UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE CASTANHAL
FACULDADE DE COMPUTAÇÃO
CURSO DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

CLÁUDIO ELZIO PAIXÃO DE OLIVEIRA JUNIOR

ENGENHARIA DE TRÁFEGO APLICADO À SIMULAÇÃO DE UMA REDE BACKBONE DE UM PROVEDOR DE INTERNET REGIONAL UTILIZANDO O PROTOCOLO MPLS TE

Castanhal
2022

UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE CASTANHAL
FACULDADE DE COMPUTAÇÃO
CURSO DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

CLÁUDIO ELZIO PAIXÃO DE OLIVEIRA JUNIOR

ENGENHARIA DE TRÁFEGO APLICADO À SIMULAÇÃO DE UMA REDE BACKBONE DE UM PROVEDOR DE INTERNET REGIONAL UTILIZANDO O PROTOCOLO MPLS TE

Trabalho de Conclusão de Curso apresentado para obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. José Jailton Henrique Ferreira Junior

Castanhal
2022

CLÁUDIO ELZIO PAIXÃO DE OLIVEIRA JUNIOR

ENGENHARIA DE TRÁFEGO APLICADO À SIMULAÇÃO DE UMA REDE BACKBONE DE UM PROVEDOR DE INTERNET REGIONAL UTILIZANDO O PROTOCOLO MPLS TE

Trabalho de Conclusão de Curso apresentado para obtenção do grau de Bacharel em Engenharia de Computação.

Banca Examinadora

Prof. Dr. José Jailton Henrique Ferreira Junior
Faculdade de Computação - UFPA
Orientador

Prof. Dr. Tássio Costa de Carvalho
Faculdade de Computação - UFPA
Membro da Banca

Prof. Dr. Igor Ruiz Gomes
Faculdade de Computação - UFPA
Membro da Banca

Castanhal
2022

Este trabalho é dedicado aos meus pais, Ana Maria e Cláudio Elzio. Sem eles eu não teria conseguido chegar até aqui.

AGRADECIMENTOS

Os agradecimentos principais são direcionados aos meus pais, Ana Maria Martins de Oliveira e Cláudio Elzio Paixão de Oliveira, por fazerem o possível e o impossível pelo meu crescimento educacional e profissional.

A minha noiva, Débora Nepomuceno, pelo apoio e companheirismo em todos os momentos.

Ao meu orientador, Prof. Jose Jailton, pelas orientações e conselhos na idealização deste trabalho.

Aos meus amigos de faculdade, Neverton Sousa e Erick Fernandes, por toda ajuda e parceria durante a graduação.

Aos meus colegas de trabalho, Cydiane Oliveira, Ana Carolina e Thiago Gonçalves, por todo incentivo e apoio.

E a todos que colaboraram indiretamente na execução deste trabalho.

RESUMO

As taxas de tráfego nos provedores de internet atuais têm aumentado consideravelmente devido à presença cada vez maior de enlaces de fibra óptica. Com este cenário, os desafios atuais dos ISPs (*Internet Service Providers*) não estão mais relacionadas à velocidade de transmissão, mas sim em garantir qualidade de serviço (QoS - Quality of Service) para aplicações que exigem critérios a mais na transmissão de dados, como menor delay, jitter, perda de pacotes, entre outros. Este trabalho tem como objetivo implementar, através do software de simulação EVE-NG, uma rede baseada em uma topologia de backbone de um provedor da cidade de Castanhal, do Estado do Pará, a fim de comparar o desempenho do tráfego para dois tipos de serviços distintos (Streaming e Voip), com o uso de mecanismos de engenharia de tráfego proporcionados pelo protocolo MPLS, também chamado de MPLS TE. (Multi-Protocol Label Switching Traffic Engineering). Através das simulações, será avaliado o tráfego para esses dois serviços em uma cenário sem e com MPLS TE implementado. Os resultados obtidos mostram uma otimização do tráfego no cenário de rede com MPLS TE, através da determinação de caminhos explícitos na rede backbone para diferentes tipos de tráfego.

Palavras-chave: ISPs, QoS, MPLS, Engenharia de Tráfego.

ABSTRACT

Traffic rates on current internet providers have increased considerably due to the increasing presence of fiber optic links. With this scenario, the current challenges for ISPs (*Internet Service Providers*) are no longer related to transmission speed, but to guarantee quality of service (QoS - Quality of Service) for applications that require more criteria in transmission. data, such as lower delay, jitter, packet loss, among others. This work aims to implement, through the EVE-NG simulation software, a network based on a backbone topology of a provider in the city of Castanhal, in the State of Pará, in order to compare the traffic performance for two types of services (Streaming and VoIP), using traffic engineering mechanisms provided by the MPLS protocol, also called MPLS TE. (Multi-Protocol Label Switching Traffic Engineering). Through the simulations, the traffic for these two services will be evaluated in a scenario without and with MPLS TE implemented. The results obtained show an optimization of the traffic in the network scenario with MPLS TE, through the determination of explicit paths in the backbone network for different types of traffic.

Keywords: ISPs, QoS, MPLS, Traffic Engineering

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

AS	Autonomous System
ASIC	Application-Specific Integrated Circuits
BE	Best-Effort
CBR	Constraint Based Routing
CE	Customer Edge
CHR	Cloud Hosted Router
CR-LDP	Constraint-Based LDP
CR-LSP	Constraint-Based Routing LSP
CSPF	Constrained Shortest Path First
DHCP	Dynamic Host Configuration ProtocoL
DNS	Domain Name System
ENSP	Enterprise Network Simulation Platform
ER-LSP	Explicit route - LSP
FEC	Forwarding Equivalence Class
GNS3	Graphical Network Simulator 3
IETF	Internet Engineering Task Force
IGP	Interior Gateway protocol
LDP	Label Distribution Protocol
LER	Label Edge Router
LSR	Label Switching Router
MBPS	Megabit per second
NS2	Network Simulator version 2
PE	Provider Edge
RFC	Request for Comments
RSVP-TE	Resource Reservation Protocol - Traffic Engineering

SIP	Session Initiaion Protocol
SPF	Shortest Path First
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
WAN	Wide Area Network
RTT	Round-Trip Time
SNMP	Simple Network Management Protocol
EVE-NG	Emulated Virtual Environment Next Generation
IP	Internet Protocol
IPv6	Internet Protocol version 6
ISP	Internet Service Provider
LSA	Link-State Advertisement
LSP	Label Switch Path
MPLS TE	Multi-Protocol Label Switching Traffic Engineering
MPLS	Multiprotocol Label Switching
OSPF	Open Shortest Path First
QoS	Quality of Service
TE	Traffic Engineering
VOIP	Voice over Internet Protocol
VPN	Virtual private network

SUMÁRIO

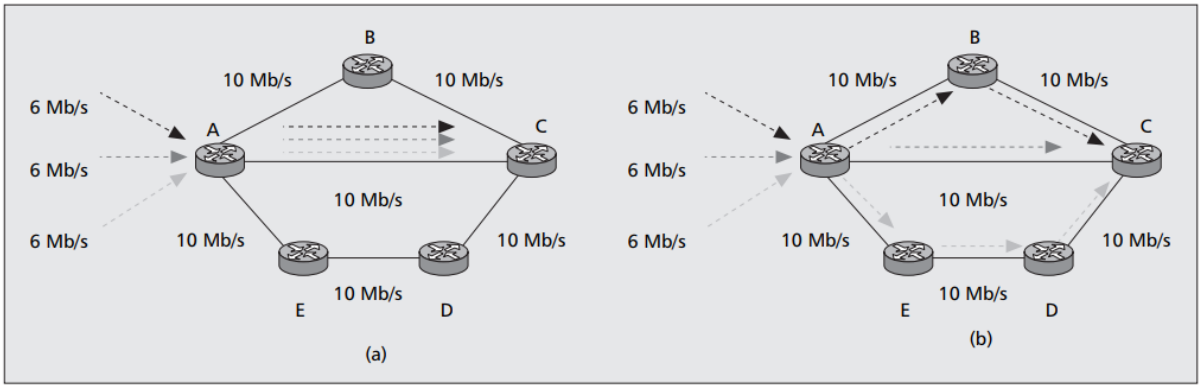
1 INTRODUÇÃO

A internet já foi um sistema de comunicação baseado somente nas políticas de melhor esforço (BE - *Best Effort*), que consiste em um sistema de serviço que não oferece nenhuma garantia de que a entrega de dados atenda a alguns requisitos de QoS *(Quality of Service)*. Atualmente, ela está se voltando a atender a demanda de tráfego exigidas por diferentes aplicações, como largura de banda, menor delay, jitter e perda de pacotes (KUROSE, 2005). Essas demandas podem ser atendidas através da implementação da Engenharia de tráfego em redes backbone.

A Engenharia de Tráfego (TE - *Traffic Engineering*) é um conceito importante para provedores de internet que buscam otimizar a performance da rede e a entrega de tráfego aos seus diversos clientes. As vantagens na aplicação da engenharia de tráfego está na possibilidade da otimização de roteamento, a fim de orientar o tráfego para atender às diversas demandas e fazer uso dos recursos da rede de maneira eficiente. Os autores Lee e Mukherjee (LEE; MUKHERJEE, 2004) definem que o objetivo do TE é "*...inserir o tráfego onde há largura de banda disponível*". O TE pode ser classificado por diversos critérios. Entre eles, podemos classificar de duas formas: *MPLS-based TE* (TE baseado em MPLS) e *IP-Based TE* (TE baseado em IP).

O *MPLS-Based TE* possibilita a implementação de roteamento explícito e divisão de fluxos de dados em diferentes caminhos de uma rede backbone para a otimização de tráfego e largura de banda. Isso é possível devido à criação de várias LSPs (*Label Switch Path*) que são caminhos preestabelecidos no roteador de entrada de uma rede MPLS. Na Figura 1, temos um exemplo de uma rede onde em um lado (a - sem TE) o fluxo de tráfego da rede segue um único caminho, entre os roteadores A e C, baseado no modelo de roteamento de menor distância. Dessa forma, haverá congestionamento na rede devido à subutilização de determinados enlaces. Já no lado oposto (b - com TE), há uma distribuição do tráfego por diferentes caminhos, utilizando de modo eficiente os recursos da rede.

Figura 1 – Topologia sem MPLS TE (a) e com MPLS TE (b)



Fonte: (WANG et al., 2008)

A desvantagem da engenharia de tráfego baseado em MPLS está no *overhead* (sobrecarga) que ela pode gerar com a criação de LSPs em uma rede backbone de grande porte. Além

disso, a criação de caminhos de *backup* (secundários) são obrigatórios no TE baseado em MPLS, visto que em caso de falha de link, o tráfego não poderá ser entregue automaticamente por caminhos alternativos.

O *IP-Based TE* consiste na otimização do valor de peso de protocolos de roteamento dinâmico (IGP - *Interior gateway protocol*), como o OSPF, de acordo com a topologia da rede e a demanda por largura de banda (FORTZ; THORUP, 2000). Ao contrário do TE baseado em MPLS, este não possibilita a implementação de roteamento explícito dedicado para fluxos individuais, pois as alterações do peso do link IGP podem afetar os padrões de roteamento de todo o conjunto de fluxos de tráfego.

A vantagem do *IP-Based TE* consiste na melhor escalabilidade, devido à ausência de *overhead* causada pela criação de LSPs, e resiliência, pois o fluxo de tráfego pode ser distribuído automaticamente por caminhos alternativos em caso de falha de link, sem a necessidade de especificar caminhos de backup. Contudo, sua desvantagem consiste no fato de que essas mudanças automáticas para caminhos alternativos podem gerar novos congestionamentos de tráfego, uma vez que o modelo de roteamento segue o padrão de menor custo ou menor distância.

1.1 Justificativa

O acesso a internet no Brasil está cada vez mais comum devido ao aumento na quantidade de ISPs (*Internet Service Provider*), principalmente na entrega de enlaces de fibra óptica. De acordo com o Cetic (Centro de Estudos sobre as Tecnologias da Informação e da Comunicação), a cada dez provedores de acesso à Internet no Brasil, nove oferecem fibra óptica aos clientes (Cetic.br, 2021). Com o aumento da oferta de internet de alta velocidade, é comum que determinados clientes busquem provedores que se destaquem e atendam as demandas para serviços como jogos online, streaming, entre outros. Com base nesse cenário, é comum que o foco dos ISPs atuais esteja voltado em melhorar a distribuição e priorização do tráfego para diferentes aplicações em sua rede backbone.

Este trabalho visa aplicar a engenharia de tráfego baseado em MPLS em uma topologia de rede simulada, que tem como base a topologia parcial de uma rede backbone de um provedor de Internet da cidade de Castanhal do estado do Pará. Atualmente, nesse provedor já existe o MPLS implementado para atender às aplicações de VPN (*Virtual private network*) porém o mesmo não é utilizado para a orientação do tráfego da rede.

Com o cenário simulado, podemos avaliar a importância da aplicação do MPLS para a engenharia de tráfego na rede e identificar em que situações elas podem ser aplicadas e os benefícios que ela pode trazer às redes de backbone de ISPs.

1.2 Objetivos

1.2.1 Objetivo Geral

O presente trabalho tem como objetivo avaliar os benefícios do uso de engenharia de tráfego baseado em MPLS em uma topologia que tem como base uma rede backbone parcial de um provedor da cidade de Castanhal do Estado do Pará. O software utilizado para a simulação será o EVE-NG (*Emulated Virtual Environment Next Generation*) e nela será reproduzido cenários de tráfego sem o MPLS TE e com o MPLS TE implementado.

No tráfego dos cenários será avaliado dois tipos de aplicações: serviços de *streaming de vídeo* e serviços de *voip*. O tráfego de duas aplicações será simulado pois cada uma exige diferentes requisitos de qualidade de serviços. Uma exige uma melhor largura de banda (*streaming de video*) e outra exige um menor delay (*voip*).

1.2.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Efetuar um estudo sobre o MPLS TE (*MPLS Traffic Engineering*)
- Descrever as etapas de configuração da topologia de rede no software EVE-NG
- Reproduzir tráfego com cenário sem MPLS TE e com MPLS TE
- Comparar os dois cenários através dos resultados gerados pela simulação

1.3 Estrutura do Trabalho

A organização desse trabalho está dividida em capítulos. No segundo capítulo é dada uma introdução sobre o protocolo OSPF, MPLS e MPLS TE, necessárias para o entendimento da topologia simulada. No terceiro capítulo serão apresentados projetos similares ao presente trabalho para fins de comparação. No quarto capítulo será abordado as tecnologias e softwares utilizados, como também as configurações realizadas para a criação da topologia. No quinto capítulo será realizada uma comparação dos resultados da simulação dos dois cenários. E por fim, temos o último capítulo com as considerações finais.

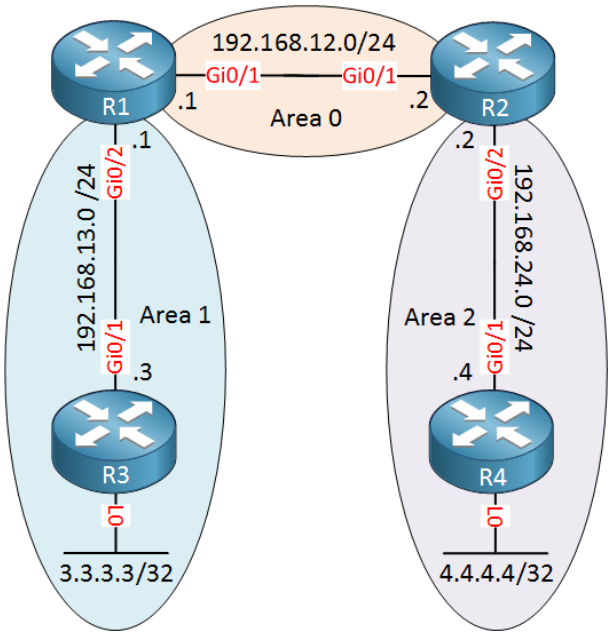
2 REFERENCIAIS TEÓRICOS

Este capítulo aborda os conceitos fundamentais para o desenvolvimento e compreensão deste trabalho, sendo eles os protocolos de rede OSPF (*Open Shortest Path First*), MPLS (*Multi-protocol Label Switching*) e MPLS TE (*Multiprotocol Label Switching Traffic Engineering*).

2.1 OSPF (*Open Shortest Path First*)

O OSPF (*Open Shortest Path First*) é um protocolo de roteamento dinâmico desenvolvido pelo IETF (*Internet Engineering Task Force*) e é definido na RFC 2328 (MOY et al., 1998). Esse protocolo é classificado como IGP (*Interior Gateway Protocol*), o que implica que este é utilizado para a distribuição de rotas entre roteadores que pertencem a um mesmo *Autonomous System (AS)*. O protocolo é baseado no algoritmo SPF (*Shortest Path First*) e possibilita que as suas rotas sejam calculadas baseadas na quantidades de nós (roteadores) até o destino e no estado do link de cada nó da rede durante o caminho. Dessa forma, parâmetros como largura da banda serão considerados na escolha do melhor caminho.

