

Gerenciamento de recursos de serviços de Voz sobre IP baseado em Redes Definidas por *Software*

Paulo Roberto Vieira Jr ¹, Adriano Fiorese ¹, Guilherme P. Koslovski¹,
Anderson H. S. Marcondes¹

¹Programa de Pós-Graduação em Computação Aplicada (PPGCA)
Universidade do Estado de Santa Catarina (UDESC) — Joinville, SC — Brasil

paulorvj@gmail.com, {adriano.fiorese, guilherme.koslovski}@udesc.br
anderson.marcondes@sfs.ifc.edu.br

Abstract. *The Voice Over IP technology (VoIP) allows people to communicate voice using the Internet as a means of transmission. VoIP calls between clients using the same Codec consume communication and computational resources from VoIP server. Software Defined Networking (SDN) has been disseminating the decoupling between control and data plane in network technologies. Therefore, this paper aims to present a management approach based on SDN to reduce the use of resources in a VoIP server. Scenarios were emulated with and without the use of SDN to assess CPU utilization and bandwidth used in a VoIP server. The experimental results obtained indicate reduction on the consumption of the mentioned resources.*

Resumo. *A tecnologia Voz sobre IP (VoIP) permite que pessoas possam se comunicar com voz utilizando a Internet como meio de transmissão. Chamadas VoIP entre clientes que possuem o mesmo Codec consomem recursos de comunicação e computacionais dos servidores VoIP. Software Defined Networking (SDN) difundiu o desacoplamento entre o plano de controle e de dados. Assim, este trabalho apresenta uma solução baseada em SDN para reduzir a utilização de recursos em um servidor VoIP. Foram emulados cenários com e sem o uso de SDN, quantificando a utilização de CPU e largura de banda em um servidor VoIP. Resultados experimentais obtidos indicam redução no consumo dos recursos citados.*

1. Introdução

As redes de comutação de pacotes, inicialmente projetadas para realização de tarefas sem requisitos de qualidade de serviço, como tráfego de correio eletrônico e transferência de arquivos, foram rapidamente difundidas. Um grande número de *switches*, roteadores, *firewalls* e outros equipamentos foram e ainda são implantados, desenvolvidos por diversos fornecedores. Usualmente, cada equipamento deve ser configurado pelo administrador através de comandos de baixo nível no *software* embarcado no equipamento. Esse crescimento tornou as redes fechadas, proprietárias e verticalmente integradas, ossificando a infraestrutura [Handley 2006] e limitando a experimentação de novos protocolos e soluções de gerenciamento [McKeown et al. 2008].

Os protocolos inicialmente propostos cumpriram seu papel para o objetivo traçado. Entretanto, no contexto atual, as redes de computadores desempenham papéis

que vão além do inicialmente projetado e são essenciais para empresas, universidades, centros de pesquisa, ambientes governamentais e atividades diárias. Além do tráfego inicialmente previsto, as redes transportam pacotes com dados oriundos de diversas fontes, como vídeos, voz e outras aplicações que requerem qualidade de serviço (QoS, *Quality of Service*) durante sua execução. Em resposta às limitações da infraestrutura física de redes tradicionais, surgiu o paradigma das Redes Definidas por *Software* (*Software Defined Networking* - SDN), quebrando a integração vertical e separando a camada de dados da camada de controle. Assim, novas ideias e protocolos podem ser experimentados sem prejudicar o tráfego de produção [Jain and Paul 2013] [Kreutz et al. 2014].

O restante do trabalho está organizado da seguinte forma: A Seção 2 apresenta o embasamento teórico para desenvolvimento do trabalho. A Seção 3 discute os trabalhos relacionados, enquanto a Seção 4 apresenta a solução proposta. A Seção 5 apresenta a análise experimental e a Seção 6 as considerações finais.

2. Voz Sobre IP e Redes Definidas por *Software*

A área de telecomunicações sofreu alterações importantes relacionadas com a representação e transporte dos dados de voz. A disponibilização de Voz sobre o Protocolo Internet (*Voice over Internet Protocol*, VoIP) [Goode 2002] vem aproveitando a expansão das redes de comutação de pacotes para permitir o tráfego de voz utilizando a Internet como meio de transmissão. Atualmente, VoIP é utilizado por usuários domésticos e por empresas geograficamente distribuídas, permitindo a comunicação através de diversos dispositivos (*e.g.*, telefones IP, *smartphones*, celulares, *tablets*, *softphone*).

Vários fatores influenciam a qualidade de uma chamada VoIP, como: o protocolo de sinalização, a escolha do codificador/decodificador (*Codec*), atributos de segurança, a habilidade de cruzar *firewalls* e servidores de tradução de endereços (*Network Address Translation* - NAT), entre outros. Sobretudo, fatores como a configuração do servidor VoIP ou a localização dos clientes na rede, fazem com que os pacotes de voz trafeguem pelo servidor VoIP consumindo CPU e largura de banda.

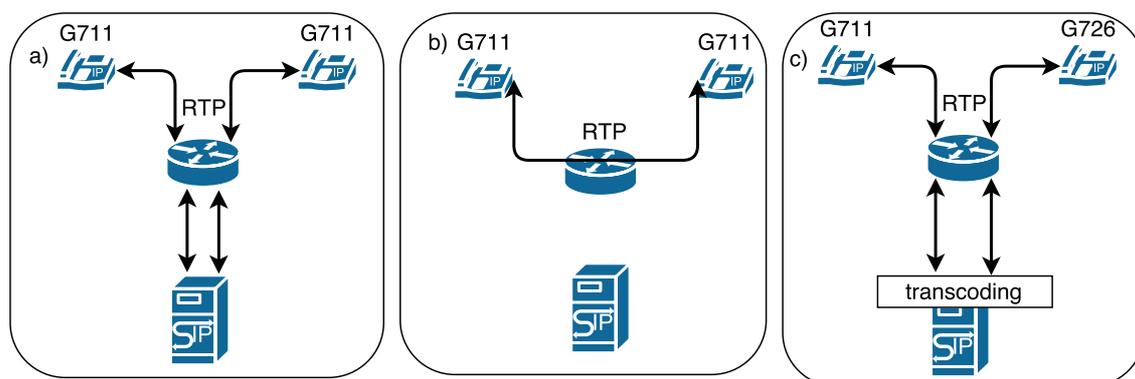


Figura 1. Exemplos de tráfego VoIP com *Codec* único (a e b) e com *Codecs* distintos (c).

