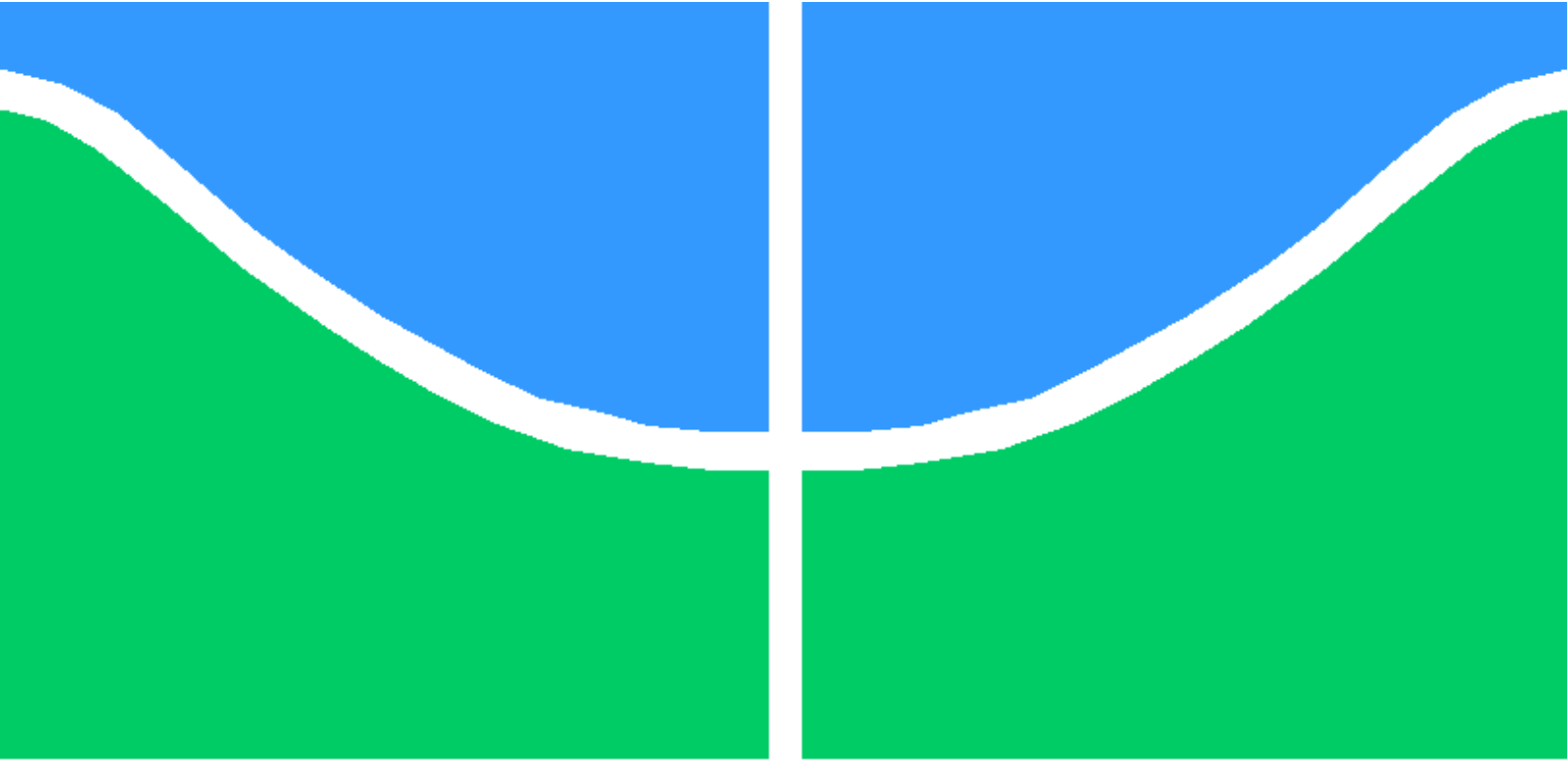

TESE DE DOUTORADO

PROPOSTA DE ARQUITETURA PARA PROVEDORES DE SERVIÇOS DE REDES FIXAS E MÓVEIS UTILIZANDO SDN OPENFLOW

Msc. Fernando López Rodríguez

Brasília, julho de 2019

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

PROPOSTA DE ARQUITETURA PARA PROVEDORES DE SERVIÇOS DE REDES FIXAS E MÓVEIS UTILIZANDO SDN OPNFLOW

FERNANDO LÓPEZ RODRÍGUEZ

ORIENTADOR: UGO SILVA DIAS
COORIENTADOR: DIVANILSON RODRIGO DE SOUSA CAMPELO

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO: PPGENE.TD - 148/2019

BRASÍLIA/DF: JULHO – 2019

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

PROPOSTA DE ARQUITETURA PARA PROVEDORES DE SERVIÇOS DE REDES FIXAS E MÓVEIS UTILIZANDO SDN OPENFLOW

FERNANDO LOPEZ RODRIGUEZ

TESE DE DOUTORADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR.

APROVADA POR:

UGO SILVA DIAS, Dr., ENE/UNB
(PRESIDENTE DA COMISSÃO)

ADONIRAN JUDSON DE BARROS BRAGA, Dr., ENE/UNB
(EXAMINADOR INTERNO)

AGOSTINHO LINHARES DE SOUZA FILHO, Dr., ANATEL
(EXAMINADOR EXTERNO)

JOEL JOSÉ PUGA COELHO RODRIGUES, Dr., INATEL
(EXAMINADOR EXTERNO)

Brasília, 22 de julho de 2019.

FICHA CATALOGRÁFICA

LOPEZ RODRIGUEZ, FERNANDO

Proposta de Arquitetura para Provedores de Serviços de Redes Fixas e Móveis Utilizando SDN OpenFlow [Distrito Federal] 2019.
124p., 210 x 297 mm (ENE/FT/UnB, Doutor, Tese de Doutorado – Universidade de Brasília.

Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

1.Redes definidas por software	2.MPLS
3.Arquitetura 5G	4.NFV
5.Arquitetura para redes de transporte	6.Deteção de tráfego QoS
7. Gerenciamento de QoS	
I. ENE/FT/UnB	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

LOPEZ., F. (2019). Proposta de Arquitetura para Provedores de Serviços de Redes Fixas e Móveis Utilizando SDN OpenFlow. Tese de Doutorado em Engenharia Elétrica, Publicação PPGENE.TD-148/2019, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 124p.

CESSÃO DE DIREITOS

AUTOR: Fernando López Rodríguez.

TÍTULO: Proposta de Arquitetura para Provedores de Serviços de Redes Fixas e Móveis Utilizando SDN OpenFlow.

GRAU: Doutor ANO: 2019

Fernando López Rodríguez
CPF: 701.290.151-99
RNE: G296173-2

Dedicatória

A minhas filhas Sofía Ema e María Eugenia.

Msc. Fernando López Rodríguez

Agradecimentos

Ás minhas filhas e esposa, por sua compreensão e apoio durante as longas jornadas de estudo.
Á minha família toda que é a grande motivadora para minha carreira.
À minha esposa María Noel por toda sua ajuda durante estes anos de trabalho participando de todas as correções realizadas.
Ao Professor Doutor Ugo Silva Dias, por seus importantes aportes e conselhos durante estes anos.
Ao Professor Doutor Divanilson Rodrigo de Sousa Campelo por aportes e ter me motivado para iniciar este longo caminho.
Ao Johnathan Dezan Vago que achou interessante minha pesquisa e realizou testes adicionais que contribuíram com meu trabalho.
Aos funcionários do Departamento de Engenharia Elétrica da UnB, pelo apoio nas necessidades administrativas durante o Doutorado.
Ao querido Brasil, que me deu a possibilidade de me especializar nesta importante Universidade.

Msc. Fernando López Rodríguez

RESUMO

As redes de grande porte, como Provedores de Serviço, são arquiteturas robustas, capazes de dar suporte a grandes volumes de tráfego com características muito diferentes. Seus equipamentos dão suporte a cargas elevadas de processamento e ao mesmo tempo, são responsáveis por construir a lógica de roteamento e por encaminhar o tráfego. Por terem o controle implementado de forma distribuída e por serem construídas com equipamentos de um limitado número de fabricantes, estas redes apresentam limitações de controle e engenharia de tráfego, dificultando assim, a diferenciação entre os serviços que os diversos provedores fornecem. Adicionalmente, a inteligência da rede está oculta nos equipamentos, tornando as inovacões muito lentas e amarradas aos interesses dos fabricantes. Como alternativa a este cenário, este trabalho propõe uma arquitetura de rede SDN-*OpenFlow* que tenta solucionar os problemas previamente mencionados, bem como os inconvenientes da característica centralizadora que o *OpenFlow* possui. É apresentada uma arquitetura de rede *OpenFlow* robusta, capaz de dar suporte a tempos de resposta elevados e a quedas do Controlador, sem adição de tempos de espera no estabelecimento de novos fluxos e com significativa redução na carga submetida ao Controlador. Como prova de conceito, é implementado um protótipo utilizando o OpenvSwitch como *software* para a virtualização dos clientes *OpenFlow*, o Mininet para a criação da topologia e o Ryu como Controlador, todos com suporte *OpenFlow* 1.3. ou superior.

