,2
10	12,41	56,60	57,17
11	3,46	32,56	52,47
12	0,80	76,78	24,88
13	1,59	44,60	10,92
14	3,83	27,57	51,09
15	5,30	46,04	46,51

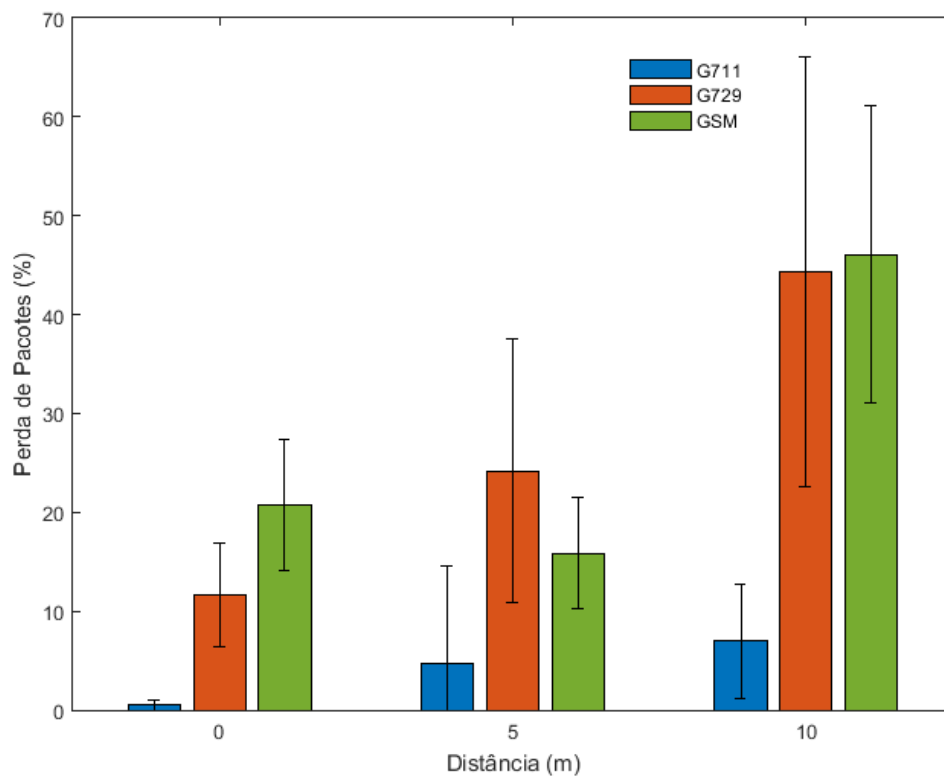


Figura 14 – Média de Perda de Pacotes.

O codec G711 obteve em todos casos o melhor desempenho neste quesito de perda de pacotes, e isso pode ter ocorrido devido ao codec G711 ter um maior payload

em relação aos outros dois codecs, que apresentam um payload bem menor. Logo, eventos de perda de pacote são mais raros.

No codec G729 houve um aumento significativo de perda de pacotes para cada distância. Ele aproximadamente dobrou a perda de pacotes quando a distância era dobrada. Em alguns casos, é possível notar que pela barra de erro ele variou muito, como no caso de 10m, em que as perdas podem ter chegado próximo dos 70%.

No codec GSM ocorreu que a perda de pacotes próximo à central foi maior que a perda de pacotes no caso 5m. Porém, se for considerada a barra de erro de ambos os casos, é possível afirmar que elas foram quase a mesma. Já no caso 10m, a perda cresceu de forma significativa, atingindo uma porcentagem muito grande de pacotes.

Para todos os codecs, conforme se aumentou a distância, as barras de erros também cresceram bastante. Isso ocorreu devido à natureza das redes sem fio, em que pequenos fatores externos alteram muito o valor dos dados, deixando com que os mesmos fiquem bastante aleatórios.

Em alguns experimentos houve uma grande variação de perda de pacotes, em que em um evento específico o valor de perda de pacotes ficou muito acima da média, como no caso da Tabela 8 no experimento 11 do codec G711. Essas variações ocorrem devido a natureza do Wi-Fi, em que diversos fatores podem causar uma diminuição momentânea na qualidade do sinal.

6.4.Jitter

Nas Tabela 10,11 e 12 são apresentados os dados de jitter que ocorreram entre a central e o cliente. O jitter foi calculado conforme descrito no Capítulo 4, pelo programa Wirshark. Na Figura 15 é mostrado o jitter médio para todos os casos. Em cada experimento o programa utilizou a média de jitter ao longo da transmissão.

Tabela 10 – Jitter da central para o cliente em ms ao longo de cada experimento para os codecs G711, G729 e GSM na distância de 0m.

Experimento	G711	G729	GSM
1	2,58	4,93	3,46
2	1,86	3,68	3,63
3	1,65	3,60	3,71
4	2,13	2,82	4,13
5	2,29	3,23	5,55
6	13,10	2,80	3,19
7	2,28	3,52	3,96
8	3,71	2,52	3,24
9	2,63	3,34	3,62
10	2,19	3,32	4,15
11	2,42	3,17	3,52
12	2,67	2,23	4,36
13	2,26	3,08	4,15
14	2,47	3,42	5,82
15	2,51	3,60	4,05

Tabela 11 – Jitter da central para o cliente em ms ao longo de cada experimento para os codecs G711, G729 e GSM na distância de 5m.

Experimento	G711	G729	GSM
1	2,62	3,35	4,07
2	3,14	2,99	5,68
3	3,61	3,58	5,16
4	3,14	3,82	6,80
5	4,15	4,53	3,94
6	3,91	4,12	4,82
7	2,49	5,34	5,47
8	2,89	4,17	5,13
9	3,03	4,17	5,33
10	3,61	3,89	3,63
11	18,66	4,69	4,22
12	5,99	4,24	4,15
13	4,56	4,77	4,66
14	3,63	4,71	6,32
15	16,77	3,79	5,48

Tabela 12 – Jitter da central para o cliente em ms ao longo de cada experimento para os codecs G711, G729 e GSM na distância de 10m.