A Figura 1 apresenta três maneiras pelas quais pode ocorrer o tráfego de dados em chamadas VoIP: (a) Os dados de uma chamada com o mesmo *Codec* de áudio (G711, por exemplo) são trafegados de um cliente para o outro através do servidor VoIP. Nesse

caso o servidor está no caminho do áudio e dois canais são criados, um para cada cliente. O servidor VoIP lê o áudio de um canal e escreve no outro, aumentando o uso de memória e de CPU durante a leitura e escrita dos canais. Essa configuração é comumente utilizada para monitorar chamadas, coletando informações relacionadas com a qualidade do atendimento prestado por centrais de vendas. (b) O tráfego de sinalização de clientes com o mesmo *Codec* de áudio e os dados de voz são trafegados diretamente entre os clientes. Nesse caso, o servidor VoIP não está no caminho do áudio e não é possível gravar a chamada, acrescentar música de fundo, transferir ou colocar a chamada em espera. Entretanto, essa abordagem reduz o uso de CPU, memória alocada e largura de banda utilizada pelo servidor. (c) Clientes com *Codecs* diferentes e os dados são trafegados sempre pelo servidor VoIP. Semelhante ao cenário (a), o servidor está no caminho do áudio e dois canais são criados, um para cada cliente. Entretanto, o servidor VoIP efetua a tradução dos pacotes de voz de acordo com os *Codecs* utilizados para que os clientes possam receber e ouvir o áudio [Goode 2002], aumentando o consumo dos recursos do servidor.

Clientes localizados em domínios administrativos protegidos por *firewall* ou utilizando NAT aumentam a complexidade do problema. O servidor VoIP não é capaz de resolver os endereços IPs privados dos clientes em uma chamada. Assim, todo o tráfego de voz é baseado nos IPs públicos dos clientes e nesse caso, o tráfego passa em sua totalidade pelo servidor VoIP, consumindo CPU e largura de banda [Karapantazis and Pavlidou 2009, Lin et al. 2010, Park et al. 2008, Chen et al. 2008, Yeryomin et al. 2008, Khlifi et al. 2006, Khirul et al. 2007].

Especificamente, o consumo de largura de banda é uma preocupação quando se discute sobre a adoção da tecnologia VoIP. Empresas com matriz e filiais podem adotar a tecnologia VoIP para reduzir custos com chamadas telefônicas. Porém, muitos planos de utilização da Internet oferecem largura de banda variada e muitas vezes com diferentes taxas para *download* e *upload*. A largura de banda disponível é importante para o provisionamento do serviço influenciando diretamente na quantidade e qualidade de chamadas simultâneas que são suportadas.

Nesse contexto, o presente trabalho tem como objetivo o desenvolvimento de uma solução baseada em SDN que permita reduzir o consumo de recursos na disponibilização do serviço VoIP, aplicável em ambiente LAN. A solução é baseada em SDN e faz com que o tráfego de áudio apresentado na Figura 1 (a) ocorra da maneira apresentada na Figura 1 (b), ou seja, permite que chamadas entre clientes que utilizem o mesmo *Codec* possam ser realizadas com o tráfego de voz fluindo diretamente entre os clientes. Dessa maneira há redução no consumo de CPU e largura de banda no servidor e conseqüentemente no segmento de rede onde o servidor VoIP está localizado.

VoIP é uma tecnologia que permite o tráfego de voz sobre redes comutadas, inclusive na Internet [Karapantazis and Pavlidou 2009]. Dentre as funcionalidades oferecidas pela tecnologia, destacam-se chamadas de longa distância e internacionais, *voice-mail*, identificação de chamadas, conferência, monitoração de conversas, chamada em espera, encaminhamento de chamadas e integração com a rede pública de telefonia (*Public Switched Telephone Network* - PSTN), entre outras. Em suma, oferece serviços com a eficiência de uma rede IP e com a qualidade de uma rede tradicional de telefonia. Além dessas funcionalidades a principal razão para sua adoção é a redução de custos, uma vez que uma única rede suporta ambos os serviços: voz e dados.

O funcionamento de chamadas VoIP é similar a uma chamada telefônica tradicional e envolve sinalização, codificação e decodificação. Uma chamada pode ser iniciada em um telefone IP, no qual a voz é digitalizada e transformada em pacotes através de codificadores. Esses pacotes são encaminhados na rede IP passando por roteadores e *switches* até o servidor VoIP ou diretamente até o destinatário (conforme representando pela Figura 1).

2.1. Estabelecimento de Sessões VoIP

O estabelecimento de conexões e o tráfego de dados VoIP é usualmente regido pelos protocolos *Session Initiation Protocol* (SIP), *Real-time Transport Protocol* (RTP) e *Real-Time Transport Control Protocol* (RTCP). SIP é um protocolo de controle (sinalização) da camada de aplicação que pode estabelecer, modificar e terminar sessões com um ou mais participantes [Rosenberg et al. 2002]. Em suma, qualquer tipo de sessão pode ser estabelecida utilizando o protocolo SIP, como por exemplo distribuição de multimídia e conferências, além de chamadas VoIP. Baseado em UDP, SIP utiliza o protocolo *Session Description Protocol* (SDP) para representar os metadados necessários para o estabelecimento das sessões [G. Camarillo 2006].

Por sua vez, RTP é um protocolo da camada de aplicação que provê entrega de dados fim-a-fim com características de tempo-real, como áudio e vídeo interativo. O pacote RTP carrega, além dos dados de seu *payload* a identificação do tipo de mídia, número de sequência, data e hora e informações para monitoramento da entrega [Schulzrinne et al. 2003]. Utilizando UDP, RTP não garante qualidade de serviço e nem provê um mecanismo para assegurar a entrega temporal da informação. Assim um número de sequência é incluído no pacote RTP permitindo ao receptor a reconstrução da sequência de pacotes quando necessário.

