



Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
Campus Campina Grande
Coordenação do Curso Superior de Tecnologia em Telemática

SEGMENT ROUTING: ROTEAMENTO DE TRÁFEGO BASEADO NA ORIGEM

JOSÉ SILVESTRE DA SILVA GALVÃO

Orientador: Dr. Marcelo Portela Sousa

Campina Grande, Dezembro de 2023

© José Silvestre da Silva Galvão



Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
Campus Campina Grande
Coordenação do Cursos Superior de Tecnologia em Telemática

SEGMENT ROUTING: ROTEAMENTO DE TRÁFEGO BASEADO NA ORIGEM

JOSÉ SILVESTRE DA SILVA GALVÃO

Monografia apresentada à Coordenação do Curso de Telemática do IFPB - Campus Campina Grande, como requisito parcial para conclusão do curso de Tecnologia em Telemática.

Orientador: Dr. Marcelo Portela Sousa

Campina Grande, Dezembro de 2023

G182s Galvão, José Silvestre da Silva

Segment routing: roteamento de tráfego baseado na origem / José Silvestre da Silva Galvão. - Campina Grande, 2023.

44f. : il.

Trabalho de Conclusão de Curso (Curso Superior de Tecnologia de Telemática) - Instituto Federal da Paraíba, 2023.

Orientador: Prof. Dr. Marcelo Portela Sousa

1. Telemática 2. Segment routing - Engenharia de tráfego I. Sousa, Marcelo Portela II. Título.

CDU 004.7

SEGMENT ROUTING: ROTEAMENTO DE TRÁFEGO BASEADO NA ORIGEM

JOSÉ SILVESTRE DA SILVA GALVÃO



DR. MARCELO PORTELA SOUSA

Orientador

Documento assinado digitalmente
 DAVID CANDEIA MEDEIROS MAIA
Data: 16/01/2024 13:43:04-0300
Verifique em <https://validar.iti.gov.br>

DR. DAVID CANDEIA MEDEIROS MAIA

Membro da Banca

Documento assinado digitalmente
 KATYUSCO DE FARIAS SANTOS
Data: 17/01/2024 14:01:34-0300
Verifique em <https://validar.iti.gov.br>

DR. KATYUSCO DE FARIAS SANTOS

Membro da Banca

Campina Grande, Paraíba, Brasil

Dezembro/2023

Este trabalho é dedicado a todos aqueles que, de alguma forma, contribuíram para a minha
jornada.

If you never bleed, you're never gonna grow.

Taylor Swift

Agradecimentos

Agradeço primeiramente à minha família, pelo constante apoio, incentivo e compreensão ao longo desta jornada acadêmica. Suas palavras de encorajamento, o amor e a compreensão por estar ausente em vários momentos, foram fundamentais para superar os desafios e alcançar este objetivo. Isso tudo é por vocês.

Aos meus amigos, que estiveram ao meu lado durante toda a trajetória acadêmica, compartilhando experiências, superando desafios e celebrando conquistas. A minha gratidão especial vai para Maria Helena e Ruan Miguel, que sempre estiveram dispostos a me ouvir, oferecer apoio e compartilhar valiosos conselhos.

Um agradecimento especial também aos amigos que conheci durante o projeto EDGE-5G, Jefferson Maxmiliano e Mikael Marinho. Sem a colaboração e ajuda de vocês, este trabalho não teria sido possível. Agradeço por compartilharem seus conhecimentos e contribuírem de maneira significativa para o desenvolvimento deste projeto.

Quero expressar minha gratidão aos professores que, ao longo do curso, dedicaram seu tempo e conhecimento para fornecer uma educação de qualidade. Seu comprometimento e paixão pelo ensino foram inspiradores e fundamentais para meu desenvolvimento acadêmico.

Um agradecimento especial ao professor Marcelo Portela, meu orientador, pela orientação excepcional, suporte e valiosas contribuições ao desenvolvimento deste trabalho. Sua expertise e dedicação foram essenciais para a condução deste projeto.

Aos professores Katysco Santos e David Candeia que compuseram a banca examinadora, agradeço pela disponibilidade em avaliar este trabalho e pelas valiosas sugestões que contribuíram para sua melhoria.

Em resumo, agradeço a todos que de alguma forma estiveram presentes em minha jornada acadêmica, tornando possível a conclusão deste trabalho. O aprendizado e as experiências compartilhadas foram inestimáveis, moldando não apenas meu conhecimento, mas também minha jornada como estudante.

Resumo

Este trabalho apresenta a implementação do *Segment Routing* (SR) em uma topologia de *backbone* de um provedor de serviços utilizando o simulador de redes EVE-NG. O SR é uma tecnologia que simplifica o uso do *Multiprotocol Label Switching* (MPLS), eliminando a necessidade de protocolos de distribuição de rótulos, como o LDP, e utiliza apenas o *Interior Gateway Protocol* (IGP) para atribuir e anunciar os *labels*. A metodologia incluiu a compreensão do funcionamento do SR, a montagem da topologia no EVE-NG, a configuração dos roteadores com um protocolo de roteamento dinâmico e a ativação do SR-MPLS nos roteadores da topologia. Os resultados mostraram diferenças de implementação entre o SR-BE e o SR-TE e evidencia o cenário de uso de cada modo do SR. O estudo contribui para a compreensão e aplicação prática do SR como uma abordagem que oferece maior controle e eficiência no tráfego em redes.

Palavras-chave: Engenharia de tráfego. SR-MPLS. SR-TE. SR-BE. *Segment Routing*.

Abstract

This work presents the implementation of Segment Routing (SR) in a backbone topology of a service provider using the EVE-NG network simulator. SR is a technology that simplifies the use of Multiprotocol Label Switching (MPLS), eliminating the need for label distribution protocols such as LDP, and relying solely on the Interior Gateway Protocol (IGP) to assign and advertise labels. The methodology included understanding the functioning of SR, setting up the topology in EVE-NG, configuring routers with a dynamic routing protocol, and activating SR-MPLS on the routers in the topology. The results showed implementation differences between SR-BE and SR-TE, highlighting the scenarios for each SR mode. The study contributes to the understanding and practical application of SR as an approach that provides greater control and efficiency in traffic management in networks.

Keywords: Traffic Engineering. SR-MPLS. SR-TE. SR-BE. Segment Routing.

Sumário

Lista de Abreviaturas	xi
Lista de Figuras	xii
Lista de Tabelas	xiii
1 Introdução	1
1.1 Objetivos	2
1.1.1 Objetivo Geral	2
1.1.2 Objetivos Específicos	2
2 Referencial Teórico	3
2.1 EVE-NG	3
2.2 Protocolos de Roteamento	3
2.2.1 <i>Open Shortest Path First</i> (OSPF)	4
2.3 <i>Multiprotocol Label Switching</i>	4
2.4 <i>Segment Routing</i>	5
3 Metodologia	9
3.1 Estudo de caso	9
3.1.1 Topologia SR-MPLS	10
3.1.2 OSPF	12
3.1.3 SR-MPLS	13
4 Resultados	15
4.1 <i>Segment Routing - Best Effort</i>	15
4.2 <i>Segment Routing - Traffic Engineering</i>	17
4.3 <i>Discussão</i>	21
5 Considerações Finais e Sugestões para Trabalhos Futuros	23
5.1 Sugestões para Trabalhos Futuros	23
A Configurações dos Roteadores	24

Referências Bibliográficas

Lista de Abreviaturas

AS	<i>Autonomous System</i>
ATM	<i>Asynchronous Transfer Mode</i>
CE	<i>Customer Edge</i>
EVE-NG	<i>Emulated Virtual Environment – Next Generation</i>
FRR	<i>Fast Reroute</i>
IETF	<i>Internet Engineering Task Force</i>
IGP	<i>Interior Gateway Protocol</i>
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
LDP	<i>Label Distribution Protocol</i>
LSA	<i>Link State Advertisement</i>
LSP	<i>Label Switched Path</i>
MPLS	<i>Multiprotocol label switching</i>
OAM	<i>Operations, Administration and Maintenance</i>
OSPF	<i>Open Shortest Path First</i>
P	<i>Provider</i>
PE	<i>Provider Edge</i>
QoS	<i>Quality of Service</i>
RSVP	<i>Resource Reservation Protocol</i>
SDN	<i>Software-Defined Network</i>
SID	<i>Segment Identifier</i>
SPF	<i>Shortest Path First</i>
SR	<i>Segment Routing</i>
SR-BE	<i>Segment Routing - Best Effort</i>
SR-MPLS	<i>Segment Routing - Multiprotocol label switching</i>
SR-TE	<i>Segment Routing - Traffic Engineering</i>
SRv6	<i>Segment Routing - IPv6</i>
TCC	<i>Trabalho de Conclusão de Curso</i>
TE	<i>Traffic Engineering</i>
VPN	<i>Virtual Private Network</i>

Lista de Figuras

2.1	Topologia do OSPF com diferentes áreas. FONTE: [ANDRÉ ORTEGA 2023]	4
2.2	Topologia que representa uma <i>Segment List</i> . FONTE: [Juniper 2023]	7
2.3	Topologia que representa as operações do SR. FONTE: [Ventre <i>et al.</i> 2021]	7
3.1	Topologia de <i>backbone</i> FONTE: [Lins, Oliveira e Mendonça]	10
3.2	Implementação FONTE: O autor (2023)	12
3.3	Topologia com as redes IPs e <i>Loopbacks</i> . FONTE: O autor (2023)	12
4.1	Informações do OSPF. FONTE: O autor (2023)	16
4.2	Caminho do pacote com o SR-BE. FONTE: O autor (2023)	16
4.3	Pacote do teste de <i>traceroute</i> no <i>Wireshark</i> . FONTE: O autor (2023)	17
4.4	Caminho do pacote com o SR-TE. FONTE: O autor (2023)	19
4.5	Pacote do teste de <i>ping</i> no <i>Wireshark</i> . FONTE: O autor (2023)	20
4.6	Pacotes ICMP <i>Request</i> no roteador R4. FONTE: O autor (2023)	21

Lista de Tabelas

3.1	Configuração de endereçamento no roteador de origem. FONTE: O autor (2023)	13
3.2	Configuração do OSPF ponto a ponto. FONTE: O autor (2023)	13
3.3	Configuração do SR-MPLS. FONTE: O autor (2023)	14
4.1	Criação da <i>Segment List</i> . FONTE: O autor (2023)	18
4.2	Aplicação da <i>Segment List</i> . FONTE: O autor (2023)	19
A.1	Configuração do <i>Segment Routing - Best Effort</i> . FONTE: O autor (2023)	24
A.2	Configuração do <i>Segment Routing - Traffic Engineering</i> . FONTE: O autor (2023)	25
A.3	Configuração completa do roteador R2. FONTE: O autor (2023)	26
A.4	Configuração completa do roteador R3. FONTE: O autor (2023)	27
A.5	Configuração completa do roteador R4. FONTE: O autor (2023)	28
A.6	Configuração completa do roteador R5. FONTE: O autor (2023)	29
A.7	Configuração completa do roteador R6. FONTE: O autor (2023)	30

Capítulo 1

Introdução

As redes de computadores têm se tornado cada vez mais presentes nas sociedades, o que permite a comunicação e o compartilhamento de recursos entre dispositivos em diferentes locais. Elas são compostas por diversos componentes interconectados, tais como computadores, servidores, *switches*, roteadores, entre outros [Tanenbaum 2003]. Esses componentes são responsáveis por garantir o correto fluxo de dados entre os dispositivos conectados à rede.