Experimento	G711	G729	GSM
1	6,15	7,59	4,76
2	9,99	5,14	5,39
3	14,75	8,60	5,65
4	7,44	3,80	7,26
5	3,54	8,50	9,58
6	7,68	6,66	8,16
7	8,90	6,05	6,28
8	5,72	4,69	7,45
9	13,04	5,18	6,06
10	13,54	4,53	6,57
11	6,34	6,20	7,70
12	3,37	4,84	3,59
13	3,93	3,90	3,08
14	5,99	6,56	6,17
15	6,48	9,20	6,98

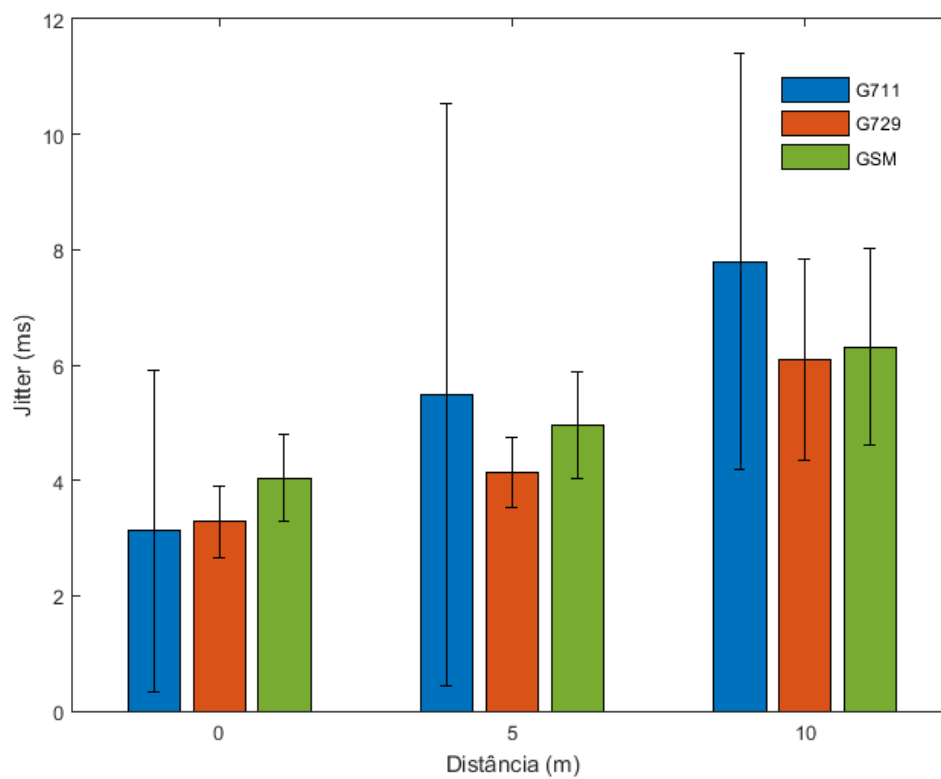


Figura 15 – Jitter Médio.

O codec G711 para o caso do jitter não sofreu tanta variação quanto na perda de pacotes. Ele aumentou aproximadamente 2ms a cada 5m. Porém, como ele foi o codec que teve menos perdas de pacotes, seu jitter pode ter variado muito, pois ele recebeu a maioria dos pacotes, e seus pacotes são maiores do que os outros dois codecs, assim fazendo com que a diferença de tempo de chegada de cada pacote seja maior do que no caso dos outros dois codecs.

No caso do codec G729 a variação de jitter foi pequena. Se for considerada a barra de erro, é possível notar que o desempenho de jitter ficou bem próximo em todos os casos, tendo uma variação aproximada de pouco mais de 1ms a cada 5m.

No codec GSM, similarmente ao codec G729, a variação de jitter foi baixa, principalmente para as duas primeiras distâncias, em que foi de aproximadamente 1ms. Já para a distância de 10m a variação foi maior de, aproximadamente, 2ms.

Como descrito acima, o desempenho do codec G711 foi o pior dos três codecs na análise de jitter. Porém, este resultado já era esperado devido ao tamanho de cada pacote, que é cerca de 8 vezes maior que os outros dois codecs.

6.5.Latência

Nas Tabelas 13, 14 e 15 são apresentados os dados de latência da central para o cliente, para cada experimento. Esse parâmetro foi obtido por meio de cálculos feitos pelo software Wireshark, em que se obtém uma amostra de latência a cada pacote recebido e posteriormente realiza-se a média. Na Figura 16 é apresentada a latência média da central para o cliente.

Tabela 13 – Latência da central para o cliente em ms ao longo de cada experimento para os codecs G711, G729 e GSM na distância de 0m.

Experimento	G711	G729	GSM
1	20,12	27,80	24,47
2	20,02	22,84	26,15
3	20,07	22,25	25,38
4	20,25	22,12	26,07
5	20,05	22,21	27,15
6	20,04	21,44	23,31
7	20,00	21,94	23,58
8	20,05	23,48	22,88
9	20,08	21,68	21,59
10	20,04	21,97	23,48
11	20,07	23,81	28,85
12	20,11	23,18	26,72
13	20,03	22,47	27,36
14	20,04	21,89	30,13
15	20,45	22,30	24,23

Tabela 14 – Latência da central para o cliente em ms ao longo de cada experimento para os codecs G711, G729 e GSM na distância de 5m.

Experimento	G711	G729	GSM
1	20,10	22,47	21,45
2	20,13	25,29	27,28
3	20,19	23,79	23,22
4	20,36	22,84	24,67
5	20,17	28,98	21,67
6	20,19	39,61	23,69
7	20,17	40,61	22,10
8	20,36	30,87	23,40
9	20,21	26,26	24,18
10	20,54	21,99	26,37
11	31,95	31,58	24,70
12	20,71	23,78	23,18
13	20,16	26,44	23,53
14	20,23	22,94	25,38
15	23,92	22,18	23,18

Tabela 15 – Latência da central para o cliente em ms ao longo de cada experimento para os codecs G711, G729 e GSM na distância de 10m.

Experimento	G711	G729	GSM
1	21,25	26,98	28,62
2	21,81	44,96	33,49
3	24,37	65,38	39,43
4	21,82	33,13	44,06
5	20,05	39,75	58,64
6	21,07	36,73	52,08
7	22,07	24,21	35,48
8	20,77	23,00	34,88
9	24,77	29,04	52,24
10	22,83	40,65	47,59
11	20,71	29,72	42,20
12	20,16	44,99	26,62
13	20,32	36,09	22,45
14	20,79	27,61	40,89
15	21,11	33,43	37,38