ABSTRACT

Large scale networks, such as Service Providers, are robust architectures, capable of supporting large volumes of traffic with very different characteristics. Their network equipment have significant processing load, being responsible for building both a routing logic and a routing traffic at the same time. By having the network control implemented in a distributed manner and being built with a limited number of vendors, these networks have limitations of control and traffic engineering, hindering the differentiation between Service Providers. Additionally, the network intelligence is hidden in the network equipment, making the innovations very slow and conditioned to the vendors interests. As an alternative option, this work proposes an SDN-*OpenFlow* network architecture that tries to improve the previously mentioned problems, and at the same time solves the arising difficulties related to the SDN network centralizing feature. With the proposed architecture, a robust *OpenFlow* network is created to support high Controller response times and Controller shut down, without additional delays in the creation of flows and with significant reduction of Controller's load. A prototype has been constructed using Open vSwitch as a virtualization software for *OpenFlow* clients, Mininet for the topology construction and Ryu as the Controller, all with *OpenFlow* 1.3 support or higher.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SÍMBOLOS

Siglas

3GPP	3rd. Generation Partnership Project
ACL	Access Control List
ADC	Application Detection and Control
AMF	Access and Mobility Management Function
AF	Application Function
AMBR	Aggregate Maximum Bit Rate
API	Application Programming Interface
ARP	Allocation and Retention Priority
ATM	Asynchronous Transfer Mode
BBERF	Bearer Binding and Event Reporting Function
BGP	Border Gateway Protocol
CDR	Charging Data Record
CR-LDP	Constraint-based Routing - Label Distribution Protocol
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DoS	Denial Of Service
DSCP	Differentiated Services Code Point
E-UTRAN	Evolved UMTS Terrestrial Radio Access Networks
eNB	Evolved eNodeB
EPC	Evolved Packet Core
EPS	Evolved Packet System
ER-LSP	Explicitly-Routed Label Switched Path
EXP	Experimental Bits
FEC	Forwarding Equivalence Class
FIB	Forwarding Information Base
gNB	next Generation eNodeB
GRB	Guaranteed Rate Bit
GTP	GPRS Tunneling Protocol
HSI	High Speed Internet
HSS	Home Subscriber Server
IMS	IP Multimedia Subsystem
ISIS	Intermediate System to Intermediate System

ISIS-TE	ISIS - Traffic Engineering
IP	Internet Protocol
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
LAN	Local Area Network
LDP	Label Distribution Protocol
LER	Label Edge Router
LSP	Label-Switched Path
LSR	Label Switch Router
LTE	Long-Term Evolution
MBR	Maximum Bit Rate
MME	Mobility Management Entity
MPLS	Multiprotocol Label Switching
MPLS-TE	MPLS - Traffic Engineering
MPLS-VPN	MPLS - Virtual Private Network
NFV	Network Function Virtualization
NG-RAN	Next Generation Radio Access Network
OCS	Online Charging System
OF	OpenFlow
OF-GW	OpenFlow Gateway
OF-NB	OpenFlow eNB or OpenFlow gNB
OF-Switch	OpenFlow Switch
OFCS	Offline Charging System
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
OSPF-TE	OSPF - Traffic Engineering
P-GW	PDN Gateway
PCC	Policy and Charging Control
PCF	Policy COntrol Function
PCEF	Policy and Charging Enforcement Function
PCRF	Policy and Charging Rules Function
PDN	Packet Data Network
PGW	PDN Gateway
PGW-C	PDN Gateway Control
PHP	Penultimate Hop Popping
POP	Points of Presence
QoS	Quality of Service
QSI	QoS Class Identifier
RIB	Routing Information Base
RSVP	Resource Reservation Protocol
S-GW	Serving Gateway
SDD	Service Detector Device
SDF	Service Data Flow

SDN	Software Defined Network
SDP	Session Description Protocol
DSGW	Serving Gateway
SIP	Session Initiation Protocol
SMF	Service Management Function
SP	Service Provider
SPR	Subscriber Profile Repository
STP	Spanning Tree Protocol
TC	Traffic Class
TCP	Transport Control Protocol
TDF	Traffic Detection Function
TDM	Time Division Multiplexing
TE	Traffic Engineering
TEID	Tunnel Endpoint Identifier
TFT	Traffic Flow Template
TLS	Transport Layer Security
ToS	Type of Service
TTL	Time to Live
UDM	Unified Data Management
UDP	User Datagram Protocol
UDR	User Data Repository
UE	User Equipment
UMTS	Universal Mobile Telecomminication Syestem
Uu	UMTS air interface
VC	Virtual Circuits
VLAN	Virtual Local Area Network
VM	Virtual Machine
VNF	Virtual Network Function
VoIP	Voice over IP
VPLS	Virtual Private LAN Services
VPN	Virtual Private Network
VPRN	Virtual Private Routed Network
VRF	Virtual Routing and Forwarding
WDM	Wavelength Division Multiplexing

Capítulo 1

Introdução

1.1 Definição do problema

Os Provedores de Serviço (*Service Providers* - SP) são organizações que comercializam acesso a Internet e serviços associados tanto para empresas e usuários finais, quanto para outros provedores. Em geral, um SP compra um serviço denominado trânsito IP de outros SP já conectados à rede; depois de estabelecida a relação de trânsito, um SP pode se conectar à Internet Global através de outro SP. Os provedores são os blocos fundamentais da arquitetura da Internet, compostos por um amplo número de equipamentos de rede como roteadores, *switches*, estações rádio-base, elementos lógicos, comutadores óticos assim como enlaces e servidores entre outros elementos.

O accesso à Internet ou a uma rede privada é oferecido a través de grande diversidade de alternativas de interconectividade como a rede de acceso móvel (domínio móvel de SP), rede de acceso fixa (domínio fixo), ou através de DataCenter (dómínio de DataCenter). Todas estas alternativas de acesso se interconectam entre elas mediante a rede de transporte do SP (domínio de transporte), que adicionalmente aos serviços de interconetividade oferecidos aos diferentes domínios, é encarregado de proporcionar acesso à Internet.

Além dos serviços de acesso, os provedores cada vez mais oferecem uma diversidade de serviços associados, como serviços de VoIP (*Voice over IP*), televisão digital, chamadas de vídeo, jogos *online*, interconexões privadas, serviços de núvem, entre outros. Todos estes diversos requisitos fazem que os SP devam dar suporte a uma grande variedade de serviços com requisitos de QoS (*Quality of Service*) diferentes, os quais, adicionados ao grande volume de tráfego e à dimensão das redes dos SP, tornam o gerenciamento destas redes uma complexa tarefa.

As redes de transporte dos SP possuem mecanismos de controle automáticos e totalmente distribuídos, tais como protocolos de roteamento (por exemplo, *Border Gateway Protocol* – BGP, *Open Shortest Path First* – OSPF, *Intermediate System to Intermediate System* – ISIS) e de sinalização (por ex., *Label Distribution Protocol* – LDP, *Resource Reservation Protocol* – RSVP). Estes mecanismos de controle geram arquiteturas robustas, capazes de recalcular automaticamente sua topologia com a queda de qualquer um dos seus equipamentos. Contudo, os mecanismos geram equipamentos mais complexos, responsáveis por construir tabelas de roteamento e pelo

encaminhamento do tráfego ao mesmo tempo, o que resulta em equipamentos de rede com altas cargas de processamento.

No que refere às redes móveis, existe uma separação do plano de dados e do plano de controle nas interfaces e nos protocolos. Entretanto, todas as interfaces são implementadas em *hardware* proprietário, o que incrementa os custos e dificulta a melhora e/ou a introdução de novos serviços. Adicionalmente, a arquitetura das redes móveis gera centralização de suas funcionalidades. Particularmente todas as políticas são decididas pelo PCRF (*Policy and Charging Control*), e o monitoramento e a implementação de QoS depende exclusivamente do PDN-GW (ou P-GW), o que introduz desafios de escalabilidade importantes nestes nós.

Como características adicionais, as redes de SP apresentam custos elevados em infraestrutura e operação, tendo que manter grupos de administradores diferentes para cada tipo de redes como rede IP (*Internet Protocol*), redes WDM (*Wavelength Division Multiplexing*), redes móveis, redes de acesso fixo e DataCenter. Os equipamentos utilizados são de um grupo reduzido de fabricantes e possuem um número de funcionalidades limitadas reduzindo sensívelmente as possibilidades de diferenciação e engenharia de tráfego. A inteligência da rede está oculta nos equipamentos, tornando as inovações extremamente lentas e amarradas aos interesses dos fabricantes.

O caminho alternativo ao controle distribuído (redes atuais) são as redes definidas por *software* SDN (*Software Defined Networks*), que têm como proposta separar o controle das redes para um dispositivo central denominado Controlador. Dentro destas encontra-se o protocolo *OpenFlow* [1, 2, 3], que proporciona uma arquitetura centralizada com uma interface aberta e padronizada, através da qual os equipamentos clientes podem interagir com um Controlador dotado de uma visão global da rede e encarregado de construir as tabelas de fluxo para o roteamento de pacotes em todos os equipamentos de rede (clientes *OpenFlow*). Dessa forma, é possível separar o plano de dados (constituido pelos clientes *OpenFlow*) do plano de controle (Controlador). O protocolo *OpenFlow* proporciona inúmeras possibilidades para engenharia de tráfego [4, 5, 6], unificação do plano de controle para diferentes tipos de redes, como IP e WDM [7, 8, 9, 10], alternativas para a computação em núvem [11, 12], gerenciamento de mobilidade [13, 14], entre outras. Tais possibilidades têm atraído o interesse de diferentes fabricantes de redes, que já estão incorporando *OpenFlow* em seus produtos [15].

No entanto, apesar das vantagens indicadas no parágrafo anterior o *OpenFlow* possui desvantagens [16]. Particularmente para um SP, a arquitetura *OpenFlow* padrão não é apropriada em razão de:

- sua característica centralizadora que gera dependência excessiva do Controlador, o que reduz em grande medida a robustez da rede;
- o requisito de que cada pacote de um novo fluxo seja processado e encaminhado pelo Controlador, gerando assim possíveis sobrecargas neste;
- a degradação de desempenho, devido a que cada fluxo com requrimentos específicos de QoS (*Quality of Service*) tem que aguardar a resposta do Controlador para seu roteamento.

1.2 Objetivos do trabalho

Este trabalho propõe duas arquiteturas, uma para o domínio de transporte do SP, e outra para o domínio móvel. No que refere ao domínio móvel, a arquitetura proposta oferece um plano de dados simplificado que reduz as fortes exigências sobre elementos como o P-GW e PCRF, incrementa as posibilidades de gerenciamento oferecendo serviços com garantia de QoS fim a fim, proporcionando um controle mais flexível com uma comunicação padronizada e aberta que gera um ambiente propício para as inovações. Adicionalmente, a simplificação e a estandarização do plano de controle da rede móvel reduz a dependência com os fabricantes e diminui os custos dos equipamentos utilizados.

No que refere à arquitetura para a rede de transporte proposta, é elaborada uma alternativa SDN-*OpenFlow* robusta, capaz de dar suporte a tempos de resposta elevados e a quedas do Controlador. Isto se consegue sem adição de tempos de espera no estabelecimento de novos fluxos e com significativa redução na carga submetida ao Controlador, tanto para o tráfego com e sem requerimentos de QoS. Ao mesmo tempo a arquitetura proposta é capaz de aprimorar as limitações que as redes atuais com controle distribuido possuem como limitações na engenharia de tráfego, inovações amarradas aos interesses dos fabricantes, equipamentos de rede mais complexos e com maior processamento.