Com a proliferação de dispositivos conectados à *Internet*, que variam de *smartphones* a dispositivos de *Internet* das Coisas (do inglês, *Internet of Things* – IoT), as redes enfrentam novos desafios: acomodar e gerenciar um volume cada vez maior de tráfego de dados e serviços. Esse aumento na demanda por conectividade tem incrementado a complexidade das redes e a necessidade de soluções que sejam não apenas escaláveis, mas também eficientes em termos de utilização de recursos. Nesse contexto dinâmico e desafiador, o *Segment Routing* (SR) surge para simplificar o uso do *Multiprotocol label switching* (MPLS), eliminando a necessidade de protocolos de distribuição de rótulos, como o LDP (do inglês, *Label Distribution Protocol*) e RSVP (do inglês, *Resource Reservation Protocol*) e utiliza apenas o IGP (do inglês, *Interior Gateway Protocol*) para atribuir e anunciar os *labels*.

Esta pesquisa tem como foco tratar um tema pouco abordado na academia e implementar o *Segment Routing* com foco no SR-MPLS (do inglês, *Segment Routing - Multiprotocol Label Switching*), que explora a questão fundamental de como o *Segment Routing* pode ser utilizado para encaminhamento de tráfego baseado na origem. Esta pesquisa também contribui para a compreensão e aplicação prática do SR como uma abordagem que oferece maior controle e eficiência no tráfego em redes.

Como parte da metodologia deste estudo, a implementação do SR-MPLS foi realizada no ambiente de simulação de redes EVE-NG e roteadores Cisco. Essa abordagem permite avaliar o desempenho do *Segment Routing* em um ambiente controlado. Com os resultados obtidos é possível demonstrar a viabilidade e os benefícios do SR, com ênfase na otimização do encaminhamento com base na origem. Além disso, esta pesquisa contribui para o avanço do conhecimento na área de redes de computadores e oferece perspectivas práticas para a implementação do SR.

Este Trabalho de Conclusão de Curso (TCC) está estruturado da seguinte maneira. O Capítulo 2 descreve os principais conceitos necessários ao entendimento desta pesquisa. O Capítulo 3 detalha a metodologia utilizada para condução do TCC. O Capítulo 4 discute os resultados gerados a partir da implementação. Por fim, o Capítulo 5 apresenta as considerações finais e perspectivas de trabalhos futuros.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo principal deste trabalho é implementar o *Segment Routing* em uma topologia de um *backbone* de um provedor de serviços utilizando o simulador de redes EVE-NG.

1.1.2 Objetivos Específicos

- Entender o funcionamento do *Segment Routing*.
- Construir uma topologia de *backbone* no EVE-NG.
- Configurar os roteadores da topologia.
- Analisar os pacotes enviados e recebidos.

Capítulo 2

Referencial Teórico

Neste capítulo, os conceitos fundamentais sobre EVE-NG, protocolos de roteamento, *Multiprotocol Label Switching* e *Segment Routing*, são abordados.

2.1 EVE-NG

O *Emulated Virtual Environment – Next Generation* (EVE-NG) é um emulador de rede *multivendor* que possibilita a criação de cenários de redes de maneira virtualizada. Ele disponibiliza ferramentas para o uso na interconexão de dispositivos, sejam eles virtuais ou físicos [EVE-NG 2023]. Essa ferramenta usa o mesmo princípio de softwares como o *packet tracer*, *HP Network Simulator* e *eNSP*, que simulam equipamentos como roteadores e *switches* de fabricantes específicos [Oliveira 2020].

A criação dos laboratórios é realizada por meio de uma interface gráfica, com acesso *web*. Os usuários podem conectar e configurar equipamentos de diferentes fabricantes, que podem ser encontrados no *site* da marca ou em bibliotecas de modelos. Além de dispositivos de rede, é possível executar sistemas operacionais (SO) *Linux* e *Windows* [EVE-NG 2023]. No *site* do EVE-NG há uma biblioteca com várias distribuições Linux compatíveis para instalação.

2.2 Protocolos de Roteamento

Para que os dados de um dispositivo sejam transmitidos entre redes distintas, o processo de roteamento é essencial [Tanenbaum 2003]. O roteamento envolve a tomada de decisão ao encaminhar um pacote, com a seleção das rotas mais adequadas para permitir a comunicação entre um ou mais nós da rede. Essa seleção é geralmente realizada por meio de um protocolo de roteamento por meio de um algoritmo, o que resulta na criação de tabelas específicas nos dispositivos de rede. Para manter essas informações atualizadas, protocolos dinâmicos são utilizados [Baker 1995].

2.2.1 Open Shortest Path First (OSPF)

O OSPF (do inglês, *Open Shortest Path First*) é um protocolo dinâmico desenvolvido pelo IETF (do inglês, *Internet Engineering Task Force*). Esse protocolo é classificado como um IGP, para a distribuição de rotas entre elementos de camada de rede que pertencem ao mesmo AS (do inglês, *Autonomous System*). Ele é baseado no algoritmo SPF (do inglês, *Shortest Path First*), que calcula as rotas com base no estado de *link* de cada nó durante o caminho [Moy 1998].

O OSPF também é caracterizado pela rápida detecção de mudança topológica na rede, como falha em interface, e recalcula uma nova rota, sem a ocorrência de *loop* após convergência rápida [Moy 1998]. Isso é possível, pois o funcionamento é baseado no fato de que cada roteador mantém um banco de dados dos estados de enlace que participam do OSPF. Esses estados se referem às interfaces de rede ativas dos vizinhos próximos, em que os estados são armazenados a partir do envio e recebimentos de mensagem *Link State Advertisement* (LSA) [HUAWEI 2023].

Uma característica do OSPF é possibilitar a divisão da rede em diferentes áreas, como representado na Figura 2.1. Os roteadores que estão dentro de uma determinada área, são denominados de roteadores internos, e cada AS terá o seu *backbone* (área 0) a qual as outras áreas estão conectadas. O *backbone* é responsável por distribuir informações de roteamento entre áreas a que estão conectadas [Moy 1998]. A vantagem da criação de diferentes áreas é a possível redução do tempo de convergência para aprender ou mudar rotas e a redução de troca de mensagens LSA.

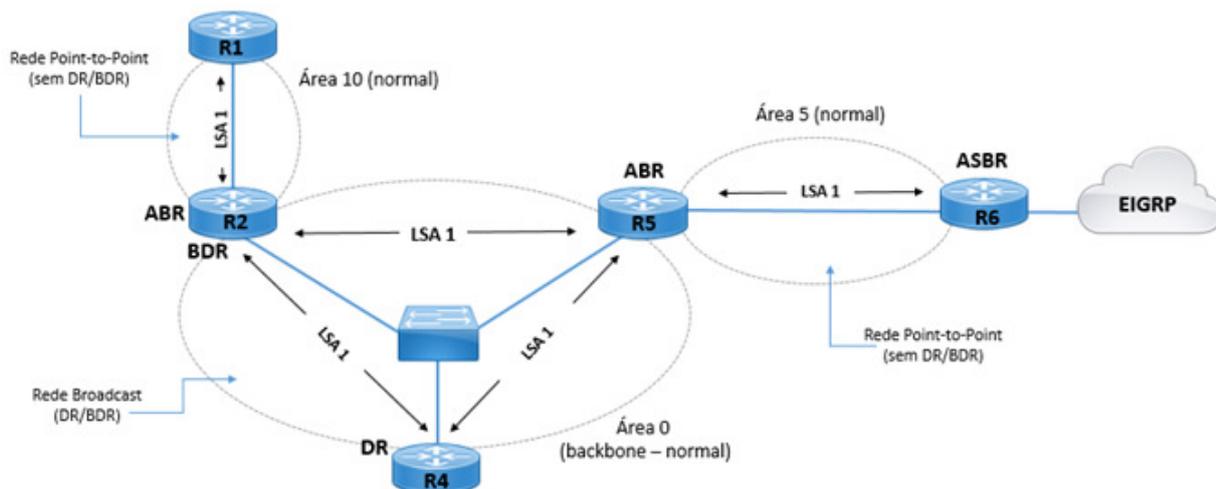


Figura 2.1: Topologia do OSPF com diferentes áreas. FONTE: [ANDRÉ ORTEGA 2023]

2.3 Multiprotocol Label Switching

O MPLS (do inglês, *Multiprotocol Label Switching*) é um protocolo de rede que utiliza etiquetas (*labels*) para encaminhar pacotes de dados. Ao contrário de outros protocolos,

como o IP (do inglês, *Internet Protocol*), o MPLS permite que o roteamento seja feito com base em caminhos pré-definidos, em vez de ser determinado a cada nó da rede [Viswanathan, Rosen e Callon 2001]. Isso torna o MPLS uma solução escalável e eficiente para redes de grande porte.

O funcionamento do MPLS é baseado na criação de caminhos pré-definidos, chamados de LSPs (*Label Switched Paths*), que são criados pelos roteadores da rede. Cada LSP é identificado por uma *label* única, que é adicionada ao cabeçalho dos pacotes de dados. Quando um pacote de dados é enviado pela rede, os roteadores utilizam a *label* para determinar o caminho que o pacote deve seguir.

Uma das principais vantagens do MPLS é a sua capacidade de oferecer diferentes níveis de serviço QoS (do inglês, *Quality of Service*) para diferentes tipos de tráfego [Porwal, Yadav e Charhate 2008]. Os roteadores da rede podem definir diferentes prioridades para os pacotes, de acordo com critérios como a largura de banda, a disponibilidade de recursos e a sensibilidade do tráfego. Isso permite que o MPLS seja utilizado em redes corporativas e de provedores de serviços de *Internet*, garantindo um alto grau de qualidade de serviço [Viswanathan, Rosen e Callon 2001].

