

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DO CURSO DE GRADUAÇÃO EM TECNOLOGIA EM
SISTEMAS PARA INTERNET**

MATHEUS GARCIA BESSEGATO

**ESTUDO DE CASO DE VIABILIDADE DE IMPLANTAÇÃO DE
SISTEMA DE TELEFONIA VOIP COM ASTERISK NA
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ -
CÂMPUS TOLEDO**

TRABALHO DE CONCLUSÃO DE CURSO

TOLEDO

2016

MATHEUS GARCIA BESSEGATO

**ESTUDO DE CASO DE VIABILIDADE DE IMPLANTAÇÃO DE
SISTEMA DE TELEFONIA VOIP COM ASTERISK NA
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ -
CÂMPUS TOLEDO**

Trabalho de conclusão de curso de graduação, apresentada à Coordenação do Curso de Graduação em Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Me. Alexandre Huff

TOLEDO

2016

AGRADECIMENTOS

Primeiramente, agradeço a Deus pelo dom da vida e por todas as oportunidades que Ele tem me proporcionado. Agradeço também a meus pais e irmãos por todo apoio prestado. Agradeço minha namorada por toda ajuda e paciência.

Agradeço meu orientador e aos demais membros da banca por toda contribuição para a realização do presente trabalho. Agradeço meus colegas do curso, colegas de trabalho e a todos demais amigos e familiares que de alguma forma estiverem presentes e contribuíram para a realização desse trabalho.

RESUMO

BESSEGATO, Matheus. ESTUDO DE CASO DE VIABILIDADE DE IMPLANTAÇÃO DE SISTEMA DE TELEFONIA VOIP COM ASTERISK NA UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - CÂMPUS TOLEDO. 67 f. Trabalho de conclusão de curso – Coordenação do Curso de Graduação em Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Toledo, 2016.

A tecnologia de comunicação Voz sobre o Protocolo de Internet (VoIP) é uma tecnologia que tem sido amplamente adotada e utilizada, especialmente em ambientes corporativos, para a modernização ou ampliação de redes telefônicas convencionais, por proporcionar uma potencial redução de custos e por ser facilmente escalável. A Universidade Tecnológica Federal do Paraná (UTFPR) Câmpus Toledo possui uma central telefônica convencional, em funcionamento desde o ano de 2007. O objetivo desse trabalho é realizar um estudo de caso sobre a viabilidade do processo de migração da telefonia da UTFPR para a tecnologia VoIP, com a avaliação da estrutura atual e da estrutura necessária para a concretização da mudança, além de testes práticos em laboratório. O sistema de telefonia atual da universidade já não suporta a demanda da mesma e não pode ser ampliado por falta de placas de expansão e por a central não possuir mais espaço para a adição de novas placas. Foi configurado um projeto piloto de um servidor VoIP com o *framework open source* Asterisk e realizados testes por usuários. O estudo mostra que a infraestrutura de rede da UTFPR oferece suporte à implantação da tecnologia, em substituição ao sistema telefônico atual da universidade.

Palavras-chave: VoIP, IPBX, Asterisk, SIP

ABSTRACT

BESSEGATO, Matheus. CASE STUDY OF VOIP TELEPHONY SYSTEM DEPLOYMENT VIABILITY WITH ASTERISK IN UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ. 67 f. Trabalho de conclusão de curso – Coordenação do Curso de Graduação em Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Toledo, 2016.

Voice over Internet Protocol (VoIP) is a technology that has been widely adopted and used, especially in corporate environments for the modernization or expansion of conventional telephone networks, by providing a potential cost reduction and for being easily scalable. The Federal Technological University of Paraná (UTFPR) Campus Toledo is operating a standard telephone switching system, since the year of 2007. The aim of this work is to conduct a case study on the process of migrating the UTFPR current telephony system to VoIP technology, with the assessment of the current structure and the necessary structure for the implementation of the change, as well as practice tests in the laboratory. The current University's telephone system no longer supports its demand and cannot be expanded due to lack of expansion boards and the fact that the central office has no more space for adding new boards. A pilot project of a VoIP server with the open source Asterisk framework was configured and experienced by users. The study shows that UTFPR's network infrastructure supports the deployment of the technology, replacing the university's current telephone system.

Keywords: VoIP, IPBX, Asterisk, SIP

LISTA DE FIGURAS

FIGURA 1	– Cabeçalho IPv4	16
FIGURA 2	– Cabeçalho IPv6	17
FIGURA 3	– Cabeçalho UDP	19
FIGURA 4	– Cabeçalho TCP	21
FIGURA 5	– <i>Handshake</i> Triplo	22
FIGURA 6	– Fluxo Básico Seção SIP	30
FIGURA 7	– Relacionamentos Componentes Asterisk	34
FIGURA 8	– Atual Topologia de Rede Física da UTFPR - Toledo	41
FIGURA 9	– Atual Rede de Telefonia da UTFPR - Toledo	42
FIGURA 10	– Respostas Questionário de Funcionalidades VoIP	43
FIGURA 11	– Processo de Execução de um <i>script</i> AGI no Asterisk	45
FIGURA 12	– Manual de Uso das Funcionalidades do IPBX	46
FIGURA 13	– Topologia de Rede Física da UTFPR - Toledo com VoIP	48
FIGURA 14	– Taxa de Saída do <i>Switch Core</i> (Mbps) - 31/10/2016	50
FIGURA 15	– Taxa de Entrada do <i>Switch Core</i> (Mbps) - 31/10/2016	50
FIGURA 16	– Taxa de Saída <i>Switch Core</i> (Mbps) - 01/11/2016	51
FIGURA 17	– Taxa de Entrada <i>Switch Core</i> (Mbps) - 01/11/2016	52
FIGURA 18	– Taxa de Saída <i>Switch Core</i> (Mbps) - 03/11/2016	52
FIGURA 19	– Taxa de Entrada <i>Switch Core</i> (Mbps) - 03/11/2016	53
FIGURA 20	– Estatísticas da Rede em Ligação VoIP	54
FIGURA 21	– Contagem de Equipamentos Utilizados para os Testes	55
FIGURA 22	– Contagem do Tipo de Rede Utilizada para os Testes	55
FIGURA 23	– Avaliação de presença de eco	56
FIGURA 24	– Avaliação de atraso de voz	56
FIGURA 25	– Avaliação de ruído	57
FIGURA 26	– Avaliação de falhas na transmissão da voz	57
FIGURA 27	– Avaliação da autenticação	58
FIGURA 28	– Avaliação do estacionamento de chamadas	59
FIGURA 29	– Avaliação da transferência de chamadas	59
FIGURA 30	– Avaliação da captura de chamadas	60
FIGURA 31	– Avaliação do identificador de chamadas	60
FIGURA 32	– Avaliação do não perturbe	61
FIGURA 33	– Avaliação do <i>voicemail</i>	61
FIGURA 34	– Avaliação geral de qualidade	62

LISTA DE TABELAS

TABELA 1	– Camadas Modelos de Referência	15
TABELA 2	– Comparativo entre difentes <i>Codecs</i>	28
TABELA 3	– Protocolos Utilizados pelo H.323	28
TABELA 4	– Versões Asterisk	33
TABELA 5	– Módulos Asterisk	35
TABELA 6	– Estados dos Dispositivos	38
TABELA 7	– Estados de Presença	39

LISTA DE SIGLAS

UTFPR	Universidade Tecnológica Federal do Paraná
VoIP	<i>Voice over Internet Protocol</i>
NVT	<i>Network Voice Protocol</i>
NVCP	<i>Network Voice Control Protocol</i>
IETF	<i>Internet Engineering Task Force</i>
PBX	<i>Private Branch Exchange</i>
IPBX	<i>Internet Protocol Private Branch Exchange</i>
OSI	<i>Open Systems Interconnection</i>
ISO	<i>International Standards Organization</i>
TCP	<i>Transport Control Protocol</i>
IP	<i>Internet Protocol</i>
HTTP	<i>Hypertext Transfer Protocol</i>
RTP	<i>Real-time Transport Protocol</i>
IPv4	<i>Internet Protocol version 4</i>
IPv6	<i>Internet Protocol version 6</i>
IHL	<i>Internet Header Length</i>
DSCP	<i>Differentiated Services Code Point</i>
QoS	<i>Quality of Service</i>
ECN	<i>Explicit Congestion Notification</i>
TTL	<i>Time-to-Live</i>
ICMP	<i>Internet Control Message Protocol</i>
UDP	<i>User Datagram Protocol</i>
DNS	<i>Domain Name System</i>
RFC	<i>Request for Comments</i>
MTU	<i>Maximum Transfer Unit</i>
ITU	<i>International Telecommunication Union</i>
SIP	<i>Session Initiation Protocol</i>
SDP	<i>Session Description Protocol</i>
URI	<i>Uniform Resource Identifier</i>
API	<i>Application Program Interface</i>
AGI	<i>Asterisk Gateway Interface</i>

SUMÁRIO

1	INTRODUÇÃO	10
2	OBJETIVOS	12
2.1	OBJETIVO GERAL	12
2.2	OBJETIVOS ESPECÍFICOS	12
3	JUSTIFICATIVA	13
4	REFERENCIAL TEÓRICO	14
4.1	MODELO TCP/IP	14
4.2	IP (<i>INTERNET PROTOCOL</i>)	15
4.3	PROTOCOLOS DA CAMADA DE TRANSPORTE	18
4.3.1	UDP (<i>User Datagram Protocol</i>)	18
4.3.2	TCP (<i>Transmission Control Protocol</i>)	19
4.3.2.1	Estabelecimento de Conexões TCP	22
4.4	RTP (<i>REAL-TIME TRANSPORT PROTOCOL</i>)	23
4.5	RTCP (<i>REAL-TIME CONTROL PROTOCOL</i>)	24
4.6	VOIP (<i>VOICE OVER INTERNET PROTOCOL</i>)	24
4.7	CODECS	26
4.7.1	Codec G.711	27
4.7.2	Codec G.729	27
4.7.3	Codec G.723.1	28
4.8	H.323	28
4.9	SIP (<i>SESSION INITIATION PROTOCOL</i>)	29
4.10	IAX (<i>INTER-ASTERISK EXCHANGE</i>)	31
4.11	ASTERISK	31
4.11.1	Versões	32
4.11.2	Arquitetura do Asterisk	33
4.11.3	Objetos	34
4.11.4	Comentários	35
4.11.5	<i>Templates</i>	36
4.11.6	Banco de Dados Asterisk	36
4.11.7	Conceitos Chave	37
4.11.7.1	<i>Channels</i>	37
4.11.7.2	<i>Bridges</i>	37
4.11.7.3	<i>States and Presence</i>	37
4.11.8	Estados dos Dispositivos	38
4.11.9	Estados das Extensões	38
5	MATERIAIS E MÉTODOS	40
5.1	INFRAESTRUTURA ATUAL DA UTFPR	40
5.2	FUNCIONALIDADES DO IPBX	42
5.3	IMPLEMENTAÇÃO E TESTES	44
5.4	METODOLOGIA DE AVALIAÇÃO DE QUALIDADE DA CHAMADA	46
5.5	METODOLOGIA DE AFERIÇÃO DO CONSUMO DE BANDA INTERNA	47

5.6	INFRAESTRUTURA NECESSÁRIA AO VOIP NA UTFPR	47
6	RESULTADOS	49
6.1	CONSUMO DE BANDA INTERNO DA UTFPR	49
6.2	AVALIAÇÃO DE QUALIDADE DO IPBX COM ASTERISK	54
7	CONCLUSÃO	63
	REFERÊNCIAS	65

1 INTRODUÇÃO

As formas de comunicação por meio da Internet estão em constante modificação, visto que foram criadas inicialmente para uso militar e posteriormente acadêmico. Hoje estão presentes em grande parte dos equipamentos eletrônicos (PASSITO et al., 2004).

Tratando-se da transmissão de pacotes de voz por meio da Internet, os experimentos iniciaram-se em 1974 por meio da ARPANET com o protocolo NVT (*Network Voice Protocol*). Dois anos depois ocorreu a primeira áudio conferência utilizando o protocolo NVCP (*Network Voice Control Protocol*). Já no ano de 1983 foi criado o primeiro *Ethernet Phone* pela empresa Xerox. No início da década de 1990 iniciou-se a padronização dos protocolos de transporte de informações multimídia pelo IETF (*Internet Engineering Task Force*) e em 1995 foi publicada a proposta do padrão RTP (*Real-time Transport Protocol*) (BALBINOT et al., 2003).

Ainda no ano de 1995, a empresa Israelense Vocaltec criou a tecnologia VoIP (*Voice over Internet Protocol*) (magicJack VocalTec Ltd., 2016). Essa tecnologia possibilita o estabelecimento de chamadas e o transporte de voz digitalizada utilizando a rede IP (BORDIM, 2010). Nas últimas décadas a tecnologia VoIP vem sendo largamente utilizada, especialmente em ambientes corporativos, com o objetivo de modernizar ou expandir a rede telefônica existente e também reduzir os custos com telefonia, visto que reduz a dependência das operadoras de telefonia tradicionais (HARFF, 2008).

O câmpus Toledo da Universidade Tecnológica Federal do Paraná possui uma central telefônica PBX (*Private Branch Exchange*) com ramais analógicos que está em funcionamento desde o ano de 2007. Esse câmpus da UTFPR deseja trocar seu sistema totalmente para a tecnologia VoIP devido às vantagens que essa tecnologia pode oferecer. Além disso, a central telefônica em funcionamento já não supre a demanda da universidade e não possui as condições mínimas necessárias para ser aprimorada.

O presente trabalho tem como objetivo a realização de um estudo de caso de verificação da viabilidade da migração do atual sistema de telefonia deste câmpus da universidade para a tecnologia VoIP. Esse estudo compreende o levantamento da infraestrutura atual, a im-

plementação e testes de um projeto piloto para a implantação de um sistema de telefonia VoIP utilizando o IPBX (*Internet Protocol Private Branch Exchange*) Asterisk, análises da rede, além do levantamento da infraestrutura necessária para a sua implantação.

O texto está organizado da seguinte forma: o capítulo 2 descreve os objetivos do trabalho; o capítulo 3 mostra a justificativa para a realização do trabalho; o capítulo 4 descreve os principais protocolos utilizados na Internet e as tecnologias estudadas e utilizadas para a realização do trabalho; o capítulo 5 define as tecnologias utilizadas e exhibe como foram realizados os estudos, implementação e testes; o capítulo 6 descreve os resultados, e; o capítulo 7 mostra as conclusões do trabalho e trabalhos futuros.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Atualmente a UTFPR câmpus Toledo possui uma central telefônica PBX (*Private Branch Exchange*) da empresa SIEMENS, modelo Hipath 3550, com ramais analógicos e deseja trocar seu sistema totalmente para a tecnologia VoIP, com o uso do *framework* Asterisk e terminais IP.

O objetivo do presente trabalho é realizar um estudo de caso sobre o processo de migração do sistema atual de telefonia da UTFPR câmpus Toledo para a tecnologia VoIP (*Voice over Internet Protocol*), partindo da infraestrutura atual da universidade, com as seguintes etapas: levantamento bibliográfico/prático para a concretização da mudança; testes práticos com um servidor VoIP com Asterisk.

2.2 OBJETIVOS ESPECÍFICOS

- Fazer o levantamento da infraestrutura de telefonia atual da Universidade Tecnológica Federal do Paraná.
- Configurar servidor Asterisk com as principais funcionalidades de um IPBX (*Internet Protocol Private Branch Exchange*).
- Realizar testes em laboratório das funcionalidades do servidor.
- Calcular o consumo de banda aproximado que ligações VoIP irão utilizar e avaliar o suporte da rede da UTFPR mediante ao consumo de banda estimado.
- Avaliar a qualidade das ligações e das funcionalidades implementadas por meio de testes com voluntários.
- Efetuar o levantamento da infraestrutura necessária para a implantação da telefonia VoIP na Universidade Tecnológica Federal do Paraná.

3 JUSTIFICATIVA

A central telefônica PBX (*Private Branch Exchange*) da UTFPR está em funcionamento desde o ano de 2007. Devido ao tempo de uso da central telefônica, há uma grande dificuldade para encontrar a placa de expansão de ramais no mercado. Além disso, há mais dois problemas que tornam impossível o crescimento da rede telefônica do câmpus: esgotaram-se todos os ramais da central; e, a central não possui mais espaço físico para a instalação de novas placas. Todo esse cenário configura a atual necessidade da Universidade em trocar seu sistema de telefonia. O *framework* Asterisk foi selecionado por ser *open source*, possuir uma vasta documentação, ser amplamente utilizado, e também por apresentar suporte às mais importantes funcionalidades esperadas em uma central telefônica.

4 REFERENCIAL TEÓRICO

4.1 MODELO TCP/IP

No ano de 1970 surgiu o modelo de referência OSI (*Open Systems Interconnection*), criado pela ISO (*International Standards Organization*), mas que foi oficializado apenas em 1983. Então, no ano 1974 foi definido pela primeira vez o modelo de referência TCP/IP, que leva esse nome devido aos seus principais protocolos, que são o TCP (*Transport Control Protocol*) e o IP (*Internet Protocol*). Esse modelo surgiu nos Estados Unidos durante a Guerra Fria e partiu da necessidade do Departamento de Defesa de conectar várias redes de maneira uniforme para que houvesse comunicação de diferentes tipos de mídia, inclusive em tempo real. Esta comunicação deveria ocorrer em todos os pontos estratégicos do país de forma que a comunicação não fosse interrompida caso uma rede parasse de funcionar (TANENBAUM; WETHERALL, 2011). O modelo TCP/IP possui 4 camadas e protocolos bem definidos para cada uma delas. Já o modelo ISO/OSI possui 7 camadas.

Segundo Tanenbaum e Wetherall (2011), o modelo ISO/OSI é um modelo melhor definido que o modelo TCP/IP. Contudo, o modelo ISO/OSI não possui definição de protocolos, o que ocorre no modelo TCP/IP. Como cada modelo possuía um ponto forte, foi largamente adotado um modelo híbrido, constituído pelos dois, que possui 5 camadas: aplicação, transporte, rede, enlace e física. Cada camada possui a função de executar serviços para a camada acima dela utilizando seus próprios protocolos. A Tabela 1 faz uma comparação dos três modelos de referência.

As camadas do Modelo Híbrido mostradas na Tabela 1 são descritas resumidamente a seguir:

1. Camada Física: a tarefa dessa camada é especificar como os *bits* serão transmitidos de um *host* para outro, o que pode variar de acordo com o meio em que serão transmitidos.

2. Camada de Enlace: essa camada deve especificar como enviar mensagens de tamanhos definidos e com níveis de confiabilidade também definidos entre os *hosts* diretamente

Tabela 1: Comparação entre as camadas dos modelos de referência ISO/OSI, TCP/IP e Híbrido

CAMADAS MODELOS DE REFERÊNCIA					
Nº	ISO/OSI	Nº	TCP/IP	Nº	HÍBRIDO
7	Aplicação	4	Aplicação	5	Aplicação
6	Apresentação				
5	Sessão				
4	Transporte	3	Transporte	4	Transporte
3	Rede	2	Rede	3	Rede
2	Enlace de Dados	1	Acesso ao Meio	2	Enlace
1	Física			1	Física

Fonte: Modificado de (TANENBAUM; WETHERALL, 2011)

conectados.