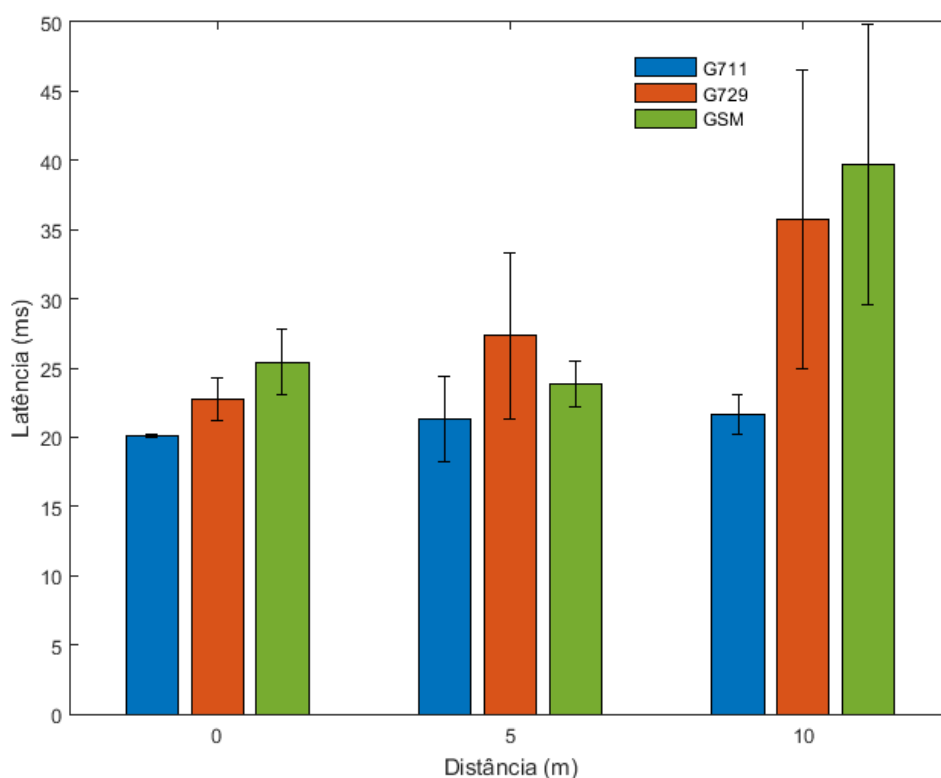


Figura 16 – Média de Latência.

O codec G711 obteve o melhor desempenho de latência, entre todos os codecs. Ele apresentou pouquíssima variação para todos a distâncias, ficando com

uma latência quase constante se consideradas as barras de erro. Isso também deve-se à baixa perda de pacotes, em que a taxa de chegada ficou constante.

A latência do codec G729 cresceu razoavelmente quando a distância chegou a 10m, fazendo com que a latência média chegasse próximo aos 40ms e tivesse uma grande variação, como é possível notar pela barra de erro. Comparando os dois primeiros casos, o crescimento foi pequeno, sendo que ambos estão dentro da barra de erro do outro, mostrando que em alguns casos o valor deles foi o mesmo.

A latência do codec GSM, como no caso da perda de pacotes em 5m, teve um desempenho melhor. Porém, a variação foi pequena levando-se em consideração a barra de erro. Pode-se considerar que a latência foi praticamente a mesma. Já no caso de 10m, a latência cresceu muito e variou bastante, como é possível notar pela barra de erro.

6.6.MOS

Na Figura 17 é apresentado o MOS, que foi calculado teoricamente como descrito no Capítulo 5. As barras de erros foram obtidas por meio da utilização da fórmula padrão de propagação de erro [21].

$$\sigma_{\bar{R}} = \sqrt{\left(\frac{\partial R}{\partial a_1}\right)^2 (\sigma_{\bar{a}_1})^2 + \left(\frac{\partial R}{\partial a_2}\right)^2 (\sigma_{\bar{a}_2})^2 + \dots + \left(\frac{\partial R}{\partial a_n}\right)^2 (\sigma_{\bar{a}_n})^2} \quad (8)$$

Esta fórmula leva em consideração a variância de cada variável, gerando um valor de erro mais próximo do real.

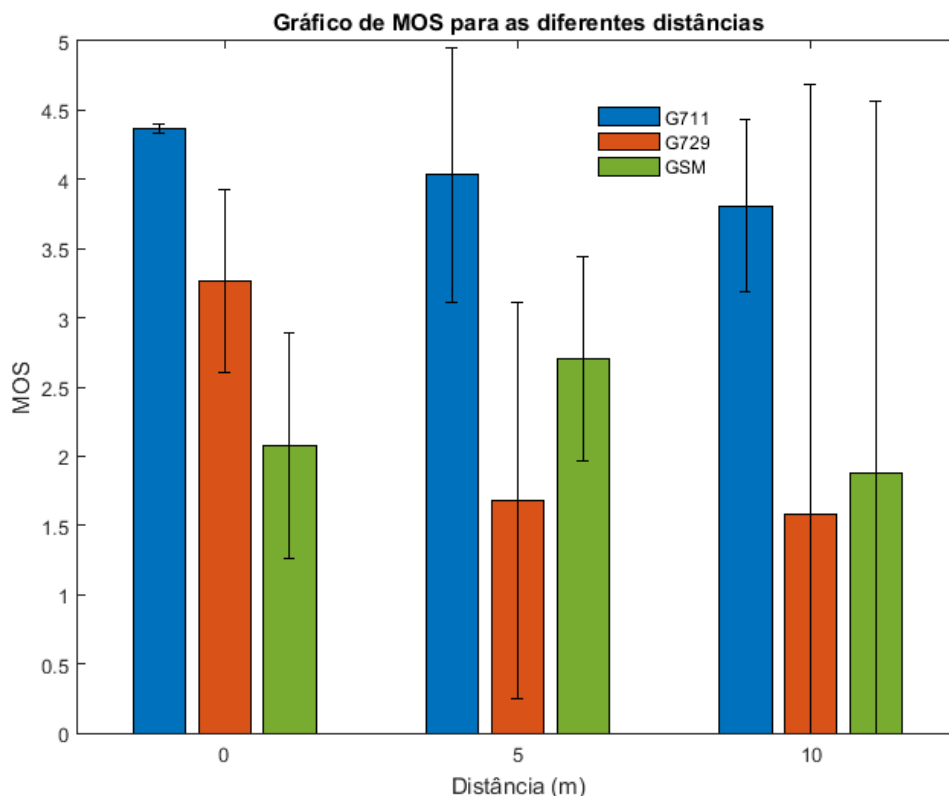


Figura 17 –MOS médio.

Para o codec G711, os valores de MOS em todos os casos foram muito bons, ficando acima de 4 para as duas primeiras distâncias, e na distância maior ficou próximo de 4. Comparando com o valor da Tabela 2, em que o valor de MOS foi 4,1, obtido através da metodologia descrita no capítulo 2, que é feita em um cenário ideal. Então o valor obtido para a rede sem fio foi muito bom, mostrando que a aproximação do modelo de cálculo utilizado no estudo gera um valor próximo ao subjetivo. A variação no codec G711 não foi tão grande, exceto no caso de 5m. Esses resultados mostram que este codec apresentou um excelente desempenho em todos os casos, mostrando que todas as conversas apresentariam um áudio muito bom.

O codec G729 para o primeiro caso apresentou um desempenho razoável, ficando com o MOS acima de 3, apresentado uma variação razoável, já quando a distância foi aumentada o MOS ficou menor que 2, que é um valor considerado muito ruim. No caso de 5m, a variância ficou grande, mas ainda aceitável. Já no caso de 10m a variância ficou muito grande, sendo muito maior que o próprio valor do resultado, mostrando que nessas condições esse codec não funciona, gerando resultados não confiáveis. Comparando esses

resultados obtidos com o valor da Tabela 2, 3,7, que é no caso ideal, o desempenho foi muito inferior, mostrando que a qualidade da ligação desse codec com a rede sem fio não é satisfatória.