Para estabelecimento de uma chamada de áudio, um par de portas lógicas são utilizadas. Inicialmente, uma porta UDP é direcionada para o serviço RTP para que possa enviar e receber áudio. Uma segunda porta, com numeração em sequência à porta utilizada pelo RTP, é necessária para controle dos pacotes RTP, realizado por meio do protocolo RTCP. O último é utilizado para transmissão periódica de pacotes de controle para todos os participantes da sessão, provendo informações sobre a qualidade da transmissão dos dados.

2.2. Codificação e Decodificação de Voz

Um *Codec* é um algoritmo que permite o transporte de um sinal analógico através de linhas/canais digitais. Existem vários *Codecs* que diferem em complexidade, largura de banda necessária e qualidade de representação da voz. Quanto maior a qualidade obtida pelo *Codec*, maior é a largura de banda necessária para tráfego dos dados [Karapantazis and Pavlidou 2009]. Diferentes *Codecs* podem ser utilizados em uma chamada. Porém, é necessária a transcodificação da chamada, ou seja os dados precisam ser decodificados e recodificados para que o áudio seja ouvido nos terminais da chamada. Este processo consome processamento nos servidores e clientes VoIP, afetando em diversos casos a qualidade percebida da voz [Goode 2002].

A *International Telecommunication Union-Telecommunication* (ITU-T) é o órgão que controla a aprovação de um *Codec*. A ITU-T avalia e atribui uma pontuação ao *Codec*

após um processo de testes. A pontuação é chamada de *Mean Opinion Score* (MOS) com valores de 1 (ruim) até 5 (excelente) [Union 2005b].

2.3. Redes Definidas por Software

SDN é um paradigma de rede emergente que separa a camada de controle da camada de dados, prometendo melhorar a utilização dos recursos de rede, simplificar o gerenciamento de rede, reduzir custos de operação e promover inovação e evolução [Akyildiz et al. 2014]. Em uma arquitetura SDN, o nível mais baixo representa a infraestrutura de rede ou camada de dados, ou seja, nessa camada encontram-se apenas os dispositivos que apenas encaminham o tráfego baseado nas decisões da camada de controle. A camada intermediária é a camada de controle e o elemento chave dessa camada é o controlador, responsável por gerenciar todos os dispositivos da rede, adicionando, removendo ou alterando as entradas nas tabelas de encaminhamento.

A comunicação entre o controlador e os dispositivos encaminhadores pode ser feita através de diversos protocolos. Entretanto, OpenFlow destaca-se como um protocolo aberto que padroniza a comunicação entre as camadas de controle e de dados, definindo como um *software* pode programar a tabela de fluxos em diferentes *switches* [McKeown et al. 2008]. De acordo com a especificação do protocolo [Open Networking Foundation 2012], a arquitetura de um *switch* SDN deve possuir os seguintes componentes: uma ou mais tabelas de fluxos, uma tabela de grupo, um canal de comunicação seguro para um controlador, uma tabela de métricas e um ou mais controladores. Cada tabela contém um conjunto de registros de fluxos e cada fluxo consiste em regras de correspondência, contadores e um conjunto de ações que devem ser aplicadas nos pacotes¹ correspondentes.

O controlador é responsável por adicionar, atualizar e remover registros da tabela de fluxos e pode ser centralizado ou distribuído. Em SDN, as aplicações gerenciam a rede através das interfaces de programação (*Application Programming Interface* - API) definidas pelos controladores. Estas APIs fornecem um nível de abstração que permite às aplicações não depender de detalhes específicos da infraestrutura, facilitando a utilização das funcionalidades disponibilizadas pelo controlador para executarem suas funções adequadamente. Como exemplos de aplicações nesta camada da arquitetura SDN pode-se citar os protocolos de roteamento, balanceamento de carga, monitoração, detecção de ataques, aplicação de políticas e *firewall*.

Para o propósito deste trabalho, um fluxo corresponde à tupla de informações de identificação dos terminais de comunicação VoIP (cliente origem e destino), e deve ser instalado nos *switches* SDN que fazem parte do caminho entre eles. Essas informações referem-se aos endereços MAC, IP e portas lógicas utilizadas na sessão SIP, no tráfego de dados de voz por meio do protocolo RTP e no tráfego de dados de controle por meio do protocolo RTCP.

3. Trabalhos Relacionados

Até a data de escrita deste trabalho não encontramos nenhuma literatura estreitamente relacionada. Apenas trabalhos que de alguma forma melhoram o desempenho de redes VoIP

¹Uma série de bytes compreendendo um cabeçalho, um *payload* e um *trailer*, nessa ordem, tratado como unidade para fins de processamento e encaminhamento. O tipo de pacote padrão é um quadro *Ethernet*, porém outros tipos de pacote também são suportados.

ou trabalhos que dimensionam a carga de um servidor VoIP sem o uso de SDN. Portanto, as opções de comparação de resultados são limitadas. [Maribondo and Fernandes 2016] propuseram um método de adaptação de *Codecs* em redes SDN (*Codec Adaptation over SDN - CAoS*). O trabalho é baseado na arquitetura SDN e segundo os autores implementa uma abordagem para evitar a degradação da voz em redes corporativas. A solução proposta reduz os efeitos do congestionamentos de rede em SDN, reduzindo a qualidade da voz em chamadas para um *Codec* de menor consumo de largura de banda quando ocorre congestionamentos. De acordo com os resultados obtidos, o CAoS permite que uma quantidade maior de chamadas possam ser efetuadas em períodos de muita utilização sem degradar exageradamente a qualidade da voz. A presente proposta pode ser incorporada ao CAoS. Originalmente, os autores não quantificaram ou consideraram a sobrecarga de servidores VoIP, fator que pode degradar o QoS das chamadas.