Outra vantagem do MPLS é a sua integração com a TE (do inglês, *Traffic Engineering*), que permite que os administradores de rede criem rotas personalizadas para o tráfego de dados. A TE é realizada por meio da criação de políticas de roteamento, que permitem que os roteadores definam caminhos específicos para diferentes tipos de tráfego. Isso possibilita uma maior eficiência na utilização da largura de banda e na priorização do tráfego.

Além disso, o MPLS é compatível com outros protocolos de rede, como o IP e o ATM (do inglês, *Asynchronous Transfer Mode*). Ele também pode ser utilizado para implementar serviços avançados, como a VPN (do inglês, *Virtual Private Network*), que proporciona um alto grau de segurança e privacidade para as comunicações empresariais [Tanenbaum 2003].

2.4 Segment Routing

O *Segment Routing* (SR) é um protocolo de roteamento de pacotes em redes IP que permite definir caminhos personalizados na rede. Ele é introduzido com a proposta de tornar a engenharia de tráfego mais eficiente, simplificando as operações de plano de controle [Moreno, Beghelli e Cugini 2017].

O SR funciona com a aplicação de pilhas de identificadores de segmento (*Segment Identifier – SID*), em que cada segmento representa um nó ou um conjunto de nós da rede. A informação do segmento é anexada ao cabeçalho do pacote e o encaminhamento é realizado de forma eficiente [Filsfils *et al.* 2018].

O SR tem várias vantagens em relação a outras técnicas de encaminhamento, como a flexibilidade na definição de caminhos, a escalabilidade, a simplicidade da arquitetura da rede e a possibilidade de introduzir novas funcionalidades sem a necessidade de mudanças

no protocolo de roteamento.

O roteamento por segmentos utiliza um IGP para distribuir as marcações e computar os caminhos sem a necessidade de alterar a arquitetura MPLS já existente, podendo assim, coexistir com o MPLS [Filsfils *et al.* 2018]. A RFC-8402 ainda destaca que a implementação da arquitetura SR pode ser feita tanto no MPLS (SR-MPLS), como no protocolo IPv6 (SRv6).

O SR-MPLS faz com que protocolos como o LDP e RSVP não sejam mais necessários. Esses protocolos causam alguns problemas na rede MPLS. Como o LDP depende do IGP, mas eles são independentes um do outro, pode acontecer a dessincronização entre o LDP e o IGP, o que pode causar o bloqueio do tráfego na rede. Outra desvantagem é o uso do RSVP-TE que tem uma implementação e manutenção complexa e também envolve estados de protocolo complexos e exige que os dispositivos troquem muitos pacotes para manter os estados do túnel, dificultando a implantação em larga escala [Liu Yan 2023]. Como proposta, o SR-MPLS soluciona essas dificuldades encontradas no MPLS, utilizando extensões do IGP para fazer distribuição e sincronização dos *labels* ao invés de utilizar os protocolos LDP e RSVP-TE.

Independente do plano de dados, o SR funciona com o mesmo princípio. Para isso, alguns conceitos são abordados:

SR *policy*: A SR *policy* é uma lista ordenada de segmentos. A lista de segmentos pode ser especificada de forma explícita no SR-MPLS usando uma pilha de *labels*, por exemplo. A política SR pode ser usada para diferentes fins: TE; *Operations, Administration and Maintenance* (OAM), ou para o *Fast Reroute* (FRR) que é a forma de refazer a rota caso alguma falha aconteça no caminho [Filsfils *et al.* 2018].

Operação PUSH: A operação consiste na inserção em uma lista de segmentos. Em redes SR-MPLS pode ser a inserção do *label* na pilha e em redes SRv6 pode significar a inserção de SID na primeira posição e redirecionar o ponteiro ao topo da lista [Filsfils *et al.* 2018].

Operação NEXT: Quando uma lista de segmentos está completa, o NEXT ativa um próximo segmento da lista. No SR-MPLS, NEXT significa remover o *label* do topo da pilha. Já no SRv6, significa incrementar o ponteiro [Filsfils *et al.* 2018].

Operação CONTINUE: É a continuação do segmento atual que está ativo, mesmo que esteja incompleto. Em SR-MPLS é implementado como um **SWAP** (uma operação de encaminhamento que consiste em buscar um *label* de entrada para determinar o *label* de saída [Viswanathan, Rosen e Callon 2001]). Em redes SRv6, seria a ação de encaminhamento de um pacote IPv6 de acordo com o endereço de destino [Filsfils *et al.* 2018].

Nas Figuras 2.2 e 2.3 duas topologias com os principais conceitos são ilustradas. É possível notar que na Figura 2.2 há uma *Segment List*, que deixa o roteador de origem (*Ingress*), passa pelos nós associados à política aplicada e ingressa ao seu destino (*Egress*). A lista de segmentos é inserida no pacote do roteador de origem [Ventre *et al.* 2021]. Na Figura 2.3, as operações PUSH, NEXT e CONTINUE são representadas. A operação PUSH no *headend*

node indica que a *Segment List* foi inserida. Na operação NEXT indica que o *label* do topo foi retirado e logo em seguida a operação CONTINUE é executada para continuar o caminho do pacote sem adicionar novos *labels*.

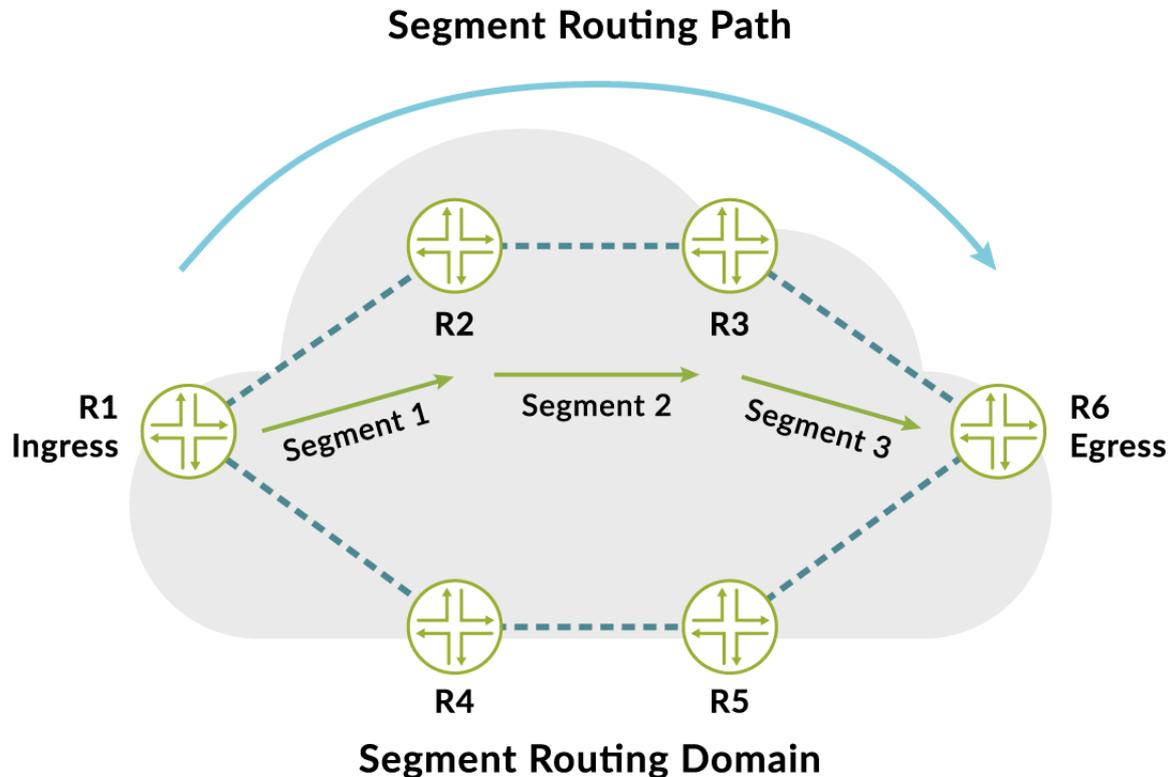


Figura 2.2: Topologia que representa uma *Segment List*. FONTE: [Juniper 2023]

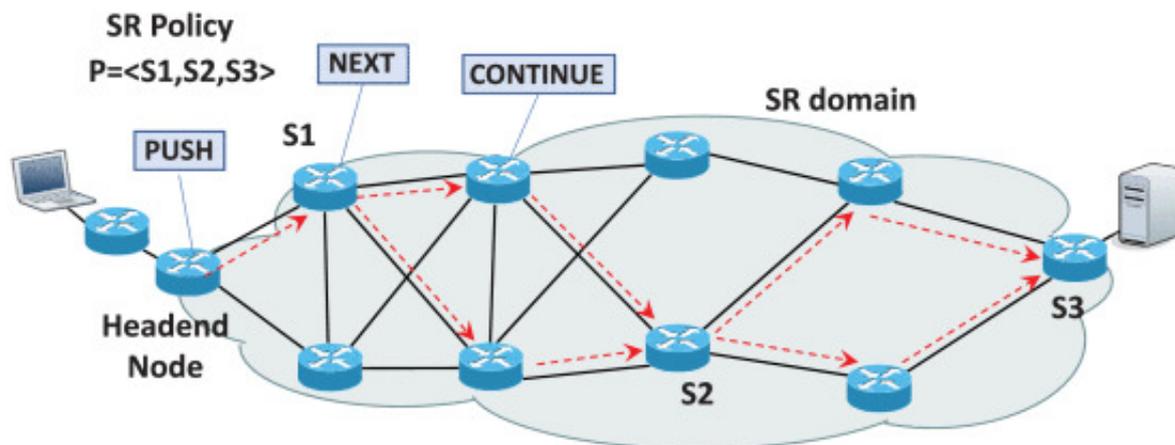


Figura 2.3: Topologia que representa as operações do SR. FONTE: [Ventre et al. 2021]

SDN (do inglês, *Software-Defined Network*), é uma arquitetura emergente que é dinâmica, gerenciável, econômica e adaptável, tornando ela ideal para as novas demandas do mercado. SDN pode ser resumida a uma frase: Uma rede na qual o plano de controle está fisicamente separado do plano de encaminhamento e um único plano de controle controla

vários dispositivos de encaminhamento [McKeown 2009]. A arquitetura do SDN separa as funções de plano de controle e plano de dados; com isso, a rede se torna programável, ágil e pode ser gerenciada de forma centralizada com o uso de controladoras SDN [Feng *et al.* 2022]. As controladoras mantêm uma visão global da rede e, com isso, podem gerenciar de um único ponto todos os dispositivos da rede. A integração do SR com SDN representa uma abordagem sinérgica que pode potencializar as vantagens do roteamento de tráfego baseado na origem.