3. Camada de Rede: a tarefa dessa camada é realizar o roteamento dos pacotes em diferentes redes para chegar até o destino final, de forma que sejam entregues ao *host* correto.

4. Camada de Transporte: essa camada é responsável por realizar a comunicação fim a fim entre *hosts*. Ela tem a funcionalidade de interconectar os processos dos dispositivos de origem e destino.

5. Camada de Aplicação: essa camada contém programas responsáveis pela interação com o usuário e que necessitam utilizar a rede para se comunicar com outras aplicações em *hosts* distintos. Inclui protocolos como o HTTP (*Hypertext Transfer Protocol*), o TELNET e o RTP (*Real-time Transport Protocol*) (TANENBAUM; WETHERALL, 2011).

4.2 IP (*INTERNET PROTOCOL*)

O protocolo IP é o elemento que mantém a Internet conectada (TANENBAUM; WETHERALL, 2011). Ele deve fornecer a melhor maneira de transportar os pacotes de um *host* de origem até um *host* de destino, independentemente se os dois estão na mesma rede ou em redes distintas e distantes. A Figura 1 permite que seja visualizado o cabeçalho do Protocolo IP.

O cabeçalho IPv4, mostrado na Figura 1, possui 20 *bytes*, mas pode ser maior com a presença do campo *Options*. Já o cabeçalho IPv6 possui 40 *bytes* e não possui o campo *Options* (FALL, STEVENS, 2012).

O primeiro campo do cabeçalho IP é o *Version*, que transporta a versão do protocolo, que pode ser 4 para o IPv4 ou 6 para o IPv6 e possui 4 *bits* de tamanho. O próximo campo é o IHL (*Internet Header Length*), que informa o tamanho do cabeçalho em palavras de 32 *bits*,

VERSION (4 BITS)	IHL (4 BITS)	DSFIELD (6 BITS)	ECN (2 BITS)	TOTAL LENGTH (16 BITS)	
IDENTIFICATION (16 BITS)			FLAGS (3 BITS)	FRAGMENT OFFSET (13 BITS)	
TIME-TO-LIVE (TTL) (8 BITS)		PROTOCOL (8 BITS)		HEADER CHECKSUM (16 BITS)	
SOURCE IP ADDRESS (32 BITS)					
DESTINATION IP ADDRESS (32 BITS)					
OPTIONS (IF ANY)					
IP DATA (IF ANY)					

Figura 1: Cabeçalho do protocolo IPv4.

Fonte: Adaptado de Fall e Stevens (2011)

já que o tamanho do cabeçalho IPv4 pode variar. O cabeçalho do IPv6 não possui esse campo pois tem tamanho fixo.

Em seguida, há o campo de 6 *bits* DSCP (*Differentiated Services Code Point*) que possui a função de definir as diferentes classes de serviço usadas para aplicar regras de qualidade de serviço (QoS - *Quality of Service*). Após esse campo há o campo ECN (*Explicit Congestion Notification*), que é utilizado para enviar notificações fim a fim sobre o congestionamento da rede, sem a perda de pacotes. No caso de congestionamento, os roteadores podem marcar os pacotes IP ao invés de descartá-los e os *hosts* devem saber interpretar essa marcação.

Há também no cabeçalho IPv4 um campo de 16 *bits* chamado *Total Length* que carrega o tamanho total do datagrama. Ele não existe no IPv6 devido ao seu tamanho fixo. Conforme Fall e Stevens (2011), utilizando esse campo de comprimento e o campo IHL é possível saber exatamente onde a porção de dados do datagrama inicia e qual seu comprimento.

O campo *Identification* serve para implementar fragmentação e remontagem de datagramas IP. Datagramas com grande tamanho são divididos em componentes menores para trafegar por *links* com unidade de transmissão máxima (MTUs) limitadas. Dentro de um determinado endereço de origem, destino e protocolo, fragmentos de um datagrama são agrupados baseado em suas identificações IPv4. Isso requer que as identificações sejam únicas dentro do endereço de origem, destino e protocolo quando a fragmentação for possível ou quando já tiver ocorrido (TOUCH, 2013). O contador chamado *Fragment Offset* tem seu valor incrementado em 1 a cada fragmento enviado, e por meio dele é possível saber se ocorreu fragmentação do datagrama, pois seu valor será maior que 0.

O campo seguinte do cabeçalho é o TTL (*Time-to-Live*), que determina o número de

roteadores que esse datagrama IP poderá passar na rede até ser descartado. Esse valor é atribuído ao datagrama e, a cada roteamento realizado, ele é decrementado em um, evitando que um datagrama fique em *loop* infinito na rede.

O campo seguinte do cabeçalho IPv4 é o *Protocol*. Ele carrega um número que indica qual é o tipo de dados da carga útil do datagrama IP. Após ele, no cabeçalho IPv4, há o campo *Header Checksum*, que é um campo para prevenir a entrega de dados corrompidos no cabeçalho IPv4. Conforme Fall e Stevens (2011), outros protocolos como o ICMP (*Internet Control Message Protocol*), o TCP (*Transmission Control Protocol*) e o UDP (*User Datagram Protocol*), com seus próprios mecanismos de checagem de integridade de dados, ajudam o protocolo IP para que os dados sejam entregues corretamente.

Por fim, há os campos *Source IP Address*, *Destination IP Address*, *Options* e *IP Data*. Os dois primeiros campos representam os endereços IP de origem e de destino dos *hosts* em comunicação, e os dois últimos campos opcionais podem conter opções e a carga útil do pacote IP, respectivamente, com tamanhos variáveis. A Figura 2 permite que seja visualizado o cabeçalho IPv6.

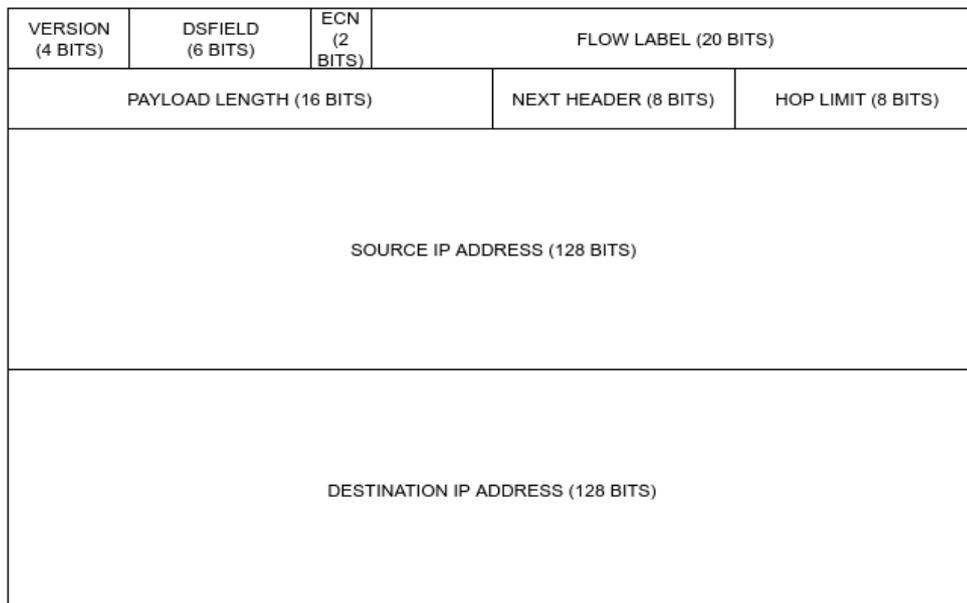


Figura 2: Cabeçalho do protocolo IPv6.

Fonte: Adaptado de Fall e Stevens (2011)

4.3 PROTOCOLOS DA CAMADA DE TRANSPORTE

Há dois protocolos principais que operam na camada de transporte: o UDP (*User Datagram Protocol*) e o TCP (*Transmission Control Protocol*). O primeiro é um protocolo não orientado à conexão, o que significa que não há a necessidade de estabelecimento de uma conexão para o envio de datagramas e também não há confirmação ou garantia da entrega, e o segundo é orientado à conexão, o que significa que antes da transmissão de segmentos deve-se estabelecer uma conexão entre os *hosts* e então haverá garantia de entrega e confirmação do recebimento (TANENBAUM; WETHERALL, 2011).

O protocolo UDP permite o envio de datagramas entre aplicações, assim como também permite que outros protocolos de transporte sejam implementados sobre ele, que é o caso do RTP (*Real-time Transmission Protocol*) que será visto mais a frente.

Já o protocolo TCP é capaz de fazer o envio de segmentos com controle de entrega, controle de fluxo e controle de congestionamento.

4.3.1 UDP (*USER DATAGRAM PROTOCOL*)

O UDP é um protocolo de transmissão de dados não orientado à conexão, que não oferece mecanismo de controle de fluxo e nem controle de entrega (PETERSON; DAVIE, 2011).

O protocolo UDP não comunica o remetente para enviar menos dados a cada instante de tempo, devido à falta de controle de fluxo. Quando os dados enviados pelo remetente chegam ao destinatário, o protocolo coloca o datagrama no final de uma fila e, caso a fila esteja cheia, os dados são perdidos e o remetente não é comunicado para enviá-los novamente. Quando a aplicação quer receber uma mensagem, então é removida uma do início da fila. Caso a fila esteja vazia, o processo é bloqueado até que haja mensagens disponíveis (PETERSON; DAVIE, 2011). A Figura 3 permite que seja visualizado o cabeçalho UDP.

Os datagramas UDP possuem um cabeçalho de 8 bytes, seguidos pela carga útil. Esse cabeçalho é formado por 4 (quatro) partes, que incluem a Porta de Origem (*Source Port Number*), a Porta de Destino (*Destination Port Number*), o Comprimento do UDP (*Length*) e o *Checksum* do UDP.

As portas de origem e destino são responsáveis pela identificação dos processos que os *hosts* estão utilizando para se comunicar. Quando um pacote chega ao destinatário, sua carga útil é entregue à aplicação que estiver associada à porta de destino do cabeçalho. Sem as portas, a camada de transporte não saberia o que fazer com o pacote que chega. Já a porta de origem tem

SOURCE PORT NUMBER (2 BYTES)	DESTINATION PORT NUMBER (2 BYTES)
LENGTH (2 BYTES)	CHECKSUM (2 BYTES)
PAYLOAD DATA (IF ANY)	

Figura 3: Cabeçalho do protocolo UDP.

Fonte: Adaptado de Fall e Stevens (2011)

sua principal utilidade quando é necessário o remetente receber uma resposta do destinatário.

O campo *Length* do UDP inclui o cabeçalho de 8 *bytes* e dados. Seu tamanho mínimo é 8 *bytes*, por ser o tamanho do cabeçalho, mas pode chegar até 65515 *bytes*, que é o tamanho da carga útil do IP. Já o campo de *Checksum* é opcional, e pode ser utilizado para aumentar a confiabilidade do datagrama transmitido. Caso a qualidade dos dados transmitidos não tenham tanta relevância, como na transmissão de voz digitalizada, o campo de *Checksum* pode ser desconsiderado (TANEMBAUM, 2011).

O UDP não realiza controle de fluxo, controle de congestionamento ou controle de entrega. Uma área que utiliza bastante UDP é a de aplicações multimídias em tempo real, e um exemplo de serviço que utiliza UDP é o DNS (*Domain Name System*) (TANEMBAUM, 2011).

4.3.2 TCP (*TRANSMISSION CONTROL PROTOCOL*)

O TCP é um protocolo que oferece um mecanismo confiável, orientado à conexão, de fluxo de dados, que tem como objetivo garantir a entrega dos dados e garantir que estes cheguem na mesma ordem de envio, com segurança, ao seu destinatário. O protocolo TCP, segundo Tanenbaum e Wetherall (2011), é o principal elemento da Internet. Esse protocolo permite que múltiplas aplicações em um mesmo *host* possam, simultaneamente, manter comunicação com seus *hosts* de destino (PETERSON; DAVIE, 2011).

A definição inicial do protocolo TCP foi na RFC (*Request for Comments*) 793 em

1981, mas passou por diversas melhorias e correções em diversas outras RFCs. A RFC 4614 foi publicada como um roteiro para os documentos de especificação do TCP (TANENBAUM; WETHERALL, 2011).

O protocolo de controle de transmissão é *full-duplex*, o que significa que cada conexão TCP suporta dois fluxos de dados, um em cada direção. Essa conexão é ponto a ponto, visto que todas conexões possuem apenas dois pontos terminais e não são permitidos processos *multicasting* ou *broadcasting* (TANENBAUM; WETHERALL, 2011). O TCP inclui um mecanismo de controle de fluxo que permite ao receptor determinar quanto dado o remetente pode transmitir a cada intervalo de tempo, e também um mecanismo de controle de congestionamento que tem o objetivo de determinar a quantidade de dados que podem ser enviados simultaneamente, com a função de não permitir uma sobrecarga no destinatário e também de não causar uma sobrecarga na rede (PETERSON; DAVIE, 2011).

Um *host* em uma conexão TCP possui um *buffer* que recebe dados e os armazena antes do envio, aguardando outros dados para enviar um fluxo maior, conforme suas necessidades. Contudo, há uma *flag PUSH* transportada no cabeçalho dos segmentos que pode forçar a entrega imediata. Outra *flag* do TCP é a *URGENT*, que serve para transmitir para a aplicação, com urgência, dados de alta prioridade, sem ficar acumulando mais dados para a transmissão de uma vez só (TANENBAUM; WETHERALL, 2011).

Os dados trocados entre os *hosts* transmissores e receptores do TCP são chamados segmentos. Cada segmento TCP possui um cabeçalho fixo de 20 *bytes* seguido por um número variável de *bytes* de dados. O próprio TCP define o tamanho dos segmentos, que não pode ser maior que 65515 *bytes*, que é o tamanho da carga útil do IP, e nem maior que a MTU (*Maximum Transfer Unit*).

Embora exista um tamanho máximo para os segmentos a serem transferidos, os pacotes IP que transportam os segmentos TCP podem ser fragmentados em algum enlace da rede que possivelmente a MTU seja menor do que o tamanho do pacote. Atualmente, as aplicações utilizam uma técnica de descoberta da MTU do caminho. A técnica utiliza mensagens de erro ICMP para encontrar a menor MTU no caminho e então reduz o tamanho do pacote para o tamanho da menor MTU encontrada, de forma que não haja fragmentação e nem perda de desempenho. A Figura 4 permite que seja visualizado o cabeçalho TCP.

Assim como no cabeçalho UDP, o cabeçalho TCP também inicia com as informações da origem (*Source Port*) e do destino (*Destination Port*) do segmento.

O campo *Sequence Number* tem a função de identificar a posição do segmento enviado

SOURCE PORT (16 BITS)				DESTINATION PORT (16 BITS)							
SEQUENCE NUMBER (32 BITS)											
ACKNOWLEDGMENT NUMBER (32 BITS)											
HEADER LENGTH (4 BITS)	RESV (0) (4 BITS)	C W R	E R E	U R G	A C K	P S H	R S T	S Y N	F I N	WINDOW SIZE (16 BITS)	
TCP CHECKSUM (16 BITS)						URGENT POINTER (16 BITS)					
OPTIONS (VARIABLE)											

Figura 4: Cabeçalho do protocolo TCP.

Fonte: Adaptado de Fall e Stevens (2011)

em meio ao fluxo de *bytes* que segue na mesma direção de transmissão do segmento e possui 32 *bits*. Já o campo *Acknowledgement Number* define o próximo *bit* esperado na transmissão e também é um campo de 32 *bits*.

O campo *Header Length* possui a informação do número de palavras de 32 *bits* presentes no cabeçalho, já que essa informação é variável pois o campo *Options* é variável.

O próximo campo é reservado com tamanho de 4 *bits* que não é utilizado. Inicialmente ele possuía 6 *bits*, mas 2 *bits* foram reivindicados para uso (TANENBAUM; WETHERALL, 2011).

Após esse campo reservado há oito campos de 1 *bit*, que são as *flags* que podem ser utilizadas para passar informações adicionais. Conforme Tanenbaum e Wetherall (2011), as *flags* CWR (*Congestion Window Reduced*) e ECE (*ECN-Echo*) são utilizadas para sinalizar congestionamento (ECN - *Explicit Congestion Notification*). Quando o receptor detectar congestionamento na rede, a *flag* ECE solicita uma redução de velocidade ao transmissor. Quando o transmissor reduzir a velocidade, a *flag* CWR sinaliza o receptor que a velocidade foi reduzida e que não precisa mais enviar mensagens de congestionamento. A *flag* URG serve para indicar urgência nesse fluxo de dados, e é identificada como ativa quando está com o valor atribuído 1. Para indicar que o número de confirmação de um pacote é válido, a *flag* ACK deve estar com o valor 1 atribuído a ela, o que é o mais comum. Quando os dados precisam chegar até a aplicação logo que recebidos sem ficarem parados no *buffer* deve-se utilizar o *bit* 1 na *flag* PSH. A *flag* RST tem a função de reiniciar a conexão por qualquer motivo que possa ter atrapalhado a conexão ou para rejeitar um segmento inválido ou para recusar uma tentativa de conexão de outro *host*. Para iniciar uma conexão deve-se enviar um pacote com a *flag* SYN com valor 1 e a *flag* ACK com valor 0. Para aceitar uma conexão deve-se responder SYN com valor 1 e ACK com valor também 1. Por último, a *flag* FIN tem a função de encerrar a conexão. Uma conexão

só é encerrada quando os dois *hosts* que estão se comunicando não possuem mais dados para transmitir e os dois enviarem a *flag* FIN.

Há outro campo que informa quantos *bytes* podem ser enviados, que é o tamanho de janela (*window size*). Caso seja 0 indica que o emissor não deve mais enviar dados para o destinatário.

No TCP também acontece a verificação de integridade do cabeçalho e da carga útil, assim como no UDP, e o campo utilizado para isso é o *Checksum*. Em seguida há um campo *Options*, que tem a utilidade de oferecer recursos extras não previstos e possui tamanho variável. Ao final do cabeçalho seguem os dados enviados no segmento, que são opcionais e que podem variar para cada transmissão de segmento (TANENBAUM; WETHERALL, 2011).

4.3.2.1 ESTABELECIMENTO DE CONEXÕES TCP

O estabelecimento de conexões TCP se dá por meio do processo conhecido como *handshake* triplo ou de três vias. A Figura 5 permite que seja visualizado como ocorre o *handshake* triplo. O *Client* manda uma primitiva *CONNECT* com a *flag* SYN ativa e com a *flag* ACK não ativa para um endereço e porta do *Server*, que deve possuir uma primitiva *LIS-TEN*. Então o *Server* envia uma primitiva *ACCEPT* com a *flag* SYN e a *flag* ACK ativas de volta para o *Client*. Por último, o *Client* devolve um ACK ativo para o *Server*, estabelecendo finalmente a conexão TCP e permitindo que inicie o envio de pacotes de dados entre os dois *hosts*.