Figura 2 – Topologia OSPF divida em áreas



Fonte: (Network Lessons, 2022a)

De acordo com a RFC 2328, esse protocolo é capaz de detectar rapidamente mudanças na topologia da rede dentro do AS, possibilitando que novas rotas sejam recalculadas. O seu funcionamento se baseia no fato de cada roteador manter uma banco de dados com os estados de todos os roteadores que participam do OSPF. Estes estados se referem às interfaces de rede ativas e seus vizinhos próximos, que são construídas através do envio e recebimento de mensagens do LSA (*Link State Advertisement*). Com isso, cada roteador consegue construir uma árvore de

melhores caminhos considerando ele como a raiz dessa árvore. Esses melhores caminhos são construídos através do algoritmo SPF.

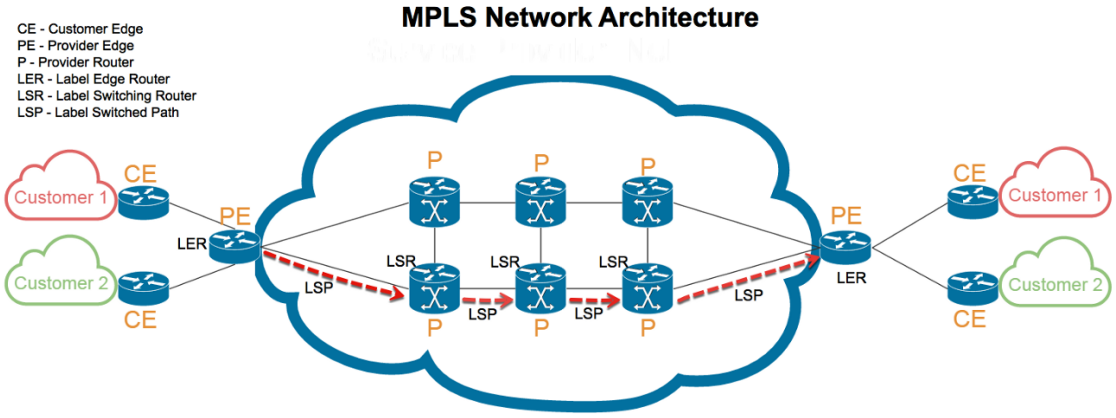
Outra característica do OSPF é a possibilidade de dividir uma rede em grupos diferentes, chamados áreas. Em cada área uma instância OSPF estará ativa que será totalmente independente da outra. Dessa forma, é possível reduzir o tempo de convergência para aprendizagem ou mudanças de rotas e o tráfego ocasionado pela troca de mensagens LSAs. Na Figura 2 apresentada acima, temos uma topologia OSPF de exemplo divida em três áreas diferentes.

2.2 MPLS (*Multiprotocol Label Switching*)

O MPLS (*Multiprotocol Label Switching*) é um protocolo de redes definido pela RFC 3031 (ROSEN et al., 2001). Este protocolo possibilita a criação e distribuição de rótulos (*labels*) para a comutação de pacotes. Esses rótulos seriam adicionados ao topo do cabeçalho IP permitindo que roteadores efetuem o encaminhamento do pacote com base nos rótulos e não nos endereços IPs de destino. Os pacotes seguem o caminho pela nós da rede através da adição e/ou troca dos rótulos que irão indicar para qual interface o pacote deve ser enviado.

Os roteadores que compõem uma rede MPLS são chamados de LSRs (*Label Switching Routers*). Um LSR é um roteador de núcleo da rede MPLS, que participa do estabelecimento de LSPs (*Label Switching Paths*) através de protocolos de sinalização de rótulos. Uma LSP é o caminho percorrido por pacotes MPLS entre dois LSRs quaisquer. Outra definição utilizada para roteadores de uma rede MPLS são os chamados LER (*Label Edge Router*), que são os roteadores de borda do domínio MPLS. Na Figura 3, temos um exemplo de uma rede MPLS com os seus componentes.

Figura 3 – Arquitetura de uma rede MPLS



Fonte: (Steve Jacob, 2018)

Os objetivos iniciais para a criação do MPLS estavam relacionados em possibilitar uma maior velocidade no encaminhamento de pacotes (Huawei Technologies, 2022). O processo tradicional de encaminhamento era considerado mais lento se comparado ao uso de rótulos,

visto que o roteador precisaria desencapsular o pacote para verificar o endereço IP de destino e encontrar a melhor rota para aquele destino na tabela de roteamento. Em contrapartida, através do MPLS, o roteador possui um mapeamento de rótulos e interfaces que é criado previamente, permitindo assim efetuar o encaminhamento apenas verificando o rótulo que é adicionado ao topo do pacote.

Com o avanço do hardware dedicados a tarefas de processamento de pacotes em roteadores para encaminhamento (data plane), velocidade não é mais uma razão para a adoção do MPLS. Confome documentado no site da Huawei (Huawei Technologies, 2022), através do uso de ASIC (Application-Specific Integrated Circuits), o encaminhamento de pacotes tradicional não é mais o gargalo no desenvolvimento da rede. Portanto, o uso do MPLS nos dias atuais está voltado aos diversos benefícios que ela traz para aplicações de VPN (Virtual Private Networks), QoS e engenharia de tráfego.

2.2.1 FEC (Forwarding Equivalence Class)

Uma FEC consiste em um conjunto de parâmetros que irão determinar um caminho para os pacotes IPs. Os pacotes associados a uma mesma FEC serão encaminhados pelo mesmo caminho. A FEC pode ser determinada por um ou mais parâmetros. Alguns desses parâmetros são:

- Endereço IP de origem e/ou destino
- Interface de origem e/ou destino
- Protocolo IP
- QoS

Em uma rede MPLS, o pacote é mapeado para a FEC somente no roteador de entrada (LER). Em nenhum momento, através do LSP, um roteador irá determinar novamente a FEC a que o pacote pertence, até que o mesmo chegue ao host final.

2.2.2 LDP (Label Distribution Protocol)

Conforme já explicado, o MPLS efetua o encaminhamento de pacotes com base nos rótulos. No momento em que um pacote entra numa rede MPLS, este é associado a uma classe de equivalência (FEC) e assim é criado um LSP relacionado a essa FEC. Após determinado ao qual LSP um pacote pertence, o roteador de entrada da rede MPLS (*LER - Label Edge Router*) irá associar um rótulo àquele pacote. Os roteadores de núcleo (*LSR - Label Switch Router*) só terão o trabalho de efetuar a troca dos rótulos, encaminhando assim o pacote de acordo com o LSP determinado anteriormente. Quando um LER está na saída da rede MPLS, ele é responsável pela remoção do rótulo do pacote IP antes de realizar a entrega ao nó de destino.

Para que seja possível a criação de LSPs e permitir que todos os roteadores tenham conhecimento a respeito destas e dos rótulos à elas associados, os roteadores precisam trocar informações de rótulos entre eles. Isso é possível através de protocolos de sinalização. O protocolo LDP, desenvolvido pela IEFT através da RFC 3036, é um protocolo projetado especificamente para a distribuição de rótulos e criação de LSPs.

Assim como o OSPF, o protocolo LDP é responsável pelo envio periódico de mensagens entre os roteadores para descoberta de vizinhos e estabelecimento de comunicação que é feita via TCP e UDP. Essa comunicação ocorre somente com os vizinhos diretamente conectados, deste modo, um LSR não precisa ter em suas tabelas todas as informações sobre todos os nós da rede. As sessões LDP são criadas e mantidas através de um grupo de mensagens que são enviados periodicamente, a saber:

- *Discovery messages*: Utilizadas para anunciar e manter a presença de um LSR em uma rede MPLS.
- *Session Messages*: utilizadas para iniciar, manter e terminar sessões LDP entre pares de roteadores.
- *Advertisement messages*: Cria, modifica e remove rótulos mapeados para determinadas FECs.
- *Notification messages*: Fornece informações sobre erros.

O LDP usa o protocolo TCP para transmitir *Session Messages*, *Advertisement messages* e *Notification messages* garantindo assim uma transmissão confiável dessas mensagens. O LDP usa o protocolo UDP apenas para transmitir *Notification messages* (Huawei Technologies Co, 2018).

2.3 MPLS TE (Traffic Engineering)

Existem duas maneiras de estabelecer um LSP numa rede MPLS. A primeira é a do tipo *Control Driven*, que é estabelecido entre todos os nós utilizando protocolos de sinalização como o LDP conforme abordado no tópico anterior. A segunda maneira é estabelecendo ER-LSP (*Explicit route - LSP*) ou CR-LSP (*Constraint-based Routing LSP*), utilizada para aplicações de engenharia de tráfego.

Quando um LSP do tipo *Control Driven* é estabelecido, cada LSR determina a próxima interface para o LSP baseado na tabela de encaminhamento IP (*Forwarding Table*) e envia uma requisição de rótulo ao roteador do próximo salto a fim de que este saiba qual rótulo associar ao LSP em questão. Portanto, um LSP do tipo *Control Driven* segue o caminho que um pacote usando o encaminhamento IP tradicional seguiria.

Esses tipos de LSPs irão sofrer do mesmo problema que existe em uma rede IP tradicional, no que se refere ao uso de largura de banda ineficiente, devido à subutilização de caminhos secundários em favorecimento de melhores caminhos - calculados pelos protocolos de roteamento IGP - para o encaminhamento do tráfego. Para evitar o uso ineficiente de largura de banda, que é a causa de congestionamento e perda de pacotes na rede, pode-se aplicar o MPLS TE (*MPLS Traffic Engineering*).

2.3.1 CR-LSP (*Constraint-based Routing LSP*)

O CR-LSP (*Constraint Based Routed LSP*) define caminhos baseados em *constraints* (restrições), que são parâmetros adicionais que serão considerados para o estabelecimento de caminhos na rede MPLS, como largura de banda e políticas administrativas. Dessa forma, ele pode selecionar rotas que sejam mais longas em relação às rotas mais curtas - preferidas pelo OSPF - se estas atenderem as restrições exigidas para esta LSP.

Há dois protocolos que podem ser utilizados para o estabelecimento de CR-LSPs: CR-LDP e RSVP-TE. O CR-LDP (*Constraint-based LDP*) contém extensões para o protocolo LDP que possibilita utilizar o LDP para estabelecer LSPs baseados em restrições. No entanto, esse protocolo não é muito utilizado visto que o RSVP-TE prevaleceu como resultado da análise comparativa realizada no âmbito do IETF, e tornada pública através da RFC 3468. Além disso, o RSVP-TE é adotado pelos principais fabricantes líderes do mercado como Huawei, Cisco e Mikrotik (RIZZETTI et al., 2014).

2.3.2 RSVP-TE (*Resource Reservation Protocol -TE)*

O RSVP-TE *(Resource Reservation Protocol -TE)* é uma extensão do protocolo RSVP versão 1 definido na RFC 2205, e se trata de um protocolo de sinalização com o propósito de possibilitar o estabelecimento de LSPs e túneis unidirecionais em redes MPLS para prover reserva de recursos como largura de banda, espaço no buffer dos equipamentos e entre outros, ao longo de todo LSP. É importante deixar claro que as sessões estabelecidas pelo RSVP-TE ou qualquer outro protocolo de sinalização são unidirecionais, portanto, caso haja necessidade de reservar recursos em ambas as direções é necessário a criação de duas sessões.

2.3.2.1 Tipos de Mensagens

Para estabelecer uma sessão, uma troca de mensagens é efetuada entre os roteadores LSR, explicitando os recursos solicitados. Segue abaixo as principais mensagens geradas pelo RSVP-TE (Cisco Press, 2006).

- *Path Message*: Gerado pelo roteador inicial da rede MPLS TE e encaminhado pela rede ao longo do caminho de um futuro *TE LSP*. A cada salto, a mensagem *PATH* verifica a disponibilidade dos recursos solicitados e armazena essas informações.

- *Reservation Message*: Criado pelo roteador final na rede MPLS TE e usado para confirmar a solicitação de reserva que foi enviada anteriormente com as mensagens *PATH*.
- emphError Messages: Em caso de indisponibilidade dos recursos solicitados, o roteador gera mensagens de erro RSVP e as envia ao roteador de onde a solicitação ou resposta foi recebida. Existem dois tipos de mensagens de erro: *PathErr* e *ResvErr*.
- *Tear Messages*: RSVP cria dois tipos de mensagens *tear*, a saber, a mensagem *Path tear* e a mensagem *Reservation tear*. Essas mensagens limpam os estados *Path* ou *Reservation* no roteador instantaneamente, possibilitando a reutilização de recursos no roteador para outras solicitações.

Os túneis RSVP-TE podem ser estabelecidos através de caminhos explícitos estáticos ou dinâmicos. No caso de caminhos estáticos, cada roteador ao longo do túnel irá definir o *next-hop* baseado na rota explícita que foi especificada no *Path Message*. Essa rota explícita pode ser completa (especifica todos os nós ao longo do caminho) ou parcial (específica apenas alguns nós que devem ser atravessados). Se nenhuma rota é encontrada ou as interfaces não atenderem as constraints solicitadas, o túnel não pode ser estabelecido.

Para estabelecer um túnel dinamicamente, o protocolo CSPF (*Constrained Shortest Path First*) é utilizado. Este é uma extensão dos protocolos de roteamento IGP para engenharia de tráfego. Com o CSPF, o roteador de entrada da rede MPLS calcula o caminho que satisfaz os requisitos e produz um caminho explícito para o *Path Message*. Se o caminho correspondente às restrições não puder ser calculado, o túnel não poderá ser estabelecido.

3 TRABALHOS CORRELATOS

3.1 Engenharia de Tráfego com Constraint Based Routing em redes MPLS

Os autores Hodzic e Zoric (HODZIC; ZORIC, 2008) abordam sobre a importância do MPLS TE para evitar congestionamento de banda, devido à subutilização de links em uma rede backbone. Os autores fornecem uma leve introdução a respeito dos protocolos de sinalização utilizados para a implementação do CBR (*Constraint Based Routing*), a saber: CR-LDP e RSVP-TE, como também explicam as diferenças, vantagens e desvantagens de cada protocolo.

Os autores utilizaram o simulador NS2 (*Network Simulator version 2*) para criar uma topologia com MPLS-TE e sem MPLS-TE. No primeiro caso, houve um congestionamento devido ao tráfego ser direcionado somente em uma rota (rota ótima de acordo com protocolos de roteamento internos). E no segundo caso, com aplicação de engenharia de tráfego através do uso de *Explicit Routes* (Rotas Explícitas), a rota que estava ociosa foi adicionada para um dos fluxos de tráfego, resolvendo assim, o problema de congestionamento.

O trabalho citado se assemelha bastante com o presente trabalho, uma vez que faz uso de dois cenários, com MPLS-TE e sem MPLS-TE, e aborda os problemas de congestionamento que ocorre com rotas ociosas, devido à escolha de rotas ótimas através de protocolos de roteamento dinâmico. A diferença deste em relação ao presente trabalho está no software de simulação utilizado, no detalhamento das configurações aplicadas e no estudo de caso, visto que este trabalho tem como foco o tráfego para duas aplicações distintas.

3.2 Análise e Otimização de Roteamento em Backbones OSPF Utilizando MPLS-TE

Em sua dissertação, o autor Oliveira (OLIVEIRA, 2011) propõe aplicar o MPLS TE a uma rede backbone IP/MPLS de uma operadora de telecomunicações a fim de otimizar o tráfego da rede, direcionando certos fluxos de dados por meio de caminhos secundários não utilizados pelo OSPF. Para realizar o testes, o autor não fez uso de softwares de simulação mas implementou fisicamente um fragmento de uma topologia backbone com equipamentos físicos pertencentes a um ponto de presença de uma operadora de telecomunicações. Os equipamentos utilizados foram da fabricante Cisco dos modelos 7206, 2611 e 3725 para a simulação dos CEs (*Customer Edges*), do modelo 7204 para a simulação do PEs (*Provider Edges*) e do modelo 7206 para a simulação do Ps (*Provider*).

Inicialmente, o autor implementou a rede IP/MPLS para análise do tráfego baseado no roteamento OSPF. Através do software de gerenciamento *PRTG* e do analisador de protocolo

Acterna, o autor pôde coletar informações de tráfego e *jitter* e constatar a má distribuição do tráfego e a superutilização de apenas uma rota entre o roteador de borda e os roteadores de núcleo do backbone. Após isso, foi aplicada o MPLS-TE na rede backbone e constatado uma melhoria no congestionamento da rede através do balanceamento do tráfego pelas rotas subutilizadas.

Além disso, o autor implementou em sua topologia um servidor *voip* completo para análise de performance mediante os testes de tráfego. O mesmo constatou uma redução dos níveis de *jitter* na aplicação do MPLS TE, nas redes com aplicações *voip* controladas pelos protocolos SIP e H.255.

O trabalho do autor se assemelha bastante com o presente projeto referente aos objetivos de análise do MPLS-TE para melhoria do tráfego. A diferença está na abordagem prática visto que o autor fez uso de equipamentos físicos e nas ferramentas para análise de diversos parâmetros de tráfego.