Capítulo 3

Metodologia

Este estudo tem como objetivo a implementação do SR-MPLS como técnica de roteamento baseado na origem. Para isso, foi usado o simulador de redes EVE-NG, como ferramenta principal para a implementação.

Um ponto importante para a simulação no EVE-NG, é definir o modelo dos dispositivos. Para isso, foi feita uma pesquisa de dispositivos compatíveis com a tecnologia e que tem suas imagens disponíveis para uso. Os dispositivos escolhidos para essa pesquisa, foram os roteadores virtuais da Cisco, o XRV 9000 e o XRV K9.

Com essas escolhas, a implementação de uma topologia foi efetuada, na qual a configuração foi executada e testada. Para os testes, utilizou-se a integração do EVE-NG com o *sniffer* de rede *Wireshark*, para comprovar que os segmentos foram devidamente encaminhados pelo cenário e os pacotes estejam seguindo o caminho configurado. Além dos testes com o *Wireshark* foram realizados testes com o *ping* e o *traceroute*.

O *hardware* utilizado para instalação do emulador e das imagens dos dispositivos de rede é um computador *desktop* da sala de capacitação do laboratório Assert no Instituto Federal da Paraíba *Campus* Campina Grande. O computador tem um processador *Intel Core i7-6700* de 3.40 GHz, 32 GB de memória RAM e sistema operacional *Windows 10* de 64 *bits*.

3.1 Estudo de caso

Nesta seção é apresentado um estudo de caso sobre o atual funcionamento do *backbone* dos provedores de serviços. O cenário representado pela Figura 3.1 é baseada em uma topologia do livro "Redes MPLS fundamentos e aplicações" [Lins, Oliveira e Mendonça].

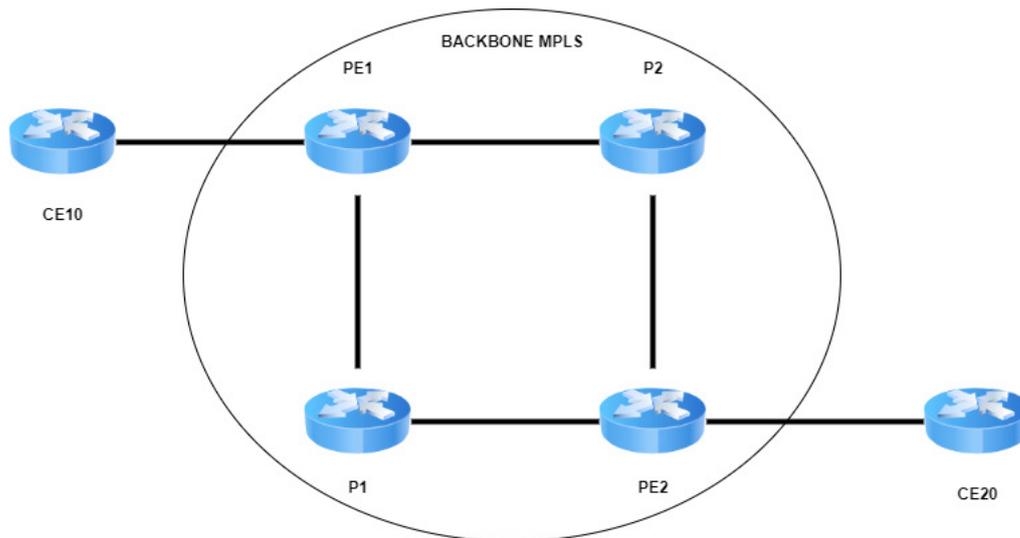


Figura 3.1: Topologia de backbone FONTE: [Lins, Oliveira e Mendonça]

Os roteadores estão nomeados de acordo com a arquitetura MPLS:

- CE (do inglês, *Customer Edge*) está localizado na borda da rede do cliente. Ele é responsável pela comunicação direta com os dispositivos finais dos clientes;
- PE (do inglês, *Provider Edge*) estão localizados na borda da rede dos provedores de serviços. A sua função é como ponto de transição entre o CE e a rede do provedor;
- P (do inglês, *Provider*) são responsáveis pelo encaminhamento de pacotes com base nos rótulos MPLS [Huawei 2023].

Nesse cenário, inicialmente são utilizadas algumas configurações e protocolos para o roteamento entre os vários nós da rede. A configuração e protocolos utilizados em uma arquitetura MPLS são: o próprio MPLS, LDP (do inglês, *Label Distribution Protocol*) para distribuição das *labels* MPLS, OSPF ou IS-IS (do inglês, *Intermediate System to Intermediate System*) como IGPs e configurações de endereçamento IP [Kaur e Kumar 2010].

A implementação do SR é realizada com base nas configurações iniciais citadas nessa seção.

3.1.1 Topologia SR-MPLS

Os dispositivos utilizados no EVE-NG tem requisitos mínimos para o funcionamento. O XRV 9000, necessita de no mínimo 16 GB de vRAM para o bom funcionamento, enquanto o XRV K9 necessita de no mínimo 4 GB [EVE-NG 2023]. Tendo em vista a quantidade de recursos que o XRV 9000 consome, ele foi utilizado como o nó de origem, já que é nele que serão feitas as configurações de caminhos do SR. Os demais roteadores são modelos XRV K9.

Um dos motivos para a escolha do XRV 9000 para o roteador de origem (R1), é o suporte a configuração de rotas do SR, que pode aplicar preferências de encaminhamento

e fazer a execução das políticas baseadas na *Segment List* especificada no nó. Os demais dispositivos são do modelo XRV K9 que possuem suporte ao SR, mas de forma limitada, o que impossibilita a criação de políticas e *Segment Lists*.

Na Figura 3.2 tem-se a implementação no EVE-NG da topologia baseada na Seção 3.1. A implementação do SR-MPLS foi feita a partir do uso do IGP OSPF em que o caminho mais curto é preferível para o tráfego dos pacotes. O caminho que os pacotes seguem quando deixam a origem (R1) é: R1 -> R2 -> R3 -> R6 (destino). Nesse cenário a rota é determinada de maneira dinâmica e eficiente pelo SR-BE (do inglês, *Segment Routing - Best Effort*), seguindo os caminhos disponíveis na rede sem a necessidade de otimizações específicas. Isso proporciona simplicidade na operação, menor *overhead*¹ de configuração e manutenção, especialmente em ambientes onde a rede é relativamente simples ou os requisitos de tráfego são mais flexíveis. Ao aplicar o SR-TE (do inglês, *Segment Routing - Traffic Engineering*), tem-se o objetivo de alcançar o destino passando pelos roteadores R4 e R5, com o caminho explícito: R1 -> R2 -> R4 -> R5 -> R3 -> R6, como na Figura 3.2. Apesar do caminho explícito ser mais longo e parecer não trazer benefícios, essa rota pode trazer mais otimização da rede em alguns casos: quando há variação no link entre R2 e R3 por congestionamento, por exemplo, o uso do SR-TE pode ser justificado. Caso o pacote enviado seja um serviço que tenha algum tipo de requisito rigoroso de latência a rota pode ser utilizada para otimizar o caminho e direcionar o tráfego para links que ofereçam menor atraso dos pacotes.

O SR também se beneficia do SDN, uma vez que há um gerenciamento centralizado que proporciona uma visão geral da rede. Isso favorece o tratamento, manipulação, controle e configuração da rede, permitindo que, a partir de um único ponto de origem, sejam definidos os melhores caminhos para o tráfego. Com isso, observa-se uma melhoria na performance e gerenciabilidade da rede.

¹Refere-se ao custo adicional ou à carga adicional associada a uma operação, processo ou protocolo. O *overhead* representa recursos extras, como tempo, largura de banda, processamento ou armazenamento, que são consumidos além do necessário para realizar uma tarefa específica.

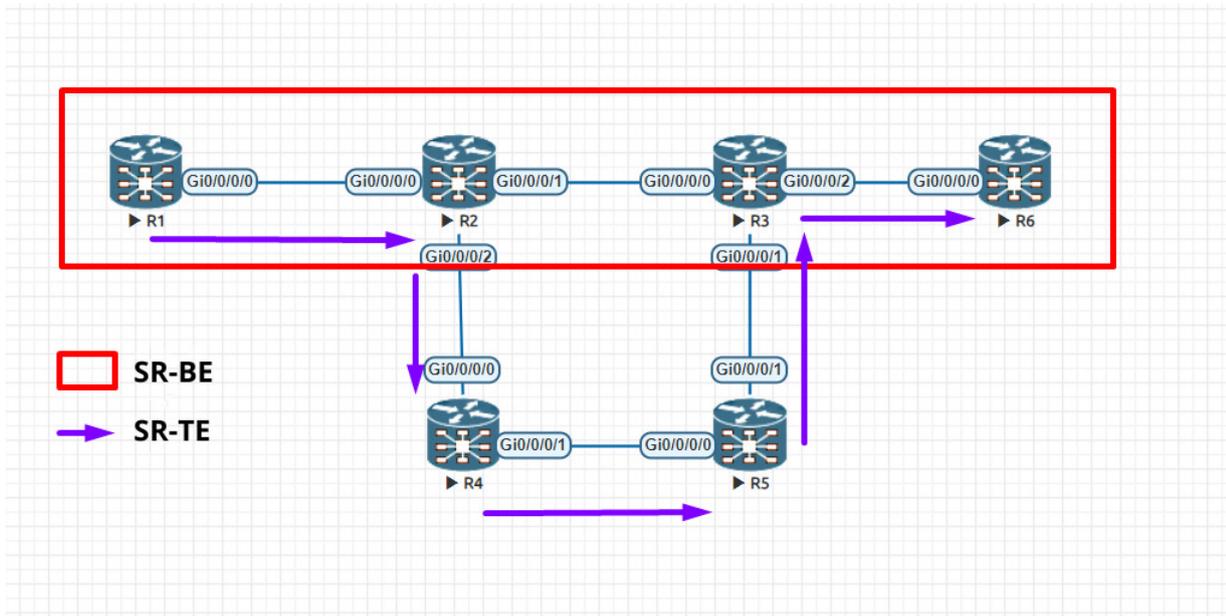


Figura 3.2: Implementação FONTE: O autor (2023)

3.1.2 OSPF

Para habilitar o OSPF nos roteadores, é necessário, adicionar endereço IP nas interfaces dos roteadores, incluindo a interface de *loopback*. Para isso, foram definidos os IPs de cada rede e também das *loopbacks*.

