Capítulo 5

Roteamento por Segmentos: Conceitos, Desafios e Aplicações Práticas

Antonio José Silvério (UFRJ/Embratel), Miguel Elias M. Campista (UFRJ) e Luís Henrique M. K. Costa (UFRJ)

Abstract

Currently, among the main problems of telecom operators are the huge number of states in the routers, manual configuration of traffic engineering and restoration paths in the network core. In this direction, the Segment Routing is an emerging proposal to simplify the routing and configuration of these networks. In Segment Routing, the flow states are kept only on the network edge nodes and configuration of the IP/MPLS core network, commonly used by operators, can be automated. To do this, Segment Routing uses the programming benefits of software defined network, by control plane centralization. The centralized controller view allows the dynamic calculation of the segments and the route construction between the edge nodes. Among the challenges to be explored in the short course are the problems faced by current operators and its solution through the Segment Routing.

Resumo

Atualmente, dentre os principais problemas das operadoras de telecomunicações estão o enorme número de estados nos roteadores, configurações manuais de engenharia de tráfego e restauração de caminhos no núcleo da rede. Nessa direção, o Roteamento por Segmentos é uma proposta emergente para simplificação do roteamento e da configuração dessas redes. No Roteamento por Segmentos, os estados por fluxo são mantidos apenas nos nós de borda da rede e a configuração das redes de núcleo IP/MPLS, comumente utilizadas pelas operadoras, pode ser automatizada. Para tal, o Roteamento por Segmentos utiliza os benefícios da programação das redes definidas por software através da centralização do plano de controle. A visão centralizada do controlador permite o cálculo dinâmico dos segmentos e a construção de rotas entre os nós de borda. Dentre os desafios a serem explorados no minicurso estão os problemas enfrentados pelas operadoras atuais e a solução através do Roteamento por Segmento.

5.1. Introdução

Na arquitetura TCP/IP o encaminhamento dos pacotes é feito salto a salto. Os campos do cabeçalho do pacote IP são analisados consultando-se a tabela de roteamento que contém os prefixos para as redes de destino. Quanto maior a tabela de roteamento, mais processamento a busca de rotas exige [De Ghein, 2007], aumentando a latência e jitter dos fluxos. Assim, um dos objetivos da tecnologia MPLS (*Multi Protocol Label Switiching*) foi lidar com o aumento do número de rotas nos roteadores de núcleo. Através do encaminhamento por rótulos de tamanho fixo, o MPLS possibilita acelerar a comutação dos pacotes [Santos et al., 2007].

As redes MPLS vem sendo adotadas como tecnologia típica da rede de roteadores de núcleo das operadora de telecomunicações. A tecnologia IP/MPLS provê um encaminhamento orientado a conexão através de circuitos virtuais unidirecionais. Inicialmente, tais circuitos eram estabelecidos seguindo os caminhos calculados pelo IGP (*Interior Gateway Protocol*) da rede, que tipicamente indicam o caminho mais curto. A inserção e remoção dos rótulos requer que os roteadores de borda conheçam o mapeamento entre rótulos MPLS e prefixos IP, configurados previamente.

Um problema que ocorre ao utilizar os caminhos calculados pelo IGP é a prevenção de situações de congestionamento ou do uso de caminhos de maior latência mesmo que com poucos saltos. Para contornar esse desafio, técnicas de engenharia de tráfego são frequentemente utilizadas para que os fluxos de dados utilizem rotas específicas por exemplo evitando congestionamento ou atendendo requisitos de qualidade de serviço (*Quality of Service* - QoS) [Sadok e Kamienski, 2000].

A engenharia de tráfego em redes MPLS é referenciada como MPLS-TE (MPLS – *Traffic Engineering*) [Systems, 2001]. O MPLS-TE permite a construção manual ou dinâmica de túneis que funcionam como caminhos da rede MPLS. O túnel é identificado por um rótulo por onde passam os fluxos de dados. As operadoras utilizam túneis para balanceamento de tráfego e para proteção de determinados fluxos com túneis primários (*primary tunnels*) e de proteção (*backup tunnels*). A Figura 5.1 mostra o balanceamento de tráfego utilizando MPLS-TE, bem como o desacoplamento das informações de enlaces físicos com a camada IP/MPLS. Esse desacoplamento é um problema atual, já que exige a configuração manual de túneis. Outro problema ocorre em caso de falhas nos enlaces físicos por longos períodos, já que torna-se necessário reconfigurar todos os túneis que passam pelo enlace defeituoso. Além disso, em geral, o plano de controle da rede IP/MPLS executa em hardware proprietário e de alto custo.

Os diferentes problemas fizeram com que, com o passar do tempo, o uso do MPLS nas redes de operadoras se tornasse um entrave. Nesse sentido, o paradigma das Redes Definidas por Software (*Software Defined Networks* - SDN) torna-se atrativo. Nas redes SDN, com a centralização do controle, é possível programar o roteamento da rede IP/MPLS por fluxo baseado nos objetivos da engenharia de tráfego. Nesse caso, a programação dos caminhos dos fluxos da rede pode ser realizada considerando, inclusive, mais de um parâmetro de configuração, como banda disponível, latência e grupos de risco compartilhado de enlaces físicos. Os roteadores de núcleo tornam-se comutadores "programáveis" e toda a lógica de programação é transferida para o controlador SDN, que pode executar em servidores de uso comum (*Commercial off the shelf* - COTS) e não mais

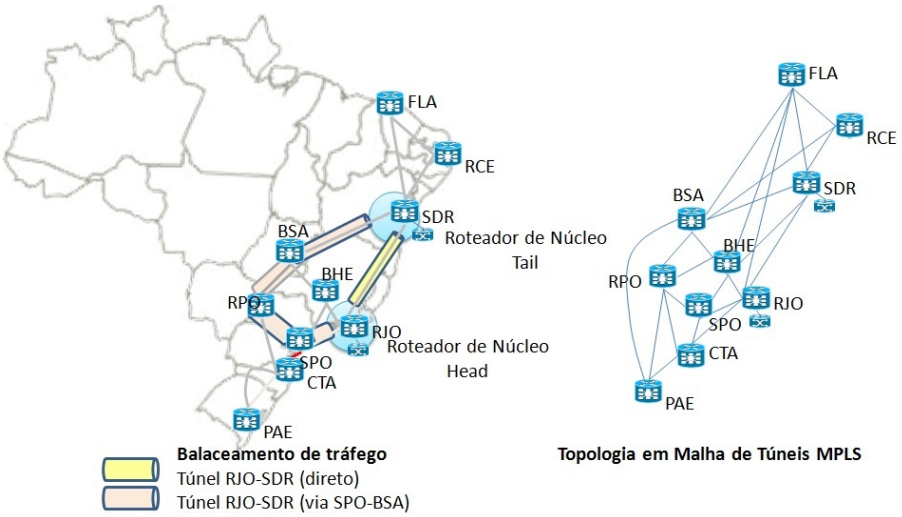


Figura 5.1. Configuração manual de túneis MPLS-TE.

em hardwares proprietários, de alto custo [Kreutz et al., 2015]. As interfaces (*northbound* e *southbound*) comunicam-se com o gerenciamento da rede através de APIs (*Application Programming Interfaces*) abertas ou proprietárias. Essa simplificação reduz as limitações da configuração de túneis TE das redes MPLS.

O roteamento por segmentos é uma proposta para simplificação do roteamento e configuração, mantendo estados por fluxo apenas nos nós de borda da rede, apoiando-se na programação das redes SDN. Em outras palavras, o roteamento por segmentos emprega roteamento pela fonte. Assim, o roteador de entrada da rede possui uma lista ordenada dos segmentos que definem o caminho na rede até o roteador de saída. O cálculo do caminho é feito pelo Elemento de Computação de Caminhos SDN (*SDN Path Computation Element*), ou através de uma aplicação SDN, podendo considerar objetivos de engenharia de tráfego. Em uma rede IP/MPLS, os segmentos são rótulos MPLS, que agrupados nos roteadores de núcleo de borda, informam o caminho do fluxo. Assim, apenas os roteadores de borda contêm a informação dos rótulos, não sendo necessário mantê-la nos roteadores intermediários. Essa característica permite reduzir a complexidade do plano de controle dos roteadores de núcleo.

O restante deste capítulo está organizado da seguinte forma. A Seção 2 introduz os problemas atuais das redes das operadoras na configuração e operação de redes IP/MPLS. A Seção 3 aborda o roteamento em redes SDN, apresentando uma visão geral da arquitetura SDN, detalhando as interfaces northbound e southbound usadas no roteamento por segmentos. A Seção 4 descreve os conceitos básicos do roteamento por segmentos, o funcionamento do plano de dados MPLS com roteamento por segmentos, e o plano de controle IGP para o roteamento por segmentos. Também são mostrados exemplos de roteamento por segmentos em IPv6 e a interoperabilidade de redes IP/MPLS legadas. Na Seção 5 são abordados simuladores de redes SDN, em especial o simulador Mininet e experimentos com controlador SDN OSHI SRTE (*Open Source Hybrid IP/SDN networking with Segment Routing and Traffic Engineering*). A Seção 6 conclui este minicurso destacando outras iniciativas e direções futuras.

5.2. Limitações das Tecnologias das Redes de Núcleo

As limitações da comutação de pacotes IP exigiram da indústria o desenvolvimento de tecnologias que trouxessem desempenho às redes de roteadores. As soluções inicialmente propostas utilizavam comutadores ATM com roteamento baseado em gerência de redes, evoluindo para o desenvolvimento de um hardware que ampliasse a capacidade de comutação, separando os protocolos de controle da comutação de pacotes, sendo esta última, o embrião das redes MPLS. A tecnologia MPLS [Rose, 2014] surgiu como uma proposta de evolução das redes públicas de comutação e encaminhamento de pacotes, motivada pela convergência da comunicação digital (voz, dados e vídeo) em uma infraestrutura comum. A ideia era integrar o MPLS com as redes IP e com outros tipos de redes legadas como o Ethernet, o Frame Relay e o próprio ATM [El-Sayed e Jaffe, 2002].

A rede IP/MPLS tem planos de controle e de encaminhamento de dados implementados em cada nó da rede. O plano de controle é mantido através de protocolos como o LDP (*Label Distribution Protocol*) para distribuição dos rótulos e um protocolo IGP, como o OSPF (*Open Shortest Path First*), para descoberta dos caminhos mais curtos denominados LSP (*Label Switched Path*). No caso de falha da rede, o OSPF encontra um novo caminho, desviando o tráfego para este [Santos et al., 2007, De Ghein, 2007].

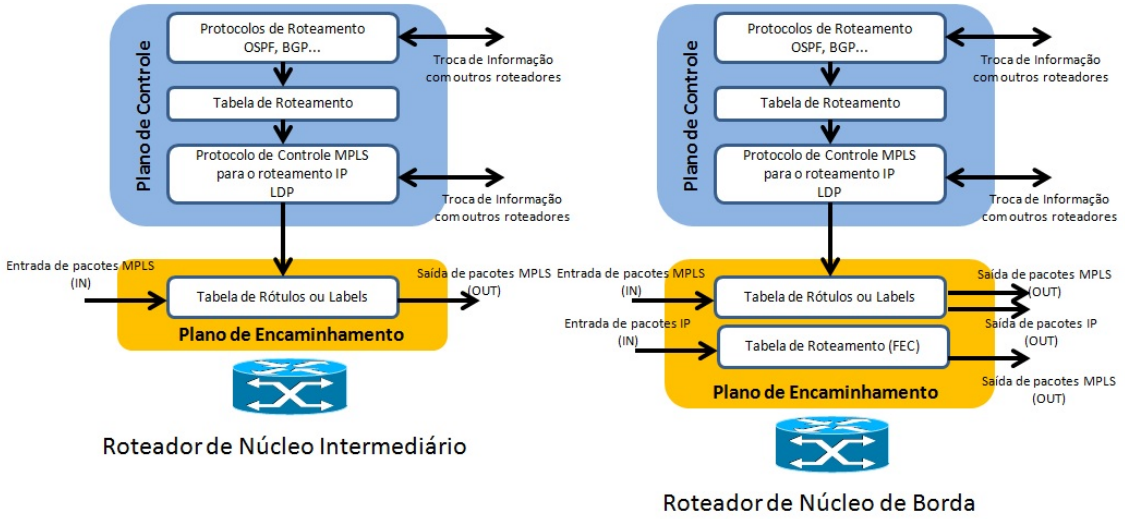
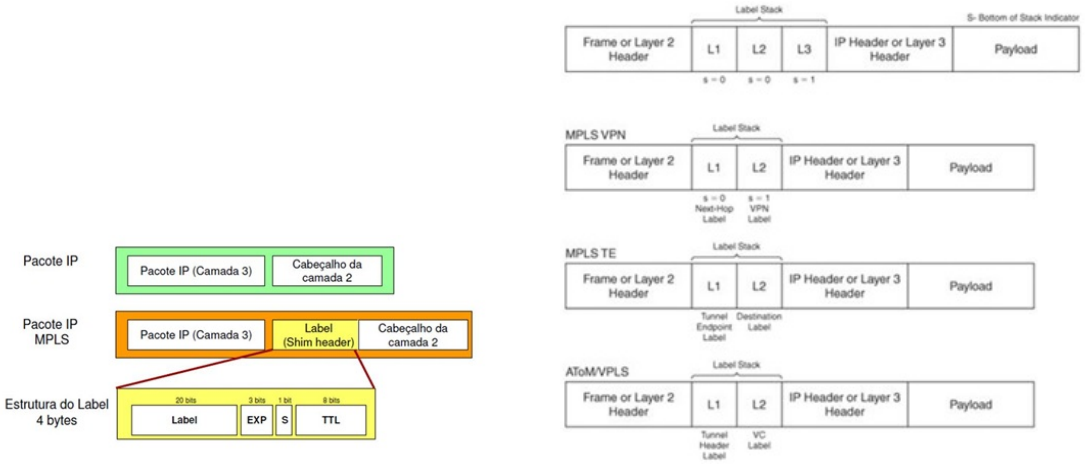


Figura 5.2. Representação de um roteador IP/MPLS.

5.2.1. Fundamentos do MPLS

Em uma rede MPLS, o pacote IP recebe um rótulo (*label*), inserido entre o cabeçalho IP e o cabeçalho do protocolo de camada inferior. Um rótulo MPLS possui 32 bits. Os primeiros 20 bits são o valor do rótulo. Os três próximos bits são experimentais, usados para marcação de classes de serviço. Esses bits são seguidos pelo bit S (*Bottom of Stack*), cujo valor é 0 se o rótulo é o inferior da pilha de rótulos ou 1 caso contrário. Os últimos 8 bits servem para limitar o tempo de vida do pacote (*Time to Live* - TTL) de forma similar ao IP [Marzo et al., 2003]. A Figura 5.3(a) ilustra o rótulo MPLS.

Os roteadores MPLS podem processar um conjunto de rótulos no pacote, organizados em pilha sendo o primeiro denominado de externo ou superior *(top label)*



(a) Estrutura do rótulo MPLS. (b) Empilhamento de rótulos MPLS.

Figura 5.3. Encapsulamento MPLS.

e o último rótulo, interno ou inferior *(bottom label)* (Figura 5.3(b)). Assim, são possíveis aplicações como redes privativas virtuais (*Vitual Private Networks* - VPN) e linhas alugadas virtuais (*Virtual Leased Lines* - VLL ou *Virtual Private LAN Service* - VPLS) [Sadok e Kamienski, 2000]. O transporte de outras tecnologias sobre o MPLS (*AToM - Any Transport Over MPLS*) é uma característica fundamental para o Roteamento por Segmentos [Pepelnjak e Guichard, 2003, Asati, 2012].

A tecnologia MPLS introduziu mecanismos como qualidade de serviço, engenharia de tráfego [Sadok e Kamienski, 2000], bem como novos serviços do tipo VPN na rede de núcleo das operadoras de telecomunicações. Existem dois tipos principais de VPN: de camada 2 (*VPN Layer 2*), por exemplo, o Ethernet ou o ATM, e o de camada 3 *(VPN Layer 3)*, que transporta protocolos de camada 3, mais especificamente, o IP. A rede IP/MPLS é muitas vezes considerada como uma rede multisserviço, já que oferece suporte a redes IP, VPN, Frame Relay, etc.

5.2.2. Arquitetura de Redes IP/MPLS

A arquitetura de uma rede IP/MPLS é composta por roteadores de borda denominados PE (*Provider Edge*) ou LER (*Label Edge Router*) que inserem e retiram rótulos dos pacotes. Os roteadores intermediários (P – *Provider* ou LSR – *Label Switch Router*) realizam a comutação de rótulos, enviando o pacote enlace a enlace. O CE (*Customer Edge*) representa o roteador IP que pertence a um determinado sistema autônomo [De Ghein, 2007, Systems, 2001]. A sequência de rótulos que forma o circuito virtual é um LSP (*Label Switched Path*), unidirecional (Figura 5.4). Ao separar o plano de controle do plano de encaminhamento de pacotes, o MPLS permite que novas facilidades sejam inseridas no plano de controle, sem necessidade de criar um novo plano de encaminhamento. Essa é a base para que aplicações como a engenharia de tráfego sejam criadas a partir do encaminhamento baseado em rótulos. A tabela FEC (*Fowarding Equivalence Class*) associa o endereço IP de destino com o rótulo de entrada na rede MPLS.

Os roteadores de borda (LER) recebem os pacotes IP provenientes do roteador

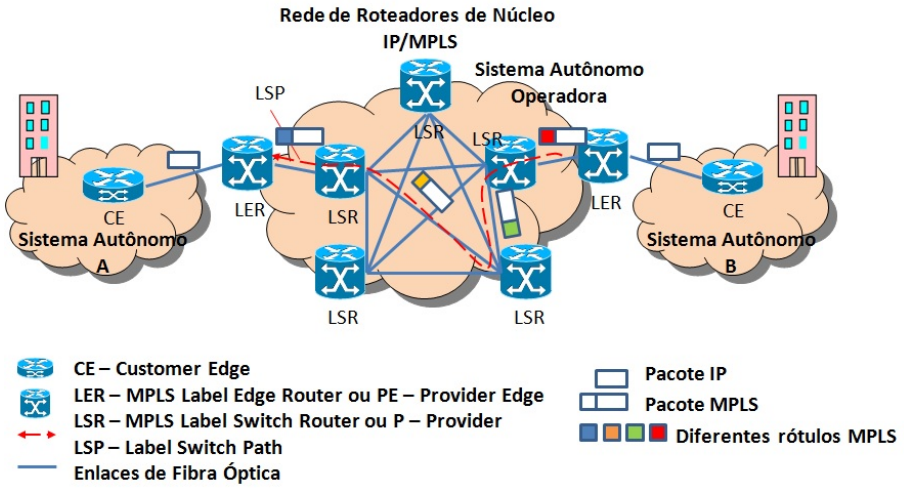


Figura 5.4. Arquitetura de Rede IP/MPLS.

cliente (CE) de origem, inserindo/retirando um rótulo adequado ao seu IP de destino e classe de serviço. Essas operações denominam-se inserção/remoção de rótulo (*push/pop*). No interior da rede, o pacote é comutado nos roteadores LSR intermediários apenas com base na informação dos rótulos. A informação de rótulos em uso está contida em uma base de instâncias de rótulos de encaminhamento (*Label Forwarding Information Base* - LFIB) em cada um dos roteadores intermediários do núcleo da rede. Na outra extremidade, o roteador MPLS de borda faz a operação inversa, retirando o rótulo e entregando o pacote para a rede do roteador cliente de destino. O processo de comutação de rótulos é mostrado na Figura 5.5. A tabela FEC associa o endereço IP de destino com o rótulo de entrada e saída na rede MPLS. O protocolo LDP (*Label Distribution Protocol*) é responsável pela distribuição de rótulos, trocando informações da FEC e rótulos associados entre os roteadores MPLS de borda e intermediários. Dessa forma, procura-se manter a coerência entre os LSPs formados e os prefixos IP dos pacotes encaminhados [Rose, 2014].

As tabelas FEC (Figura 5.5) são preenchidas por algum protocolo de roteamento (ex. OSPF, BGP) entre o CE de origem e o de destino. O protocolo LDP identifica seus vizinhos através de mensagens `HELLO`. Para cada entrada da tabela de roteamento é criada uma entrada na FEC, associando um rótulo de entrada. Os roteadores de núcleo (LERs e LSRs) anunciam aos seus vizinhos o par FEC/Rótulo de Entrada, sendo esta operação denominada distribuição de rótulos. Cada roteador de núcleo monta uma tabela completa com FEC/Rótulo de Entrada/Rótulo de Saída/Porta, denominada LFIB (Figura 5.5). Dessa forma, o plano de encaminhamento é responsável pela comutação dos pacotes baseado apenas nos rótulos envolvendo as operações de comutação, inserção e retirada de rótulos. No plano de controle, o protocolo LDP traduz as tabelas de rótulos em informações da tabela de roteamento. Ainda no plano de controle, o cálculo de caminhos na rede MPLS é feito por um IGP.

5.2.3. Serviços de Rede IP/MPLS

O principal serviço em redes IP/MPLS é o VPN, que pode ser configurado em topologias do tipo malha completa ou parcial, ou matriz-filial (*hub and spoke*). O MPLS

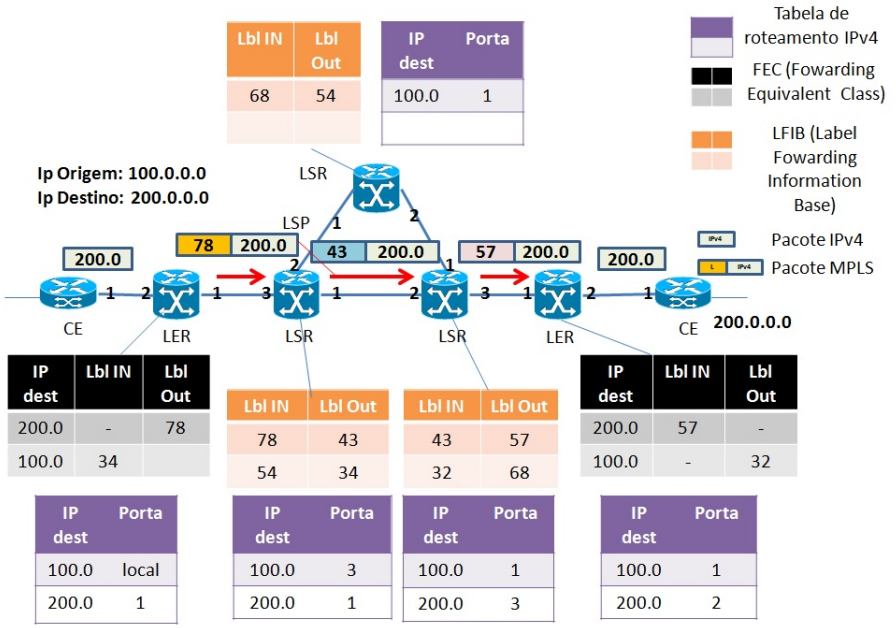


Figura 5.5. Comutação MPLS.

permite dois tipos de VPN [Pepelnjak e Guichard, 2003, Asati, 2012], a VPN de camada 3 (VPN L3) e a de camada 2 (VPN L2).