3.3 Implantação de Engenharia de tráfego com MPLS-TE em rede WAN

Em sua monografia, a autora Ana (SCHARF, 2017) propõe aplicar o MPLS TE como uma forma de atender quatro demandas típicas de redes WAN, a saber:

- Facilitar a implantação de enlaces para clientes.
- Incrementar a robustez a falhas, para maximizar a conectividade dos clientes.
- Aumento do aproveitamento das capacidades dos enlaces da rede.
- Tratamento diferenciado para tráfegos em função de seus tipos ou dos clientes que os geraram.

A topologia que a autora usou como modelo para o seu projeto foi baseada em uma parte da rede de telecomunicações da empresa Eletrosul. Para a simulação, foi utilizado o software GNS3 com roteadores Cisco da série 7200, em conjunto com o VirtualBox para a simulação de terminais, que representam os clientes.

A autora elaborou três experimentos com dois tipos de tráfegos gerados em cada um deles. Utilizou-se o software *iperf* para a transmissão de pacotes IPv4 a fim de gerar tráfego de melhor esforço durante um intervalo de 5 minutos. O outro tipo de tráfego foi gerado através da ferramenta PJSUA, que serve para a construção de aplicações multimídia *Session Initiaion Protocol* (SIP) de forma simples.

Os resultados obtidos pela autora podem ser resumidos na figura abaixo. Além da taxa de transferência ser diferente para cada experimento, houve uma mudança de cenários para cada um

deles, sendo que no primeiro foi simulado a comunicação entre um par de clientes, no segundo foi adicionado mais uma chamada SIP em relação ao primeiro cenário e no terceiro experimento houve comunicação com dois pares de clientes.

Figura 4 – Resultados obtidos pela autora Ana Luiza Scharf

Taxa do tráfego do melhor esforço	Resultados esperados	Resultados obtidos
1 Mbps	–Sem perdas de pacotes da chamada SIP. –Sem oscilações no tráfego de melhor esforço.	–Sem perdas de pacote para o tráfego da chamada SIP. –Variações mínimas no tráfego de melhor esforço.
2 Mbps	–Perdas de pacotes do SIP –Oscilações no tráfego de melhor esforço.	–Sem perdas de pacote para o tráfego SIP, exceto o Cenário3. –Variações significativas no tráfego de melhor esforço.
3 Mbps	–Perdas de pacotes do SIP. –Várias oscilações no tráfego de melhor esforço	–Perdas de pacote para o tráfego SIP. –Várias oscilações no tráfego de melhor esforço

Fonte: (SCHARF, 2017)

O trabalho da autora se assemelha ao presente trabalho em relação à aplicação e simulações do MPLS TE em uma topologia de rede. No entanto, ela difere não somente nos softwares utilizados, como também no tipo de abordagem adotado visto que a autora não efetuou comparações com cenários sem o MPLS TE.

4 METODOLOGIA

O presente trabalho foi baseado em uma topologia de rede de backbone de um provedor de internet na cidade de Castanhal, no estado do Pará. Para a simulação desta rede foi empregada a plataforma EVE-NG, que permite a emulação de switches e roteadores de diferentes fabricantes, na qual foi adotado o roteador do fabricante Mikrotik. Isso possibilitou simular e analisar o comportamento da rede com e sem o MPLS-TE.

Além do EVE-NG, foi utilizado também o software de monitoramento Zabbix que permite a captura de diversos dados via protocolo SNMP possibilitando a geração de gráficos do tráfego da rede para análise dos resultados.

4.1 Ferramentas

4.1.1 EVE-NG

O EVE-NG (*Emulator Virtual Environment - New Generation*) é um emulador de rede que possibilita a reprodução de cenários de rede com dispositivos *multivendor*. A plataforma permite a emulação de equipamentos de rede, ao contrário da simulação, que é o padrão adotado por softwares como Packet Tracer, HP Network Simulator e ENSP, que apenas simulam o comportamento de roteadores e switches de seus respectivos fabricantes (OLIVEIRA, 2020).

A ferramenta fornece uma interface web para acesso e criação de laboratórios. Os usuários podem adicionar, conectar e configurar nós de rede a partir de uma biblioteca de modelos. É possível executar softwares de dispositivos comerciais no *Dynamips* e *IOU* (como os da Cisco, por exemplo) e outros dispositivos de rede, como roteadores de código aberto, no QEMU. Por ser uma máquina virtual, o EVE-NG pode ser instalado e configurado em qualquer sistema operacional, como Windows, Linux ou Mac OS.

O destaque dessa plataforma em relação às outras ferramentas de simulação de redes está precisamente no fato de possibilitar a emulação de dispositivos de diferentes fabricantes, uma vez que a heterogeneidade de dispositivos em redes backbones de provedores de internet é bastante comum. No entanto, o EVE-NG não é a primeira ferramenta a possibilitar a emulação de dispositivos *multivendor*. O GNS3 (*Graphical Network Simulator*), que é um simulador de redes gráfico, também oferece suporte para a emulação de dispositivos de diferentes fabricantes (DAYANAND; GHORBANI; VAGHRI, 2016). No entanto, a sua instalação e uso é mais complexa se comparado ao EVE-NG, que além de uma instalação mais simples, fornece uma interface de gerenciamento web intuitiva e que pode ser acessada via IPv4 e IPv6 de qualquer dispositivo que esteja na mesma rede, podendo também ser instalada na nuvem, como o Google Cloud.

Para este projeto, o EVE-NG foi instalado em um máquina com sistema operacional Win-

dows versão 10, com 6GB de RAM dedicado para a máquina virtual. O sistema de virtualização utilizado foi o *VMware Workstation 16 Player* (versão não comercial).

4.1.2 Zabbix

O Zabbix é um software de monitoramento open-source, criado por Alexei Vladishev e suportado nativamente pelo Zabbix SIA. Este software consegue monitorar vários parâmetros de uma rede e a saúde e integridade dos servidores. Isso permite uma reação rápida aos problemas de qualquer dispositivo monitorado. Além disso, o Zabbix oferece excelentes recursos de alertas, relatórios e visualização de dados com base nos dados armazenados (SHOKHIN, 2015).

No cenário deste projeto será necessário avaliar parâmetros como o tráfego de uma interface e o tempo de resposta (*RTT - Round Trip Time*) para determinada aplicação e, através do Zabbix, é possível capturar essas informações em tempo real. Neste projeto, o Zabbix está instalado em outro servidor e irá efetuar as capturas de informações dos nós da topologia através de conectividade IPv6. Dessa forma, é possível ter o EVE-NG sendo executado em uma máquina e o Zabbix em outra, capturando as informações dos hosts do ambiente simulado através da Internet.

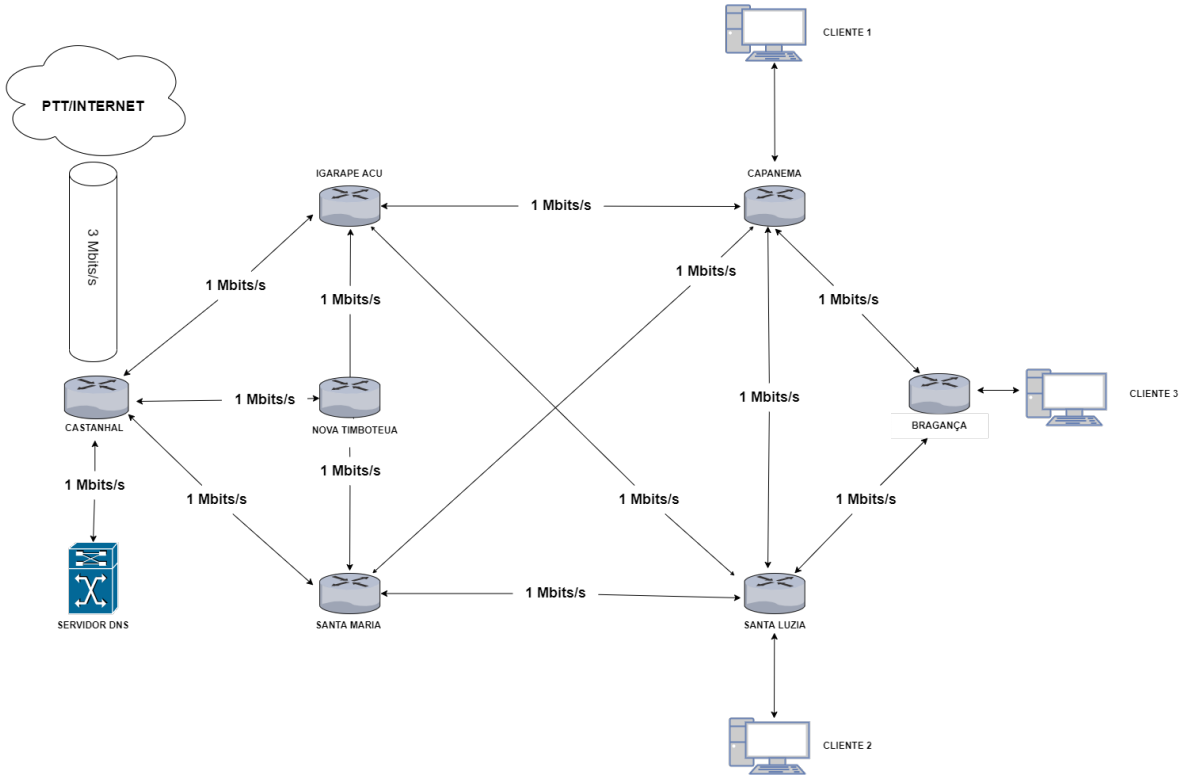
A versão do software utilizada neste trabalho foi a 6.0.2, instalada em uma máquina virtual com sistema operacioal Ubuntu, versão 20.04 LTS.

4.2 Topologia

O cenário representado pela Figura 5 é baseado em uma topologia de backbone de um provedor regional da cidade de Castanhal/PA, como já mencionado. Cada roteador está nomeado com a localidade em que cada um opera. O modelo escolhido para os roteadores nessa topologia foi o *Cloud Hosted Router* (CHR), que é uma versão do RouterOS da Mikrotik destinada a ser executada como uma máquina virtual. Ele suporta a arquitetura x86 de 64 bits e pode ser usado na maioria dos hipervisores populares, como VMWare, Hyper-V, VirtualBox, KVM e outros (Mikrotik Wiki, 2020a).

Os hosts indicados pelos nomes *Cliente 1*, *Cliente 2* e *Cliente 3* também são equipamentos com o RouterOS CHR e estão presentes nessa topologia para representar o tráfego originado por clientes das regiões de Capanema, Santa Luzia e Bragança, respectivamente. Os enlaces possuem velocidade de 1 Mbps devido às limitações da licença grátis do CHR.

Figura 5 – Topologia de Backbone



Fonte: O Autor (2022)

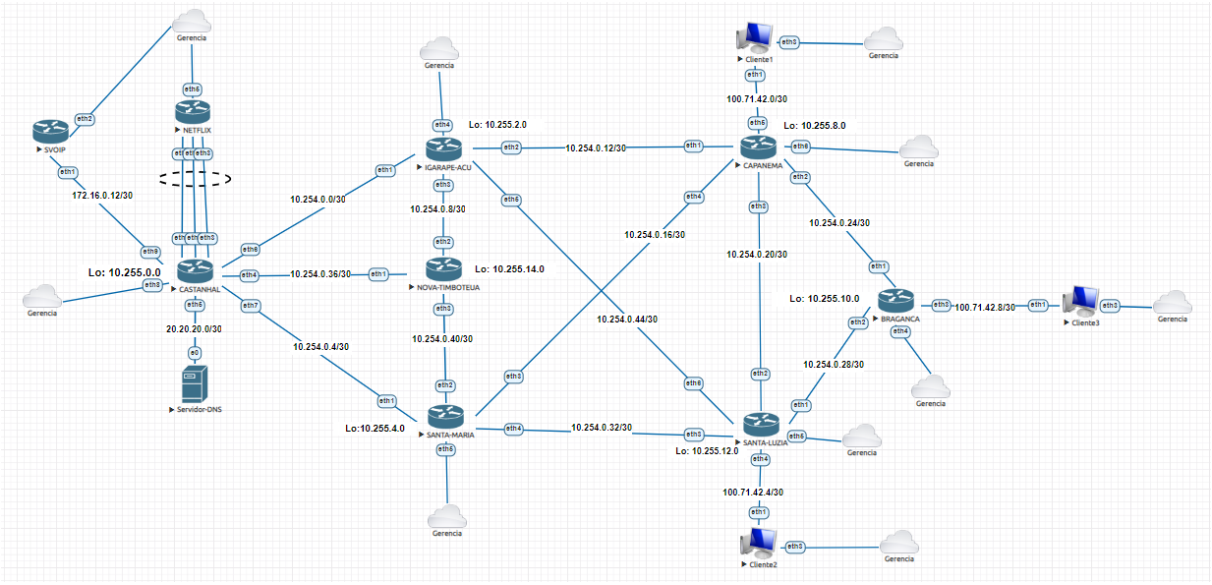
4.2.1 Implementação

Na Figura 6 temos a implementação, no EVE-NG, da topologia apresentada anteriormente. Inicialmente, observa-se que alguns hosts possuem conexão com a figura de uma nuvem. Essa é uma ferramenta disponibilizada pelo EVE-NG que permite a conexão destas máquinas emuladas com os adaptadores *Ethernet* virtuais que são configuradas na máquina do usuário no momento da instalação do Vmware. Dessa forma, é possível criar uma *Bridge* (ponte) que permite a comunicação da máquina do usuário com as máquinas simuladas no EVE-NG.

Além disso, se a interface de rede do EVE-NG estiver configurada em *Bridge*, como exemplificado na Figura 7, as máquinas virtuais podem receber um endereço IP da mesma faixa da rede interna do usuário via DHCP (*Dynamic Host Configuration Protocol*), possibilitando a comunicação com qualquer máquina dentro daquela rede, podendo, ainda, fornecer acesso à internet para as máquinas virtualizadas. Esse recurso é de vital importância pois dessa forma é possível estabelecer a comunicação dessas máquinas virtualizadas com o servidor Zabbix que está instalado em outra máquina.

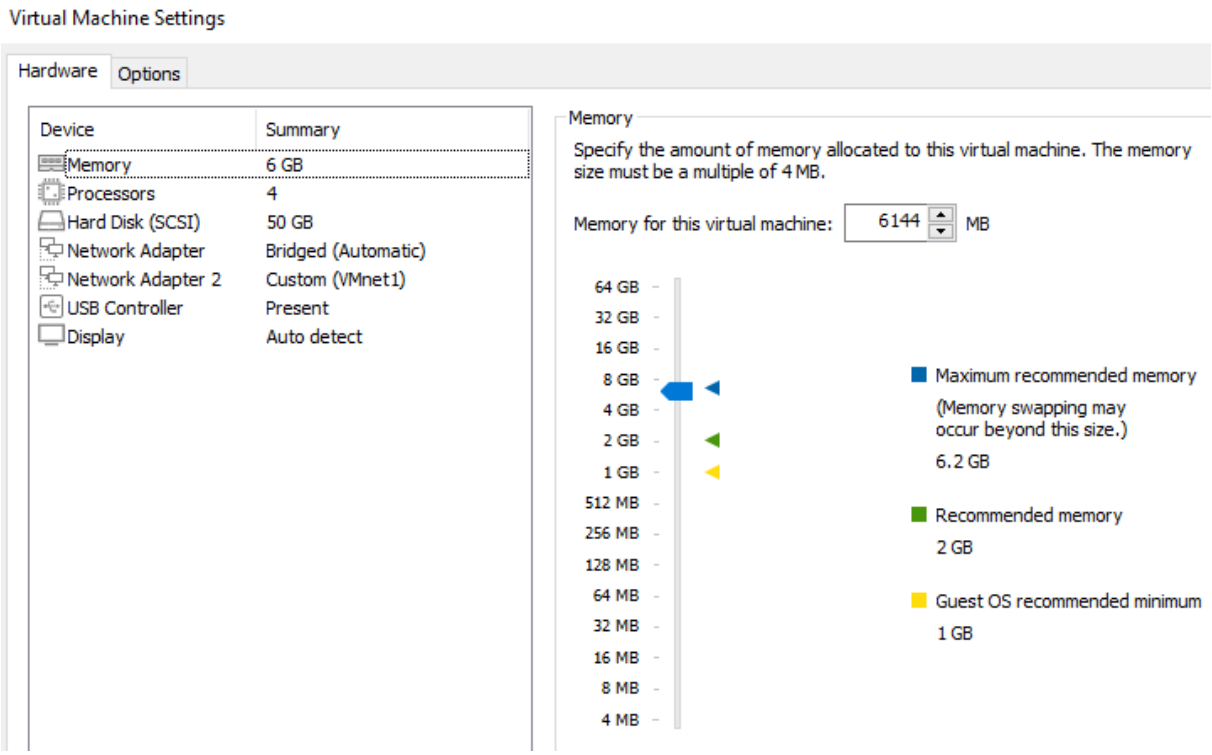
Os hosts nomeados como *Netflix* e *Svoip* da Figura 6 estão simbolizando os serviços acessados pelos clientes, que foi simplificado em uma nuvem na topologia da Figura 5. Como o tráfego para o host *Netflix* requer uma largura de banda maior e será utilizado pelos três clientes simultaneamente, foi necessário fornecer um link de maior capacidade, uma vez que há um

Figura 6 – Topologia implementada no EVE-NG



Fonte: O Autor (2022)

Figura 7 – Configurações da máquina virtual



Fonte: O Autor (2022)

limite de de velocidade de 1 Mbps por link. Para isso, foi adicionado três links entre os hosts de *Castanhal* e *Netflix* para evitar problemas de congestionamento por falta de banda durante a simulação de tráfego.