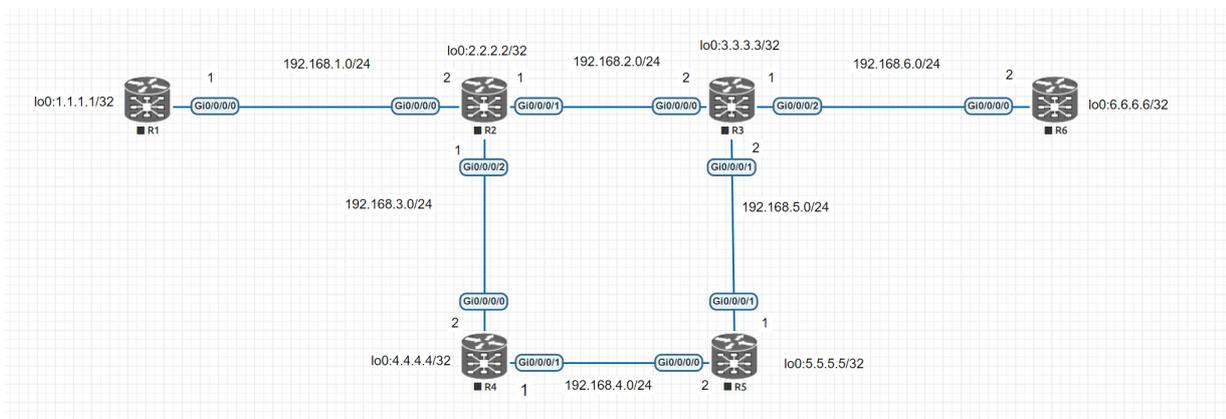


Figura 3.3: Topologia com as redes IPs e Loopbacks. FONTE: O autor (2023)

A seguir há um exemplo de como foi feita a configuração de IP na interface Gi0/0/0/0 de R1, bem como, a configuração e ativação da interface *loopback* 0.

```
RP/0/RP0/CPU0:R1(config)#interface g0/0/0/0
RP/0/RP0/CPU0:R1(config-if)#ip addr 192.168.1.1 255.255.255.0
RP/0/RP0/CPU0:R1(config-if)#no shutdown
RP/0/RP0/CPU0:R1(config-if)#exit
RP/0/RP0/CPU0:R1(config)#interface loopback 0
RP/0/RP0/CPU0:R1(config-if)#ip addr 1.1.1.1 255.255.255.255
RP/0/RP0/CPU0:R1(config-if)#commit
```

Figura 3.1: Configuração de endereçamento no roteador de origem. FONTE: O autor (2023)

Após a configuração de todas as interfaces, a configuração do OSPF foi realizada. A Figura 3.2 apresenta a configuração de alguns parâmetros necessários para a operação do OSPF, entre eles, a identificação do processo OSPF (`ospf 1`), a identificação do roteador (`router-id 1.1.1.1`) e a criação da área *backbone* (`area 0`). Após a criação do processo e da área, é possível adicionar as interfaces à área utilizando o comando *interface g0/0/0/0* dentro da configuração da área OSPF. Ao adicionar a interface é aplicado o comando *network point-to-point* para garantir uma comunicação OSPF ponto-a-ponto.

```
RP/0/RP0/CPU0:R1(config)#router ospf 1
RP/0/RP0/CPU0:R1(config-ospf)#router-id 1.1.1.1
RP/0/RP0/CPU0:R1(config-ospf)#area 0
RP/0/RP0/CPU0:R1(config-ospf-ar)#interface loopback 0
RP/0/RP0/CPU0:R1(config-ospf-ar-if)#exit
RP/0/RP0/CPU0:R1(config-ospf-ar)#interface g0/0/0/0
RP/0/RP0/CPU0:R1(config-ospf-ar-if)#network point-to-point
RP/0/RP0/CPU0:R1(config-ospf-ar-if)#commit
```

Figura 3.2: Configuração do OSPF ponto a ponto. FONTE: O autor (2023)

3.1.3 SR-MPLS

A configuração do SR-MPLS foi realizada em todos os roteadores. Na configuração do processo 1 do OSPF, o SR e o recurso de encaminhamento por *labels* MPLS foram habilitados. Na área 0 do mesmo processo, na interface de *loopback* foi adicionado a SID do roteador. O *prefix-SID* tem um intervalo que vai de 16000 a 1048575 e ele deve ser único em toda a rede [Cisco 2023]. Na Figura 3.3 há a configuração feita no roteador R1 e que segue a mesma base para configurar os demais roteadores.

```
RP/0/RP0/CPU0:R1(config)#router ospf 1
RP/0/RP0/CPU0:R1(config-ospf)#segment-routing mpls
RP/0/RP0/CPU0:R1(config-ospf)#segment-routing forwarding mpls
RP/0/RP0/CPU0:R1(config-ospf)#area 0
RP/0/RP0/CPU0:R1(config-ospf-ar)#interface loopback 0
RP/0/RP0/CPU0:R1(config-ospf-ar-if)#prefix-sid absolute 16001
RP/0/RP0/CPU0:R1(config-ospf-ar-if)#commit
```

Figura 3.3: *Configuração do SR-MPLS. FONTE: O autor (2023)*

Após concluir as configurações de OSPF e SR-MPLS é possível analisar os resultados da implementação e também fazer as configurações de algumas variações do SR-MPLS.

Capítulo 4

Resultados

4.1 *Segment Routing - Best Effort*

Após a conclusão das configurações de OSPF e SR-MPLS, o SR já está em funcionamento no seu modo SR-BE. O SR-BE é o modo mais puro do SR-MPLS, em que o OSPF calcula a rota mais curta para o pacote seguir [Huawei 2023]. O SR-BE é adequado para redes em que o tráfego não requer um controle rigoroso de caminho ou para redes que não sejam sensíveis a requisitos rigorosos de latência. O SR-BE também pode ser preferível em redes que tenham uma alta demanda por escalabilidade. Em uma rede de escala massiva, por exemplo, a topologia pode mudar dinamicamente devido a adições de novos pontos de presença ou reconfigurações. O SR-BE permite uma adaptação mais simples a essas mudanças sem a necessidade de reconfigurações complexas.

O resultado da implementação pode ser observado na Figura 4.1. É possível notar algumas informações importantes do OSPF e informações pertinentes ao SR, como a área e processo OSPF, como também a identificação atribuída ao roteador, que no caso do roteador R1 é 1.1.1.1. É possível de ver o *index SID* do roteador, que nesse caso é 1, esse número está relacionando a configuração do *prefix-SID*, feita na Seção 3.1.3.

```

R1
RP/0/RP0/CPU0:R1#show ospf database opaque-area 7.0.0.1 self-originate
Wed Dec  6 11:30:04.677 UTC

      OSPF Router with ID (1.1.1.1) (Process ID 1)

      Type-10 Opaque Link Area Link States (Area 0)

LS age: 588
Options: (No TOS-capability, DC)
LS Type: Opaque Area Link
Link State ID: 7.0.0.1
Opaque Type: 7
Opaque ID: 1
Advertising Router: 1.1.1.1
LS Seq Number: 80000011
Checksum: 0xc5c0
Length: 44

  Extended Prefix TLV: Length: 20
    Route-type: 1
    AF       : 0
    Flags    : 0x40
    Prefix   : 1.1.1.1/32

  SID sub-TLV: Length: 8
    Flags    : 0x0
    MTID     : 0
    Algo     : 0
    SID Index: 1
RP/0/RP0/CPU0:R1#

```

Figura 4.1: Informações do OSPF. FONTE: O autor (2023)

Ao fazer o comando *traceroute* com o destino para a interface de *loopback* de R6, pode-se notar que o SR-BE tem preferência pelo caminho mais curto por usar o OSPF como meio de cálculo de rotas. Então o caminho mais provável que o pacote percorre nesse cenário é saindo do R1 passando por R2, R3 e ingressando no R6, chegando ao seu destino final. Na Figura 4.2 ainda é possível de ser observado a informação do *label* de destino, no caso do R6, 16006.

```

R1
RP/0/RP0/CPU0:R1#traceroute 6.6.6.6
Wed Dec  6 11:54:08.513 UTC

Type escape sequence to abort.
Tracing the route to 6.6.6.6

 1  192.168.1.2 [MPLS: Label 16006 Exp 0] 91 msec  87 msec  73 msec
 2  192.168.2.2 [MPLS: Label 16006 Exp 0] 88 msec  89 msec  68 msec
 3  192.168.6.2 91 msec  * 69 msec
RP/0/RP0/CPU0:R1#

```

Figura 4.2: Caminho do pacote com o SR-BE. FONTE: O autor (2023)

Ao capturar os pacotes do *traceroute* com o *Wireshark* na interface *GigabitEthernet0/0/0/0* no roteador R1 com destino R6, é possível observar alguns parâmetros.

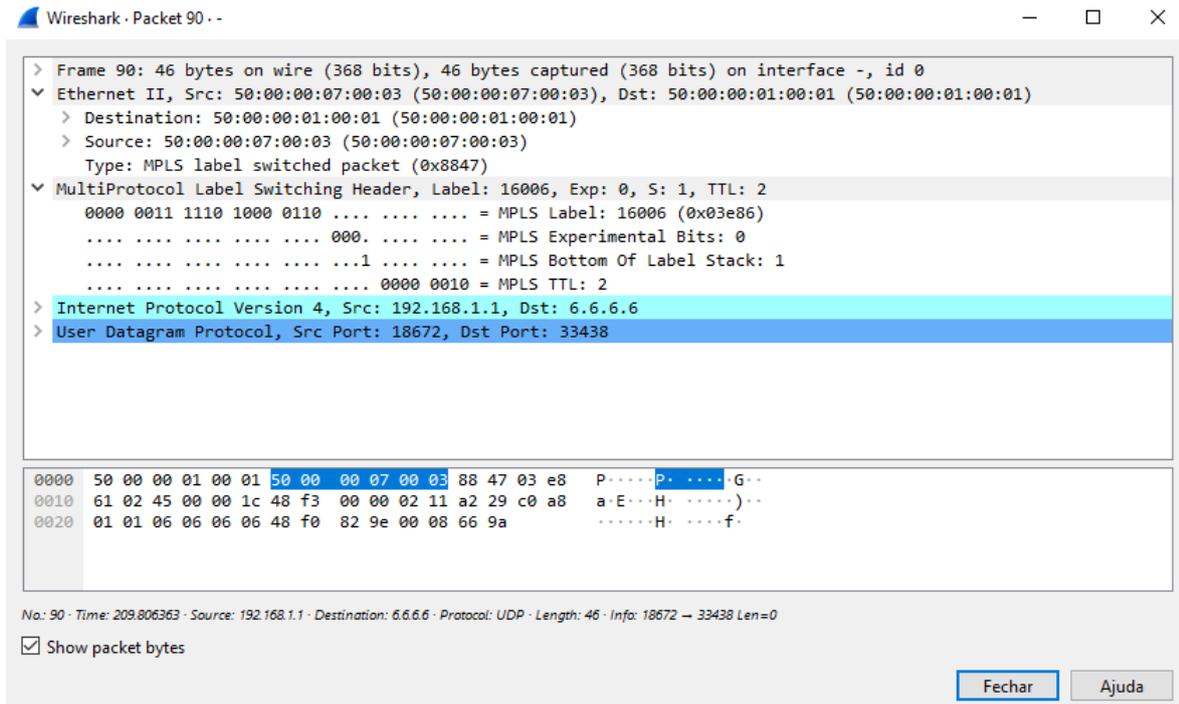


Figura 4.3: Pacote do teste de traceroute no Wireshark. FONTE: O autor (2023)