As VPNs L3 permitem a troca de protocolos de roteamento "*peering*" entre os roteadores cliente (CE) e os roteadores de borda do núcleo (LER). As portas do LER são associadas a uma VPN através de uma instância de roteamento chamada VRF (*Virtual Routing and Forwarding tables*). As VRFs são utilizadas para definir os LSPs. Como cada VPN utiliza sua própria VRF, os CEs de diferentes VPNs podem ter o mesmo endereço IP. Os CEs utilizam protocolos de roteamento tradicionais, como OSPF, RIP e BGP, com os LERs aos quais estão diretamente conectados. Os roteadores de borda (LER) trocam informações aprendidas pelos roteadores de cliente (CEs) diretamente conectados através do protocolo MP-BGP (*MultiProtocol-BGP*). Na percepção do roteador do cliente, a rede IP/MPLS funciona como uma rede IP dedicada. O processo de encaminhamento das VPNs utiliza dois rótulos, ou seja, o roteador de núcleo de borda insere dois rótulos MPLS entre o cabeçalho de camada 2 e o cabeçalho IP, empilhando rótulos. O rótulo mais externo refere-se ao roteador de núcleo (LER) de destino, enquanto o rótulo mais interno refere-se à VPN. O MPLS também possui a funcionalidade PHP (*Penultimate Hop Popping*) onde o penúltimo LSR (*penultimate LSR*) retira o rótulo mais externo antes de entregar o pacote ao LER. Esse processo é útil em VPNs L3, pois reduz a carga de processamento do LER, eliminando um rótulo para ser processado. O LER anuncia o rótulo de valor 3 que significa a funcionalidade de PHP. Esse rótulo é chamado *implicit-null*, que quando usado leva à perda das informações de QoS dos bits experimentais. A solução é o uso de rótulos especiais denominados *explicit-null* (valor 0), que é lido pelo LER para fins de QoS, mas removido da FIB. A Figura 5.6 ilustra uma VPN L3.

As VPNs L2 trocam com os roteadores de núcleo apenas informações de camada 2, sendo as informações de camada 3 trocadas apenas entre os CEs. As VPNs L2 podem

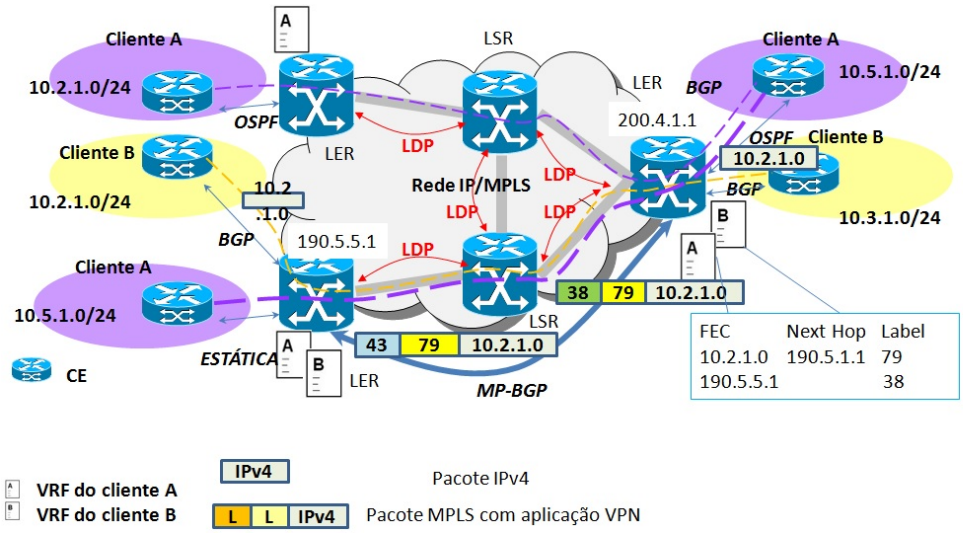


Figura 5.6. Serviço de rede privativa VPN L3.

ser entendidas como um enlace virtual, que pode ser ponto-a-ponto, no caso do VLL; e ponto-multiponto, no caso do VPLS. O transporte de quadros L2 introduziu o conceito de circuito virtual (*Virtual Circuit* - VC) no MPLS. Um LSP age como um túnel que permite carregar múltiplos VCs, enquanto o VC permite o transporte de quadros L2. O VC é implementando utilizando empilhamento de rótulos [IETF MPLS documents, 2001]. A Figura 5.7 mostra uma VPN L2 ponto-a-ponto, com um LSP conectando os roteadores de borda dos clientes A e B. O roteador de núcleo de borda A informa ao B que o roteador A receberá pela porta 1 os quadros L2 com outro rótulo, enviando os quadros com dois rótulos, fazendo com que chegue ao seu destino na porta correta.

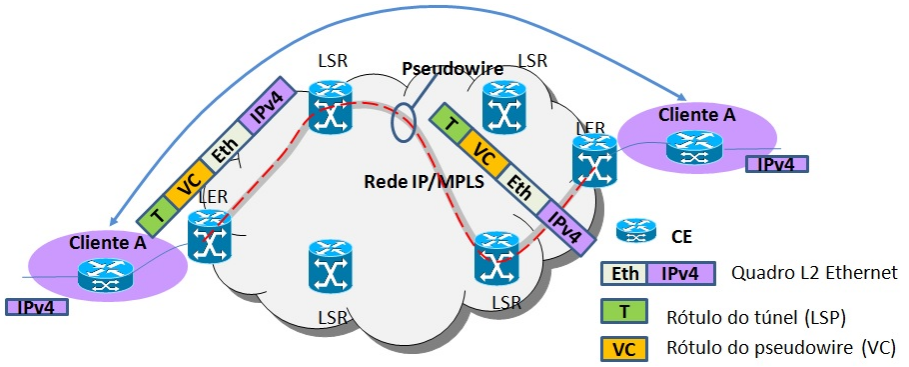


Figura 5.7. Serviço de rede privativa VPN L2 ponto a ponto (VLL).

5.2.4. QoS, Engenharia de Tráfego e Restauração em Redes IP/MPLS

O protocolo MPLS, assim como o protocolo IP, permite utilizar o modelo de Diff-Serv (*Differentiated Services*) [Marzo et al., 2003] para implementação de QoS. O Diff-Serv gerencia os recursos de rede e cria classes de serviço atendidas em filas diferentes conforme o nível de prioridade do fluxo de dados [IETF MPLS documents, 2001]. Enquanto no IPv4 a classe de serviço é indicada no campo DSCP (*Differentiated Services*

Code Point, anteriormente, ToS – *Type of Service*) [Marzo et al., 2003], no MPLS, a informação é definida no campo EXP bits do rótulo.

O conceito de engenharia de tráfego, MPLS-TE (*MPLS – Traffic Engineering*), foi desenvolvido para resolver problemas de congestionamento, otimização da banda e convergência da rede com proteção do tráfego por reserva de recursos. A estratégia do MPLS-TE é calcular caminhos baseados em restrições como banda disponível, latência e perda de pacotes, sugerindo uma rota diferente do IGP, que tipicamente não realiza engenharia de tráfego [Xiao et al., 2000, Alvarez, 2016]. O cálculo de caminhos baseados em restrições é feito pelo PCE (*Path Computation Element*), papel desempenhado pelo plano de controle dos roteadores de núcleo (Figura 5.8). O caminho calculado pela engenharia de tráfego é implementado como um túnel unidirecional (túnel TE) iniciado no roteador de núcleo de origem (*head end LSR*) e terminado no roteador de núcleo de destino (*tail end LSR*). O túnel é identificado a partir de um rótulo a mais além dos rótulos do protocolo LDP. A reserva de recursos para o caminho é feita pelo protocolo RSVP-TE (*Resource reSerVation Protocol – Traffic Engineering*), uma extensão do RSVP [Rose, 2014]. O RSVP-TE é usado para o estabelecimento de LSPs, que podem ser configurados manualmente (*Explicit LSP*) ou dinamicamente (*Dynamic LSP*) através de um IGP com extensões para engenharia de tráfego, como o OSPF-TE [Rose, 2014]. Na Figura 5.8, o roteador R1 utiliza a base de dados da topologia TE, que contém múltiplas métricas por enlace, para calcular o caminho até R8 através do PCE presente em R1. O RSVP-TE cria LSPs adicionais distribuindo os rótulos entre os roteadores R1 e R8. As mensagens RSVP-TE de caminho são enviadas periodicamente pelo roteador de núcleo de origem para cada roteador de núcleo intermediário pertencente ao túnel, contendo uma lista de atributos a serem analisados. Esses atributos possuem os valores das métricas utilizadas no cálculo dos caminhos. As mensagens de reserva de recursos são enviadas pelo roteador de destino após receber uma mensagem de estabelecimento de caminho, iniciando o processo de distribuição de rótulos, no qual cada roteador intermediário informa o rótulo a ser usado pelo antecessor. Se um dos roteadores intermediários não puder confirmar a reserva, uma mensagem de erro é enviada ao roteador de núcleo de origem do túnel [Alvarez, 2016].

Um dos requisitos da engenharia de tráfego é a capacidade de re-rotear um túnel TE baseado em políticas administrativas. O mecanismo de re-roteamento rápido (*Fast Reroute* – FRR) previne falhas de nós e de enlaces. O mecanismo pode ser usado para trocar para rotas com melhor desempenho, em caso de falhas de recursos do túnel, ou utilizar novas rotas por decisões administrativas. A Figura 5.9 mostra a criação do túnel primário formado pelos roteadores A, B, D, E e do túnel de proteção, provisionado nos roteadores B, C e D [Osborne e Simha, 2002, Alvarez, 2016]. A engenharia de tráfego também pode ser usada para balanceamento de tráfego através de múltiplos túneis TE. A malha de túneis e sua banda podem ser configurados manual ou automaticamente, através de um roteiro repetitivo de configurações [Osborne e Simha, 2002]. Em uma topologia em malha, tanto para balanceamento de tráfego, quanto para proteção, os caminhos precisam ser disjuntos. Entretanto, alguns túneis podem utilizar o mesmo recurso físico, como o mesmo cabo de fibra óptica. No exemplo da Figura 5.10, o SRLG (*Shared Risk Link Group*) de número 10 indica a presença de um enlace físico comum entre os caminhos R2-R4/R2-R3. A informação do SRLG é usada como um atributo adicional ao caminho original fornecido pelo IGP para evitar a configuração de túneis no mesmo enlace físico [Alvarez, 2016].

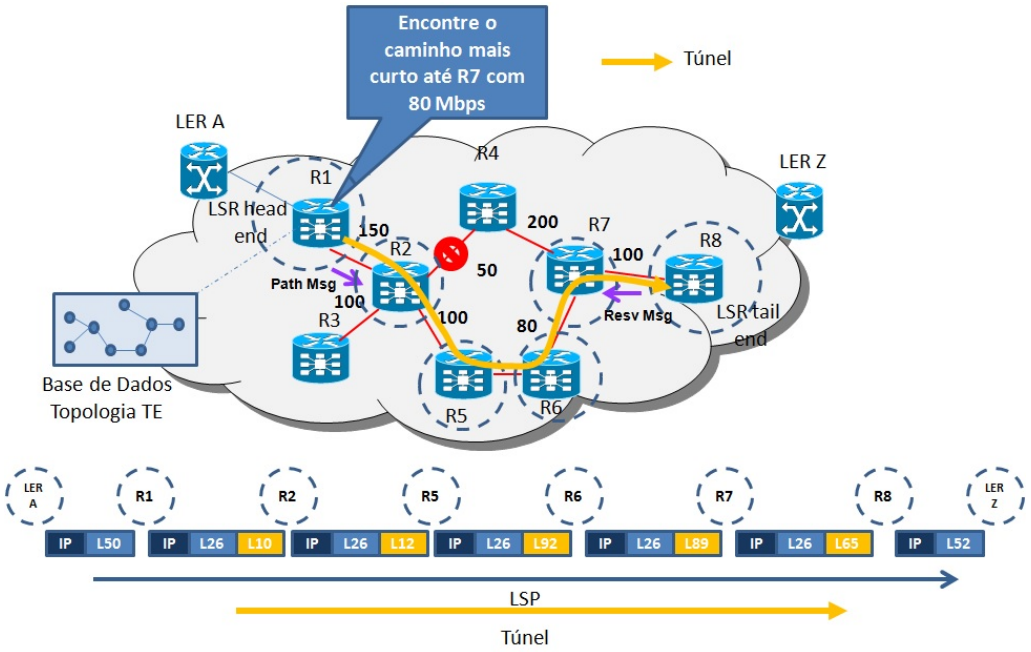


Figura 5.8. Cálculo de caminho do túnel TE.

5.2.5. Desafios para Engenharia de Tráfego e Restauração em Redes IP/MPLS

A operadora de telecomunicações utiliza a engenharia de tráfego para o tratamento de falhas nos enlaces físicos da rede. Tipicamente, a camada física ou rede de transporte é composta por equipamentos DWDM (*Dense Wavelength Division Multiplex*), ROADM (*Reconfigurable Add Drop Multiplex*) e comutadores OTN (*Optical Transport Network*). Esses elementos realizam a multiplexação e comutação do tráfego no domínio óptico e elétrico, além de realizar funções de formatação e regeneração dos sinais, de proteção e de restauração da rede. A rede de transporte possui um plano de controle próprio baseado no padrão ASON (*Automatically Switched Optical Network*) [Je e Ly, 2012] e GMPLS (*Generalized MPLS*) [J., 2011].

A rede de transporte é segregada da rede de roteadores de núcleo (camada L3), sendo necessária a configuração de SRLGs para evitar túneis primários e de proteção em um mesmo enlace físico. É comum o cenário com proteções e restaurações em ambas as redes, ocasionando desperdício da banda disponível. A Figura 5.11 ilustra a separação entre as redes de transporte e de roteadores de núcleo (rede IP/MPLS). Essa separação evidencia o desacoplamento dos planos de controle de cada camada, resultando em uma ocupação não otimizada da banda disponível como resultado da configuração de proteção de múltiplas camadas.

A configuração manual ocorre frequentemente devido às mudanças da matriz de tráfego e da topologia física, muitas vezes como decorrência de falhas nos enlaces. Em qualquer dos casos, a malha de túneis deve ser reconfigurada para se adequar ao novo padrão de tráfego. Outro problema das operadoras é o tempo de convergência da rede em caso de falhas, intrinsecamente relacionado à separação da rede de roteadores de núcleo (camada L3) da rede de transporte (camadas L0, L1 e L2). A desvinculação da topologia

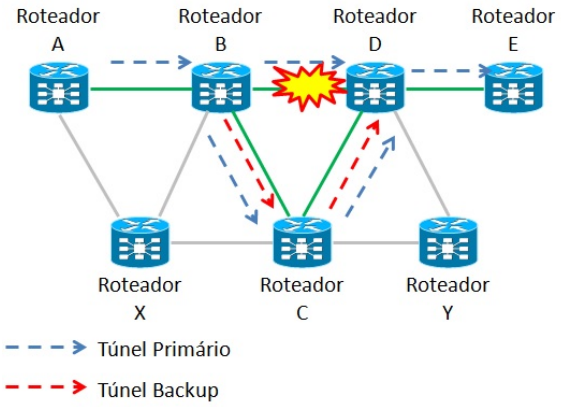


Figura 5.9. Túnel primário e de proteção.

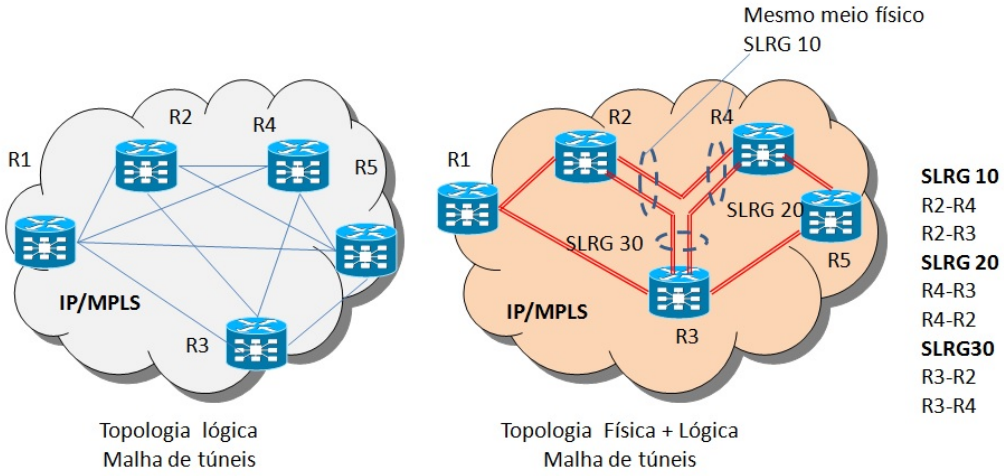


Figura 5.10. SLRG (*Shared Risk Link Group*).

de rede física da rede lógica de túneis TE é um desafio enfrentado pelas operadoras, pois uma interrupção em um enlace físico pode corresponder à interrupção de ambos os túneis, o TE principal e reserva.

5.3. Conceitos de Redes Definidas por Software

O Roteamento por segmentos utiliza aplicações de Redes Definidas por Software (SDN) para o cálculo dos caminhos dos túneis MPLS, baseado nos parâmetros de engenharia de tráfego. A configuração do túnel é representada por uma lista ordenada de segmentos. As SDNs constituem uma mudança de paradigma em redes: separando o plano de controle do plano de encaminhamento, rompe a integração vertical comum em roteadores e comutadores. A centralização do controle da rede por outro lado introduz maiores capacidade de programação e flexibilidade de utilização [Kreutz et al., 2015, Hu et al., 2014].

5.3.1. Arquitetura e Desenvolvimento da SDN

Em redes IP tradicionais os plano de controle e de dados são acoplados e embutidos no mesmo dispositivo de rede. Essa integração vertical é uma das razões pelas quais

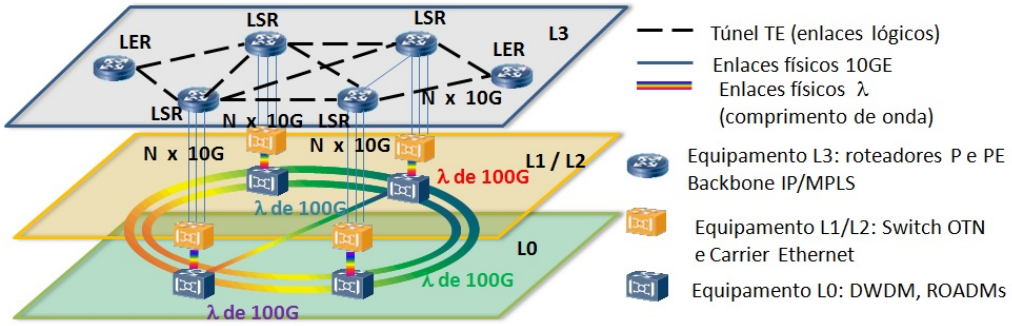


Figura 5.11. Separação de camadas de rede de transporte e IP/MPLS.

as redes IP possuem gerenciamento e controle complexos, sendo ainda pouco flexíveis à introdução de inovações tecnológicas. Para gerência de redes, a maior parte dos fabricantes oferece soluções com hardware, sistemas operacionais e programas de controle proprietários. As redes de núcleo IP/MPLS apresentam também grande complexidade de configuração devido aos inúmeros protocolos envolvidos, bem como a falta de flexibilidade em resposta a mudanças da rede, falhas e balanceamento de tráfego.

A arquitetura de rede SDN se baseia em quatro pilares: desacoplamento do plano de controle do plano de dados; decisões baseadas em fluxos ao invés do endereçamento de destino; programação de fluxos flexível, limitada ao tamanho das tabelas de fluxos; e lógica de controle em entidade externa denominada controlador. O controlador é uma plataforma de software ou um sistema operacional de rede (*Network Operating System* - NOS) executado em um servidor de uso comum COTS. O objetivo é prover os recursos essenciais de rede e abstrações para facilitar a programação de qualquer dispositivo de encaminhamento. Em uma visão simplificada da arquitetura SDN, a infraestrutura física é formada pelos dispositivos de encaminhamento, como comutadores ou roteadores sem plano de controle ou com o plano de controle reduzido e com interfaces de programação abertas. A Figura 5.12 mostra a arquitetura de uma SDN [Kreutz et al., 2015]. O plano de dados é constituído pela infraestrutura de rede e interfaces *southbound* (SBI – *southbound interface*). A infraestrutura de rede, similar a uma rede tradicional, corresponde aos dispositivos que desempenham instruções elementares de encaminhamento de tráfego. Estas instruções são definidas por interfaces abertas SBI, como por exemplo o OpenFlow [Lara et al., 2014]. O plano de controle é constituído por uma camada de virtualização, outra camada correspondente ao sistema operacional de rede e as interfaces *northbound* (NBI). O sistema operacional de rede pode estar ou não em uma infraestrutura de hardware virtualizada através do hipervisor, que permite a criação de máquinas virtuais no servidor físico onde é instalado o controlador de rede. O controlador ou sistema operacional de rede programa os dispositivos de rede. Por fim, as interfaces *northbound* oferecem APIs para desenvolvedores de aplicações. O plano de gerência corresponde ao conjunto de aplicações, utilizando linguagens de programação apropriadas para o ambiente virtualizado, que interliga todas as funcionalidades de rede para que as aplicações possam implementar a lógica e controle da rede através da NBI. O plano de dados é responsável pelo encaminhamento dos pacotes, e é composto pela infraestrutura de rede, que corresponde ao elementos físicos da rede e por interfaces NBI responsáveis pela comuni-

cação entre o plano de controle e o plano de dados. A seguir serão detalhadas cada uma das camadas da arquitetura SDN.

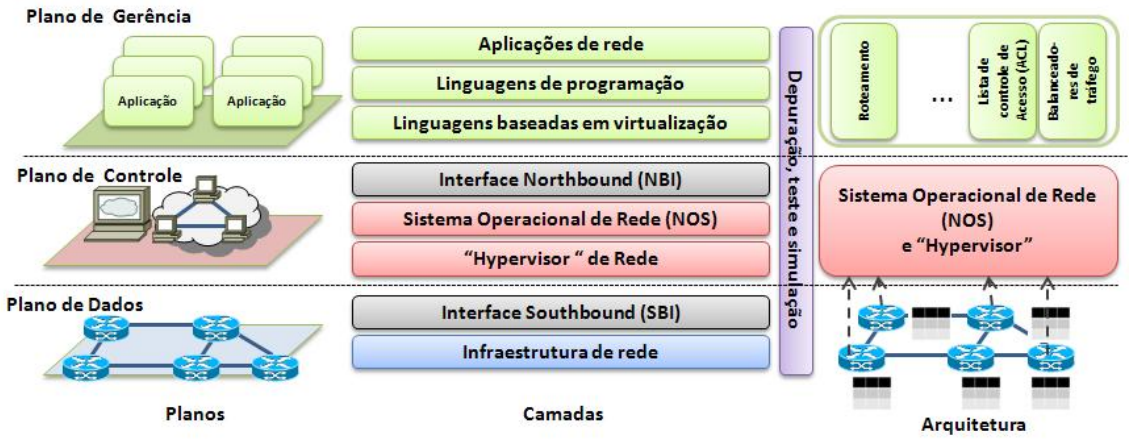


Figura 5.12. Diagrama de arquitetura de sistema de rede SDN.

5.3.2. Plano de Dados

O plano de encaminhamento de dados é composto pela infraestrutura de rede física e pelas interfaces *southbound* onde protocolos abertos ou proprietários podem ser usados para programação dos dispositivos de rede, como o CWMP (*CPE WAN Management Protocol*) definido na norma ETSI TR-069 ou o protocolo SNMP (*Simple Network Management Protocol*) padronizado pelo IETF. O SNMP foi concebido para gerenciamento de dispositivos de rede através de operações do tipo "GET" e "SET" de informações em uma base de dados (*Management Information Base* – MIB).

5.3.2.1. Infraestrutura de Rede

A infraestrutura física das redes SDN é similar à das redes tradicionais, com a diferença dos dispositivos serem simplesmente elementos de encaminhamento de dados [Nunes et al., 2014], ou seja, sem controle ou software embarcado que permita o dispositivo tomar decisões autônomas. A inteligência reside no sistema operacional de rede e nas aplicações de rede, que acessam os elementos da infraestrutura através das interfaces *southbound*, como por exemplo o OpenFlow.

