

Implementação de um Servidor VoIP Asterisk Utilizando o Protocolo SIP

Luimar Donini, Dimas Irion Alves, Renato Machado e Edson Rodrigo Schlosser

Resumo—Nos últimos anos, a utilização e o interesse em sistemas de comunicações baseados em tecnologias VoIP, cresceram significativamente, devido ao baixo custo e a possibilidade de integração entre a infraestrutura utilizada para serviços de voz e de dados. Por esta razão, diversas topologias e protocolos de comunicação foram propostos para tecnologias VoIP. Baseado neste contexto, esse trabalho tem como objetivo, servir como um tutorial simplificado para a implementação de um servidor VoIP Asterisk de baixa complexidade e baixo custo, utilizando o sistema operacional Linux Fedora.

Palavras-Chave—VoIP, Asterisk, Linux, servidor.

Abstract—Nowadays, the interest and the usage of VoIP based communications systems is increasing significantly due to the low cost and the possibility of integrating the required infrastructure for voice and data services. Additionally, a large variety of topologies and communication protocols have been proposed for VoIP technologies. In this context, this paper aims to serve as a simplified tutorial for implementing a low-complexity and low cost VoIP Asterisk server, using the Fedora Linux operating system.

Keywords—VoIP, Asterisk, Linux, server.

I. INTRODUÇÃO

A tecnologia VoIP (*Voice over IP*) surgiu em Israel no ano de 1995, quando um grupo de pesquisadores conseguiu desenvolver um sistema que permitiu a utilização dos recursos multimídia de um computador doméstico para iniciar conversas de voz através da internet. É importante salientar que a tecnologia desenvolvida apresentava qualidade precária de sinal e pouca confiabilidade. No entanto, ainda em 1995, a empresa chamada Vocaltec Inc. lançava o *internet phone software*, o qual foi o primeiro *software* dedicado à comunicação por VoIP [1]. Esse é considerado o precursor dos *softphones* atuais, os quais são programas “clientes” que, entre outros recursos, emulam as funcionalidades de um telefone físico. Desde então, o crescimento do interesse da academia e da indústria, bem como, as aplicações que utilizam tecnologias VoIP vêm crescendo significativamente, utilizado inclusive em sistemas de comunicação 4G [2].

Nos sistemas VoIP, a voz humana é captada e transformada em dados, os quais são transmitidos junto com um cabeçalho de informações de controle, que são referentes ao destino da informação. É importante salientar que os pacotes possuem tamanhos idênticos, seguindo o funcionamento dos protocolos

TCP/IP. Em computadores, os responsáveis pela captação e empacotamento dos dados de voz, são os *softphones*, que como já mencionado, são programas de computador que agem como telefones virtuais. Quando um destinatário recebe pacotes de dados de voz, esse os converte novamente para o formato analógico, para que possam ser transmitidos por um alto-falante.

Atualmente, o padrão VoIP mais difundido é o que utiliza o protocolo RTP (*Real Time Protocol*). O RTP organiza os pacotes de dados, de forma que seja possível a sua transmissão em tempo real. Caso um pacote possua um atraso maior do que o máximo atraso permitido, esse será considerado perdido. Nesse caso, o protocolo RTP realiza uma interpolação dos sinais recebidos, com o intuito de preencher as lacunas geradas pelas perdas de pacotes. Assim sendo, o RTP é o protocolo encarregado das aplicações que envolvem áudio e vídeo em sistemas VoIP. Devido a essa característica, seu funcionamento é atrelado a outro protocolo, o RTCP (*Real Time Control Protocol*), o qual é responsável pela compressão dos pacotes de dados e também atua no monitoramento dos mesmos. Existem ainda outros protocolos conhecidos como *codecs*, que servem para adicionar funcionalidades e maior qualidade à comunicação [3].

Este trabalho apresenta a implementação de um sistema VoIP Asterisk de baixo custo e baixa complexidade. O restante do trabalho está organizado da seguinte maneira. A Seção II apresenta os equipamentos utilizados para implementação, bem como características e configurações do sistema; resultados da implementação são apresentados e discutidos na Seção III; a Seção IV apresenta algumas considerações finais, referentes a implementação.