Na Figura 4.3 é notório o campo *Multiprotocol Label Switching Header* entre a camada *Ethernet II* e a camada de rede, sendo possível de comprovar que o MPLS é um protocolo que trabalha entre as duas camadas [Lins, Oliveira e Mendonça]. Nesse campo do MPLS também é possível de observar o *label* de destino e o parâmetro *MPLS Bottom Of Labels Stack* que está relacionado ao label de destino, se o resultado for 1 esse será o último nó, caso o resultado seja 0, ainda há mais nós que o pacote deve percorrer para chegar ao destino. Outras informações como IP de origem, IP de destino, MAC de origem e MAC de destino podem ser observados no bloco da camada 2 e camada 3.

4.2 Segment Routing - Traffic Engineering

Além do SR-BE, há a possibilidade da criação de rotas explícitas utilizando a engenharia de tráfego. O SR-TE é apropriado em cenários onde é necessário um controle mais preciso sobre a engenharia de tráfego, como a otimização de caminhos para cumprir requisitos específicos, como latência. Em redes que suportam várias classes de serviços, onde diferentes fluxos de tráfego podem ter requisitos distintos, o SR-TE pode ser usado para alocar recursos de forma mais eficiente e garantir que cada classe de serviço atenda às suas necessidades. Por exemplo, uma rede opera com diferentes tipos de serviços, VoIP (do inglês, *Voice over IP*), tráfego de vídeo em tempo real e transferência de dados de alta prioridade. Cada um desses serviços tem requisitos específicos em termos de largura de banda, latência e confiabilidade. O serviço de VoIP, por exemplo, requer uma baixa latência para garantir uma comunicação clara sem atrasos, com isso utilizando o SR-TE o administrador de rede consegue otimizar

o tráfego criando uma rota onde a latência do tráfego é menor [Filsfils *et al.* 2016].

A criação de políticas de encaminhamento deve ser feita no roteador de origem (R1). No modo de configuração global é habilitado o *Segment Routing* e a engenharia de tráfego. Em seguida é criada a lista de segmentos, que é uma lista formada de *labels* (*prefix-SID* configurados anteriormente) ou endereços IPv4 [Cisco 2018]. Para as configurações é essencial a criação de um nome para a lista e o caminho que o pacote deve seguir até chegar ao destino. Na Figura 4.1 está a configuração feita para a criação da lista. O nome da escolhido para a lista foi lista-1 e o caminho que o pacote deve seguir para ingressar no roteador R6 é passado pelos roteadores R2, R4, R5, R3 e ingressar no R6, como especificado na Figura 3.2

```
RP/0/RP0/CPU0:R1(config)#segment-routing
RP/0/RP0/CPU0:R1(config-sr)#traffic-eng
RP/0/RP0/CPU0:R1(config-sr-te)#segment-list lista-1
RP/0/RP0/CPU0:R1(config-sr-te-sl)#index 10 mpls label 16002
RP/0/RP0/CPU0:R1(config-sr-te-sl)#index 20 mpls label 16004
RP/0/RP0/CPU0:R1(config-sr-te-sl)#index 30 mpls label 16005
RP/0/RP0/CPU0:R1(config-sr-te-sl)#index 40 mpls label 16003
RP/0/RP0/CPU0:R1(config-sr-te-sl)#index 50 mpls label 16006
RP/0/RP0/CPU0:R1(config-sr-te-sl)#commit
```

Figura 4.1: Criação da Segment List. FONTE: O autor (2023)

Após a criação da lista, é necessário criar as políticas necessárias para que ela seja aplicada ao encaminhar os dados. A política, assim como a lista de segmentos, deve ter um nome. Alguns parâmetros devem ser configurados como a *color* que é um número entre 1 e 4294967295 e cada número pode ser usado para entrega de serviços específicos, nesse mesmo comando deve ser definido o último nó que a política vai ser aplicada, nesse caso o R6 (representado pelo seu IP de *loopback* 6.6.6.6). O comando *include all* faz com que qualquer tipo de *index* configurado anteriormente seja aceito, podendo ser tanto IPv4 quanto *label*. Deve ser especificado o tipo de encaminhamento que será usado (*candidate-path*), a preferência que aquela política vai ser aplicada, que varia de 1 a 65535 (caso haja mais de uma política, a maior preferência será ativa [Cisco 2023]) e como última instância, é necessário escolher o tipo de encaminhamento, usando o IGP OSPF ou neste caso, usando a *segment-list* criada anteriormente.

```

RP/0/RP0/CPU0:R1(config-sr-te)#policy tcc
RP/0/RP0/CPU0:R1(config-sr-te-policy)#color 20 end-point ipv4 6.6.6.6
RP/0/RP0/CPU0:R1(config-sr-te-policy)#autoroute
RP/0/RP0/CPU0:R1(config-sr-te-policy-autoroute)#include all
RP/0/RP0/CPU0:R1(config-sr-te-policy-autoroute)#exit
RP/0/RP0/CPU0:R1(config-sr-te-policy)#candidate-paths
RP/0/RP0/CPU0:R1(config-sr-te-policy-path)#preference 200
RP/0/RP0/CPU0:R1(config-sr-te-policy-path-pref)#explicit segment-list lista-1
RP/0/RP0/CPU0:R1(config-sr-te-pp-info)#commit

```

Figura 4.2: Aplicação da Segment List. FONTE: O autor (2023)

Após a aplicação das políticas, executando o comando *traceroute* com origem R1 e destino R6, é possível notar algumas mudanças na pilha MPLS que foi mostrada na Figura 4.2.

```

R1
RP/0/RP0/CPU0:R1#traceroute 6.6.6.6
Wed Dec  6 13:59:52.401 UTC

Type escape sequence to abort.
Tracing the route to 6.6.6.6

 0  192.168.1.2 [MPLS: Labels 16004/16005/16003/16006 Exp 0] 80 msec 81 msec 90 msec
 1  192.168.3.2 [MPLS: Labels 16005/16003/16006 Exp 0] 84 msec 69 msec 75 msec
 2  192.168.4.2 [MPLS: Labels 16003/16006 Exp 0] 77 msec 77 msec 77 msec
 3  192.168.5.2 [MPLS: Label 16006 Exp 0] 78 msec 73 msec 75 msec
 4  192.168.6.2 75 msec * 72 msec
RP/0/RP0/CPU0:R1#

```

Figura 4.4: Caminho do pacote com o SR-TE. FONTE: O autor (2023)

A primeira mudança notável é a pilha MPLS. Na Figura 4.4 no primeiro salto, é possível observar os *labels* que o pacote deve passar e ao longo do caminho os *labels* são submetidos a operação NEXT, que é o processo de remover o *label* do topo da pilha. Ainda na Figura 4.4 nota-se que o caminho que o pacote percorreu é o que foi especificado na lista de segmentos, passando pelas interfaces com IPs 192.168.1.2, 192.168.3.2, 192.168.4.2, 192.168.5.2 e 192.168.6.2 que são os roteadores R2, R4, R5, R3 e R6, respectivamente.

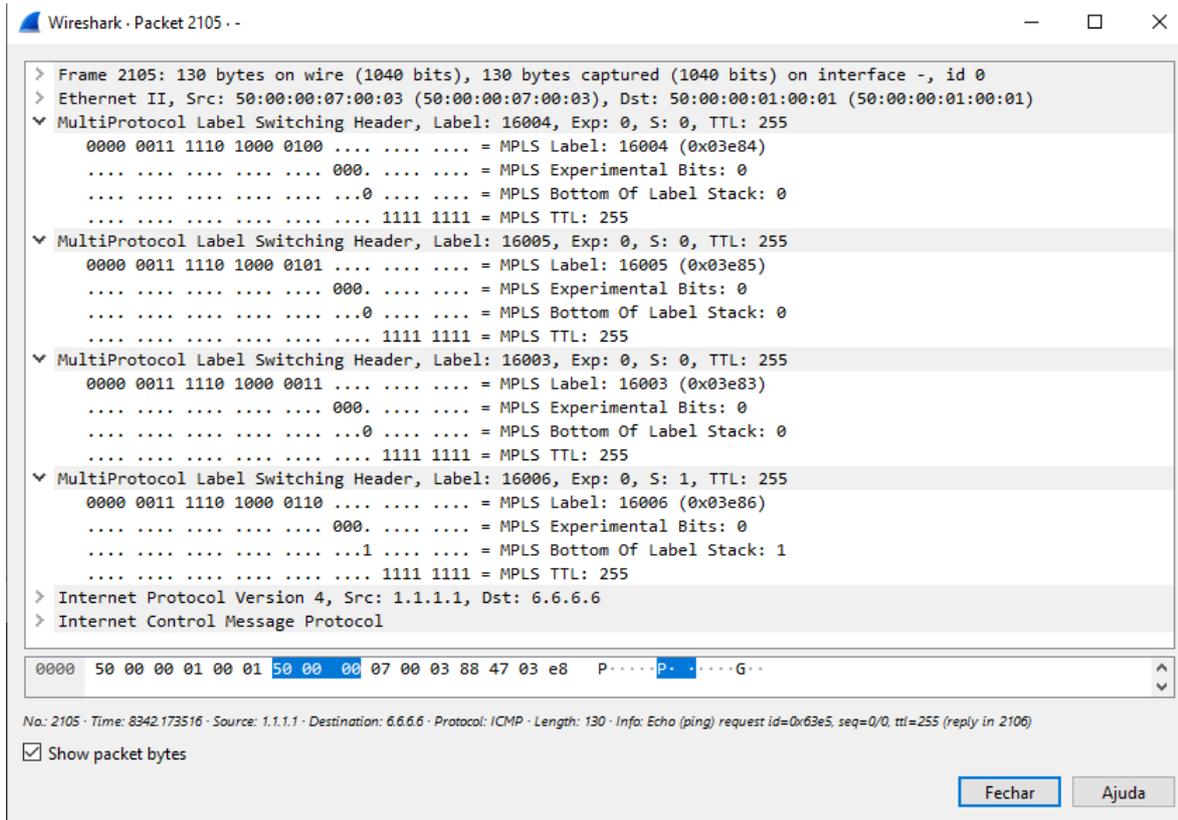


Figura 4.5: Pacote do teste de ping no Wireshark. FONTE: O autor (2023)

Ao capturar os pacotes do comando *ping* na interface *GigabitEthernet0/0/0/0* no roteador R1 com destino R6 é possível observar todos os *labels* que o pacote irá passar. Na Figura 4.5 nota-se que diferente do SR-BE, este pacote tem vários *labels*. Nas abas de MPLS observa-se os parâmetros discutidos na Seção 4.1, como também é notável o MPLS *Bottom Of Labels Stack: 0* sendo utilizado, isso significa que aquele não é o último roteador que o pacote vai passar e implica dizer que é feito a operação NEXT.