Em [Lee et al. 2014] é apresentado um *framework* chamado meSDN que habilita virtualização WLAN, QoS orientado as aplicações e melhora da eficiência energética em dispositivos Android. A solução é baseada em SDN e utiliza o Open Virtual Switch (OVS) [Pfaff et al. 2015] para monitorar e gerenciar o tráfego de aplicações móveis. Os autores argumentam que estender as capacidades de uma rede SDN à dispositivos móveis pode prover soluções para muitos problemas de rede como QoS, virtualização e diagnóstico de falhas. A solução proposta na Seção 4 reduz o consumo de largura de banda de servidores VoIP, podendo ser combinado com meSDN para reduzir o tráfego de dados.

Garg *et al.* apresentaram um estudo sobre as limitações das redes sem fio 802.11 (a/b) em suportar chamadas VoIP [Garg and Kappes 2003]. O estudo mostrou a quantidade de chamadas simultâneas que podem ser colocadas em uma célula 802.11 (a/b) da rede. Por exemplo, com a utilização do *Codec* g711 com 20ms de áudio, uma célula 802.11 (a/b) pode suportar somente de 3 a 12 chamadas. Esse fator limitante endossa o desenvolvimento da presente proposta: ao reduzir o tráfego de dados até servidores VoIP, a comunicação em uma célula pode ser utilizado ao máximo para tráfego de dados de chamadas diretamente entre clientes.

Em [Costa et al. 2015] é investigada a adequação do servidor *Private Branch Exchange* (PBX) Asterisk para fornecer capacidades de comunicação VoIP com um MOS aceitável para um grande número de chamadas. A métrica de probabilidade de bloqueio é usado para medir a capacidade do servidor PBX, enquanto o MOS é utilizado para avaliar a qualidade das chamadas de voz. Os resultados do trabalho mostraram que o servidor PBX Asterisk pode efetivamente lidar com mais de 160 chamadas de voz simultâneas com uma probabilidade de bloqueio de menos de 5%, proporcionando chamadas de voz com MOS acima da média 4. Conforme será discutido na Seção 5, o número máximo de chamadas suportadas por um servidor pode ser aumentado com o redirecionamento de tráfego (e consequentemente de carga) com SDN. Ainda, em [Costa et al. 2015], o MOS é medido por uma ferramenta chamada VoIPMonitor, que observa o tráfego VoIP e faz o cálculo do MOS seguindo a recomendação da ITU-T [Union 2005a]. A presente proposta é agnóstica as tecnologias de monitoramento utilizadas, podendo ser conectada com VoIPMonitor.

Por fim, em [Ali et al. 2009] é apresentado um estudo dos requisitos máximos de largura de banda e latência máximo de serviços VoIP em redes de *Worldwide Interopera-*

bility for Microwave Access - WiMAX, incluindo uma análise dos efeitos da compressão de cabeçalho dos pacotes, supressão do cabeçalho do *payload* e outros fatores que afetam a largura de banda. Usando SDN, a aplicação desenvolvida neste trabalho (Seção 4) reescreve informações em pacotes para diminuir o tráfego de dados até o servidor.

4. Módulo de Gerenciamento SDNVoIP

Esta seção apresenta SDNVoIP, uma aplicação para controle de tráfego VoIP em SDN. A aplicação gerencia o tráfego VoIP, reduzindo a utilização de CPU em servidores e a largura de banda utilizada em redes SDN. Para reduzir o uso de CPU nos servidores, SDNVoIP reconfigura as chamadas que utilizem o mesmo *Codec* para realizarem uma comunicação fim-a-fim, sem passar pelo servidor. Ou seja, o cenário da Figura 1(a) é transformado no cenário (b), sem a necessidade de reconfigurar os servidores VoIP.

4.1. Identificação e Encaminhamento de Fluxos

Para o entendimento de um fluxo VoIP baseado nas regras de correspondência das portas da camada de transporte, é necessária a compreensão do mapeamento entre clientes e o servidor VoIP. Assim, o mapeamento de portas em uma chamada normal é apresentado na Figura 2. Por exemplo, o cliente origem e cliente destino comunicam-se na porta SIP 20000 e 15000, respectivamente, com a porta SIP 5060 do servidor VoIP. Nesse caso, todo o áudio trafegado entre os clientes passa pelo servidor VoIP. Uma vez estabelecida a sessão SIP, o servidor recebe na porta RTP 18564 o áudio enviado pelo cliente origem pela porta 25000 e encaminha o áudio recebido pela porta 16181 para o cliente destino que está aguardando na porta RTP 30000. Como as portas RTP e RTCP são definidas em pares, os pacotes RTCP do cliente origem na porta 25001 são recebidos pelo servidor na porta 18565 e encaminhados do servidor na porta 16182 para o cliente destino que aguarda na porta RTCP 30001.

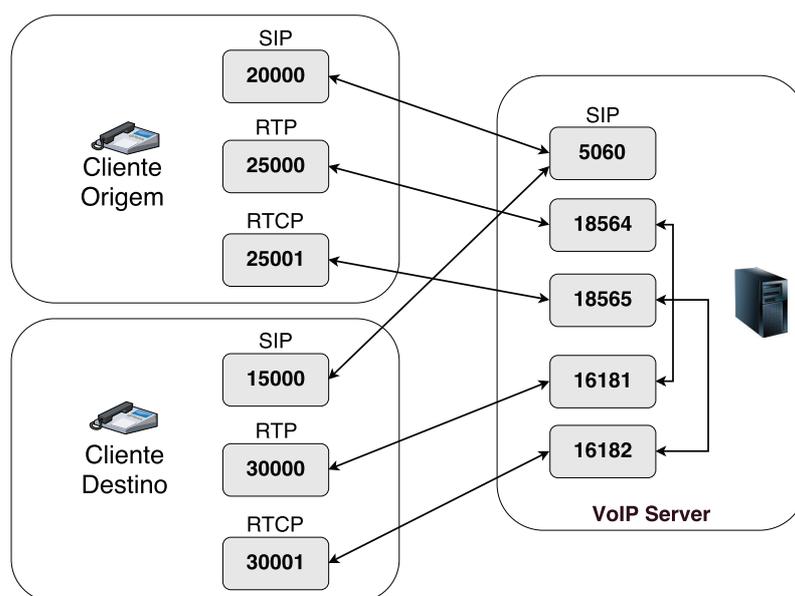


Figura 2. Mapeamento de portas em uma chamada VoIP normal.