II. METODOLOGIA

Para a realização do projeto, foi utilizado um computador *desktop* com processador Intel Dual Core de 2.2 GHz, 4 GB de memória RAM DDR2 e sistema operacional Linux Fedora x64 *release 22 update 3*. A configuração do servidor é bastante simples para os dias atuais, entretanto, é capaz de gerenciar conexões de áudio e vídeo com bastante estabilidade. O *software* utilizado como servidor foi o Asterisk, que suporta os protocolos H.323, SIP (*Session Initiation Protocol*) e o proprietário IAX (*Inter Asterisk eXchange*). O Asterisk é considerado um dos mais completos sistemas de telefonia por *software* [4],[5].

A. Instalação e configuração

O Asterisk é um *software* livre, que tem como função atuar como uma central de distribuição telefônica, baseada

Luimar Donini e Edson Rodrigo Schlosser, Universidade Federal do Pampa, Alegrete-RS, Brasil, E-mails: luimar@gmail.com e schlossertche@gmail.com. Dimas Irion Alves Universidade Federal do Pampa, Alegrete-RS, Brasil, e Universidade Federal de Santa Catarina, Florianópolis, Brasil, E-mail: dimasalves@unipampa.edu.br. Renato Machado, Universidade Federal de Santa Maria, Santa Maria-RS, Brasil, E-mail: renatomachado@ieee.org.

em endereços IP. Devido a natureza do *software* (GNU *General Public License*), seu código-fonte é disponibilizado gratuitamente, além de já estar presente em grande parte dos sistemas Linux, na forma de um pacote pré-compilado. Para a instalação do pacote Asterisk em sistemas Linux deve-se adotar os seguintes passos, utilizando um terminal como administrador: i) Atualizar o sistema; ii) Instalar os pacotes de dependências; iii) Instalar o Asterisk; iv) Instalar os pacotes de protocolos (e.g., SIP, IAX e H.323).

Com o servidor instalado, deve-se proceder com a sua configuração para que ele possa gerenciar as requisições de conexão. Para tanto, deve-se editar os arquivos de configuração “SIP.conf” (caso o protocolo utilizado seja o SIP) e “extensions.conf”. No fedora, esses arquivos ficam em “/etc/Asterisk”. No arquivo SIP.conf, são inseridos parâmetros como: porta onde o servidor irá “escutar”; IP (tanto interno quanto externo) onde o servidor estará ativo; se o servidor está atrás de um NAT (*Network Address Translation*); intervalo de portas para o protocolo RTP; *codecs* de áudio e vídeo; cadastro dos ramais com usuário e senha e se o usuário pode efetuar e receber, ou só receber ou só efetuar ligações. Já no arquivo extensions.conf são incluídos os ramais previamente cadastrados no arquivo SIP.conf para que estes consigam realizar e/ou receber ligações [6]. Depois de feitas as configurações nos arquivos, é necessário reiniciar o servidor.

Se o sistema operacional do servidor estiver atrás de um *firewall*, é preciso abrir as portas necessárias para o funcionamento do servidor e dos protocolos. No roteador e no modem (conexão ADSL) da rede, é preciso configurar o redirecionamento de portas. As portas a serem desbloqueadas e redirecionadas são a 5060 UDP (porta padrão do servidor Asterisk) e o intervalo de portas de 10000 a 20000 UDP (portas para o protocolo RTP). Essas portas podem ser trocadas de acordo com a necessidade, basta editá-las no arquivo de configuração do protocolo SIP.

III. RESULTADOS

Durante os testes, foi utilizado o *softphone* cliente Zoiper, o qual possui baixa complexidade para o usuário, além de ser gratuito e um dos *softwares* mais utilizados em sistemas VoIP. Para utilizar o Zoiper, basta cadastrar uma nova conta SIP, H.323 ou IAX (de acordo com o protocolo utilizado) com os dados de usuário, senha e ip cadastrados no servidor Asterisk. Finalizados os passos anteriores, dado que tudo foi configurado corretamente, o servidor está disponível para atender às requisições dos clientes. Para efetuar uma chamada, basta digitar o número do ramal (previamente cadastrado nos arquivos de configuração) desejado no cliente Zoiper e clicar em *CALL*. A Fig. 1 mostra a sinalização SIP, o estabelecimento da conexão entre dois ramais (6600 e 6601) e o fechamento da conexão. Além disso, a Fig. 1 apresenta a alocação dos canais de comunicação, realizada pelo servidor. A Fig. 2 mostra, através do comando “*sip show peers*”, os usuários que estão conectados ao servidor com seus respectivos IP’s, ramais, portas utilizadas e latência.

