# Técnicas de Processamento de Sinais e Telecomunicações

Henrique Ajuz Holzmann
(Organizador)

Atena Editora
Ano 2019

**Henrique Ajuz Holzmann**

(Organizador)

# Técnicas de Processamento de Sinais e Telecomunicações

Atena Editora
2019

2019 *by Atena Editora*

Editora Executiva: Profª Drª Antonella Carvalho de Oliveira
Diagramação: Karine de Lima
Edição de Arte: Lorena Prestes
Revisão: Os Autores





Atena Editora
Ano 2019

**Dados Internacionais de Catalogação na Publicação (CIP)**
**(eDOC BRASIL, Belo Horizonte/MG)**

T255 Técnicas de processamento de sinais e telecomunicações [recurso eletrônico] / Organizador Henrique Ajuz Holzmann. – Ponta Grossa, PR: Atena Editora, 2019.

Formato: PDF
Requisitos de sistema: Adobe Acrobat Reader
Modo de acesso: World Wide Web
Inclui bibliografia
ISBN 978-85-7247-449-8
DOI 10.22533/at.ed.498190807

1. Tecnologia da informação. 2. Telecomunicações. I. Holzmann, Henrique Ajuz.

CDD 338.47

**Elaborado por Maurício Amormino Júnior – CRB6/2422**

Atena Editora
Ponta Grossa – Paraná - Brasil
www.atenaeditora.com.br
contato@atenaeditora.com.br

Atena Editora
Ano 2019

## APRESENTAÇÃO

A obra Técnicas de Processamento de Sinais e Telecomunicações está organizada de maneira a atender a temas atuais sobre a área de telecom e processamento de sinais de maneira sucinta e otimizada, sendo dividido em 17 capítulos sequenciais.

A transmissão de dados juntamente com suas vertentes representa um dos principais pilares para o progresso econômico de uma nação e para o atendimento de inúmeras necessidades da humanidade, estando presente nos mais diversos setores. Desenvolve-la de maneira eficiente é uma busca constante de grandes empresas e pesquisadores, buscando otimizar e agilizar o processo de troca de informações.

Produzir conhecimento nestas áreas é de extrema importância, a fim de gerar desenvolvimento e ampliar possibilidades nos mais diversos campos. Desta forma um compendio de temas e abordagens que facilitam as relações entre temas referentes a comunicação e processamento de sinais em diferentes níveis de profundidade em pesquisas, envolvendo aspectos técnicos, científicos e humanos é trazido nesta obra.

Boa leitura!

Henrique Ajuz Holzmann

## SUMÁRIO

# CAPÍTULO 9

# SERVIÇO DE L2VPN EM REDES DE *BACKBONE* IP: ESTUDO DE CASO DA REDECOMEP-RIO

**Pedro Henrique Diniz da Silva**
Centro Brasileiro de Pesquisas Físicas, Coordenação de Desenvolvimento Tecnológico
RedeRio/FAPERJ, Centro de Engenharia e Operações
Rede Nacional de Ensino e Pesquisa, Ponto de Presença do Rio de Janeiro
Rio de Janeiro - RJ

**Natália Castro Fernandes**
Universidade Federal Fluminense, Departamento de Engenharia de Telecomunicações
Niterói - RJ

**Nilton Alves Jr.**
Centro Brasileiro de Pesquisas Físicas, Coordenação de Desenvolvimento Tecnológico
RedeRio/FAPERJ, Centro de Engenharia e Operações
Rio de Janeiro - RJ

**Márcio Portes de Albuquerque**
Centro Brasileiro de Pesquisas Físicas, Coordenação de Desenvolvimento Tecnológico
RedeRio/FAPERJ, Centro de Engenharia e Operações
Rio de Janeiro - RJ

**RESUMO:** As redes OpenFlow introduzem o conceito de controladores centralizados para gerenciamento do comportamento do encaminhamento dos elementos de rede. Esse conceito permite que diversos serviços em redes de *backbone*, atualmente implementados através de soluções proprietárias, distribuídas pelos diversos elementos e de grande complexidade de operação, sejam simplificados facilitando a operação de rede. Esse trabalho propõe uma nova aplicação para provimento de conexões de Redes Privadas Virtuais de Camada 2 (L2VPN, do inglês *Layer 2 Virtual Private Network*) em redes de *backbone*, para caso de uso do *backbone* acadêmico, de pesquisa e de governo da cidade do Rio de Janeiro (REDECOMEP-Rio). São apresentados os algoritmos implementados através do protocolo OpenFlow e controlador Ryu. A solução proposta é avaliada através de cenários emulados por meio do emulador Mininet e de um estudo de caso executado na rede de produção da REDECOMEP-Rio.
**PALAVRAS-CHAVE:** L2VPN; Redes Definidas por Software; OpenFlow; Controlador Ryu.

**ABSTRACT:** OpenFlow networks introduce the concept of centralized controllers for managing the forwarding behavior of network elements. This concept allows multiple services in backbone networks, currently implemented through proprietary solutions, distributed by the various elements, and of highly complex operation, to be simplified to facilitate network operation. This paper proposes a new application for provision of Layer 2 Virtual Private Networks (L2VPN) in

backbone networks for use case of academic, research, and government backbone network of the city of Rio de Janeiro (REDECOMEP-Rio). We present the algorithms implemented using the OpenFlow protocol and Ryu controller. We also evaluate the proposed solution through emulated scenarios using Mininet emulator and a case study implemented in the production network of Redecomep-Rio.
**KEYWORDS:** L2VPN; Software Defined Networks; OpenFlow; Ryu Controller.

## 1 I INTRODUÇÃO

A RedeRio Metropolitana (REDECOMEP-Rio) (Moraes *et al.*, 2015) é uma infraestrutura de fibras óticas próprias que formam uma rede de alta velocidade para as instituições de ensino, ciência, tecnologia, inovação e governo na cidade do Rio de Janeiro. Apesar de ser um *backbone* que atende instituições acadêmicas, a REDECOMEP-Rio, que opera como uma rede de núcleo IP tradicional puramente roteada, também atende a diversos outros tipos de instituições, as quais incluem empresas ligadas à prefeitura, ao estado e ao governo federal. Além de serviços de conectividade de alta velocidade à Internet e de alta confiabilidade, essas instituições também carecem de serviços que simulem redes privadas em cima dessa rede de núcleo compartilhada entre os diversos afiliados. Esse tipo de rede privada é também conhecido como Rede Privada Virtual (VPN, do inglês *Virtual Private Network*).

Vários tipos de VPNs estão disponíveis para a comunidade atualmente e assim sendo existem alguns tipos de classificação para tais. Um tipo de classificação é a divisão em dois modelos de serviços (Knight e Lewis, 2004): VPNs de camada 2 do modelo de referência TCP/IP (L2VPNs do inglês *Layer 2 Virtual Private Networks*); e VPNs de camada 3 (L3VPNs do inglês *Layer 3 Virtual Private Networks*). As VPNs mais comuns são as de camada 3 (Knight e Lewis, 2004) capazes de fazer com que dois elementos de rede pareçam estar diretamente conectados via uma rede roteada, ainda que existam diversos elementos de rede (roteadores) no meio do caminho. Já as VPNs de camada 2 são conexões ponto-a-ponto que simulam um circuito físico de camada 2. Sua principal vantagem é a transparência do enlace virtual formado ponta-a-ponta, permitindo que qualquer protocolo, e não somente o IP, possa trafegar.