Para caracterizar um fluxo de dados, o tráfego SIP, as portas RTP e RTCP utilizadas originalmente na comunicação entre cliente origem-servidor VoIP e servidor VoIP-

cliente destino são descobertas. Da mesma forma, os endereços de enlace e IP envolvidos, são identificados. Essa descoberta subsidia a instalação dos fluxos com as ações. Assim, após o SDNVoIP completar a instalação das entradas nas tabelas de encaminhamento dos *switches*, todo o tráfego RTP e RTCP é direcionado diretamente entre os clientes. Ou seja, os pacotes enviados pela porta 25000 do cliente origem são encaminhados diretamente para a porta 30000 do cliente destino e os pacotes enviados pela porta 25001 do cliente origem são encaminhados para a porta 30001 do cliente destino.

4.2. Implementação do Módulo

SDNVoIP foi desenvolvido como um módulo do controlador Floodlight [Project Floodlight 2016]. Dentre os controladores SDN existentes, Floodlight foi selecionado por apresentar um estágio maduro de desenvolvimento e ser adotado em diversos cenários experimentais e industriais. Quanto a implementação, *Floodlight* é um controlador centralizado desenvolvido em linguagem *Java*, que suporta *switches* físicos e virtuais, baseados no protocolo OpenFlow. O módulo SDNVoIP mantém informações dos clientes que estão executando chamadas, permitindo a instalação dos fluxos nos *switches* OpenFlow no caminho entre eles. Assim, pode-se fazer a instalação dos fluxos em dois momentos: quando o cliente destino envia a mensagem SIP/SDP de resposta para o servidor VoIP, respondendo a chamada feita pelo cliente origem; ou quando inicia o tráfego RTP entre o cliente origem e cliente destino.

Quatro fluxos por chamada são instalados nos *switches* SDN. Dois fluxos para o caminho no sentido origem-destino e destino-origem dos pacotes RTP; e dois fluxos para o caminho no sentido origem-destino e destino-origem dos pacotes RTCP. Para cada fluxo VoIP, sete ações [Open Networking Foundation 2012] são aplicadas nos pacotes RTP e RTCP que trafegam no *switch* correspondente:

- (i) Alterar o endereço MAC de origem para o endereço MAC do servidor VoIP;
- (ii) Alterar o endereço MAC de destino para o endereço MAC do cliente destino;
- (iii) Alterar o endereço IP de origem para o endereço IP do servidor VoIP;
- (iv) Alterar o endereço IP de destino para o endereço IP do cliente destino;
- (v) Alterar a porta RTP/RTCP de origem para a porta RTP/RTCP do servidor VoIP;
- (vi) Alterar a porta RTP/RTCP de destino para a porta RTP/RTCP do cliente destino;
- (vii) Saída do pacote em uma porta física do *switch*.

As entradas nas tabelas de fluxos devem ser instaladas nos *switches* que compõem o caminho da chamada com um tempo de expiração. Quando o término de uma chamada é sinalizado pelos clientes, a entrada é removida. Em caso de interrupção abrupta da chamada, as entradas relacionadas são automaticamente removidas dos *switches* após o tempo de expiração.

5. Análise Experimental

Esta seção apresenta a análise experimental realizada com SDNVoIP. Duas métricas são coletadas: o consumo de CPU nos servidores VoIP e a largura de banda consumida para tráfego de chamadas VoIP.

5.1. Ambiente de Testes

Para o desenvolvimento deste trabalho foram utilizados os seguintes recursos: (i) Como controlador foi utilizado o Floodlight em sua versão 1.2 e algumas modificações em seu código foram realizadas para que os protocolos SIP, RTP e RTCP fossem manipulados pelo módulo SDVoIP. (ii) Como servidor VoIP foi utilizado o Asterisk versão 1.13, executado em um computador com processador Intel QuadCore de 3.2GHz com 8GB de memória RAM. Para gerar a carga (chamadas VoIP) no servidor VoIP foi utilizado o *softphone* Linphone versão 3.6.1. (iii) O *switch* utilizado foi o TP-Link WR1043 com *firmware* personalizado com o Open Wireless Router (OpenWRT) versão 15.01 e Open Virtual Switch (OVS) versão 2.3. A versão 1.3 do OpenFlow foi a utilizada.

5.2. Cenários Experimentais

Dois cenários foram utilizados para realização dos experimentos, com e sem a utilização de SDN, ambos em ambiente LAN.

Chamadas VoIP sem SDN

O cenário sem SDN, apresentado na Figura 3, possui o servidor VoIP, o *switch* TP-Link WR1043 (com *firmware* nativo) e um PC onde foram inicializadas as instâncias do *softphone* Linphone. Um *script* auxiliar faz a inicialização do *softphone* e recebe como parâmetro a quantidade de instâncias que devem ser executadas no *Linux Host*. Cada instância do *softphone* possui um cliente com nome de usuário *u* seguido por um número sequencial, assim os clientes possuem como nome de usuário *u1*, *u2* e assim sucessivamente até o limite informado como parâmetro.

As chamadas são geradas por outro *script*, de acordo com a seguinte lógica: cliente *u51* realiza uma chamada para o cliente *u1*, cliente *u52* faz uma chamada para o cliente *u2* e assim até o cliente final, *u100* chamar o cliente *u50*. Assim, 50 chamadas simultâneas são geradas entre 100 clientes. O *softphone* foi configurado para atender às chamadas recebidas automaticamente e logo após o atendimento deverá realizar a reprodução de um arquivo de áudio. Todas as chamadas utilizaram o mesmo *Codec* (G711) e todo o áudio é trafegado pelo servidor VoIP. Dessa maneira a carga gerada (uso da CPU) no servidor VoIP é estabelecida com chamadas simultâneas entre dois clientes, trafegando áudio e simulando uma conversa em que os dois participantes falam e ouvem.