Utilizando os conceitos de SDN e NFV, este trabalho propõe uma arquitetura para a rede móvel que aborda com uma nova metodologia a detecção de tráfego QoS e seu gerenciamento, dois dos requerimentos principais da próxima geração de redes 5G. Se o tráfego com requerimentos de QoS é detectado de forma mais rápida e eficiente, este poderá ser processado rápidamente. Por tanto, com um tratamento de QoS especializado, pode ser criado um fluxo de QoS ótimo, fim a fim, que cumpra tais características. Os resultados mostram que a lógica proposta é flexível e programavel, aplicada muito próximo do lugar onde o tráfego é gerado (próximo dos UE – *User Equipment*), e que herda as possibilidades de engenharia de tráfego das arquiteturas SDN.

A seguir são listados cada um dos objetivos do trabalho com maior detalhe:

A. Criar uma proposta de arquitetura SDN-*OpenFlow* para redes de transporte de SP e com o plano de dados MPLS (*Multiprotocol Label Switching*) com as seguintes características:

   - Maiores possibilidades de engenharia de tráfego comparada com as redes de transporte atuais.
   - Utilização de mecanismos que proporcionem garantia de QoS dentro da rede de transporte.
   - Não adição de atrasos na criação de novos fluxos.
   - Baixa carga no Controlador comparada com a alternativa SDN-*OpenFlow* padrão.
   - Utilização de equipamentos de redes mais simples e com menor carga de processamento do que os utilizados nas redes atuais.
   - Sem mecanismos de distribuição de informação complexos que incrementam os tempos de convergência da rede.

- Redução sensivel da dependência da rede no Controlador, criando assim uma rede SDN-*OpenFlow* mais robusta, capaz de continuar sua operação durante quedas ou tempos de resposta altos do Controlador.
- Protocolos de comunicação padrão abertos que facilitem a substituição de equipamentos de diferentes fabricantes sem inconvenientes.
- Incrementação da velocidade de inovação, permitindo o aporte de toda a comunidade cientifica.
- Criação de um protótipo que implemente a lógica proposta como prova de conceito.

B. Criar uma proposta de arquitetura SDN-*OpenFlow* para redes móveis do SP com o plano de dados MPLS com as características listadas as seguir:

- Garanta requisitos de QoS dentro da rede móvel, gerenciando todos os equipamentos lógicos e físicos envolvidos no trajeto fim a fim.
- Criar um mecanismo de deteção de tráfego prográmavel e flexível que ao mesmo tempo se encontre distribuido na rede.
- Tentar evitar que as interfaces entre o plano de dados e de controle da rede móvel sejam implementadas em hardware proprietário.
- Facilitar a incorporação de novos serviços e funcionalidades.
- Reduzir a centralização de diferentes funcionalidades em equipamentos como o caso da definição de políticas nos PCRF e o monitoramento e implementação de QoS nos PDN-GW.
- Utilizar protocolos de comunicação abertos e padrão que facilitem a substituição de equipamentos de diferentes fabricantes sem inconvenientes.
- Proporcionar serviços de mobilidade dentro de um domínio móvel e entre domínios móveis do mesmo SP transparentes para os UE no processo.
- Incrementar a velocidade de inovação para este tipo de redes, que permita se adaptar às fortes exigências que estas possuem.
- Proporcionar caminhos otimizados para a comunicação com requisito de QoS entre dois UE dentro do mesmo domínio.
- Não adição de atrasos na criação de fluxos (*bearers* específicos), com a finalidade de garantir requisitos QoS.
- Proporcionar novos mecanismos de detecção de sinalização e de detecção de requisitos de QoS localizados nos eNodeB e nos P-GW, que permitam de forma dinâmica, flexível e ampla a detecção dos diferentes serviços. Adicionalmente esta detecção é realizada sem adição de atrasos e mediante a consideração do perfil de cada usuário que permite otimizar o processo.
- Criar um protótipo que implemente a lógica proposta como prova de conceito.

C. Proporcionar uma interação entre as redes móveis e a rede de transporte do mesmos SP, com as características listadas a seguir:

- Permitir a criação de serviços pre-definidos na rede de transporte para ser utilizados pelas redes móveis exigindo o mínimo de interação entre os Controladores dos diferentes tipos de redes.
- Evitar parcialmente o problema de triangulação no encaminhamento do tráfego quando o UE se encontra fora do seu próprio domínio móvel, mas dentro de outro domínio móvel do mesmo SP.
- Proporcionar garantias de QoS fim a fim em uma comunicação entre dois UEs pertenecentes a diferentes domínios móveis do mesmo SP, os quais estão interconectadas através da rede de transporte. Esta comunicação envolve todos os equipamentos do caminho, assim como os diferentes tipos de redes que requeiram ser utilizadas.
- Aprimorar os mecanismos de encaminhamento entre domínios.
- Incrementar as possibilidades de engenharia de tráfego entre as redes móveis e a rede de transporte do SP.

1.3 Principias contribuições da Tese

No transcurso do trabalho é desenvolvida a arquitetura proposta e é mostrado como os objetivos previamente indicados são atingidos. Nesta seção destacam-se as principais contribuições do trabalho.

A proposta propõe o encaminhamento de pacotes com requerimento de QoS por duas linhas de processamento simultaneamente. Uma encarregada do encaminhamento imediato como tráfego sem requerimentos de QoS, e outra que encaminha uma cópia do pacote para seu processamento que permite construir caminhos ótimos que cumpram com todos os requerimentos de QoS.

Adicionalmente destaca-se como a lógica proposta consegue encaminhar os pacotes por duas linhas de processamento, mas em nenhum caso os dois métodos são utilizados para o envio de um mesmo pacote. Os primeiros pacotes serão só encaminhados pelas regras gerais, e só quando o caminho ótimo seja criado os sucessivos pacotes do mesmo fluxo serão enviados pelo caminho ótimo.

Destaca-se assim a técnica de processamento QoS, tanto para redes de transporte quanto para redes móveis. Esta cria um método de encaminhamento que é simultaneamente proativo, utilizando regras gerais pre-configuradas, e reativo, que permite estabelecer caminhos ótimos fluxo a fluxo. Ambos os métodos são simultaneamente aplicados sobre os mesmos pacotes.

A proposta realizada apresenta uma alternativa de implementação mediante equipamentos híbridos (os que implementam o controle centralizado e distribuído simultaneamente). Em esta lógica, os pacotes inicialmente processados pelo plano de controle centralizado podem ser encaminhados ao destribuído utilizando a instruções do tipo *Output* a porta reservada *Normal*. Esta alternativa combina as vantagens das arquiteturas distribuídas e as arquiteturas centralizadas, criando uma solução robusta, capaz de continuar operativa ainda com quedas do Controlador ou tempos de resposta elevados. Ao mesmo tempo incorpora a flexibilidade e programabilidade que

as arquiteturas SDN possuem.

O mecanismo de deteção de tráfego para as redes móveis proposto é também um dos focos do trabalho, proporcionando um método flexível, programável e adaptável composto por dos níveis para a deteção do tráfego com requerimentos de QoS. O primeiro é construído de forma distribuída trasladando a deteção do tráfego aos extremos da rede móvel nos eNodeB. E o segundo é implementado de forma parcialmente centralizada, permitindo detetar requerimentos de QoS mediante entidades criadas para esse fim.

1.4 Publicações realizadas

As principais contribuições referentes à arquitetura para redes de transporte proposta foram publicadas na conferencia GLOBCOM, no trabalho denominado: *A Robust SDN Network Architecture for Service Providers* [17].

Por outra parte, as contribuições referentes à arquitetura para redes móveis propostas foram publicadas na revista SENSOR, no trabalho denominado: *QoS Management and Flexible Traffic Detection Architecture for 5G Mobile Networks* [18].

1.5 Apresentação do manuscrito

O presente trabalho é organizado conforme descrito a seguir:

- O Capítulo 2 começa realizando uma divisão da rede de SP em domínios de controle, fundamental para a construção das arquiteturas apresentadas ao longo do trabalho. Posteriormente é apresentado um caso de uso de SP, o que proporciona uma base de referência para o trabalho. Finalmente é realizado um referencial teórico com todos os conceitos prévios requeridos para a compreensão das propostas. Isto inclui, conceitos de SDN e particularmente uma descrição do protocolo *OpenFlow* (o escolhido para a contrução das arquiteturas propostas no trabalho), características de MPLS com suas aplicações mais relevantes, e a descrição geral de todos os aspectos referentes às redes 4G utilizadas no trabalho.
- O Capítulo 3 aborda a arquitetura proposta para a rede de transporte do SP (ou domínio de transporte), a responsável da interconexão dos diferentes domínios do SP. O capítulo começa com uma descrição da lógica geral de funcionamento, os mecanismos proativos que permitem alcançar todas as redes destino para o encaminhamento, e a lógica proposta para o gerenciamento do tráfego com requisitos de QoS. A arquitetura apresentada gera uma rede robusta, com menor dependência do Controlador e sem adição de tempos de espera na criação de novos fluxos. Também no capítulo é sugerida uma arquitetura alternativa que utiliza equipamentos híbridos, capazes de implementar simultâneamente o controle centralizado *OpenFlow* e o controle distribuído atual.
- O Capítulo 4 aborda a construção da arquitetura para a rede móvel do SP (denominada

domínio móvel). Inicialmente são apresentadas as caraterísticas gerais e os mecanismos proativos para a construção da arquitetura. Seguidamente são apresentados os procedimentos para a criação dos enlaces de rádio iniciais, a metodologia para a contrução de caminhos com requisitos de QoS específicos, e mecanismos para a detecção da sinalização que permite conhecer os requisitos de QoS do serviço. Finalmente são abordados aspectos de mobilidade dentro do domínio e entre domínios móveis do mesmo SP, assim como alternativas para garantir os requisitos de QoS dos diferentes serviços fim a fim.

- O Capítulo 5 tem como objetivo a descrição dos *softwares* utilizados, assim como mostrar os protótipos desenvolvidos, os diferentes testes realizados e os principais resultados do trabalho mediante diversas comparativas. Isto é realizado tanto para um protótipo que implementa a arquitetura de transporte quanto para um protótipo que implementa a lógica de rede móvel proposta.
- O Capítulo 6 apresenta as principais conclusões do trabalho, suas limitações, e as perspectivas de trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo tem como objetivo apresentar os conceitos gerais considerados para a realização das propostas de arquitetura desenvolvidas no presente trabalho. Seguidamente é apresentado um caso de uso de rede SP que proporciona um ponto de referência para este trabalho. Finalmente é realizado um referencial teórico com os conceitos requeridos para facilitar a compreensão das diferentes temáticas abordadas no trabalho.