Diferente do SR-BE que utiliza o OSPF para calcular o caminho de ida e de volta, o SR-TE utiliza a lista de segmentos para fazer o caminho, porém esse caminho que é criado é unidirecional [Cisco. 2023], isso quer dizer que, o OSPF vai ser utilizado para fazer o ICMP (do inglês, *Internet Control Message Protocol*) *Reply*. A confirmação de que a rota de resposta ICMP não está seguindo o caminho especificado na *segment list* é feita a partir da captura do tráfego na interface *GigabitEthernet0/0/0/1* do roteador R4, por exemplo. Na Figura 4.6 nota-se que não há resposta ICMP passando pelo R4.

The screenshot shows a network traffic capture in Wireshark. The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help) and a toolbar. A display filter is applied: "Apply a display filter ... <Ctrl-/>". The packet list pane shows 27 packets. Packets 1-3 are OSPF Hello Packets (94 bytes). Packets 4-7 are ICMP Echo (ping) requests (122 bytes) with ID 0x6639, originating from 1.1.1.1 and destined for 6.6.6.6. Packets 8-27 are OSPF Hello Packets (94 bytes) originating from 192.168.4.2 and destined for 224.0.0.5. The packet details pane is currently empty.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.4.2	224.0.0.5	OSPF	94	Hello Packet
2	3.579275	192.168.4.1	224.0.0.5	OSPF	94	Hello Packet
3	5.920417	1.1.1.1	6.6.6.6	ICMP	122	Echo (ping) request id=0x6639,
4	6.015226	1.1.1.1	6.6.6.6	ICMP	122	Echo (ping) request id=0x6639,
5	6.091284	1.1.1.1	6.6.6.6	ICMP	122	Echo (ping) request id=0x6639,
6	6.172271	1.1.1.1	6.6.6.6	ICMP	122	Echo (ping) request id=0x6639,
7	6.245457	1.1.1.1	6.6.6.6	ICMP	122	Echo (ping) request id=0x6639,
8	9.609099	192.168.4.2	224.0.0.5	OSPF	94	Hello Packet
9	13.398539	192.168.4.1	224.0.0.5	OSPF	94	Hello Packet
10	19.098680	192.168.4.2	224.0.0.5	OSPF	94	Hello Packet
11	22.948147	192.168.4.1	224.0.0.5	OSPF	94	Hello Packet
12	28.757667	192.168.4.2	224.0.0.5	OSPF	94	Hello Packet
13	32.497583	192.168.4.1	224.0.0.5	OSPF	94	Hello Packet
14	38.526786	192.168.4.2	224.0.0.5	OSPF	94	Hello Packet
15	41.646877	192.168.4.1	224.0.0.5	OSPF	94	Hello Packet
16	47.566385	192.168.4.2	224.0.0.5	OSPF	94	Hello Packet
17	51.086122	192.168.4.1	224.0.0.5	OSPF	94	Hello Packet
18	57.395669	192.168.4.2	224.0.0.5	OSPF	94	Hello Packet
19	61.086678	192.168.4.1	224.0.0.5	OSPF	94	Hello Packet
20	67.275150	192.168.4.2	224.0.0.5	OSPF	94	Hello Packet
21	71.074891	192.168.4.1	224.0.0.5	OSPF	94	Hello Packet
22	76.945962	192.168.4.2	224.0.0.5	OSPF	94	Hello Packet
23	80.344338	192.168.4.1	224.0.0.5	OSPF	94	Hello Packet
24	86.583521	192.168.4.2	224.0.0.5	OSPF	94	Hello Packet
25	89.373355	192.168.4.1	224.0.0.5	OSPF	94	Hello Packet
26	96.173587	192.168.4.2	224.0.0.5	OSPF	94	Hello Packet
27	99.192696	192.168.4.1	224.0.0.5	OSPF	94	Hello Packet

Figura 4.6: Pacotes ICMP Request no roteador R4. FONTE: O autor (2023)

Para que o pacote siga o mesmo caminho na ida e na volta a configuração de SR-TE deve ser feita nos dois nós (R1 e R6), assim haverá o caminho explícito para a resposta dos pacotes.

4.3 Discussão

A convergência entre *Segment Routing* e Redes Definidas por *Software* representa uma abordagem que favorece a otimização do desempenho e a operação das redes. Ao integrar o SR no contexto do SDN, são observadas melhorias substanciais na flexibilidade e eficiência do encaminhamento. O SR, com sua capacidade de inserir identificadores de segmentos nos pacotes de dados, proporciona uma maneira inovadora de definir caminhos específicos na rede. A visão global mantida pelas controladoras SDN oferece um entendimento abrangente das condições da rede, permitindo ao SR traçar caminhos de maneira inteligente, com base em informações em tempo real [Kukreja *et al.* 2016]. Isso não apenas simplifica a gestão e configuração dos caminhos, mas também promove uma resposta ágil a mudanças na topologia da rede ou na demanda de serviços. A centralização do plano de controle proporcionada pelo SDN permite a coordenação eficiente de políticas de encaminhamento e a orquestração dinâmica de serviços, resultando em uma rede mais adaptável, escalável e capaz de oferecer suporte a uma variedade de requisitos de tráfego [Bannour, Souihi e Mellouk 2018]. Em síntese, a combinação de SR e SDN não apenas potencializa a eficiência operacional, mas

também oferece uma abordagem inteligente e centrada em políticas para a definição de trajetórias, contribuindo significativamente para a evolução das infraestruturas de rede.

Capítulo 5

Considerações Finais e Sugestões para Trabalhos Futuros

Este trabalho teve como objetivo o estudo da tecnologia *Segment Routing* (SR). Durante a pesquisa foi possível entender o funcionamento do MPLS e do SR e assim, fazer a implementação do SR em um *backbone* de um provedor de serviços, que a princípio, utilizava apenas o MPLS. Para a implementação, foi utilizado o simulador de redes EVE-NG, ao qual foram inseridas as imagens dos roteadores Cisco XRV 9K e Cisco XRVK9. A partir disso, foi possível fazer a topologia do *backbone* no ambiente simulado. Com a construção da topologia no EVE-NG, foi possível fazer a configuração do SR na sua forma SR-BE e SR-TE.

Com a implementação da tecnologia, foi possível observar o comportamento da rede com a captura de pacotes utilizando o *sniffer* de rede *Wireshark*. A partir do estudo do SR-BE e SR-TE e das implementações feitas é possível definir casos de usos para os dois modos abordados de SR-MPLS. O SR-BE é mais utilizado em redes onde há uma escalabilidade maior ocorrendo variação na topologia e em redes que não tem uma demanda por baixa latência. Enquanto o SR-TE pode ser utilizado para redes que oferecem diferentes tipos de serviços e precisam de prioridades diferentes para cada serviço. O SR também é compatível com o SDN que traz benefícios de flexibilidade, gerenciabilidade e controle de forma centralizada.

Espera-se que este trabalho contribua para a compreensão do SR-MPLS, e a partir dele, seja possível que profissionais e estudiosos da área consigam aplicar os conceitos apresentados.

5.1 Sugestões para Trabalhos Futuros

Como trabalho futuro busca-se fazer testes e comparar os cenários de rede com e sem o SR e avaliar métricas como latência, perda de pacote e utilizar diferentes tipos de serviços para a avaliação. Além de explorar sinergias entre *Segment Routing* e SDN (do inglês, *Software-Defined Network*), investigando como essas duas abordagens podem ser integradas para melhorar a flexibilidade e o controle da rede.

Apêndice A

Configurações dos Roteadores

Configuração dos roteadores utilizados para as implementações feitas.

```
!! IOS XR Configuration version = 6.5.1
!
hostname R1
!
interface Loopback0
ipv4 address 1.1.1.1 255.255.255.255
!
interface GigabitEthernet0/0/0/0
ipv4 address 192.168.1.1 255.255.255.0
!
router ospf 1
router-id 1.1.1.1
segment-routing mpls
segment-routing forwarding mpls
area 0
interface Loopback0 prefix-sid absolute 16001
!
interface GigabitEthernet0/0/0/0
network point-to-point
!
!
!
end
```

Figura A.1: Configuração do Segment Routing - Best Effort. FONTE: O autor (2023)

```
!! IOS XR Configuration version = 6.5.1
!
hostname R1
!
interface Loopback0
ipv4 address 1.1.1.1 255.255.255.255
!
interface GigabitEthernet0/0/0/0
ipv4 address 192.168.1.1 255.255.255.0
!
router ospf 1
router-id 1.1.1.1
segment-routing mpls
segment-routing forwarding mpls
area 0
interface Loopback0
prefix-sid absolute 16001
!
interface GigabitEthernet0/0/0/0
network point-to-point
!
!
!
segment-routing
traffic-eng
segment-list lista-1
index 10 mpls label 16002
index 20 mpls label 16004
index 30 mpls label 16005
index 40 mpls label 16003
index 50 mpls label 16006
!
policy tcc
color 20 end-point ipv4 6.6.6.6
autoroute
include all
!
candidate-paths
preference 200
explicit segment-list lista-1
!
!
!
!
!
!
end
```

Figura A.2: *Configuração do Segment Routing - Traffic Engineering. FONTE: O autor (2023)*

```
!! IOS XR Configuration 6.0.1
!
hostname R2
interface Loopback0
ipv4 address 2.2.2.2 255.255.255.255
!
interface GigabitEthernet0/0/0/0
ipv4 address 192.168.1.2 255.255.255.0
!
interface GigabitEthernet0/0/0/1
ipv4 address 192.168.2.1 255.255.255.0
!
interface GigabitEthernet0/0/0/2
ipv4 address 192.168.3.1 255.255.255.0
!
router ospf 1
router-id 2.2.2.2
segment-routing mpls
network point-to-point
segment-routing forwarding mpls
area 0
interface Loopback0
prefix-sid absolute 16002
!
interface GigabitEthernet0/0/0/0
network point-to-point
!
interface GigabitEthernet0/0/0/1
network point-to-point
!
interface GigabitEthernet0/0/0/2
network point-to-point
!
!
!
end
```