O codec GSM, obteve um resultado próximo para os dois primeiros casos considerando a variância de ambos. O desempenho é considerado ruim, ou seja, a qualidade da ligação seria ruim podendo atrapalhar consideravelmente uma conversação. Já no caso de 10m o desempenho ficou um pouco pior que os anteriores, porém a variância ficou muito grande, sendo muito maior que o próprio valor. A grande variação mostra que o desempenho para essa condição seria muito ruim, apresentando muitos erros.

Para os codecs testados o G711 foi o codec que apresentou o melhor desempenho, tendo um desempenho muito bom para todos os casos. O G729 só obteve um desempenho aceitável para o caso próximo, enquanto para os outros dois casos a conversação ficaria impraticável. E o GSM obteve um desempenho ruim em todos casos, ficando com um MOS menor que 3 em todos os casos.

Com as análises dos resultados obtidos, é possível assim avaliar como foi o desempenho da central VoIP utilizando as redes sem fio. Essa conclusão será feita no próximo capítulo.

7 Conclusão

A metodologia utilizada nesse estudo, levou em consideração diversos casos de teste, como a utilização de três codecs (G711, G729 e GSM) em três distâncias diferentes (próximo, a 5m e a 10m). Com isso, foi possível obter uma variedade de resultados de desempenho da central VoIP utilizando as redes sem fio. Baseado nessa variedade de resultados é possível avaliar para em quais condições essa central poderia operar utilizando uma rede sem fio.

As primeiras análises se deram na medição da quantidade de ligações que essa central VoIP suportaria utilizando uma rede cabeada. Esta análise foi feita como forma de comparação do número de ligações que a central suportaria utilizando a rede cabeada e sem fio. Para o codec G711 o número de ligações suportadas pode superar 100 ligações, enquanto com os outros dois codecs não é possível ultrapassar esse número. Quando comparados os números de ligações utilizando a rede sem fio, o número de ligações utilizado foi menor, de 10 ligações. Esse número de ligações menor ocorreu, pois em casos em que a distância aumentou, quanto maior foi o número de ligações mais perdas apresentavam, tendo casos em que não era nem possível estabelecer a conexão. Essas perdas se justificam pela análise seguinte, que mostra uma queda grande do sinal conforme aumentou-se a distância, e quanto menor a intensidade, menor fica a taxa de transferência. Com isso, pode-se concluir que se forem utilizadas redes sem fio, o número de ligações simultâneas suportadas irá cair bastante, pois o limitante não mais será a capacidade de processamento da central, e sim a qualidade da rede.

As análises seguintes são referentes às perdas que ocorreram nos parâmetros citados no estudo. Comparando todos os casos, o codec G711 foi o que apresentou o melhor desempenho. Ele apresentou menos perdas em praticamente todos os parâmetros, só ficando atrás no quesito de jitter, que como foi explicado já era de se esperar.

O codec G729, que foi um codec desenvolvido para funcionar sobre circunstâncias em que a qualidade da conexão não é boa, apresentou o segundo melhor desempenho. Porém, não foi um desempenho satisfatório quando as distâncias aumentaram, existiram picos em que praticamente todos os pacotes foram perdidos e o jitter chegou a valores grandes.

O codec GSM apresentou o pior desempenho dos três codecs. Era de se esperar que o desempenho deste codec fosse um pouco inferior que o dos outros dois, em que pode ser constatado na literatura que seu desempenho é menor. Porém, o desempenho acabou sendo ruim em todos os casos, e para praticamente todos os parâmetros.

A última análise se deu com o cálculo do MOS teórico, para o três codecs, em que o codec G711 apresentou o melhor valor, tendo um resultado bom até para a maior distância. Já o codec G729 apresentou um resultado bom para perto, e desastroso para distâncias maiores. E o codec GSM para as duas primeiras distâncias apresentou um resultado ruim, porém seria possível ter uma conversação. Já para a maior distância ficou muito ruim, o que impediria qualquer conversação.

Com todas as considerações feitas, pode-se concluir que para os experimentos realizados o único codec que seria possível utilizar nessas condições seria o G711, que apresentou um resultado bom para todas as condições.

A central VoIP do estudo utilizando rede sem fio com o codec G711, pelos resultados, poderia ser utilizada em aplicações em que o número de ligações simultâneas não fosse grande, por exemplo, em empresas pequenas. Mas como abordado, o Wi-Fi apresenta grandes variações na qualidade, tanto que em muitos resultados os erros foram grandes, então a aplicação que optasse pela utilização dessa central não poderia ser sensível a quedas repentinas na qualidade das ligações.

Existem alguns artigos que realizam experimentos relacionados a este estudo, como [22] que apresenta um estudo teórico, porém mais estudos práticos, como o [23] [24], na área seriam necessários para criar uma base consistente de conhecimento das aplicações, e do desempenho de centrais VoIP em redes sem fio. Algumas modificações futuras que poderiam ser feitas desse estudo, seria a utilização de um hardware que apresentasse uma placa de rede sem fio melhor que a Raspberry e a utilização de roteadores com uma potência de sinal maior. Além disso, pode-se testar a utilização de outros codecs existentes. Outro ponto que poderia ser alterado, seria o ambiente de teste, em que poderia ser utilizado um ambiente sem obstáculos, o que faria com que a qualidade do sinal melhorasse, podendo assim melhorar o desempenho de todos os codecs.

8 Referência Bibliográficas

- [1] DAVIDSON, Jonathan; PETERS, James. **Voice over IP Fundamentals**. Indianapolis: Cisco Press, 2000.
- [2] HALLOCK, Joe. A Brief History of VoIP. 2004. Disponível em: <http://www.joehallock.com/edu/pdfs/Hallock_J_VoIP_Past.pdf>. Acesso em: 14 nov. 2017.
- [3] GOODE, B.. Voice over Internet protocol (VoIP). **Proceedings Of The Ieee**, [s.l.], v. 90, n. 9, p.1495-1517, set. 2002. Institute of Electrical and Electronics Engineers (IEEE).
- [4] INTERNARIONAL TELECOMMUNICATION UNION - TELECOMMUNICATION STANDARDIZATION SECTOR. **P.800: Methods of subjective determination of transmission quality**. Geneva: ITU-T, 1996. Disponível em: <<https://www.itu.int/rec/T-REC-P.800-199608-I/en>>. Acesso em: 6 nov. 2017.
- [5] RASPBERRY PI FOUNDATION. **Raspberry Pi 3 model B**. Londres, 2016. Disponível em: <<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>>. Acesso em: 2 abr. 2017.
- [6] RASPBERRY PI FOUNDATION, 2016. Disponível em: <<https://www.raspberrypi.org/app/uploads/2017/05/Raspberry-Pi-3-1-1619x1080.jpg>>. Acesso em: 20 abr. 2017
- [7] GAST, Matthew. **802.11® Wireless Networks: The Definitive Guide**. Sebastopol: O'reilly, 2002.
- [8] TANENBAUM, Andrew S.; WETHERALL, Davido. **Redes de Computadores**. 5. ed. São Paulo: Pearson Educational do Brasil, 2011.
- [9] NEWTON, Rusty. **A Brief History of the Asterisk Project**. Disponível em: <<https://wiki.asterisk.org/wiki/display/AST/A+Brief+History+of+the+Asterisk+Project>>. Acesso em: 10 ago. 2017.
- [10] BRYANT, Russell; MADSEN, Leif; VAN MEGGELEN, Jim. **Asterisk™: The Definitive Guide**. 4. ed. Sebastopol: O'reilly Media, 2013.
- [11] GAYRAUD, Richard; JACQUES, Olivier; DAY, Robert. **SIPp reference documentation**. Disponível em: <<http://sipp.sourceforge.net/doc/reference.pdf>>. Acesso em: 10 ago. 2017.
- [12] WIRESHARK (Org.). **RTP statistics**. 2009. Disponível em: <https://wiki.wireshark.org/RTP_statistics>. Acesso em: 6 nov. 2017.