Vale ressaltar que apenas incluindo os três links para o host *Netflix* não implica que os três serão utilizados igualmente. Para evitar que um dos links ficasse sobrecarregado e outros ocioso,

foi considerado a possibilidade de agregar os três links em um *Bonding*. Nos roteadores Mikrotik, o *Bonding* é uma tecnologia que permite a agregação de múltiplas interfaces *Ethernet* em uma interface virtual, possibilitando alta taxas de transferências de dados e o recurso de *Failover* que, em caso de queda de uma das interfaces, possibilita que o tráfego continue sendo transmitido pela interface ainda ativa (Mikrotik Wiki, 2020b). No entanto, possivelmente devido às limitações da virtualização, houve congestionamento com testes de tráfego utilizando diferentes algoritmos de balanceamento do *Bonding*.

Para solucionar esse problema, foi dedicado exclusivamente um link para cada cliente entre o host de *Castanhal* e *Netflix*. Para que o tráfego de cada cliente fosse encaminhado obrigatoriamente em uma interface específica, foi necessário criar regras de roteamento com a ferramenta *Mangle* do Mikrotik, que permite, além de outras coisas, efetuar marcações nos pacotes de entrada e saída. Dessa forma, se um pacote que chega ao roteador *Castanhal* com destino ao endereço IP da *Netflix* e que tem como origem o endereço IP de um dos clientes, recebe uma marcação específica.

Com essas marcações é possível criar uma rota estática para cada cliente direcionando o tráfego através do *Gateway* vinculado à interface que foi dedicada ao cliente. As configurações dessas rotas estáticas estão detalhadas na Figura 8.

Figura 8 – Configuração de firewall e rotas estáticas

```
[admin@CASTANHAL] > ip firewall mangle print where dst-address="23.246.50.1"
Flags: X - disabled, I - invalid, D - dynamic
 0    ;;; Marca Rota - Cliente 1
      chain=prerouting action=mark-routing new-routing-mark=cl1
      passthrough=no src-address=100.71.42.2 dst-address=23.246.50.1

 1    ;;; Marca Rota - Cliente 2
      chain=prerouting action=mark-routing new-routing-mark=cl2
      passthrough=no src-address=100.71.42.6 dst-address=23.246.50.1

 2    ;;; Marca Rota - Cliente 3
      chain=prerouting action=mark-routing new-routing-mark=cl3
      passthrough=no src-address=100.71.42.10 dst-address=23.246.50.1
[admin@CASTANHAL] > ip route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 0 A S  ;;; ROTA NETFLIX - CL1
        23.246.50.1/32                     172.16.0.1                1
 1 A S  ;;; ROTA NETFLIX - CL2
        23.246.50.1/32                     172.16.0.5                1
 2 A S  ;;; ROTA NETFLIX - CL3
        23.246.50.1/32                     172.16.0.9                1
```

Fonte: O Autor (2022)

O servidor DNS apresentado na topologia da Figura 6 é uma máquina linux, configurada para responder às requisições DNS de todos os hosts dessa rede. Ela foi inserida na topologia para que cada host pudesse ter um nome de domínio. Dessa forma, durante testes de *Traceroute* seria mais fácil visualizar o caminho que o pacote está percorrendo, visto que há muitos IPs de

ponto a ponto. Os domínios criados seguem um padrão. Por exemplo, o IP "10.255.10.0"do host *Bragança* é representado pelo domínio "braganca-0-10-255-10".

Na máquina linux, foi instalado e configurado o software *DNSmasq*, para que o servidor fosse capaz de responder às requisições DNS. Além disso, os domínios criados foram adicionados no diretório /etc/hosts. Cada roteador na rede possui configurado o IP 20.20.20.2 como servidor de DNS.

Ainda na Figura 6, os roteadores representados pelo nome *Castanhal*, *Igarapé-Açu*, *Nova Timboteua*, *Santa Maria*, *Santa Luzia*, *Capanema* e *Bragança* fazem parte da rede de backbone, onde o roteamento é realizado. Nessa rede foi implementado inicialmente os protocolos OSPF e MPLS, a fim de verificar os caminhos empregados para o transporte de dados dos clientes às aplicações de destino e analisar o desempenho no acesso simultâneo dos três clientes para cada aplicação.

4.3 Cenário Inicial

Neste tópico será apresentado as configurações dos protocolos OSPF e MPLS apenas dos roteadores *Castanhal*, *Capanema*, *Santa Luzia* e *Bragança*, uma vez que neles há configurações adicionais de roteamento pois são os roteadores de borda que conectam diretamente aos clientes. Nos outros roteadores da topologia as configurações são semelhantes.

4.3.1 OSPF

Para habilitar o OSPF no roteador é necessário criar uma instância onde é definido os parâmetros para o estabelecimento da vizinhança em uma área OSPF. A área adotada em toda a rede backbone é a área 0 (*backbone*), que é a área padrão.

Na Figura 9, podemos observar a configuração de instância OSPF no roteador *Castanhal*. Assim como em todos os outros roteadores, o endereço IP escolhido para o *Router ID* é o endereço IP de loopback. O *Router ID* é o identificador único de um roteador em uma rede OSPF (Network Lessons, 2022b). As métricas, que são utilizadas pelos roteadores vizinhos para avaliar o custo das rotas por cada interface, permanecem no valor padrão (valor 20 no Mikrotik). O que foi adicionado nesta instância foi a redistribuição de rotas diretamente conectadas (*Redistribute-Connected*) para que os endereços IPs dos hosts *Neflix* e *Svoip*, que estão diretamente conectados ao roteador *Castanhal*, pudessem ser divulgados para os outros roteadores via OSPF.

Para que uma vizinhança OSPF seja estabelecida, é preciso definir as interfaces que participarão da sessão OSPF. No Mikrotik, podemos fazer isso de diversas formas. Uma delas é anunciando os endereços de rede dos enlaces entre os roteadores. Dessa forma, o roteador entende que a interface onde está configurado o IP de uma determinada rede irá participar da sessão OSPF e também que aquela rede deve ser anunciada para os seus vizinhos. No caso

Figura 9 – Instância OSPF

```
[admin@CASTANHAL] > routing ospf instance print
Flags: X - disabled, * - default
 0  * name="default" router-id=10.255.0.0 distribute-default=always-as-type-2
      redistribute-connected=as-type-2 redistribute-static=as-type-2
      redistribute-rip=no redistribute-bgp=no redistribute-other-ospf=no
      metric-default=1 metric-connected=20 metric-static=20 metric-rip=20
      metric-bgp=auto metric-other-ospf=auto mpls-te-area=backbone
      mpls-te-router-id=Loopback in-filter=ospf-in out-filter=ospf-out
```

Fonte: O Autor (2022)

do roteador *Castanhal*, temos os endereços IPs referente às conexões com os roteadores de *Igarapé-Açu*, *Nova Timboteua* e *Santa Maria*.

Figura 10 – Anúncio de rotas

```
[admin@CASTANHAL] > routing ospf network print
Flags: X - disabled, I - invalid
 #   NETWORK            AREA
 0   10.255.0.0/32      backbone
 1   10.254.0.0/30      backbone
 2   10.254.0.4/30      backbone
 3   10.254.0.36/30     backbone
```

Fonte: O Autor (2022)

Além disso, para que outras redes não sejam anunciadas via OSPF, uma vez que a redistribuição de rotas diretamente conectadas está ativada, foi criado um filtro de roteamento que pode ser observado na Figura 11. Deste modo, no roteador *Castanhal*, apenas as redes do servidor DNS (20.20.20.0/30) e os endereços IPs de loopback dos hosts *Netflix* (23.246.50.1/32) e *Svoip* (185.2.220.1/32) serão anunciadas. Por conseguinte, outras rotas diretamente conectadas, como, por exemplo, as utilizadas para a conexão ponto a ponto com os hosts *Svoip* e *Netflix* (172.16.0.X), não serão redistribuídas pois são descartadas através do filtro de roteamento.

Figura 11 – Filtro de roteamento

```
[admin@CASTANHAL] > routing filter print
Flags: X - disabled
 0   chain=ospf-out prefix=20.20.20.0/30 invert-match=no action=accept
     set-bgp-prepend-path=""

 1   chain=ospf-out prefix=23.246.50.1 invert-match=no action=accept
     set-bgp-prepend-path=""

 2   chain=ospf-out prefix=185.2.220.1 invert-match=no action=accept
     set-bgp-prepend-path=""

 3   chain=ospf-out prefix=0.0.0.0/0 prefix-length=0-32 invert-match=no
     action=discard set-bgp-prepend-path=""
```

Fonte: O Autor (2022)

A mesma lógica foi utilizada para a configuração OSPF nos roteadores de *Capanema*, *Santa Luzia* e *Bragança*, onde é feita a conexão com o *Cliente 1*, *Cliente 2* e *Cliente 3*, respectivamente. A redistribuição de rotas diretamente conectadas foi habilitada para anunciar os endereços

IPs de ponto a ponto entre os clientes e os roteadores de borda. Um filtro de roteamento foi habilitado para que apenas esses endereços IPs (100.71.42.X) fossem redistribuídos. Nas figuras abaixo temos as configurações efetuadas nos três roteadores.

Figura 12 – Configuração OSPF em Capanema

```
[admin@CAPANEMA] > routing ospf instance print
Flags: X - disabled, * - default
 0  * name="default" router-id=10.255.8.0 distribute-default=never
      redistribute-connected=as-type-2 redistribute-static=no
      redistribute-rip=no redistribute-bgp=no redistribute-other-ospf=no
      metric-default=1 metric-connected=20 metric-static=20 metric-rip=20
      metric-bgp=auto metric-other-ospf=auto mpls-te-area=backbone
      mpls-te-router-id=Loopback in-filter=ospf-in out-filter=ospf-out
[admin@CAPANEMA] > routing ospf network print
Flags: X - disabled, I - invalid
 #    NETWORK            AREA
 0    10.254.0.24/30     backbone
 1    10.255.8.0/32      backbone
 2    10.254.0.12/30     backbone
 3    10.254.0.16/30     backbone
 4    10.254.0.20/30     backbone
[admin@CAPANEMA] > routing filter print
Flags: X - disabled
 0    chain=ospf-out prefix=100.71.42.0/30 invert-match=no action=accept
      set-bgp-prepend-path=""

 1    chain=ospf-out prefix=0.0.0.0/0 prefix-length=0-32 invert-match=no
      action=discard set-bgp-prepend-path=""
```

Fonte: O Autor (2022)

Figura 13 – Configuração OSPF em Santa Luzia

```
[admin@SANTA-LUZIA] > routing ospf instance print
Flags: X - disabled, * - default
 0  * name="default" router-id=10.255.12.0 distribute-default=never
      redistribute-connected=as-type-2 redistribute-static=no
      redistribute-rip=no redistribute-bgp=no redistribute-other-ospf=no
      metric-default=1 metric-connected=20 metric-static=20 metric-rip=20
      metric-bgp=auto metric-other-ospf=auto mpls-te-area=backbone
      mpls-te-router-id=Loopback in-filter=ospf-in out-filter=ospf-out
[admin@SANTA-LUZIA] > routing ospf network print
Flags: X - disabled, I - invalid
 #    NETWORK            AREA
 0    10.254.0.32/30     backbone
 1    10.254.0.20/30     backbone
 2    10.254.0.28/30     backbone
 3    10.255.12.0/32     backbone
 4    10.254.0.44/30     backbone
[admin@SANTA-LUZIA] > routing filter print
Flags: X - disabled
 0    chain=ospf-out prefix=100.71.42.4/30 invert-match=no action=accept
      set-bgp-prepend-path=""

 1    chain=ospf-out prefix=0.0.0.0/0 prefix-length=0-32 invert-match=no
      action=discard set-bgp-prepend-path=""
```

Fonte: O Autor (2022)

Figura 14 – Configuração OSPF em Bragança

```
[admin@BRAGANCA] > routing ospf instance print
Flags: X - disabled, * - default
 0  * name="default" router-id=10.255.10.0 distribute-default=never
      redistribute-connected=as-type-2 redistribute-static=no
      redistribute-rip=no redistribute-bgp=no redistribute-other-ospf=no
      metric-default=1 metric-connected=20 metric-static=20 metric-rip=20
      metric-bgp=auto metric-other-ospf=auto mpls-te-area=backbone
      mpls-te-router-id=Loopback in-filter=ospf-in out-filter=ospf-out
[admin@BRAGANCA] > routing ospf network print
Flags: X - disabled, I - invalid
 #    NETWORK            AREA
 0    10.254.0.24/30     backbone
 1    10.254.0.28/30     backbone
 2    10.255.10.0/32     backbone
[admin@BRAGANCA] > routing filter print
Flags: X - disabled
 0    chain=ospf-out prefix=100.71.42.8/30 invert-match=no action=accept
      set-bgp-prepend-path=""

 1    chain=ospf-out prefix=0.0.0.0/0 prefix-length=0-32 invert-match=no
      action=discard set-bgp-prepend-path=""
```

Fonte: O Autor (2022)

4.3.2 MPLS

Para a topologia inicial, a configuração do MPLS é bastante simples. Basta habilitar o protocolo LDP (*Label Distribution Protocol*) nas interfaces que fazem parte do backbone (o que não inclui as interfaces dos hosts *Netflix Svoip* e as interfaces que conectam diretamente aos clientes). Na Figura 15, temos um exemplo da configuração MPLS no roteador *Castanhal*. O *Transport-Address* escolhido para cada sessão é o endereço IP de loopback. Ele é usado para identificação durante a sessão TCP onde o LDP é executado (Juniper Networks, 2022).

Figura 15 – Configuração MPLS em Castanhal

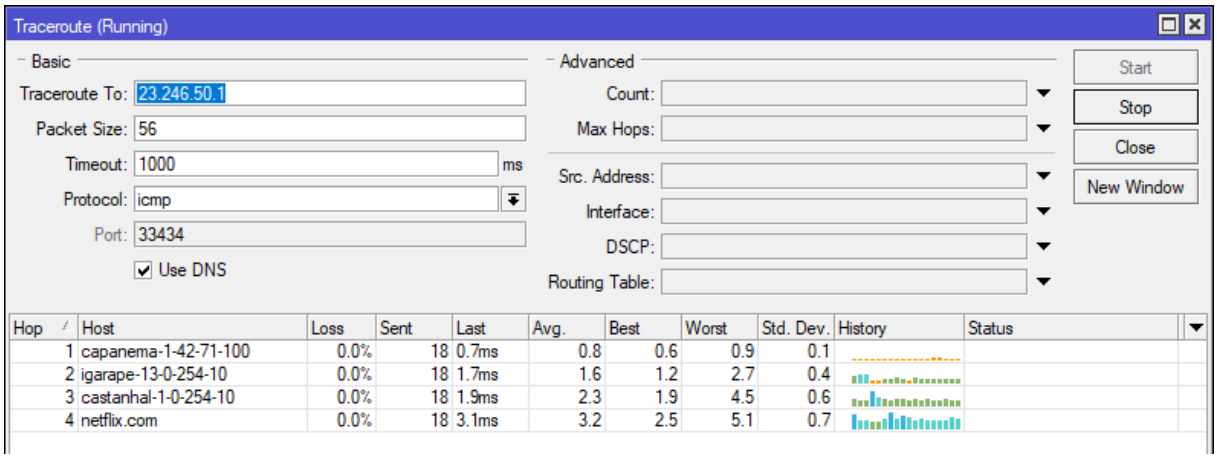
```
[admin@CASTANHAL] > mpls ldp print
                    enabled: yes
                     lsr-id: 0.0.0.0
            transport-address: 0.0.0.0
            path-vector-limit: 255
                   hop-limit: 255
                loop-detect: no
            use-explicit-null: no
  distribute-for-default-route: no
[admin@CASTANHAL] > mpls ldp interface print detail
Flags: X - disabled, I - invalid
 0   interface=ether6 hello-interval=5s hold-time=15s
     transport-address=10.255.0.0 accept-dynamic-neighbors=yes

 1   interface=ether7 hello-interval=5s hold-time=15s
     transport-address=10.255.0.0 accept-dynamic-neighbors=yes

 2   interface=ether4 hello-interval=5s hold-time=15s
     transport-address=10.255.0.0 accept-dynamic-neighbors=yes
```

Fonte: O Autor (2022)

4.3.3 Orientação do tráfego

Após implementação do cenário inicial, foi executado um *Traceroute* em cada cliente para os endereços IPs das aplicações de *Streaming* e *Voip* com o propósito de identificar os caminhos selecionados pelo protocolo OSPF. Na Figura 16, temos o *Traceroute* sendo executado no *Cliente 1* com destino ao servidor *Netflix*.