5.3.2.2. Interfaces Southbound: NETCONF, PCEP, BGP-LS, OpenFlow e outros

Atualmente, muitas interfaces abertas podem ser empregadas como interface *southbound* em redes SDN. Além do OpenFlow, pode-se utilizar os protocolos NETCONF [RFC6241, 2011], BGP-LS (*Border Gateway Protocol - Link State*) [RFC7752, 2016a], PCEP (*Path Computation Element Communication Protocol*) [RFC5440, 2009], OVSDB (*Open vSwitch Database*) [RFC7047, 2013], SNMP (*Simple Network Management Protocol*) [RFC1157, 1990], a TL1 (*Transaction Language 1*) [Telcordia GR-831, 1996] e a CLI (*Command Line Interface*) [ISO/IEC 23271:2012, 2012]. A linguagem TL1 foi

desenvolvida especificamente para equipamentos de telecomunicações, cujo objetivo é a comunicação entre máquinas. O protocolo NETCONF foi desenvolvido para ser o sucessor natural do SNMP, pois o SNMP tinha o foco no monitoramento e não na configuração da rede. O OVSDB é um protocolo de gerenciamento projetado para redes definidas por software. Originalmente o OVSDB era parte do OVS (*Open vSwitch*), um comutador virtual projetado para hipervisores Linux. O OVS representa uma evolução dos protocolos de gerenciamento de rede, permitindo programar e configurar pontes, portas e interfaces de plataformas de equipamentos SDN e de virtualização de funções de rede (*Network Functions Virtualization* – NFV) [Andreas et al., 2015]. A seguir são detalhadas os principais protocolos de interfaces SBI.

NETCONF e YANG

O NETCONF é um protocolo de gerenciamento de rede definido pelo IETF para configuração e monitoramento da rede, sendo este último papel desempenhado pelo SNMP desde o final dos anos 80. O SNMP tinha como ponto fraco a ausência de recursos de configuração de rede além da interface binária BER (*Basic Encoding Rules*) e MIBs proprietárias. O protocolo NETCONF utiliza mecanismos que permitem instalar, manipular e apagar a configuração de dispositivos de rede através de uma implementação cliente-servidor, ilustrada na Figura 5.13.

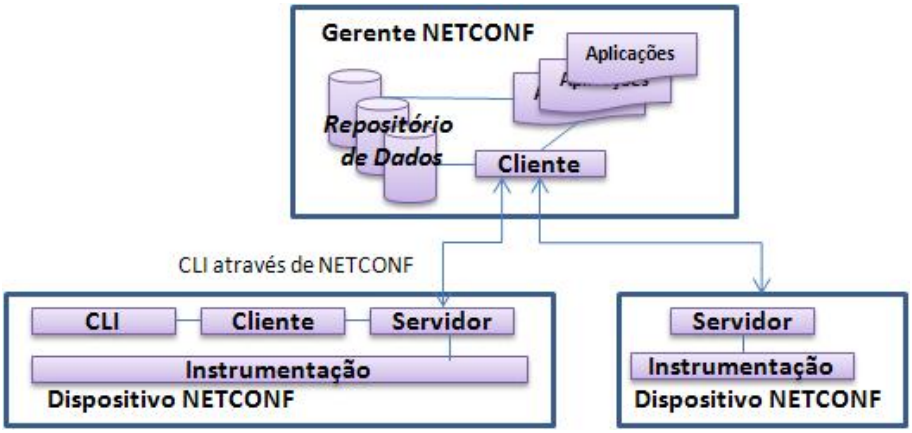


Figura 5.13. Modelo de implementação cliente-servidor do NETCONF.

Após o estabelecimento da sessão segura de transporte entre cliente e servidor, o protocolo NETCONF envia uma mensagem `HELLO` para anúncio das capacidades do protocolo e modelos de dados suportados. O NETCONF suporta ainda a subscrição e o recebimento de notificações de eventos de forma assíncrona assim como o fechamento parcial de uma configuração corrente de um dispositivo de rede. Esta funcionalidade permite múltiplas sessões de edição, agilizando o processo de configuração. O NETCONF permite o monitoramento e gerenciamento por uma entidade autônoma (o gerente NETCONF) que utiliza repositório de dados, sessões, fechamentos e estatísticas disponíveis no servidor NETCONF.

O YANG é uma linguagem formal com texto claro de modelo de dados com sin-

taxe e semântica que permitem a construção de aplicações de rede. O modelo YANG pode se traduzir em um arquivo de formato XML (*eXtensible Markup Language*) ou JSON (*JavaScript Object Notation*), estruturado em uma árvore para cada módulo, com propriedades que correspondem às funcionalidades do dispositivo; e declarações de tipos, dados, restrições e acréscimos de estruturas reutilizáveis [Nwa et al., 2010]. A Figura 5.14 mostra um diagrama representativo do modelo YANG, onde cada dispositivo de rede tem seus atributos descritos em um modelo de abstração, ilustrado na Figura 5.15. O protocolo NETCONF transporta estas informações até uma gerência de aplicações, e as aplicações podem inferir as configurações necessárias nos dispositivos de rede. A Figura 5.15 exemplifica um módulo YANG de inventário de interfaces um comutador OpenFlow. A estrutura em árvore do módulo representa os atributos de endereçamento da interface, com tipos de dados definidos [RFC6020, 2010].

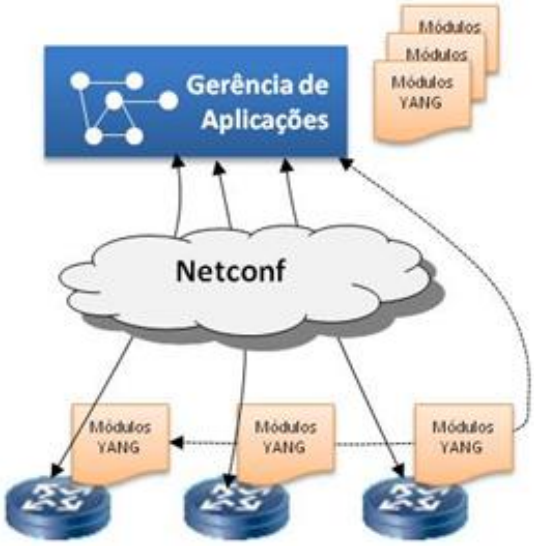


Figura 5.14. Modelo de dados YANG.

```
+--rw if:interfaces
 +--rw if:interface [name]
    ...
    +--rw ipv4
       +--rw enabled?            boolean
       +--rw ip-forwarding?      boolean
       +--rw address [ip]
          +--rw ip               inet:ipv4-address
          +--rw (subnet)?
             +--:(prefix-length)
             |  +--rw ip:prefix-length?   uint8
             +--:(netmask)
                +--rw ip:netmask?        inet:ipv4-address
```

Figura 5.15. Exemplo de módulo YANG para inventário de rede.

BGP-LS

Existem duas formas básicas de obtenção de informações de topologia de rede: protocolos de gerenciamento e roteamento. Um protocolo de roteamento responsável por obter a informação da topologia da rede é o BGP-LS (*Border Gateway Protocol - Link*

State), uma extensão do BGP [RFC7752, 2016b] que permite carregar informações dos estados dos enlaces. Essa informação é usada pelo IGP, que normalmente utiliza outra base de informações de estados dos enlaces como a TED (*Traffic Engineering Database*) usada na engenharia de tráfego, sendo que ambas provêem o mesmo conjunto de informações. No caso do BGP-LS, as informações podem ser agregadas de múltiplas áreas e de diferentes sistemas autônomos, permitindo uma análise abrangente do estado de toda rede. O BGP-LS foi desenvolvido especificamente para melhorar a escalabilidade do BGP como o controle baseado em fluxos TCP e no uso estratégico de roteadores RR (*Router Reflectors*). Em ambos os casos é necessário adquirir informações de topologia multi-área, o que é feito tradicionalmente por um elemento da rede de um sistema autônomo que reúne as informações dos demais elementos de outros sistemas autônomos através de meios manuais. Um controlador de engenharia de tráfego, por exemplo, uma aplicação de servidor PCE (*Path Computation Server* - PCS) implementa o BGP-LS para adquirir a topologia de rede a ser utilizada no roteamento da mesma [Casellas et al., 2015]. O BGP-LS suporta também mecanismos de políticas que limitam o uso de certos nós e enlaces da rede, ou seções de topologia particionadas pelo operador da rede. A Figura 5.16 mostra um controlador SDN interagindo com dispositivos de rede de diferentes ASes, com um IGP tradicional e com o BGP-LS. Note como o BGP-LS tem visão mutidomínio através do BGP-LS RR (*BGP-LS Router Reflector*) quando comparado à solução com IGP tradicional.

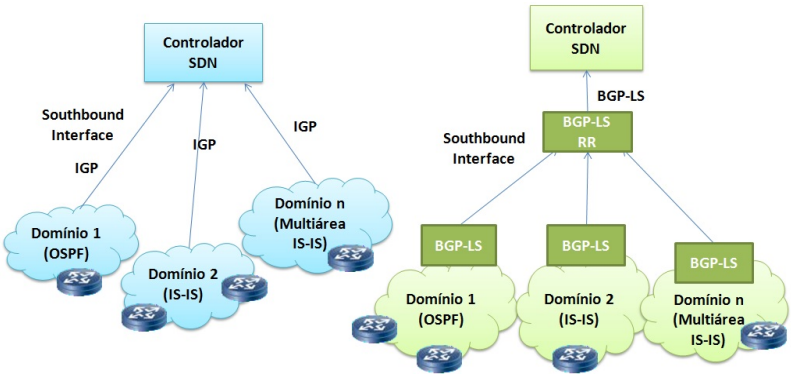


Figura 5.16. Southbound interface com IGP e com BGP-LS.

PCEP

O protocolo PCEP (*Path Computation Element Protocol*) é usado para comunicação entre o PCC (*Path Computation Client*) e o PCE (*Path Computation Element*), utilizando informações da base de estados do enlace, conforme a Figura 5.17. O PCE de controle dinâmico (*stateful PCE*) e o PCE iniciado pelo LSP são extensões ainda em discussão no IETF para habilitar o emprego destes protocolos em SDNs [Paolucci et al., 2013]. As extensões ainda não avançaram como uma proposição de padrão final devido à dificuldade de garantir nos padrões a orquestração entre PCEs de diferentes fornecedores [Casellas et al., 2015].

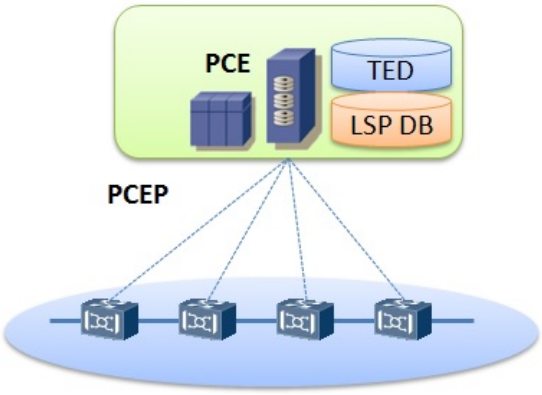


Figura 5.17. Arquitetura de Rede com Protocolo PCEP.

OpenFlow

Uma das principais tendências de SBI aberta para SDNs é o protocolo OpenFlow. O OpenFlow utiliza uma tabela com regras de tratamento de pacotes, na qual cada regra permite ações como encaminhamento, descarte e modificação do fluxo. O OpenFlow permite controlar os fluxos de dados encaminhando e processando os pacotes, conforme ações e regras configuradas pelo controlador nos comutadores OpenFlow. Os comutadores OpenFlow e o controlador da rede são interligados através de um canal de comunicação TLS (*Transport Layer Security*), conforme a arquitetura mostrada na Figura 5.18.

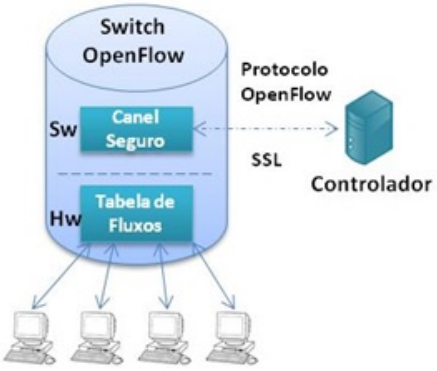


Figura 5.18. Arquitetura de uma rede OpenFlow.

A Figura 5.19 resume o funcionamento da SBI OpenFlow, com suas entradas na tabela de fluxo, cada uma definindo uma regra, uma ação e contadores de estatísticas. As regras são baseadas em informações do cabeçalho das camadas de enlace, rede e transporte. A partir da versão 1.1, o OpenFlow passou a suportar VLAN e Q-in-Q, portas virtuais e túneis, roteamento multi-percurso, ECMP (*Equal-Cost Multi-Path*) e MPLS.

5.3.3. Plano de Controle

O controlador representa o plano de controle em uma SDN [Xiea et al., 2015]. O controlador e as aplicações de rede podem ser instalados ou não em uma infraestrutura virtualizada. O controle da rede e as funções são abstraídas com a introdução de um

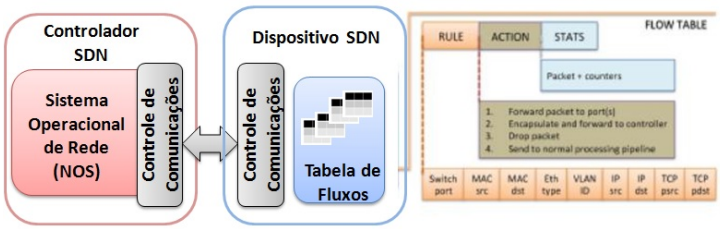


Figura 5.19. Tabelas de fluxos OpenFlow.

hipervisor [Andreas et al., 2015], criando uma rede independente do tipo de aplicação. O gerenciamento e orquestração da rede podem ser realizados por um orquestrador, como por exemplo, o OpenStack usado em nuvens computacionais [Huang et al., 2014]. A ideia é permitir que servidores e a infraestrutura da rede sejam virtualizados e implementados na nuvem [Matias et al., 2015]. O hipervisor permite a criação, remoção e movimentação das máquinas virtuais dos controladores SDN e de aplicações de rede. A Figura 5.20 mostra uma arquitetura comparativa de uma rede tradicional e de rede de SDN através de APIs existentes, onde o ambiente de virtualização é utilizado na camada de controle e gerenciamento, e finalmente a rede SDN com virtualização de funções de rede (NFV), denominado SDN overlay onde o SDN e a NFV atuam em conjunto [Xia et al., 2015].

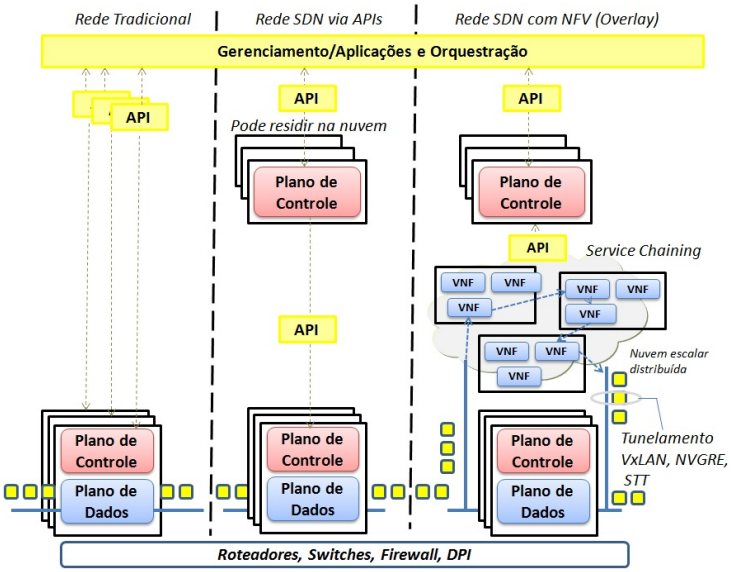
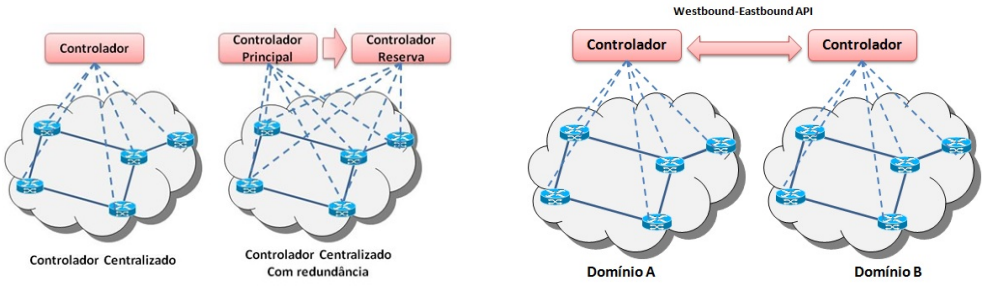


Figura 5.20. SDN via API e SDN Overlay.

5.3.3.1. Controladores SDN: Opendaylight, ONOS e outros

O controlador SDN, corresponde ao sistema operacional de rede, provêem APIs de alto nível para acessar um nível de abstração mais baixo dos dispositivos de rede. Tais instruções de nível mais baixo são específicas do dispositivo de rede e, na maioria das vezes, sistemas operacionais proprietários, como o IOS da Cisco ou o Junos da Juniper.

Atualmente, os projetistas de protocolos de roteamento precisam lidar com algoritmos distribuídos complexos para a solução de problemas em redes. As redes SDN facilitam o gerenciamento das redes e as soluções de problemas através da lógica centralizada. Assim, o controlador é o elemento crítico de uma arquitetura SDN. Existe uma grande diversidade de controladores e plataformas de controle com diferentes projetos e arquiteturas. As arquiteturas relevantes são com controladores centralizados ou distribuídos (Figura 5.21(a)). Um controlador centralizado tem a desvantagem de ser um ponto único de falhas, com escalabilidade limitada. No entanto, a centralização significa simplicidade de operação e melhor visão do comportamento da rede [Scott-Hayward, 2015]. Os controladores de código aberto NOX [Gude et al., 2016], POX [Kaur et al., 2016], Floodlight [Haleplidis et al., 2015] e Ryu [development team, 2016] são do tipo centralizado.



(a) Controlador centralizado e centralizado com redundância.

(b) Controlador distribuído..

Figura 5.21. Arquitetura de controladores.

Os controladores distribuídos podem ser organizados em *cluster* ou em um conjunto de controladores distribuídos em diferentes domínios de rede [Civanlar et al., 2015]. São exemplos de controladores distribuídos o Opendaylight [Medved et al., 2014] e o ONOS [Stancu et al., 2015]. Para controladores distribuídos em diferentes domínios, as APIs de fronteira leste e oeste (*eastbound-westbound*) [Pingping et al., 2015] são importantes. Atualmente, cada controlador implementa sua API de fronteira leste-oeste com o objetivo de troca de dados, verificação da consistência do modelo de dados e monitoramento de notificações. A maioria dos controladores distribuídos oferece uma consistência semântica baixa, ou seja, as atualizações de nós distintos são eventualmente atualizadas em todos os controladores. Isso implica períodos de tempo onde controladores distintos possuem visões diferentes para a mesma propriedade. A consistência forte, por outro lado, assegura que todos os controladores possuem a visão mais atualizada possível, depois de uma operação de escrita. Embora tenha impacto no desempenho do sistema, a consistência forte é uma aplicação de simples implementação. O controlador ONOS é um exemplo de controlador com consistência forte [Kreutz et al., 2015]. Atualmente existe um grande número de implementações de controladores SDN disponíveis, incluindo de código aberto ou comerciais. Os de código aberto possuem interfaces *northbound* (NBI) e *southbound* (SBI) abertas, permitindo a pesquisa de métodos inovadores de operação da rede [Rowshanrad et al., 2014]. Adicionalmente à pesquisa e experimentação, as interfaces abertas permitem que equipamentos de fabricantes diferentes possam interoperar. A seguir são enumerados alguns dos principais controladores existentes:

- **VMware/Nicira**: Plataforma de virtualização (*Network Virtualization Plataform* – NVP), atualmente comercializado como VMware NSX, utiliza um comutador virtual aberto (*Open Virtual Switch* – OVS) e o OpenFlow como interface SBI.

- **NOX/POX**: Controlador de código aberto que utiliza o OpenFlow 1.0. A pesquisa continuada pelo ON.LAB (*Open Networking Lab*) deu origem ao controlador ONOS aplicado atualmente na indústria de equipamentos de redes de telecomunicações. O controlador NOX foi desenvolvido em C++ não tendo sido implementado de forma massiva. O controlador POX foi o sucessor do NOX com interface gráfica escrita na linguagem Python, facilitando o desenvolvimento e experimentação.

- **Ryu**: Controlador com implementação em Python. Possui um serviço de mensagens de componentes implementadas em outras linguagens de programação, como por exemplo, bibliotecas do OpenFlow, gerenciamento de aplicações, serviços de infraestrutura e bibliotecas reutilizáveis como do protocolo NETCONF.

- **Beacon**: Controlador implementado em JAVA e integrado à IDE do Eclipse. O Beacon foi o primeiro controlador a criar um ambiente de trabalho de SDN, mas é limitado à topologia em estrela.

- **Big Switch Networks/Floodlight**: é uma ramificação do controlador Beacon. Foi inicialmente implementado usando o Apache Ant, uma ferramenta de desenvolvimento popular, tornando-o de fácil uso e flexível. O controlador Floodlight possui uma comunicadade ativa e um grande número de funcionalidades que podem ser adicionadas para requisitos específicos de uma empresa. Possui uma interface gráfica baseada em JAVA, e a maioria de suas funcionalidades está exposta em APIs REST.

- **Opendaylight**: é um projeto colaborativo da Linux Foundation, suportado pela Cisco e diversas outras empresas. Tal como o Floodlight, o Opendaylight foi escrito em JAVA. É orientado a API REST e interface web. Sua segunda versão (Helium) inclui suporte a NFV e redes de grandes dimensões. Também possui módulos plugáveis que podem ser utilizados de acordo com a necessidade. O Opendaylight é um pouco diferente dos demais controladores por oferecer outros protocolos como interfaces *southbound* além do OpenFlow, como o BGP e PCEP. Adicionalmente, o Opendaylight oferece interfaces com o Openstack e Open vSwitch (OVSDB). O Opendaylight é baseado em uma arquitetura de microsserviços, através do compartilhamento de estruturas de dados baseados em YANG para armazenamento de dados e troca de mensagens. Através de um modelo dirigido à camada de abstração de serviços (*Model Driven Service Abstraction Layer* - MD-SAL) qualquer aplicação ou função pode ser agregada a um serviço e carregada pelo controlador [Haleplidis et al., 2015].

- **ONOS**: controlador SDN voltado para operadoras de telecomunicações, projetado para alta disponibilidade, desempenho e escalabilidade através de instâncias distribuídas que conferem redundância ao controlador em caso de falha.

- **Opencontrail**: Controlador comercial da Juniper, baseado no Apache 2.0, com foco em NFV e com API REST.

Existem ainda outros controladores comerciais. Alguns exemplos são o Brocade Vyatta, o Cisco WAE (*WAN Automation Engine*), ambos baseados em Opendaylight com implementações próprias. O HP *Virtual Application Networks* (VAN) trabalha em conjunto com o Openstack com melhorias no controle do Open vSwitch e suporte a diferentes tipos de hipervisores e roteadores distribuídos [Kreutz et al., 2015].

5.3.3.2. Elemento de Computação – PCE

O PCE (*Path Computation Element*) é uma entidade que calcula caminhos baseado em restrições fornecidas a partir do comportamento dos roteadores, de um OSS (*Operations Support System*), ou ainda de outro PCE da rede. A arquitetura PCE já possui um cálculo de caminho centralizado para grandes redes multidomínio e multicamadas, sendo adequada como ferramenta do SDN. Quando um nó necessita calcular o caminho para um determinado LSP, faz uma requisição ao PCE através do protocolo PCEP (*Path Computation Element Protocol*). O PCE tem acesso às informações de topologia de um domínio inteiro da rede. A arquitetura do PCE está ilustrada na Figura 5.22. Sua principal função é resolver o problema de multidomínios, pois o PCE tem a visão da rede como um todo, construindo uma base de dados através destes múltiplos domínios. A sessão entre o PCC (*Path Computation Client*) e o PCS (*PCE Server*) ou simplesmente PCE, é estabelecida sobre TCP, assim como o BGP. Uma vez a sessão estabelecida, o PCE constrói a base de dados de topologia TED (*Traffic Engineering Database*) usando um IGP como o OSPF ou o IS-IS, ou ainda através do BGP-LS. Este último possui estruturas TLV (*Type-Length-value*) que permitem ao PCE construir a base de dados. Quando o PCC solicita um LSP com certas restrições, o PCE calcula o caminho baseado na base de dados e responde com o caminho apropriado. Essa abordagem centralizada auxilia o cálculo e o provisionamento do caminho aumentando a flexibilidade e permitindo melhor uso dos recursos de rede. O PCE pode ser do tipo sem controle de estados (*Stateless PCE*) e com controle de estados (*Stateful PCE*), como ilustrado na Figura 5.23.