- [13] RASPBX. [S.l.: s.n.], 2017. Disponível em: <<http://www.raspberry-asterisk.org/downloads>>. Acesso em: 10 ago. 2017.
- [14] VANILLA Installation of Asterisk 13 on Raspberry Pi 3 with G729 Codec. Disponível em: <<http://www.raspberry-asterisk.org/http://blog.pztop.com/2016/11/01/Asterisk13-on-RaspberryPi3//>>. Acesso em: 10 ago. 2017.
- [15] LEWIS, Tony. **Installing FreePBX with the Official Distro**. 2015. Disponível em: <<https://wiki.freepbx.org/display/FOP/Installing+FreePBX+with+the+Official+Distro>>. Acesso em: 6 nov. 2017.
- [16] FREE SOFTWARE FOUNDATION. **Sipp 'uac' scenario with pcap (rtp) play**. Disponível em: <https://github.com/saghul/sipp-scenarios/blob/master/sipp_uac_pcap_g711a.xml>. Acesso em: 10 ago. 2017.
- [17] AHMED, Mohiuddin; MANSOR, Abdul Malik. CPU dimensioning on performance of Asterisk VoIP PBX. **Proceedings Of The 11th Communications And Networking Simulation Symposium On - Cns '08**, [s.l.], p.139-146, 2008.
- [18] EPIC. Produção de Bensound. [s.l.]: [s.i], 2017. Son., P&B. Disponível em: <<http://www.bensound.com/royalty-free-music/track/epic>>. Acesso em: 10 ago. 2017.
- [19] PINGMAN TOOLS. **How is MOS calculated in PingPlotter Pro?** Disponível em: <<https://www.pingman.com/kb/article/how-is-mos-calculated-in-pingplotter-pro-50.html>>. Acesso em: 6 nov. 2017.
- [20] COCKER, E.; GHAZZI, F.; SPEIDEL, U.. Quality Trend Measurement of Long-distance VoIP Communication via Estimated Mean Opinion Score. In: INTERNATIONAL CONFERENCE ON INFORMATION, COMMUNICATIONS AND SIGNAL PROCESSING (ICICS), 10., 2015, Singapura. [S.I]: IEE, 2015.
- [21] TOGINHO FILHO, D. O; ANDRELLO, A.c. **Medição e propagação de erros**. 2009. Disponível em: <<http://www.esalq.usp.br/departamentos/leb/aulas/lce5702/medicao.pdf>>. Acesso em: 6 nov. 2017.
- [22] HOLE, David P.; TOBAGI, Fouad A.. Capacity of an IEEE 802.11b Wireless LAN supporting VoIP. In: IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS, 2004, Paris. **Paper**. Paris: Ieee, 2004.
- [23] VILLACÍS, David; ACOSTA, Freddy R.; CUEVA, Román A. Lara. Performance Analysis of VoIP Services over WiFi-based systems. In: IEEE COLOMBIAN

CONFERENCE ON COMMUNICATIONS AND COMPUTING, 2013,
Medelin. **Paper.** Medelin: Ieee, 2013.

[24] WANG, W.; LIEW, S.c.; LI, V.o.k.. Solutions to Performance Problems in VoIP Over a 802.11 Wireless LAN. **Ieee Transactions On Vehicular Technology**, [s.l.], v. 54, n. 1, p.366-384, jan. 2005. Institute of Electrical and Electronics Engineers (IEEE).

ANEXO A – Cenário 1 do SIPp utilizando o codec G711

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">
<!-- This program is free software; you can redistribute it and/or -->
<!-- modify it under the terms of the GNU General Public License as -->
<!-- published by the Free Software Foundation; either version 2 of the -->
<!-- License, or (at your option) any later version. -->
<!-- -->
<!-- This program is distributed in the hope that it will be useful, -->
<!-- but WITHOUT ANY WARRANTY; without even the implied warranty of -->
<!-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the -->
-->
<!-- GNU General Public License for more details. -->
<!-- -->
<!-- You should have received a copy of the GNU General Public License -->
<!-- along with this program; if not, write to the -->
<!-- Free Software Foundation, Inc., -->
<!-- 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA -->
<!-- -->
<!-- Sipp 'uac' scenario with pcap (rtp) play -->
<!-- -->
<scenario name="UAC with media">
<!-- In client mode (sipp placing calls), the Call-ID MUST be -->
```



```
<!-- generated by sipp. To do so, use [call_id] keyword. -->
<send retrans="500">
<![CDATA[
  INVITE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
  Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
  From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]
  To: sut <sip:[service]@[remote_ip]:[remote_port]>
  Call-ID: [call_id]
  CSeq: 1 INVITE
  Contact: sip:sipp@[local_ip]:[local_port]
  Max-Forwards: 70
  Subject: Performance Test
  Content-Type: application/sdp
  Content-Length: [len]
  v=0
  o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
  s=-
  c=IN IP[local_ip_type] [local_ip]
  t=0 0
  m=audio [auto_media_port] RTP/AVP 8 101
  a=rtpmap:8 PCMA/8000
  a=rtpmap:101 telephone-event/8000
  a=fmtp:101 0-11,16
]]>
```

```

</send>

<recv response="100" optional="true">

</recv>

<recv response="180" optional="true">

</recv>

<!-- By adding rrs="true" (Record Route Sets), the route sets -->

<!-- are saved and used for following messages sent. Useful to test -->

<!-- against stateful SIP proxies/B2BUAs. -->

<recv response="200" rtd="true" crlf="true">

</recv>

<!-- Packet lost can be simulated in any send/recv message by -->

<!-- by adding the 'lost = "10"'. Value can be [1-100] percent. -->

<send>

<![CDATA[

    ACK sip:[service]@[remote_ip]:[remote_port] SIP/2.0

    Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]

    From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]

    To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]

    Call-ID: [call_id]

    CSeq: 1 ACK

    Contact: sip:sipp@[local_ip]:[local_port]