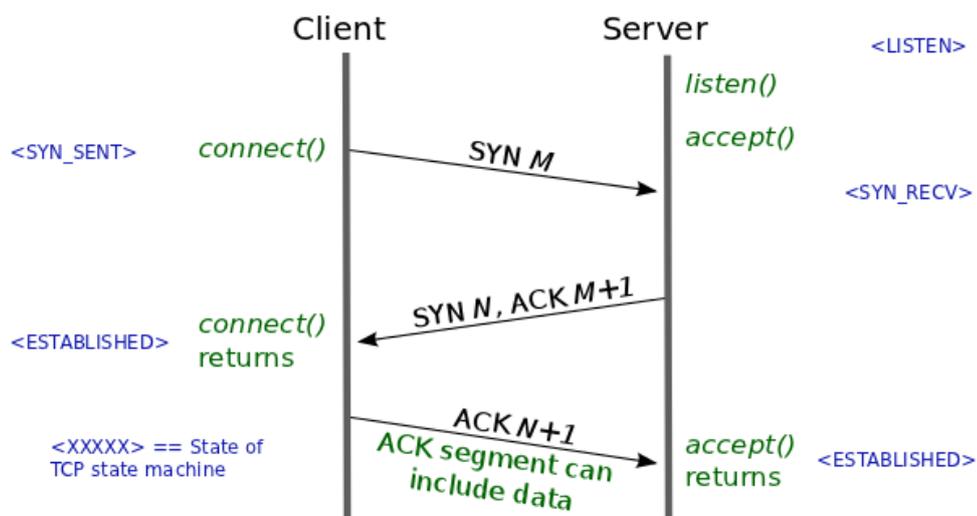


Figura 5: Processo *Handshake* Triplo para o estabelecimento de conexão TCP.

Fonte: Retirado de Kerrisk (2012)

Para finalizar uma conexão TCP ativa, o processo é parecido ao de abertura. Primeiramente o *Client* manda uma *flag* FIN ativa para o *Server*. O segundo deve então responder com uma *flag* ACK ativa, para confirmar a finalização daquele canal de transmissão. Contudo, como a comunicação é *full-duplex*, isso significa, ocorre em dois sentidos, o *Server* pode continuar transmitindo seus dados. Ao terminar a transmissão de seu fluxo de dados, o *Server* envia então uma *flag* FIN ativa para o *Client*, que responde com uma *flag* ACK ativa, confirmando a finalização da conexão. A partir desse momento os canais estão liberados para iniciarem outras conexões.

4.4 RTP (*REAL-TIME TRANSPORT PROTOCOL*)

Pelo menos desde 1981 iniciaram-se experimentos para analisar o transporte de tráfego na rede em tempo real, tais como amostras de voz digitalizada sobre redes de comutação de pacotes (PETERSON; DAVIE, 2011).

Assim como amostras de voz digitalizada, aplicações de tempo real são aplicações que requerem entrega praticamente imediata da informação, em tempo hábil para a comunicação. VoIP (*Voice over Internet Protocol*) é um exemplo clássico de aplicações de tempo real, pois é extremamente difícil que se estabeleça uma comunicação entre duas pessoas se houver grande atraso na entrega da voz nas extremidades da rede (PETERSON; DAVIE, 2011).

O protocolo RTP é descrito na RFC 1889 e possui como função básica multiplexar fluxos de dados em tempo real sobre um único fluxo de datagramas UDP (TANENBAUM; WETHERALL, 2011). A RFC 3550 (SCHULZRINNE et al., 2003) define o RTP e o RTCP (*Real-time Control Protocol*), que é um protocolo diretamente ligado ao RTP. Ambos foram criados pelo IETF (*Internet Engineering Task Force*). O RTP é utilizado para a troca de dados multimídia, enquanto o RTCP é utilizado para enviar periodicamente informações de controle associadas a um determinado fluxo de dados (PETERSON; DAVIE, 2011).

O protocolo RTP é encapsulado por protocolos de camadas baixas, mais comumente pelo UDP. Ele é um protocolo de transmissão que executa sobre outro protocolo de transmissão (PETERSON; DAVIE, 2011).

Para que uma amostra de áudio possa ser transmitida, por exemplo, primeiramente o RTP deve encapsular a amostra, em seguida o RTP é encapsulado pelo UDP, que por último é encapsulado pelo IP, para então ser transmitido. Os pacotes RTP utilizam apenas UDP, sem que haja nenhuma forma de tratamento especial no roteamento de seus pacotes, de forma que o RTP não garante a entrega temporizada dos seus pacotes. Isso torna fundamental a utilização

de mecanismos de QoS (*Quality of Service*) nos casos de necessidade de melhoria na qualidade da aplicação em tempo real (COMER, 2015).

4.5 RTCP (*REAL-TIME CONTROL PROTOCOL*)

O protocolo RTCP é definido junto com o protocolo RTP na RFC 3550 (SCHULZ-RINNE et al., 2003) e fornece um fluxo de controle que é associado a um fluxo de dados para uma aplicação multimídia (TANENBAUM; WETHERALL, 2011). Ele é responsável pelo *feedback* do desempenho da aplicação e da rede, por relacionar e sincronizar diferentes fluxos de mídias que saem do mesmo remetente e também por transmitir a identidade do remetente para exibição em uma interface do usuário (PETERSON; DAVIE, 2011).

O RTCP possui 5 tipos de mensagens, que servem para o controle da transmissão entre os *hosts*. O primeiro tipo de mensagem é o relatório do emissor que fornece registros de data/hora para que o receptor consiga executar dois fluxos simultaneamente, sincronizando os fluxos por meio do tempo. O segundo tipo de mensagem é o relatório do receptor, que possui a função de informar ao emissor sobre as condições da recepção, como o nível de perda de pacotes e o *jitter* (variação do atraso) entre as chegadas de relatórios na origem. O terceiro tipo de mensagem é destinado à leitura por humanos e ele transmite informações gerais sobre o usuário que controla a origem. O quarto tipo de mensagem é uma mensagem de despedida enviada quando encerra-se um fluxo. Por último, o quinto tipo de mensagem serve para especificar um novo tipo de mensagem.

Os *feedbacks* do RTCP são utilizados para parametrizar o atraso, o *jitter*, a largura da banda e o congestionamento da rede para a origem, o que permite o aumento da taxa de dados quando a rede estiver com um bom tráfego ou uma redução quando a rede estiver com um tráfego ruim, melhorando a qualidade do transporte.

Como os fluxos de diferentes mídias podem possuir *clock* diferentes, o RTCP também lida com a sincronização entre esses fluxos, melhorando a entrega dos pacotes e, conseqüentemente, a comunicação (PETERSON; DAVIE, 2011).

4.6 VOIP (*VOICE OVER INTERNET PROTOCOL*)

A tecnologia VoIP foi inventada em 1995 pela empresa Israelense Vocaltec (magicJack VocalTec Ltd., 2016). VoIP é uma tecnologia que possibilita o estabelecimento de chamadas e o transporte de voz digitalizada utilizando a rede IP (BORDIM, 2010).

Segundo Bordim (2010), a tecnologia VoIP traz diversas vantagens à telefonia, como a redução dos custos em ligações, visto que pode-se evitar o uso de operadoras telefônicas tradicionais. Outra vantagem é a simplificação da estrutura, pois os serviços de comunicação como telefonia, fax e internet ficam centralizados na mesma estrutura. Além disso, o VoIP é portátil, uma vez que é possível utilizar seu ramal a partir de qualquer lugar por meio de um *Softphone*, desde que haja conexão à Internet, e ainda permite a integração entre centrais telefônicas distintas.

Mesmo com inúmeras vantagens, a tecnologia VoIP também impõe algumas barreiras e dificuldades. Dentre suas desvantagens estão o alto custo inicial para a estruturação de redes totalmente IP, incluindo o custo de modernização e aquisição de nova aparelhagem, a falta de mão de obra especializada no mercado e também problemas relacionados à qualidade da ligação, como o atraso para entrega dos pacotes ou o excesso de perdas dos mesmos.

Outro elemento que está associado diretamente com as ligações de VoIP é a qualidade de serviço (QoS - *Quality of Service*). Um grande número de fatores influenciam na obtenção de uma alta qualidade em ligações VoIP. Esses fatores incluem o *codec* (acrônimo para codificador/decodificador) de áudio, o empacotamento, a perda de pacotes, o atraso, a variação do atraso (*jitter*) e a arquitetura da rede para fornecer QoS. Outros fatores envolvidos para a realização de uma ligação VoIP de sucesso incluem o protocolo de sinalização, controle de admissão de chamadas, preocupações com segurança e questões de NAT (*Network Address Translate*) e *firewall* (GOODE, 2002).

Para que haja boa qualidade nas ligações VoIP, a rede deve estar superdimensionada, isso significa que deve haver mais banda disponível do que será consumida, sem a ocorrência de congestionamento de pacotes. Caso não exista sobra da banda na rede, algumas práticas de QoS podem ser tomadas para a melhoria das ligações, como dar preferência na rede aos pacotes de voz, o que permite reduzir o atraso e as perdas desses pacotes até chegar ao destinatário.

Conforme Bordim (2010), a comunicação por meio de VoIP pode ocorrer de três maneiras diferentes, que são:

Softphone: é um software que emula o funcionamento de telefones convencionais. Há disponível no mercado versões pagas e versões gratuitas de *softphones* e todos possuem a mesma função, contudo podem haver algumas diferenças, como os *codecs* que são suportados.

ATA (Analog Telephone Adapter): é um equipamento que permite conectar aparelhos telefônicos analógicos à rede IP. O equipamento ATA deve estar configurado com a identificação do ramal que está conectado a ele. Há no mercado ATAs que suportam de 2 até 30 aparelhos

telefônicos e são uma boa alternativa para continuar utilizando os aparelhos telefônicos já existentes em uma infraestrutura convencional que pretende migrar para telefonia IP.

Telefone IP: é um equipamento diferente dos telefones convencionais. Este equipamento possui o hardware necessário para ser conectado na rede de computadores. Há diferentes modelos de telefones IP no mercado e grande variação de preços. Alguns aparelhos possuem visor LCD, conexão *Ethernet Gigabit* ou conexão *Fast Ethernet*, suporte a *codec* de áudio pago, teclas programáveis, dentre outras funções. Esse tipo de equipamento possui um custo elevado se comparado aos aparelhos tradicionais e é um dos itens que mais eleva o custo inicial de implantação de telefonia VoIP.

4.7 CODECS

Para que seja possível enviar voz ou vídeo em redes de comutação de pacotes é necessário converter o sinal analógico em sinal digital. Os *codecs* (acrônimo para codificador/decodificador) possuem essa função de codificar e decodificar sinais e as técnicas mais comuns, segundo Bordim (2010), são as de codificação de forma de onda (*waveform coder*), paramétricas e híbridas.

O codificador de forma de onda mede a amplitude do sinal de entrada em intervalos regulares e converte essas amostras em valores digitais. Para retornar um sinal digital para sinal analógico, as amostras digitais devem passar por um decodificador, que possui a função de recriar um sinal analógico similar ao original, de acordo com a taxa de amostras utilizada pelo codificador (COMER, 2015).

O número de amostras realizadas no momento da codificação deve prezar pela qualidade da reprodução e também pelo tamanho da representação digital, visto que um fator influencia diretamente sobre o outro. Quanto mais amostras, melhor será a qualidade do sinal gerado, porém maior será o tamanho do arquivo de saída.

Segundo Comer (2015), há três maneiras diferentes de reduzir a quantidade de dados gerados em uma codificação digital: realizar uma menor amostragem por segundo; utilizar menos *bits* para codificar cada amostra; ou utilizar um esquema de compactação digital. O problema é que as duas primeiras formas reduzem a qualidade do áudio, e a compactação pode causar retardo na entrega, pois a saída digitalizada precisa ser mantida durante a compactação.

4.7.1 CODEC G.711

O *codec* G.711 é um modulador por códigos de pulso (PCM) de frequências de voz e foi criado pelo ITU (*International Telecommunication Union*) em 1972. Ele possui uma taxa recomendada de amostragem de 8000 amostras de áudio por segundo, com uma tolerância de aproximadamente 50 partes por milhão (ppm) (ITU-T, 1988).

Cada uma das 8000 amostras de áudio é codificada por uma sequência de 8 *bits*, o que faz com que a taxa de saída deste codificador seja de 64 kbit/s, desconsiderando qualquer *overhead* (processamento ou armazenamento em excesso) (SOUZA, 2006).

O *codec* G.711 é um dos codificadores mais utilizados por ser livre de licença e por não comprimir o sinal, o que permite seu uso em transmissões de fax. Por causa da alta taxa de amostragem, o sinal pode ser restaurado perfeitamente após a decodificação. Ele deve ser utilizado em redes que possuam boa largura de banda disponível, já que utiliza uma alta taxa de transmissão de *bits* (BORDIM, 2010).

4.7.2 CODEC G.729

O *codec* G.729 é um codificador de áudio desenvolvido pelo ITU, inicialmente para redes de comutação de circuito, que necessita a compra de licença para uso (BORDIM, 2010). Ele utiliza um algoritmo de codificação eficiente na compressão e na redução do consumo de banda, capaz de comprimir a carga útil em 8 kbit/s, 8 vezes menos que o G.711 (ITU-T, 2012).

O algoritmo utilizado pelo G.729 é o CS-ACELP (*Conjugate Structure Algebraic Code Excited Linear Prediction*), que é um codificador híbrido baseado no modelo de codificação CELP (*Code Excited Linear Prediction*). Esse algoritmo opera com o sinal de voz de entrada já convertido para o formato PCM e trabalha com quadros de 10 ms (ou 80 amostras) que são divididos novamente em dois quadros de 5 ms (ou 40 amostras). Os quadros de 10 ms são analisados e extraídos os parâmetros CELP. Os parâmetros de excitação de voz são determinados nos quadros de 5 ms, codificados e transmitidos (BORDIM, 2010).

Devido a esse processo que o áudio passa, o sinal gerado e transmitido possui uma latência maior que o G.711, além de possuir uma qualidade inferior, visto que há compressão do áudio. A grande diferença está na taxa de transmissão, pois o G.729 utiliza apenas 8 kbit/s enquanto o G.711 utiliza 64 kbit/s (SOUZA, 2006).

4.7.3 CODEC G.723.1

O *codec* G.723.1 também é um padrão criado pelo ITU e possui taxas de transmissão de 6,3 e 5,3 kbit/s (SOUZA, 2006). Ele codifica sinais de áudio em quadros de 30 ms e possui um atraso adicional de 7,5 ms, totalizando 37,5 ms (ITU-T, 2006).

O *codec* G.723.1 possui altas taxas de compressão, maiores que a do G.729, o que faz dele o *codec* com menor taxa de transmissão, mas também possui a maior latência por causa do seu processo de compressão, o que dificulta seu uso em qualquer aplicação sensível a atraso.

A Tabela 2 permite visualizar um comparativo entre os três *codecs* citados.

Tabela 2: Comparação entre os *codecs* G.711, G.729 e G.723.1

<i>Codec</i>	G.723.1	G.729	G.711
Taxa de <i>bits</i> (kbit/s)	5,3/6,3	8	64
Tamanho do quadro (ms)	30	10	0,125
<i>Lookahead Delay</i> (ms)	7.5	5	Inexistente
Comprimento do Quadro (<i>bytes</i>)	20/24	10	1

Fonte: Retirado de Souza (2006)

4.8 H.323

Criado pelo ITU, o H.323 é um padrão de sinalização que permite a transmissão de pacotes de voz em redes IP. O H.323 é um protocolo e define como unir outros protocolos para estabelecer um sistema de comunicação IP (PETERSON; DAVIE, 2011).

O H.323 fornece protocolos que especificam a forma que os participantes de uma comunicação devem transferir e compartilhar dados. A Tabela 3 mostra os quatro protocolos básicos utilizados pelo H.323 para telefonia IP:

Tabela 3: Protocolos Utilizados pelo H.323

PROTOCOLO	FUNÇÃO
H.225.0	Sinalização usada para estabelecer uma chamada
H.245	Controle e <i>feedback</i> durante uma chamada
RTP	Transferência de dados em tempo real
T.120	Troca de dados associados a uma chamada

Fonte: Modificado de Comer (2015)

4.9 SIP (*SESSION INITIATION PROTOCOL*)

O protocolo de inicialização de sessão (SIP) é um protocolo da camada de aplicação, baseado em um modelo requisição/resposta (PETERSON; DAVIE, 2011). O SIP foi criado pelo IETF como uma alternativa ao H.323, contudo ele serve apenas para sinalização, não realizando tudo que o H.323 proporciona (COMER, 2015).

Para poder fornecer informações sobre uma chamada, o SIP se baseia no protocolo SDP (*Session Description Protocol*) que é fundamental para a criação de áudioconferências, pois permite a entrada e saída de participantes de forma dinâmica (PETERSON; DAVIE, 2011).

Conforme Comer (2015), o SIP utiliza a arquitetura cliente-servidor com dois tipos de servidores: o Servidor Agente de Usuário, que é um servidor executado em um telefone SIP; e o Servidor Intermediário, que é um servidor que pode funcionar como um *proxy* ou como um servidor de redirecionamento.

O protocolo SIP fornece algumas funcionalidades que, conforme Peterson e Davie (2011), podem ser agrupadas em cinco categorias:

Localização do Usuário: determina o dispositivo correto com o qual se comunicar para chegar a um determinado usuário;

Disponibilidade do Usuário: determina se o usuário está apto ou disponível para uma seção de comunicação particular;

Capacidade do Usuário: determina alguns itens como a escolha da mídia e o esquema de comunicação a ser utilizado;

Configuração de Seção: estabelece os parâmetros da seção como os números de portas que serão utilizados na comunicação;

Gerenciamento de Seção: um intervalo de funções que inclui transferência de seção e modificação de parâmetros da seção.

Um *proxy* SIP pode ser imaginado como um ponto de contato para um usuário para o qual são enviados as requisições iniciais para comunicação. Cada usuário possui um nome no formato *nome@domínio*. Para iniciar uma seção SIP um usuário inicia enviando sua mensagem SIP para o *proxy* local desse domínio. Dentre outras coisas, a mensagem inicial conta com uma URI (*Uniform Resource Identifier*) SIP que fornece a identificação completa de um usuário, mas não fornece sua localização.

Um usuário pode se registrar com um serviço de localização enviando uma mensagem

de registro SIP ao “*registrar*” para seu domínio. Essa mensagem cria uma ligação entre o endereço de registro e o endereço do contato. A Figura 6 mostra o processo de *Registration* (registro), em que é possível notar o envio da mensagem de solicitação de registro e a resposta 200, confirmando o registro e também permite a visualização do estabelecimento e finalização de uma seção SIP.