Chamadas VoIP com SDN

O cenário com SDN é similar ao cenário apresentado na Figura 3. As únicas diferenças são a adição do controlador Floodlight e a reconfiguração do *switch* TP-Link WR1043 com OpenWRT e OVS habilitados e comunicando-se com o controlador (*out-of-band*). Assim como no cenário sem SDN, todo o áudio entre os clientes é trafegado pelo servidor VoIP. Entretanto, após o controlador obter todas as informações necessárias, quatro fluxos para cada chamada são instalados no *switch*, fazendo com que o áudio das chamadas não trafegue pelo servidor VoIP, fluindo diretamente entre os clientes.

Embora a solução atue em LAN, um cenário como o de empresas com matriz e filiais (ambiente WAN), pode obter benefícios da solução apresentada no presente trabalho. A Figura 4 apresenta uma topologia de rede de uma empresa que possui uma matriz e uma filial. Uma chamada entre um cliente origem (CO) e um cliente destino (CD), ambos localizados na Filial 1, possui uma rota passando pelos *switches* F, A e D. Em uma rede não

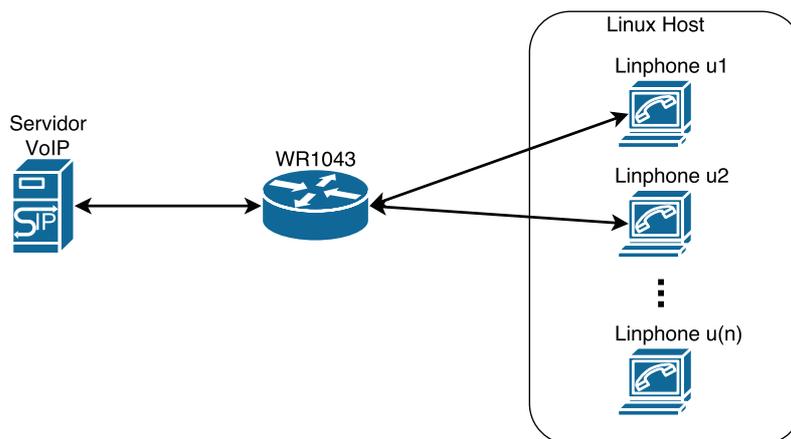


Figura 3. Cenário experimental sem SDN.

SDN, todo o tráfego de voz de CO para CD e vice-versa irá trafegar pelo servidor VoIP localizado na matriz, aumentando a utilização de CPU e consumindo largura de banda.

Já com implantação de *switches* OpenFlow e a utilização do módulo SDNVoIP, pode-se reduzir a utilização de CPU do servidor VoIP e o consumo de largura de banda no *link* Internet, mantendo-se uma arquitetura centralizada para o servidor VoIP, facilitando sua manutenção em termos de configuração. O módulo detecta a chamada feita entre CO e CD e instala os fluxos necessários no *switch* F fazendo com que o tráfego de voz ocorra diretamente entre os clientes na filial 1. Em suma, em ambiente WAN, pode-se obter redução na utilização de CPU no servidor VoIP e redução na utilização da largura de banda dos *links* de interconexão, fazendo com que o tráfego de voz ocorra somente na subrede onde estão localizados o CO e CD, quando for o caso.

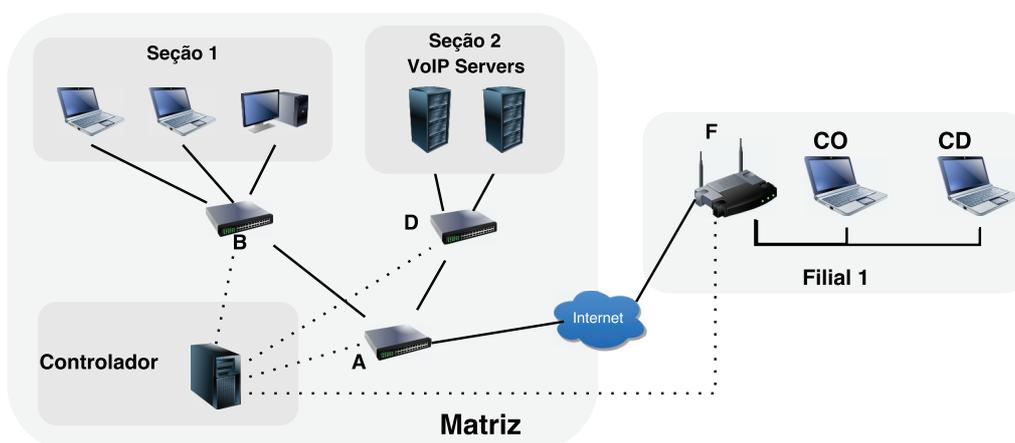


Figura 4. Cenário experimental com SDN: chamada entre clientes na Filial 1.

5.3. Análise dos Resultados

A taxa de utilização de CPU foi coletada com a ferramenta *mpstat* em intervalos de 1 segundo. Foram experimentados 5 minutos de conversação em chamadas simultâneas e os experimentos repetidos dez vezes, sobre os quais calculou-se a média aritmética e desvio padrão. A Tabela 1 apresenta os dados obtidos a respeito da utilização de CPU em

um servidor VoIP com as respectivas cargas. Observa-se que 300 chamadas simultâneas utilizando o mesmo *Codec* e todo o tráfego passando pelo servidor VoIP, gerando uma taxa de utilização de CPU de aproximadamente 27%. Já com o uso de SDN, todo o tráfego gerado pelas chamadas é encaminhado diretamente entre os clientes, fazendo com que a CPU do servidor VoIP fique aproximadamente 99% do tempo ociosa.

Tabela 1. Comparação do uso de CPU de um servidor VoIP.