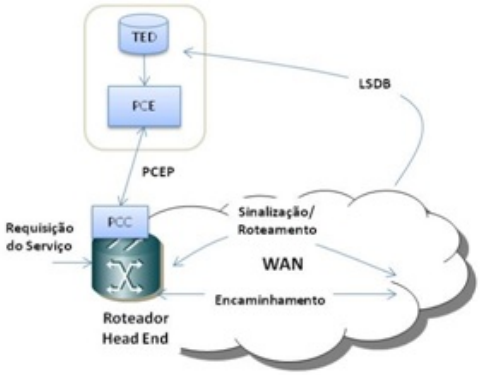


Figura 5.22. Arquitetura do PCE.

O PCE sem controle de estados tem habilidade reduzida para otimizar recursos de rede, não possui conhecimento prévio dos caminhos pré-estabelecidos. Ele é útil entre domínios do MPLS-TE, mas não é adequado para emprego em SDN. Já o PCE com controle de estados tem um controle do cálculo de caminho ótimo (por exemplo, estado

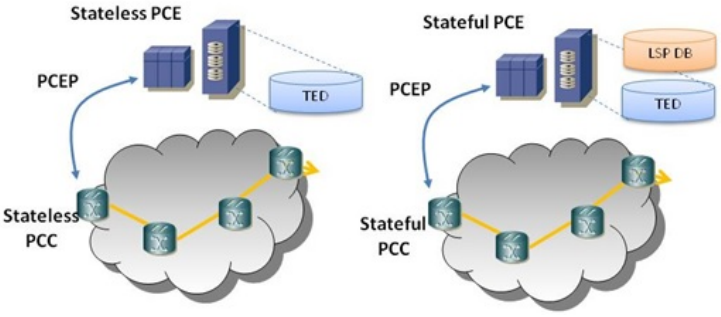


Figura 5.23. Stateless e Stateful PCE.

do LSP, recursos, políticas, análise de redes), possibilita a inicialização de caminhos e atualizações de controle, e requer sincronismo do estados da base de dados dos LSPs. O PCE com controle de estados pode ainda operar em modo passivo ou ativo. No modo passivo, o PCC inicia a configuração do caminho, atualizando as informações de controle, e o PCE aprende o estado do LSP para otimização do cálculo do caminho. No modo ativo, tanto o PCC quanto o PCE podem iniciar a configuração do caminho, mas a atualização do controle é feita pelo PCE, delegada pelo PCC. No modo ativo, o LSP pode ser iniciado pelo PCE, sendo mais integrado às demandas das aplicações, e o PCE pode fazer parte do controlador SDN determinando quando e quais caminhos serão configurados. No modo ativo, o LSP pode ser iniciado no PCC baseado nos estados que estão distribuídos na rede e pode ser usado em conjunto com os LSPs iniciados pelo PCE, sendo uma abordagem de rede híbrida com controle IP/MPLS e SDN [Paolucci et al., 2013]. A base de dados utilizada pelo PCE pode ser construída através de multidomínios, basicamente por quatro métodos distintos: cálculo de caminhos por domínio, cooperação entre PCEs, cálculo de caminhos de forma recursiva ou PCE hierárquico utilizado em multicamada IP/óptica.

Uma proposta de SDN para o PCE é uma solução ideal para operadoras de telecomunicações, desacoplando o plano de controle do plano de encaminhamento de dados, e com mecanismos de controle centralizado que controlam a configuração dos caminhos. Empregar o PCE com políticas apropriadas provenientes do OSS (*Operational Support Systems*), é a forma mais rápida e fácil para introduzir a SDN nas redes IP/MPLS e de transporte da operadora de telecomunicações. Com o emprego de PCEs, as redes de roteadores de núcleo têm um ganho substancial em benefícios de roteamento intra-domínios. Entende-se por domínio diferentes áreas de sistemas autônomos e também diferentes redes, como a rede IP e a de transporte óptica. Em redes legadas o PCE é implementado em servidores do sistema de gerência da rede, individualmente para cada domínio. A cooperação entre PCEs de diferentes domínios é uma questão de difícil solução técnica, e o SDN desponta como uma solução para o PCE multidomínios [Farrel, 2006].

5.3.3.3. Interfaces Northbound: REST, RESTFUL, NETCONF e outros

As interfaces *northbound* ainda são objeto de esforço de padronização e pesquisa para que garanta a interoperabilidade entre diferentes plataformas de controle. Alguns controladores como o Opendaylight e FloodLight propõem suas próprias NBIs. A API

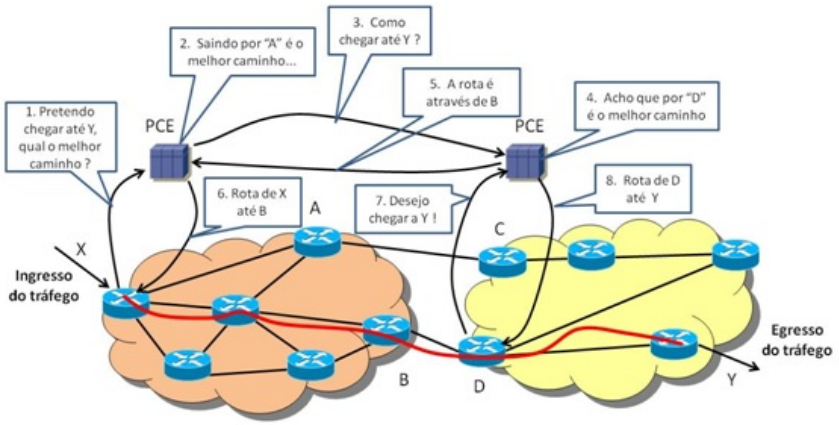


Figura 5.24. Exemplo de PCE intradomínios.

REST utiliza HTTP ou HTTPS, tendo sido desenvolvida inicialmente para acessar informações de serviços web. O uso do REST em SDN se deve a questões de simplicidade, flexibilidade, extensibilidade e segurança. Para acesso aos dados que envolvem recursos nos dispositivos de destino é simplesmente utilizada uma URL. A flexibilidade é consequência das entidades requisitantes poderem acessar os componentes de configuração definidas em um dispositivo usando os recursos REST em URLs diferentes. Não existem esquemas complexos ou MIBs para tornar o dado acessível. Já a extensibilidade do REST é devida ao suporte a novos recursos que não requerem a recompilação de esquemas ou MIBs, mas somente a chamada da URL apropriada pela aplicação. Por fim, usando o protocolo HTTPS, os aspectos de segurança são considerados e permite-se atravessar firewalls. A flexibilidade e extensibilidade podem representar um risco potencial do uso do REST pela falta de formalismo comparado a outros métodos. No entanto, a relação custo-benefício tende a indicar o REST como uma API propícia para utilização em SDN. A API REST pode trabalhar com diferentes formatos de arquivos como XML e JSON.

A NBI RESTCONF [Bierman et al., 2015] é um protocolo similar ao REST que roda sobre HTTP para acessar dados definidos em YANG e repositórios de dados em NETCONF. A RESTCONF descreve como mapear YANG em uma interface RESTful. A RESTCONF opera com repositórios de dados conceituais com a modelagem de dados YANG. O servidor lista cada módulo YANG suportado em um tipo de recurso da API de alto nível usando estruturas baseadas em módulos YANG habilitadas com URI. A RESTCONF pode enviar dados de requisição e resposta no formato JSON ou XML. O NETCONF possibilita a passagem de dados e comandos de e para dispositivos de rede, podendo ser usado interface SBI e NBI.

5.3.4. Plano de Gerenciamento

O plano de gerenciamento corresponde às aplicações utilizadas em Redes Definidas por Software e às linguagens de programação usadas para implementar estas aplicações. A subseção a seguir enumera algumas aplicações denominadas casos de uso em Redes Definidas por Software.

5.3.4.1. Aplicações de Rede

Entre as aplicações de rede destacam-se balanceadores de carga, lista de acesso para segurança, detecção de ataques, monitoramento da rede, virtualização da rede e roteamento. Neste último, se encaixa o Roteamento por Segmentos. A implementação da lógica de controle pode ser traduzida em comandos e instalados no plano de encaminhamento de dados ditando o comportamento dos dispositivos de encaminhamento.

- Engenharia de Tráfego: o principal objetivo desta aplicação é minimizar consumo de energia, maximizar a utilização agregada da rede, balanceamento de carga e outras técnicas de otimização de tráfego.
- Mobilidade e Redes Sem-fio: atualmente o plano de controle distribuído de redes sem-fio é subótimo face ao gerenciamento do espectro limitado, alocação de recursos de rádio, implementação de mecanismos de *handover* e balaceamento de tráfego entre células.
- Medição e Monitoramento: um exemplo de aplicações de medição é a melhoria da visibilidade do desempenho fim a fim da rede. As aplicações de monitoramento envolvem diferentes técnicas de amostragem e estimativas a serem aplicadas, reduzindo a interferência desnecessária do plano de controle nos dispositivos de encaminhamento, com o objetivo de coletar informações e calcular estatísticas do plano de dados.
- Segurança e Dependência: um conjunto de propostas diversas de segurança e dependência está emergindo no contexto de SDN. As vantagens do SDN para melhoria dos serviços depende de redes e sistemas seguros, como execução de políticas (controle de acesso,*firewall*, *middlebox*, a detecção e mitigação de ataques do tipo negação de serviço (*Denial of Service* - DoS), inspeção pormenorizada de pacotes (*Deep Packet Inspection* - DPI) e detecção de anomalias de tráfego. Existem basicamente duas abordagens de segurança: uma utilizando o SDN para prover segurança como serviço de valor adicionado, e outra, para provimento de segurança na própria arquitetura da SDN.
- Redes de Datacenter: as redes de centros de dados são beneficiadas de forma significativa solucionando problemas como migração de rede viva, gerência de rede avançada, implantação rápida do planejamento de redes para redes em produção. As aplicações em SDN caminham para um conceito de loja de aplicativos (SDN App Store), este conceito lançado pela HP, o controlador HP utiliza OpenFlow para acessar aplicações on-line e selecioná-las para serem dinamicamente descarregadas e instaladas no controlador. As linguagens de programação permitem prover um grande conjunto de poderosas abstrações [Casado et al., 2014] e mecanismos como composição de aplicações, tolerância a falhas no plano de dados e blocos de construção básicos. Um exemplo de linguagem de programação é o Pyretic que oferece uma abstração de alto nível da topologia e do conjunto de políticas da rede.

Ainda na visão simplificada da arquitetura de rede SDN, a plataforma de controle se comunica com as aplicações de rede através da interface *northbound*, que oferece uma API

para desenvolvedores de aplicação, como a API REST, RESTCONF, Restful e linguagens de programação com Java, Python, C e C++ [Hodzic e Zoric, 2008].

5.3.4.2. Roteamento como serviço: Routeflow e RCP

A utilização de SDNs para decisão de encaminhamento [Adrian et al., 2015], como o Roteamento por Segmentos, é uma tendência como por exemplo o RCP e RouteFlow. O RCP (*Routing Control Plataform*) é uma alternativa primitiva de roteamento como serviço [Feamster et al., 2004]. O servidor RCP divulga as rotas aos roteadores usando protocolos existentes como o BGP. A visão da rede é obtida pelo IGP. O RPC opera de forma distribuída, dividindo a o projeto em componentes, visualizando o estado global da rede por componente. O RCP possui objetivos similares aos das redes SDN, tais como melhor escalabilidade, gerenciamento e separação do software do hardware do roteador. O projeto RouteFlow é considerado um caso de uso de implementação de uma rede de teste para roteamento como serviço (*IP Routing-as-a-Service*) em código aberto, demonstrando a migração de uma rede tradicional de camada 3 para uma rede OpenFlow de camada 3. Os equipamentos roteadores legados interoperam com implementações simplificadas de roteamento intra e interdomínio. O projeto RouteFlow [Rothenberg et al., 2011] foi particularmente importante pois demonstrou um método de integração de redes tradicionais com inúmeros protocolos de rede com redes definidas por software baseada em OpenFlow [Jingjing et al., 2014]. Os conceitos de SDN são importantes para a compreensão do roteamento por segmentos, visto que o roteamento por segmentos é uma aplicação SDN.

5.4. Roteamento por Segmentos

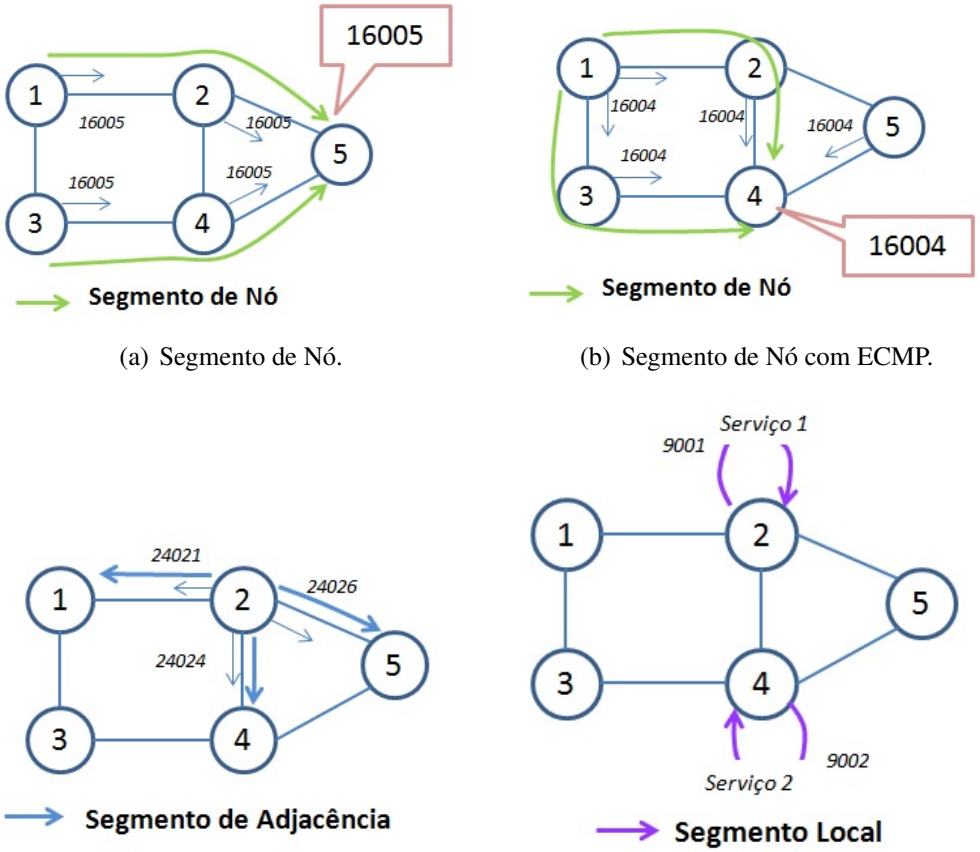
O Roteamento por Segmentos ou SR (*Segment Routing*) é um conceito que pode ser entendido como um protocolo de roteamento suportado por aplicações de Redes Definidas por Software para a definição de caminhos de forma eficiente e automatizada. O Roteamento por Segmentos melhora o encaminhamento de pacotes sem necessitar da manutenção de estados por fluxo dentro da rede de roteadores de núcleo, reduzindo a complexidade dos planos de controle e de dados. A engenharia de tráfego é um exemplo de aplicação no contexto de Roteamento por Segmentos. O Roteamento por Segmentos permite a configuração, a modificação e a remoção de caminhos TE dentro de um domínio de rede, operando somente na borda da rede. O plano de controle de Roteamento por Segmentos pode ser mantido de forma centralizada ou distribuída.

5.4.1. Conceitos preliminares

O Roteamento por Segmentos define um protocolo de roteamento pela origem, ou seja, a origem do fluxo de dados escolhe e codifica no cabeçalho do pacote a lista de segmentos a serem percorridos pelos pacotes até o destino. O segmento, por sua vez, é um identificador genérico para qualquer tipo de instrução: um serviço, contexto, localizador ou um caminho baseado no IGP ou no BGP. Ainda, um segmento pode ser representado por um índice local ou global. Por exemplo, uma lista de segmentos para a rede de roteadores de núcleo IP/MPLS é representada por uma pilha de rótulos (*Label Stack*), enquanto no IPv6 é representada pela extensão do cabeçalho para rotea-

mento [Previdi et al., 2015b]. O identificador de segmento denomina-se SID (*Segment Routing Identifier*), e em redes MPLS, o SID é um rótulo MPLS. A cadeia de SIDs é chamada de caminho de Roteamento por Segmentos (*Segment Routing Path - SR path*).

Os segmentos podem ser divididos em Segmentos de Nó, Segmentos de Adjacência e Local. A Figura 5.25 mostra o identificador de Segmento de Nó, de Adjacência e Local. O Segmento de Nó tem uma numeração única e global denominada prefixo SID retirado do SRGB (*Segment Routing Global Block*) [SPRING, 2015] dentro do domínio do IGP, que também corresponde a um domínio do Roteamento por Segmentos. Os Segmentos de Nó estão contidos em uma numeração global de segmentos. A numeração global de segmentos, em redes MPLS, é formada por uma faixa de rótulos reservada, somada a um índice referente ao nó (prefixo), compondo a numeração do Segmento de Nó. O Segmento de Adjacência tem significado local e está relacionado a uma ou mais adjacências do nó. Os Segmentos de Adjacência têm rótulo no formato 240xy para uma determinada adjacência xy. Os rótulos de 90000 a 99999 são usados pelas extensões dos protocolos LDP, RSVP e BGP. O Segmento Local corresponde ao segmento suportado apenas no nó que foi gerado. Assim, nenhum outro nó pode instalar este SID em sua FIB.



(a) Segmento de Nó.

(b) Segmento de Nó com ECMP.

(c) Segmento de Adjacência.

(d) Segmento Local.

Figura 5.25. Segmento de Nó, Adjacências e Local.

Na Figura 5.25(a), o rótulo 16005 corresponde ao Segmento de Nó 5. O identificador do segmento é composto pelo SRGB no valor de 16000 a ser detalhado na Subseção 5.4.3 e mais um valor de um prefixo do nó, neste caso 5. Nota-se que o SID 16005

é o mesmo para qualquer nó de origem, a partir de qualquer nó de origem utiliza-se o Segmento de Nó 16005 para chegar ao nó 5 de destino e a divulgação do SID é feita pelo IGP para todos os nós. Na Figura 5.25(b), existem dois caminhos de mesmo custo entre o nó 1 e 4, obtidos pelo IGP. A escolha do caminho depende de outras restrições como latência e banda, obtida por uma aplicação SDN que possua visão completa da rede. A Figura 5.25(c) ilustra os Segmentos de Adjacências do nó 2 para o nó 1, 4 e 5 com SIDs 24021, 24024 e 24025, respectivamente, e representa uma identificação para cada enlace diretamente conectado ao nó. Neste exemplo, a numeração escolhida para os Segmentos de Adjacência foi 240xy (faixa de identificador de segmento fora do SRGB que identifica o Segmento de Nó conforme descrito na Subseção 5.4.3). Para cada Segmento de Adjacência foi utilizado o identificador xy, onde xy é uma adjacência que identifica o enlace entre os nós x e y. A Figura 5.25(d) exemplifica um Segmento Local no nó 2 de SID 9001, indicando um serviço existente no nó 2. Em particular para o Segmento Local, nenhum outro nó pode configurar este SID na SR-FIB (*Segment Routing-Fowarding Information Base*) sob pena de conflito de SIDs. Os segmentos de Nó e de Adjacência podem ser combinados, dirigindo o tráfego através de qualquer caminho na rede.

O caminho é especificado como uma lista de segmentos no cabeçalho do pacote através de um empilhamento de rótulos MPLS (Figura 5.26). Não existem estados por fluxo nos roteadores intermediários, utilizando-se apenas o IGP, e consequentemente, dispensando o protocolo de distribuição de rótulos (LDP) de uma rede MPLS convencional. Os estados por fluxo são mantidos apenas na origem, pois o roteador de origem conhece todos os SIDs para alcançar o destino. A quantidade de entradas na tabela SR-FIB em um determinado nó é da ordem de $N + A$, onde N é o é numero de Segmentos de Nó e A é o número de Segmentos de Adjacências, como observada na Figura 5.26. Ainda na Figura 5.26, a informação do Segmento de Nó 4 com SID de 16004 é anunciada aos demais roteadores pelo IGP, correlacionando o SID com a interface loopback do nó 4. O Roteamento por Segmentos possui operações similares ao das redes IP/MPLS. O segmento ativo é definido como o segmento que vai aplicar a instrução corrente no pacote. Nas redes MPLS, o SID do segmento ativo é o rótulo mais externo; enquanto no IPv6, é um ponteiro para o SID. No exemplo da Figura 5.26, na interface de egresso do nó 1 em direção ao nó 2, o segmento ativo é o com SID 16004, que é o Segmento de Nó que vai encaminhar o pacote até o nó 4, compondo uma parte do caminho. As operações que podem ser efetuadas em Roteamento por Segmentos são as seguintes:

- `PUSH`: Inserção em uma lista de segmentos. Para redes MPLS, significa inserção do rótulo na pilha; enquanto para o IPv6, significa inserir o SID na primeira posição e redirecionar o ponteiro o topo da lista.

- `NEXT`: Ativação do próximo segmento da lista, uma vez que o atual está completo. Nas redes MPLS, significa remover o rótulo mais externo; enquanto para o IPv6 significa incrementar o ponteiro.

- `CONTINUE`: Continuação do atual segmento ativo, mesmo sem estar completo. Nas redes MPLS, há correspondência com o processo de comutação de rótulos. Logo, se o próximo salto está no mesmo SRGB, o valor do rótulo é mantido. No IPv6, significa não incrementar o ponteiro.

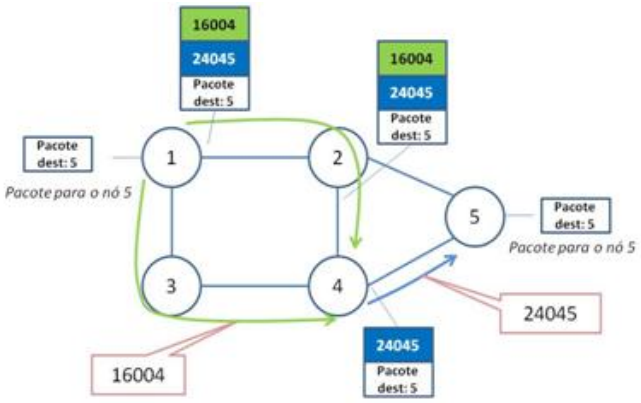


Figura 5.26. Combinação de Segmentos de Nó e de Adjacências.

A Figura 5.27 mostra as operações `PUSH` (Insere), `NEXT` (Próximo) e `CONTINUE` (Continue) implementadas no Roteamento por Segmentos referente ao exemplo da Figura 5.26. O segmento ativo é aquele cujo SID corresponde ao rótulo mais externo, neste caso nos enlaces 1-2, 2-3 o SID ativo é do Segmento de Nó global 16004. Nota-se que no nó 2, o rótulo passa por um processo de comutação MPLS, sem no entanto, alterar o SID. No nó 4, o rótulo 16004 é removido e o segmento ativo passa a ser o Segmento de Adjacência com SID 24045. Finalmente, no nó 5 é feita a remoção do rótulo atual mais externo com SID 24045, entregando o pacote à rede de destino.