Analisando a Fig. 1, percebe-se, nas linhas 1 e 2, que a chamada utilizará o protocolo RTP, além de *codecs* de vídeo.

```

== Using SIP VIDEO CoS mark 6
== Using SIP RTP CoS mark 5
-- Executing [6600@grupo1:1] Dial("SIP/6601-00000000", "SIP/6600,25") in new
stack
== Using SIP VIDEO CoS mark 6
== Using SIP RTP CoS mark 5
-- Called SIP/6600
-- SIP/6600-00000001 is ringing
-- SIP/6600-00000001 answered SIP/6601-00000000
-- Channel SIP/6600-00000001 joined 'simple_bridge' basic-bridge <d7b859b5-f
f59-4fcc-b1f4-806dbb4e09f8>
-- Channel SIP/6601-00000000 joined 'simple_bridge' basic-bridge <d7b859b5-f
f59-4fcc-b1f4-806dbb4e09f8>
-- Channel SIP/6600-00000001 left 'simple_bridge' basic-bridge <d7b859b5-ff5
9-4fcc-b1f4-806dbb4e09f8>
-- Channel SIP/6601-00000000 left 'simple_bridge' basic-bridge <d7b859b5-ff5
9-4fcc-b1f4-806dbb4e09f8>
== Spawn extension (grupo1, 6600, 1) exited non-zero on 'SIP/6601-00000000'

```

Fig. 1. Sinalização SIP, estabelecimento e fechamento de conexão entre ramais.

```

fedora22*CLI> sip show peers
Name/username      Host                               Dyn Forcerport
Comedia            ACL Port      Status      Description
6600/6600          58667        OK (1 ms)   192.168.0.100    D Yes
Yes
6601/6601          5060         OK (72 ms)  192.168.1.100    D Yes
Yes
6602/6602          48113        OK (100 ms) 201.66.206.146  D Yes
Yes
6603/6603          0            (Unspecified) UNKNOW          D Yes
Yes
4 sip peers [Monitored: 3 online, 1 offline Unmonitored: 0 online, 0 offline]

```

Fig. 2. Clientes conectados ao servidor e suas respectivas informações.

Isso significa que um dos clientes (ou ambos) suporta(m) *streaming* de vídeo. As linhas 3 e 4 sinalizam o começo da nova chamada, mostrando os ramais envolvidos e o grupo ao qual pertencem. As linhas 7 e 8 sinalizam que o ramal 6600 está sendo chamado através do protocolo SIP. A linha 9 mostra que o ramal 6600 atendeu a ligação do ramal 6601. As demais linhas, com exceção da última, sinalizam a alocação de canais por onde se dará a conexão. Por fim, a última linha sinaliza o encerramento da chamada pelo ramal 6600.

IV. CONCLUSÕES

O artigo apresentou uma metodologia básica para a implementação de um servidor VoIP Asterisk utilizando sistema operacional Linux Fedora. O servidor implementado pode servir uma rede local com confiabilidade, qualidade e estabilidade. Além disso, é importante salientar que a metodologia proposta apresenta baixa complexidade de implementação e baixo custo. Por fim, a partir da metodologia apresentada é possível realizar a expansão do sistema para redes de maior porte, através de pequenas modificações nas configurações e com a contratação de um provedor VoIP. É importante salientar que os arquivos de configuração podem ser obtidos através de contato direto com o autor via e-mail.

REFERÊNCIAS

- [1] T. Keating, “Internet phone release 4,” *Computer Telephony Interaction Magazine*, vol. 1, no. 1, pp. 28–32, Jul 1996.
- [2] C. Booth, *Hope, Hype and VoIP: Riding the Library Technology Cycle*, 1st ed., ser. Library technology reports. ALA TechSource, 2010, vol. 16, no. 5.
- [3] J. T. Inc., *Network Protocols Handbook*, 2nd ed. Javvin Technologies Inc, 2005.
- [4] T. Carmona, “VoIP com asterisk,” *Linux Pocket Pro magazine*, Out 2008.
- [5] N. E. M. A. Munadi, R and R. M. Rumani.
- [6] W. A. L. Alexander, A. L. and K. R. K., “Implementing a VoIP SIP server and user agent on a bare PC,” in *Second International Conference on Future Computational Technologies and Applications*, Nov 2010, pp. 8–13.