Figura A.3: *Configuração completa do roteador R2. FONTE: O autor (2023)*

```
!! IOS XR Configuration 6.0.1
!
hostname R3
interface Loopback0
ipv4 address 3.3.3.3 255.255.255.255
!
interface GigabitEthernet0/0/0/0
ipv4 address 192.168.2.2 255.255.255.0
!
interface GigabitEthernet0/0/0/1
ipv4 address 192.168.5.2 255.255.255.0
!
interface GigabitEthernet0/0/0/2
ipv4 address 192.168.6.1 255.255.255.0
!
router ospf 1
router-id 3.3.3.3
segment-routing mpls
segment-routing forwarding mpls
area 0
interface Loopback0
prefix-sid absolute 16003
!
interface GigabitEthernet0/0/0/0
network point-to-point
!
interface GigabitEthernet0/0/0/1
network point-to-point
!
interface GigabitEthernet0/0/0/2
network point-to-point
!
!
!
end
```

Figura A.4: *Configuração completa do roteador R3. FONTE: O autor (2023)*

```
!! IOS XR Configuration 6.0.1
!
hostname R4
interface Loopback0
ipv4 address 4.4.4.4 255.255.255.255
!
interface GigabitEthernet0/0/0/0
ipv4 address 192.168.3.2 255.255.255.0
!
interface GigabitEthernet0/0/0/1
ipv4 address 192.168.4.1 255.255.255.0
!
interface GigabitEthernet0/0/0/2
shutdown
!
router ospf 1
router-id 4.4.4.4
segment-routing mpls
segment-routing forwarding mpls
area 0
interface Loopback0
prefix-sid absolute 16004
! interface GigabitEthernet0/0/0/0
network point-to-point
!
interface GigabitEthernet0/0/0/1
network point-to-point
!
!
!
end
```

Figura A.5: *Configuração completa do roteador R4. FONTE: O autor (2023)*

```
!! IOS XR Configuration 6.0.1
!
hostname R5
interface Loopback0
ipv4 address 5.5.5.5 255.255.255.255
!
interface GigabitEthernet0/0/0/0
ipv4 address 192.168.4.2 255.255.255.0
!
interface GigabitEthernet0/0/0/1
ipv4 address 192.168.5.1 255.255.255.0
!
interface GigabitEthernet0/0/0/2
shutdown
!
router ospf 1
router-id 5.5.5.5
segment-routing mpls
segment-routing forwarding mpls
area 0
interface Loopback0
prefix-sid absolute 16005
!
interface GigabitEthernet0/0/0/0
network point-to-point
!
interface GigabitEthernet0/0/0/1
network point-to-point
!
!
!
end
```

Figura A.6: *Configuração completa do roteador R5. FONTE: O autor (2023)*

```
!! IOS XR Configuration 6.0.1
!
hostname R6
interface Loopback0
ipv4 address 6.6.6.6 255.255.255.255
!
interface GigabitEthernet0/0/0/0
ipv4 address 192.168.6.2 255.255.255.0
!
interface GigabitEthernet0/0/0/1
shutdown
!
interface GigabitEthernet0/0/0/2
shutdown
!
router ospf 1
router-id 6.6.6.6
segment-routing mpls
segment-routing forwarding mpls
area 0
interface Loopback0
prefix-sid absolute 16006
!
interface GigabitEthernet0/0/0/0
network point-to-point
!
!
!
end
```

Figura A.7: *Configuração completa do roteador R6. FONTE: O autor (2023)*

Referências Bibliográficas

ANDRÉ ORTEGA. *Tipos de LSAs e áreas OSPF*. 2023. Disponível em: <<https://k3gsolutions.freshdesk.com/support/solutions/articles/16000063352-tipos-de-lsas-e-areas-ospf>> Acesso em: 19 de Setembro 2023. xii, 4

BAKER, E. F. *Requirements for IP Version 4 Routers*. 1995. 3

BANNOUR, F.; SOUIHI, S.; MELLOUK, A. Distributed sdn control: Survey, taxonomy, and challenges. *IEEE Communications Surveys Tutorials*, v. 20, n. 1, p. 333–354, 2018. 21

Cisco. *Segment Routing Configuration Guide for Cisco ASR 9000 Series Routers, IOS XR Release 6.4.x*. 2018. Disponível em: <https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k-r6-4/segment-routing/configuration/guide/b-segment-routing-cg-asr9000-64x/b-segment-routing-cg-asr9000-64x_chapter_0111.html> Acesso em: 19 de Setembro 2023. 18

Cisco. *Compreender o OSPF (Open Shortest Path First) - Guia de design*. 2023. Disponível em: <https://www.cisco.com/c/pt_br/support/docs/ip/open-shortest-path-first-ospf/7039-1.html> Acesso em: 19 de Setembro 2023. 13, 18

Cisco. *Segment Routing Configuration Guide for Cisco ASR 9000 Series Routers, IOS XR Release 7.5.x*. 2023. Disponível em: <<https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k-r7-5/segment-routing/configuration/guide/b-segment-routing-cg-asr9000-75x.html>> Acesso em: 05 de Dezembro 2023. 20

EVE-NG. *EVE-NG Community Cookbook*. 2023. Disponível em: <<https://www.eve-ng.net/index.php/documentation/community-cookbook/>> Acesso em: 22 de Setembro 2023. 3, 10

FENG, L. *et al.* Strong Anonymous Communication System Based on Segment Routing Over SDN. *The Computer Journal*, p. bxac151, 11 2022. ISSN 0010-4620. Disponível em: <<https://doi.org/10.1093/comjnl/bxac151>>. 8

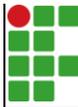
FILSFILS, C. *et al.* The segment routing architecture. In: . [S.l.]: Institute of Electrical and Electronics Engineers (IEEE), 2016. p. 1–6. 18

FILSFILS, C. *et al.* *Segment Routing Architecture*. RFC Editor, 2018. RFC 8402. (Request for Comments, 8402). Disponível em: <<https://www.rfc-editor.org/info/rfc8402>>. 5, 6

Huawei. *BGP/MPLS IP VPN Fundamentals*. 2023. Disponível em: <<https://support.huawei.com/enterprise/en/doc/EDOC1000178179/b5f87131/bgp-mpls-ip-vpn-fundamentals>> Acesso em: 19 de Setembro 2023. 10

Huawei. *BGP/MPLS IP VPN Fundamentals*. 2023. Disponível em: <<https://support.huawei.com/enterprise/en/doc/EDOC1100092117>> Acesso em: 28 de Novembro 2023. 15

- HUAWEI. *OSPF Fundamentals*. 2023. Disponível em: <<https://support.huawei.com/enterprise/en/doc/EDOC1100034072/32a400ec/ospf-fundamentals>> Acesso em: 19 de Setembro 2023. 4
- Juniper. *What is segment routing?* 2023. Disponível em: <<https://www.juniper.net/us/en/research-topics/what-is-segment-routing.html>> Acesso em: 19 de Setembro 2023. xii, 7
- KAUR, G.; KUMAR, D. Mpls technology on ip backbone network. *International Journal of Computer Applications*, v. 5, p. 13–16, 8 2010. ISSN 09758887. 10
- KUKREJA, N. *et al.* Demonstration of sdn-based orchestration for multi-domain segment routing networks. In: . [S.l.]: IEEE Computer Society, 2016. v. 2016-August. ISBN 9781509014675. ISSN 21627339. 21
- LINS, R. D.; OLIVEIRA, J. M. A. M. de; MENDONÇA, R. L. Redes-mpls-fundamentos-e-aplicacoes. xii, 9, 10, 17
- Liu Yan. *What is SR-MPLS*. 2023. Disponível em: <<https://info.support.huawei.com/info-finder/encyclopedia/en/SR-MPLS.html>> Acesso em: 13 de Dezembro 2023. 6
- MCKEOWN, N. Software-defined networking. *INFOCOM Keynote Talk*, v. 17, p. 30–32, 01 2009. 8
- MORENO, E.; BEGHELLI, A.; CUGINI, F. Traffic engineering in segment routing networks. *Computer Networks*, Elsevier B.V., v. 114, p. 23–31, 2 2017. ISSN 13891286. 5
- MOY, J. *OSPF Version 2*. RFC Editor, 1998. RFC 2328. (Request for Comments, 2328). Disponível em: <<https://www.rfc-editor.org/info/rfc2328>>. 4
- 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, p. e1199119562, 11 2020. ISSN 2525-3409. 3
- PORWAL, M. K.; YADAV, A.; CHARHATE, S. Traffic analysis of mpls and non mpls network including mpls signaling protocols and traffic distribution in ospf and mpls. In: . [S.l.]: IEEE, 2008. p. 187–192. ISBN 978-0-7695-3267-7. 5
- TANENBAUM, A. *Redes de computadores*. [S.l.]: Elsevier, 2003. 1, 3, 5
- VENTRE, P. L. *et al.* Segment routing: A comprehensive survey of research activities, standardization efforts, and implementation results. *IEEE Communications Surveys Tutorials*, v. 23, n. 1, p. 182–221, 2021. xii, 6, 7
- VISWANATHAN, A.; ROSEN, E. C.; CALLON, R. *Multiprotocol Label Switching Architecture*. RFC Editor, 2001. RFC 3031. (Request for Comments, 3031). Disponível em: <<https://www.rfc-editor.org/info/rfc3031>>. 5, 6

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus Campina Grande
	R. Tranquílino Coelho Lemos, 671, Dinamérica, CEP 58432-300, Campina Grande (PB)
	CNPJ: 10.783.898/0003-37 - Telefone: (83) 2102.6200

Documento Digitalizado Restrito

Entrega do trabalho de conclusão de curso

Assunto:	Entrega do trabalho de conclusão de curso
Assinado por:	Jose Galvao
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Restrito
Hipótese Legal:	Direito Autoral (Art. 24, III, da Lei no 9.610/1998)
Tipo da Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- José Silvestre da Silva Galvão, ALUNO (202021210010) DE TECNOLOGIA EM TELEMÁTICA - CAMPINA GRANDE, em 17/01/2024 14:13:46.

Este documento foi armazenado no SUAP em 17/01/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1054195

Código de Autenticação: ab516b497b