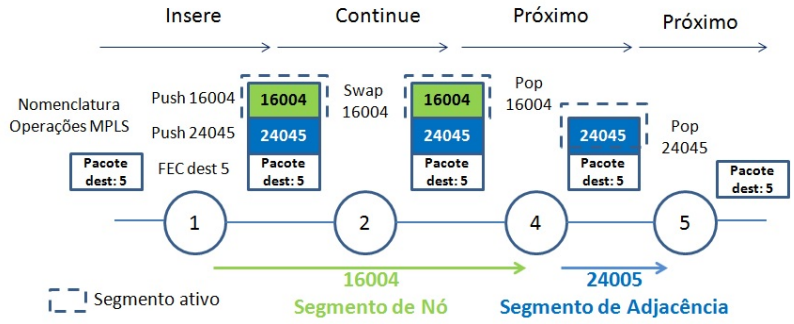


Figura 5.27. Operações em Roteamento por Segmentos.

Uma boa prática para a Operadora de Telecomunicações é alocar o mesmo SRGB em um domínio de Roteamento por Segmentos (*SR-Domain*). Um SR-Domain é um conjunto de nós conectados em uma infraestrutura física correspondente a uma rede de Operadora de Telecomunicações, e confinada dentro de uma instância do IGP, chamada SR-IGP (*Segment Routing IGP*). Por exemplo, uma rede típica de 500 nós (500 Segmentos de Adjacência) com 5000 segmentos de nó globais (pertencentes ao SRGB), para um determinado fluxo f, somente o nó de ingresso do tráfego detém os estados para f, ou seja 5500 entradas na SR-FIB. Mesmo que se tenha $n \times f$ fluxos, a quantidade de entradas na SR-FIB é a mesma. Outra facilidade do Roteamento por Segmentos é o agrupamento (*bundle*) através de Segmentos de Adjacência. Os Segmentos de Adjacências permitem

agrupar múltiplos enlaces físicos, permitindo balanceamento de tráfego, ou configurar o SID para um enlace específico da adjacência. Por exemplo, na Figura 5.28, os Segmentos de Adjacências 24045, 24145 são configurados em enlaces físicos distintos, na mesma adjacência, enquanto o SID 24245 representa um grupo (*bundle*) de enlaces físicos, permitindo balancear o tráfego de forma simples.

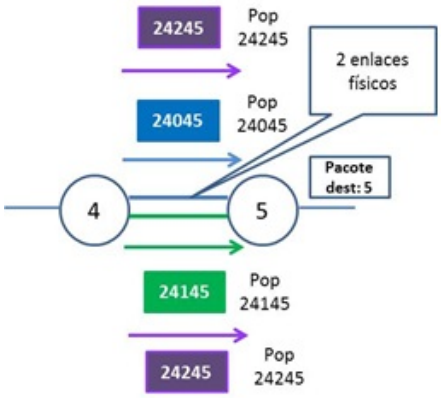


Figura 5.28. Segmentos de Adjacência (*bundle*).

O Roteamento por Segmentos tem foco no plano de encaminhamento de dados e a inteligência da escolha do melhor caminho é realizada pela Rede Definida por Software. O Roteamento por Segmentos simplifica a operação da rede da Operadora de Telecomunicações, reduzindo a quantidade de protocolos de controle e estados nos roteadores intermediários do núcleo da rede e, consequentemente, permitindo a criação de serviços baseado em aplicações através do SDN. O roteamento aumenta a capacidade da rede, aproveitando melhor a infraestrutura existente e evitando a duplicação de túneis TE para salvamento do tráfego. Comutando em sub-50ms, o uso do Roteamento por Segmentos em redes IP/MPLS permite uma vasta gama de aplicações para Operadoras de Telecomunicações e Operadoras OTT (*Over The Top*) através de redes de acesso banda larga e centro de dados. A Tabela 5.1 mostra as diferenças entre a arquitetura IP/MPLS e a arquitetura das redes que utilizam Roteamento por Segmentos (*SR-based MPLS*), destacando as vantagens da arquitetura.

5.4.2. Plano de dados: MPLS e IPv6

Em redes MPLS, o Roteamento por Segmentos utiliza o plano de encaminhamento de dados do MPLS, onde o segmento é representado por um rótulo; e a lista de segmentos, é representado pelo empilhamento de rótulos. O Roteamento por Segmentos possui as funcionalidades de PHP (*Penultimate Hop Popping*) e rótulos *explicit-null*. A imposição do rótulo representa o prefixo do SID e tem preferência em casos do prefixo de destino não possuir um rótulo associado através do protocolo LDP. O encaminhamento do pacote e as operações de Roteamento por Segmentos são ilustrados na Figura 5.29. O nó 4 utiliza prefixo IPv4 ou IPv6 de sua interface loopback:1.1.1.4/32, com prefixo de SID associado 16004, assumindo que o protocolo LDP não esteja habilitado. O nó 4 requisita a funcionalidade PHP por padrão. Esse prefixo é uma entrada na FIB do roteador remoto 1 e a operação executada é a inserção do rótulo (*push*). O prefixo SID também é uma

Tabela 5.1. MPLS com Roteamento por Segmentos *vs.* MPLS tradicional.

Característica	Rede MPLS com SR	Rede MPLS tradicional
Transporte MPLS básico	IGP	IGP+LDP
Sincronismo entre o LDP/IGP	Não se aplica	Difícil de gerenciar
Comutação FRR em 50ms	IGP	IGP+LDP
Túneis TE adicionais para suportar FRR	Não precisa	Precisa
Otimização de caminho de backup	Sim	Não
ECMP para criação de túneis TE	Sim	Não
Estados apenas no *head-end* do TE	Sim	Não, dependendo do número de nós (complexidade $O(n^2)$ nos pontos intermediários)
Interoperabilidade com a rede MPLS tradicional	Sim	Não se aplica
Projetado para SDN	Sim	Não

entrada da LFIB do roteador 2, com a operação de comutação do rótulo (*swap*). No roteador 3, o prefixo SID é uma entrada remota de sua LFIB. Como o roteador 3 é o penúltimo salto (PHP), então a operação executada sob o rótulo é de remoção (*pop*). No roteador 4, o pacote chega sem o rótulo baseado no endereço IP ou o rótulo do serviço (Segmento Local).

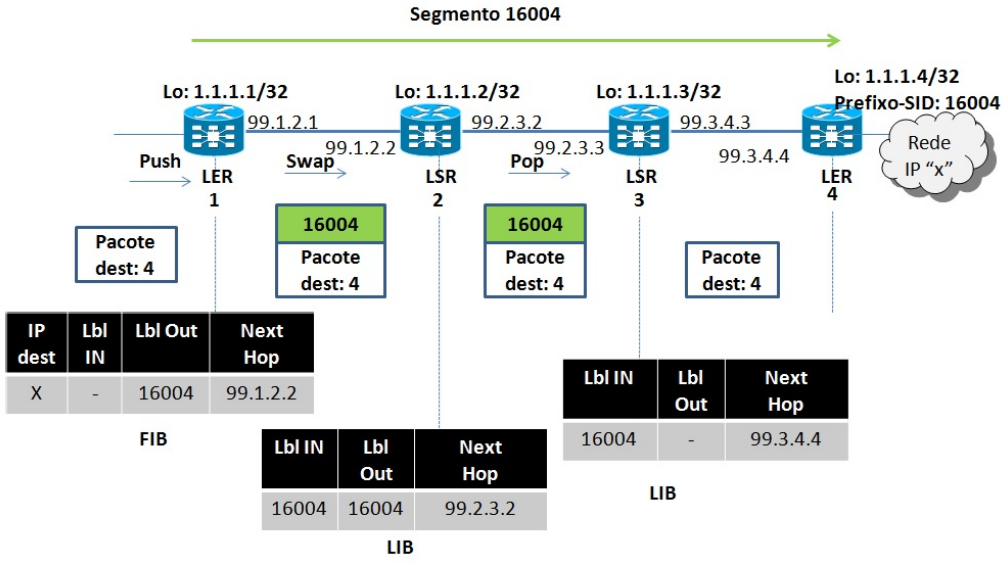


Figura 5.29. Roteamento de Segmentos - Plano de dados MPLS.

A opção de *explicit-null* é configurável para um prefixo SID, sendo que o vizinho do roteador de origem do SID comuta o prefixo SID mais externo com o rótulo *explicit-null*. Na Figura 5.29, a LIB do roteador 3, o rótulo de saída (*Label Out*) seria *explicit-null*, sendo útil para repassar os EXP bits do LSR 3 para o LER 4.

O plano de dados MPLS em Roteamento por Segmentos é o mesmo das redes IP/MPLS legadas. Para serviços VPN L3, a rede MPLS se torna um serviço de transporte

baseado nos prefixos de segmentos. A Figura 5.30 ilustra um serviço VPN L3 sobre uma rede implementada com Roteamento por Segmentos. Os prefixos SID correspondem aos Segmentos de Nó, que são SIDs globais. Nota-se que para os nós adjacentes ao nó 3, os SID globais são removidos. Para o nó 6 existem duas possibilidades de caminho de custo igual (ECMP) com mesmo SID global. Do nó 3 para seus vizinhos (nó 1 e 4) são configurados Segmentos de Adjacência com significado local, sendo removidos.

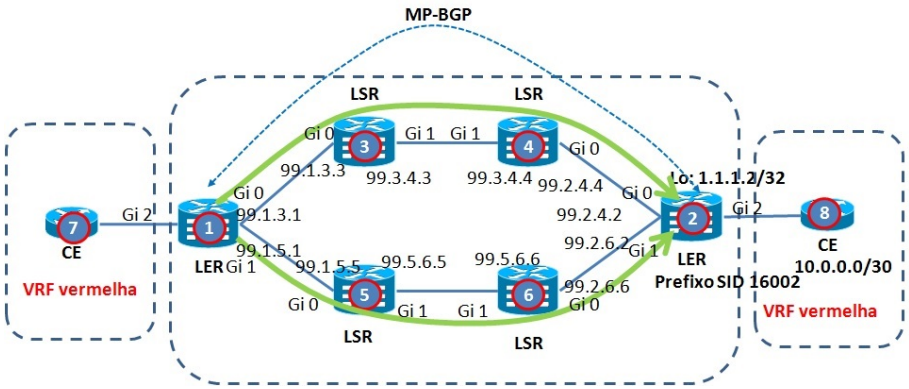


Figura 5.30. VPN L3 com Roteamento por Segmentos.

Além das redes MPLS, o Roteamento por Segmentos pode ser aplicado no plano de dados IPv6, onde a lista de segmentos está codificada na extensão do cabeçalho para roteamento pela fonte. A lista de segmentos pode ser baseada no IGP ou no BGP. A divulgação dos segmentos na rede com IPv6 é feita a partir de extensões dos protocolos IGP, BGP, BGP-LS e PCEP. O Roteamento por Segmentos em IPv6 não necessita de atualização de toda a rede, permitindo interoperar com e sem Roteamento por Segmentos. Essa característica possibilita a adoção gradual da tecnologia. O caminho é expresso de forma explícita, onde os nós representam roteadores, servidores, instâncias de aplicações, serviços, cadeias de serviços, etc. O roteamento em segmentos em IPv6 é um roteamento pela origem não-estrito.

Figura 5.31. Cabeçalho de Roteamento por Segmentos SRH do IPv6.

No cabeçalho do IPv6 (*Segment Routing Header* - SRH) visto na Figura 5.31, os campos de "Segment List", descrevem o caminho do pacote. O segmento é representado

por um endereço IPv6. A seguir são detalhados os campos do cabeçalho de Roteamento por Segmentos:

- `Next Header`: é um seletor de 8 bits que identifica o tipo de cabeçalho imediatamente seguido pelo SRH.
- `Hdr Ext Len`: define o tamanho do cabeçalho de SRH em octectos, descontando os primeiro oito octetos.
- `Routing Type`: ainda depende de definição pelo IETF.
- `Segments Left`: contém o índice da lista de segmentos, indicando o próximo a ser inspecionado. É decrementado a cada inspeção.
- `First Segment`: é um "offset" no SRH, não incluindo os 8 primeiros octetos. É expresso em múltiplos de 16 octetos apontando para o último elemento da lista de segmentos. Representa o primeiro segmento da lista.
- `Flags`: 16 bits para flags. Os bits de 4 a 15 definem o tipo de codificação dos endereços IPv6 na lista de políticas (Policy List).
- `Segment List[n]`: é o endereço IPv6 que representa cada segmento do caminho. A lista de segmentos é codificada de forma reversa, ou seja, o último segmento é o primeiro da lista.
- `Policy List[n]`: são endereços que representam nós específicos no SR-Path: "Ingress SR PE" (nó que insere o cabeçalho SRH) e "Egress SR PE" (endereço do nó de egresso do domínio de Roteamento por Segmentos).
- `HMAC`: autenticação SRH, ainda em versão "draft" pelo IETF.

O SRH é um novo tipo em um cabeçalho de roteamento existente no IPv6, idêntico ao RH0, que tornou-se obsoleto por questões de segurança. O SRH utiliza o HMAC como solução de segurança usado no ingresso de um domínio de Roteamento por Segmentos segundo o "draft" do IETF *draft-vyncke-6man-segment-routing-security*. Dentro de um domínio controlado de Roteamento por Segmentos, o HMAC não é necessário. No Roteamento por Segmentos do IPv6, o segmento ativo é aquele que está designado como endereço MAC de destino no pacote. Em cada ponto final do segmento (*endpoint*), o endereço MAC de destino é atualizado com o próximo segmento ativo na lista de segmentos conforme a Figura 5.32. Na topologia apresentada na mesma figura, o nó A é chamado nó de Roteamento por Segmentos habilitado (*SR Capable*), capaz de criar a lista de segmentos ou recebê-la de um controlador SDN. Os nós intermediários do caminho são chamados de nós de trânsito (*Transit Nodes*), que podem não executar o Roteamento por Segmentos, nós com Roteamento por Segmentos intra-segmento (*Intra-Segment Nodes*) e nós finais (*Endpoint Nodes*). Os nós B e G são nós de trânsito sem Roteamento por Segmentos, fazendo o encaminhamento dos pacotes baseados no endereço IPv6 sem inspecionar o SRH. Se ocorresse a inspeção do SRH, esses nós seriam chamados de nós intra-segmentos. Os nós C, F e H são nós finais do Roteamento por Segmentos, pois inspecionam o SRH, e também são nós de Roteamento por Segmentos habilitados.

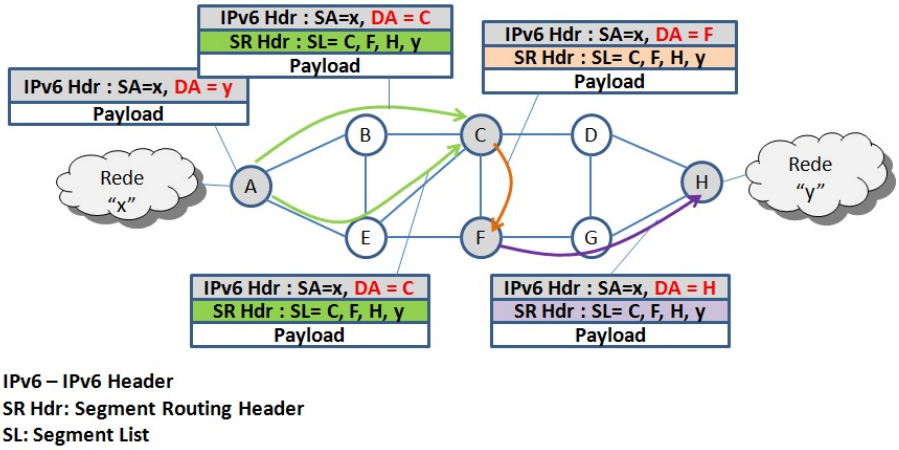
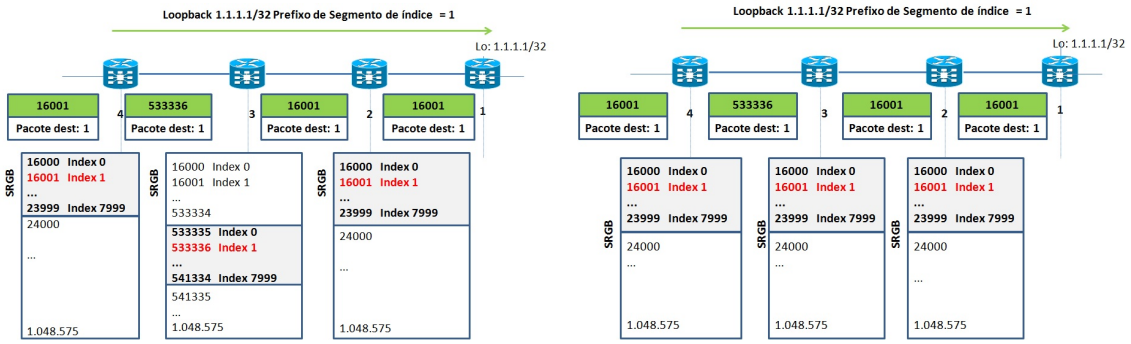


Figura 5.32. Exemplo de plano de dados IPv6 para Roteamento de segmentos.

5.4.3. Bloco Global de Roteamento por Segmentos (SRGB)

O Bloco Global de Roteamento por Segmentos (*Segment Routing Global Block* - SRGB) é uma faixa de rótulos reservada para os Segmentos de Nó ou globais, pois tem valor absoluto em um domínio de Roteamento por Segmentos. O prefixo SID é anunciado com um índice único em um domínio de Roteamento por Segmentos. O primeiro prefixo começa em zero, e o rótulo é formado pelo prefixo do SID somado à base do SRGB, representando o Segmento do Nó. Por exemplo, um roteador com interface loopback 1.1.1.65/32 tem prefixo SID 65 com rótulo 16065. Uma boa prática é usar o mesmo SRGB em todos os nós do mesmo domínio de Roteamento por Segmentos, pois diferentes SRGBs podem complicar a solução de problemas na rede. O SRGB não padrão pode ser alocado com rótulos entre 16000 a 1048575 ou até onde o roteador permitir, sendo o tamanho máximo de 64 kBytes. A Figura 5.33(a) mostra uma possível alocação de SRGB não recomendada, com SRGBs diferentes. Os SRGBs diferentes podem ter conflito com rótulos distribuídos pelo LDP em uma rede mista com Roteamento por Segmentos e IP/MPLS legada (Subseção 5.4.5). A Figura 5.33(b) mostra a alocação recomendada com o mesmo SRGB, o que simplifica a programação na rede SDN.



(a) Não recomendada. (b) Recomendada.

Figura 5.33. Alocação de SRGBs.

Os rótulos que representam os SIDs globais e locais são gerenciados por uma base

de comutação de rótulos (*Label Switching Database* - LSD) que aloca dinamicamente rótulos para as aplicações do MPLS, tais como os protocolos LDP, RSVP, BGP, TE, redes virtuais privadas de camada 2 e os Segmentos de Adjacência obtidos pelo IGP. O LSD preserva a faixa de rótulos do SRGB (de 16000 a 23999) e aloca dinamicamente os rótulos a partir de 24000. O LSD pode alocar rótulos dinamicamente do SRGB em situações de emergência, quando a faixa de rótulos dinâmicos foi consumida, ou se a faixa reservada para o SRGB não for usada. A LSD permite que futuras ativações de Roteamento por Segmentos em roteadores de núcleo não necessitem de um "reboot", coexistindo com rótulos já alocados pela rede IP/MPLS legada. No primeiro momento da ativação do LSD, este aguarda que o IGP solicite o SRGB, assim o LSD aloca a faixa de rótulos do SRGB, e o IGP já pode usá-lo. A Tabela 5.2 mostra a alocação de rótulos para o MPLS e para o Roteamento por Segmentos.

Tabela 5.2. Faixa de rótulos do MPLS e do Roteamento por Segmentos.

Uso	Faixa de Rótulos
Reservada para uso especial	0 a 15
Reservada para rótulos MPLS estáticos	16 a 1599
Reservada para Roteamento por Segmentos	16000 a 23999
Reservada para roteamento dinâmico	24000 a 1048575

5.4.4. Plano de controle IGP do Roteamento por Segmentos: IS-IS e OSPF

O plano de controle IGP é responsável pela configuração e distribuição dos segmentos, aplicando os segmentos em redes multi-área (OSPF) e multinível (IS-IS) e verificando os anúncios de rotas, através de extensões desses protocolos. O IS-IS suporta o plano de controle IPv4 e IPv6, com roteamento considerando múltiplos níveis de rede (nível 1 e 2 do IS-IS). Utiliza o prefixo do SID para representar as interfaces loopback dos roteadores em IPv4 e IPv6 e utiliza os Segmentos de Adjacência para identificar as adjacências dos nós. O anúncio do prefixo para o SID é feito pelo servidor de mapeamento (*Mapping Server*). O Roteamento por Segmentos com IS-IS torna possível a introdução do suporte de sub-TLVs (*Type-length-value*) em extensões do protocolo IS-IS [Previdi et al., 2015a]. Para o OSPF, a versão que suporta as extensões para Roteamento por Segmentos é o OSPFv2, com multi-área, no qual o prefixo de SID representa as interfaces loopback dos roteadores em IPv4 e os Segmentos de Adjacência identificam as adjacências do nó conforme o "draft" do IETF *draft-ietf-ospf-segment-routing-extensions-05*. As extensões do OSPF para Roteamento por Segmentos adicionam anúncios de estado de enlace opaco (*Opaque LSA*), que permitem a transmissão arbitrária de dados que o OSPF não necessariamente utilize. Os anúncios de estados de enlace opaco adicionados oferecem suporte ao Roteamento por Segmentos, permitindo o envio de informações como o algoritmo usado no roteamento e as faixas de rótulos (*Opaque LSA Type 4*), SID de Segmentos de Nó (*Opaque LSA Type 7*) e SID de Segmentos de Adjacência (*Opaque LSA Type 8*). O prefixo do SID do Segmento do Nó usa a informação do SRGB, que á anunciado pelos LSAs opacos. O SRGB pode ser o padrão que utiliza rótulos MPLS de 16000 a 23999 ou algum outro que utilize rótulos fora do padrão. O SRGB escolhido pode ser configurado em cada instância do IGP, sendo que, as diferentes instâncias podem usar SRGB iguais (*Overlapping SRGB*) ou diferentes (*Non-overlapping SRGBs*).

O Prefixo do SID pode ser configurado como um valor absoluto ou índice, sendo que o índice representa um incremento na base do SRGB. Por exemplo, considerando um prefixo com índice igual a 1 e um SRGB igual a 16000, então o SID é $16000 + 1 = 16001$. Esse valor de SID representa o rótulo, tem valor global e é único em um SR-Domain. O Prefix SID é configurado manualmente e equivale a atribuir um endereço a uma interface loopback do roteador de núcleo. Por exemplo, o roteador originador pode anunciar as faixas de rótulos $[100, 199]$, $[1000, 1099]$ e $[500, 599]$ fora do SGRB padrão. Nesse caso, os roteadores de destino que receberem essas faixas, as concatenam formando a SRGB $\{[100, 199], [500, 599], [1000, 1099]\}$. Dessa forma, os índices utilizam várias faixas, como por exemplo: Índice 0 = Rótulo 100, Índice 1 = Rótulo 101, ..., Índice 99 = Rótulo 199, Índice 200 = Rótulo 500, Índice 201 = Rótulo 501, ..., Índice 299 = Rótulo 599, Índice 300 = Rótulo 1000, Índice 301 = Rótulo 1001, ..., Índice 399 = Rótulo 1099.

O Segmento de Adjacência tem significado local e é automaticamente alocado para cada adjacência, sempre sendo um valor absoluto não indexado. As extensões do OSPF possuem TLVs e sub-TLVs com flags para o anúncio e propagação dos prefixos. A Figura 5.34 ilustra as extensões do OSPF e respectivos flags para suporte aos SIDs.