Apesar de cada uma das tecnologias apresentadas oferecer diversas vantagens e desvantagens uma em relação a outra, lidar com a variedade de protocolos para a criação de VPNs torna a gerência e operação da infraestrutura de rede mais complexa para os seus operadores. Essas tecnologias são difíceis de implementar, ou operacionalmente quase impossíveis de realizar em larga escala em redes IP puramente roteadas (Osborne e Simha, 2002), como o caso da REDECOMEP-Rio, devido principalmente a complexidade de configuração dos equipamentos envolvidos.

Com o enfoque em simplificar e otimizar o processo de operação de uma rede de núcleo puramente roteada, propõe-se uma solução de estabelecimento de circuitos L2VPN, denominada de *Virtual Circuits Flow* (VCFlow), que evita a complexidade de

ter protocolos de encapsulamento adicionais para o estabelecimento das conexões ponto-a-ponto, sem necessidade de configurações adicionais nos equipamentos envolvidos e centraliza a configuração das conexões em um ponto único da rede. Essa solução escala naturalmente ao passo que a mesma depende, basicamente, do processo de descoberta de topologia do protocolo OpenFlow.

A aplicação desenvolvida para o estabelecimento de L2VPNs instala regras de fluxos OpenFlow reativamente a cada circuito que passa pelo *backbone*, baseado em um arquivo de configuração centralizado. Essa aplicação visa ser implementada na operação da REDECOMEP-Rio como um serviço na rede de produção.

Na próxima seção, são apresentados os trabalhos relacionados. Na seção subsequente é apresentada a arquitetura proposta de estabelecimento de L2VPNs. Apresenta-se, também, na Seção 4, uma avaliação do protótipo e um estudo de caso na REDECOMEP-Rio. O artigo é concluído na Seção 5.

## 2 I TRABALHOS RELACIONADOS

Nos últimos anos, a comunidade de Redes de Educação e Pesquisa (REN, do inglês *Research and Education Networks*) tem investido no desenvolvimento de diversas arquiteturas, protocolos e *softwares* controladores para suportar os serviços de provisionamento de circuitos virtuais (Rao *et al.*, 2005; Zheng *et al.*, 2005; Bobyshev *et al.*, 2006; Guok *et al.*, 2006; Yang *et al.*, 2006; Katramatos *et al.*, 2007), com enfoque principalmente voltado a projetos de *grids* computacionais para física de altas energias. O projeto *Lambda Station (Bobyshev et al., 2006)*, idealizado pelo *Fermi National Accelerator Laboratory* (FermiLab) e o *California Institute of Technology* (Caltech), foi uma das primeiras iniciativas para provisionamento dinâmico de circuitos para aplicações científicas e utilizava técnicas de *Policy Based Routing* (PBR) para encaminhamento de tráfego entre *clusters* de computadores. Já o projeto *Terapaths* (Katramatos *et al.*, 2007) , financiado pelo Departamento de Energia dos Estados Unidos, propôs a criação de caminhos virtuais fim a fim com garantias de banda, combinando o uso de redes locais (LAN) baseadas em técnicas de *DiffServ* e redes de longa distância (WAN) baseadas em túneis *Layer 2 Virtual Private Network* (L2VPN) *Pseudowire* em redes *Multi Protocol Label Switching* (MPLS). Essa solução se baseava na utilização de técnicas de estabelecimento de circuitos conhecidas como L2VPN, que permitem a criação de túneis virtuais sobrepostos a uma rede IP por exemplo, onde para o seu funcionamento é necessário a utilização de protocolos de encapsulamento. Por consequência, esse tipo de protocolos opera através de cabeçalhos adicionais incorporados aos pacotes tradicionais, o que faz com que mais recursos de rede sejam consumidos para o estabelecimento dos circuitos virtuais fim a fim. Soluções como UltraScience (Rao *et al.*, 2005), também financiado pelo Departamento de Energia dos Estados Unidos, e CHEETAH (Zheng *et al.*, 2005), proposto por pesquisadores da Universidade da Virginia e do Laboratório Nacional de Oak Ridge, se baseavam

no estabelecimento de canais óticos dedicados, também conhecidos como L1VPNs óticos, em redes *Synchronous Optical Networking* (SONET) através do protocolo *Generalized Multi Protocol Label Switching* (GMPLS). Entretanto, esse tipo de implementação carece também de cabeçalhos adicionais para encapsulamento, além de um conjunto de *softwares* e protocolos de sinalização e controle (*e.g.* OSPF-TE e RSVP-TE) que possuem grande complexidade para serem implementados para permitir o provisionamento de canais dedicados de forma dinâmica.

No caso de redes IP convencionais, que não implementam nenhuma técnica para provisionamento de circuitos dinâmicos ou VPNs, seja uma rede de *campus* ou até mesmo um provedor de serviços que não implementa MPLS, muitas das vezes um *software* de controle tem de ser empregado para exercer funções de monitoramento e configuração dos *switches,* roteadores e enlaces, para prover algum tipo de circuito estático. Tipicamente, esse tipo de *software* é implementado utilizando tecnologias proprietárias, permitindo acesso somente aos operadores de rede do provedor, além de carecer muitas das vezes de configurações manuais em vários *switches* e roteadores no caminho (dispositivos esses, em geral, produzidos pelo mesmo fabricante). Além disso, cada tipo de técnica para provisionamento de circuitos virtuais de maneira estática demanda um conjunto de procedimentos específicos para o fornecimento do mesmo serviço, como o caso de protocolos *Layer 2 Tunneling Protocol* (L2TP) e túneis *Generic Routing Encapsulation* (GRE). Isso acaba por aumentar e muito a sua complexidade de implantação.

Atualmente, os serviços de provisionamento de circuitos existentes e em operação, como OSCARS (Guok *et al.*, 2006; Esnet, 2016), adotado pelo *backbone* da *Energy Sciences Network* (ESnet) do Departamento de Energia dos Estados Unidos, e AutoBAHN (Geant, 2017), adotado pela rede de educação e pesquisas da Europa denominada GEANT, se baseiam na utilização de policiamento de tráfego para prover Qualidade de Serviço (QoS) em redes que operam por meio de diferentes tipos de tecnologia, como *Ethernet*, SONET e MPLS por exemplo. Ambas as soluções se baseiam, basicamente, na utilização de protocolos abertos como *Inter Domain Controller Protocol* (IDCP) e *Network Services Interface* (NSI), mas cada um desses serviços disponibiliza um conjunto de *Application Programming Interfaces* (API) desenvolvidas para cada cenário com suas especificidades, o que dificulta sua utilização em outras redes.