Figura 16 – Traceroute do Cliente 1 para Netflix

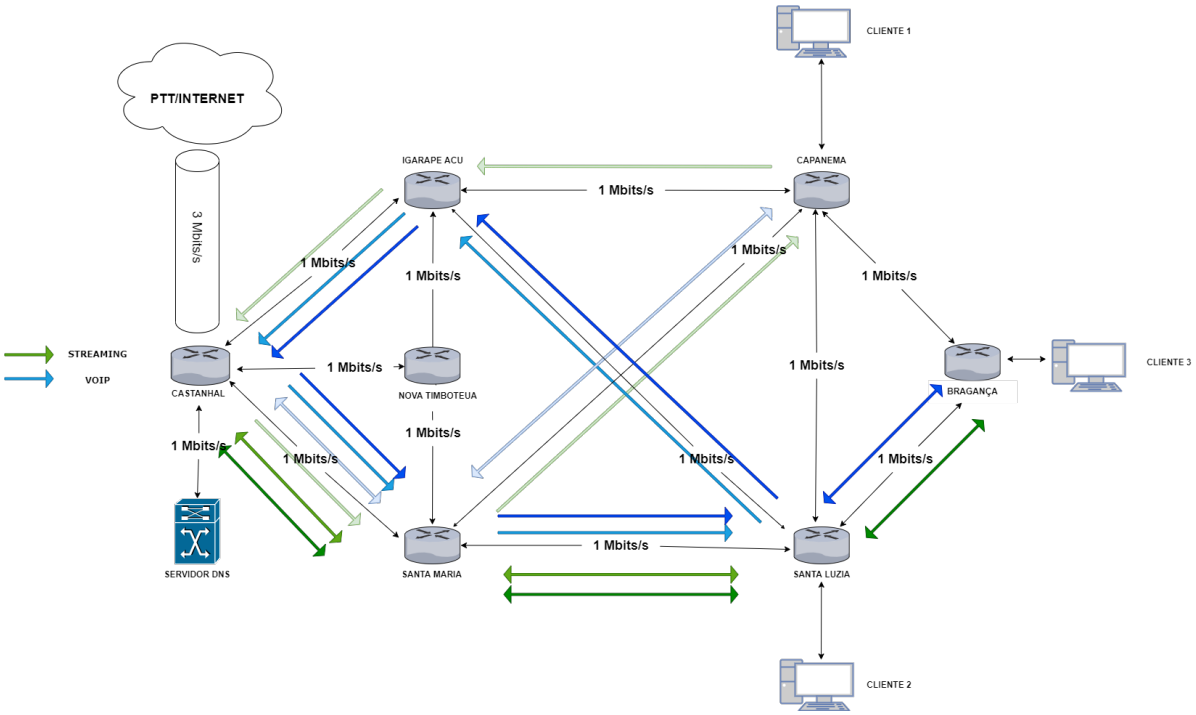


Fonte: O Autor (2022)

Como todos os links entre os roteadores possuem velocidade limitada a 1 Mbps, o OSPF fez o cálculo com base no número de saltos até o destino. Na Figura 17, temos a representação dos caminhos designados pelo OSPF para o encaminhamento do tráfego de cada cliente às aplicações de destino. Os indicadores em verde estão relacionados aos caminhos de cada cliente

para acesso ao serviço de *Streaming*. A cor mais clara está vinculada ao *Cliente 1*, a intermediária ao *Cliente 2* e a mais escura ao *Cliente 3*. O mesmo raciocínio vale para os indicadores em azul referente ao serviço de *Voip*.

Figura 17 – Orientação do tráfego na topologia inicial



Fonte: O Autor (2022)

Como pode ser observado pela Figura 17, alguns links do backbone acabam ficando ociosos enquanto outros ficam sobrecarregados. Um exemplo disso é o link entre *Castanhal* > *Santa Maria* > *Santa Luzia* > *Bragança* que está sendo utilizado para o tráfego de *Streaming* do *Cliente 2* e *Cliente 3* e ao mesmo tempo para o tráfego *Voip* do *Cliente 3*.

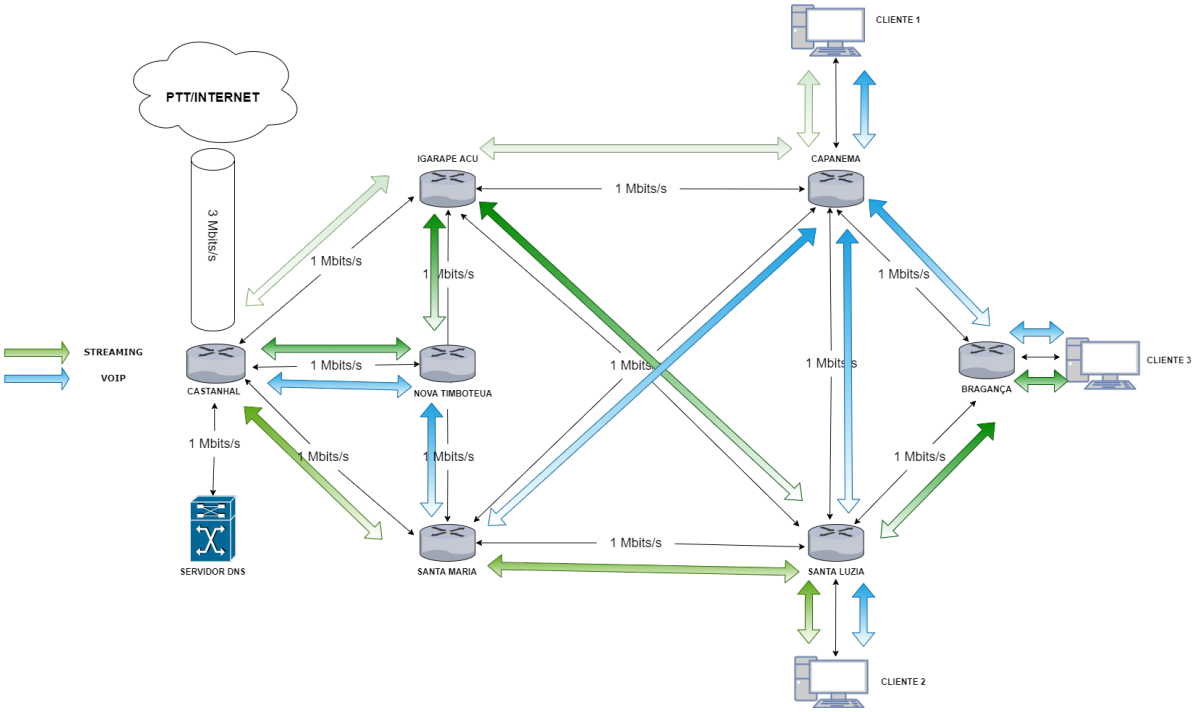
4.4 Cenário Final

As configurações da topologia que serão apresentadas neste tópico estão relacionadas ao cenário ideal proposto por este trabalho, onde será aplicado o MPLS Traffic Engineering (TE). Conforme já abordado em tópicos anteriores, através do MPLS TE podemos aplicar engenharia de tráfego na rede backbone através da inclusão de outras restrições ou parâmetros para análise de melhor caminho, como, por exemplo, a largura de banda. Além do mais, com o protocolo RSVP TE é possível definir, manualmente, rotas explícitas para orientar o tráfego da rede.

No cenário inicial, com apenas os protocolos OSPF e MPLS configurados, observou-se que os caminhos adotados para a orientação do tráfego geraram subutilização de determinados enlaces em relação a outros. Para contornar este problema, podemos planejar uma topologia com caminhos explícitos para cada região (cliente), com base nas aplicações que os mesmos irão

utilizar, através da criação de túneis unidirecionais com reserva de banda. Na Figura 18, temos novamente a topologia da rede porém com as orientações de tráfego planejadas.

Figura 18 – Topologia final com planejamento de tráfego



Fonte: O Autor (2022)

Com o planejamento demonstrado pela figura acima, é possível ter um link dedicado de tráfego *Streaming* para cada cliente. Para o tráfego destinado à aplicação de *Voip*, uma vez que este não requer uma alta largura de banda, pode-se dedicar um link específico para o acesso de todos os clientes à essa aplicação. Ela não sofrerá congestionamento pois o tráfego de *Streaming* não irá percorrer esse caminho - apenas entre os enlaces de *Castanhal* e *Nova Timboteua* - e irá atender as exigências para um rápido tempo de resposta e zero perda de pacotes.

Para a execução desse planejamento, é necessário criar túneis MPLS TE nos roteadores *Castanhal*, *Capanema*, *Santa Luzia* e *Bragança*. Nesses túneis serão especificados caminhos completos, pela rede backbone, para orientar o tráfego de cada cliente. Como os túneis são unidirecionais, é necessário criar um túnel nos roteadores de borda (*Capanema*, *Santa Luzia* e *Bragança*) para o tráfego de saída do cliente e túneis em *Castanhal* para o tráfego de retorno ao cliente. Na Figura 19, temos a configuração de caminhos explícitos para o cliente em Capanema. São esses caminhos que serão incorporados na criação dos túneis.

O primeiro caminho foi denominado como *Streaming-primary* onde os *hops* (saltos) estabelecidos são *Igarapé-Açu* (10.254.0.13) > *Castanhal* (10.254.0.1) para o acesso *Streaming* do *Cliente 1*. Esse caminho é primário, pois, além dele, foi criado um caminho secundário (*Streaming-backup*), onde os hops não serão definidos manualmente. Para este, será aplicado o protocolo CSPF (*Constrained Shortest Path First*) que será responsável pelo estabelecimento de um caminho de backup em caso de indisponibilidade do caminho primário.

Figura 19 – Caminhos ("Paths") criados em Capanema

```
[admin@CAPANEMA] > mpls traffic-eng tunnel-path print detail
Flags: X - disabled
 0    name="streaming-primary" use-cspf=no
      hops=10.254.0.13:strict,10.254.0.1:strict

 1    name="streaming-backup" use-cspf=yes hops=""

 2    name="voip-primary" use-cspf=no
      hops=10.254.0.17:strict,10.254.0.41:strict,10.254.0.37:strict

 3    name="voip-backup" use-cspf=yes hops=""
```

Fonte: O Autor (2022)

O mesmo padrão de configuração é adotado para o caminho destinado a aplicação de *Voip*, porém o caminho explícito segue a topologia planejada: *Santa Maria* (10.254.0.17) > *Nova Timboteua* (10.254.0.41) > *Castanhal* (10.254.0.37). Este também possui um caminho de backup que será definido via CSPF.

É importante notar que em todos os caminhos explícitos, os *hops* são definidos em modo *Strict*. No Mikrotik, existem dois modos que podem ser adotados: *Strict* e *Loose*. No modo *Loose*, é pressuposto que haja outros roteadores entre o *hop* anterior e o próximo *hop*, enquanto que no *Strict* é pressuposto que não haja nenhum roteador entre o *hop* anterior e o próximo *hop* (Mikrotik Wiki, 2022a). Podemos observar um exemplo disso entre os roteadores *Capanema* e *Santa Maria* onde não há nenhum *hop* entre eles. O caminho é diretamente conectado. Após a definição desses caminhos, é criada a interface túnel TE no roteador *Capanema* conforme demonstrado pela Figura 20.

Figura 20 – Interface TE em Capanema

```
[admin@CAPANEMA] > interface traffic-eng print
Flags: X - disabled, D - dynamic, R - running
 0    name="Streaming-path" mtu=1500 disable-running-check=no
      from-address=10.255.8.0 to-address=10.255.0.0 bandwidth=100kbps
      primary-path=streaming-primary secondary-paths=streaming-backup
      primary-retry-interval=1m record-route=yes bandwidth-limit=disabled
      auto-bandwidth-range=0bps auto-bandwidth-reserve=0%
      auto-bandwidth-avg-interval=5m auto-bandwidth-update-interval=1h

 1    name="Voip-path" mtu=1500 disable-running-check=no
      from-address=10.255.8.0 to-address=10.255.0.0 bandwidth=30kbps
      primary-path=voip-primary secondary-paths=voip-backup
      primary-retry-interval=1m record-route=yes bandwidth-limit=disabled
      auto-bandwidth-range=0bps auto-bandwidth-reserve=0%
      auto-bandwidth-avg-interval=5m auto-bandwidth-update-interval=1h
```

Fonte: O Autor (2022)

Como pode-se observar, o túnel será estabelecido entre o endereço (*From-address*) 10.255.8.0 que é o endereço IP de loopback de *Capanema* e o endereço (*To-address*) 10.255.0.0 que é o IP de loopback de *Castanhal*. A banda reservada para esse túnel é de 100kbps. Como

os túneis são unidirecionais e o tráfego de upload para o serviço de *Streaming* é menor que o de download, é reservado apenas 100Kbps para esse túnel. Sendo assim, a banda maior será reservada no túnel do roteador *Castanhal* em direção à *Capanema*.

Os parâmetros *Primary-path* e *Secondary-path* , observados na Figura 20, receberão os caminhos criados anteriormente. Vale ressaltar que o parâmetro *Bandwidth* não é um limitador de banda para o túnel, portanto um fluxo de dados maior ou menor pode passar por ele. Esse parâmetro serve para administração e planejamento, e caso seja identificado que há outras reservas de banda no mesmo caminho e a requisição de banda definida não pode ser atendida, o túnel não será estabelecido.

Além disso, um endereço IP de máscara /30 é atribuído nessas interfaces como pode ser observado na Figura 21. O outro endereço IP será definido nas interfaces de túneis criadas em *Castanhal*.

Figura 21 – Endereços IPs dos túneis TE em Capanema

```
[admin@CAPANEMA] > ip address print where interface=Streaming-path
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         INTERFACE
 0   10.99.99.2/30      10.99.99.0      Streaming-path
[admin@CAPANEMA] >
[admin@CAPANEMA] > ip address print where interface=Voip-path
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         INTERFACE
 0   10.88.88.2/30      10.88.88.0      Voip-path
```

Fonte: O Autor (2022)

Na Figura 22, temos as configurações realizadas em castanhal para os túneis sentido *Aplicações* > *Cliente 1*.

Figura 22 – Caminhos ("Paths") criados em Castanhal

```
[admin@CASTANHAL] > mpls traffic-eng tunnel-path print detail
Flags: X - disabled
 0   name="cliente1-streaming-primary" use-cspf=no
     hops=10.254.0.2:strict,10.254.0.14:strict

 1   name="cliente1-streaming-backup" use-cspf=yes hops=""

 6   name="cliente1-voip-primary" use-cspf=no
     hops=10.254.0.38:strict,10.254.0.42:strict,10.254.0.18:strict

 7   name="cliente1-voip-backup" use-cspf=yes hops=""
```

Fonte: O Autor (2022)

A configuração demonstrada pela Figura 23 segue o mesmo princípio criado em *Capanema*, porém na direção oposta. A banda reservada para *Streaming* também é maior (800Kbps) visto que o tráfego sentido *Downstream* é maior. Um endereço IP também é definido para a interface túnel criada, conforme pode ser observado na Figura 24.

Figura 23 – Interface TE em Castanhal

```
[admin@CASTANHAL] > interface traffic-eng print
Flags: X - disabled, D - dynamic, R - running
 0  R name="Cliente1-Streaming" mtu=1500 disable-running-check=no
      from-address=10.255.0.0 to-address=10.255.8.0 bandwidth=800kbps
      primary-path=cliente1-streaming-primary
      secondary-paths=cliente1-streaming-backup primary-retry-interval=1m
      record-route=yes bandwidth-limit=disabled auto-bandwidth-range=0bps
      auto-bandwidth-reserve=0% auto-bandwidth-avg-interval=5m
      auto-bandwidth-update-interval=1h

 1  R name="Cliente1-Voip" mtu=1500 disable-running-check=no
      from-address=10.255.0.0 to-address=10.255.8.0 bandwidth=30kbps
      primary-path=cliente1-voip-primary secondary-paths=cliente1-voip-backup
      primary-retry-interval=1m record-route=yes bandwidth-limit=disabled
      auto-bandwidth-range=0bps auto-bandwidth-reserve=0%
      auto-bandwidth-avg-interval=5m auto-bandwidth-update-interval=1h
```

Fonte: O Autor (2022)

Figura 24 – Endereços IPs na interface TE em Castanhal

```
[admin@CASTANHAL] > ip address print where interface="Cliente1-Streaming"
Flags: X - disabled, I - invalid, D - dynamic
 #    ADDRESS            NETWORK         INTERFACE
 0    ;;; TUNNEL STR CLIENTE 1
      10.99.99.1/30      10.99.99.0      Cliente1-Streaming
[admin@CASTANHAL] >
[admin@CASTANHAL] > ip address print where interface="Cliente1-Voip"
Flags: X - disabled, I - invalid, D - dynamic
 #    ADDRESS            NETWORK         INTERFACE
 0    ;;; TUNNEL VOIP CLIENTE 1
      10.88.88.1/30      10.88.88.0      Cliente1-Voip
```

Fonte: O Autor (2022)

Com os endereços IPs definidos em ambos os lados, e havendo comunicação entre os túneis - as interfaces túneis podem não subir caso a requisição de banda não possa ser atendida - podemos criar regras de roteamento que identifiquem tráfegos para determinadas aplicações e encaminhem o tráfego através desses túneis. Essas políticas de roteamento devem ser criadas em ambos os lados. Na Figura 25 temos as configurações em *Capanema*.