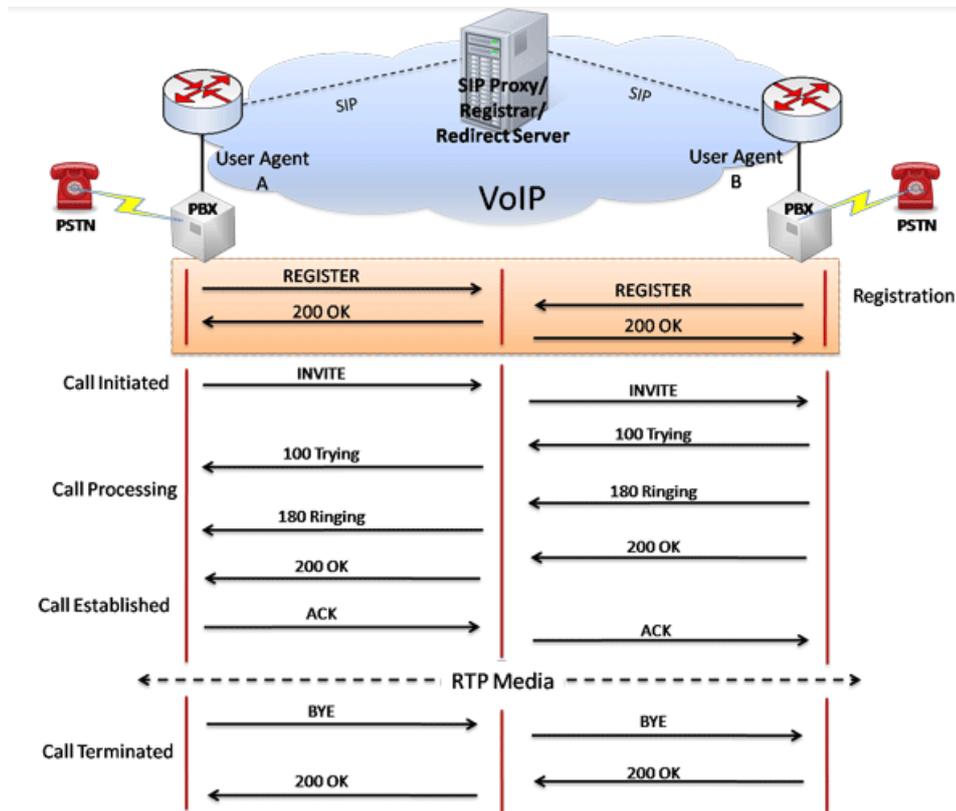


Figura 6: Fluxo básico de uma seção SIP.

Fonte: Retirado de GL Communications Inc. (2014)

O fluxo básico de uma seção SIP, como mostra a Figura 6, inicia com o envio de uma mensagem *Invite* para o *proxy*, que a repassa para o destinatário. Se não houver erro na mensagem, a primeira resposta é *100 trying* (tentando). Uma vez que o convite é entregue sem erros ao destinatário, este responde a mensagem *180 ringing* (tocando), que significa que pode ter sido gerado um toque no telefone. Considerando que o destinatário esteja livre para atender a chamada, então ele atende e é enviada a mensagem *200 OK*. Quem iniciou a chamada responde com um *ACK* e então a chamada é estabelecida, o que torna possível o envio de mídia (fluxo de áudio RTP encapsulado) entre as duas partes. Ao final da conversação, qualquer uma das partes pode desligar seu dispositivo e então é enviada uma mensagem *BYE* e o outro lado responde com um *200 OK* novamente, encerrando definitivamente a seção SIP (PETERSON; DAVIE, 2011).

4.10 IAX (*INTER-ASTERISK EXCHANGE*)

O IAX é um protocolo aberto desenvolvido pela Digium (BORDIM, 2010), com a função de proporcionar comunicação entre servidores Asterisk. O Asterisk é abordado na próxima seção. A versão 2 do protocolo (IAX2) foi definida na RFC 5456 e, de acordo com a RFC, o protocolo não é um padrão do IETF (*Internet Engineering Task Force*).

O protocolo IAX utiliza a porta UDP 4569 para sinalização e também para o transporte de voz. Ele utiliza um mecanismo particular para transporte e controle do canal de comunicação de áudio e vídeo, o que torna desnecessário o uso do RTP para transporte de mídia.

Conforme Bordim (2010), o IAX busca minimizar o uso de banda em chamadas entre servidores Asterisk, pois utiliza o mesmo cabeçalho IP e UDP para transmitir várias chamadas utilizando apenas um pacote, o que causa um acréscimo de apenas 4 bytes por chamada. Ele utiliza uma porta fixa (UDP 4569) para transmissão de todas as chamadas entre dois servidores. Adicionalmente, o IAX suporta três formas de autenticação de usuários, que são por texto puro, *hashing* MD5 ou com criptografia RSA.

Além disso, o IAX provê maior facilidade de uso na presença de *firewalls* do que o SIP e o H.323, já que não precisa realizar alocação dinâmica de canais de mídia quando a aplicação negocia novas portas para ligações estabelecidas, e torna possível a transmissão de informações sobre o plano de discagem (BORDIM, 2010).

4.11 ASTERISK

No ano de 1999, Mark Spencer, proprietário da *Linux Suport Services*, sentiu a necessidade de um sistema telefônico funcional, que fosse ao mesmo tempo robusto e que reduzisse custos quando comparado aos serviços que eram oferecidos no mercado na época. Por não encontrar uma solução, Mark decidiu criar seu próprio sistema telefônico e começou a codificá-lo. Em poucos meses ele já havia escrito o código original do núcleo do Asterisk e publicado de forma *open source* na Internet, disponibilizando-o sob os termos da licença GPL.

Pouco tempo após a disponibilização do código, centenas de desenvolvedores que gostaram da ideia de existir um PBX (*Private Branch Exchange*) *open source*, ou mesmo desenvolvedores que ajudavam em outros projetos similares que nunca foram bem aceitos e disseminados, começaram a enviar o código de novas características e funcionalidades para o Asterisk. Atualmente o Asterisk conta com milhares de colaboradores desenvolvedores, entre ajudantes individuais e empresas.

No ano de 2001, a empresa *Linux Suport Services* trocou seu nome para Digium e continuou a desenvolver o Asterisk em colaboração com sua comunidade open source. A Digium vem oferecendo serviços e suporte à comunidade de desenvolvimento, bem como vem construindo produtos e serviços para venda em torno do Asterisk. Com isso, a empresa Digium e o software Asterisk cresceram muito e buscam até hoje atender a todas as necessidades apresentadas pelos usuários, possuindo um vasto leque de ferramentas para comunicação (DAVENPORT, 2010d).

O Asterisk, conforme definição de sua própria documentação, é um canivete suíço da telefonia (DAVENPORT, 2010b). Surgiu como um PBX *open source* para pequenas empresas, que era o caso da *Linux Suport Services*. Após seu crescimento, ele tornou-se uma ferramenta para construção de aplicações de comunicação, fornecendo funções além de um PBX convencional, como *gateways* VoIP, sistemas de *call center*, pontes de conferências, servidores *voicemail* e diversos outros tipos de aplicações que envolvam comunicação em tempo real.

O Asterisk é um servidor de comunicação capaz de manipular detalhes de baixo nível de transferência de dados, por meio de diversos protocolos de comunicação, como o RTP e o RTCP. Além disso, o Asterisk permite ao desenvolvedor a opção de criar suas próprias aplicações de comunicação (DAVENPORT, 2010b).

4.11.1 VERSÕES

No ano de 2005 foi lançada a versão 1.2 do Asterisk e a partir daí foram lançadas novas versões praticamente todos os anos, mantendo sempre um período de suporte a problemas e atualizações de segurança.

Cada nova versão lançada é definida com um tipo que pode ser *Standard* ou LTS (*Long Time Support*). As versões *Standard* possuem garantia de suporte completo por um ano e mais um ano de garantia de atualizações de segurança. Já as versões LTS possuem garantia de suporte completo por quatro anos e mais um ano com garantia de atualizações de segurança. A versão LTS mais atual do Asterisk é a 13, lançada em 2014, com previsão de fim de vida em outubro de 2019.

A Tabela 4, que está presente na documentação do Asterisk, mostra as versões lançadas a partir de 2005 e também as previsões de lançamento de novas versões até o ano de 2017. Também mostra o tipo de versão, a data do lançamento, a data em que a versão passou a ter apenas atualizações de segurança e a data de fim da vida do produto (*End of Life* - EOL).

Para o desenvolvimento de aplicações para empresas costuma-se utilizar sempre a úl-

Tabela 4: Tabela de versões do Asterisk, lançadas e previsões

Release Series	Release Type	Release Date	Security Fix Only	EOL
1.2.X		21/11/2005	07/08/2007	21/11/2010
1.4.X	LTS	23/12/2006	21/04/2011	21/04/2012
1.6.0.X	Standard	01/10/2008	01/05/2010	01/10/2010
1.6.1.x	Standard	27/04/2009	01/05/2010	27/04/2011
1.6.2.x	Standard	18/12/2009	21/04/2011	21/04/2012
1.8.x	LTS	21/10/2010	21/10/2014	21/10/2015
10.x	Standard	15/12/2011	15/12/2012	15/12/2013
11.x	LTS	25/10/2012	25/10/2016	25/10/2017
12.x	Standard	20/12/2013	20/12/2014	20/12/2015
13.x	LTS	24/10/2014	24/10/2018	24/10/2019
14.x	Standard	26/09/2016	26/09/2017	26/09/2018
15.x	LTS	2017-10 (tentative)	2021-10 (tentative)	2022-10 (tentative)

Fonte: Retirado de Bryant (2010)

tima versão LTS disponível, uma vez que possui o maior tempo possível de atualizações de segurança e não haver a necessidade de troca do servidor em períodos curtos. Uma possível desvantagem das versões LTS é que elas podem contar com menos funcionalidades implementadas (BRYANT, 2010).

4.11.2 ARQUITETURA DO ASTERISK

A Figura 7 mostra um diagrama que busca representar os principais relacionamentos da maioria dos componentes do Asterisk entre si e também com outras entidades externas ao Asterisk.

O Asterisk possui um núcleo que pode interagir com diversos módulos. Ele é o componente essencial para o funcionamento do PBX, que fornece a infraestrutura necessária para o funcionamento de diversas funções, como a leitura dos arquivos de configuração e o carregamento de todos outros módulos distintos que desempenham alguma funcionalidade.

Dentre os diferentes módulos, há os chamados *Channel Drivers* que fornecem canais que acompanham o *dialplan* (plano de discagem) do Asterisk para executar o comportamento programado e facilitar a comunicação entre dispositivos ou programas externos do Asterisk. Para se comunicar com outros canais, os *Channels* geralmente utilizam a infraestrutura de ponte.

Os módulos são em sua maioria distribuídos pelo próprio Asterisk, mas também podem ser distribuídos por membros da comunidade ou outras empresas que fazem módulos comerciais. A instalação dos módulos distribuídos pelo Asterisk é opcional, visto que o Asterisk

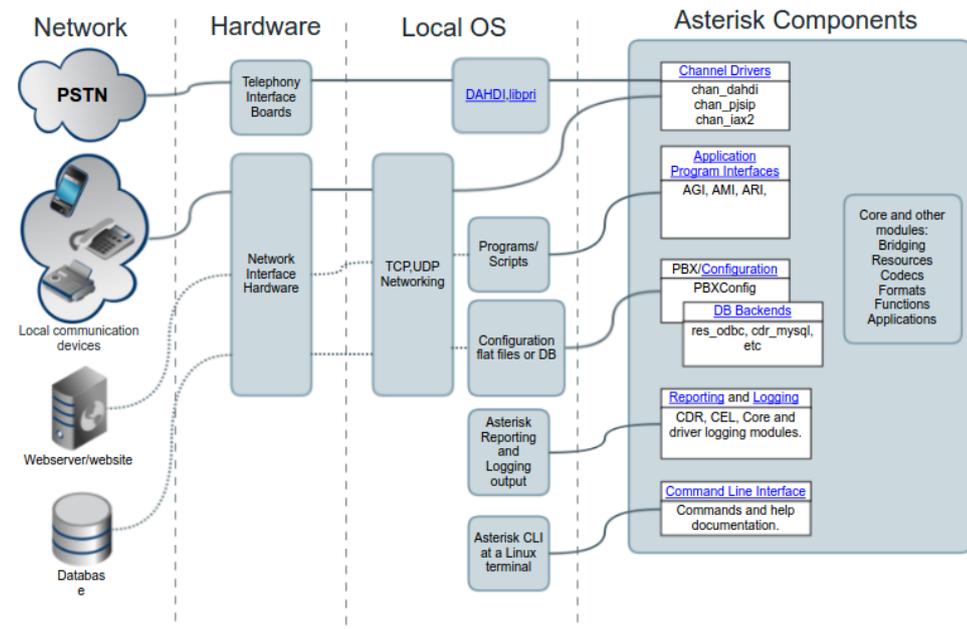


Figura 7: Relacionamentos entre os principais componentes do Asterisk e componentes externos.

Fonte: Retirado de Davenport (2010a)

permite ao desenvolvedor escolher se quer ou não instalar os módulos, durante a criação do servidor. Além disso, após instalados cada módulo pode ser configurado de forma independente em seus arquivos de configuração.

Geralmente os arquivos de configuração dos módulos possuem a extensão `.so` e ficam no diretório `/usr/lib/asterisk/modules`. Esses arquivos são carregados e suas funcionalidades adicionadas ao sistema sempre que o Asterisk é iniciado. É possível visualizar todos os módulos instalados digitando `module show` na CLI (*Command Line Interface*) do Asterisk (DAVENPORT, 2010a). A Tabela 5 mostra os diferentes tipos de Módulos do Asterisk e uma breve descrição de suas funcionalidades.

Todas as configurações do Asterisk são salvas em arquivos de configuração em texto puro, que podem ser editadas em qualquer editor de texto comum. Esses arquivos podem ser divididos em várias *sections* (seções). Os nomes de seções ficam entre colchetes e não podem conter espaços, além de diferenciarem letras maiúsculas e minúsculas. Dentro dessas seções são definidos os valores das configurações do arquivo (DAVENPORT, 2010g).

4.11.3 OBJETOS

Dentro dos arquivos de configuração do Asterisk também são criados objetos, os quais possuem uma sintaxe que difere da sintaxe das configurações. Para criar um objeto deve-se

Tabela 5: Tabela de Módulos do Asterisk

Tipo de Módulo	Descrição
Channel Drivers	Comunicar com dispositivos externos do Asterisk e traduzir a sinalização específica ou o protocolo ao núcleo.
Dialplan Applications	Aplicações fornecem a função de chamadas ao sistema. Elas podem atender chamadas, tocar música de espera, desligar chamadas, dentre outras funções.
Dialplan Functions	Funções são usadas para recuperar, definir ou manipular várias configurações em uma chamada.
Resources	Fornecer recursos ao Asterisk e seus módulos. Música em espera e estacionamento de chamadas são exemplos de recursos.
CODECs	Módulos para codificação e decodificação de áudio e vídeo.
File Format Drivers	Drivers para salvar e converter mídias em formatos especificados.
Call Detail Record Drivers (CDR)	Módulo para escrita de logs de chamadas no disco ou em um banco de dados.
Call Event Log Drivers (CEL)	Similar ao CDR, contudo armazena mais detalhes do que acontece durante uma ligação.
Bridge Drivers	Drivers utilizados para criação de pontes entre outros módulos ou mesmo chamadas, na arquitetura do Asterisk

Fonte: Modificado de (DAVENPORT, 2010j)

especificar o tipo do objeto, inserir uma seta, esta formada pelos caracteres igual e maior que ($=>$), e então as configurações para o objeto. Para facilitar a criação dos objetos nos arquivos de configuração, o Asterisk passou a aceitar também apenas o símbolo de igual ($=$), com a mesma função da flecha (DAVENPORT, 2010f). Um exemplo de seção é mostrado na sequência:

```
[nome-secao]
objeto => configuracao
objeto = configuracao
```

4.11.4 COMENTÁRIOS

Por costume, assim como no desenvolvimento de sistemas, é comum o uso de comentários nos arquivos de configuração, que podem auxiliar na manutenção do arquivo e também para desabilitar temporariamente certas configurações. Os comentários podem ser de apenas uma linha, utilizando ponto e vírgula (;), ou blocos de várias linhas, demarcados com ponto e vírgula seguido de dois hifens no início (;-) e dois hifens seguidos de ponto e vírgula no final do bloco (-;) (DAVENPORT, 2010e). Abaixo um exemplo de comentário em linha e de bloco.

```
[nome-secao]
```

```

configuracao = true ; aqui configuração é verdadeira
;- este é um bloco de comentário capaz de comentar várias linhas,
se estendendo desde a marcação inicial até a marcação final -;

```

4.11.5 *TEMPLATES*

O Asterisk oferece *templates* prontos para a maioria dos arquivos de configuração. Um *template* é uma seção de um arquivo de configuração que é utilizada como base ao desenvolvedor para a criação de novas seções (DAVENPORT, 2010i).

Um *template* é definido em uma seção inserindo após a declaração da mesma um ponto de exclamação entre parênteses. Dessa forma, o Asterisk não realizará o carregamento dessa seção para o uso de suas configurações, ela servirá apenas de modelo para o desenvolvedor criar as demais (DAVENPORT, 2010h).

De acordo com a necessidade de utilizar um *template* em uma nova seção, o desenvolvedor pode colocar o nome do *template* entre parênteses após declarar a mesma, conforme exemplo abaixo.

```

[secao-template] (!) ; declaração do template
configuracao = valor
[secao-usa-template] (secao-template)
configuracao2 = valor2

```

Essa forma de uso de *templates* serve também para o uso de outras seções como modelo, mesmo que estas não sejam *templates*, basta colocar o nome da seção que deseja-se utilizar entre parênteses.

A nova seção criada irá receber como herança todos os valores e objetos definidos em *template* ou em outra seção que queira usar como base. Ainda, todos as configurações e objetos definidos na nova seção irão sobrescrever os herdados (DAVENPORT, 2010k).

4.11.6 BANCO DE DADOS ASTERISK

O Asterisk, quando instalado, já vem com um banco de dados integrado para uso interno. Ele é usado pelo Asterisk para o armazenamento de informações necessárias para a persistência entre *reloads/restarts*, quando vários módulos utilizam dele automaticamente.

Outro propósito desse banco de dados é que ele pode ser usado por desenvolvedores e administradores para o armazenamento de dados arbitrários, que pode ser feito utilizando funções e aplicações para isso (NEWTON, 2014a).