Entretanto, novas soluções para provisionamento de circuitos virtuais fim a fim têm surgido mais recentemente a partir da introdução do novo paradigma de Redes Definidas por *Software* (SDN, do inglês *Software Defined Networks*). Esse paradigma reduz a complexidade para implantação de serviços de provisionamento de circuitos virtuais dinâmicos já que permite o desacoplamento do *software* de plano de controle do dispositivo de encaminhamento. Isso faz com que esse *software* de controle, que era integrado fortemente nos equipamentos individuais de rede, possa ser implementado em controladores SDN externos (logicamente centralizados) em vez de nos próprios

*switches,* permitindo a abstração da infraestrutura de rede para as aplicações e serviços de rede (Chaves Filho, 2015). Esse mecanismo de funcionamento permite que os controladores SDN tenham uma visão centralizada da rede e regras de encaminhamento sejam aplicadas aos *switches,* baseando-se em uma classificação de tráfego em fluxos de rede a partir somente das informações dos protocolos tradicionalmente utilizados em LAN e WAN, como *Ethernet*, IP e TCP. No momento atual, o OpenFlow é o padrão de interfaces de SDN para conexão entre controladores e dispositivos de encaminhamento mais amplamente aceito e implementado por fabricantes de dispositivos de rede, que disponibilizam recursos de SDN. Esse padrão prove, basicamente, a especificação para o canal de comunicação entre *switches* e controladores, e é baseado nele que a maioria das soluções/aplicações de rede disponibilizadas pela comunidade se baseiam (Kreutz *et al.*, 2015).

Essa abordagem de serviços de provisionamento de circuitos baseada em SDN/OpenFlow traz como uma de suas principais vantagens a não carência de protocolos de encapsulamento adicionais para estabelecimento de circuitos, visto que podem fazer uso de técnicas de modificação dos cabeçalhos já utilizados por padrão em redes locais, como é o caso do cabeçalho de VLAN. Outra vantagem também muito importante é a possibilidade de que quando esse serviço de provisionamento for utilizado, por exemplo para transferências de arquivos, as perdas de pacotes se mantenham baixas, visto que os recursos computacionais podem ser estritamente atribuídos a fluxos de dados específicos, levando a disponibilização de serviços de *Bandwidth on Demand* (BoD). Exemplos desse tipo de solução de provisionamento baseado em SDN/OpenFlow são o OESS, (Tepsuporn *et al.*, 2015; Globalnoc, 2016), utilizado inicialmente no *backbone* de educação e pesquisas da Internet2 nos Estados Unidos, e o DynPaC (Mendiola *et al.*, 2015), desenvolvido para utilização no *backbone* da GEANT.

Contudo, as soluções baseadas em SDN/OpenFlow como OESS e DynPaC, visam o provisionamento desse tipo de circuitos em redes **integralmente** SDN/OpenFlow. Isso é um problema visto que para o estabelecimento desse tipo de rede é, em geral, necessário a aquisição de novos recursos, tanto de novos enlaces para operar em paralelo a rede de produção IP tradicional quanto de novos equipamentos (*switches*) que suportem tecnologias de SDN/OpenFlow. Uma solução para esse problema é a utilização de redes SDN/OpenFlow **híbridas**, onde o tráfego IP tradicional de propósitos gerais roteado tradicionalmente pode ser encaminhado em paralelo ao tráfego destinado a uma aplicação específica, mas que se baseia nas regras de decisão de SDN/OpenFlow. Esse tipo de solução permite a utilização de equipamentos que já estejam em operação, mas que proveem algumas funcionalidades de SDN/OpenFlow.

Este trabalho, então, visa o desenvolvimento de uma solução para serviços de provisionamento de circuitos virtuais dinâmicos do tipo L2VPN *Ethernet*/VLAN em redes de *backbone* SDN/OpenFlow híbridas já em operação, em que o encaminhamento dos pacotes dos circuitos virtuais compartilha os mesmos recursos de rede já

disponíveis, como enlaces físicos e roteadores/*switches*. Essa solução é uma solução aberta destinada a redes de *backbone* de provedores de serviço de rede em geral (não somente voltada a redes de educação e pesquisa) ou até mesmo para redes de *campus*, que inclui como principal vantagem um serviço de conectividade privada sem custos adicionais e de alta capacidade baseada na agregação das vantagens oferecidas por uma tecnologia bem compreendida, como o *Ethernet*, aliada ao novo paradigma de SDN/OpenFlow.

## 3 I CARACTERÍSTICAS PRINCIPAIS DA APLICAÇÃO PROPOSTA

Uma rede de *backbone* IP consiste de múltiplos roteadores do tipo *Provider Edge* (PE), que conectam o roteador de borda *Customer Edge* (CE) à rede de núcleo do provedor, e roteadores do tipo *Provider* (P), os quais funcionam como um roteador de trânsito na rede de núcleo entre os PE, conforme mostrado na Figura 1.

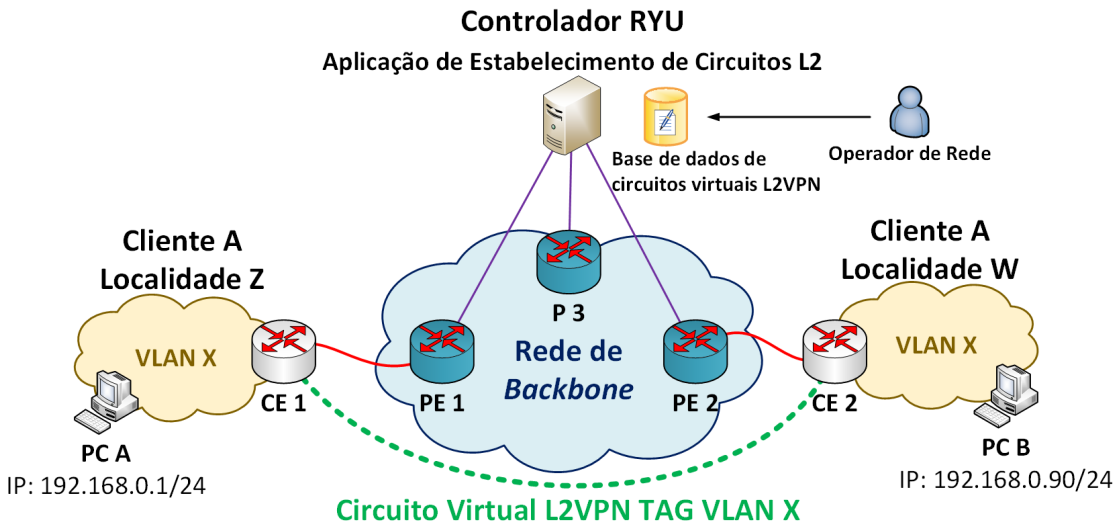



Figura 1: Modelo básico da visão da aplicação de estabelecimento de circuitos L2.

Na solução proposta, os administradores de rede do cliente informam aos operadores de rede do provedor as *tags* de VLAN 802.1q (Ieee (Institute of Electrical and Electronics Engineers), 2014) utilizadas em cada localidade nas quais desejam estabelecer um circuito L2VPN, e os operadores somente precisam armazenar as informações do circuito (como, por exemplo, *tag* de VLAN e roteadores PE envolvidos) em uma base de dados. A aplicação, então, é responsável por direcionar o tráfego de maneira adequada através da rede de núcleo, baseando-se nas informações obtidas através da base de dados e utilizando o algoritmo de escolha de menor caminho *Shortest Path First* (SPF) (Misa e Frana, 2010).