Figura 25 – Rota estática via túnel em Capanema

```
[admin@CAPANEMA] > ip route print detail where distance=1
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 0    S  ;;; ROTA STR VIA TUNNEL
         dst-address=23.246.50.1/32 gateway=10.99.99.1
         gateway-status=10.99.99.1 unreachable distance=1 scope=30
         target-scope=10

 1    S  ;;; ROTA VOIP VIA TUNNEL
         dst-address=185.2.220.1/32 gateway=10.88.88.1
         gateway-status=10.88.88.1 unreachable distance=1 scope=30
         target-scope=10
```

Fonte: O Autor (2022)

Podemos notar que todo tráfego destinado ao endreço IP da Netflix (23.246.50.1/32) será encaminhado pelo *Gateway* do túnel *Streaming* que conecta em *Castanhal* (10.99.99.1). O mesmo vale para o tráfego com destino ao servidor *Voip* (185.2.220.1/32) que será encaminhado através do *Gateway* do túnel *Voip* em Castanhal (10.88.88.1). Na Figura 26, temos a mesma política aplicada em *Castanhal* para o tráfego de retorno.

Figura 26 – Rota estática via túnel em Castanhal

```
[admin@CASTANHAL] > ip route print detail where dst-address=100.71.42.2/32
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 0 A S  ;;; ROTA STR CLIENTE 1 VIA TUNNEL
        dst-address=100.71.42.2/32 gateway=10.99.99.2
        gateway-status=10.99.99.2 reachable via  Cliente1-Streaming
        distance=1 scope=30 target-scope=10 routing-mark=cl1-str

 1 A S  ;;; ROTA VOIP CLIENTE 1 VIA TUNNEL
        dst-address=100.71.42.2/32 gateway=10.88.88.2
        gateway-status=10.88.88.2 reachable via  Cliente1-Voip distance=1
        scope=30 target-scope=10 routing-mark=cl1-voip
```

Fonte: O Autor (2022)

As mesmas configurações foram aplicadas para os roteadores de *Santa Luzia* e *Bragança*, seguindo a ideia da topologia apresentada na Figura 18. Em *Castanhal* foram criados outros túneis e políticas de roteamento para o retorno do tráfego aos clientes dessas regiões.

4.5 Simulação de Tráfego

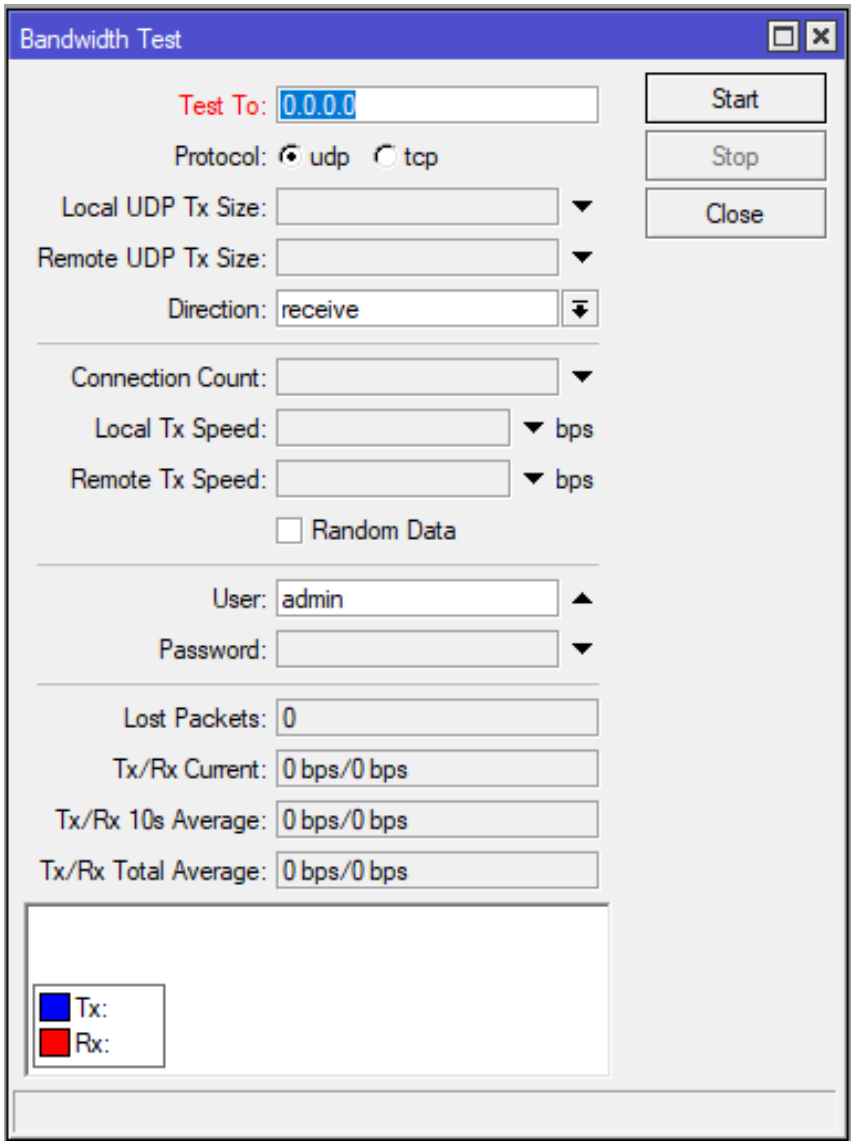
Com base nos dois cenários apresentados previamente, foram executadas simulações de tráfego destinadas às aplicações de *Streaming* e *Voip*, originadas simultaneamente dos clientes das três regiões, com o intuito de avaliar o desempenho da rede backbone. Para esta finalidade, foi utilizado a ferramenta de teste de velocidade disponível nos roteadores Mikrotik, para simular o tráfego de *Streaming*, e empregado o recurso de *UserParameter*, que será descrito mais à frente, para simular o tráfego *Voip*.

4.5.1 Simulação do Tráfego Streaming

O *Bandwidth Test* está disponível na aba *Tools* (Ferramentas) dos roteadores Mikrotik e com ele é possível executar testes de largura de banda para qualquer servidor disponível, inclusive para outro roteador Mikrotik que esteja com o servidor de teste habilitado (Mikrotik Wiki, 2022c). Esta ferramenta foi adotada para simular o tráfego de clientes para serviços de *Streaming*, visto que o consumo para esses tipos de aplicações gera um tráfego alto.

Como pode-se observar na Figura 27, existe um campo para adicionar o endereço IP onde se deseja realizar o teste. Em cada cliente foi inserido o endereço IP do servidor *Netflix* que está habilitado para receber requisições de teste de banda.

Figura 27 – Ferramenta de teste de banda no Mikrotik



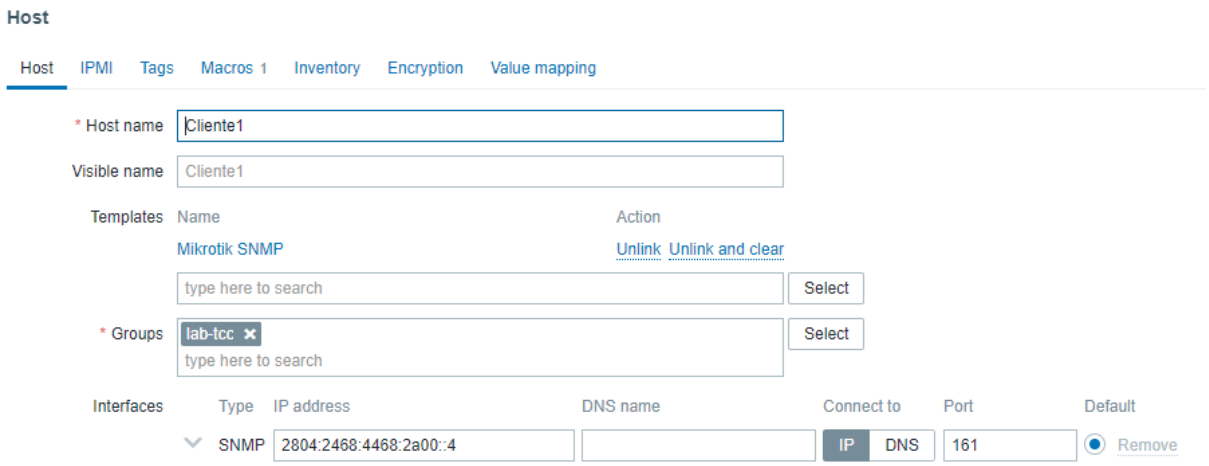
Fonte: O Autor (2022)

Os testes foram executados simultaneamente em cada cliente durante 15 minutos. No servidor Zabbix foi criado um item com o objetivo de gerar gráficos das taxas de transferência nas interfaces de cada cliente. Deste modo, é possível avaliar o tráfego de cada cliente após a finalização dos testes. Na Figura 28 temos a configuração de um host no Zabbix para o *Cliente 1*.

O tipo de captura escolhido foi o SNMP na porta 161. Esta porta é a padrão do protocolo SNMP (*Simple Network Management Protocol*) e está habilitada em todos os roteadores da topologia. Além disso, foi adicionado o template da Mikrotik que vem com alguns itens pré-configurados, o que inclui um item que realiza a captura de bytes de entrada e saída de uma interface e gera gráficos a partir dessa captura.

Observa-se que o parâmetro *IP address* recebeu um endereço IPv6. Esse endereço foi atribuído ao *Cliente 1* através da conexão com a nuvem no EVE-NG que põe os hosts em modo *Bridge* com o adaptador de *Ethernet* do computador local, conforme já explicado em tópicos

Figura 28 – Monitoramento para o Cliente 1 (Zabbix)



Fonte: O Autor (2022)

anteriores. Como o servidor Zabbix está localizado em outra máquina, foi necessário estabelecer essa comunicação via IPv6.

Para realizar a avaliação posterior dos resultados, podemos definir que o valor ideal de consumo simultâneo dos clientes para aplicação *Streaming* deverá chegar ao valor aproximado de 800Kbps - será necessário avaliar com unidades de grandeza menores devido às limitações de banda do ambiente de simulação - e para isso, foi adicionado três *Queues* no roteador *Netflix*, que limita o tráfego de cada cliente para 800Kbps, uma vez que a ferramenta de *Bandwidth Test* executa os testes com base na limite de banda da interface. As *Queues* nos roteadores Mikrotik são utilizadas para limitar ou priorizar determinados tráfegos (Mikrotik Wiki, 2022b).

4.5.2 Simulação do tráfego Voip

Para o teste de tráfego *Voip*, como a métrica a ser avaliada é o delay ou tempo de resposta (*Round Trip Time*), o teste ideal é efetuar um ping durante o intervalo de tempo do teste de tráfego streaming (15 minutos). Seria possível executar um ping ininterrupto em cada cliente, porém, não há uma maneira ideal de capturar - através do Zabbix - os pacotes transmitidos em cada cliente para avaliação posterior.

Para solucionar esse problema, foi adotado o recurso de *UserParameter*, que permite realizar a captura de parâmetros que não estão pré-definidos no Zabbix. O *UserParameter* deve ser adicionado no arquivo de configuração “zabbix_agentd.conf” que está localizada no diretório */etc/zabbix/* do servidor onde o Zabbix está instalado. Para adicionar o *UserParameter*, deve-se utilizar a seguinte sintaxe “UserParameter=<chave>,<comando>” (Zabbix SIA, 2022). A chave pode ser um nome qualquer e é utilizada para a criação do item no Zabbix.

Além da chave, é necessário adicionar o comando no *UserParameter*. O comando adicionado para este trabalho permite que o servidor Zabbix acesse os hosts Cliente 1, 2 e 3 via

protocolo SSH e execute um ping para o servidor *Svoip*. Esse acesso é feito a cada 30 segundos. Na Figura 29 temos o comando criado para cada cliente.

Figura 29 – User Parameter no servidor Zabbix

```
UserParameter=rttone.check,sshpass -p "12345678" ssh zabbix@2804:2468:4468:2a00::4 "ping svoip.com count
 1" | grep = | awk '{print $4}' | tr -dc '0-9' | awk '{printf($1/1000)}'

UserParameter=rtttwo.check,sshpass -p "12345678" ssh zabbix@2804:2468:4468:2a00::9 "ping svoip.com count
 1" | grep = | awk '{print $4}' | tr -dc '0-9' | awk '{printf($1/1000)}'

UserParameter=rttthree.check,sshpass -p "12345678" ssh zabbix@2804:2468:4468:2a00::8 "ping svoip.com cou
nt 1" | grep = | awk '{print $4}' | tr -dc '0-9' | awk '{printf($1/1000)}'
```

Fonte: O Autor (2022)

Para que esse acesso funcionasse, foi criado um usuário chamado “zabbix” em cada cliente. Assim que o acesso SSH é estabelecido, o comando “ping svoip.com count 1” é executado, fazendo com um que um pacote ICMP (*Internet Control Message Protocol*) seja enviado ao destino. Após esse comando, é obtido um resultado como ilustrado na Figura 30.

Figura 30 – Comando via sshpass

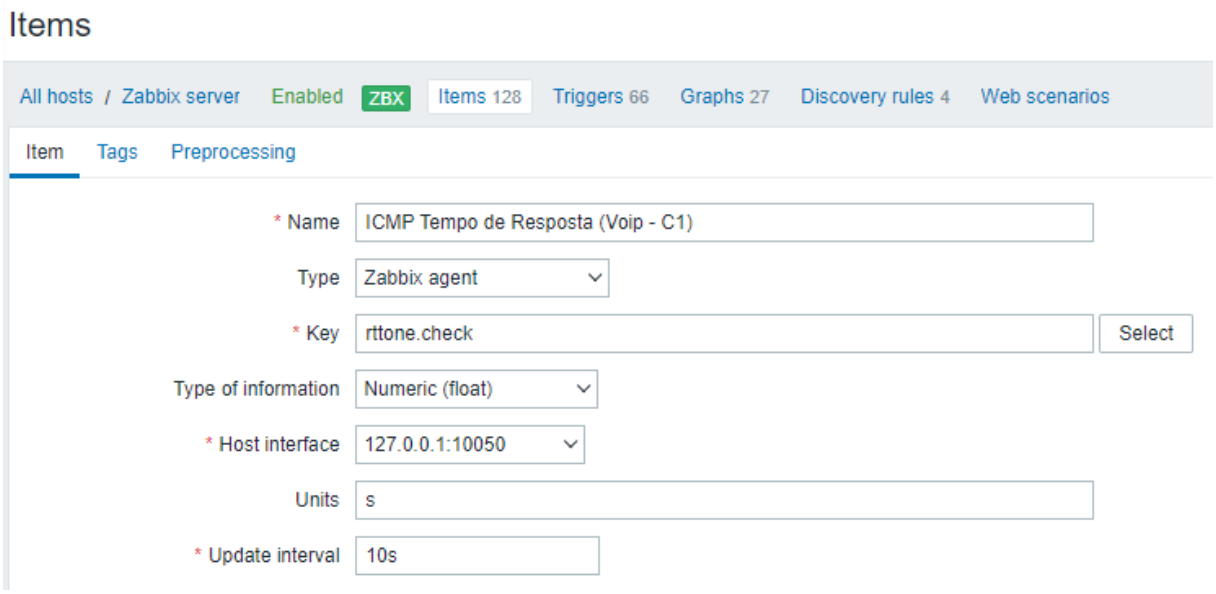
```
elzio@DESKTOP-QT8MC6L:~$ sshpass -p "" ssh admin@192.168.0.199 "ping svoip.com count 1"
  SEQ HOST                                     SIZE TTL TIME  STATUS
    0 185.2.220.1                                56  64 1ms
    sent=1 received=1 packet-loss=0% min-rtt=1ms avg-rtt=1ms max-rtt=1ms
```

Fonte: O Autor (2022)

O dado que interessa para a captura e armazenamento no Zabbix é do valor "rtt"(o resultado de "min", "avg"e "max"são todos iguais pois apenas um pacote é enviado). Para extrair somente esse valor, foi utilizado comandos do Linux para filtros do resultado, como *awk* e *tr*. No final, o valor é divido por 1000 pois o Zabbix é configurado para capturar o valor em segundos e o resultado que obtemos está em milisegundos.