4.11.7 CONCEITOS CHAVE

O Asterisk possui alguns conceitos chave, que possuem grande importância para a criação de aplicações de comunicação. Abaixo estão listados os conceitos chaves de *Bridges*, *Channels* e *States and Presence*:

4.11.7.1 CHANNELS

Um *channel* (canal) age como um caminho da comunicação entre o Asterisk e outros dispositivos. Nada acontece no Asterisk sem que tenha envolvido ao menos um *channel*. Uma chamada, por exemplo, pode ser um ou mais *channels* criando um caminho de comunicação ou atividade por meio do Asterisk. Geralmente, os *channels* são utilizados para a passagem de áudio, mas também podem passar por eles outros tipos de dados, como mensagens de texto e vídeos. A partir do momento que um *channel* é estabelecido, os eventos que ocorrerem sobre ele são dependentes da sinalização e dos outros protocolos implementados nele por meio do *channel driver* (NEWTON, 2014b).

4.11.7.2 BRIDGES

Uma *bridge* (ponte) é uma construção que compartilha mídia com *channels*. Ela é a forma em que os caminhos de comunicação são compartilhados. As *bridges* são criadas quando o Asterisk sabe que dois ou mais *channels* querem se comunicar. Em uma ligação, por exemplo, os canais de quem faz a ligação e de quem recebe a ligação são conectados por meio de uma *bridge*. Quando a aplicação que criou uma *bridge* é encerrada, os *channels* deixam a *bridge* (DAVENPORT, 2010c).

4.11.7.3 STATES AND PRESENCE

O Asterisk é capaz de receber informações sobre o estado de dispositivos, *extensions* (extensões, que pode ser entendido como ramais) e também sobre os usuários dos dispositivos. Isso é possível por meio dos conceitos de *Device State* (estado do dispositivo), *Extension State* (estado da extensão) e *Presence State* (estado de presença, que é referente ao usuário) (NEWTON, 2014e).

4.11.8 ESTADOS DOS DISPOSITIVOS

Há no Asterisk diversos provedores de estado dos dispositivos, como o `chan_sip.so`, que fornece informações de estado de seus recursos. Na Tabela 6 estão todos os possíveis estados que um dispositivo pode ter.

Tabela 6: Tabela de Estados dos Dispositivos

ESTADO	DESCRIÇÃO
<i>UNKNOWN</i>	estado do dispositivo é desconhecido
<i>NOT_INUSE</i>	o dispositivo não está em uso
<i>INUSE</i>	o dispositivo está em uso
<i>BUSY</i>	o dispositivo está ocupado
<i>INVALID</i>	dispositivo inválido
<i>UNAVAILABLE</i>	dispositivo indisponível
<i>RINGING</i>	dispositivo está tocando
<i>RINGINUSE</i>	toque do dispositivo em uso
<i>ONHOLD</i>	dispositivo em espera

Fonte: Modificado de Newton (2014c)

O comando `devstate` da CLI permite listar ou modificar algum estado de dispositivo especificamente (NEWTON, 2014c).

4.11.9 ESTADOS DAS EXTENSÕES

O estado da extensão é o conjunto do estado do dispositivo a partir dos dispositivos mapeados para a extensão por meio de uma diretiva.

Para que o Asterisk seja capaz de fornecer o estado de uma extensão, é necessário definir primeiramente um *alias* (apelido) para esta extensão no arquivo de configuração do *dialplan* do Asterisk, o `extensions.conf`. Então, quando o Asterisk carrega o arquivo de configuração, ele cria *alias* na memória para cada *alias* definido no *dialplan*, o que permite que esses *alias* sejam consultados ou manipulados por funções e por comando da CLI. O estado de cada *alias* é atualizado constantemente baseado nas mudanças de estado do dispositivo mapeado para ele. O comando `core show hints` da CLI lista o estado da extensão para todos *alias* definidos, bem como uma lista do estado do dispositivo ou o estado de presença mapeados para cada *alias* (NEWTON, 2014d).

Os conceitos de estado do dispositivo e estado de presença são um pouco parecidos, contudo o estado de dispositivo reflete especificamente o aspecto físico de um dispositivo, indicando se ele está conectado ao Asterisk ou não.

O estado de presença foi inserido apenas a partir da versão 11 do Asterisk e reflete o estado do usuário de um dispositivo, o que pode ser diferente. Mesmo que um dispositivo esteja conectado e seu estado seja disponível, seu usuário pode estar distante do aparelho e seu estado ser indisponível.

O comando `presencestate` da CLI lista ou modifica algum estado de presença fornecido pelo provedor *CustomProvider*. A Tabela 7 permite visualizar todos os possíveis estados de presença:

Tabela 7: Tabela de Estados de Presença

<i>NOT_SET</i>	não há estado de presença para esta entidade
<i>UNAVAILABLE</i>	esta entidade está presente mas temporariamente indisponível para comunicação
<i>AVAILABLE</i>	esta entidade está disponível para comunicação
<i>AWAY</i>	esta entidade não está presente e está indisponível para comunicação
<i>XA</i>	no momento, esta entidade não está presente e não há expectativa de retorno
<i>CHAT</i>	esta entidade está disponível para comunicação, porém prefere utilizar mensagens instantâneas para se comunicar
<i>DND</i>	esta entidade não quer ser perturbada

Fonte: Modificado de Michelson (2012)

5 MATERIAIS E MÉTODOS

Essa seção dedica-se a mostrar os materiais utilizados para a realização do trabalho e a forma como foram aplicados, além do meio que foram realizados os estudos para comprovar ou não a viabilidade da implantação de um servidor VoIP com Asterisk na UTFPR de Toledo. O texto desse capítulo está dividido em seis seções, dispostas de forma que se cumprissem os objetivos propostos no trabalho, da seguinte forma: a seção 5.1 descreve o levantamento da infraestrutura de rede atual da UTFPR; a seção 5.2 mostra como foi realizada a pesquisa sobre as funcionalidades do servidor VoIP que deveriam ser implementadas; a seção 5.3 descreve o processo de implementação, com todas as tecnologias utilizadas para isso, e como ocorreram os testes das funcionalidades implementadas; a seção 5.4 descreve a forma de avaliação da qualidade dos serviços oferecidos pelo IPBX configurado; a seção 5.5 mostra a metodologia utilizada para aferição do uso da rede da UTFPR, e; a seção 5.6 descreve o levantamento da infraestrutura de rede necessária para a implantação do VoIP na instituição.

5.1 INFRAESTRUTURA ATUAL DA UTFPR

O câmpus de Toledo da Universidade Tecnológica Federal do Paraná conta atualmente com três blocos de salas de aula, além de um prédio do restaurante universitário e um prédio de biblioteca, laboratórios e ginásio. Quanto à Tecnologia da Informação, a instituição possui uma boa infraestrutura de rede, tanto por seu padrão de cabeamento, que é totalmente estruturado (utiliza fibra ótica para a interligação dos principais ativos da rede e UTP - *Unshielded Twisted Pair* - categoria 6 para a maioria das estações de trabalho), quanto por seus ativos de rede, que em sua maioria possuem conexão *gigabit* e são gerenciáveis.

Em toda a estrutura física da instituição o cabeamento é estruturado, o que garante padronização na conectorização e nos meios de transmissão da rede. Além disso, estão instalados e organizados na rede diversos equipamentos de rede, de modo que seja possível estabelecer comunicação entre todos os dispositivos finais e à Internet.

A universidade dispõe de cinco servidores para a disponibilização de todos os serviços

necessários aos funcionários e à comunidade acadêmica. Dois desses servidores são virtualizados, o que permite o compartilhamento de seus recursos entre diversas máquinas virtuais.

A rede conta ainda com um roteador e mais de 25 *switches* distribuídos entre o *data center*, os *racks* de cada prédio e também os *racks* dos laboratórios de informática. A grande maioria dos *switches* são *gigabit ethernet*, gerenciáveis e suportam VLANs (*Virtual Local Area Network*). A rede da UTFPR de Toledo está dividida logicamente em várias VLANs, que atendem ao setor administrativo, aos acadêmicos, laboratórios de informática e câmeras de monitoramento, dentre outros usuários e serviços.

A Figura 8 mostra de forma simplificada a topologia da rede física da universidade. É possível visualizar a distribuição dos *switches* principais e secundários em cada bloco, bem como a comunicação com os pontos de acesso (*Access Points*) e com os demais dispositivos finais da rede, além da interligação com o *data center*, que é onde ocorre a comunicação com os servidores, com o roteador e também a saída para a WAN (*Wide Area Network*).

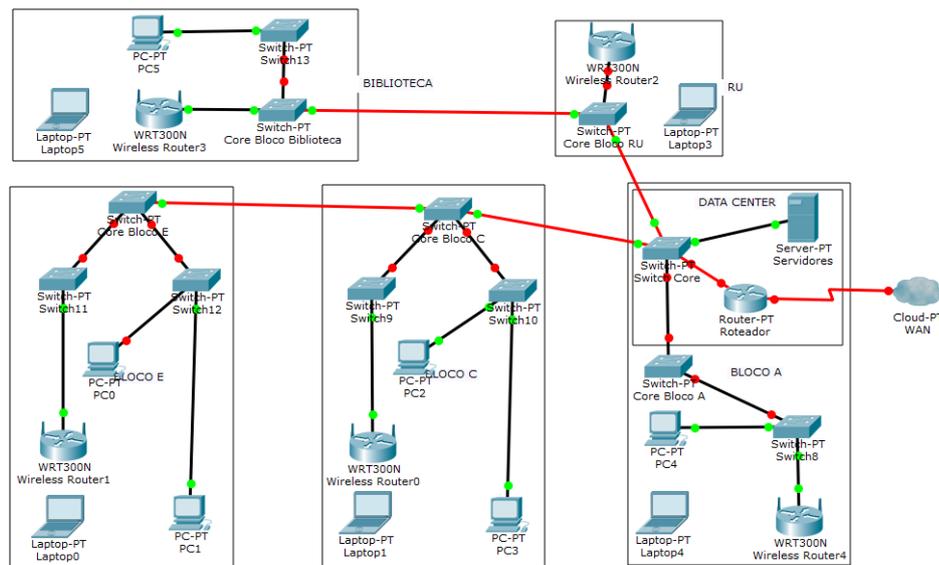


Figura 8: Atual Topologia de Rede Física da UTFPR - Toledo

Fonte: Conteúdo próprio, gerado com o *software* de ensino *CISCO Packet Tracer*.

Devido ao cabeamento estruturado, a telefonia do câmpus pode utilizar da mesma estrutura para a ligação dos aparelhos. A Figura 9 mostra como é feita essa distribuição dos ramais.

No *data center* há a central telefônica *SIEMENS* e a distribuição dos seus ramais em dois *patch panels* (PP). A partir desses dois, os ramais são distribuídos entre os blocos pelos *voice panels* de interligação dos blocos. No *rack* do *data center* há cinco *voice panels* para a

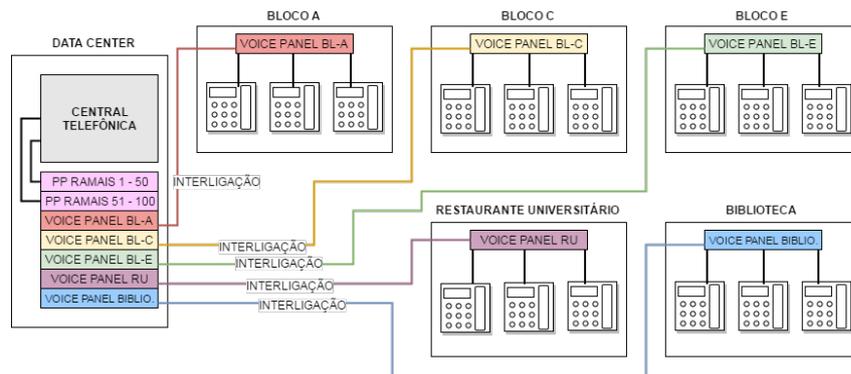


Figura 9: Atual Rede de Telefonia da UTFPR - Toledo

Fonte: Conteúdo próprio.

ligação dos ramais de cada bloco e nos *racks* de cada bloco há um *voice panel* para ligar os ramais no interior do bloco.

5.2 FUNCIONALIDADES DO IPBX

Para a produção do presente trabalho foi realizado o estudo dos principais protocolos de comunicação em rede, estudo da tecnologia VoIP e estudo da arquitetura do *framework* Asterisk. A partir do conhecimento obtido, foi realizado um questionário com os funcionários da UTFPR, de forma que apontassem quais das principais funcionalidades de telefonia poderiam ser utilizadas na universidade.

O questionário ficou aberto para respostas entre os dias 07 de Abril de 2016 e 29 de Abril de 2016, sendo divulgado na lista de e-mail de todos os servidores nos dias 19 e 27 de Abril, e obteve 67 respostas. Segundo informação obtida com a Coordenadoria de Gestão de Recursos Humanos do câmpus Toledo (COGERH-TD), até a data em que o questionário estava aberto, o número de servidores do câmpus, entre efetivos e substitutos, era de 177. Então, o número de respostas representa aproximadamente 38% do universo de servidores.

A Figura 10 possibilita a visualização de um resumo das respostas a respeito da utilidade de cada recurso solicitado no questionário. Os recursos mencionados no questionário e que aparecem no gráfico, em ordem, são:

1. Autenticação: com a função de autenticação o usuário pode realizar chamadas com sua senha a partir de qualquer aparelho da instituição;

2. Encaminhamento de chamadas quando ocupado: essa função fará que, quando um ramal estiver ocupado, chamadas realizadas para ele sejam encaminhadas para outro ramal

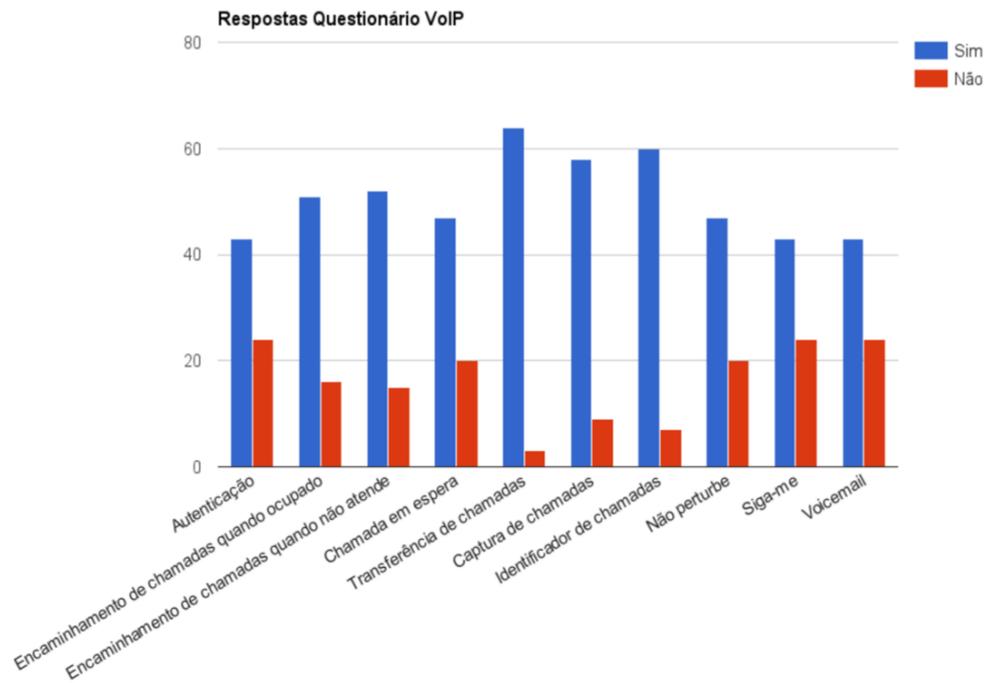


Figura 10: Respostas do Questionário de Funcionalidades VoIP realizado com os servidores da UTFPR - Câmpus Toledo.

Fonte: Conteúdo próprio.

do mesmo grupo;

3. Encaminhamento de chamadas quando não atende: essa função fará que, quando uma chamada para um ramal não for atendida, ela seja encaminhada para outro ramal do mesmo grupo;

4. Chamada em espera: função que permite ao usuário colocar uma chamada em espera e recuperar mais tarde;

5. Transferência de chamadas: essa função permite a transferência manual de chamadas pelo usuário;

6. Captura de chamadas: essa função permite ao usuário realizar a captura de chamadas de outro ramal do mesmo grupo;

7. Identificador de chamadas: essa função permite ao usuário saber o número que está ligando para ele;

8. Não perturbe: essa função permite ao usuário a opção de não ser perturbado pelo telefone em momentos que não pode atender;

9. Siga-me: essa função permite que as ligações para seu ramal toquem em outro local;

10. Voicemail: essa função permite o armazenamento de mensagens de voz para posterior recuperação pelo telefone;

Assim como é possível notar na Figura 10, em todas as questões a maioria dos usuários optou pela opção “Sim”, de que a implementação da funcionalidade seria útil para o desempenho do trabalho dentro da universidade.

5.3 IMPLEMENTAÇÃO E TESTES

Para a criação de um IPBX que atendesse às necessidades dos usuários levantadas por meio do questionário, foi utilizado um servidor *HP ProLiant DL320 G5p* com 2 GB de memória RAM, 400 GB de HDD, com processador *Intel(R) Xeon(R) CPU E3110 3.00GHz* e duas interfaces de rede *ethernet gigabit*. Além disso, foi instalado o sistema operacional *Debian 7.8 Wheezy x64* e realizadas as configurações necessárias no servidor, como a configuração das interfaces de rede e da lista de repositórios do sistema.

Em seguida, foi instalado o Asterisk em sua última versão (13.10) e configurado os seus principais arquivos de configuração. Foram utilizados os protocolos de transporte RTP e UDP e também o protocolo de sinalização SIP.

Como a rede da UTFPR tem banda disponível para o tráfego de pacotes de voz, foi escolhido para o projeto o *codec G.711a*, que é um *codec* que não possui atraso e possui maior qualidade que os demais, visto que não há compressão do áudio, porém consome mais banda.

Após a configuração necessária para a implementação de cada funcionalidade foram realizados testes com ramais configurados em *Softphones*, para depois disponibilizar para demais usuários voluntários testarem.

Para implementar a função de autenticação foi instalado o *mysql-server* no servidor, para a criação de um banco de dados com os ramais e senhas dos usuários, e também instalado o PHP 5.4.45 para a execução de um *script* de autenticação. O *script* de autenticação possui a finalidade de, quando um usuário tentar realizar uma chamada externa, conferir se o ramal e a senha informados pelo usuário são os mesmos dados salvos no banco de dados. O Asterisk possui uma API (*Application Program Interface*) independente de linguagem para o processamento de chamadas, nomeada AGI (*Asterisk Gateway Interface*). Isso permite que sejam criados simples e complexos programas para manipular e rotear chamadas. A Figura 11 mostra, de forma sequencial, como é executado um *script* AGI no Asterisk.