Nesta seção, primeiramente, são descritas as características principais do protocolo OpenFlow utilizadas nessa solução. Em seguida, é descrito como o operador de rede configura sua base de dados de estabelecimento de circuitos L2VPN. Em sequência, é descrito a utilização do algoritmo de menor caminho SPF

para estabelecimento entre os roteadores PE.

**A. Características Principais do Protocolo OpenFlow**

O OpenFlow é um protocolo que permite que tabelas de fluxos em *switches* e roteadores sejam remotamente gerenciadas por um controlador. O protocolo define um fluxo como uma tupla com valores tais como: porta física de entrada, endereço MAC de origem e destino, campo tipo do cabeçalho *Ethernet*, identificador de VLAN, endereços IP de origem e de destino, campo protocolo do cabeçalho IP, porta de origem e destino do protocolo de camada de transporte. O *pipeline* de tabelas de fluxos em um *switch* OpenFlow mapeia a definição de um fluxo através dessa tupla em uma ação a ser tomada nos pacotes que pertencem a esse fluxo. Algumas dessas ações podem ser descartar os pacotes, encaminhá-los em uma porta física específica ou em um conjunto delas, ou enviar os pacotes para o controlador. Em caso de um pacote não corresponder a nenhuma das entradas da tabela de fluxos, o *switch* pode ser configurado para armazenar em um *buffer*, encapsular e enviar o pacote ao controlador para inspeção. Quando o controlador toma a decisão referente ao que fazer com todos os pacotes com aquela característica descrita pela tupla de campos do cabeçalho, ele adiciona uma entrada para esse fluxo na tabela de fluxos para armazenar essa decisão. Além disso, existe também a opção do controlador instalar entradas pró-ativamente para que esse processo não seja novamente repetido. Essas entradas são também conhecidas como regras.

O OpenFlow também implementa um conjunto de mensagens de sinalização e controle trocadas entre o controlador e o *switch*. Essas mensagens são responsáveis por realizar diversas ações como: verificação de características, configuração, modificação de estados, leitura de estados, envio de pacotes e mensagens de barreira. Dentre os tipos de mensagens mais relevantes estão o *packet_in*, o *packet_out* e o *flow_mod*. O *packet_in* é uma mensagem assíncrona enviada do *switch* para o controlador para notificar a chegada de um fluxo não classificado. O *packet_out* é uma mensagem enviada do controlador para o *switch*, em resposta a um *packet_in*, indicando qual ação deve ser tomada para aquele pacote. Já o *flow_mod* é também enviado do controlador para o *switch* para modificar o estado do mesmo, podendo realizar diversos comandos como adição e modificação de entradas na tabela de fluxos.

Em nossa solução de L2VPN, o controlador realiza algumas funções principais como: o procedimento de descoberta de topologia; verificação dos *endpoints* (*switches* PE) do circuito virtual L2VPN a ser estabelecido; escolha do menor caminho entre todos os switches na rede e encaminhamento do pacote na porta do *switch* associada ao caminho entre os *endpoints*; realização do VLAN *stitching*, garantindo a consistência do circuito fim-a-fim. Essas funções serão descritas em mais detalhes adiante.

**B. Verificação dos *Endpoints* e VLAN *Stitching***

A aplicação proposta se baseia nos dados armazenados em duas bases de

dados: (i) gerenciada pelo operador de rede para armazenar a configuração do circuito virtual; (ii) utilizada para o processo de VLAN *stitching* no núcleo da rede. No caso da base de configuração a cada cinco minutos, cada entrada de configuração é checada em sequência em busca de modificações.

Na base de dados de configuração, cada entrada é representada por uma tupla de seis valores: porta *switch* PE 1, *switch* PE 1, VLAN ID de acesso no *switch* PE 1, porta *switch* PE 2, *switch* PE 2 e VLAN ID de acesso no *switch* PE 2.

Para lidar com as situações em que um mesmo cliente utiliza *tags* de VLANs distintas nos locais em que é atendido, e para decisão de qual *tag* é utilizada no núcleo da rede sem que haja sobreposição, outra base de identificação é utilizada. Nela, cada circuito passa a ser identificado por uma tupla de quatro valores: *switch* PE 1, VLAN ID de acesso no *switch* PE 1, *switch* PE 2, VLAN ID de acesso no *switch* PE 2. A partir desses dados, seleciona-se sequencialmente em função das *tags* já em uso no núcleo, qual a próxima a ser utilizada e armazena-se essa informação na base, conforme ilustrado na Figura 2.

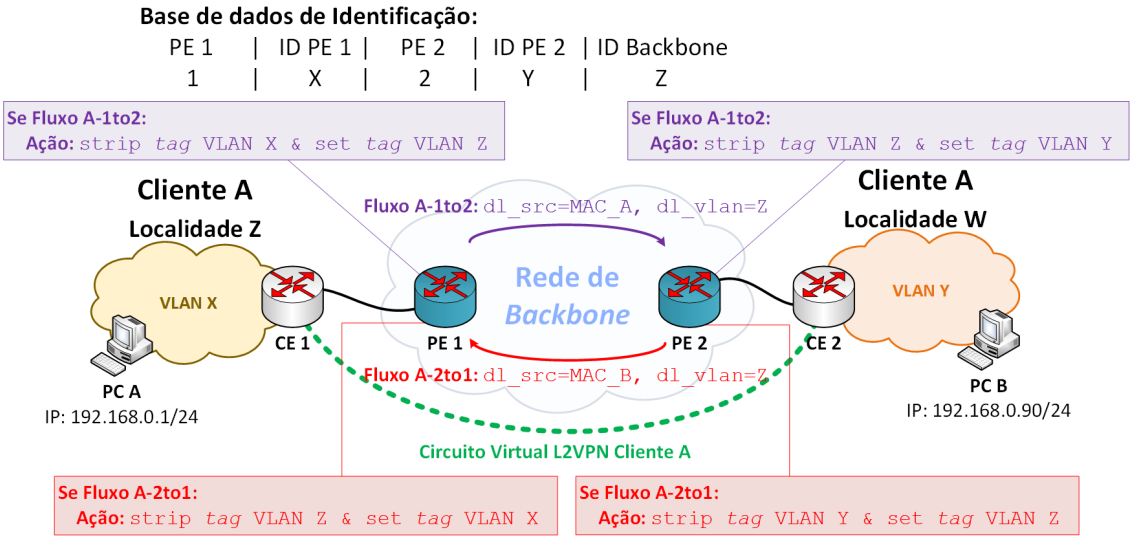



Figura 2: Modelo básico da visão da aplicação de estabelecimento de circuitos L2 para processo de VLAN *stitching*.