    Max-Forwards: 70

    Subject: Performance Test

    Content-Length: 0
  ]]>

```

```
]]>

</send>

<!-- Play a pre-recorded PCAP file (RTP stream) -->

<nop>

<action>

  <exec play_pcap_audio="AudioG711.pcap"/>

</action>

</nop>

<!-- Pause 660 seconds, which is approximately the duration of the -->
<!-- PCAP file -->

<pause milliseconds="660000"/>

<!-- The 'crlf' option inserts a blank line in the statistics report. -->

<send retrans="500">

<![CDATA[

  BYE sip:[service]@[remote_ip]:[remote_port] SIP/2.0

  Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]

  From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]

  To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]

  Call-ID: [call_id]

  CSeq: 2 BYE

  Contact: sip:sipp@[local_ip]:[local_port]

  Max-Forwards: 70

  Subject: Performance Test

  Content-Length: 0
```

```
]]>
</send>

<recv response="200" crlf="true">
</recv>

<!-- definition of the response time repartition table (unit is ms) -->
<ResponseTimeRepartition value="10, 20, 30, 40, 50, 100, 150, 200"/>
<!-- definition of the call length repartition table (unit is ms) -->
<CallLengthRepartition value="10, 50, 100, 500, 1000, 5000, 10000"/>
</scenario>
```

ANEXO B – Cenário 2 do SIPp utilizando o codec G711

```

<?xml version="1.0" encoding="ISO-8859-1" ?>

<!DOCTYPE scenario SYSTEM "sipp.dtd">

<!-- This program is free software; you can redistribute it and/or -->
<!-- modify it under the terms of the GNU General Public License as -->
<!-- published by the Free Software Foundation; either version 2 of the -->
<!-- License, or (at your option) any later version. -->

<!-- -->

<!-- This program is distributed in the hope that it will be useful, -->
<!-- but WITHOUT ANY WARRANTY; without even the implied warranty of -->
<!-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the -->
-->

<!-- GNU General Public License for more details. -->

<!-- -->

<!-- You should have received a copy of the GNU General Public License -->
<!-- along with this program; if not, write to the -->
<!-- Free Software Foundation, Inc., -->
<!-- 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA -->

<!-- -->

<!-- Sipp 'uac' scenario with pcap (rtp) play -->

<!-- -->

<scenario name="UAC with media">

<!-- In client mode (sipp placing calls), the Call-ID MUST be -->
<!-- generated by sipp. To do so, use [call_id] keyword. -->

```

```
<send retrans="500">
<![CDATA[
  INVITE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
  Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
  From: sipp2 <sip:sipp2@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]
  To: sut <sip:[service]@[remote_ip]:[remote_port]>
  Call-ID: [call_id]
  CSeq: 1 INVITE
  Contact: sip:sipp2@[local_ip]:[local_port]
  Max-Forwards: 70
  Subject: Performance Test
  Content-Type: application/sdp
  Content-Length: [len]
  v=0
  o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
  s=-
  c=IN IP[local_ip_type] [local_ip]
  t=0 0
  m=audio [auto_media_port] RTP/AVP 8 101
  a=rtpmap:8 PCMA/8000
  a=rtpmap:101 telephone-event/8000
  a=fmtp:101 0-11,16
  ]]>
</send>
```

```

<recv response="100" optional="true">

</recv>

<recv response="180" optional="true">

</recv>

<!-- By adding rrs="true" (Record Route Sets), the route sets -->
<!-- are saved and used for following messages sent. Useful to test -->
<!-- against stateful SIP proxies/B2BUAs. -->

<recv response="200" rtd="true" crlf="true">

</recv>

<!-- Packet lost can be simulated in any send/recv message by -->
<!-- by adding the 'lost = "10"'. Value can be [1-100] percent. -->

<send>

<![CDATA[

ACK sip:[service]@[remote_ip]:[remote_port] SIP/2.0

Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]

From: sipp2 <sip:sipp2@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]

To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]

Call-ID: [call_id]

CSeq: 1 ACK

Contact: sip:sipp2@[local_ip]:[local_port]

Max-Forwards: 70

Subject: Performance Test

Content-Length: 0

]]>

```

```
</send>

<!-- Play a pre-recorded PCAP file (RTP stream) -->

<nop>

  <action>

    <exec play_pcap_audio="AudioG711.pcap"/>

  </action>

</nop>

<!-- Pause 600 seconds, which is approximately the duration of the -->
<!-- PCAP file -->

<pause milliseconds="600000"/>

<!-- The 'crlf' option inserts a blank line in the statistics report. -->

<send retrans="500">

  <![CDATA[

    BYE sip:[service]@[remote_ip]:[remote_port] SIP/2.0

    Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]

    From: sipp2 <sip:sipp2@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]

    To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]

    Call-ID: [call_id]

    CSeq: 2 BYE

    Contact: sip:sipp2@[local_ip]:[local_port]

    Max-Forwards: 70

    Subject: Performance Test

    Content-Length: 0

  ]]>
```



```
</send>  
  
<recv response="200" crlf="true">  
  
</recv>  
  
<!-- definition of the response time repartition table (unit is ms) -->  
  
<ResponseTimeRepartition value="10, 20, 30, 40, 50, 100, 150, 200"/>  
  
<!-- definition of the call length repartition table (unit is ms) -->  
  
<CallLengthRepartition value="10, 50, 100, 500, 1000, 5000, 10000"/>  
  
</scenario>
```

ANEXO C – Cenário 1 do SIPp utilizando o codec G729

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">
<!-- This program is free software; you can redistribute it and/or -->
<!-- modify it under the terms of the GNU General Public License as -->
<!-- published by the Free Software Foundation; either version 2 of the -->
<!-- License, or (at your option) any later version. -->
<!-- -->
<!-- This program is distributed in the hope that it will be useful, -->
<!-- but WITHOUT ANY WARRANTY; without even the implied warranty of -->
<!-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the -->
-->
<!-- GNU General Public License for more details. -->
<!-- -->
<!-- You should have received a copy of the GNU General Public License -->
<!-- along with this program; if not, write to the -->
<!-- Free Software Foundation, Inc., -->
<!-- 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA -->
<!-- -->
<!-- Sipp 'uac' scenario with pcap (rtp) play -->
<!-- -->
<scenario name="UAC with media">
<!-- In client mode (sipp placing calls), the Call-ID MUST be -->
```

```
<!-- generated by sipp. To do so, use [call_id] keyword. -->
<send retrans="500">
<![CDATA[
  INVITE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
  Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
  From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]
  To: sut <sip:[service]@[remote_ip]:[remote_port]>
  Call-ID: [call_id]
  CSeq: 1 INVITE
  Contact: sip:sipp@[local_ip]:[local_port]
  Max-Forwards: 70
  Subject: Performance Test
  Content-Type: application/sdp
  Content-Length: [len]
  v=0
  o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
  s=-
  c=IN IP[local_ip_type] [local_ip]
  t=0 0
  m=audio [auto_media_port] RTP/AVP 18 101
  a=rtpmap:18 G729/8000
  a=rtpmap:101 telephone-event/8000
  a=fmtp:101 0-11,16
]]>
```

```

</send>

<recv response="100" optional="true">

</recv>

<recv response="180" optional="true">

</recv>

<!-- By adding rrs="true" (Record Route Sets), the route sets -->

<!-- are saved and used for following messages sent. Useful to test -->

<!-- against stateful SIP proxies/B2BUAs. -->

<recv response="200" rtd="true" crlf="true">

</recv>

<!-- Packet lost can be simulated in any send/recv message by -->

<!-- by adding the 'lost = "10"'. Value can be [1-100] percent. -->

<send>

<![CDATA[

    ACK sip:[service]@[remote_ip]:[remote_port] SIP/2.0

    Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]

    From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]

    To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]

    Call-ID: [call_id]

    CSeq: 1 ACK

    Contact: sip:sipp@[local_ip]:[local_port]

    Max-Forwards: 70

    Subject: Performance Test

    Content-Length: 0
  ]]>