A criação do Item no Zabbix para o *Cliente 1* está representada na Figura 31. O item é criado no host *Zabbix Server* e o tipo utilizado é *Zabbix agent*. Observe que a chave utilizada para o *Cliente 1* é a que foi criada no *UserParameter*. A interface de captura é no próprio *localhost*, que é o servidor onde o Zabbix está instalado, e a unidade de medida é em segundos, conforme já mencionado.

Figura 31 – Criação de item para captura do tráfego Voip (Zabbix)



Fonte: O Autor (2022)

Outros itens foram criados para o Cliente 2 e 3. Deste modo, podemos monitorar o tempo de resposta para o servidor *SVoip* a todo momento, inclusive durante os testes de tráfego de *Streaming*, e avaliar, através de gráficos gerados a partir desse item, se houve falhas ou congestionamento para essa aplicação.

Para a avaliação de resultados dos testes de tráfego *voip*, não foi definido valores ideais. Para este caso, será avaliado os valores de RTT (*Round Trip Time*) nos resultados de ambos cenários e se houve perda de pacotes.

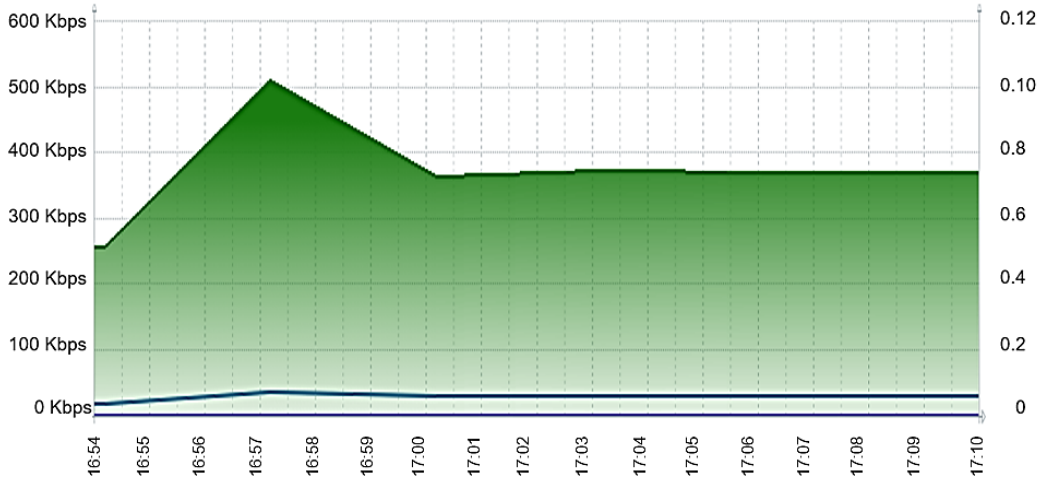
5 RESULTADOS

Neste tópico será apresentado os resultados das simulações de tráfego executadas nos dois cenários (com MPLS TE e sem MPLS TE), conforme descrito no tópico 4.5.

5.1 Sem MPLS TE (Tráfego Streaming)

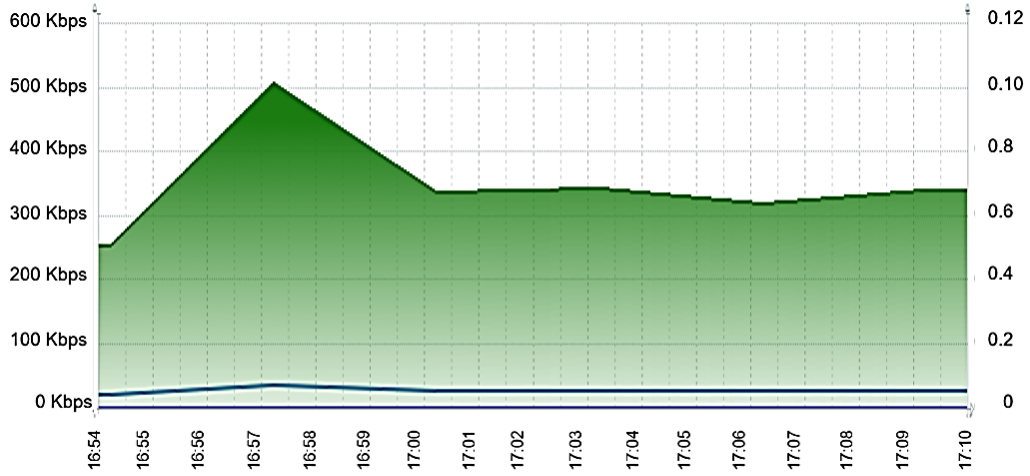
O gráfico da Figura 32 está relacionado ao tráfego do *Cliente 1*, destinado à aplicação de *Streaming*, durante um intervalo de tempo de 15 minutos. Nota-se que o uso de banda alcançou valores na média de 370Kbps, com pico máximo de 500Kbps. Bem abaixo do ideal definido para a aplicação neste cenário (800Kbps). O resultado para o *Cliente 2*, apresentado na Figura 33, se assemelha ao do *Cliente 1*.

Figura 32 – Tráfego streaming do cliente 1



Fonte: O Autor (2022)

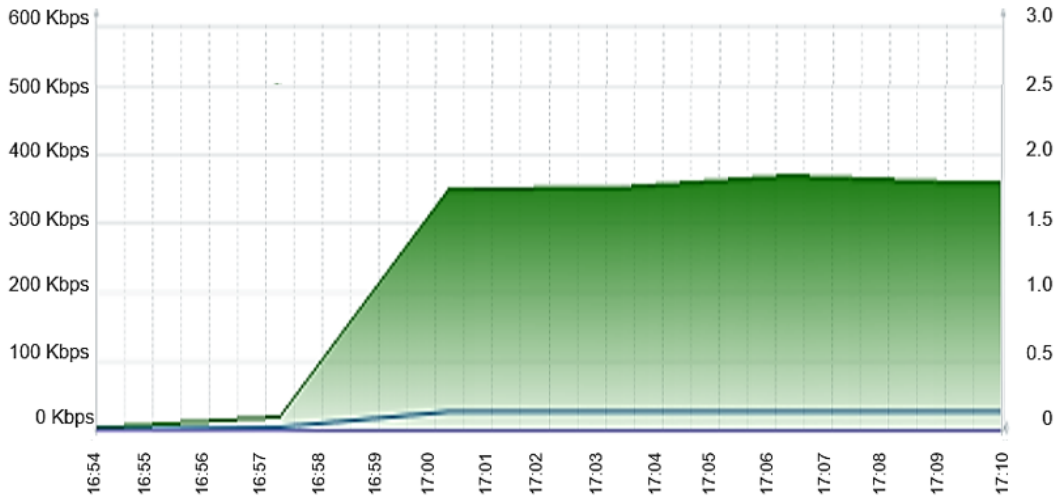
Figura 33 – Tráfego streaming do cliente 2



Fonte: O Autor (2022)

Na Figura 34, temos os resultados de tráfego do *Cliente 3*, que difere um pouco dos dois primeiros clientes. Observa-se que o cliente ficou um período de 3 minutos sem nenhum consumo de banda. Ao verificar os caminhos escolhidos pelo OSPF, demonstrados na Figura 17 que foi apresentada no subtópico 4.3.3, é possível notar que o link entre *Castanhal* > *Santa Maria* foi o escolhido para o encaminhamento do tráfego entre o servidor *Netflix* e os 3 clientes. Esse é o enlace que possui maior congestionamento do backbone. Além disso, apesar dos testes de tráfego serem realizados simultaneamente em cada cliente, o teste de banda no *Cliente 3* foi iniciado por último. Dessa forma, o mesmo foi afetado inicialmente pois não havia banda disponível.

Figura 34 – Tráfego streaming do cliente 3



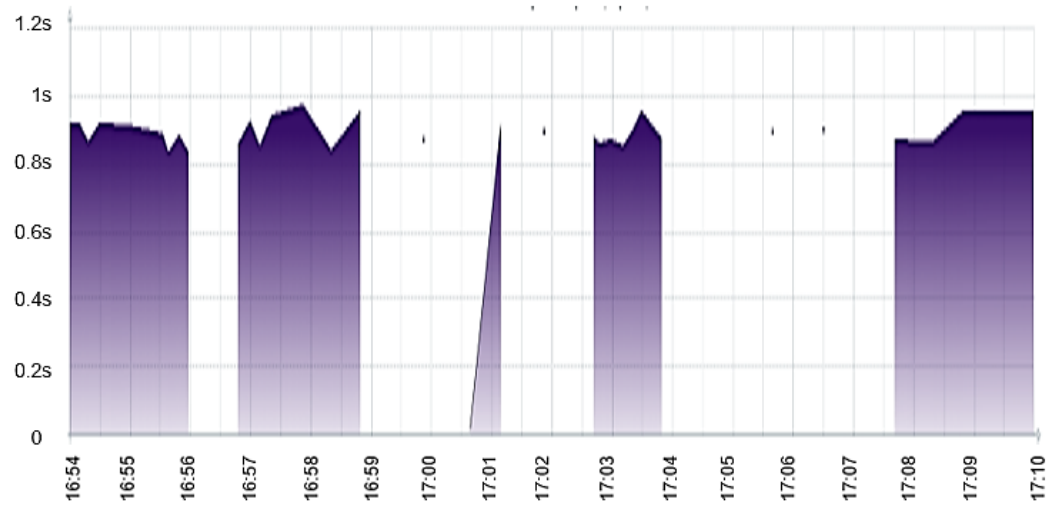
Fonte: O Autor (2022)

Decorrido o tempo de 3 minutos, o *Cliente 3* conseguiu gerar um tráfego na média de 366Kbps. É notório que essa mudança se reflete nos gráficos do Cliente 1 e 2 onde houve uma diminuição na taxa de dados transmitida a partir do tempo 16:57. A conclusão que pode ser tirada disso é que, durante o teste, houve uma alteração automática no caminho OSPF adotado para um dos clientes. Como não havia consumo para o *Cliente 3*, o roteador pode ter interpretado que houve indisponibilidade no link e adotou um caminho alternativo. De qualquer forma, mesmo com essa correção o resultado para os três clientes foi bem abaixo do ideal definido para esse cenário.

5.2 Sem MPLS TE (Tráfego Voip)

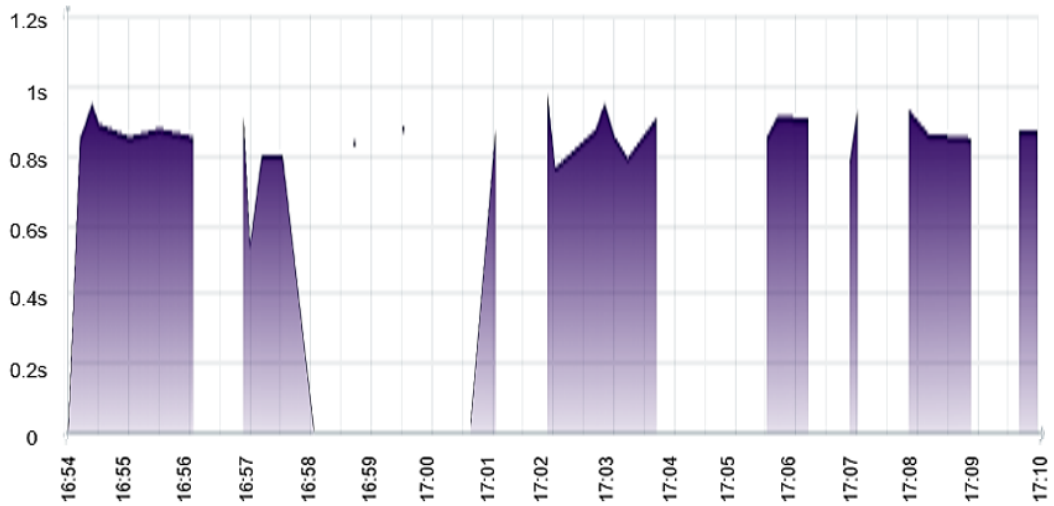
Ao observar a orientação do tráfego apresentada na Figura 17 do subtópico 4.3.3, é possível deduzir que os caminhos escolhidos pelo OSPF não só irão gerar congestionamento para serviços de *Streaming* como também para os serviços de *Voip*. Nas figuras abaixo temos o gráfico de testes de ping, que são efetuados em intervalos de 30 segundos, para os Clientes 1, 2 e 3, respectivamente.

Figura 35 – Tráfego voip do cliente 1



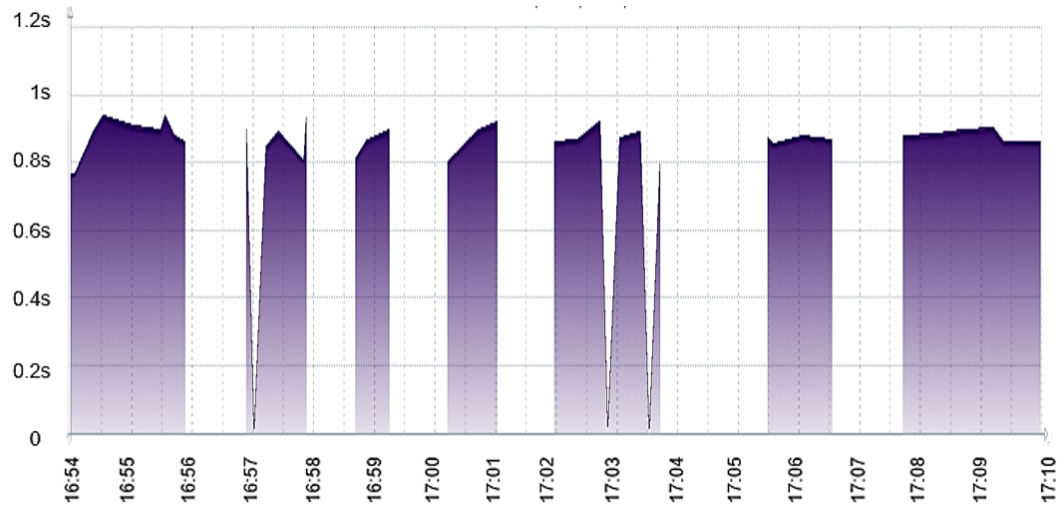
Fonte: O Autor (2022)

Figura 36 – Tráfego voip do cliente 2



Fonte: O Autor (2022)

Figura 37 – Tráfego voip do cliente 3



Fonte: O Autor (2022)

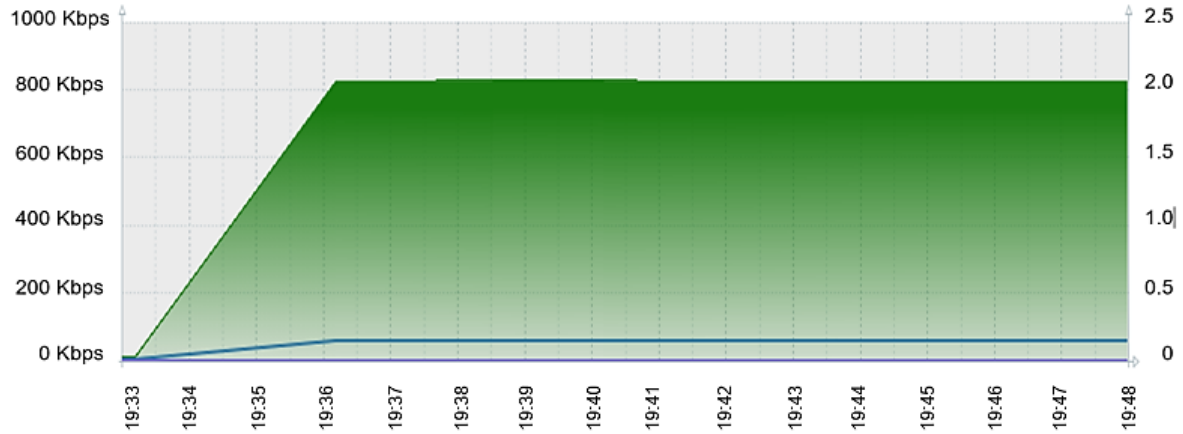
Como pode-ser observar nos gráficos acima, o tempo de resposta dos testes de ping, efetuadas a partir dos três clientes para o servidor voip, obtiveram valores consideravelmente altos, visto que a comunicação entre dispositivos de redes, mesmo em áreas geográficas diferentes, é medida em milissegundos (ms) e o resultados obtidos mostram os valores chegando a 1 segundo. Os espaços em branco nos gráficos apresentados mostram que, em determinados períodos, os pacotes não chegaram ao destino ou não retornaram, ou seja, houve perda de pacotes.

Esses resultados são consequência do congestionamento de enlaces na rede simulada, decorrentes dos caminhos adotados pelo protocolo OSPF. Com base nesses resultados, é válido afirmar que o presente cenário não é o ideal em uma rede backbone, principalmente para aplicações como voip que precisam de um tempo de resposta menor e que esteja livre de perda de pacotes.