No que refere ao referêncial teórico, inicialmente são abordados conceitos relacionados à arquitetura SDN e particularmente ao protocolo *OpenFlow*, o protocolo escolhido para a realização das arquiteturas tanto para o domínio de transporte (capítulo 3) quanto para o domínio móvel (capítulo 4) do SP. A seguir são aprofundados conceitos sobre MPLS, a tecnologia escolhida para a implementação do plano de dados. Por fim, são aprofundados conceitos das redes móveis 4G atuais, que são utilizados para a construção da arquitetura móvel desenvolvida.

2.1 Conceitos gerais do trabalho

Como foi previamente indicado, os SP são os blocos fundamentais na arquitetura da Internet. A interconexão entre os SP junto com o protocolo de roteamento global BGP (*Border Gateway Protocol*) e protocolos de roteamento interno, permitem a troca de informações entre quaisquer pontos do mundo através da Internet. Desta forma, os SP têm a capacidade de proporcionar acceso à Internet aos seus clientes, o que pode ser oferecido por diferentes alternativas como conexões fixas, conexões móveis ou através de acessos proporcionados em DataCenters.

A conexão à Internet proporcionada, permite aos seus clientes acessar a inúmera variedade de serviços como navegação web, correio eletrônico, comunicações de VoIP e video, jogos *online*, *streaming* de vídeo, etc. Alguns de estes serviços não têm significativos requisitos de QoS, mas outros podem requerer largura de bandas mínimas garantidas, atrasos totais menores a um determinado valor, disponibilidade do serviço maior a uma determinada porcentagem, etc. Mas, em quais condições estes requisitos podem ser satisfeitos? Existem mecanismos que permitam garantir fim a fim estes requisitos nas redes dos SP atuais?

O presente trabalho propõe a realização de mudanças importantes nas redes atuais utilizando a arquitetura SDN-*OpenFlow*. Como será demonstrado, a arquitetura proposta incrementa sensívelmente as posibilidades de engenharia de tráfego, permitindo assim cumprir com os requisitos de QoS dos diferentes tipos de tráfego fim a fim. Para isto, a rede do SP é subdividida em domínios, criando mecanismos que diferenciam o tráfego e aplicam técnicas específicas para seu tratamento, em cada domínio. Com estas mudanças é criada uma rede mais flexível, com inúmeras possibilidades de gerenciamento e rápidamente adaptáveis aos novos requisitos de serviço que surgem constantemente.

Com o objetivo geral estabelecido, se subdividiu este objetivo em problemas de menor porte. Acredita-se que um Controlador ou grupo de Controladores, interatuando como um só, difícilmente possam gerenciar todos os elementos que compõem a rede de um SP. Mas por outra parte, a alternativa de subdividir um domínio de controle em subdomínios com Controladores em cada um deles, também apresenta significativas dificuldades. Esta última alternativa requer de singificativas interações entre os Controladores pertencientes aos subdomínios que compõem o caminho fim a fim, o que dificulta o processo e gera adicionais atrasos no encaminhamento dos novos fluxos.

A escolha dos subdomínios de controle é um dos aspectos chave na procura do objetivo, tendo que manter a independência entre os subdomínios para reduzir a comunicação necessária entre eles, mas ao mesmo tempo tem que permitir a criação de serviços que cumpram com os requisitos de QoS ainda entre domínios. Tendo isto em consideração, é estratégicamente escolhida uma divisão do domínio de controle do SP como é mostrado a seguir [19]:

- Domínio de acesso móvel do SP: é o responsável pelo gerenciamento da rede móvel do SP. Isto inclui a rede de acesso, *backhaul* e núcleo da rede móvel do SP assim como os elementos lógicos desta rede.
- Domínio de acesso fixo do SP: é o responsável pelo gerenciamento de todos os elementos que compõem a rede fixa de acesso do SP.
- Domínio de DataCenter do SP: é o domínio que pode brindar um lugar físico para a interconexão de clientes, e o responsável por brindar todos os serviços relacionados com a núvem.
- Domínio de transporte do SP: é o encarregado de proporcionar a interconetividade entre os diferentes domínios previamente indicados, e entre clientes de grande porte que podem estar conetados diretamente a este domínio. Adicionalmente, é o domínio encarregado de proporcionar acesso à Internet aos outros domínios e clientes.

A Figura 2.1 representa uma rede de SP que ilustra a subdivisão nos domínios previamente indicados. É importante notar que existem domínios móveis, domínios de Datacenter e domínios fixos isolados entre eles e gerenciados por Controladores indenpendentes. Todos estes domínios se encontram interligados através do domínio de transporte do SP, que proporciona interconectividade entre eles e acesso à Internet. Entretanto, a existência de interações entre Controladores de diferentes domínios possibilita a entrega de serviços otimizados fim a fim, mas estas interações

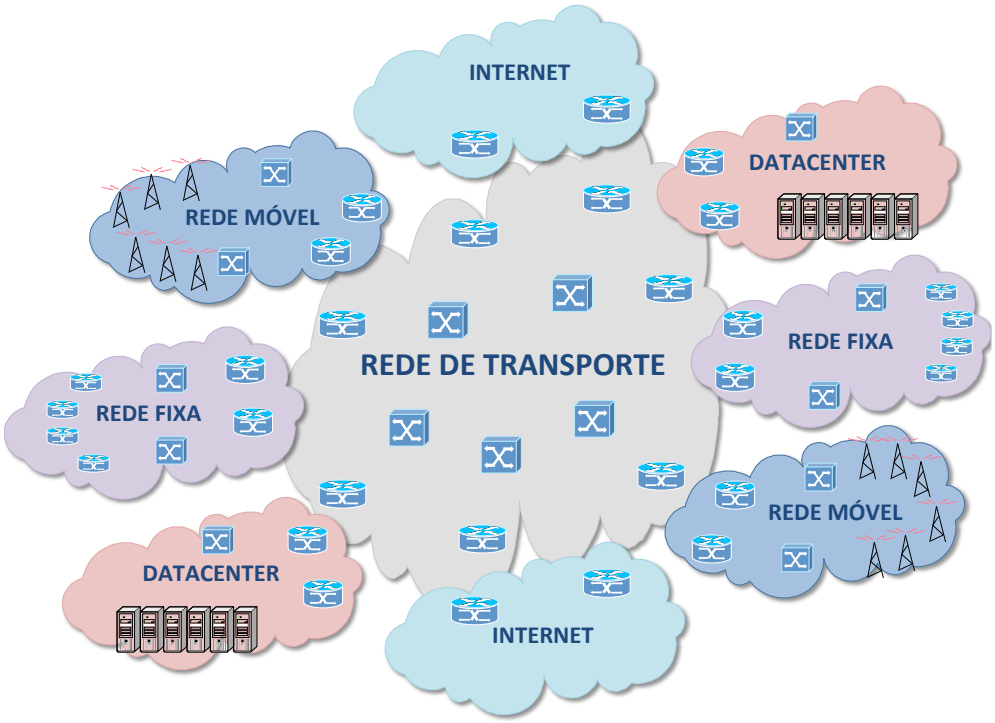

Figura 2.1: Visão geral de uma rede de SP e seus domínios de controle

devem ser proativas e reduzidas à menor quantidade possível. No capítulo 4 são indicados exemplos de esse tipo de interações tanto entre o domínio móvel e o de transporte preconfigurando serviços especializados quanto entre domínios móveis do mesmo SP para gerenciar os aspectos de mobilidade entre eles.

O trabalho cria a arquitetura para o gerenciamento de serviços para dois de estes domínios. Em primeira instância, para o domínio de transporte do SP (capítulo 3), e seguidamente para os domínios móveis do SP (capítulo 4). Com estes dois domínios interligados são criados serviços que cumprem com os requisitos de QoS fim a fim. Muitos dos conceitos que serão desenvolvidos podem ser aplicáveis ao domínio de acesso fixo do SP, mas este domínio é considerado fora dos objetivos do trabalho.

2.2 Características gerais da rede de SP considerada como caso de estudo

Nesta seção são descritas as características gerais da rede de SP, adotadas como referência ao longo do trabalho. Estas características tentam ser gerais para descrever um grande número de SP, e definem uma variedade de serviços que a arquitetura dos SP deve estar capacitada a prover. As características escolhidas são as siguintes:

- Consta de um número significativo de equipamentos interconectados mediante diferentes tipos de conexões, como *Ethernet*, SDH/SONET, ATM (*Asynchronous Transfer Mode*), *Frame*

Relay, WDM, conexões por rádio, etc;

- Utiliza protocolo de roteamento de estado de enlace como ISIS (*Intermediate System to Intermediate System*) ou OSPF (*Open Shortest Path First*);
- Tem implementado MPLS (*Multiprotocol Label Switching*) em toda a rede de transporte do SP, com equipamentos LER (*Label Edge Router*) e equipes LSR (*Label Switch Router*). Os LER proporcionam os pontos de entrada e saída da rede de transporte, sendo estes os encarregados de adicionar o cabeçalho MPLS à entrada da rede e remover este cabeçalho na saída. Por outra parte, os LSR são os equipamentos internos à rede MPLS e baseam toda sua decisão de encaminhamento utilizando os rótulos MPLS;
- Utiliza a funcionalidade MPLS-TE (*Multiprotocol Label Switching - Traffic Engineering*), para realizar engenharia de tráfego;
- Presta serviço de VPN mediante MPLS-VPN (*Multiprotocol Label Switching - Virtual Private Network*);
- Dependendo do tipo de cliente (residencial ou empresarial), presta serviço de conectividade à Internet através de sua rede de acesso fixa mediante tecnologias como *Ethernet*, *Frame Relay*, ATM, xDSL (*Digital Subscriber Line*), TDM (*Time Division Multiplexing*), ou conexão por fibra;
- Possui clientes de porte considerável, interconectados mediante sessões BGP diretamente contra à rede de transporte;
- Possui diversas conexões a diferentes Tier1 (redes de trânsito que interconectam os ISPs), administradas mediante sessões BGP, proporcionando-lhe conectividade global;
- Dá suporte ao serviço *triple play* (vídeo, VoIP e HSI) aos clientes que o solicitarem.
- Dispõe de uma rede de telefonia móvel distribuída em um grande número de pontos geográficos (domínios móveis), interconectados mediante a rede de transporte do SP.
- Oferece serviço de DataCenter em várias localidades.