Sem SDN						
Chamadas simultâneas	50	100	150	200	250	300
Uso de CPU%	4,03	9,22	13,92	19,35	22,78	27,92
Variância	0,53	5,59	1,33	1,04	1,37	2,74
Desv. Padrão	0,13	0,37	0,21	0,17	0,21	0,30
Com SDN						
Uso de CPU%	0,75	0,73	0,71	0,67	0,71	0,66
Variância	0,01	0,00	0,00	0,00	0,00	0,00
Desv. Padrão	0,08	0,01	0,03	0,06	0,03	0,02

A Figura 5 apresenta graficamente a taxa de utilização de CPU. O eixo x mostra a quantidade de chamadas simultâneas, enquanto que o eixo y apresenta a utilização de CPU em %. As barras verticais apresentam em preto a taxa de utilização de CPU sem o uso de SDN, enquanto que as barras na cor cinza apresentam a utilização de CPU com o uso de SDN. A solução com o uso de SDN reduz a utilização de CPU. Dessa maneira a CPU permanece mais tempo disponível para outras tarefas como a transcodificação entre clientes que efetuam chamadas com *Codecs* diferentes. Quanto maior a quantidade de chamadas, maior é o consumo de CPU. A solução apresentada neste trabalho com o uso de SDN pode reduzir o uso CPU para cargas maiores, desde que os clientes possuam o mesmo *Codec*.

A Tabela 2 apresenta as métricas de largura de banda sem o uso de SDN coletadas no servidor Asterisk utilizado, sendo rx a vazão de recepção, $rx\ pps$ a taxa de recepção de pacotes, em pacotes por segundo, tx a vazão de transmissão e $tx\ pps$ a taxa de transmissão em pacotes por segundo. Observa-se que para 50 chamadas simultâneas a vazão típica de recepção e transmissão é de 1,03Mbps e a vazão máxima de recepção foi de 1,04Mbps e 1,05Mbps para a transmissão. As taxas típicas de recepção e transmissão de pacotes foram de 5.09Kbps e 5.04Kbps respectivamente.

Tabela 2. Largura de banda em um servidor VoIP sem o uso de SDN.

Chamadas	50	100	150	200	250	300
Rx	1,03Mbps	2,05Mbps	3,07Mbps	4,10Mbps	5,23Mbps	6,05Mbps
Rx pps	5,10Kbps	10,13Kbps	15,19Kbps	20,23Kbps	25,84Kbps	28,44Kbps
Tx	1,03Mbps	2,07Mbps	3,10Mbps	4,14Mbps	5,28Mbps	6,11Mbps
Tx pps	5,04Kbps	10,08Kbps	15,12Kbps	20,16Kbps	25,73Kbps	28,32Kbps
Max Rx	1,04Mbps	2,05Mbps	3,09Mbps	4,11Mbps	6,15Mbps	6,48Mbps
Max Tx	1,05Mbps	2,07Mbps	3,12Mbps	4,14Mbps	6,20Mbps	6,54Mbps

Já para 300 chamadas simultâneas observa-se que a vazão típica de recepção e transmissão é de 6,05Mbps e 6,11Mbps respectivamente, e a vazão máxima de recepção

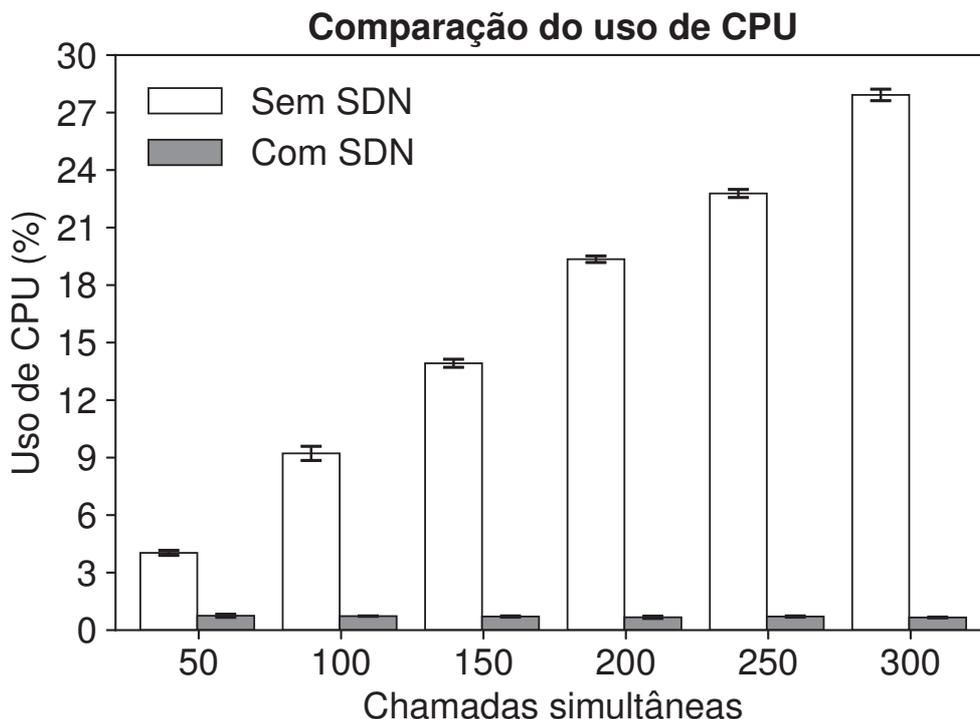


Figura 5. Comparação do uso de CPU em um servidor VoIP.

foi de 6,48Mbps e 6,54Mbps para a transmissão. As taxas de recepção e transmissão de pacotes foram de 28,44Kbps e 28,32Kbps respectivamente. A solução baseada em SDN proposta no presente trabalho reduz no maior cenário experimentado (300 chamadas com duração de 5 minutos), 6,05Mbps e 6,11Mbps de largura de banda de recepção e transmissão, respectivamente no segmento de rede do servidor VoIP.