### C. Escolha do Menor Caminho e Estabelecimento do Circuito

A cada pacote do tipo *packet_in*, o controlador analisa as informações dos cabeçalhos do pacote, verificando para cada entrada de configuração se: (i) o switch de entrada do pacote é um dos *endpoints* do circuito; (ii) a porta de entrada do *switch* é referente a um dos circuitos; (iii) a *tag* de VLAN do pacote é igual ao VLAN ID de entrada de algum circuito. Caso essas informações se confirmem, o *switch* realiza as ações de associar o endereço MAC de origem do pacote ao *endpoint*, de substituir a *tag* de entrada pela ID de *backbone* do circuito e de encaminhar o pacote. Ao chegar no *endpoint* de destino a operação inversa é realizada. Caso o *switch* que recebe o pacote não seja um *endpoint*, mas a *tag* seja associada a um ID de circuito no *backbone* e o MAC de origem esteja vinculado a um dos *endpoints*, o pacote é encaminhado

para o outro *endpoint*. Então, para estabelecer o caminho entre o *switch* que envia a mensagem de *packet_in* e o *switch* PE de destino do circuito, a aplicação utiliza o algoritmo de *Shortest Path First (Misa e Frana, 2010)*. Dessa forma, a aplicação é capaz de estabelecer o circuito L2VPN fim-a-fim reativamente a cada mensagem de *packet_in*, encaminhando o fluxo à porta de saída de cada *switch*, obtida em função do algoritmo SPF em conjunto com as bases de dados de configuração e de identificação de circuitos. Para evitar que ocorram *loops* na topologia utilizou-se também o protocolo *Spanning Tree* (STP).

A aplicação proposta instala uma regra com *soft-timeout* de 60 segundos para poder lidar com o processo de alteração da configuração de um circuito. Então, após uma mudança por parte do operador de rede, há a necessidade de aguardar: (a) um período de 60 segundos de inatividade para que a entrada da tabela de fluxo de cada *switch* no caminho seja deletada; (b) um período de 5 minutos para que a configuração possa ser verificada novamente e o novo circuito possa ser estabelecido a partir de novas mensagens de *packet_in* - podendo ambos os intervalos (a) e (b) ocorrerem concorrentemente.

## 4 I IMPLEMENTAÇÃO E AVALIAÇÃO DO FUNCIONAMENTO DA APLICAÇÃO

Nesta seção, buscou-se avaliar o tempo de convergência da aplicação visto pelo switch CE, definido como o intervalo de tempo entre o início do funcionamento da aplicação e o momento em que os protocolos STP e SPF já convergiram, ou seja, quando os *switches* PE OpenFlow iniciam o encaminhamento de pacotes baseados nas mensagens de *packet_in* e *packet_out*. Para tal, os resultados foram subdivididos em dois cenários: topologias emuladas e estudo de caso da implantação no *Backbone* REDECOMEP-Rio. A avaliação de desempenho ilustra como o procedimento para o estabelecimento do circuito é eficaz e como o sistema escala adequadamente, aparentando ter pouco *overhead* em função da complexidade da rede. A complexidade, nesse caso, é considerada como o número total de nós, isto é, o número total de *switches* mais *hosts*.

Para validar a proposta nos cenários de topologias emuladas, construiu-se um protótipo utilizando o OpenVswitch (um *switch* OpenFlow via *software*) e o controlador OpenFlow Ryu, por meio do emulador de redes Mininet (Lantz *et al.*, 2010). A aplicação Ryu instala as regras nos *switches* do caminho do circuito L2VPN por meio do protocolo OpenFlow v1.3. Nesse protótipo, todos os procedimentos de verificação do circuito e de escolha de menor caminho foram desenvolvidos conforme descritos nas Seções 3.b e 3.c.

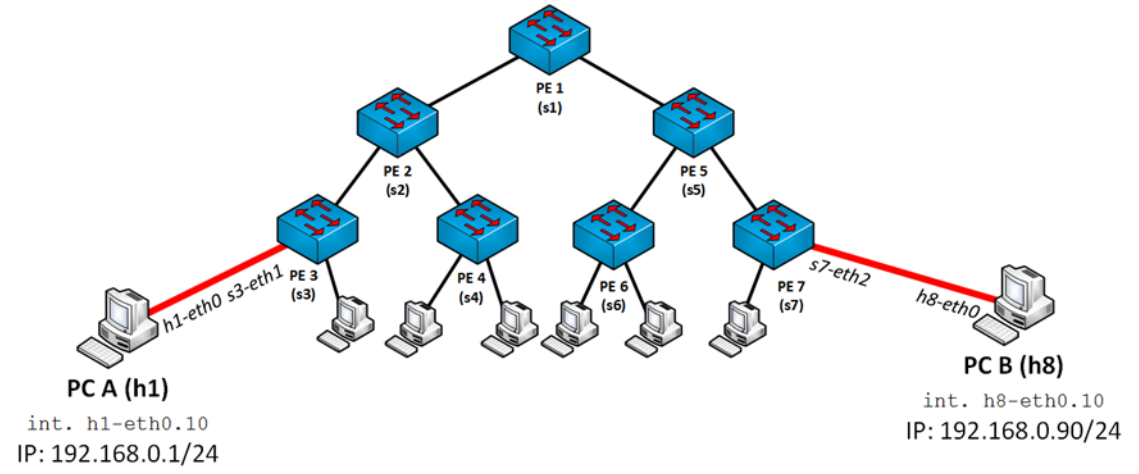
Para avaliar o funcionamento da aplicação, foi medido o tempo de convergência na rede, considerando o uso do algoritmo STP, disponibilizado por padrão no controlador Ryu, com o algoritmo SPF, desenvolvido para a aplicação, e com algoritmo proposto de estabelecimento de conexões L2VPN. Emulou-se o processo de estabelecimento

de uma conexão entre dois *hosts* de um mesmo cliente, assumindo que o circuito já estava previamente descrito no arquivo de configuração. Foi medido o intervalo de tempo entre o primeiro pacote ARP *Request* enviado e sem resposta e o primeiro pacote ARP *Request* que recebe resposta. Os resultados são apresentados com um intervalo de confiança de 95%.

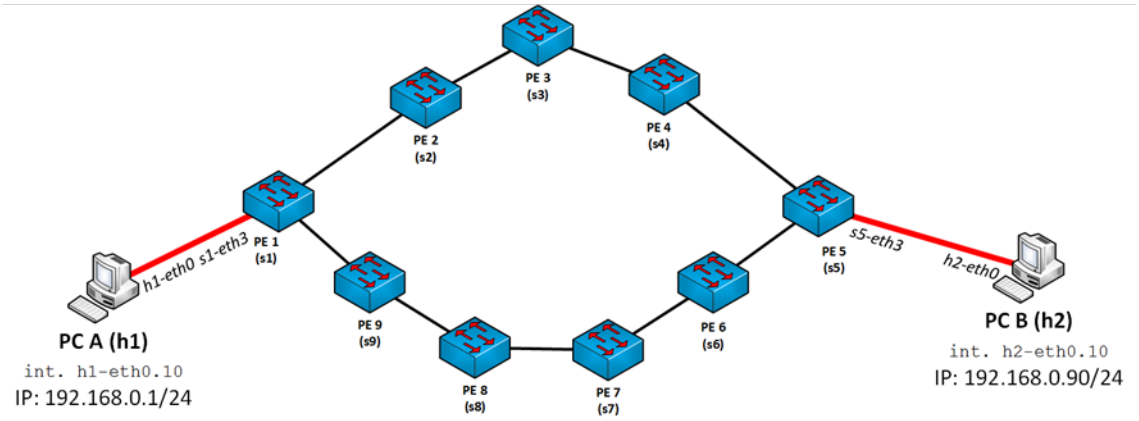
**A. Cenários Emulados**

Para realizar a avaliação do funcionamento da aplicação, emulou-se três diferentes topologias. Utilizou-se duas topologias padrão do Mininet e uma customizada, conforme descritas em detalhes adiante. Para geração dos pacotes para estabelecimento de conexão e geração dos pacotes ARP, utilizamos a ferramenta FPING (Schemers, 2007), com intervalos entre mensagens ICMP *Echo Request* de 10 ms. Os intervalos entre os pacotes ARP foram checados através da ferramenta *tcpdump* (Fuentes e Kar, 2005), capturando pacotes na interface do *host* de origem.