As características listadas proporcionam uma idéia mais aprofundada da complexidade das redes de SP atuais, assim como mostram a diversidade de serviços que os SP oferecem.

2.3 SDN

As redes definidas por software (SDN) são uma abordagem para o gerenciamento de redes de computadores que evoluiu a partir do trabalho realizado na Universidade de Berkeley e Stanford no ano de 2008 [1]. As SDN permitem que os administradores de rede gerenciem serviços de redes através da abstração das funcionalidades de nível inferior. Isto é conseguido com a separação do plano de controle (onde são construidas as decisões de roteamento) do plano de dados (o nível mais

baixo composto pelos dispositivos físicos responsáveis de routear o tráfego). Nas SDN, o plano de controle é trasladado para um dispositivo central denominado Controlador, que possui a visão global da rede e é o responsável pela configuração do plano de dados dos dispositivos de redes por meio de um protocolo padrão aberto. Adicionalmente, o Controlador permite a criação de uma camada de alto nível de abstração (camada SDN), possibilitando a programação dos serviços a serem configurados na rede, conforme ilustrado na Figura 2.2.

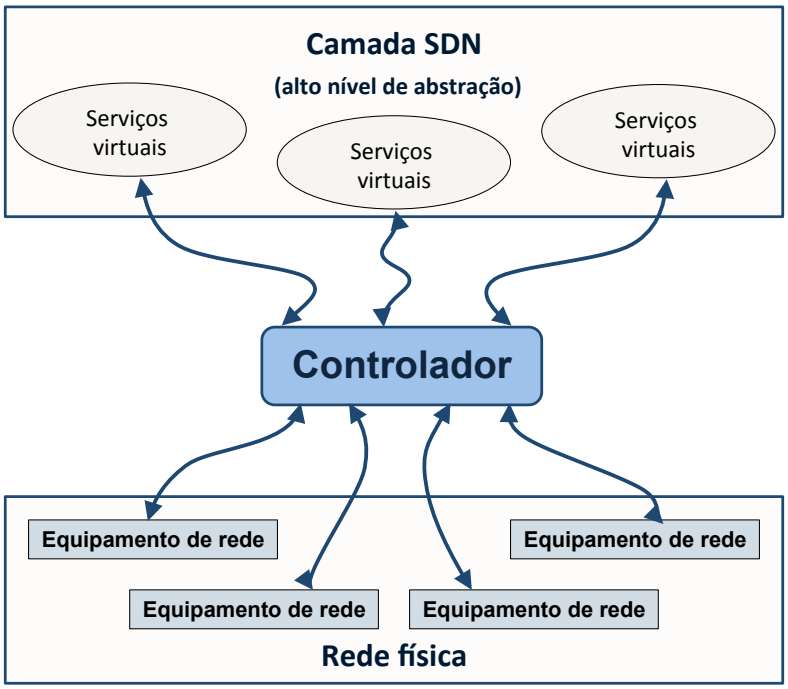

Figura 2.2: Arquitetura SDN

Para uma melhor compreensão do conceito de SDN, se mostra a arquitetura interna atual de um roteador e as mudanças que as SDN propõem.

A arquitetura atual de um roteador, como pode ser vista na Figura 2.3, pode ser dividida em duas partes:

1. o plano de dados: geralmente implementado em *hardware* onde se encontra a tabela de fluxos denominada FIB (*Forwarding Information Base*) dos dispositivos de rede. Esta tabela é constituida pela informação necessária para identificar univocamente os fluxos e as ações a realizar para cada um deles (por exemplo, sair por determinada porta);

2. o plano de controle: é implementado em *software*, lugar onde se encontram a lógica de roteamento (o protocolo de roteamento), e a tabela construida por ele, denominada RIB (*Routing Information Base*).

Cada primeiro pacote de um novo fluxo que chega a um roteador é inicialmente enviado ao plano de controle, e este decide o roteamento e a ação a ser tomada. Esta informação é colocada

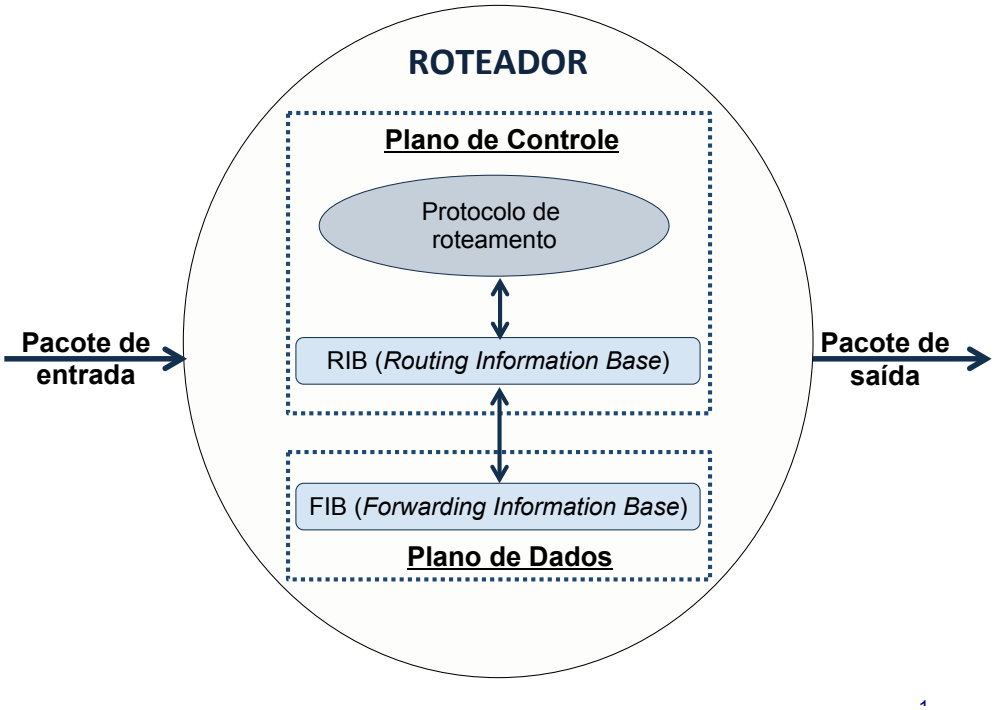

Figura 2.3: Arquitetura de um roteador

como uma nova entrada na tabela de fluxos FIB dentro do plano de dados. Com isto, no caso de chegar outro pacote desse mesmo fluxo, é o plano de dados quem encaminha o pacote diretamente, sem a necessidade de gerar uma nova consulta ao plano de controle, agilizando em grande medida o roteamento.

A idéia de uma SDN [20] é que os dispositivos de rede só tenham o plano de dados previamente descrito, diminuindo sensívelmente a complexidade destes dispositivos. Por outra parte, o plano de controle é colocado em um novo elemento de rede, denominado Controlador. O Controlador possui a visão e o controle global da rede, gerando assim, inúmeras novas possibilidades de controle e engenharia de tráfego.

Para uma melhor compreensão de uma SDN, se considera a situação descrita na Figura 2.4. Quando o primeiro pacote de um novo fluxo chegar a um dispositivo de rede, este ainda não tem pré-configurado um fluxo em seu plano de dados que lhe indique como encaminhar o pacote. Então, o dispositivo de rede envia uma consulta ao Controlador, que calcula o melhor caminho para o fluxo específico. Com esta informação e mediante um protocolo de comunicação apropriado, o Controlador configurará o fluxo específico no plano de dados de todos os equipamentos intervenientes no trajeto fim a fim, possibilitando assim, o roteamento do pacote. É importante notar que os sucessivos pacotes do mesmo fluxo, agora já dispõem de um fluxo específico pré-configurado no plano de dados, e portanto, não terão que realizar uma nova consulta ao Controlador.

Para que o funcionamento de uma SDN seja possível, é necessário que exista um protocolo de comunicação entre os diferentes elementos de rede e o Controlador, de forma que este último seja capaz de manipular as tabelas de fluxos dos elementos da rede. O protocolo mais amplamente

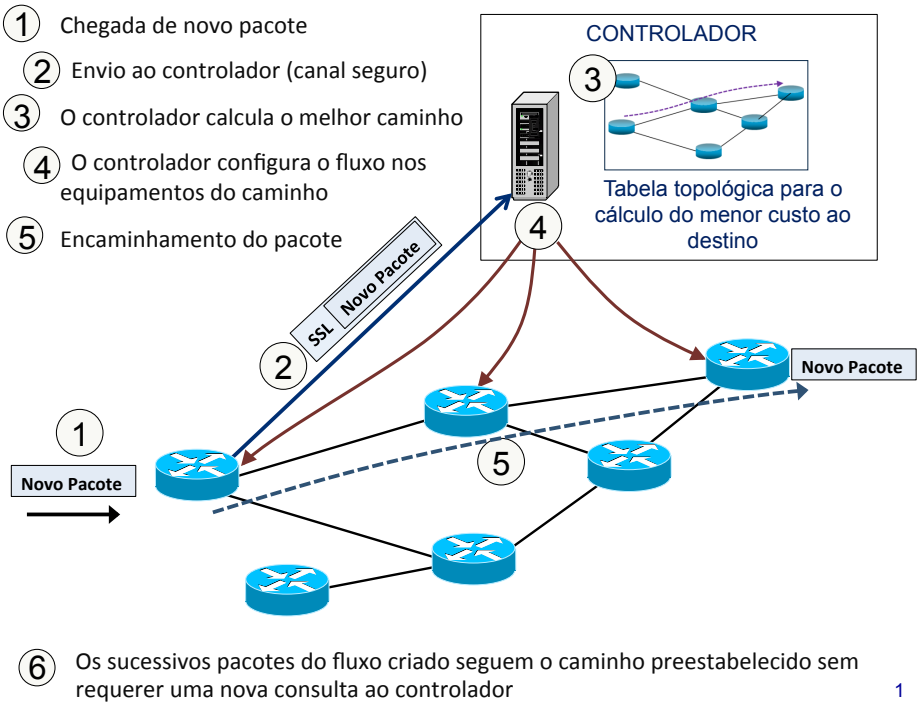

Figura 2.4: Modo de funcionamento de uma SDN

difundido e utilizado, e com estágio de desenvolvimento mais avançado com essa finalidade é o protocolo aberto *OpenFlow* [1] [2], escolhido no presente trabalho. *OpenFlow* proporciona uma interface aberta padrão para que a comunicação seja possível.