5.3 Com MPLS TE (Tráfego Streaming)

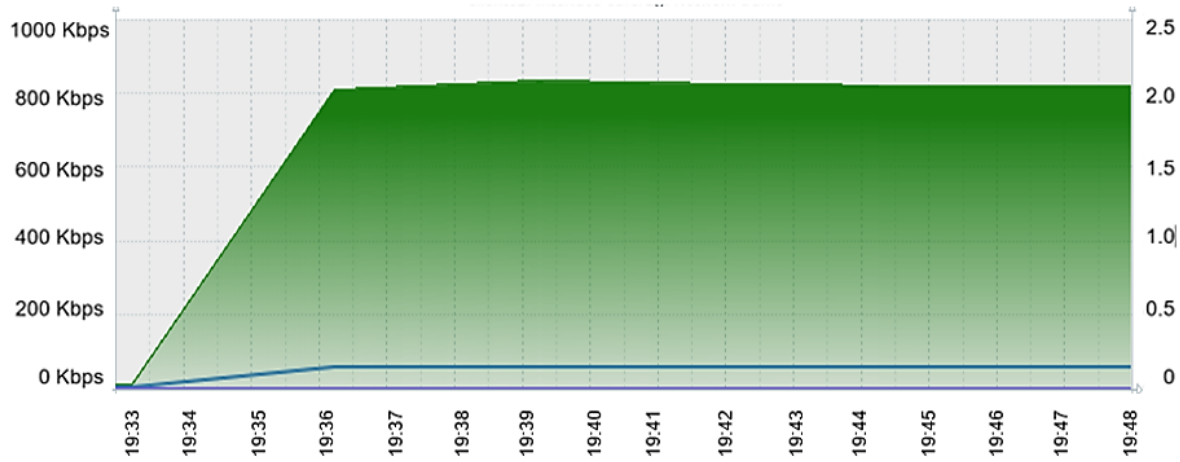
Com as configurações aplicadas e descritas no tópico 4.4, foi realizada outra simulação de tráfego, com tempo de 15 minutos, com destino a aplicação de *Streaming*. Abaixo temos o gráfico dos clientes 1, 2 e 3, respectivamente.

Figura 38 – Tráfego streaming com MPLS TE do cliente 1



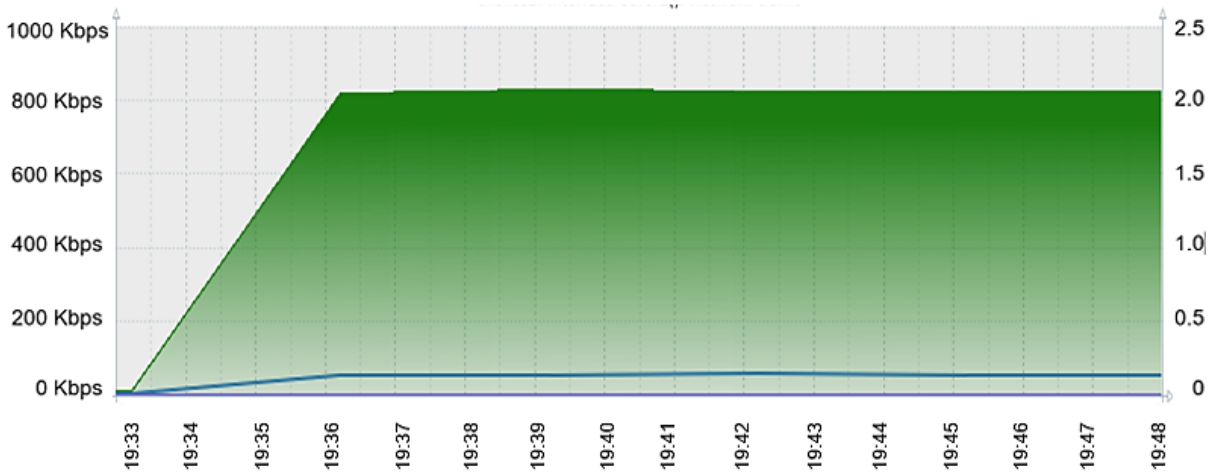
Fonte: O Autor (2022)

Figura 39 – Tráfego streaming com MPLS TE do cliente 2



Fonte: O Autor (2022)

Figura 40 – Tráfego streaming com MPLS TE do cliente 3



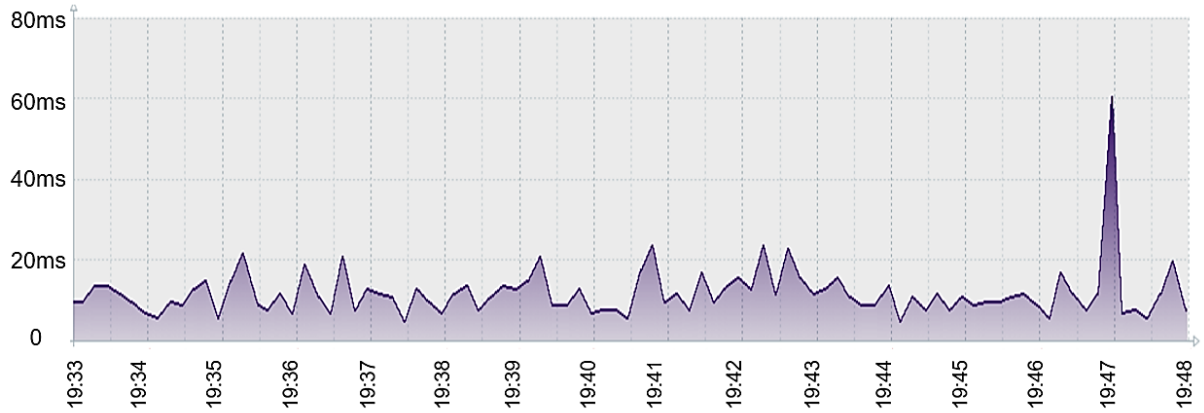
Fonte: O Autor (2022)

Com o uso de rotas explícitas e reserva de banda, foi possível alcançar uma estabilidade no tráfego dos três clientes, evitando a subutilização dos links do backbone. Dessa forma, foi possível alcançar a taxa de transmissão ideal definida para esse cenário, que é de 800Kbps.

5.4 Com MPLS TE (Tráfego Voip)

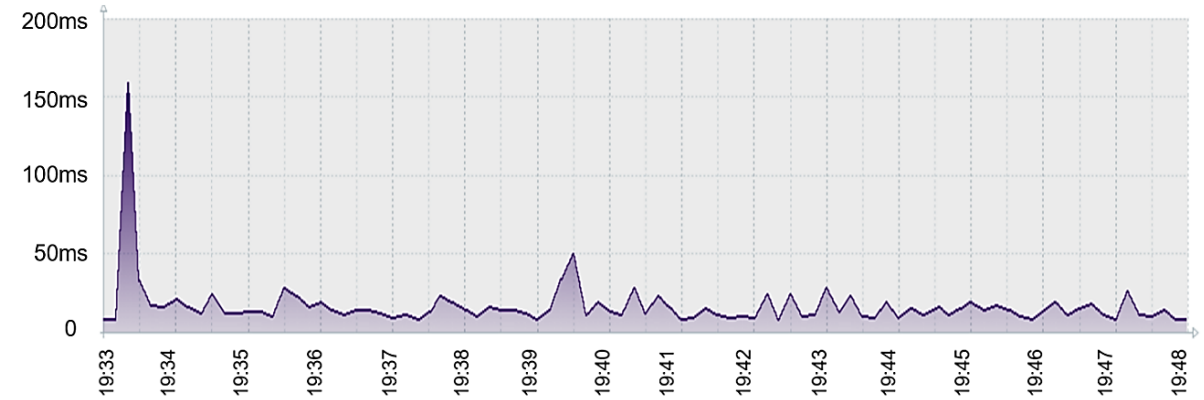
Com o uso dedicado de um link para o tráfego *voip*, descrita no tópico 4.4, foi possível alcançar valores de RTT (*Round Trip Time*) satisfatórios, com valores na média de 20ms e zero perdas de pacotes. As figuras abaixo mostram os resultados do tempo de resposta para o cliente 1, 2 e 3, respectivamente.

Figura 41 – Tráfego voip com MPLS TE do cliente 1



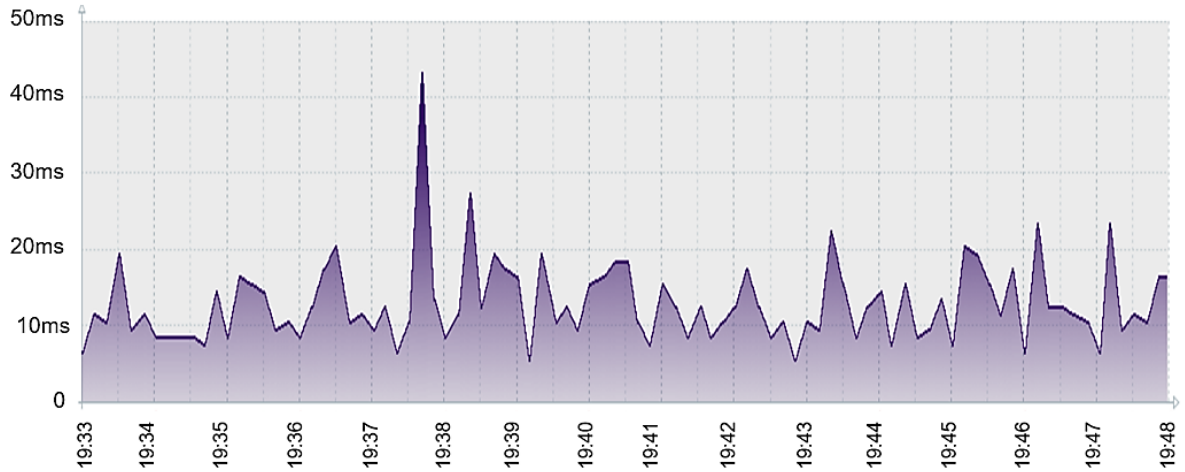
Fonte: O Autor (2022)

Figura 42 – Tráfego voip com MPLS TE do cliente 2



Fonte: O Autor (2022)

Figura 43 – Tráfego voip com MPLS TE do cliente 3



Fonte: O Autor (2022)

6 CONCLUSÃO

O presente estudo teve como objetivo avaliar o desempenho do tráfego em uma topologia de uma rede backbone, comparando a aplicação do protocolo MPLS TE, para a orientação do tráfego da rede, com os métodos de roteamento baseado somente em IP/MPLS.

Com base nos resultados obtidos, identificou-se que no cenário com MPLS TE implementado, houve um melhor desempenho no tráfego para as duas aplicações testadas (*streaming* e *voip*) pois, através de estabelecimento de túneis TE, foi possível orientar o tráfego a partir de caminhos planejados para diferentes endereços IPs de destino. Com isso, foi possível alcançar taxas de tráfego para os serviços de streaming na faixa de 800Kbps (limite estabelecido para a simulação) e tempos de resposta de ICMP na média de 20ms, sem perdas de pacotes.

Podemos expandir esse trabalho para cenários reais de um provedor de internet, em que determinadas aplicações requerem maior prioridade e qualidade de serviço (QoS), como streaming de vídeo, chamadas de voz e vídeo, jogos online etc. Para atender essas demandas, não basta apenas ter links de maior capacidade, mas faz-se necessário aplicar engenharia de tráfego a fim de definir melhores caminhos que não apenas sigam o modelo de "caminho com menor custo", mas também levem em consideração critérios como largura de banda, menor delay, jitter, perda de pacote entre outros.

Para trabalhos futuros, este projeto pode utilizar a mesma topologia para avaliar o estabelecimento de caminhos definidos apenas pelo CSPF (*Constrained Shortest Path First*) e verificar a eficiência destes em comparação com os caminhos definidos manualmente.

REFERÊNCIAS

Cetic.br. **A cada dez provedores de acesso à Internet no Brasil, nove oferecem fibra óptica aos clientes, revela pesquisa do Cetic.br**. 2021. Disponível em: <https://cetic.br/pt/noticia/a-cada-dez-provedores-de-acesso-a-internet-no-brasil-nove-oferecem-fibra-optica-aos-clientes-revela-pesqu>. Acesso em: 25 de maio 2022.

Cisco Press. **MPLS Traffic Engineering**. 2006. Disponível em: <https://www.ciscopress.com/articles/article.asp?p=426640&seqNum=2>. Acesso em: 13 de maio 2022.

DAYANAND, L. N.; GHORBANI, B.; VAGHRI, S. A survey on the use of gns3 for virtualizing computer networks. IASET, p. 50, 2016.

FORTZ, B.; THORUP, M. Internet traffic engineering by optimizing ospf weights. In: IEEE. **Proceedings IEEE INFOCOM 2000. conference on computer communications. Nineteenth annual joint conference of the IEEE computer and communications societies (Cat. No. 00CH37064)**. [S.l.], 2000. v. 2, p. 519–528.

HODZIC, H.; ZORIC, S. Traffic engineering with constraint based routing in mpls networks. In: IEEE. **2008 50th International Symposium ELMAR**. [S.l.], 2008. v. 1, p. 269–272.

Huawei Technologies. **What Is MPLS?** 2022. Disponível em: <https://support.huawei.com/enterprise/br/doc/EDOC1100118961>. Acesso em: 10 de maio 2022.

Huawei Technologies Co. **LDP Working Mechanism**. 2018. Disponível em: <https://support.huawei.com/enterprise/en/doc/EDOC1100116685/22b0901d/ldp-working-mechanism#EN-US_CONCEPT_0177107509>. Acesso em: 14 de maio 2022.

Juniper Networks. **transport-address**. 2022. Disponível em: <https://www.juniper.net/documentation/us/en/software/junos/mpls/topics/ref/statement/transport-address-edit-protocols-ldp.html>. Acesso em: 20 de maio 2022.

KUROSE, J. F. **Computer networking: A top-down approach featuring the internet, 3/E**. [S.l.]: Pearson Education India, 2005.

LEE, Y.; MUKHERJEE, B. Traffic engineering in next-generation optical networks. **IEEE communications surveys & tutorials**, IEEE, v. 6, n. 3, p. 16–33, 2004.

Mikrotik Wiki. **Manual:CHR**. 2020. Disponível em: <>. Acesso em: 13 de maio 2022.

Mikrotik Wiki. **Manual:Interface/Bonding**. 2020. Disponível em: <https://wiki.mikrotik.com/wiki/Manual:Interface/Bonding>. Acesso em: 20 de maio 2022.

Mikrotik Wiki. **Manual:MPLS/Traffic-eng**. 2022. Disponível em: <https://wiki.mikrotik.com/wiki/Manual:MPLS/Traffic-eng>. Acesso em: 20 de maio 2022.

Mikrotik Wiki. **Manual:Queue**. 2022. Disponível em: <https://wiki.mikrotik.com/wiki/Manual:Queue>. Acesso em: 20 de maio 2022.

Mikrotik Wiki. **Manual:Tools/Bandwidth Test**. 2022. Disponível em: <https://wiki.mikrotik.com/wiki/Manual:Tools/Bandwidth_Test>. Acesso em: 20 de maio 2022.

MOY, J. et al. Ospf version 2 (rfc 2328). **Ascend Communications Inc**, 1998.

Network Lessons. **OSPF Multi-Area Configuration**. 2022. Disponível em: <https://networklessons.com/cisco/ccna-routing-switching-icnd2-200-105/ospf-multi-area-configuration>. Acesso em: 13 de maio 2022.

Network Lessons. **OSPF Router ID**. 2022. Disponível em: <https://networklessons.com/ospf/ospf-router-id>. Acesso em: 20 de maio 2022.

OLIVEIRA, J. Mário Alexandre Melo de. **Análise e otimização de roteamento em Backbones OSPF utilizando MPLS-TE**. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2011.

OLIVEIRA, V. C. Simulador eve-ng em projetos de redes heterogêneas: um estudo sobre a importância da simulação em redes de computadores. **Research, Society and Development**, v. 9, n. 11, p. e1199119562–e1199119562, 2020.

RIZZETTI, T. A. et al. Methods of availability assurance for communication of pmu in a smart grid based on ip protocol. In: IEEE. **2014 49th International Universities Power Engineering Conference (UPEC)**. [S.l.], 2014. p. 3.

ROSEN, E. et al. Multiprotocol label switching architecture. rfc 3031, January, 2001.

SCHARF, A. L. Implantação de engenharia de tráfego com mpls-te em rede wan. 2017.

SHOKHIN, A. Network monitoring with zabbix. Mikkelin ammattikorkeakoulu, 2015.

Steve Jacob. **What is Multiprotocol Label Switching (MPLS) ??** 2018. Disponível em: <https://medium.com/@blogstevej327stuff/what-is-multiprotocol-label-switching-mpls-f9e9cc7fe43b>. Acesso em: 13 de maio 2022.

WANG, N. et al. An overview of routing optimization for internet traffic engineering. **IEEE Communications Surveys & Tutorials**, IEEE, v. 10, n. 1, p. 36–56, 2008.

Zabbix SIA. **User parameters**. 2022. Disponível em: <https://www.zabbix.com/documentation/current/en/manual/config/items/userparameters>. Acesso em: 22 de maio 2022.