**1. Topologia Linear com 2 *Switches***: Como avaliação inicial, utiliza-se a topologia mais simples disponível no emulador com dois *switches*. Os *switches* são conectados diretamente e dois *hosts* são conectados cada um a um dos *switches*. **2. Topologia *Fat-tree* com Três Níveis e Sete *Switches*:** A topologia *fat-tree* é um tipo de topologia voltada para *data centers*. Realizou-se a avaliação de funcionamento em uma topologia de três níveis com sete *switches* e oito *hosts* conectados de acordo com a Figura 3(a). **3. Topologia Emulada do *Backbone* REDECOMEP-Rio:** Como a aplicação visa ser implementada na rede em produção da REDECOMEP-Rio, optou-se por realizar uma emulação em uma infraestrutura similar à sua topologia real. A topologia de *backbone* da REDECOMEP-Rio, atualmente, é composta por nove Pontos de Presença (PoP, do inglês *Point of Presence*) espalhados por uma topologia em anel com múltiplas redundâncias. A Figura 3(b) exibe o *layout* da topologia da rede em produção simplificada. Foram conectados nove *switches* em anel com dois *hosts*. Nesse cenário, cada *host* faz o papel dos *switches* CE que devem ser conectados aos *switches* PE de *backbone*. Vale ressaltar que como a topologia é em anel, ocorrem *loops* do *Spanning Tree*.



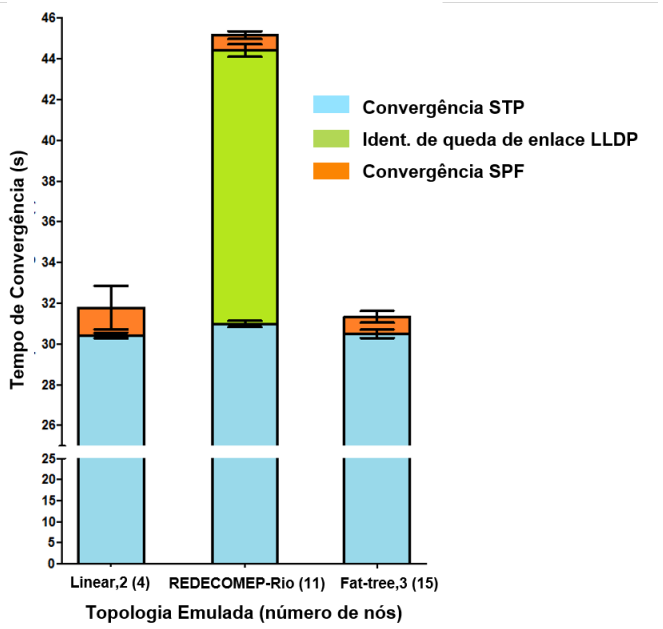

(a) *Fat-tree* com sete *switches* e oito *hosts*.



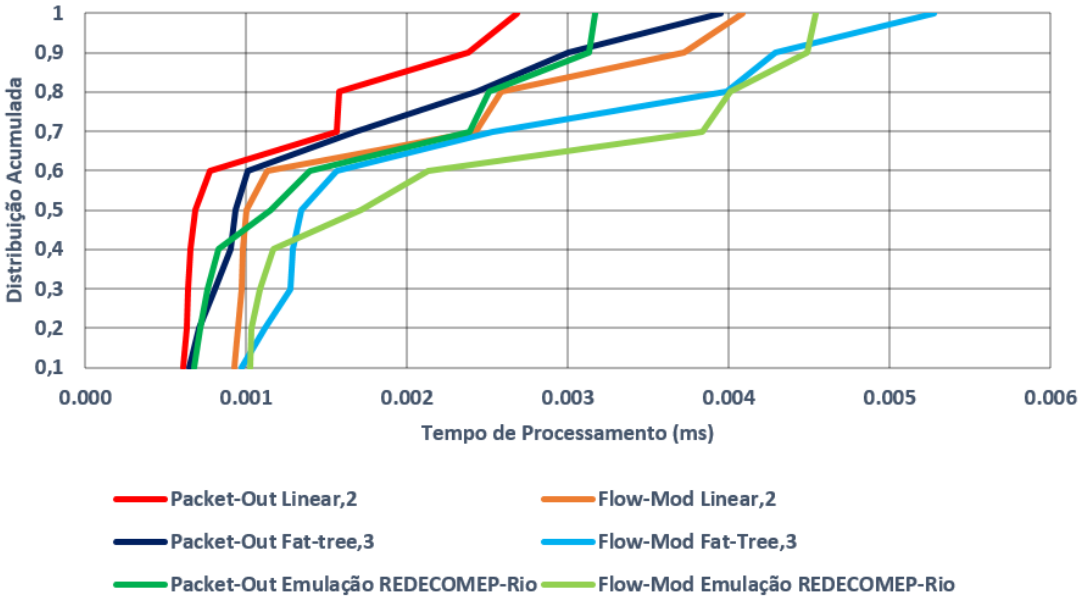

(a) Emulação da REDECOMEP-Rio com nove *switches* e dois *hosts*.

Figura 3: *Layout* das topologias emuladas.

Os resultados referentes aos cenários emulados podem ser visualizados na Figura 4(a). O tempo de convergência é em grande parte dominado pelo tempo de convergência do STP, que para todos os casos apresentados é de 30s. Isso ocorre de acordo com o apresentado em (Ryu, 2016). A partir do instante em que o STP converge a aplicação demora somente 1,5s para iniciar o encaminhamento de pacotes. Assim como os casos em que não ocorrem *loops*, o tempo de convergência é em grande parte dominado pelo tempo de convergência do STP, cerca de 30s. Porém, após uma porta entrar em estado BLOCKED pelo STP, é necessário mais 13,5s para que o LLDP, implementado por padrão no Ryu, identifique que o enlace está bloqueado e a aplicação, a partir de então, possa recalcular os caminhos. Desde o momento em que o enlace passa a estar bloqueado, a aplicação leva cerca de 1s para encaminhar os pacotes.




(a) Tempo total de convergência em relação ao número total de nós na rede.




(b) CDF do tempo de processamento das mensagens de *packet_out* e *flow_mod*.

Figura 4: Resultados obtidos para os cenários emulados.