2.4 Protocolo *OpenFlow*

Um elemento de rede que implementa o *OpenFlow*, conforme ilustrado na Figura 2.5, tem que satisfazer três características fundamentais:

- ter uma tabela de fluxos, e para cada fluxo, uma ação a tomar;
- opcionalmente uma conexão TcP ou um canal seguro TLS (*Transport Layer Security*), que será utilizado para toda a comunicação entre os elementos de rede e o Controlador *OpenFlow*;
- utilizar o protocolo *OpenFlow* para toda a comunicação previamente mencionada, o que proporcionará uma interface padrão que permitirá a comunicação entre os elementos de rede (de qualquer fabricante que o implemente) e o Controlador.

Para cada fluxo entrante, os *switches OpenFlow* devem executar no mínimo três ações básicas:

- encapsular e reenviar o pacote ao Controlador através de um canal seguro. Isto será típicamente realizado para todos os pacotes que ainda não têm uma regra específica preenchida na tabela de fluxos;

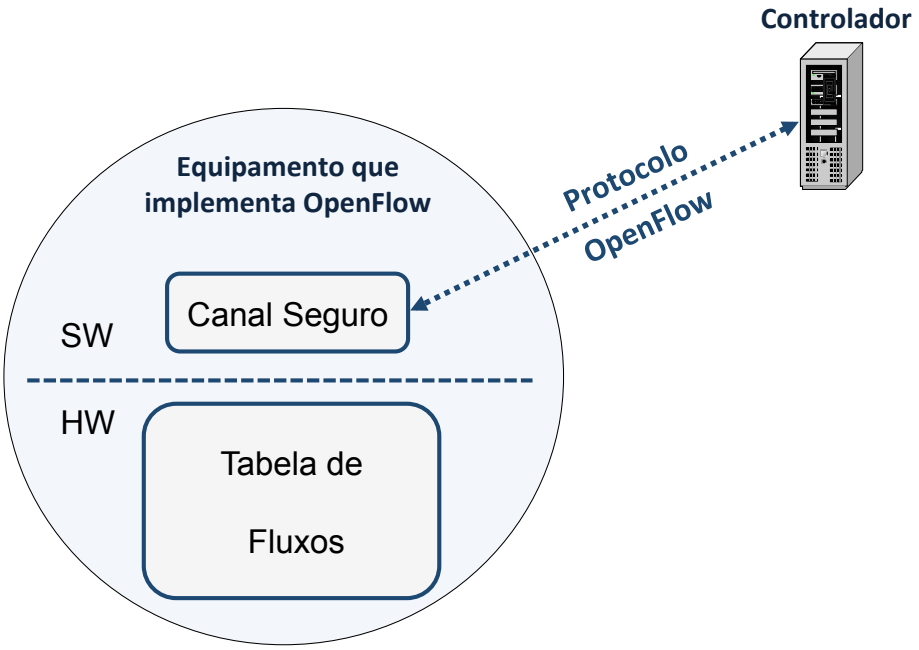

Figura 2.5: Arquitetura de equipamento que implementa *OpenFlow*

- encaminhar o pacote utilizando uma regra pré-estabelecida na tabela de fluxos (caso geral, uma vez que o Controlador configura o fluxo no equipamento de rede);
- descartar pacotes impedindo o roteamento e determinados fluxos na rede.

2.4.1 Definições do protocolo *OpenFlow*

A seguir são definidas as diferentes nomenclaturas utilizadas pelo protocolo *OpenFlow*, com o auxílio de alguns exemplos que ilustram melhor o conceito:

- Porta (*Port*): Ponto no qual um pacote ingressa ou sai da linha de processamento de *OpenFlow*. Pode ser uma porta física, uma porta lógica definida pelo *switch* (por exemplo, agrupamento de várias portas físicas como uma só porta lógica) ou portas reservadas, todas definidas no protocolo *OpenFlow*.
- Linha de processamento (*Pipeline Processing*): é composta por um grupo de tabelas de fluxos interligadas.
- Tabela de fluxos (*Flow Table*): Etapas da linha de processamento compostas por linhas de coincidência (entradas de fluxo ou regras).
- Entradas de fluxo (*Flow Entry*): São elementos na tabela de fluxos usados para processar pacotes buscando coincidências. Estas contêm um grupo de campos de coincidência para os

pacotes processados, uma prioridade de busca, um grupo de contadores para registrar todas as ocasiões que as coincidências ocorrerem, além de ter um grupo de instruções a aplicar.

- Campos de coincidência (*Match Fields*): campos que são examinados em procura de coincidência. Estes incluem os cabeçalhos dos pacotes, as portas de ingresso, valores de metadata, etc. Os campos de coincidência podem ser específicos na procura de um pacote particular ou podem procurar grupo de pacotes utilizando máscaras.
- Metadata: é o valor de um registro, usado para transportar informação de uma tabela para outra, dentro da linha de processamento.
- Instruções (*Instructions*): Integram as entradas de fluxo e são utilizadas para descrever o processamento que ocorre quando os pacotes coincidem com regras da tabela de fluxo. Uma instrução pode modificar a linha de processamento (direcionar o pacote para outra tabela de fluxo com número de sequência superior) ou conter um grupo de ações a serem executadas.
- *Action*: Estas podem ser adicionadas ao *action set* e executadas ao final da linha de processamento, ou podem ser aplicadas imediatamente ao pacote (*apply-action*).
- *Apply-Action*: Ação que é executada imediatamente, no momento que é estabelecida a coincidência com uma regra de qualquer tabela dentro da linha de processamento.
- *Action Set*: é um grupo de ações que são cumulativas durante o processamento do pacote através das tabelas e só são executadas quando o pacote chega ao final da linha de processamento.
- Grupo: é uma lista de *action buckets* e uma forma de escolher uma ou várias *action buckets* para cada pacote.
- *Action Bucket*: é um grupo de ações e parâmetros associados e definidos para grupos.
- Controlador: Uma entidade que interage com os *switches OpenFlow*, usando o protocolo *OpenFlow*. É o responsável da construção e gestão das tabelas de fluxo de cada equipamento da rede.
- *Meter*: Elemento do equipamento de rede que pode medir e controlar a taxa de transmissão para fluxos de pacotes.

2.4.2 Linha de processamento *OpenFlow*

Uma tabela de fluxos é formada por entradas de fluxo (regras), as quais são constituidas por: campos de coincidência, prioridade, contadores, instruções, tempos de espera e *cookie*, conforme ilustrado na Figura 2.6, em que as duas primeiras (campos de coincidência e prioridade) determinam univocamente uma entrada na tabela de fluxos. *OpenFlow* também estabelece uma linha de coincidência denominada *miss-table*, a qual pode ser colocada para processar os fluxos que não têm coincidência na tabela.

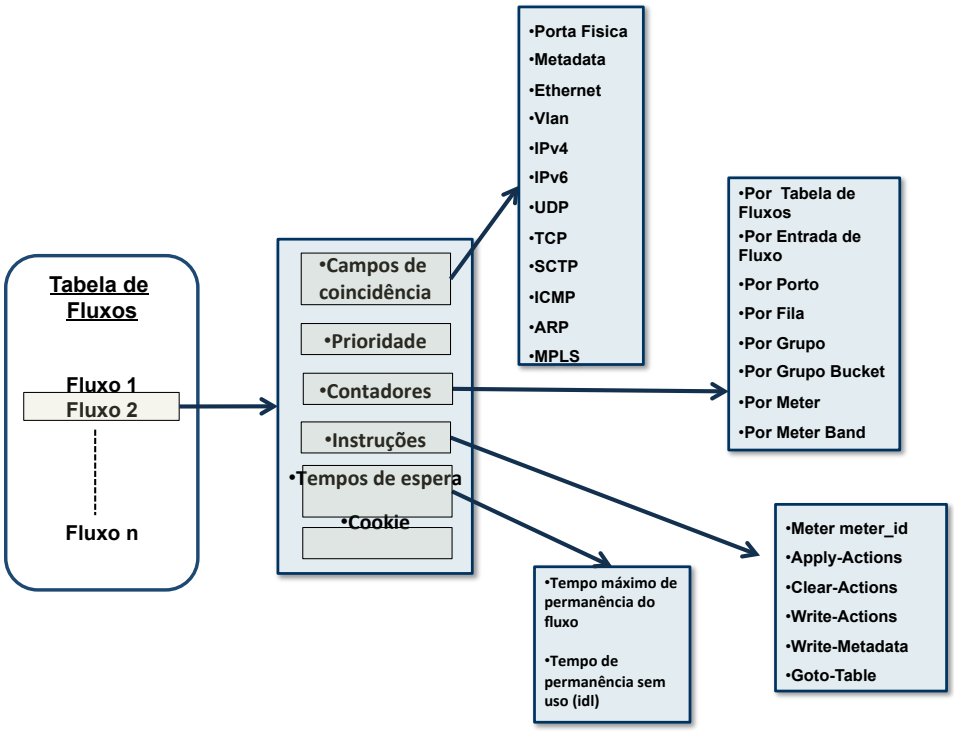

Figura 2.6: Elementos componentes das entradas de fluxo

A Figura 2.7 ilustra o processamento *OpenFlow* que ocorre na chegada de um novo pacote. Quando um novo pacote chega à rede, o equipamento *OpenFlow* realiza uma inspeção na Tabela 0 (primeira tabela do processamento), procurando coincidências dos campos do pacote com as regras da tabela de fluxos. No caso de encontrar coincidência, a regra terá que processar as instruções correspondentes, as quais terão ações associadas. Dentro das instruções, encontra-se o direcionamento para outra tabela de numeração superior, o que possibilita a realização de ligações sucessivas entre tabelas, formando a linha de processamento *OpenFlow*. O processamento de um pacote finaliza quando coincide com uma entrada de fluxo que não tem direcionamento para outra tabela, executando as *action-set*.

2.4.2.1 Tipos de instruções *OpenFlow*

Como mencionado anteriormente, cada entrada de fluxo tem instruções ligadas, conforme ilustrado na Figura 2.6, pudendo ser dos seguintes tipos:

- *Meter*: Encaminha o pacote para um medidor de tráfego, proporcionando um método para construir limitadores de tráfego.
- *Apply-Actions*: Ações de execução imediata, sem modificar o *action set*. Estas ações são utilizadas para modificar o pacote entre tabelas. Se a *apply-action* indica uma ação *Output*, uma cópia do pacote é encaminhada imediatamente e o pacote original continua o processamento normal. Esta última é uma característica muito utilizada na arquitetura proposta no trabalho.