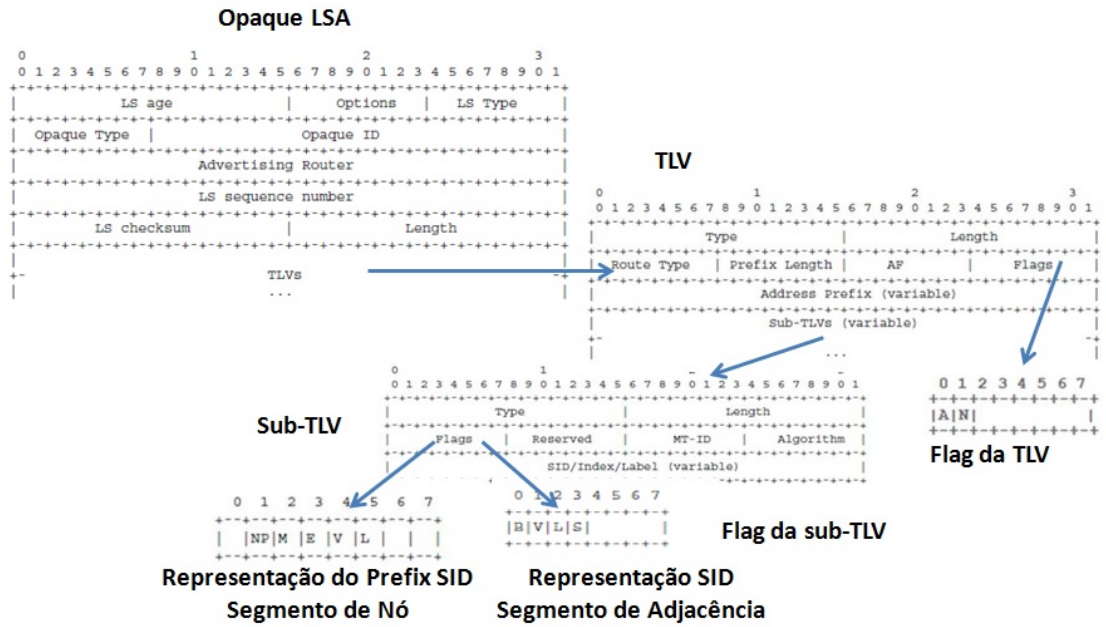


Figura 5.34. Extensões do OSPF para prefixos SID.

O campo de flags da TLV é composto por dois bits, sendo que o bit A sinaliza o uso de uma extensão TLV para prefixo SID inter-área e o bit N indica que o prefixo identifica o nó. No campo de flags da sub-TLV, o bit NP configura o PHP, o que significa que o prefixo SID não pode ser removido depois do encaminhamento do pacote; o bit M indica se os SIDs são anunciados pelo servidor de mapeamento; o bit E indica que o penúltimo salto deve trocar o prefixo SID por um rótulo *explicit-null*, o bit V indica que o prefixo SID é um valor absoluto e, finalmente, o bit L indica quando o prefixo SID tem significado local, ou seja, é um índice.

Existem Prefixos SID denominados de *Anycast* por serem anunciados a múltiplos

nós da rede. O tráfego é encaminhado pelo originador do prefixo SID *Anycast*, escolhendo a melhor rota baseada no IGP. Os nós que anunciam o mesmo prefixo SID *Anycast* tem o mesmo SGRB. Esses prefixos são divulgados entre diferentes áreas através dos roteadores ASBR (*Autonomous System Boundary Router*), responsáveis por distribuir rotas de outros roteadores de outras áreas e de outros sistemas autônomos.

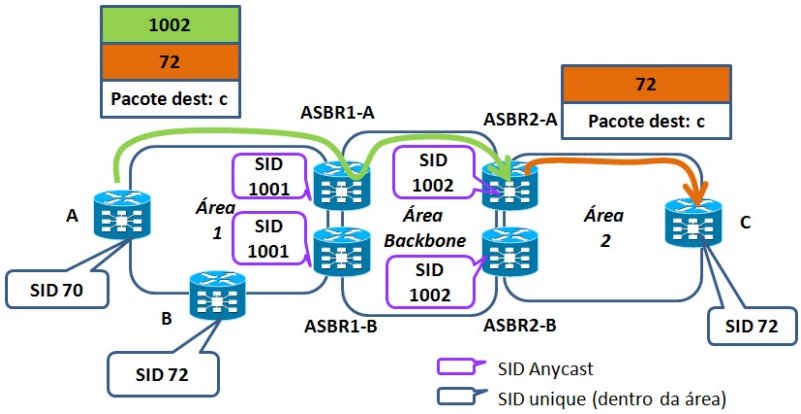


Figura 5.35. Prefixo SID *Anycast*.

A Figura 5.35 mostra um exemplo de aplicação de prefixo SID *Anycast*. Em redes multi-áreas, os SIDs dos ASBRs são *Anycast*, pois são únicos em todo domínio e são redistribuídos em cada região. Apesar dos SIDs serem únicos internamente nas áreas não-backbone, eles podem ser reutilizados em outras áreas. Por exemplo, o rótulo 72 leva até C dentro da área 2 e B dentro da área 1. Porém, os rótulos {1001,72} levam até B de qualquer lugar, e {1002,72} levam até C de qualquer lugar. Os rótulos 1001 e 1002 são divulgados em todo o domínio SR, possuindo re-roteamento rápido nativo (*Fast Reroute* - FRR) quando ocorrer falha em algum dos ASBRs. No OSPF, quando um nó anuncia seu prefixo SID, ele inclui este prefixo nas *Opaque LSAs* que são enviadas a seus vizinhos. Ao propagar um prefixo, o OSPF anuncia um prefixo recebido ou originado de outra área. Quando um nó cria um prefixo, anunciando-o como local (Segmento Local), o nó é proprietário deste prefixo. No OSPF, os prefixos SID são propagados entre áreas, enquanto os de adjacências não são. O OSPF não tem como identificar quais nós originaram o prefixo e quais propagaram este prefixo. As flags também transportam informações do comportamento do nó originador do prefixo, como o *explicit-null*. Esse comportamento não deve ser aplicado aos nós propagadores do prefixo, mas apenas ao originador. Em multi-áreas OSPF, o ABR (*Area Border Router*) ou ASBR propaga o prefixo SID não-locais com o bit NP=0 (*No PHP*) e o bit E=0 (*no-explicit-null*). Já para prefixos locais, o prefixo SID é propagado com bit NP=1 e o bit E=0. A Figura 5.36 mostra um exemplo de multi-área OSPF com prefixos SID locais e não-locais que atravessam as áreas OSPF.

Os Segmentos de Adjacências tem significado local e são alocados dinamicamente a partir de um conjunto de rótulos conforme mostrado na Figura 5.28. A alocação automática de rótulos pode ser feita por adjacência, como por exemplo uma adjacência com proteção e sem proteção. No IS-IS, isso significa ter dois SIDs diferentes para adjacências L1 e L2 entre os mesmos vizinhos; no OSPF, o mesmo SID de adjacência é aplicado em todas as áreas de uma adjacência multi-área, o que representa múltiplas adjacências para

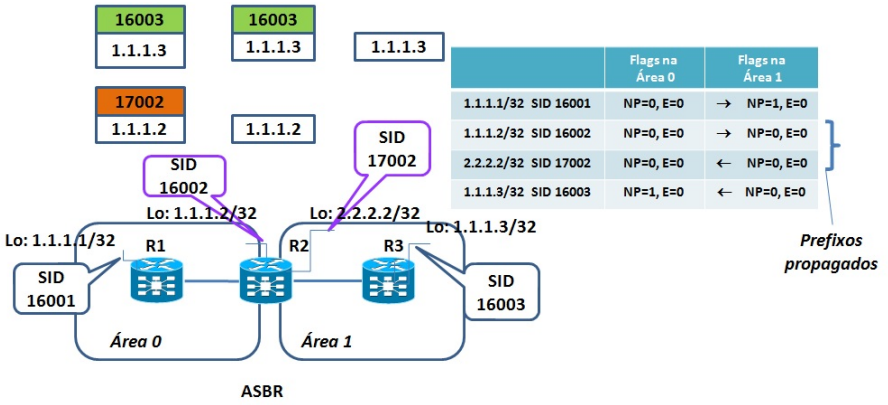


Figura 5.36. Propagação de prefixos SID e flags.

cada área diferente, em uma mesma interface. Os Segmentos de Adjacência têm persistência de rótulo, ou seja, o mesmo rótulo é alocado depois da recuperação de uma falha. No OSPF, os Segmentos de Adjacências são representados nas "flags" sub-TLVs do *Opaque LSA*, com os seguintes bits quando configurado com "1": B é o bit de Backup, que indica uma adjacência protegida; o bit V significa que o SID de adjacência transporta um valor e não um índice; o bit L significa que o SID de adjacência tem significado local e o bit S quando o SID de adjacência se refere a um conjunto de adjacências, usado para balanceamento de carga conforme mostrado na Figura 5.36. No Roteamento por Segmentos, os nós que estejam interligados através de uma rede local necessitam conduzir os fluxos de dados nesta rede. Para tal, os nós necessitam de Segmentos de Adjacência através da rede local. A solução é a criação de um "pseudo-nó"representando a rede LAN. Os Segmentos de Adjacência são associados aos nós ligados à LAN e ao pseudo-nó (Figura 5.37).

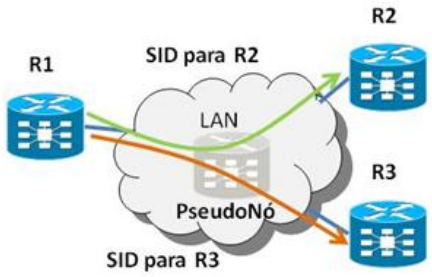
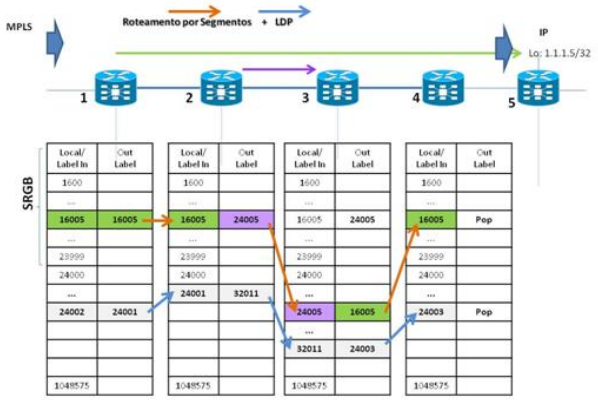


Figura 5.37. Segmento de Adjacências - Pseudo-nó.

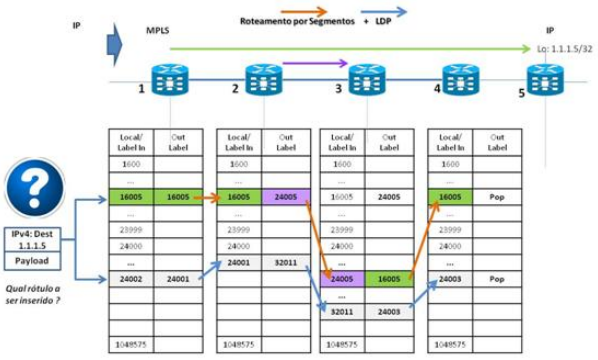
5.4.5. Coexistência do Roteamento por Segmentos e LDP

A arquitetura de redes IP/MPLS permite a coexistência de múltiplos protocolos para distribuição de rótulos como o LDP, RSVP-TE e o Roteamento por Segmentos, sem interação entre si. Cada nó de rede reserva uma faixa de rótulos dentro do SRGB para o plano de controle do Roteamento por Segmentos e rótulos fora do SRGB para alocação dinâmica pelo plano de controle MPLS tradicional. As entradas dos rótulos provenientes do LDP e do Roteamento por Segmentos são indexadas na FIB e LFIB dos roteadores. A Figura 5.38(a) mostra o sentido de um pacote da rede MPLS para a IP com rótulos

configurados pelo Roteamento por Segmentos (dentro da faixa do SRGB) e pelo LDP. As múltiplas entradas advindas do Roteamento por Segmentos ou pelo LDP podem ser programadas para o mesmo prefixo de destino conforme mostra a figura. Para o sentido do pacote da rede IP para a MPLS com Roteamento por Segmentos e LDP coexistindo da Figura 5.38(b), o rótulo a ser imposto ao pacote só pode ser recebido de um dos protocolos, não de forma simultânea. Se existem vários caminhos para o destino, a entrada da tabela deve definir se o caminho selecionado pelo protocolo de Roteamento por Segmentos ou pelo LDP deve ser usado. Por padrão, a escolha da entrada é a do protocolo LDP, sendo necessário ativar a preferência pelo Roteamento por Segmentos. Uma proposta de migração do LDP para Roteamento por Segmentos em uma rede de roteadores de núcleo IP/MPLS é inicialmente manter em execução os dois planos de controle independentes. Os roteadores de núcleo então são atualizados para Roteamento por Segmentos, e os roteadores são configurados para preferencialmente utilizar a imposição dos rótulos através do Roteamento por Segmentos. O passo final é remover o LDP dos roteadores, simplificando assim rede. O IETF vem trabalhando na elaboração de normas para padronizar a interoperabilidade de redes IP/MPLS com Roteamento por Segmentos e o protocolo LDP [Filsfils et al., 2015], permitindo a coexistência das redes legadas tradicionais IP/MPLS.



(a) Caso MPLS para IP.



(b) Caso IP para MPLS.

Figura 5.38. Coexistência entre o Roteamento por Segmentos e o LDP.

5.4.6. Servidor de Mapeamento

O Servidor de Mapeamento (*Mapping Server*) é usado na interoperabilidade entre nós habilitados e não-habilitados ao Roteamento por Segmentos. O mapeamento dos prefixos de rede e SIDs são configurados no Servidor de Mapeamento, de forma similar ao Refletor de Rotas (*Router Reflector* - RR). Esse mapeamento é realizado no servidor de mapeamento e é necessário para interoperabilidade das redes MPLS tradicionais que usam o protocolo LDP com as redes implementadas com Roteamento por Segmentos. O servidor de mapeamento é um mecanismo de controle, não necessitando estar localizado no plano de dados, e deve ser redundante. O cliente do mapeamento recebe e analisa os mapeamentos de prefixos para SIDs do servidor, que usa as entradas aprendidas e configuradas localmente nas tabelas RIB para construção de uma política válida e consistente de mapeamento de SIDs. A instância do IGP utiliza as políticas do mapeamento de SIDs para recalcular alguns ou todos prefixos SID. Uma regra de consenso é quando o servidor de mapeamento for usado, todos os nós devem se comportar como clientes para recebimento do mapeamento de prefixos de tal forma que não recebam os LSAs através de nós que não são habilitados com Roteamento por Segmentos. Os anúncios do servidor de mapeamento não são nem propagados entre áreas OSPF, nem tampouco entre áreas IS-IS. Os anúncios do servidor de mapeamento, tanto no IS-IS quanto no OSPF, são implementados nas extensões desses protocolos. É possível haver múltiplos servidores para anúncio de mapeamentos de prefixos para SIDs. Espera-se, porém, que o conjunto de mapeamentos seja igual para os servidores, a fim de se manter consistência. Caso não haja consistência, o cliente do servidor de mapeamento escolhe a entrada que não tenha sobreposição como política ativa. A arquitetura cliente-servidor de mapeamento é mostrada na Figura 4.16.

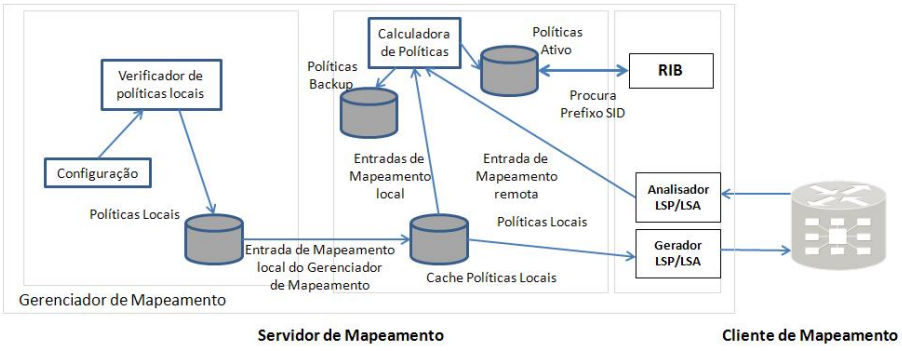
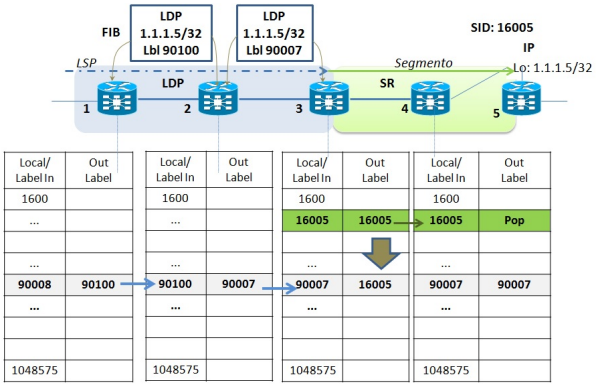


Figura 5.39. Arquitetura Cliente-Servidor de Mapeamento.

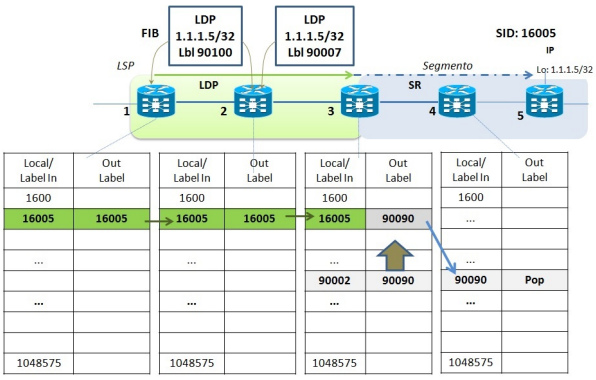
5.4.7. Interoperabilidade entre Roteamento por Segmentos e o protocolo LDP

Existem alguns modelos propostos para interoperabilidade do Roteamento por Segmentos com o protocolo LDP: Roteamento por Segmentos para LDP, LDP para Roteamento por Segmentos, Roteamento por Segmentos sobre LDP e LDP sobre Roteamento por Segmentos. O LDP para Roteamento por Segmentos conecta um LSP criado pelo LDP com um prefixo SID, em qualquer nó na borda do domínio LDP para Roteamento por Segmentos através de uma entrada na LFIB, configurada de forma automática em cada um dos domínios. O rótulo de entrada do nó de borda é obtido pelo LDP e o rótulo de saída é o do segmento, copiado do que seria o rótulo de entrada para o trecho

do segmento conforme mostrado na Figura 5.40(a). No Roteamento por Segmentos para LDP, como o destino é um domínio que não tem Roteamento por Segmentos habilitado (domínio LDP), é necessário anunciar no domínio LDP os prefixos SID na tabela LFIB destes roteadores. Isso é feito através do servidor de mapeamento, que informa o prefixo e a partir do prefixo IP é conhecido o rótulo para chegar ao destino conforme mostrado na Figura 5.40(a). Já no Roteamento de Segmentos sobre LDP, a cada borda de rede que utilize o Roteamento por Segmentos/LDP, o prefixo SID é mapeado em um LSP obtido do protocolo LDP. Na borda LDP/Roteamento por Segmentos o LSP é mapeado em um prefixo SID. Se o caminho terminar em um nó com domínio LDP, é necessário utilizar o servidor de mapeamento. Por fim, no LDP sobre Roteamento por Segmentos, na fronteira LDP/Roteamento por Segmentos, o LSP criado pelo LDP é mapeado em um prefixo SID, sendo portanto, necessário o servidor de mapeamento.



(a) LDP para Roteamento por Segmentos.



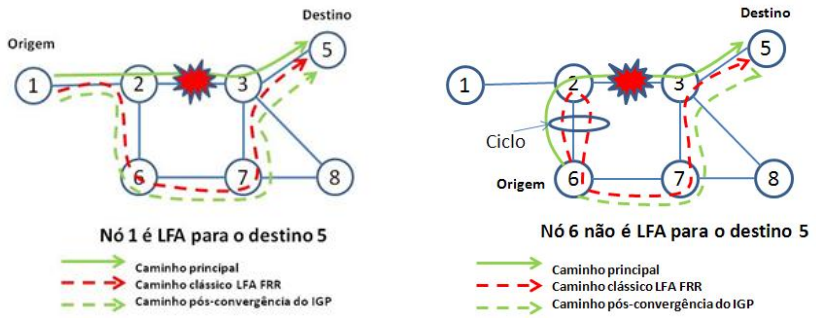
(b) Roteamento por Segmentos para LDP.

Figura 5.40. Interoperabilidade entre o Roteamento por Segmentos e o LDP.

5.4.8. Mecanismo de proteção - Topology Independent LFA (TI-LFA)

O mecanismo TI-LFA (*Topology Independent Loop Free Alternate*) é uma técnica de re-roteamento para convergência rápida [Francois et al., 2015]. O tempo de convergência deve ser em menos de 50ms, tempo inferior à convergência do IGP da rede. O LFA clássico em redes IP/MPLS depende da topologia, nem sempre provendo o melhor caminho de proteção. No mecanismo clássico LFA-FRR (*Loop Free Alternate Fast Reroute*), o IGP pré-calcula um caminho de proteção para cada caminho primário, instalando

o caminho de proteção no plano de dados. Na ocorrência de uma falha, todos os caminhos de proteção dos destinos impactados são habilitados com prefixos existentes na rede, convergindo em 50ms. O mecanismo clássico LFA-FRR possui algumas desvantagens como cobertura incompleta e caminho de proteção nem sempre ótimo. O LFA clássico é dependente da topologia e nem toda topologia é livre de ciclos para todos os destinos conforme mostra a Figura 5.41. Na Figura 5.41(a), o caminho principal sai do nó de origem 1 ao nó de destino 5. O caminho de proteção tem os nós intermediários 2, 6 e 7 livres de ciclos, coincidindo com o caminho que o IGP tomaria após a convergência da rede (maior que 50ms). Na Figura 5.41(b), um exemplo de caminho principal saindo do nó de origem 6 para o nó de destino 5. Nesse caso, o LFA clássico tem cobertura incompleta (nem todos os nós são LFA) e é dependente da topologia, nem sempre provendo um caminho ótimo de proteção, com a presença de ciclos. Uma topologia independente provê 100% de cobertura de nós com LFA e esse tipo de mecanismo de proteção é comumente usado em Roteamento por Segmentos.



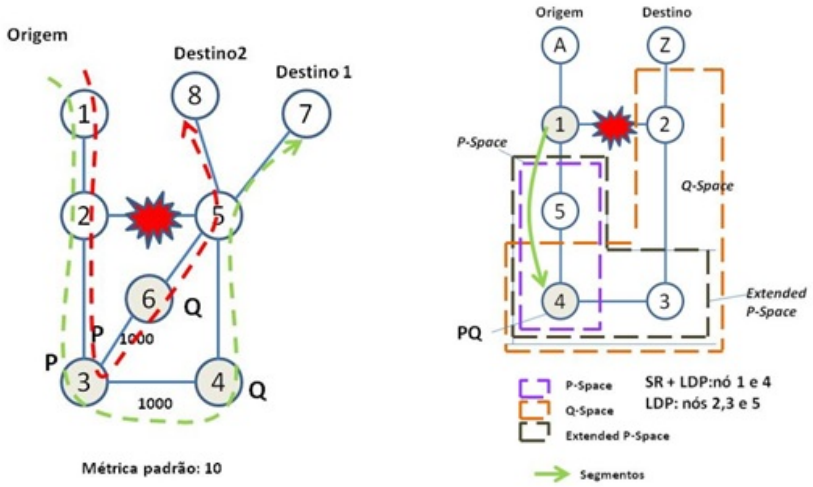
(a) Exemplo sem ciclo (LFA). (b) Exemplo com ciclo.

Figura 5.41. Mecanismo LFA clássico.

O TI-LFA usa o caminho pós-convergência com um caminho de proteção de reroteamento rápido. O mecanismo TI-LFA foi elaborado para ter 100% de cobertura dos nós sem ciclos e com tempo de convergência $\leq$50ms, prevenindo congestionamento transiente e roteamento não-ótimo. O caminho otimizado e natural após uma falha seria o caminho pós-convergência calculado pelo IGP, utilizado pelo TI-LFA. No entanto, não existe uma garantia de convergência rápida ($\leq$50ms), além dos nós serem livres de ciclos para este novo caminho. O Roteamento por Segmentos força o caminho de pós-convergência através de uma lista de segmentos. O mecanismo do TI-LFA influencia o mecanismo clássico de LFA, forçando a ausência de ciclos em um caminho ótimo, encontrado pelo IGP após o período de convergência da rede.