É possível visualizar que o usuário, ao fazer uma chamada, interage com o Asterisk,

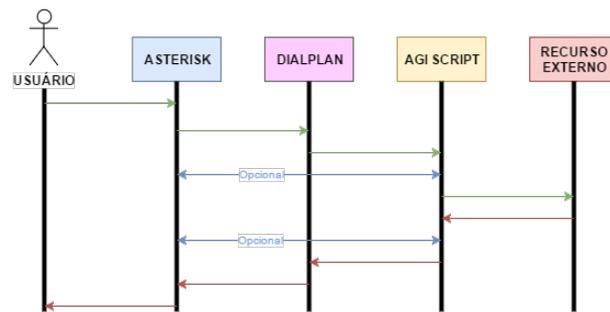


Figura 11: Processo de Execução de um *script* AGI no Asterisk.

Fonte: Conteúdo próprio.

que por sua vez interage com o plano de discagem (*dialplan*). Nesse caso, para o usuário conseguir realizar uma ligação externa o mesmo precisa se autenticar, então o plano de discagem irá solicitar a execução do *script* AGI. Esse último acessa recursos externos, como um banco de dados, por exemplo, e pode ou não comunicar-se com o Asterisk para trocar informações necessárias à sua execução. Ao fim da execução do *script* AGI é dado um retorno ao plano de discagem, que fará o roteamento da chamada e retornará para o Asterisk que, por fim, retornará para o usuário.

Optou-se realizar o encaminhamento de chamadas quando ocupado apenas para o ramal da recepção, de forma que a chamada é colocada em uma fila quando o ramal está ocupado. O encaminhamento de chamadas quando o usuário não atende foi implementado para o ramal da recepção, que aciona a função Siga-me automaticamente e toca o ramal da sala de vigilância da instituição, e também para o ramal principal da DIMASU (Divisão de Manutenção e Suporte ao Usuário), que ao não atender faz com que seja tocado o segundo ramal da Divisão. Para a maioria dos demais ramais não foram realizadas essas configurações de encaminhamento devido à funcionalidade de *voicemail*. Para as chamadas não atendidas ou que estão com a linha ocupada, a ligação é encaminhada para a caixa de mensagens do usuário, que poderá ouvir a mensagem gravada em outro momento.

No momento de criar uma conta de usuário, pode ser definido que o mesmo fará parte de um ou vários grupos. A função de captura de chamadas só pode ser realizada entre membros de um mesmo grupo. Já a função de identificação de chamadas ocorre sem a necessidade de qualquer implementação especial no Asterisk.

Os testes das funcionalidades foram realizados com o *Softphone X-LITE* para os computadores institucionais, o *Softphone Zoiper* para *Android*, equipamento ATA da marca *Linksys*, modelo *PAP2T-NA*, utilizados com telefone analógico da marca *Gigaset*, modelo *DA100*, am-

bos disponibilizados pela UTFPR câmpus Toledo e também um Telefone IP da marca *Khomp*, modelo *IPS 212*, disponibilizado pela UTFPR câmpus Londrina. Foram criados no servidor Asterisk diversos ramais e, posteriormente, configurados nos *Softphones*, ATA e Telefone IP.

5.4 METODOLOGIA DE AVALIAÇÃO DE QUALIDADE DA CHAMADA

A avaliação da qualidade dos serviços oferecidos pelo servidor IPBX com Asterisk implementado foi realizada por voluntários. Colaboraram com a avaliação funcionários da instituição e funcionários terceirizados. Foi preparado um ambiente para testes dentro da COGETI (Coordenadoria de Gestão de Tecnologia da Informação) com os mesmos equipamentos utilizados para os testes da implementação.

Os voluntários realizaram testes de ligações e das demais funcionalidades implementadas, sempre com o acompanhamento do autor do presente trabalho e um manual de uso, e depois cada um respondeu a um questionário de avaliação separadamente. A Figura 12 mostra o manual de uso disponível aos voluntários.



Figura 12: Manual de Uso das Funcionalidades do IPBX.

Fonte: Conteúdo próprio.

O questionário de avaliação foi criado com quatorze questões objetivas obrigatórias e uma questão aberta não obrigatória. As duas primeiras questões eram sobre o aparelho utilizado, as quatro seguintes sobre aspectos específicos ligados à qualidade da ligação, as próximas sete sobre o funcionamento das funcionalidades implementadas e a última objetiva era uma avaliação geral da qualidade dos serviços. Por último havia uma questão aberta na qual o voluntário podia fazer qualquer observação relevante ao projeto ou aos testes. As avaliações ocorreram entre os dias 27/10/2016 e 03/11/2016.

5.5 METODOLOGIA DE AFERIÇÃO DO CONSUMO DE BANDA INTERNA

Para avaliar se a rede interna da UTFPR-TD suportará o aumento de consumo que será proporcionado com o uso da tecnologia VoIP, foram realizadas aferições do consumo de banda interno da universidade.

Com o auxílio da ferramenta *Cisco Network Assistant* foi aferido o uso da rede sete vezes por dia (às 08h, 10h, 12h, 14h, 16h, 18h e 20h), por três dias (31/10/2016, 01/11/2016 e 03/11/2016). Foram escolhidos esses horários por abrangerem os diferentes períodos de funcionamento da universidade, tanto em momentos de grande uso da rede quanto em momentos de menor uso.

Esse aferimento foi realizado em três interfaces *gigabit* do *switch core* do *data center* da universidade, e foi considerado o tráfego de saída da interface (Tx) e o tráfego que entrava na interface (Rx). Essas interfaces são as responsáveis pela comunicação do *data center* com os demais blocos, o que indica que todo o tráfego gerado pelos usuários passa por elas.

A interface *gigabit* Gi1/0/13 do *switch core* do *datacenter* é responsável pela comunicação com o bloco A da universidade, a interface Gi1/1/1 é responsável pela comunicação com os blocos do Restaurante Universitário (RU) e da Biblioteca (DEBIB), e a interface Gi1/1/2 é responsável pela comunicação com os blocos C e E.

5.6 INFRAESTRUTURA NECESSÁRIA AO VOIP NA UTFPR

Visto que o câmpus Toledo da UTFPR possui uma boa infraestrutura de rede, não será necessária a aquisição de muitos equipamentos para a implantação da telefonia VoIP. Referente ao cabeamento, não precisa nenhuma alteração, já que a conectorização é padronizada, o que torna possível a conexão tanto dos equipamentos analógicos atuais, que utilizam conectores RJ11, quanto de telefones IP, que utilizam conectores RJ45. Não será necessária, também, a aquisição de um novo servidor para ser instalado e configurado o Asterisk, pois a universidade já conta com um equipamento disponível para essa finalidade.

A partir do cenário atual da universidade, para que possa ser utilizado o servidor VoIP, há duas possíveis modificações que podem ser realizadas: a troca de todos os telefones analógicos por telefones IP; ou a adaptação dos telefones analógicos com equipamentos ATA. Independentemente da escolha, esse será o investimento de maior custo do projeto. Uma opção que pode auxiliar na redução dos custos é a utilização de *Softphones* com *headsets* em locais que não tenha a necessidade de um aparelho físico e já tenha um computador. Partindo da infraes-

trutura atual, como vista anteriormente, a Figura 13 mostra de forma simplificada como deve ser a topologia física após a instalação dos novos aparelhos na rede.

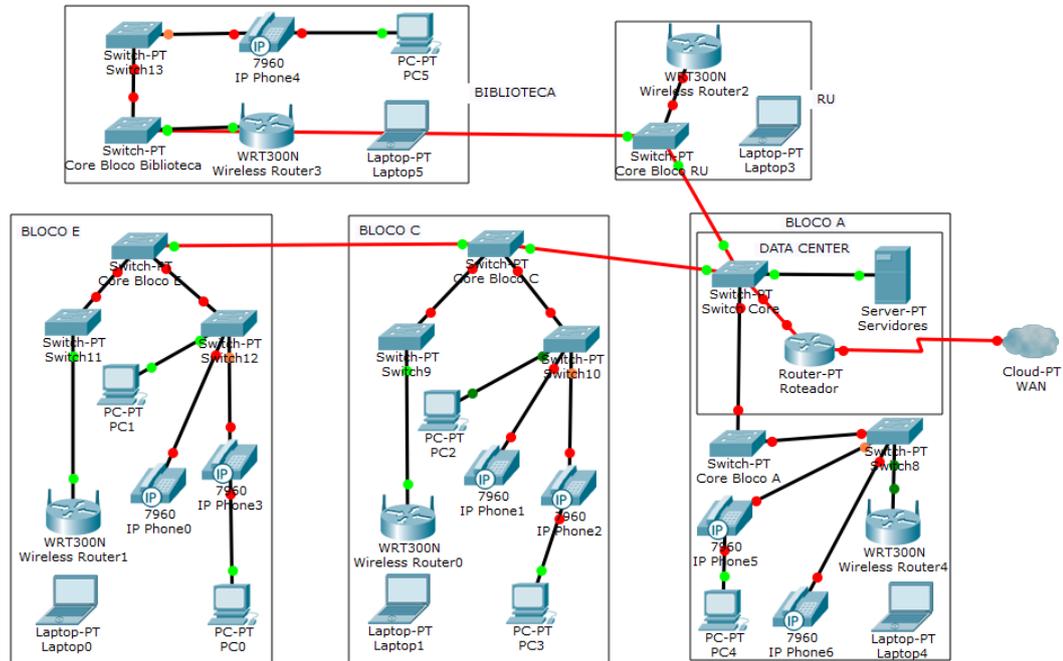


Figura 13: Topologia de Rede Física da UTFPR - Toledo com VoIP.

Fonte: Conteúdo próprio, gerado com o software de ensino *CISCO Packet Tracer*.

Os novos telefones não dependerão mais da infraestrutura de cabeamento metálico utilizada com os voice panels, mas irão demandar portas nos *switches*. Por esse motivo, deverá ser realizada a compra de equipamentos telefônicos que trabalhem como *switches*, para que seja utilizado apenas um cabo e, conseqüentemente, apenas uma porta de *switch* para que sejam ligados o telefone e o computador pessoal, onde necessário. De qualquer forma, caso necessário, há *switches fast ethernet* reservas para a ligação do número necessário de equipamentos.

6 RESULTADOS

6.1 CONSUMO DE BANDA INTERNO DA UTFPR

Conforme os valores obtidos com as aferições realizadas na rede da universidade, foram gerados gráficos que representam o consumo de banda de entrada e saída do *switch core* do *data center*.

A Figura 14 mostra o tráfego de saída do switch no dia 31/10/2016 em Mbps (megabit por segundo). É possível notar que o maior tráfego de saída é na interface Gi1/1/2 e que há variação no uso da banda durante o dia em todas as interfaces observadas. A interface Gi1/1/2 teve seu maior uso no dia às 10h, com aproximadamente 106 Mbps, e menor uso às 12h, com aproximadamente 15 Mbps. No início da tarde, às 14h, nota-se nova elevação no consumo nessa interface, que então passa a decrescer até a última aferição, às 20h. A interface Gi1/0/13 teve seu pico na aferição das 14h, quando atingiu uma taxa de 23 Mbps, e no restante do dia operou com valores mais baixos. Por último, a interface Gi1/1/1 foi a que operou com as taxas de consumo mais baixas durante o dia, com uma taxa máxima de consumo de 7 Mbps às 16h. Percebe-se então que a maior média de consumo diário foi na interface Gi1/1/2, com taxa de 54 Mbps. Visto que todas as interfaces são *gigabit*, a taxa média de consumo representou apenas 5,4% da banda disponível, enquanto o pico diário foi de 10,6%, o que pode ser considerado um baixo consumo.

Por outro lado, a Figura 15 mostra o tráfego de entrada do *switch* no dia 31/10/2016 em Mbps. Nota-se que o maior tráfego de entrada foi na interface Gi1/1/2, assim como foi para saída, mas percebe-se que nesse caso a interface Gi1/1/1 teve mais entrada do que a interface Gi1/0/13 praticamente o dia todo. A maior taxa de entrada para a interface Gi1/1/2 no dia foi às 16h, com aproximadamente 42 Mbps, e menor uso às 12h, com aproximadamente 17 Mbps. O pico de consumo da interface Gi1/0/13 foi às 20h, com uma taxa de 13 Mbps. Já a interface Gi1/1/1 obteve uma taxa máxima de consumo de 21 Mbps às 14h. É possível visualizar que a maior média de consumo diário de entrada também foi na interface Gi1/1/2, assim como de saída, com taxa de 32 Mbps. Essa taxa média de consumo representou apenas 3,2% da

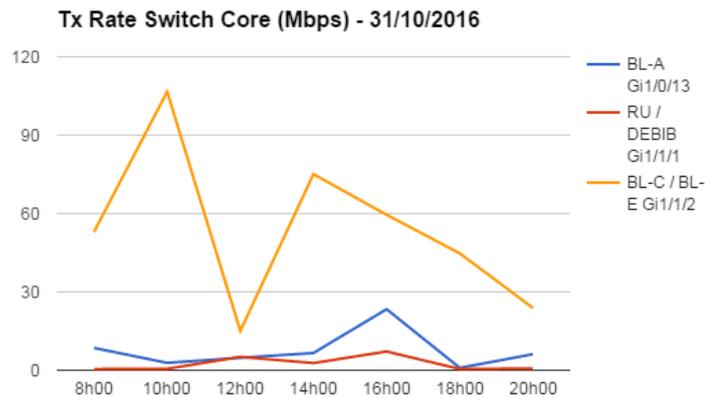


Figura 14: Taxa de Saída do *Switch Core* (Mbps) - 31/10/2016.

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

banda disponível, enquanto o pico diário aferido foi de 4,2%, o que pode ser considerado baixo. Percebe-se que a interface Gi1/1/2 teve uma taxa média de saída maior do que a taxa de entrada, enquanto as interfaces Gi1/0/13 e Gi1/1/1 tiveram maior taxa de entrada do que de saída.

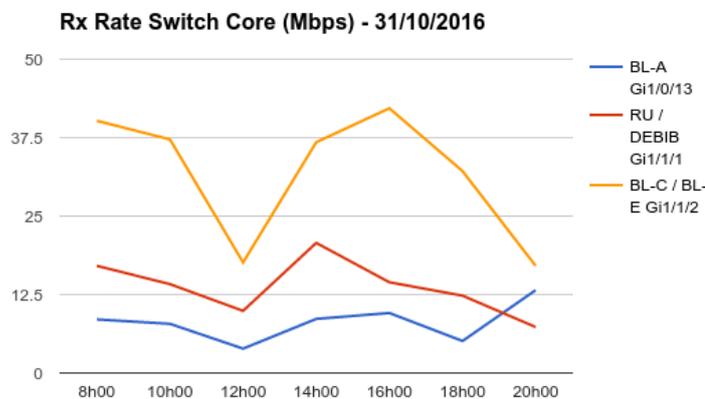


Figura 15: Taxa de Entrada do *Switch Core* (Mbps) - 31/10/2016.

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

A Figura 16 mostra o tráfego de saída do *switch core* no dia 01/11/2016 em Mbps. Assim como na aferição do dia anterior, nota-se que o maior tráfego de saída mantém-se na interface Gi1/1/2, que teve seu maior uso no dia às 16h, com consumo aproximado de 239 Mbps, e menor uso às 8h, com consumo aproximado de 43 Mbps. A média de consumo dessa interface foi maior que no dia anterior. A interface Gi1/0/13, assim como no dia anterior, operou com taxas baixas, e seu pico de consumo foi às 16h, com 21 Mbps. Às 8h o consumo dessa interface foi de apenas 2 Mbps e manteve-se baixo durante toda a parte da manhã. A interface Gi1/1/1 teve um consumo máximo de saída de 7 Mbps às 14h. É possível visualizar que a média

de consumo diário mais alta foi registrada na interface Gi1/1/2, com taxa de 97 Mbps, ou seja, apenas 9,7% da banda disponível, e a menor média foi registrada na interface Gi1/1/1, com taxa de 3,3% de ocupação da banda. O pico diário foi de 24% da banda disponível, maior que o do dia anterior, mas que também pode ser considerado um baixo consumo.

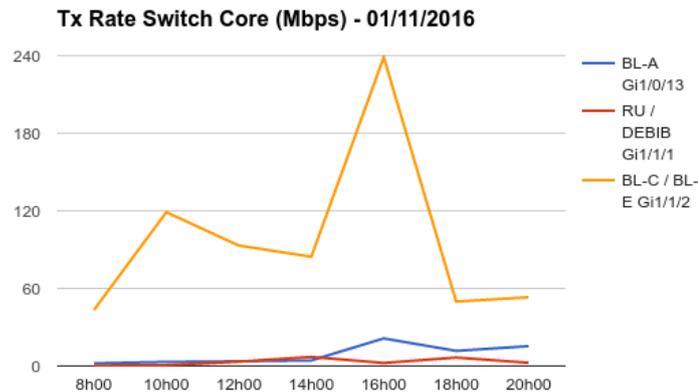


Figura 16: Taxa de Saída do *Switch Core* (Mbps) - 01/11/2016.

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

A Figura 17 exibe o tráfego de entrada do *switch core* no dia 01/11/2016 em Mbps. É possível visualizar que o maior tráfego de entrada foi novamente na interface Gi1/1/2, bem como foi para saída. A maior taxa de entrada para a interface Gi1/1/2 no dia foi novamente às 16h, com aproximadamente 141 Mbps, e menor uso às 20h, com aproximadamente 20 Mbps. O pico de consumo da interface Gi1/1/1 foi às 16h, com uma taxa máxima de consumo de 51 Mbps. Já a interface Gi1/0/13 teve um pico de 26 Mbps, também às 20h. Nota-se que as três interfaces tiveram seus picos de entrada às 16h e que a interface Gi1/1/2 teve uma taxa média de saída maior do que a taxa de entrada, enquanto as interfaces Gi1/0/13 e Gi1/1/1 tiveram maior taxa de entrada do que de saída. A taxa de consumo médio da interface que mais consumiu banda representou apenas 5,3% da banda disponível, e o pico diário aferido foi de 14,1%, considerado um gasto baixo.

A Figura 18 exibe o tráfego de saída do *switch* no dia 03/11/2016 em Mbps. Tal qual as demais aferições dos outros dias, a interface Gi1/1/2 é a que registrou o maior tráfego de saída. Nesse dia, seu pico de consumo ocorreu às 14h, com taxa aproximada de 131 Mbps, e menor consumo às 20h, com taxa aproximada de 32 Mbps. As interfaces Gi1/0/13 e Gi1/1/1 mantiveram suas baixas taxas de uso da banda disponível. A maior taxa registrada para a interface Gi1/0/13 foi de aproximadamente 9,4 Mbps, às 14h, e a menor de 1,3 Mbps, às 16h. Já a interface Gi1/1/1 teve seu pico de consumo às 12h, com 17 Mbps, e sua menor taxa às 14h, com 1,1 Mbps de consumo da banda disponível. A média de consumo de banda diário mais

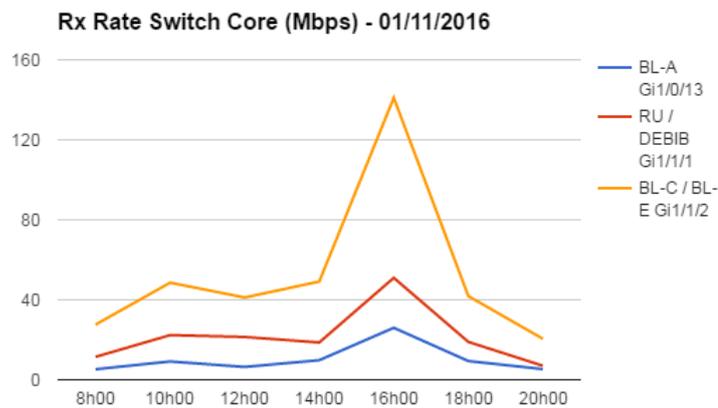


Figura 17: Taxa de Entrada do *Switch Core* (Mbps) - 01/11/2016.