Cabe ainda ressaltar que a partir do momento em que a aplicação já convergiu, o estabelecimento do circuito depende somente da latência entre os *switches* e do tempo em que as mensagens de *packet_out* e *flow_mod* demoram para serem processadas e enviadas pelo controlador. Como pode ser observado na Figura 4(b), a função de distribuição acumulada (CDF) dos tempos de processamento dessas mensagens demonstra que esse intervalo representa cerca de 0,017% do tempo total de convergência para o pior dos casos de processamento de mensagens *flow_mod*, ou seja, a topologia emulada *fat-tree*. Logo, isso mostra que a partir da convergência da aplicação, o tempo para estabelecimento do circuito é muito pequeno.

**B. Estudo de Caso da Implantação Inicial no Backbone da REDECOMEP-Rio**

Apresenta-se, a seguir, a avaliação do sistema protótipo na REDECOMEP-Rio. O estudo de caso foi executado em uma configuração com equipamentos híbridos, ou seja, realizando o encaminhamento dos pacotes da rede em produção no mesmo enlace que os pacotes SDN/OpenFlow. O cenário descrito foi avaliado utilizando os equipamentos: 2 (PE) x Cisco ASR9000 rodando IOS XR 5.1.3 equipados com pelo menos 6 GB DRAM; 1 (CE) x Cisco Catalyst WS-C3560G-24TS rodando IOS 12.2(50)SE5; 1 (CE) x Cisco Catalyst WS-C3750G-24TS-1U rodando 12.2(25)SEE2. A associação entre os roteadores Cisco ASR9000 e o controlador ocorre *in-band*. Todos os dispositivos são conectados com enlaces de 1 Gbps de capacidade, conforme mostrado na Figura 5. A conexão entre o *switch* CE e o roteador PE ocorre através de uma interface do tipo tronco com permissão de tráfego das VLANs 802.1q 10 e 100.

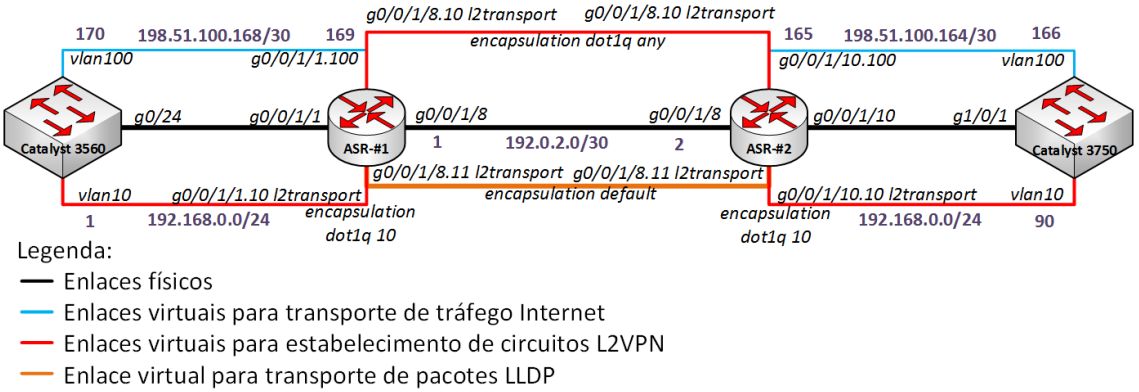



Figura 5: Cenário de estudo de caso da avaliação do funcionamento da aplicação de estabelecimento de circuitos L2VPN no *backbone* da REDECOMEP-Rio.

De acordo com o que pode ser observado na Figura 5, fez-se necessário o estabelecimento de dois enlaces virtuais entre os roteadores PE para o correto funcionamento da aplicação. Além do transporte dos pacotes com *tags* de VLAN 802.1q para o estabelecimento de circuitos, também é necessário o transporte dos pacotes LLDP para o mapeamento da topologia. Desse modo, para cada tipo de enlace virtual foi permitido um determinado tipo de encapsulamento de pacotes. Isto é, para o transporte dos pacotes dos circuitos com encapsulamento dot1q configurou-se uma interface com *encapsulation dot1q any*, em vermelho na Figura 5, enquanto que para os pacotes de controle LLDP sem encapsulamento foi configurado outro tipo de interface com *encapsulation default*, em laranja na Figura 5.

Assim como o exposto na Seção 4.a., o tempo de convergência da aplicação também foi avaliado, *i.e.*, o intervalo de tempo entre o pacotes ARP *Request* inicial e o que recebe resposta foi determinado. Entretanto, em vez de utilizar a ferramenta FPING foi utilizado o utilitário *ping* disponível por padrão nos *switches*, também com intervalo de 10ms entre ICMP *Echo Request*, e para análise dos resultados a ferramenta *tcpdump*. As requisições foram enviadas a partir do *switch* denominado Catalyst 3560 na Figura 5.

A Figura 6 exibe o resultado comparativo entre as CDFs dos tempos de convergência do cenário emulado de topologia linear descrito na Seção 4.a. e do estudo de caso descrito na presente seção. Pode-se observar uma diferença de até 8s entre o menor valor para o cenário emulado e o maior valor do estudo de caso. Essa diferença ocorre, principalmente, devido ao intervalo entre tentativas de associação entre controlador e roteador ASR9000 serem fixadas em 8s, enquanto que o intervalo referente ao OpenVswitch ser de apenas 1s.

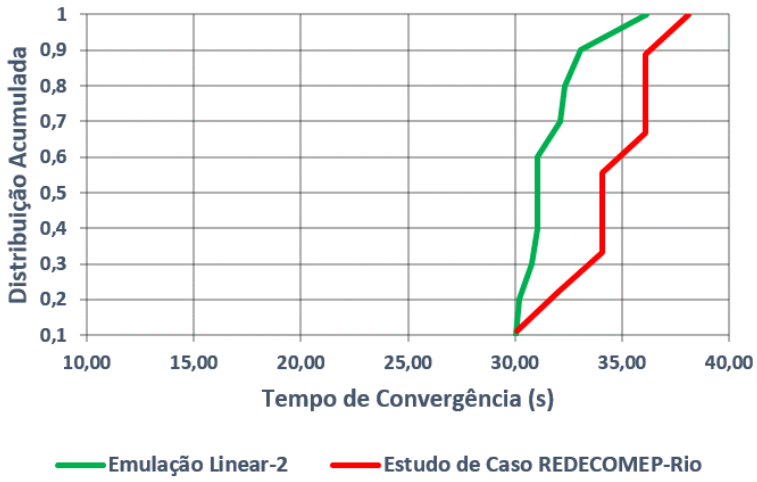



Figura 6: CDF do tempo total de convergência da aplicação para os cenários emulado linear e de estudo de caso.