O mecanismo TI-LFA utiliza o algoritmo "PQ", que encontra nós que satisfaçam as propriedades de "P" e "Q". O algoritmo é proprietário Cisco e não está no escopo na padronização do IETF para o TI-LFA. O TI-LFA pode ser usado para balanceamento de tráfego, e proteção de tráfego MPLS com protocolos de controle tradicionais como o LDP. O TI-LFA utiliza caminhos ECMP (*Equal-Cost Multi-Path routing*), permitindo o balanceamento de tráfego baseado em uma função hash. A Figura 4.21a mostra um exemplo de balanceamento de tráfego do nó origem 1 para os nós de destino 7 e 8. O enlace a ser protegido é o enlace 2-5 (enlace do caminho principal ou primário). O TI

LFA calcula dois pares de nós “P” e “Q” para ambos os destinos 7 e 8, neste exemplo para o destino 10 (P=3, Q=4) e para o destino (P=3, Q=6). O TI LFA estatisticamente faz o balanceamento do tráfego nestes pares de nós. O tráfego MPLS controlado pelo LDP e não por Roteamento por Segmentos, também pode fazer uso do TI-LFA, eliminando sessões T-LDP (*Target LDP*), e o Roteamento por Segmentos pode ser implementado em “ilhas” dentro da rede controlada pelo LDP, influenciando a proteção do TI LFA para redes MPLS com LDP. A FIB (*Fowarding Information Base*) usa a fusão de rótulos obtidos do Servidor de Mapeamento (*Mapping Server*), que anuncia os prefixos para segmentos referentes ao caminho de proteção. Os nós de onde partem os caminhos de proteção devem ter Roteamento por Segmentos habilitados, bem como o nó de destino, que deve ser associado a um prefixo SID anunciado por este nó, ou através de um Servidor de Mapeamento (*Mapping Server*). Além da origem e do destino, os nós P e Q também devem estar habilitados para Roteamento por Segmentos. O exemplo da Figura 4.21b mostra o nó 1 e 4 com Roteamento por Segmentos e LDP habilitados. O nó de destino “Z”,e os nós 2, 3 e 5 são configurados somente com LDP. O Servidor de Mapeamento (*Mapping Server*) anuncia o prefixo SID 16006 para a interface “loopback” do nó Z, cujo IP é por exemplo 1.1.1.6/32. O nó 4 é um nó do tipo PQ para proteção do nó de destino “Z” no nó 1 do caminho principal ou primário 1-2. O nó 1 e 4 utilizam as funcionalidades de interoperabilidade do Roteamento por Segmentos e do protocolo LDP para dirigir os pacotes para o caminho de proteção encontrado pelo TI LFA.



(a) Balanceamento de tráfego de proteção.

(b) TI LFA em redes MPLS com LDP.

Figura 5.42. Exemplos de implementação de TI LFA com Roteamento por Segmentos.

5.4.9. Aplicações em Roteamento por Segmentos

A Engenharia de Tráfego é um exemplo de aplicação no contexto de Roteamento por Segmentos (*Segment Routing - Traffic Engineering* - SR-TE) utilizando os benefícios das Redes Definidas por Software e do Roteamento por Segmentos [Bhatia et al., 2015]. O Roteamento por Segmentos permite a configuração, a modificação e a remoção de caminhos TE dentro de um domínio de rede, operando somente na borda da rede. O plano

de controle do Roteamento por Segmentos pode ser mantido tanto de forma centralizada através de um controlador SDN, quanto de forma distribuída no plano de controle existente nos roteadores de núcleo habilitados com Roteamento por Segmentos. Neste último, os roteadores podem pertencer a uma rede composta somente por nós que executem o Roteamento por Segmentos ou em uma mista, que execute também o LDP. Na Engenharia de Tráfego com Roteamento por Segmentos, o roteador de núcleo de origem (*Head End LSR*) é denominado roteador de núcleo de origem SRTE (*Segment Routing Traffic Engineering Head End*), onde os pacotes são classificados e onde são impostas as listas de segmentos. Todas as funcionalidades de engenharia de tráfego são influenciadas pelas decisões do Roteamento por Segmentos e pelo controlador SDN. Os roteadores intermediários (*Mid Point*), para efeito do Roteamento por Segmentos não existem em sua topologia lógica, não necessitando de estados adicionais (troca de rótulos e atualizações de tabelas LFIBs) e sinalizações (protocolo RSVP-TE). A centralização da engenharia de tráfego permite melhorar a otimização, a previsibilidade e a convergência da rede através de aplicações programáveis por APIs NBI e de mecanismos de programação da rede como o PCEP e APIs SBI. O PCE e o BGP-LS possuem extensões para Roteamento por Segmentos. O BGP-LS é usado para anunciar o estado do enlace e a base de dados TE (*Traffic Engineering Database* - TED). A informação do estado do enlace é repassado em LSAs opacos, incluindo a informação dos estados do Roteamento por Segmentos distribuídos pelo IGP. O PCE (*Path Computation Element*) especifica uma lista de SIDs a partir de uma requisição de indicação de caminho proveniente da aplicação, e o PCC (*Path Computation Client*) encaminha o tráfego impondo a lista de segmentos nos pacotes conforme mostrado na Figura 5.43. Não existe sinalização RSVP-TE e os caminhos podem ser iniciados pelo PCE ou PCC.

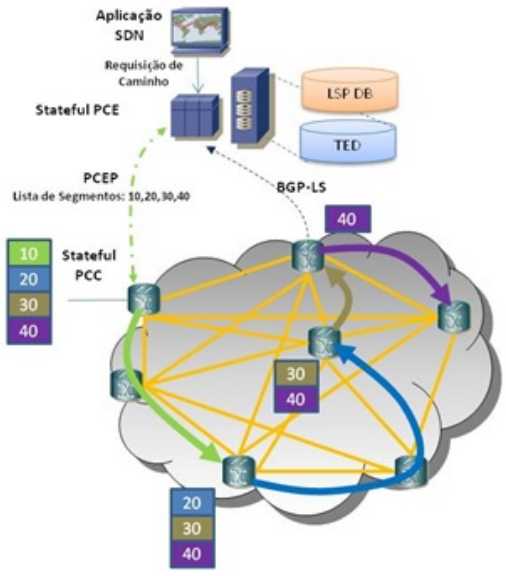


Figura 5.43. Roteamento por Segmentos com decisão de caminho por SDN.

A centralização da inteligência da engenharia de tráfego com uma visão fim a fim da rede pode ser obtida através da tecnologia de Roteamento por Segmentos e Redes Definidas por Sotware, onde controlador expressa o caminho em uma lista de segmentos, e a

rede mantém os segmentos e providencia o reroteamento rápido (*Fast Reroute* - FRR) para o mesmo, coincidente com os caminhos de menor custo (*Equal-cost multi-path routing* - ECMP).A Figura 5.44 mostra um controlador SDN fazendo a orquestração de engenharia de tráfego para descoberta de um caminho que atenda a demanda de banda de 2G para um túnel MPLS que vai de A a Z. Nas redes IP/MPLS legadas o RSVP-TE era responsável para obtenção dos recursos através de complexa sinalização. O controlador encontra o caminho a partir do dados coletados (através do BGP-LS por exemplo) usando estes dados na computação do caminho, que é uma lista de segmentos, configurados através uma "southbound interface" como NETCONF ou CLI conforme deduzido na figura.

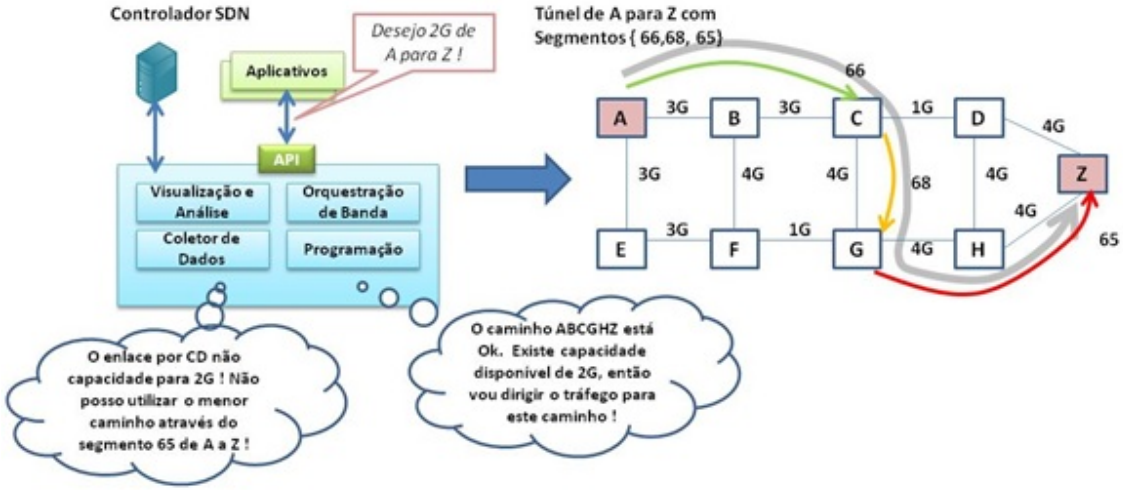


Figura 5.44. Aplicação SR TE com orquestração de banda.

Outra aplicação do SR TE é a determinação do caminho baseado em classe de serviço (*CoS-Based TE*). Nesse caso, o tráfego de dados pode ser direcionado por segmentos onde os enlaces possuem maior capacidade e menor custo por bit, enquanto o tráfego de voz sobre IP pode ser direcionado por segmentos com enlaces de menor latência. Para redes extensas e multi-áreas são usados prefixos SID "Anycast". Outra aplicação do SR TE para fins operacionais e de recuperação de rede é o controle OAM (*Operations, Administration and Maintainance*). O uso dos segmentos de adjacência permite monitorar cada enlace da rede, sendo bastante útil em redes extensas e complexas como a rede de roteadores de núcleo das operadoras de telecomunicações. A Figura 5.45 demonstra esse tipo de aplicação. O monitoramento do caminho ABCFGDH é realizado por segmentos de adjacência, permitindo localizar perda de pacotes em um enlace. Este tipo de aplicação é descrito na versão "draft" do IETF *draft-geib-spring-oam-usecas-02*.

O Roteamento por Segmentos automatiza o "peering" de roteadores de diferentes sistemas autônomos através da automação do BGP com alocação de SIDs, simplificando o "peering" da rede IP/MPLS legada. O controlador SDN aprende os SIDs dos "peers" e a topologia externa através do roteador de borda pelo BGP-LS EPE (*Egress Peering Engineering*), definidos nas extensões do BGP-LS. Esse tipo de caso de uso, definido no "draft" *draft-ietf-spring-segment-routing-central-epe*, é ilustrado na Figura 5.46.

Das aplicações com Roteamento por Segmentos com BGP, se destacam casos de

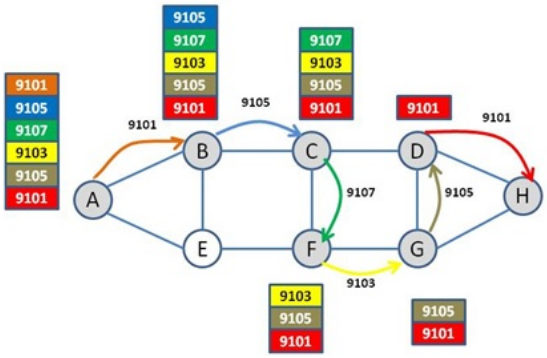


Figura 5.45. Aplicação OAM com roteamento de segmentos de adjacência.

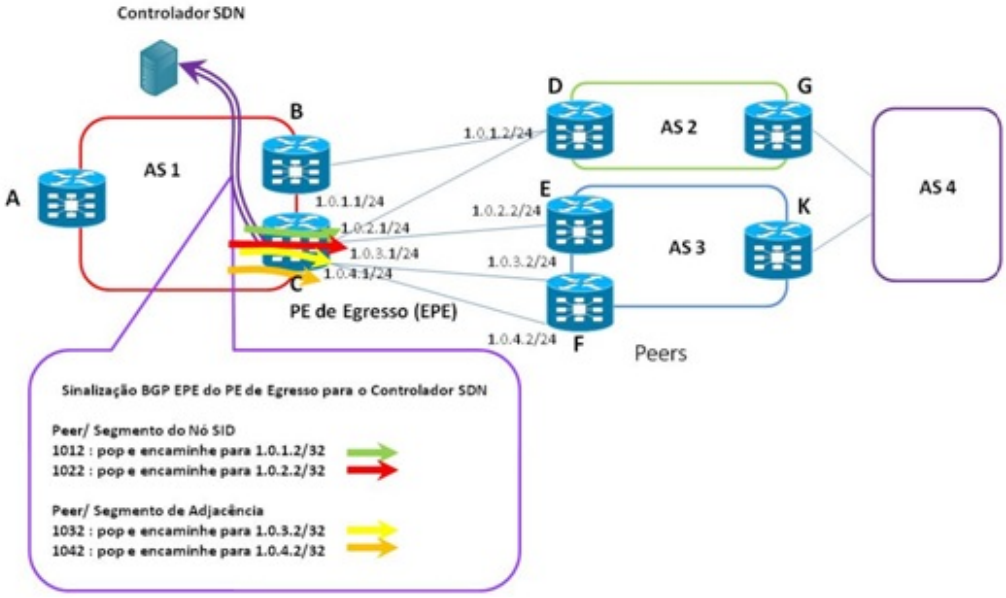


Figura 5.46. Aplicação EPE - automação de "peering" da rede.

uso para Datacenters em escala massiva (*Massive Scale DC* - MSDC) onde os prefixos de segmentos são usados no BGP assim como no IGP, definido no documento do IETF "draft" *draft-ietf-idr-bgp-prefix-sid*. Qualquer nó dentro da topologia aloca o mesmo segmento BGP para o mesmo comutador do Datacenter (*Top of Rack Switch* - TOR), com os benefícios do reroteamento rápido e engenharia de tráfego. Uma aplicação com plano de encaminhamento de dados IPv6 e também MPLS é a cadeia de serviços baseada em Roteamento por Segmentos. O nó que conecta a instância do serviço origina um SID baseado no comportamento do serviço. O nó pode ser virtual ou físico, os SIDs são conhecidos no nó de ingresso, através de um controlador SDN com múltiplas APIs como protocolos IGP e BGP, NETCONF, REST, OpenFlow, etc. Não existe sobrecarga da aplicação, nem manutenção de estados para cada cadeia, apenas um único estado por instância de serviço. Recentemente o IETF definiu uma proposta para transportar cadeias de serviço dentro de um cabeçalho NSH (*Network Service Header*) que identifica uma cadeia (*path id*) para transporte de meta-dados. O IETF e esté trabalhando para integrar o NSH com Roteamento por Segmentos como definido no "draft" Network Service Header *draft-ietf-*

sfc-nsh-04, que mapeia os segmentos dentro do *path id* como uma opção para redes de Datacenter. A Figura 5.47 mostra um exemplo de cadeia de serviços com Roteamento por Segmentos, podendo ser aplicável em NFV (*Network Functions Virtualization*). O Roteamento por Segmentos com plano de encaminhamento de dados IPv6 apresenta os mesmos estudos de caso do com plano de encaminhamento de dados MPLS.

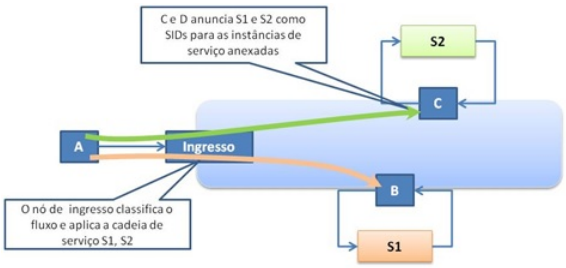


Figura 5.47. Cadeia de serviços com Roteamento por Segmentos.

5.4.10. Padronização IETF

A padronização da arquitetura de Roteamento por Segmentos está sendo conduzida pelo IETF, no grupo de trabalho SPRING WG (*Source Packet Routing in Networking Working Group*), cujas RFCs estão na versão "draft", que focam arquitetura, casos de uso e extensões dos protocolos IS-IS, OSPF, BGP, BGP-LS, PCEP e IPv6.

5.5. Experimentação Prática

Nos experimentos práticos deste minicurso, a ferramenta de simulação OSHI (*Open Source Hybrid IP/SDN networking*) [Davoli et al., 2015] é empregada. Essa ferramenta auxilia a investigação e prototipagem das redes Openflow, para que elas possam oferecer as mesmas funcionalidades das redes IP/MPLS. A ideia foi utilizar um simulador que permita trabalhar com nós de redes virtuais e reais, e com implementação de software aberto em IP e SDN. A interface *southbound* é Openflow e as interfaces *northbound* são APIs REST. O tráfego é enviado em vários tipos de túneis tais como MPLS, VLAN, Q-in-Q e Ethernet PBB (*Provider Backbone Bridge*) através do controlador SDN. Os sistemas operacionais de rede escolhidos foram o Floodlight, Ryu e ONOS [Salsano et al., 2014b, Stancu et al., 2015]. O simulador de rede híbrida IP/SDN possui três formas de implementação: no Virtual Box "run time", no ambiente de simulação do Mininet e em provas de conceito de redes SDN como o projeto OFELIA aplicando o conceito de experimento como serviço (*testbed as a service*) [Salsano et al., 2014b]. Nesta seção, o Mininet é usado [Salsano et al., 2014b, OSHI e Mininet, 2015]. O simulador utiliza nós virtuais de rede de núcleo denominados Roteadores de Núcleo em software aberto (*Open Source Label Switch Routers* - OpenLSR), que geram pacotes OSPF e LDP utilizando o Quagga e computam os rótulos MPLS que são instalados nos comutadores usando o protocolo OpenFlow. O controlador SDN também possui um módulo de software para engenharia de tráfego e roteamento por segmentos [Davoli et al., 2015].

5.5.1. Introdução ao Mininet

O Mininet [Mininet, 2015] é um emulador de rede que cria redes virtuais com servidores, comutadores, controladores e enlaces virtuais em uma única máquina física ou virtual. O Mininet possui linha de comando própria (*Command Line Interface* - CLI) e APIs que permitem a criação de rede e serviços, customização e compartilhamento com outros usuários. O Mininet é ideal para experimentos com OpenFlow e redes SDN e pode ser executado em um PC ou até mesmo em um laptop. Os passos seguintes ilustram a criação e o uso de uma topologia mínima, que inclui o controlador POX, um comutador OpenFlow e dois servidores.

Passo 1: Inicialização do Mininet com a topologia mínima.
Passo 2: Verificação dos nós da rede.
Passo 3: Verificação dos enlaces da rede.
Passo 4: Verificação dos endereços lógicos dos dispositivos da rede. Os nós são configurados em uma subrede 10.0.0.0/8 por padrão.
Passo 5: Acesso aos servidores HTTP através do terminal.
Passo 6: Teste da conectividade da rede através do `ping`.

O Mininet permite a criação de topologias mais complexas com mais comutadores e servidores. O Mininet ainda permite a construção de topologias através de APIs em Python, que é a base de construção do emulador. Os comandos passo-a-passo necessários para a execução deste primeiro experimento e de todos os outros a seguir estão disponíveis em `http://www.gta.ufrj.br/~silverio/SRArquivoSBRC2016.htm`.

5.5.2. Construção de uma rede simples com roteamento no Mininet

Este experimento visa criar uma rede simples com roteamento estático, onde o roteador recebe e processa pacotes como um roteador físico real, e depois os encaminha pelas interfaces corretas. O roteador emulado encaminha pacotes de um cliente para dois servidores HTTP.

Para tal, os seguintes passos devem ser seguidos:

Passo 1: Instalação do módulo do controlador POX.
Passo 2: Verificação dos arquivos de configuração.
Passo 4: Configuração do ambiente através da execução do arquivo `config.sh`.
Passo 5: Inicialização do controlador POX.
Passo 6: Realização dos testes propostos a partir da execução do arquivo `sr_solution`.
Passo 7: Inicialização dos testes propostos no Mininet.
Passo 8: Captura dos pacotes no formato `pcap` através do executável `sr`.

5.5.3. Open Source Hybrid IP/SDN Networking

A ferramenta OSHI (*Open Source Hybrid IP/SDN networking*) [OSHI, 2015] permite o uso de nós híbridos que combinam o encaminhamento tradicional IP com o encaminhamento SDN. O OSHI é uma ferramenta de código aberto que implementa comutadores OpenFlow (*OpenFlow Capable Switch* - OFCS), um encaminhador de pacotes IP e um *daemon* de roteamento IP. O OFCS é conectado ao conjunto de interfaces físicas pertencentes à rede IP/SDN, enquanto o encaminhador de pacotes IP é ligado a um conjunto de

portas virtuais OFCS, como visto na Figura 5.48. No nó OSHI, o OFCS é implementado usando o Open vSwitch, o encaminhador de pacotes IP é o kernel do Linux e o Quagga atua como o *daemon* de roteamento. .

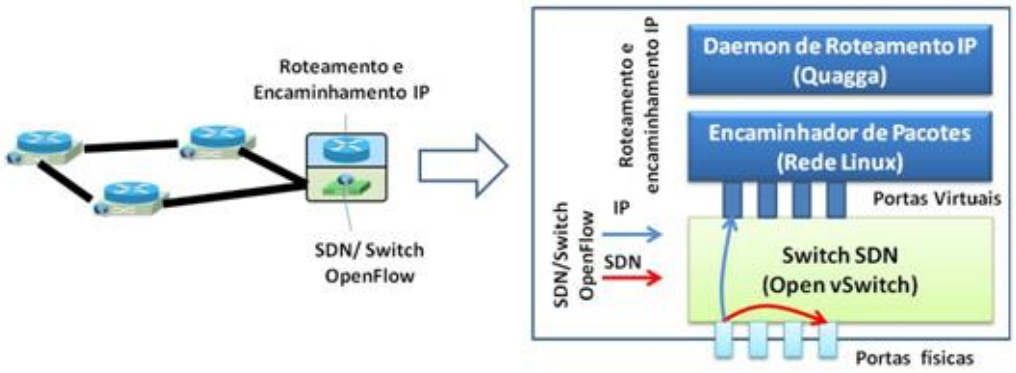


Figura 5.48. Construção de uma rede simples com roteamento no Mininet (Adaptado de [Salsano et al., 2014b]).

Os serviços a serem simulados pela ferramenta OSHI são do tipo IP ponto-a-ponto como Circuitos Virtuais Alugados (*Virtual Leased Lines* - VLL), Pseudo-fio (*Pseudowires* - PW) de camada 2 e comutadores virtuais de camada 2 (*Virtual Switched Services* -VSS) sobre um backbone IP/MPLS. Este último permite também simular engenharia de tráfego em redes IP/MPLS (túneis TE) e roteamento por segmentos. A fim de apoiar tanto o desenvolvimento, quanto os aspectos de teste e de avaliação comparativa, a ferramenta Mantoo (*Management Tools*) [Salsano et al., 2014b, Salsano et al., 2014a] foi adicionada para auxiliar experiências SDN sobre simuladores e redes experimentais distribuídas.O Mantoo inclui *scripts* Python para configuração e controle dos simuladores ou *scripts* para medições de desempenho e um visualizador de topologia 3D.A representação da topologia é através de arquivos no formato JSON e a interface gráfica possui um módulo de criação automático de topologia a partir dos dados desses arquivos. Os *scripts* de configuração incluem um analisador da topologia, extensões das bibliotecas do Mininet e o configurador OSHI.

5.5.4. Criação de um serviço MPLS através da ferramenta OSHI-TE

A ferramenta OSHI-TE utiliza nós virtuais de rede de núcleo chamados OpenLSR (*Open Source Label Switch Routers*), que geram pacotes OSPF e LDP utilizando o Quagga. Este último computa os rótulos MPLS que são instalados nos computadores usando o protocolo OpenFlow. O controlador SDN também possui um módulo de software para engenharia de tráfego e roteamento por segmentos [SPRING, 2015].