6. Considerações Finais

O presente trabalho buscou agregar as vantagens e benefícios do uso de SDN na área de telecomunicações através da proposição do módulo SDNVoIP para o gerenciamento de chamadas VoIP em ambientes de rede de computadores que suportem OpenFlow. Os resultados obtidos demonstram que é possível reduzir a utilização de CPU em um servidor VoIP através da criação/utilização de fluxos que redirecionam o tráfego de voz diretamente entre as partes envolvidas que utilizam o mesmo *Codec*. Da mesma forma, há também a redução no consumo de largura de banda no segmento de rede do servidor VoIP.

A redução da utilização de CPU permite que haja maior disponibilidade de processador para chamadas que precisem de tradução de *Codec*, assim como a redução no consumo de largura de banda pode permitir que mais chamadas sejam completadas. Por fim, é possível redimensionar a quantidade de chamadas suportadas pelos servidores VoIP existentes em uma rede, sem a necessidade de aquisição de outro servidor para suportar mais chamadas.

Como trabalhos futuros, o módulo SDNVoIP pode ser modificado para redirecionar o tráfego de clientes que possuem *Codecs* diferentes e precisam ser transcodificadas,

para um servidor VoIP com menor utilização de CPU. Outra possibilidade é redirecionar o tráfego para um servidor VoIP com melhor localização em um ambiente de nuvem. Por fim, outra linha é o redirecionamento do tráfego de chamadas para o servidor VoIP com um caminho de rede com menos tráfego, evitando congestionamentos.

Agradecimentos

O trabalho foi parcialmente financiado pelos editais de apoio da UDESC e desenvolvido no LabP2D.

Referências

- Akyildiz, I. F., Lee, A., Wang, P., Luo, M., and Chou, W. (2014). A roadmap for traffic engineering in sdn-openflow networks. *Computer Networks*, 71:1–30.
- Ali, A. A., Vassilaras, S., and Ntagkounakis, K. (2009). A comparative study of bandwidth requirements of voip codecs over wimax access networks. In *2009 Third International Conference on Next Generation Mobile Applications, Services and Technologies*, pages 197–203. IEEE.
- Chen, W.-E., Huang, Y.-L., and Chao, H.-C. (2008). Nat traversing solutions for sip applications. *EURASIP Journal on Wireless Communications and Networking*, 2008(1):1–9.
- Costa, L. R., Nunes, L. S. N., Bordim, J. L., and Nakano, K. (2015). Asterisk pbx capacity evaluation. In *Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2015 IEEE International*, pages 519–524. IEEE.
- G. Camarillo, E. (2006). Session description protocol (sdp) format for binary floor control protocol (bfc) streams. RFC 4583, RFC Editor.
- Garg, S. and Kappes, M. (2003). Can i add a voip call? In *Communications, 2003. ICC'03. IEEE International Conference on*, volume 2, pages 779–783. IEEE.
- Goode, B. (2002). Voice over internet protocol (voip). *Proceedings of the IEEE*, 90(9):1495–1517.
- Handley, M. (2006). Why the internet only just works. *BT Technology Journal*, 24(3):119–129.
- Jain, R. and Paul, S. (2013). Network virtualization and software defined networking for cloud computing: A survey. *IEEE Communications Magazine*, 51(11):24–31.
- Karapantazis, S. and Pavlidou, F. N. (2009). VoIP: A comprehensive survey on a promising technology. *Computer Networks*, 53(12):2050–2090.
- Khurul, I., Islam, M. K., and Hasan, K. A. (2007). An efficient approach for nat traversal problem on security of voice over internet protocol. In *10th International Conference on Computer and information technology, 2007*, pages 1–5. IEEE.
- Khlifi, H., Gregoire, J., and Phillips, J. (2006). Voip and nat/firewalls: issues, traversal techniques, and a real-world solution. *IEEE Communications Magazine*, 44(7):93.
- Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (2014). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76.

- Lee, J., Uddin, M., Tourrilhes, J., Sen, S., Banerjee, S., Arndt, M., Kim, K.-H., and Nadeem, T. (2014). mesdn: mobile extension of sdn. In *Proceedings of the fifth international workshop on Mobile cloud computing & services*, pages 7–14. ACM.
- Lin, Y.-D., Tseng, C.-C., Ho, C.-Y., and Wu, Y.-H. (2010). How nat-compatible are voip applications? *IEEE Communications Magazine*, 48(12):58–65.
- Maribondo, P. D. and Fernandes, N. C. (2016). Avoiding voice traffic degradation in ip enterprise networks using caos. In *Proceedings of the 2016 workshop on Fostering Latin-American Research in Data Communication Networks*, pages 34–36. ACM.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74.
- Open Networking Foundation (2012). OpenFlow Switch Specification. <http://www.cs.yale.edu/homes/yu-minlan/teach/csci599-fall12/papers/openflow-spec-v1.3.0.pdf>.
- Park, C., Jeong, K., Kim, S., and Lee, Y. (2008). Nat issues in the remote management of home network devices. *IEEE network*, 22(5):48–55.
- Pfaff, B., Pettit, J., Koponen, T., Jackson, E. J., Zhou, A., Rajahalme, J., Gross, J., Wang, A., Stringer, J., Shelar, P., Amidon, K., and Casado, M. (2015). The design and implementation of open vswitch. In *Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation, NSDI’15*, pages 117–130, Berkeley, CA, USA. USENIX Association.
- Project Floodlight (2016). Project floodlight: Open source software for building software-defined networks. <http://www.projectfloodlight.org/floodlight>.
- Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E. (2002). Sip session initiation protocol. RFC 3261, RFC Editor.
- Schulzrinne, H., Casner, R., and Frederick, V. J. (2003). RTP: A transport protocol for real-time applications. RFC 3550, IETF.
- Union, I. T. (2005a). Methods for subjective determination of transmission quality. <http://www.itu.int/rec/T-REC-P.800-199608-I/en>.
- Union, I. T. (2005b). Testing requirements and methodology. <https://www.itu.int/en/ITU-T/C-I/Pages/HFT-mobile-tests/Methodologies.aspx>.
- Yeryomin, Y., Evers, F., and Seitz, J. (2008). Solving the firewall and nat traversal issues for sip-based voip. In *Telecommunications (ICT), International Conference on*, pages 1–6. IEEE.