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

alta foi registrada na interface Gi1/1/2, com taxa de 65 Mbps, ou seja, apenas 6,5% da banda disponível, e a menor média foi registrada na interface Gi1/0/13, com taxa de 5,3% de ocupação da banda. O pico registrado no dia foi de 13% de uso da banda disponível, o que pode ser considerado um uso baixo da rede.

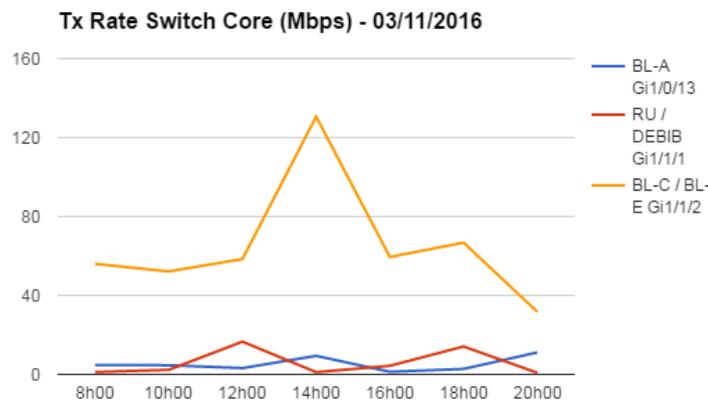


Figura 18: Taxa de Saída do *Switch Core* (Mbps) - 03/11/2016.

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

Por outro lado, a Figura 19 mostra o tráfego de entrada do *switch core* no dia 03/11/2016 em Mbps. Assim como para saída, nota-se que o maior tráfego de entrada foi na interface Gi1/1/2. A maior taxa de entrada para essa interface foi aferida às 14h, com aproximadamente 49 Mbps, e menor uso às 20h, com aproximadamente 21 Mbps. A interface Gi1/0/13 registrou seu maior consumo também às 14h, com aproximadamente 12 Mbps, e menor uso também às 20h, com pouco menos de 4 Mbps. Já a interface Gi1/1/1 registrou seu maior uso de banda às 12h, com pouco mais de 25 Mbps, e seu menor uso às 20h, com aproximadamente 10 Mbps.

Novamente, a interface Gi1/1/2 teve uma taxa média de saída maior do que a taxa de entrada, enquanto as interfaces Gi1/0/13 e Gi1/1/1 tiveram maior taxa média de entrada do que de saída. A taxa de consumo médio da interface que mais consumiu banda representou apenas 3,8% da banda disponível, e o pico diário aferido foi de 4,8%, considerado um gasto baixo.

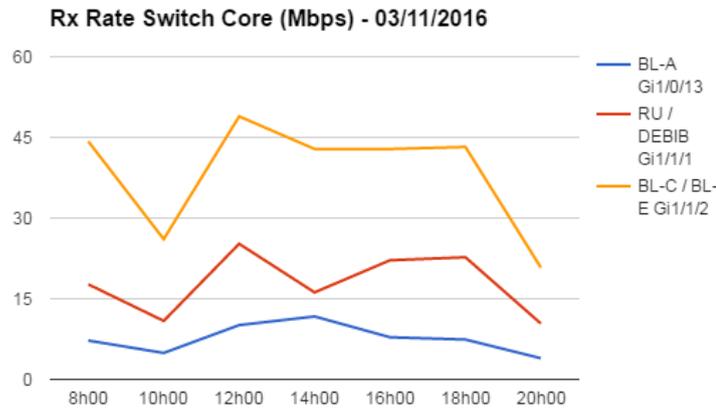


Figura 19: Taxa de Entrada do *Switch Core* (Mbps) - 03/11/2016.

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

Todos os dias de aferições mostraram que a interface Gi1/1/2 é a mais utilizada, com grande diferença das demais. Isso ocorre porque essa interface transmite e recebe dados de dois blocos da instituição (bloco C e bloco E), que contam com laboratórios de informática, várias salas de professores, alguns ambientes administrativos e também diversas salas de aula. A interface Gi1/0/13 é responsável pela comunicação com apenas um bloco (bloco A). Esse bloco possui diversas salas de professores e amplo espaço administrativo, além de salas de aula e laboratórios de eletrônica e química, contudo não possui a mesma demanda da interface Gi1/1/2. Por último, a interface Gi1/1/1 também é responsável pela comunicação com dois blocos (Restaurante Universitário e Biblioteca), contudo a demanda nesses ambientes é baixa, especialmente no RU, que possui seu horário de pico geralmente às 12h, que é o horário de almoço. A Biblioteca do câmpus conta com poucos servidores, que junto aos computadores disponibilizados aos alunos no ambiente não conseguem elevar muito o consumo dessa interface.

Haja vista as figuras e análises apresentadas das aferições realizadas, pode-se afirmar que o consumo de banda da UTFPR de Toledo é pequeno, comparado à sua capacidade. Entre os dias aferidos, não houve nenhuma interface em que sua média de consumo tenha sido superior a 10% da banda disponível.

Com o uso do *Softphone Zoiper para Android*, durante uma ligação de teste foi possível extrair algumas informações importantes referente ao uso da rede em uma ligação VoIP, como

pode ser visto na Figura 20. O relatório de estatísticas de rede gerado pelo aplicativo mostrou que uma ligação SIP com o *codec* G.711a (aLaw) e o protocolo RTP consumiu uma taxa média de transmissão de dados (*Current sent bitrate*) de 84 Kb/s, e a taxa média de recebimento de dados (*Current received bitrate*) de foi de 83 Kb/s. Com essa informação é possível estimar o aumento do consumo de banda que o VoIP trará para a instituição.

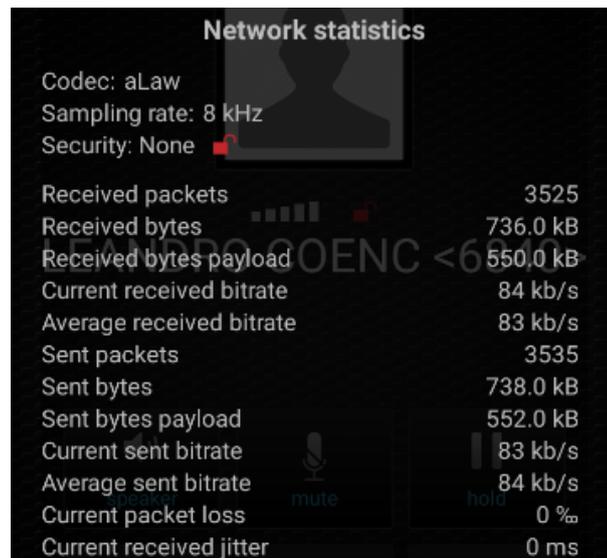


Figura 20: Estatísticas da Rede em Ligação VoIP.

Fonte: Conteúdo próprio, gerado com o *Softphone Zoiper*.

Como não é possível saber qual o número máximo de ligações simultâneas que ocorrem atualmente na universidade, já que não foram encontrados *softwares* gratuitos para a gestão de PBX convencionais disponíveis no mercado e a UTFPR Toledo não possui a licença de nenhum *software* do gênero, será estimado o consumo de banda com um número alto de ligações simultâneas. Caso ocorram 50 ligações simultâneas, hipoteticamente, com uma taxa de transmissão de 84 Kb/s, haverá um aumento de 4200 Kb/s (ou 4 Mb/s) na rede.

Visto que a rede da UTFPR é *gigabit* e que o consumo médio da banda não ultrapassa 10% da banda total disponível, um aumento de 4 Mb/s não trará uma diferença significativa para a rede. Com isso pode-se afirmar que não será necessário, a princípio, a implementação de QoS na rede da universidade.

6.2 AVALIAÇÃO DE QUALIDADE DO IPBX COM ASTERISK

Para determinar a qualidade dos serviços oferecidos pelo servidor IPBX implementado, foram realizadas onze avaliações por voluntários, de acordo com a metodologia apresen-

tada anteriormente.

A primeira pergunta do questionário foi sobre qual tipo de equipamento o usuário utilizou para os testes. Conforme a Figura 21, foram realizadas avaliações com os diferentes tipos de aparelhos disponibilizados no laboratório de testes e a maioria utilizou *Softphones*.

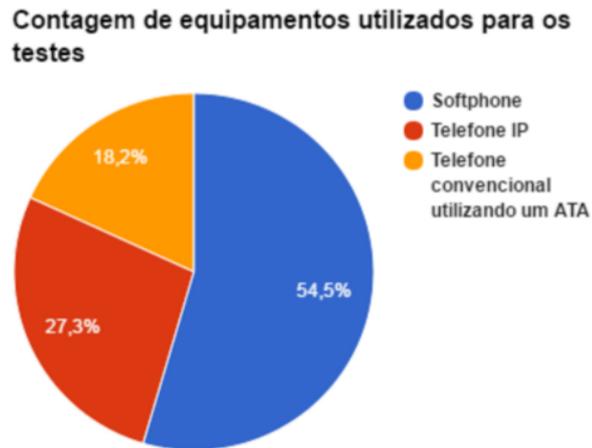


Figura 21: Contagem de Equipamentos Utilizados para os Testes.

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

A segunda questão proposta foi sobre o tipo de conexão utilizada pelo equipamento, visto que foram utilizados *Softphones* em computadores pessoais e em *Smartphones*. A Figura 22 mostra a contagem das respostas.

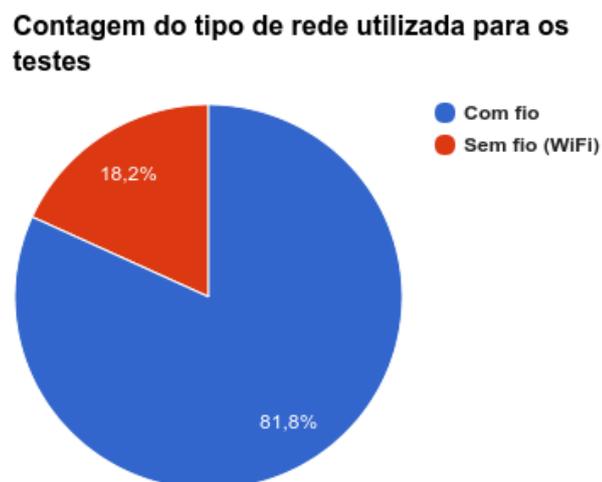


Figura 22: Contagem do Tipo de Rede Utilizada para os Testes.

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

A terceira questão já avaliava a qualidade da ligação, com o questionamento sobre a

presença ou não de eco perceptível nas ligações. A Figura 23 mostra que em nenhuma das onze avaliações foi possível detectar eco nas ligações realizadas.

Avaliação de presença de eco nas ligações

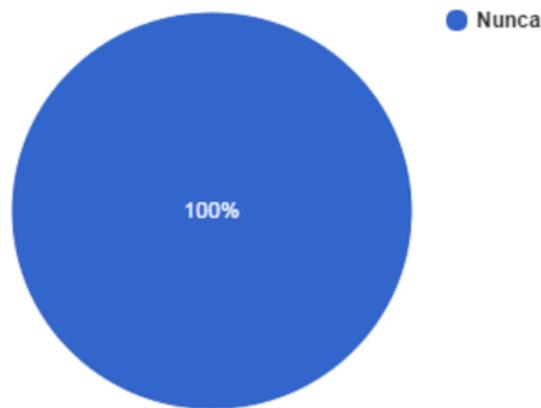


Figura 23: Avaliação de presença de eco nas ligações.

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

A Figura 24 exibe o resultado da quarta questão, que possuía a intenção de avaliar se havia ou não atraso de voz perceptível nas ligações, que fosse capaz de atrapalhar a conversação. A maioria das avaliações constatou que nunca havia atrasos nas ligações, e pequena parte respondeu ter atrasos raramente.

Avaliação de atraso da voz nas ligações

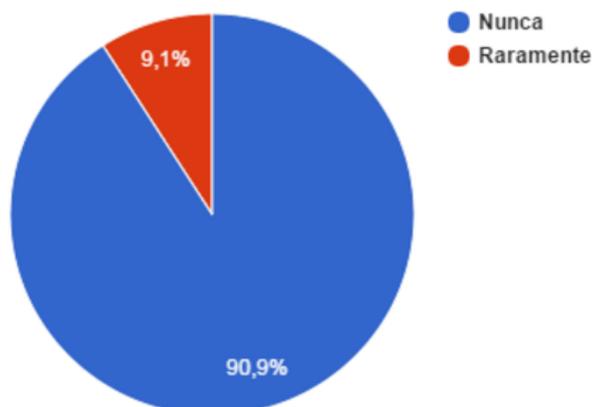


Figura 24: Avaliação de atraso de voz nas ligações.

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

A quinta questão avaliava a presença de ruídos perceptíveis nas ligações, de forma que pudesse comprometer a qualidade a conversação. A Figura 25 mostra o resultado das avaliações. Nota-se que as ligações eram limpas, visto que nunca havia ruído nas ligações avaliadas.

Avaliação de presença de ruídos nas ligações

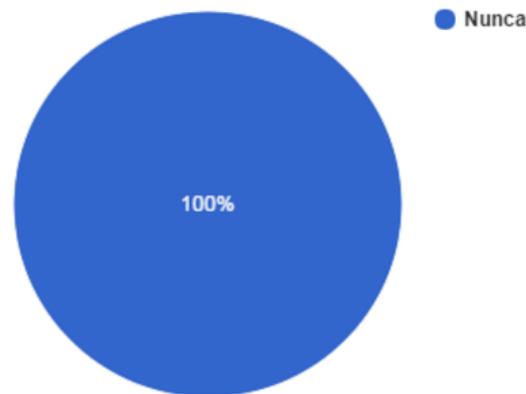


Figura 25: Avaliação de ruído nas ligações.

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

A próxima pergunta foi sobre a presença de falhas na transmissão de voz nas ligações. A Figura 26 mostra que na maioria das ligações não houve falha, porém algumas avaliações relatam que falhas perceptíveis ocorreram em alguns momentos. Conforme a resposta do tipo de equipamento utilizado para esses testes que apresentaram falhas e conforme as observações inseridas ao final do questionário, pode-se afirmar que as falhas foram perceptíveis quando utilizado *Softphone* com comunicação sem fio para a avaliação, ou que utilizando um equipamento com fio estabeleceu-se comunicação com um dispositivo sem fio. Além disso, as observações mostram que as falhas ocorriam nos primeiros segundos da ligação e depois a chamada se estabilizava sem falhas.

Avaliação de presença de falhas na transmissão da voz ("picotes") nas ligações

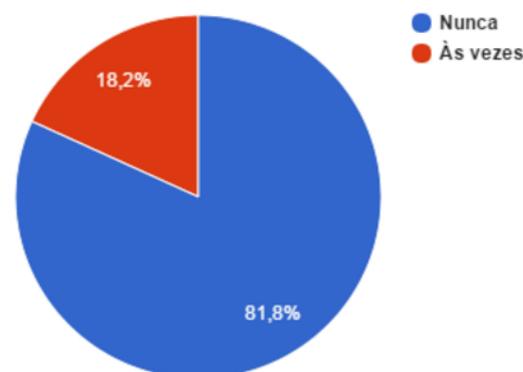


Figura 26: Avaliação de falhas na transmissão da voz (picotes) nas ligações.

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

A partir da sétima questão avaliou-se a qualidade de algumas funcionalidades implementadas no servidor IPBX. Essa pergunta avaliava a função de autenticação do usuário a partir de qualquer ramal para realizar chamadas externas. A Figura 27 mostra que a maioria dos usuários avaliou como ótimo ou excelente o funcionamento da implementação. Como a central está em fase de implementação, ela não estava conectada à rede de telefonia externa, então o usuário apenas sabia se conseguiu ou não se autenticar por meio do retorno do Asterisk, mas não ligava de fato para fora da universidade. Por esse motivo e também por nem todos os ramais terem configurada a permissão de realizar ligações externas, percebe-se que pouco menos da metade dos usuários não testou a funcionalidade, o que pode ser visto pelas respostas “Não se aplica”.

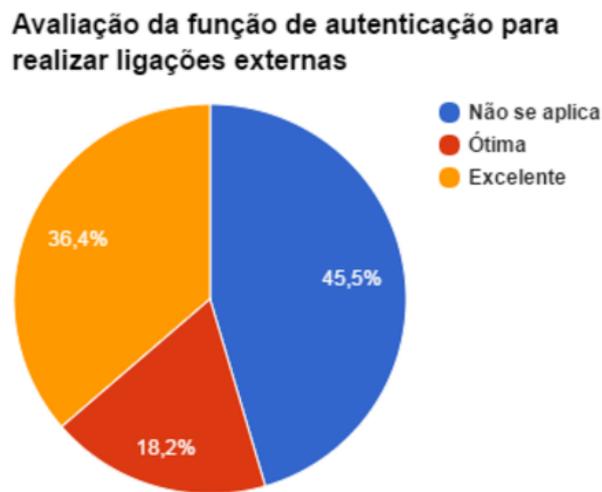


Figura 27: Avaliação da função de autenticação para realizar ligações externas.

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

A oitava pergunta foi referente à função de estacionamento de chamadas. Conforme as respostas do questionário, pode-se afirmar que a funcionalidade operou da forma satisfatória em *Softphones* e no Telefone IP, contudo apresentou problemas no ATA. Entretanto, o resultado da avaliação foi satisfatório no contexto geral de funcionamento. A Figura 28 exhibe o resultado da avaliação da funcionalidade.

As questões seguintes foram referentes à funcionalidade de transferência de chamadas e captura de chamadas, respectivamente. A Figura 29 e a Figura 30 mostram que todos os usuários que testaram as funcionalidades ficaram satisfeitos com os resultados e que apenas um usuário não testou ambas funcionalidades.