Os serviços simulados pelo OSHI são implementados como caminhos em uma rede SDN, a partir de controladores Ryu ou Floodlight centralizados. Duas propostas foram concebidas e implementadas para o estabelecimento de caminhos, a primeira baseia-se em identificadores de VLAN e a segunda em rótulos MPLS. Como exemplo, o serviço Pseudo-fio foi implementado usando apenas os rótulos MPLS, a partir do encapsulamento do pacote Ethernet cliente.O serviço de Pseudo-fio em uma rede híbrida IP/SDN tem as mesmas características dos serviços implementados em rede tradicional

IP/MPLS. O controlador utilizado foi o Ryu e um *script*, denominado `VLLPusher` que utiliza a API REST do controlador foi utilizado para recuperar a topologia de rede e, em seguida, avaliar o caminho mais curto entre os pontos finais do serviço. Nesse passo, é possível introduzir os aspectos de engenharia de tráfego [Davoli et al., 2015]. Finalmente, o *script* aloca os rótulos MPLS e usa a API REST do controlador Openflow para definir as regras para o encaminhamento de pacotes e a comutação de rótulos MPLS.

A arquitetura do nó OSHI permite não somente a criação dos serviços de Pseudo-fio, mas também de comutadores virtuais. O comutador OpenFlow suporta a inserção e retirada de rótulos, enquanto o ACE (*Access Encapsulator*) provê o túnel GRE. O ACE é implementado como uma nova instância do Open vSwitch, utilizando duas funções de virtualização: espaço de nomes de redes e pares de portas Ethernet virtuais.

A seguir são enumerados os passos para a criação de um serviço de Pseudo-fio, que pode ser executado através da linha de comando ou da interface gráfica Mantoo:

Passo 1: Execução do *script* de carga da topologia de um serviço Pseudo-fio através do Mininet ou da interface gráfica.
Passo 2: Realização de testes de conectividade e captura de pacotes com farejadores de rede.
Passo 3: Acesso remoto ao controlador e verificação das saídas dos *scripts*.
Passo 4: Teste da conectividade entre os roteadores clientes.
Passo 5: Configuração dos circuitos virtuais.
Passo 6: Criação dos circuitos virtuais.
Passo 7: Teste da conectividade (Passo 4) e verificação dos serviços operacionais.
Passo 8: Remoção dos circuitos virtuais.

5.5.5. Criação de um serviço em rede MPLS-TE com roteamento por segmentos

Considerando o exemplo de rede MPLS da Figura 5.49, o roteamento por segmentos é baseado na inserção, retirada e comutação de rótulos que representam os segmentos. Na figura, o nó C e F anunciam seus segmentos globais 69 e 90 com os endereços de suas interfaces loopback para os outros nós através do IGP. Já o nó E anuncia o rótulo 23 como segmento de adjacência. Dessa forma, se o nó A enviar um pacote para F, ele deve criar uma lista de segmentos contendo os segmentos {90, 23, 69}. Os segmentos AC e EF correspondem aos menores caminhos de A para C e de E para F e, por conseguinte, devem ser os caminhos encontrados pelo IGP para alcançar o destino.

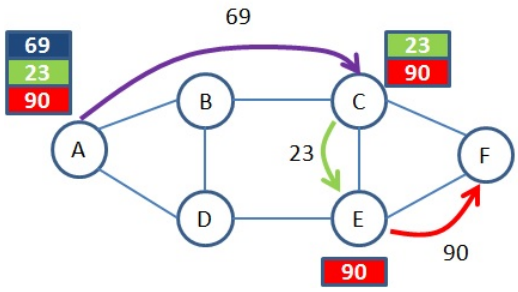


Figura 5.49. Caso de uso de roteamento por segmentos simplificado.

O projeto de rede com nós OSHI pode acomodar o roteamento por segmentos graças à característica híbrida IP/SDN da arquitetura do nó. Algumas premissas foram consideradas no ajuste da ferramenta para roteamento por segmentos: os segmentos utilizam rótulos MPLS, segmentos locais não são suportados, são utilizados os 16 bits mais significativos à direita correspondente à interface "loopback" para codificar o rótulo, cada interface OSHI em um nó tem o mesmo endereço físico e existe um mapeamento estático de MAC entre os nós OSHI usados para roteamento por segmentos. A ação de retirada do rótulo ocorre no último nó OSHI e não no penúltimo nó. Como na ferramenta não existe um plano de dados e de controle MPLS, este pode ser replicado utilizando tabelas OpenFlow e o uso de comutadores habilitadas com OpenFlow.

A seguir são enumerados os passos para a criação de um serviço VLL.

Neste exemplo, uma topologia gerada pela interface gráfica é analisada e um conjunto de fluxos é extraído para em seguida permitir a alocação de segmentos.

Passo 1: Verificação das extensões do roteamento por segmentos do Mininet.
Passo 2: Carregamento da topologia na interface gráfica.
Passo 3: Implantação da topologia.
Passo 4: Identificação do endereço IP do controlador e execução do *script* de implantação.
Passo 5: Geração de um catálogo de fluxos para ser manuseado pelo algoritmo de alocação de roteamento por segmentos.
Passo 6: Execução do algoritmo de roteamento por segmentos a partir da interface gráfica do OSHI.
Passo 7: Execução do projeto.
Passo 8: Execução do aplicativo `sr_vll_pusher`.
Passo 9: Realização dos testes de conectividade no VLL.
Passo 10: Remoção dos circuitos virtuais e término da emulação do Mininet.

5.5.6. Engenharia de Tráfego através do OSHI-TE

Para implementação de engenharia de tráfego a ferramenta OSHI utiliza um aplicativo que influencia as decisões do controlador Ryu através de uma API REST. Na entrada, é necessário um arquivo de configuração no formato JSON que descreve as relações do tráfego, enquanto que a topologia e capacidade dos enlaces são obtidas da API REST do módulo de topologia e do módulo do comutador habilitado com OpenFlow, que provê a topologia e velocidade das portas. A implementação do TE é dividida em três partes: a obtenção das entradas, algoritmos baseados em heurísticas e a instalação de regras. O último passo é alcançado pela API REST, que permite implantar as regras nos comutadores OpenFlow.

5.6. Considerações Finais e Direções Futuras

O roteamento por segmentos é uma proposta emergente para simplificação do roteamento e da configuração das redes das operadoras de telecomunicações. No Roteamento por Segmentos, os estados por fluxo são mantidos apenas nos nós de borda da rede e a configuração das redes de núcleo pode se tornar automatizada. Para tal, o Roteamento por Segmentos utiliza os benefícios da programação das redes definidas por

software através da centralização do plano de controle. A visão centralizada do controlador permite o cálculo dinâmico dos segmentos e a construção de rotas entre os nós de borda [Sgambelluri et al., 2015b].

O roteamento por segmentos tende futuramente a ser aplicado em redes de transporte ópticas com GMPLS, que utilizam instâncias hierárquicas de sessões de sinalizações. Tais sessões de sinalização devem ser estabelecidas também nos nós de trânsito das redes de transporte ópticas, tendo como consequência uma implementação complexa do plano de controle. O roteamento por segmentos tem grande potencial de integração entre as camadas de rede de transporte óptica e de roteadores de núcleo, permitindo a convergência IP e óptica [Sgambelluri et al., 2015a] e, consequentemente, proporcionando inúmeros benefícios para a operadora de telecomunicações. Dentre os benefícios estão o uso racional e otimizado da capacidade da rede, com proteção e restauração de acordo com de níveis de serviço diferenciados por fluxo de dados. Outras direções futuras indicam a aplicação do conceito de roteamento por segmentos em redes Carrier Ethernet [Cai et al., 2014, Bidkar et al., 2014], através da adição de rótulos com os segmentos no quadro Ethernet ("*Ominipresent Ethernet*"). Outras propostas de implementação do roteamento por segmentos em redes Carrier Ethernet são através do MPLS-TP (*MPLS Transport Profile*), que permite um uso misto em redes Carrier Ethernet baseadas em comutadores metro Ethernet e redes ópticas de transporte ou através do uso do encapsulamento do Ethernet com MPLS. O roteamento por segmentos ainda possui pontos em desenvolvimento, como questões de segurança nas extensões do cabeçalho IPv6; bem como integração com outras tecnologias, como por exemplo em virtualização de funções de redes. Nessa última, o roteamento por segmentos é usado para transportar a informação da cadeia de serviços, interoperando com o cabeçalho de serviços de rede, que ainda está em curso de padronização.

Referências

[Adrian et al., 2015] Adrian, G., Vasileios, K. e Xenofontas, D. (2015). Evaluating the effect of centralization on routing convergence on a hybrid bgp-sdn emulation framework. *Computer communication review*, 44:369–370.

[Alvarez, 2016] Alvarez, S. (2016). BRKMPL-2100 - deploying MPLS traffic engineering. Acessado em fev/2016 `https://www.ciscolive.com/online/connect/sessionDetail.ww?SESSION_ID=7818&backBtn=true`.

[Andreas et al., 2015] Andreas, B., Arsany, B., Martin, R. e Wolfgang, K. (2015). Survey on network virtualization hypervisors for software defined networking. *IEEE Communications Surveys and Tutorials*, 18:655–685.

[Asati, 2012] Asati, R. (2012). BRKIPM-2017 - designing IP/MPLS VPN networks. Acessado em fev/2016 `https://www.ciscolive.com/online/connect/sessionDetail.ww?SESSION_ID=2636&backBtn=true`.

[Bhatia et al., 2015] Bhatia, R., Hao, F., Kodialam, M. e Lakshman, T. (2015). Optimized network traffic engineering using segment routing. Em *IEEE Conference on Computer Communications (INFOCOM)*, p. 657–665.

[Bidkar et al., 2014] Bidkar, S., Gumaste, A., Ghodasara, P., Hote, S., Kushwaha, A., Patil, G., Sonnis, S., Ambasta, R., , Nayak, B. e Agrawal, P. (2014). Field trial of a software defined network (SDN) using carrier ethernet and segment routing in a tier-1 provider. Em *IEEE Global Communications Conference (GLOBECOM)*, p. 2166–2172.

[Bierman et al., 2015] Bierman, A., Bjorklund, M. e Watsen, K. (2015). RESTCONF Protocol. https://tools.ietf.org/html/draft-ietf-netconf-restconf-09.

[Cai et al., 2014] Cai, D., Wielosz, A. e Wei, S. (2014). Evolve Carrier Ethernet architecture with SDN and segment routing. Em *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, p. 1–6.

[Casado et al., 2014] Casado, M., Foster, N. e Guha, A. (2014). Abstractions for software defined networks. *Communications of the ACM*, 57:86–95.

[Casellas et al., 2015] Casellas, R., Munoz, R., Martinez, R., Vilalta, R., Liu, L., Tsuritani, T., Morita, I., Lopez, V., de Dios, O. G. e Fernandez-Palacios, J. P. (2015). SDN orchestration of openflow and GMPLS flexi-grid networks with a stateful hierarchical PCE. *IEEE Communications Magazine*, 7:106–117.

[Civanlar et al., 2015] Civanlar, S., Lokman, E., Kaytaz, B. e Tekalp, A. M. (2015). Distributed management of service-enabled flow-paths across multiple sdn domains. *IEEE Networks and Communications (EuCNC) 2015 European Conference on*, p. 360–364.

[Davoli et al., 2015] Davoli, L., Veltri, L., Ventre, P. L., Siracusano, G. e Salsano, S. (2015). Traffic engineering with segment routing: SDN-based architectural design and open source implementation. Em *European Workshop on Software Defined Networks (EWSDN)*, p. 111–112.

[De Ghein, 2007] De Ghein, L. (2007). *MPLS Fundamentals - A Comprehensive Introduction to MPLS Theory and Practice*. Cisco Press, 1 edição.

[development team, 2016] development team, R. (2016). Ryu Documentation Release 4.1. https://media.readthedocs.org/pdf/ryu/latest/ryu.pdf.

[El-Sayed e Jaffe, 2002] El-Sayed, M. e Jaffe, J. (2002). A view of telecommunications network evolution. *IEEE Communications Magazine*, 40:74–81.

[Farrel, 2006] Farrel, A. (2006). Introduction to the Path Computation Element. https://www.itu.int/ITU-/worksem/ngn/200604/presentation/s4_farrell.pdf.

[Feamster et al., 2004] Feamster, N., Balakrishnan, H., Rexford, J., Shaikh, A. e van der Merwe, J. (2004). The case for separating routing from routers. *Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, p. 5–12.

[Filsfils et al., 2015] Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Telecom, B., Ytti, S., Henderickx, W., Tantsura, J. e Crabbe, E. (2015). Segment Routing interoperability with LDP. https://www.ietf.org/archive/id/draft-filsfils-spring-segment-routing-ldp-interop-03.txt.

[Francois et al., 2015] Francois, P., Filsfils, C., Bashandy, A., Decraene, B. e Litkowski, S. (2015). Topology Independent Fast Reroute using Segment Routing. https://www.ietf.org/archive/id/draft-francois-spring-segment-routing-ti-lfa-02.txt.

[Gude et al., 2016] Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N. e Shenker, S. (2016). NOX: Towards an operating system for networks. http://www.cs.yale.edu/homes/jf/nox.pdf.

[Haleplidis et al., 2015] Haleplidis, E., Hadi Salim, J., Denazis, S. e Koufopavlou, O. (2015). Towards a network abstraction model for SDN. *Computer Communications*, 32:309–327.

[Hodzic e Zoric, 2008] Hodzic, H. e Zoric, S. (2008). Traffic engineering with constraint based routing in MPLS networks. Em *International Symposium ELMAR*, p. 269–272.

[Hu et al., 2014] Hu, F., Hao, Q. e Bao, K. (2014). A survey on software-defined network and openflow: From concept to implementation. *IEEE Communications Surveys & Tutorials*, 16:2181 –2206.

[Huang et al., 2014] Huang, S., Griffioen, J. e Calvert, K. L. (2014). Network hypervisors: Enhancing sdn infrastructure. *Computer Communications*, 46:87–96.

[IETF MPLS documents, 2001] IETF MPLS documents (2001). MPLS IETF WG. Acessado em fev/2016 `https://datatracker.ietf.org/wg/mpls/documents/`.

[ISO/IEC 23271:2012, 2012] ISO/IEC 23271:2012 (2012). Information technology – Common Language Infrastructure (CLI). Acessado em mar/2016 `http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=58046`.

[J., 2011] J., K. (2011). The GMPLS controlled optical networks as industry communication platform. *IEEE Transactions on Industrial Informatics*, 7:671–678.

[Je e Ly, 2012] Je, B. e Ly, O. (2012). Next-generation optical network architecture and multidomain issues. *Proceedings of the IEEE*, 100:1130–1139.

[Jingjing et al., 2014] Jingjing, Z., Di, C., Weiming, W., Rong, J. e Xiaochun, W. (2014). The deployment of routing protocols in distributed control plane of sdn. *The Scientific World Journal*, 44:1–8.

[Kaur et al., 2016] Kaur, S., Singh, J., e Ghumman, N. S. (2016). Network programmability using POX controller. http://www.sbsstc.ac.in/icccs2014/Papers/Paper28.pdf.

[Kreutz et al., 2015] Kreutz, D., Ramos, F. M. V., Veríssimo, P. E., Rothenberg, C. E., Azodolmolky, S. e Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103:14–76.

[Lara et al., 2014] Lara, A., Kolasani, A. e Ramamurthy, B. (2014). Network innovation using openflow: A survey. *IEEE Communications Survey & Tutorials*, 16:493–512.

[Marzo et al., 2003] Marzo, J. L., Calle, E., Scoglio, C. e Anjah, T. (2003). QoS online routing and MPLS multilevel protection: a survey. *IEEE Communications Magazine*, 41:126–132.

[Matias et al., 2015] Matias, J., Garay, J., Toledo, N., Unzilla, J. e Jacob, E. (2015). Toward an SDN-enabled NFV architecture. *IEEE Communications Magazine*, 53:187–193.

[Medved et al., 2014] Medved, J., Varga, R., Tkacik, A. e Gray, K. (2014). Opendaylight: Towards a model-driven SDN controller architecture. volume 46, p. 1–6.

[Mininet, 2015] Mininet (2015). Mininet tutorial. Acessado em `https://github.com/mininet/mininet/wiki/Teaching-and-Learning-with-Mininet`.

[Nunes et al., 2014] Nunes, B., Antipolis, S., Mendonca, M., Nguyen, X. N. e Obraczka, K. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, 16:1617–1634.

[Nwa et al., 2010] Nwa, S., Ju, L., Martin, R. e Phil, S. (2010). Automating network and service configuration using netconf and yang. *IEEE Communications Magazine*, 48:166–173.

[Osborne e Simha, 2002] Osborne, E. e Simha, A. (2002). *Traffic Engineering with MPLS*. Cisco Press, 2 edição.

[OSHI, 2015] OSHI (2015). OSHI - open source hybrid IP/SDN tutorial. Acessado em `http://netgroup.uniroma2.it/OSHI`.

[OSHI e Mininet, 2015] OSHI e Mininet (2015). OSHI - open source hybrid IP/SDN networking and its emulation on Mininet and on distributed SDN testbeds. Acessado em `htttp://netgroup.uniroma2.it/twiki/bin/view/Oshi/WebHome#AnchorSegRouting`.

[Paolucci et al., 2013] Paolucci, F., Cugini, F., Giorgetti, A., Sambo, N. e Castoldi, P. (2013). A survey on the path computation element (PCE) architecture. *IEEE Communications Surveys & Tutorials*, 15:1819–1841.

[Pepelnjak e Guichard, 2003] Pepelnjak, I. e Guichard, J. (2003). MPLS and VPN architectures. volume 1. Cisco Press.

[Pingping et al., 2015] Pingping, L., Jun, B., Stephen, W., Yangyang, W., Anmin, X., Ze, C., Hongyu, H., Yikai, L. e Pingping, L. (2015). A west-east bridge based sdn inter-domain testbed. *IEEE Communications Magazine*, 53:190–197.

[Previdi et al., 2015a] Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B. e Tantsura, J. (2015a). IS-IS Extensions for Segment Routing. https://tools.ietf.org/html/draft-ietf-isis-segment-routing-extensions-06.

[Previdi et al., 2015b] Previdi, S., Filsfils, C., Field, B., Leung, I., Linkova, J., Aries, E., an E. Vyncke, T. K. e Lebrun, D. (2015b). IPv6 Segment Routing Header (SRH). https://www.ietf.org/archive/id/draft-previdi-6man-segment-routing-header-08.txt.

[RFC1157, 1990] RFC1157 (1990). A Simple Network Management Protocol (SNMP). Acessado em mar/2016 `http://www.ietf.org/rfc/rfc1157.txt?number=1157`.

[RFC5440, 2009] RFC5440 (2009). Path Computation Element (PCE) Communication Protocol (PCEP). Acessado em mar/2016 `https://tools.ietf.org/html/rfc5440`.

[RFC6020, 2010] RFC6020 (2010). YANG - a data modeling language for the network configuration protocol (NETCONF). Acessado em mar/2016 `https://tools.ietf.org/html/rfc6020`.

[RFC6241, 2011] RFC6241 (2011). Network Configuration Protocol (NETCONF). Acessado em mar/2016 `https://tools.ietf.org/html/rfc6241`.

[RFC7047, 2013] RFC7047 (2013). The Open vSwitch Database Management Protocol. Acessado em mar/2016 `https://tools.ietf.org/html/rfc7047`.

[RFC7752, 2016a] RFC7752 (2016a). North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP). Acessado em mar/2016 `https://www.rfc-editor.org/rfc/pdfrfc/rfc7752.txt.pdf`.

[RFC7752, 2016b] RFC7752 (2016b). North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP. https://datatracker.ietf.org/doc/rfc7752/.

[Rose, 2014] Rose, E. (2014). BRKMPL-1101 - understanding MPLS. Acessado em fev/2016 `https://www.ciscolive.com/online/connect/sessionDetail.ww?SESSION_ID=77795&backBtn=true`.

[Rothenberg et al., 2011] Rothenberg, C. E., Nascimento, M. R., Salvador, M. R. e Magalhães, M. F. (2011). Openflow e redes definidas por software: um novo paradigma de controle e inovação em redes de pacotes. *Cad. CPqD Tecnologia*, 7:65–76.

[Rowshanrad et al., 2014] Rowshanrad, S., Namvarasl, S., Abdi, V., Hajizadeh, M. e Keshtgary, M. (2014). A survey on SDN, the future of networking. *Journal of Advanced Computer Science & Technology*, 3:232–248.

[Sadok e Kamienski, 2000] Sadok, D. e Kamienski, C. A. (2000). Qualidade de serviço na internet. Em *Minicursos do XVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, p. 4–44.

[Salsano et al., 2014a] Salsano, S., Pier, Ventre, L., Prete, L., Siracusano, G., Gerola, M., Salvadori, E., Santuari, M., Mauro, Campanella e Prete, L. (2014a). OSHI - open source hybrid IP/SDN networking and Mantoo - management tools for SDN experiments. Em *European Workshop on Software Defined Networks (EWSDN)*, p. 123–124.

[Salsano et al., 2014b] Salsano, S., Ventre, P. L., Prete, L., Siracusano, G., Gerola, M. e Salvadori, E. (2014b). OSHI - open source hybrid IP/SDN networking (and its emulation on mininet and on distributed SDN testbeds). Em *European Workshop on Software Defined Networks (EWSDN)*, p. 13–18.

[Santos et al., 2007] Santos, C. B., Fernandes, D. C. e Marchetti, B. R. B. (2007). Tutoriais TELECO MPLS: Re-roteamento dinâmico em redes IP utilizando network simulator. Acessado em fev/2016 `http://www.teleco.com.br/tutoriais/tutorialmplsrd/default.asp`.

[Scott-Hayward, 2015] Scott-Hayward, S. (2015). Design and deployment of secure, robust, and resilient sdn controllers. Em *IEEE Network Softwarization (NetSoft), 2015 1st IEEE Conference on*, p. 1–5.

[Sgambelluri et al., 2015a] Sgambelluri, A., Giorgetti1, A., Cugini, F., Bruno, G., Lazzeri, F. e Castoldi, P. (2015a). First demonstration of SDN-based segment routing in multi-layer networks. Em *Optical Fiber Communications Conference and Exhibition (OFC)*, p. 1–3.

[Sgambelluri et al., 2015b] Sgambelluri, A., Paolucci, F., Giorgetti, A., Cugini, F. e Castoldi, P. (2015b). Experimental demonstration of segment routing. *Journal of Lightwave Technology*, PP:1–1.

[SPRING, 2015] SPRING (2015). Source packet routing in networking (SPRING). Acessado em `https://datatracker.ietf.org/wg/spring/documents/`.

[Stancu et al., 2015] Stancu, A. L., Halunga, S., Vulpe, A., Suciu, G., Fratu, O. e Popovici, E. C. (2015). A comparison between several software defined networking controllers. *Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS), 2015 12th International Conference on*, p. 223–226.

[Systems, 2001] Systems, C. (2001). Implementing MPLS traffic engineering. Acessado em fev/2016 `http://www.cisco.com/c/en/us/td/docs/routers/crs/software/crs_r3-/mpls/configuration/guide/gc39crs1book_chapter4.html`.

[Telcordia GR-831, 1996] Telcordia GR-831 (1996). Operations Application Messages - Language For Operations Application Messages. Acessado em mar/2016 `http://telecom-info.telcordia.com/site-cgi/ido/docs.cgi?ID=SEARCH&DOCUMENT=GR-831&`.

[Xia et al., 2015] Xia, W., Wen, Y., Foh, C., Niyto, D. e Xie, H. (2015). A survey on software-defined networking. *IEEE Communications Surveys & Tutorials*, 17:27–51.

[Xiao et al., 2000] Xiao, X., Hannan, A., Bailey, B. e Ni, L. M. (2000). Traffic engineering with MPLS in the internet. *IEEE Network*, 14:28–33.

[Xiea et al., 2015] Xiea, J., Guo, D., Hua, Z., Qua, T. e Lvb, P. (2015). Control plane of software defined networks: A survey. *Computer communications*, 67:1–10.