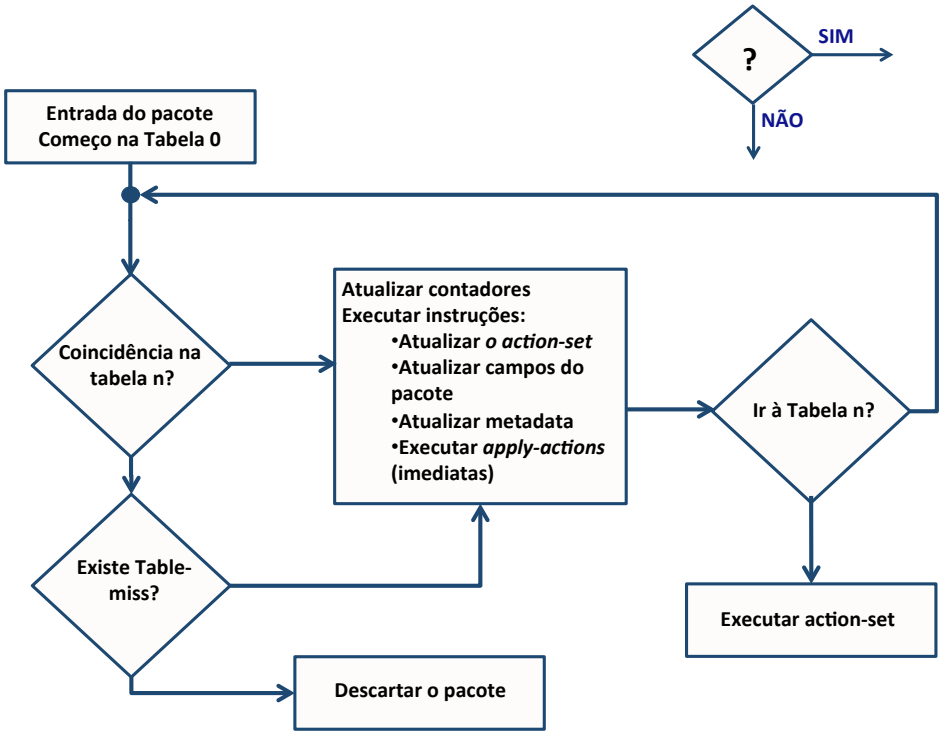

Figura 2.7: Processamento *OpenFlow*

- *Write-Actions*: Ações que vão se adicionando no *action set* e só serão executadas ao final da última tabela. Se duas regras em diferentes tabelas da linha de processamento do pacote estabelecem parâmetros diferentes para uma mesma ação, a última sobrescreve a primeira.
- *Clear-Actions*: Apaga as *action set* previamente preenchidas.
- *Write-Metadata*: Permite o envio de informação adicional entre tabelas.
- *Goto-Table*: Indica qual é a tabela que continuará o processamento do pacote.

2.4.2.2 Tipos de ações *OpenFlow*

A seguir são listadas e descritas as ações de maioir relevância para este trabalho:

- *Output*: Envía o pacote para uma porta de saída, que pode ser de um dos tipos a seguir:
  - Física: São portas definidas no equipamento que têm uma correspondência com as portas físicas do equipamento;
  - Lógica: São portas definidas no equipamento que não têm correspondência direta com portas físicas do *switch*. Estas portas são de mais alto nível de abstração e devem ser definidas utilizando métodos fora de *OpenFlow* (exemplo: *link aggregation group, tunnels, loopback interfaces*);
  - Reservada: São portas definidas por *OpenFlow* para facilitar roteamento de pacotes. A seguir são listadas as mais importantes para este trabalho:

    - *All*: Representa todas as portas do equipamento;
    - *Controller*: Representa o canal de controle com o Controlador. Quando é indicada a porta *controller*, o pacote ou o cabeçalho dele é encapsulado dentro de uma mensagem *packet-in* e é enviado ao Controlador;
    - *In-Port*: Representa a porta de entrada do pacote;
    - *Normal*: Esta saída só pode ser utilizada por equipamentos híbridos, isto é, equipamentos que tenham implementado o controle distribuído atual (não *OpenFlow*) e o controle *OpenFlow* simultaneamente. Esta saída indica que o pacote tem que sair da linha de processamento *OpenFlow* para continuar seu processamento no plano de controle distribuído tradicional. Esta funcionalidade é detalhada e utilizada na arquitetura híbrida proposta na seção 3.3;

- *Set-Queue*: Ligado a uma ação de saída, determina qual fila de saída processa o pacote, permitindo aplicar políticas de priorização de tráfego;
- *Drop*: Especifica que o pacote tem que ser descartado;
- *Group*: Indica que seja processado por uma ação grupal, permitindo ao pacote utilizar várias saídas simultaneamente, o balanceamento de tráfego, a implementação de caminhos alternativos realizando uma comutação rápida no caso de falha no caminho principal, etc. é importante ressaltar que diferentes entradas de fluxo (regras) na mesma tabela ou em diferentes podem ter a mesma ação de grupo;
- *Push-Tag/Pop-Tag*: Permite ações com os *tag*, seja para a identificação de *VLAN* ou para o rótulo *MPLS*;
- *Set-Field*: Permite modificar ou estabelecer uma grande variedade de campos do cabeçalho do pacote;
- *change-TTL*: Permite estabelecer ou copiar o *TTL* do *IP TTL* e do *MPLS TTL*.

2.4.3 As mensagens *OpenFlow*

Para a comunicação entre os diferentes elementos de rede e o Controlador, é utilizado o canal *OpenFlow*, quem funciona como interface possibilitando a configuração e o gerenciamento dos diferentes equipamentos de rede. A implementação da comunicação entre o plano de dados e o canal *OpenFlow* é específica de cada equipamento, mas o canal *OpenFlow* tem que ser padrão. O protocolo *OpenFlow* dá suporte a três tipos de mensagens: *controller-to-switch*, *asynchronous* e *symmetric*, cada um deles com uma grande variedade de subtipos, conforme ilustrado na Figura 2.8.

2.4.3.1 Mensagens *Controller-to-Switch*

Estas mensagens são iniciadas pelo Controlador, e devem ou não ser respondidas pelos equipamentos clientes. Entre estas mensagens, destacam-se os subtipos descritos a seguir:

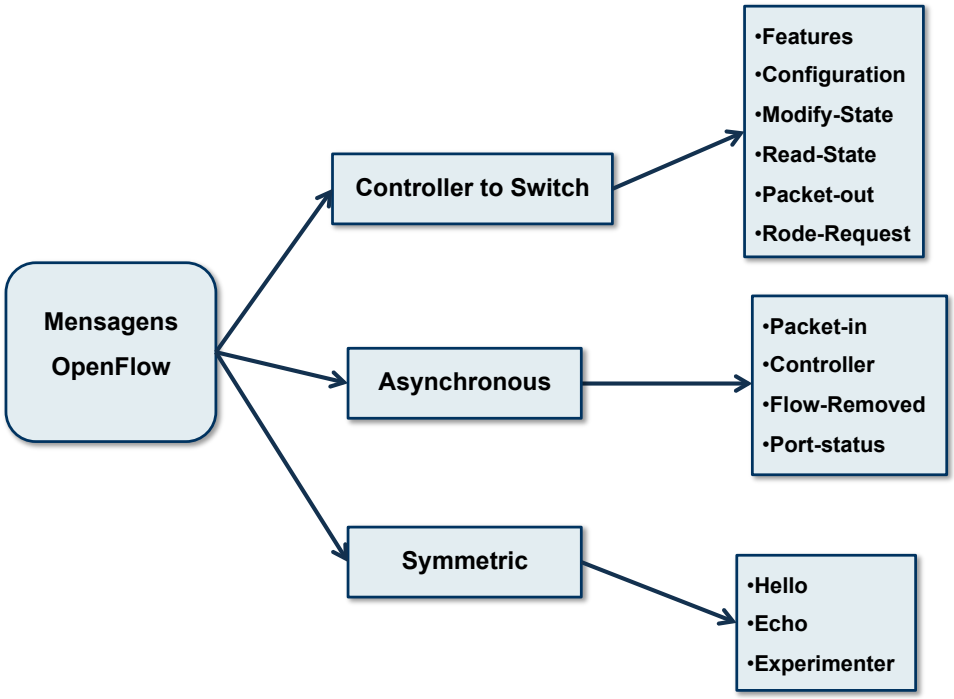

Figura 2.8: Tipos de mensagens *OpenFlow*

- *Features*: O Controlador pode solicitar as capacidades do equipamento, enviando uma solicitação *Feature*. O *switch* deve responder com um *Features-replay*, especificando assim suas capacidades. Isto é usualmente realizado após o estabelecimento do canal *OpenFlow*;
- *configuration*: O Controlador pode estabelecer e solicitar parâmetros de configuração aos equipamentos;
- *Modify-state*: Estas mensagens são enviadas pelo controlador para gerenciar o estado dos *switches*. O principal propósito é adicionar, apagar e modificar entradas de fluxo nas tabelas, e também para estabelecer propriedades nas portas dos equipamentos;
- *Read-state*: é utilizado pelo Controlador para coletar diversos tipos de informação dos clientes como configuração atual, estatísticas e capacidades do equipamento;
- *Packet-out*: Utilizado pelo Controlador para especificar qual é a porta de saída que o equipamento cliente tem que utilizar para encaminhar o pacote, o qual foi inicialmente recebido por uma mensagem *Packet-in*. O *Packet-out* pode conter o pacote completo ou uma referência a um pacote guardado no cliente. Adicionalmente, esta mensagem tem que conter uma lista de ações a serem aplicadas, mas no caso de não haver ações, a mensagem indica que o pacote tem que ser apagado. Esta última característica é muito utilizada neste trabalho;
- *Role-request*: Mensagem utilizada pelo Controlador para estabelecer seu papel no canal *OpenFlow* ou para perguntar o papel. Isto é utilizado quando um cliente é conectado a vários controladores.