A décima primeira questão foi sobre o funcionamento do identificador de chamadas. Como é uma funcionalidade nativa do Asterisk e não necessita nenhum tipo de implementação por parte do administrador do IPBX, já era esperado que a avaliação fosse satisfatória. A Fi-

Avaliação da função de estacionamento de chamadas

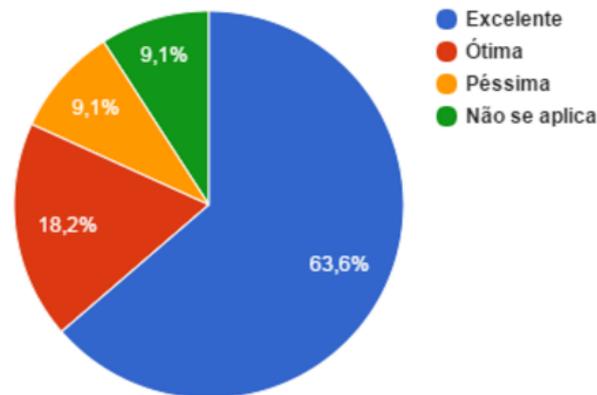


Figura 28: Avaliação da função de estacionamento de chamadas.

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

Avaliação da função de transferência de chamadas

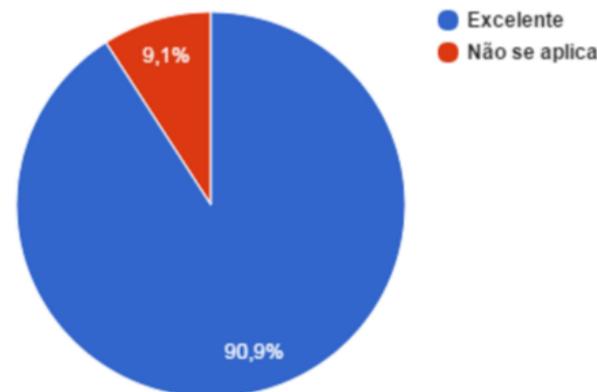


Figura 29: Avaliação da funcionalidade de transferência de chamadas.

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

gura 31 mostra que o identificador de chamadas opera dentro do esperado e pode-se afirmar que os dois usuários que não testaram a funcionalidade estavam utilizando equipamento telefônico convencional com ATA, sem visor gráfico, e por isso não conseguiram atestar o funcionamento.

A Figura 32 mostra o resultado da pergunta seguinte, referente à função de não perturbe. Apenas um usuário ficou insatisfeito com a operação dessa funcionalidade e os outros dez ficaram satisfeitos. De acordo com as repostas, sabe-se que o usuário insatisfeito testou em um equipamento ATA, que habilitou a funcionalidade normalmente, mas não conseguiu desabilitar. Constatou-se que o problema foi na configuração do equipamento ATA e após cor-

Avaliação da função de captura de chamadas

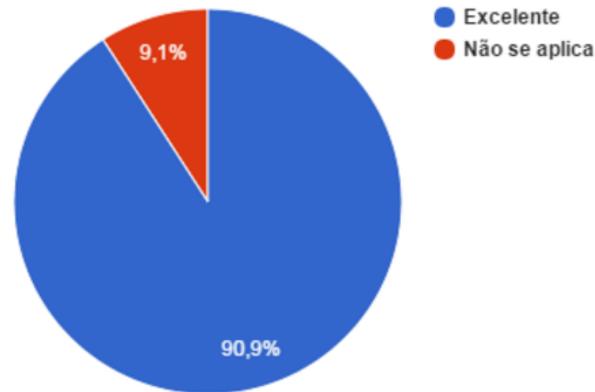


Figura 30: Avaliação da funcionalidade de captura de chamadas.

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

Avaliação da função de identificador de chamadas

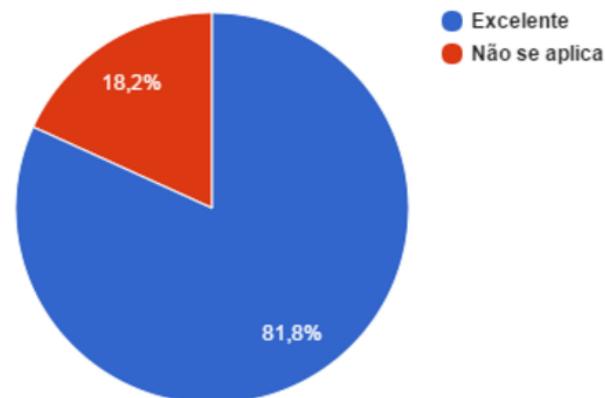


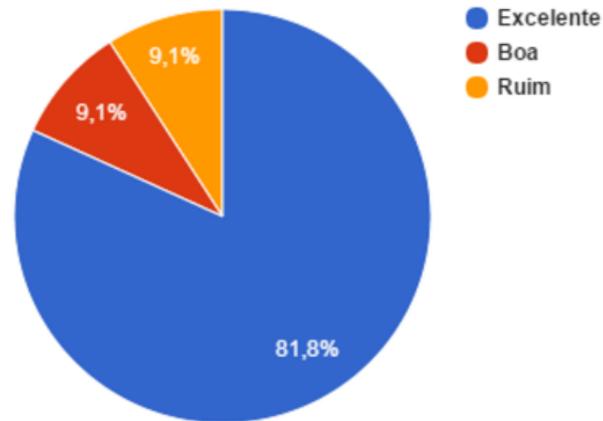
Figura 31: Avaliação da função identificador de chamadas.

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

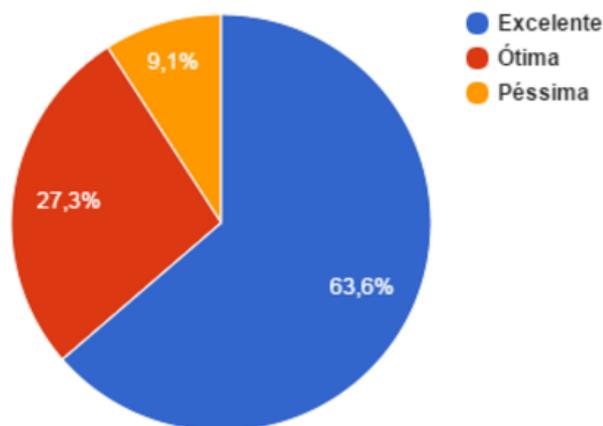
reção da configuração do equipamento, em um teste de outro voluntário, o problema já estava solucionado.

A décima terceira questão avaliou a qualidade de operação do voicemail implementado. Um voluntário que usou o *Softphone X-Lite* relatou que o *software* não capturou as teclas digitadas de ramal e senha quando solicitado. Os demais voluntários classificaram o funcionamento como satisfatório. A Figura 33 mostra o resumo das respostas dos voluntários ao questionário.

A última questão objetiva do questionário tinha o intuito de avaliar a qualidade geral

Avaliação da função de não perturbe**Figura 32: Avaliação da funcionalidade não perturbe.**

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

Avaliação da função de voicemail**Figura 33: Avaliação da funcionalidade voicemail.**

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

das ligações e dos serviços oferecidos pelo servidor IPBX com Asterisk em testes. A Figura 34 mostra que, mesmo com os problemas enfrentados por alguns voluntários em alguns tipos de equipamentos, todos ficaram satisfeitos com a qualidade das ligações e dos serviços implementados.

Avaliação geral da qualidade das ligações e dos serviços oferecidos

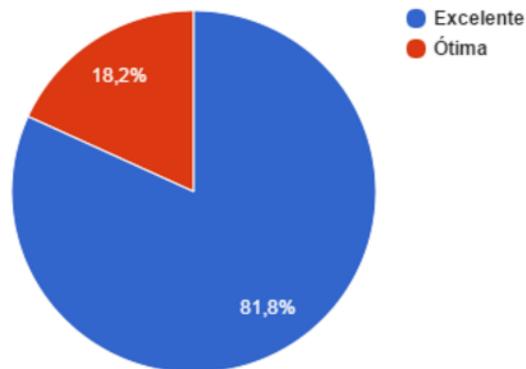


Figura 34: Avaliação geral de qualidade das ligações e serviços oferecidos.

Fonte: Conteúdo próprio, gerado com o Google Planilhas.

No campo de observações, diversos voluntários elogiaram o funcionamento do sistema em testes. Destaca-se que as ligações VoIP foram classificadas como melhores até mesmo que o atual sistema de telefonia do câmpus, especialmente por não terem ruídos e eco.

7 CONCLUSÃO

O presente trabalho apresentou o estudo e o levantamento de informações necessárias para atestar a viabilidade da implantação de um servidor IPBX com Asterisk na Universidade Tecnológica Federal do Paraná, câmpus Toledo. Foram estudados diversos protocolos e *codecs*, além do *framework* Asterisk. Nota-se que a implementação pode ser realizada de diversas formas, utilizando diferentes técnicas, protocolos e *codecs*, ou até combinações dos mesmos. Considerando a infraestrutura da UTFPR e a necessidade de compra de equipamentos para a implantação, buscou-se mostrar a viabilidade de um projeto que melhor se encaixasse à realidade da universidade.

Haja vista o resultado satisfatório da avaliação de qualidade dos serviços oferecidos pelo servidor VoIP em testes, conclui-se que o projeto piloto do IPBX com Asterisk mostrou-se capaz de atender à demanda dos usuários da universidade, e que o câmpus Toledo da Universidade Tecnológica Federal do Paraná possui uma infraestrutura adequada, capaz de receber a tecnologia VoIP.

Os resultados de avaliação apresentados mostram que a rede da instituição é superdimensionada e possui capacidade suficiente para o tráfego que pode ser gerado com a implantação do servidor VoIP, sem a necessidade de tomar medidas adicionais. Além disso, o estudo da infraestrutura atual mostrou que não há necessidade de alteração no cabeamento ou em equipamentos que fazem parte da rede. Para a implantação, há a necessidade de a universidade adquirir equipamentos ATA ou telefones IP, além de headsets para a utilização com Softphones.

Visto que o presente trabalho contempla um estudo sobre a viabilidade de implantação de um IPBX na UTFPR Toledo, é possível evidenciar outros trabalhos que podem ser realizados futuramente. Dentre eles destacam-se:

- Aplicação do Modelo E: o Modelo E é definido na recomendação G.107 do ITU-T. Esse modelo avalia os efeitos combinados de variações em diversos parâmetros de transmissão que podem afetar a qualidade de uma conversa telefônica e é indicado para avaliar a qualidade de uma ligação VoIP. A utilização do Modelo E no IPBX da universidade poderá avaliar a

qualidade da ligação em termos e valores técnicos, muitas vezes imperceptíveis aos usuários, diferentemente da forma utilizada no presente trabalho.

- Desenvolvimento de Interface Web com o estado dos ramais: a UTFPR Toledo conta com o trabalho de telefonistas, que possuem a função de transferir ligações internas e externas para os destinos solicitados. Uma interface que exiba de forma clara o estado dos ramais (disponíveis ou indisponíveis) pode auxiliar no trabalho de transferência das ligações.

- Substituição de arquivos de configuração por Banco de Dados: o funcionamento do Asterisk ocorre de acordo com as configurações realizadas em seus arquivos. Uma forma mais simples para gerenciar algumas de suas configurações seria por meio do uso de banco de dados. O Asterisk já possui suporte para manter algumas informações em bancos de dados, contudo a documentação ainda é um tanto quanto escassa. Uma boa aplicação de banco de dados com o Asterisk seria, por exemplo, para armazenar as configurações de cada ramal, que atualmente estão em arquivos de configuração individuais. Essa implementação geraria mais um trabalho futuro: a criação de uma interface web para gerenciar os ramais que incluía a criação, leitura, atualização e exclusão de ramais e suas características.

- Segurança com SRTP: de forma nativa o Asterisk utiliza o RTP para transporte dos pacotes de áudio. Contudo, esse protocolo não oferece nenhuma criptografia ao áudio da ligação, o que permite que, caso seja feita uma captura na rede, seja possível reproduzir claramente o conteúdo de uma ligação. Com a utilização do protocolo SRTP não é possível decifrar o conteúdo capturado.

A telefonia VoIP já deve ser considerada como uma realidade em nosso meio e não mais como uma ferramenta para o futuro. As redes IP estão em todos os lugares e são capazes de oferecer grande qualidade para a transmissão de pacotes de voz. A UTFPR - Toledo possui a necessidade de ampliar sua rede de telefonia e possui uma excelente infraestrutura de rede. Nota-se que não convém à instituição buscar outras formas ultrapassadas para a ampliação de sua rede de telefonia a não ser adotar o uso da telefonia de voz sobre IP.

REFERÊNCIAS

- BALBINOT, R. et al. Voz sobre ip-tecnologia e tendências. In: **Simpósio Brasileiro de Redes de Computadores**. Natal, CE: [s.n.], 2003.
- BORDIM, J. L. **Introdução à Voz sobre IP e Asterisk**. [S.l.]: Escola Superior de Redes RNP, 2010.
- BRYANT, R. **Asterisk Versions**. 2010. Disponível em: <<https://wiki.asterisk.org/wiki/display/AST/Asterisk+Versions>>. Acesso em: 27 de abril de 2016.
- COMER, D. **Interligação de Redes com TCP/IP: Princípios, Protocolos E Arquitetura**. 6. ed. [S.l.]: Elsevier Brasil, 2015.
- DAVENPORT, M. **Asterisk Architecture, The Big Picture**. 2010. Disponível em: <<https://wiki.asterisk.org/wiki/display/AST/Asterisk+Architecture>>. Acesso em: 11 de maio de 2016.
- DAVENPORT, M. **Asterisk as a Swiss Army Knife of Telephony**. 2010. Disponível em: <<https://wiki.asterisk.org/wiki/display/AST/Asterisk+as+a+Swiss+Army+Knife+of+Telephony>>. Acesso em: 20 de abril de 2016.
- DAVENPORT, M. **Bridges**. 2010. Disponível em: <<https://wiki.asterisk.org/wiki/display/AST/Bridges>>. Acesso em: 16 de maio de 2016.
- DAVENPORT, M. **A Brief History of the Asterisk Project**. 2010. Disponível em: <<https://wiki.asterisk.org/wiki/display/AST/A+Brief+History+of+the+Asterisk+Project>>. Acesso em: 20 de abril de 2016.
- DAVENPORT, M. **Comments**. 2010. Disponível em: <<https://wiki.asterisk.org/wiki/display/AST/Comments>>. Acesso em: 15 de maio de 2016.
- DAVENPORT, M. **Objects**. 2010. Disponível em: <<https://wiki.asterisk.org/wiki/display/AST/Objects>>. Acesso em: 15 de maio de 2016.
- DAVENPORT, M. **Sections and Settings**. 2010. Disponível em: <<https://wiki.asterisk.org/wiki/display/AST/Sections+and+Settings>>. Acesso em: 14 de maio de 2016.
- DAVENPORT, M. **Template Syntax**. 2010. Disponível em: <<https://wiki.asterisk.org/wiki/display/AST/Template+Syntax>>. Acesso em: 15 de maio de 2016.
- DAVENPORT, M. **Templates**. 2010. Disponível em: <<https://wiki.asterisk.org/wiki/display/AST/Templates>>. Acesso em: 15 de maio de 2016.

- DAVENPORT, M. **Types of Asterisk Modules**. 2010. Disponível em: <<https://wiki.asterisk.org/wiki/display/AST/Types+of+Asterisk+Modules>>. Acesso em: 12 de maio de 2016.
- DAVENPORT, M. **Using Templates**. 2010. Disponível em: <<https://wiki.asterisk.org/wiki/display/AST/Using+Templates>>. Acesso em: 15 de maio de 2016.
- FALL, K.; STEVENS, W. **TCP/IP Illustrated**. 2. ed. [S.l.]: Addison-Wesley, 2011.
- GL Communications Inc. **MAPS SIP Protocol Emulator**. 2014. Disponível em: <<http://www.gl.com/sip-rtp-protocol-simulator-maps.html>>. Acesso em: 28 de maio de 2016.
- GOODE, B. Voice over internet protocol (voip). **Proceedings of the IEEE**, v. 90, n. 9, p. 1495–1517, Sep 2002. ISSN 0018-9219.
- HARFF, S. Requisitos e proposta para implantação de um servidor voip. Trabalho de Conclusão de Curso, Universidade Federal do Rio Grande do Sul. Porto Alegre, RS. 2008.
- ITU-T. **ITU-T RECOMMENDATION G.711**. [S.l.]: ITU, 1988.
- ITU-T. **ITU-T RECOMMENDATION G.723.1**. [S.l.]: ITU, 2006.
- ITU-T. **ITU-T RECOMMENDATION G.729**. [S.l.]: ITU, 2012.
- KERRISK, M. **TCP Fast Open: expediting web services**. 2012. Disponível em: <<https://lwn.net/Articles/508865/>>. Acesso em: 27 de maio de 2016.
- magicJack VocalTec Ltd. **About us**. 2016. Disponível em: <<http://www.vocaltec.com/governance.cfm>>. Acesso em: 18 de maio de 2016.
- MICHELSON, M. **Presence State**. 2012. Disponível em: <<https://wiki.asterisk.org/wiki/display/AST/Presence+State>>. Acesso em: 17 de maio de 2016.
- NEWTON, R. **Asterisk Internal Database**. 2014. Disponível em: <<https://wiki.asterisk.org/wiki/display/AST/Asterisk+Internal+Database>>. Acesso em: 15 de maio de 2016.
- NEWTON, R. **Channels**. 2014. Disponível em: <<https://wiki.asterisk.org/wiki/display/AST/Channels>>. Acesso em: 16 de maio de 2016.
- NEWTON, R. **Device State**. 2014. Disponível em: <<https://wiki.asterisk.org/wiki/display/AST/Device+State>>. Acesso em: 16 de maio de 2016.
- NEWTON, R. **Extension State and Hints**. 2014. Disponível em: <<https://wiki.asterisk.org/wiki/display/AST/Extension+State+and+Hints>>. Acesso em: 16 de maio de 2016.
- NEWTON, R. **States and Presence**. 2014. Disponível em: <<https://wiki.asterisk.org/wiki/display/AST/States+and+Presence>>. Acesso em: 16 de maio de 2016.

PASSITO, A. et al. Análise de desempenho de tráfego voip utilizando o protocolo ip security. In: **I Workshop de Ciência da Computação e Sistemas de Informação da Região Sul (WORKCOMP SUL)**. Florianópolis, SC: [s.n.], 2004.

PETERSON, L.; DAVIE, B. **Computer Networks: A Systems Approach**. 5. ed. [S.l.]: Elsevier Science, 2011. (The Morgan Kaufmann Series in Networking).

SCHULZRINNE, H. et al. **RFC 3550 RTP: A Transport Protocol for RealTime Applications**. 2003. Disponível em: <<https://tools.ietf.org/html/rfc6864>>. Acesso em: 24 de maio de 2016.

SOUZA, F. N. M. d. A. **Desempenho VoIP: Codificação de Voz**. 2006. Disponível em: <http://www.teleco.com.br/tutoriais/tutorialmondesvoip/pagina_3.asp>. Acesso em: 20 de maio de 2016.

TANENBAUM, A.; WETHERALL, D. **Redes de Computadores**. 5. ed. [S.l.]: Pearson Prentice Hall, 2011.

TOUCH, J. **RFC 6864: Updated Specification of the IPv4 ID Field**. 2013. Disponível em: <<https://tools.ietf.org/html/rfc6864>>. Acesso em: 10 de maio de 2016.