```

```
]]>

</send>

<!-- Play a pre-recorded PCAP file (RTP stream) -->

<nop>

  <action>

    <exec play_pcap_audio="AudioG729.pcap"/>

  </action>

</nop>

<!-- Pause 660 seconds, which is approximately the duration of the -->
<!-- PCAP file -->

<pause milliseconds="660000"/>

<!-- The 'crlf' option inserts a blank line in the statistics report. -->

<send retrans="500">

  <![CDATA[

    BYE sip:[service]@[remote_ip]:[remote_port] SIP/2.0

    Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]

    From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]

    To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]

    Call-ID: [call_id]

    CSeq: 2 BYE

    Contact: sip:sipp@[local_ip]:[local_port]

    Max-Forwards: 70

    Subject: Performance Test

    Content-Length: 0
```

```
]]>
</send>

<recv response="200" crlf="true">
</recv>

<!-- definition of the response time repartition table (unit is ms) -->
<ResponseTimeRepartition value="10, 20, 30, 40, 50, 100, 150, 200"/>
<!-- definition of the call length repartition table (unit is ms) -->
<CallLengthRepartition value="10, 50, 100, 500, 1000, 5000, 10000"/>
</scenario>
```

ANEXO D – Cenário 2 do SIPp utilizando o codec G729

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">
<!-- This program is free software; you can redistribute it and/or -->
<!-- modify it under the terms of the GNU General Public License as -->
<!-- published by the Free Software Foundation; either version 2 of the -->
<!-- License, or (at your option) any later version. -->
<!-- -->
<!-- This program is distributed in the hope that it will be useful, -->
<!-- but WITHOUT ANY WARRANTY; without even the implied warranty of -->
<!-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the -->
-->
<!-- GNU General Public License for more details. -->
<!-- -->
<!-- You should have received a copy of the GNU General Public License -->
<!-- along with this program; if not, write to the -->
<!-- Free Software Foundation, Inc., -->
<!-- 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA -->
<!-- -->
<!-- Sipp 'uac' scenario with pcap (rtp) play -->
<!-- -->
<scenario name="UAC with media">
<!-- In client mode (sipp placing calls), the Call-ID MUST be -->
```

```
<!-- generated by sipp. To do so, use [call_id] keyword. -->
<send retrans="500">
<![CDATA[
  INVITE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
  Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
  From: sipp2 <sip:sipp2@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]
  To: sut <sip:[service]@[remote_ip]:[remote_port]>
  Call-ID: [call_id]
  CSeq: 1 INVITE
  Contact: sip:sipp2@[local_ip]:[local_port]
  Max-Forwards: 70
  Subject: Performance Test
  Content-Type: application/sdp
  Content-Length: [len]
  v=0
  o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
  s=-
  c=IN IP[local_ip_type] [local_ip]
  t=0 0
  m=audio [auto_media_port] RTP/AVP 18 101
  a=rtpmap:18 G729/8000
  a=rtpmap:101 telephone-event/8000
  a=fmtp:101 0-11,16
]]>
```



```

</send>

<recv response="100" optional="true">

</recv>

<recv response="180" optional="true">

</recv>

<!-- By adding rrs="true" (Record Route Sets), the route sets -->
<!-- are saved and used for following messages sent. Useful to test -->
<!-- against stateful SIP proxies/B2BUAs. -->
<recv response="200" rtd="true" crlf="true">

</recv>

<!-- Packet lost can be simulated in any send/recv message by -->
<!-- by adding the 'lost = "10"'. Value can be [1-100] percent. -->

<send>

<![CDATA[

ACK sip:[service]@[remote_ip]:[remote_port] SIP/2.0

Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]

From: sipp2 <sip:sipp2@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]

To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]

Call-ID: [call_id]

CSeq: 1 ACK

Contact: sip:sipp2@[local_ip]:[local_port]

Max-Forwards: 70

Subject: Performance Test

Content-Length: 0

```

```
]]>

</send>

<!-- Play a pre-recorded PCAP file (RTP stream) -->

<nop>

<action>

  <exec play_pcap_audio="AudioG729.pcap"/>

</action>

</nop>

<!-- Pause 600 seconds, which is approximately the duration of the -->
<!-- PCAP file -->

<pause milliseconds="600000"/>

<!-- The 'crlf' option inserts a blank line in the statistics report. -->

<send retrans="500">

<![CDATA[

  BYE sip:[service]@[remote_ip]:[remote_port] SIP/2.0

  Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]

  From: sipp2 <sip:sipp2@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]

  To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]

  Call-ID: [call_id]

  CSeq: 2 BYE

  Contact: sip:sipp2@[local_ip]:[local_port]

  Max-Forwards: 70

  Subject: Performance Test

  Content-Length: 0
```

```
]]>
</send>

<recv response="200" crlf="true">

</recv>

<!-- definition of the response time repartition table (unit is ms) -->
<ResponseTimeRepartition value="10, 20, 30, 40, 50, 100, 150, 200"/>

<!-- definition of the call length repartition table (unit is ms) -->
<CallLengthRepartition value="10, 50, 100, 500, 1000, 5000, 10000"/>