## 5 I CONCLUSÃO

Redes de *backbone* têm como uma de suas principais demandas o provimento de serviços de estabelecimento de redes virtuais privadas. Um dos principais tipos de redes são as VPNs de camada 2 ou L2VPNs. A solução proposta de estabelecimento desse tipo de circuitos instala regras reativamente nos *switches* da rede utilizando o controlador OpenFlow Ryu. Essa solução se difere, principalmente, das disponíveis no mercado atualmente, pois não é necessário em nenhum momento que o operador tenha conhecimento do funcionamento do protocolo, nem sequer tenha que alterar as configurações dos equipamentos. As avaliações mostram que o sistema escala bem para redes com até quinze nós de *backbone*, dependendo em grande parte do tempo de convergência do protocolo *Spanning Tree*, e se há ou não *loops* na topologia. Esse efeito ocorre devido ao fato de que o protocolo LLDP leva cerca de 13s após a convergência do STP para detectar o bloqueio de um enlace. Além disso, a diferença encontrada entre o cenário emulado e o implantado, mostra que o tempo de associação entre o controlador e o *switch* OpenFlow tem um pequeno impacto no tempo total de convergência da aplicação. Ademais, o impacto dos atrasos entre os *switches*, e entre eles e o controlador no tempo de estabelecimento de circuito é pequeno, devido ao fato dos tempos de processamento de mensagens de *packet_out* e *flow_mod* somados a esses atrasos serem muito pequenos em relação ao tempo

total de convergência. Assim, após o período de inicialização da rede, o tempo para estabelecimento automático da L2VPN é inferior a 10ms, o que é um excelente tempo quando comparado ao processo tradicional para a configuração e estabelecimento desse tipo de circuito.

Portanto, em função dos resultados obtidos, a implementação da aplicação em todo o *backbone* em produção da REDECOMEP-Rio está em fase de finalização, onde o impacto de sua utilização nos usuários da rede deve ser avaliado, posteriormente.

## REFERÊNCIAS


BOBYSHEV, A. et al. **Lambda Station: On-demand Flow Based Routing for Data Intensive Grid Applications over Multitopology Networks.** 2006 3rd International Conference on Broadband Communications, Networks and Systems, 2006, IEEE. p.1-9.

CHAVES FILHO, G. D. A. **Comutação baseada em caminhos: uma solução SDN para problema de migração de máquinas virtuais em Data Centers**. 2015. (Mestrado em Informática). Centro de Ciências Exatas e Tecnologia, Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.

ESNET. **OSCARS: On-Demand Secure Circuits and Advance Reservation System. 2016.** Disponível em: < http://www.es.net/engineering-services/oscars/ >. Acesso em: 20 de fevereiro de 2019.

FUENTES, F.; KAR, D. C. **Ethereal vs. Tcpdump: a comparative study on packet sniffing tools for educational purpose**. Journal of Computing Sciences in Colleges, v. 20, n. 4, p. 169-176, 2005. ISSN 1937-4771.

GEANT. **About AutoBAHN**. 2017. Disponível em: < http://geant3.archive.geant.net/service/autobahn/about_autoBAHN/Pages/About_autoBAHN.aspx >. Acesso em: 20 de fevereiro de 2019.

GLOBALNOC. **OESS: Open Exchange Software Suite.** 2016. Disponível em: < http://globalnoc.iu.edu/sdn/oess.html >. Acesso em: 20 de fevereiro de 2019.

GUOK, C. et al. **Intra and Interdomain Circuit Provisioning Using the OSCARS Reservation System.** 2006 3rd International Conference on Broadband Communications, Networks and Systems, 2006, IEEE. p.1-8.

IEEE (INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS). **IEEE 802: Local and Metropolitan Area Network Standards**. IEEE Standard 802.1q 2014.

KATRAMATOS, D. et al. **The TeraPaths Testbed: Exploring End-to-End Network QoS.** Testbeds and Research Infrastructure for the Development of Networks and Communities, 2007. TridentCom 2007. 3rd International Conference on, 2007, IEEE. p.1-7.

KNIGHT, P.; LEWIS, C. **Layer 2 and 3 Virtual Private Networks: Taxonomy, Technology, and Standardization Efforts.** IEEE Communications Magazine, v. 42, n. 6, p. 124-131, 2004. ISSN 0163-6804.

KREUTZ, D. et al. **Software-Defined Networking: A Comprehensive Survey.** Proceedings of the IEEE, v. 103, n. 1, p. 14-76, 2015. ISSN 0018-9219.

LANTZ, B.; HELLER, B.; MCKEOWN, N. **A network in a laptop: rapid prototyping for software-defined networks.** Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, 2010, ACM. p.19.

MENDIOLA, A. et al. **DynPaC: A Path Computation Framework for SDN.** 2015 Fourth European Workshop on Software Defined Networks, 2015, IEEE. p.119-120.

MISA, T. J.; FRANA, P. L. **An interview with Edsger W. Dijkstra.** Communications of the ACM, v. 53, n. 8, p. 41-47, 2010. ISSN 0001-0782.

MORAES, L. F. M. D.; ALBUQUERQUE, M. P. D.; RIBEIRO FILHO, J. L. **Infraestrutura Redes de Alta Velocidade no Rio de Janeiro: história e estado da arte.** In: (Ed.). A História da Telessaúde da Cidade para o Estado do Rio de Janeiro. Rio de Janeiro: EdUERJ, 2015. ISBN 978-85-7511395-0.

OSBORNE, E. D.; SIMHA, A. **Traffic engineering with MPLS**. Cisco Press, 2002. ISBN 1587050315.

RAO, N. S. et al. **UltraScienceNet: Network Testbed for Large-Scale Science Applications.** IEEE Communications Magazine, v. 43, n. 11, p. S12-S17, 2005. ISSN 0163-6804.

RYU. **Framework Community: Ryu SDN controller.** 2016. Disponível em: < https://osrg.github.io/ryu/ >. Acesso em: 20 de fevereiro de 2019.

SCHEMERS, R. J. **FPING Man Page**. 2007. Disponível em: < http://fping.sourceforge.net/man/ >. Acesso em: 20 de fevereiro de 2019.

SHARAFAT, A. R. et al. **MPLS-TE and MPLS VPNs with OpenFlow.** ACM SIGCOMM Computer Communication Review, 2011, ACM. p.452-453.

TEPSUPORN, S. et al. **A multi-domain SDN for dynamic layer-2 path service.** Proceedings of the Fifth International Workshop on Network-Aware Data Management, 2015, ACM. p.2.

YANG, X. et al. **GMPLS-based Dynamic Provisioning and Traffic Engineering of High-Capacity Ethernet Circuits in Hybrid Optical/Packet Networks.** Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications, 2006, IEEE. p.1-5.

ZHENG, X. et al. **CHEETAH: Circuit-switched High-speed End-to-End Transport ArcHitecture testbed.** IEEE Communications Magazine, v. 43, n. 10, p. S11-S17, 2005.

## SOBRE O ORGANIZADOR

**Henrique Ajuz Holzmann -** Professor da Universidade Tecnológica Federal do Paraná (UTFPR). Graduação em Tecnologia em Fabricação Mecânica e Engenharia Mecânica pela Universidade Tecnológica Federal do Paraná. Mestre em Engenharia de Produção pela Universidade Tecnológica Federal do Paraná Doutorando em Engenharia e Ciência do Materiais pela Universidade Estadual de Ponta Grossa. Trabalha com os temas: Revestimentos resistentes a corrosão, Soldagem e Caracterização de revestimentos soldados.

Agência Brasileira do ISBN
ISBN 978-85-7247-449-8