</scenario>
```

ANEXO E – Cenário 1 do SIPp utilizando o codec GSM

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">
<!-- This program is free software; you can redistribute it and/or -->
<!-- modify it under the terms of the GNU General Public License as -->
<!-- published by the Free Software Foundation; either version 2 of the -->
<!-- License, or (at your option) any later version. -->
<!-- -->
<!-- This program is distributed in the hope that it will be useful, -->
<!-- but WITHOUT ANY WARRANTY; without even the implied warranty of -->
<!-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the -->
-->
<!-- GNU General Public License for more details. -->
<!-- -->
<!-- You should have received a copy of the GNU General Public License -->
<!-- along with this program; if not, write to the -->
<!-- Free Software Foundation, Inc., -->
<!-- 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA -->
<!-- -->
<!-- Sipp 'uac' scenario with pcap (rtp) play -->
<!-- -->
<scenario name="UAC with media">
<!-- In client mode (sipp placing calls), the Call-ID MUST be -->
```

```
<!-- generated by sipp. To do so, use [call_id] keyword. -->
<send retrans="500">
<![CDATA[
  INVITE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
  Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
  From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]
  To: sut <sip:[service]@[remote_ip]:[remote_port]>
  Call-ID: [call_id]
  CSeq: 1 INVITE
  Contact: sip:sipp@[local_ip]:[local_port]
  Max-Forwards: 70
  Subject: Performance Test
  Content-Type: application/sdp
  Content-Length: [len]
  v=0
  o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
  s=-
  c=IN IP[local_ip_type] [local_ip]
  t=0 0
  m=audio [auto_media_port] RTP/AVP 3 101
  a=rtpmap:3 GSM/8000
  a=rtpmap:101 telephone-event/8000
  a=fmtp:101 0-11,16
]]>
```

```

</send>

<recv response="100" optional="true">

</recv>

<recv response="180" optional="true">

</recv>

<!-- By adding rrs="true" (Record Route Sets), the route sets -->

<!-- are saved and used for following messages sent. Useful to test -->

<!-- against stateful SIP proxies/B2BUAs. -->

<recv response="200" rtd="true" crlf="true">

</recv>

<!-- Packet lost can be simulated in any send/recv message by -->

<!-- by adding the 'lost = "10"'. Value can be [1-100] percent. -->

<send>

<![CDATA[

    ACK sip:[service]@[remote_ip]:[remote_port] SIP/2.0

    Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]

    From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]

    To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]

    Call-ID: [call_id]

    CSeq: 1 ACK

    Contact: sip:sipp@[local_ip]:[local_port]

    Max-Forwards: 70

    Subject: Performance Test

    Content-Length: 0
  ]]>

```

```
]]>
</send>
<!-- Play a pre-recorded PCAP file (RTP stream) -->
<nop>
<action>
  <exec play_pcap_audio="AudioGSM.pcap"/>
</action>
</nop>
<!-- Pause 660 seconds, which is approximately the duration of the -->
<!-- PCAP file -->
<pause milliseconds="660000"/>
<!-- The 'crlf' option inserts a blank line in the statistics report. -->
<send retrans="500">
<![CDATA[
  BYE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
  Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
  From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]
  To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]
  Call-ID: [call_id]
  CSeq: 2 BYE
  Contact: sip:sipp@[local_ip]:[local_port]
  Max-Forwards: 70
  Subject: Performance Test
  Content-Length: 0
```

```
]]>
</send>
<recv response="200" crlf="true">
</recv>
<!-- definition of the response time repartition table (unit is ms) -->
<ResponseTimeRepartition value="10, 20, 30, 40, 50, 100, 150, 200"/>
<!-- definition of the call length repartition table (unit is ms) -->
<CallLengthRepartition value="10, 50, 100, 500, 1000, 5000, 10000"/>
</scenario>
```


ANEXO F – Cenário 2 do SIPp utilizando o codec GSM

```

<?xml version="1.0" encoding="ISO-8859-1" ?>

<!DOCTYPE scenario SYSTEM "sipp.dtd">

<!-- This program is free software; you can redistribute it and/or -->
<!-- modify it under the terms of the GNU General Public License as -->
<!-- published by the Free Software Foundation; either version 2 of the -->
<!-- License, or (at your option) any later version. -->

<!-- -->

<!-- This program is distributed in the hope that it will be useful, -->
<!-- but WITHOUT ANY WARRANTY; without even the implied warranty of -->
<!-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the -->
-->

<!-- GNU General Public License for more details. -->

<!-- -->

<!-- You should have received a copy of the GNU General Public License -->
<!-- along with this program; if not, write to the -->
<!-- Free Software Foundation, Inc., -->
<!-- 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA -->

<!-- -->

<!-- Sipp 'uac' scenario with pcap (rtp) play -->

<!-- -->

<scenario name="UAC with media">

<!-- In client mode (sipp placing calls), the Call-ID MUST be -->

```

```
<!-- generated by sipp. To do so, use [call_id] keyword. -->
<send retrans="500">
<![CDATA[
  INVITE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
  Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
  From: sipp2 <sip:sipp2@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]
  To: sut <sip:[service]@[remote_ip]:[remote_port]>
  Call-ID: [call_id]
  CSeq: 1 INVITE
  Contact: sip:sipp2@[local_ip]:[local_port]
  Max-Forwards: 70
  Subject: Performance Test
  Content-Type: application/sdp
  Content-Length: [len]
  v=0
  o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
  s=-
  c=IN IP[local_ip_type] [local_ip]
  t=0 0
  m=audio [auto_media_port] RTP/AVP 3 101
  a=rtpmap:3 GSM/8000
  a=rtpmap:101 telephone-event/8000
  a=fmtp:101 0-11,16
]]>
```

```

</send>

<recv response="100" optional="true">

</recv>

<recv response="180" optional="true">

</recv>

<!-- By adding rrs="true" (Record Route Sets), the route sets -->

<!-- are saved and used for following messages sent. Useful to test -->

<!-- against stateful SIP proxies/B2BUAs. -->

<recv response="200" rtd="true" crlf="true">

</recv>

<!-- Packet lost can be simulated in any send/recv message by -->

<!-- by adding the 'lost = "10"'. Value can be [1-100] percent. -->

<send>

<![CDATA[

    ACK sip:[service]@[remote_ip]:[remote_port] SIP/2.0

    Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]

    From: sipp2 <sip:sipp2@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]

    To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]

    Call-ID: [call_id]

    CSeq: 1 ACK

    Contact: sip:sipp2@[local_ip]:[local_port]

    Max-Forwards: 70

    Subject: Performance Test

    Content-Length: 0
  ]]>

```

```
]]>

</send>

<!-- Play a pre-recorded PCAP file (RTP stream) -->

<nop>

<action>

  <exec play_pcap_audio="AudioGSM.pcap"/>

</action>

</nop>

<!-- Pause 600 seconds, which is approximately the duration of the -->
<!-- PCAP file -->

<pause milliseconds="600000"/>

<!-- The 'crlf' option inserts a blank line in the statistics report. -->

<send retrans="500">

<![CDATA[

  BYE sip:[service]@[remote_ip]:[remote_port] SIP/2.0

  Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]

  From: sipp2 <sip:sipp2@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]

  To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]

  Call-ID: [call_id]

  CSeq: 2 BYE

  Contact: sip:sipp2@[local_ip]:[local_port]

  Max-Forwards: 70

  Subject: Performance Test

  Content-Length: 0
```

```
]]>
</send>
<recv response="200" crlf="true">
</recv>
<!-- definition of the response time repartition table (unit is ms) -->
<ResponseTimeRepartition value="10, 20, 30, 40, 50, 100, 150, 200"/>
<!-- definition of the call length repartition table (unit is ms) -->
<CallLengthRepartition value="10, 50, 100, 500, 1000, 5000, 10000"/>
</scenario>
```