2.4.3.2 Mensagens *Asynchronous*

Estas mensagens são enviadas pelos equipamentos cliente para o Controlador, sem nenhuma solicitação prévia. São utilizadas para comunicar a chegada de um pacote, a mudança do estado de uma porta ou algum erro. Os principais tipos destas mensagens são listados a seguir:

- *Packet-in*: Estas mensagens permitem transferir o controle do roteamento de um pacote de entrada ao Controlador. No caso de um pacote entrante ter coincidência com uma regra ligada a uma ação *output* à porta reservada *controller*, esta encaminha o pacote ao Controlador mediante uma mensagem *Packet-in*. Esta mensagem pode conter o pacote total ou só parte de seu cabeçalho, mas neste último caso, o equipamento de rede tem que guardar o pacote completo;
- *Flow-removed*: O equipamento de rede informa ao Controlador que um fluxo foi removido de uma das tabelas de fluxo. Estas mensagens são enviadas por uma entrada de fluxo, caso tenha sido estabelecida a *flag OFPFF_SEND_FLOW_REM*, e são geradas como resultado de uma solicitação do Controlador de remoção de fluxo ou devido à expiração do *timeout* ligado à entrada de fluxo;
- *Port-status*: O cliente informa ao Controlador a mudança no estado de uma de suas portas. Estas mensagens notificam as mudanças de estado das portas, seja por uma ação de configuração no equipamento ou por um evento de mudança de estado;
- *Error*: com estas mensagens, os clientes podem notificar ao Controlador vários tipos de erros.

2.4.3.3 Mensagens *Symmetric*

Estas mensagens são enviadas pelos equipamentos de rede ou pelo Controlador, sem nenhuma solicitação prévia.

- *Hello*: São mensagens trocadas entre o Controlador e os equipamentos de rede, na inicialização da conexão.
- *Echo*: São mensagens enviadas pelo Controlador e pelos clientes. Inicialmente é enviada uma solicitação *Echo-request*, que é respondida mediante um *Echo-reply*. Estas mensagens são principalmente utilizadas para verificar o estado do canal *OpenFlow*.
- *Experimenter*: Permite implementar funcionalidades adicionais, dentro do espaço das mensagens *OpenFlow*.

2.4.4 Troca de mensagens OpenFlow

Nas subseções a seguir, é descrita uma comunicação básica entre um equipamento cliente e o Controlador.

2.4.4.1 Estabelecimento da conexão

As conexões no canal *OpenFlow*, ilustradas na Figura 2.9, são iniciadas pelo cliente, que tem previamente configurados o endereço IP e a porta do controlador. Inicialmente, é estabelecida uma conexão TLS (*Transport Layer Security*) ou uma conexão TcP (para o caso de canal sem segurança), e em seguida o controlador e o cliente *OpenFlow* passam a trocar as mensagens *Hello* que permitem acordar os parâmetros básicos (como a versão de *OpenFlow* a ser utilizada). Quando a conexão se encontra estabelecida, ambos os dispositivos trocarão periodicamente mensagens *Echo* (*request/reply*), verificando assim, o estado do canal de controle *OpenFlow*.

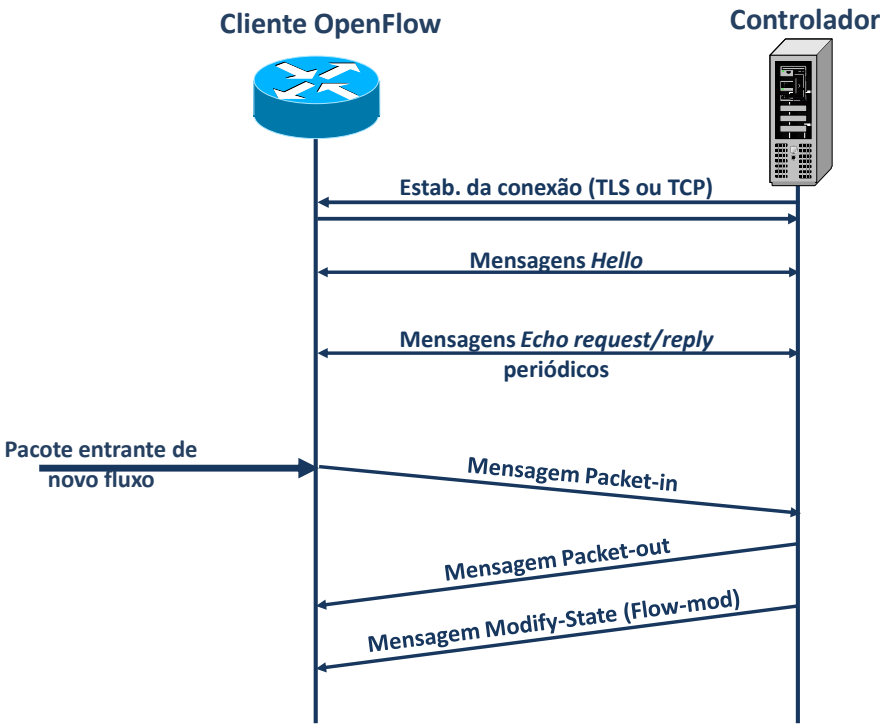

Figura 2.9: Sequência das mensagens *OpenFlow*

2.4.4.2 Chegada de um pacote

Quando um pacote pertencente a um novo fluxo chega ao cliente *OpenFlow*, este é encaminhado ao Controlador mediante uma mensagem *Packet-in* (Figura 2.9). O Controlador calcula o melhor caminho para o pacote entrante, e normalmente, executa as duas ações seguintes:

1. responde o *Packet-in* atraves de uma mensagem *Packet-out*, indicando como o equipamento deve encaminhar o pacote;

2. gera uma mensagem do tipo *Modify-state* (*Flow-Mod*) para que seja criado um fluxo específico na tabela do equipamento cliente e de todos os outros clientes intervenientes no caminho. Desta forma, os pacotes sucessivos do mesmo fluxo não têm que realizar a mesma consulta ao Controlador.

2.5 Multiprotocol Label Switching (MPLS)

MPLS é um mecanismo utilizado em redes de alta performance que baseia o encaminhamento do tráfego em função de um rótulo, em lugar de amplas direções IP e seus respectivos prefixos. Os rótulos identificam caminhos virtuais entre nós da rede e não entre os equipamentos finais. Com a adição do cabeçalho MPLS, os pacotes podem atravessar uma rede MPLS sem a necessidade de analisar o cabeçalho IP ou o cabeçãlo do que for transportado.

MPLS é um protocolo (da família das redes de comutação de pacotes) altamente escalável, independente do protocolo utilizado para seu transporte. O cabeçalho adicionado por MPLS se encontra localizado no que corresponderia a uma camada intermédia entre a 2 e a 3 no modelo OSI (*Open Systems Interconnection*), o que permite criar caminhos fim a fim entre diferentes meios de transporte. MPLS permite eliminar a dependência em relação a tecnologias de camada 2 utilizadas como ATM, *Frame Relay*, *Ethernet*, etc.

O cabeçalho MPLS é mostrado na Figura 2.10, composto dos campos descritos a seguir:

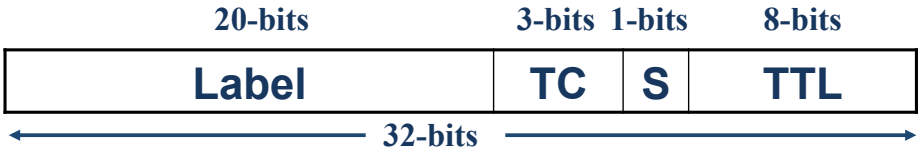

Figura 2.10: O cabeçalho MPLS

- *Label*: é um rótulo de 20 bits que atúa como um identificador local, o que permite diferenciar univocamente um caminho MPLS denominado LSP (*Label Switched Path*) com um rótulo específico.
- Tc (*Traffic class*): Identifica o tipo de tráfego que é transportado. No caso de ser utilizado para o transporte de pacotes IP, o Tc pode conter a informação do campo *IP precedence* do cabeçalho IP.
- S: Bandeira indicadora de *stack*. Está constituído por um só *bit* que quando é estabelecido em 1, indica que não é o último cabeçalho MPLS do *stack*. Com esta técnica é possível adicionar mais de um cabeçalho MPLS para cada pacote, ligando a um pacote mais de um rótulo MPLS.
- TTL (*Time to Live*): Este campo é preenchido com um valor inicial pelo nó entrante à rede MPLS (denominado LER *Label Edge Router*), e diminuido durante o encaminhamento por cada um dos nós internos à rede MPLS (denominados LSR *Label Switch Router*) até chegar ao LER de saída.

Este protocolo é amplamente utilizado nas redes de SP para o transporte de pacotes, para realização de engenharia de tráfego, e para a criação e o gerenciamento de VPN (*Virtual Private Network*). Particularmente é muito utilizado no domínio de transporte dos SP implementando os diversos serviços de transporte, mas também é utilizado no domínio de acesso fixo (em redes como as de agregação *Ethernet*), e pode ser utilizado dentro do domínio móvel para proporcionar interconectividade entre os elementos lógicos de este domínio. Sua ampla presença nestas redes, a abstração entre o tráfego transportado e rótulo ligado, e as características previamente mencionadas o indicam como um excelente protocolo para o transporte do tráfego fim a fim. MPLS é o protocolo adoptado pelas arquiteturas propostas para a implementação do plano de dados.

2.5.1 Descrição geral do funcionamento MPLS

A redes MPLS estão constituidas por dois tipos de equipamentos, os LER e os LSR. Os LER são os primeiros equipamentos que recebem um pacote proveniente de fora do domínio MPLS, examinam o cabeçalho (exemplo cabeçalho IP), e adicionam o cabeçalho MPLS para seu encaminhamento dentro da rede MPLS. Adicionalmente os LER são os últimos equipamentos dentro do caminho do pacote na rede MPLS, sendo estes os encarregados de retirar o cabeçalho MPLS prévio a seu envio fora do domínio MPLS. Os LSR (internos ao domínio MPLS), são os responsáveis do encaminhamento dos pacotes dentro da rede MPLS, utilizando para isto somente a inspeção do rótulo MPLS.

A Figura 2.11 mostra como um caminho MPLS (LSP) pode ser definido. Para isto é adicionada uma linha nas tabelas de encaminhamento de cada um dos nós intervenientes no trajeto. Como pode se observar no exemplo, os rótulos adicionados são utilizados para o encaminhamento do pacote com endereço IP 10.1.1.1, mas esse mesmo rótulo é utilizado para todos os pacotes dentro do prefixo 10.1.1.0/24. A ligação inicial entre prefixo e rótulo é realizada pelo LER de entrada (LER1), e posteriormente os LSR só examinam o rótulo MPLS para seu encaminhamento. Finalmente é o LER de saída (LER2) quem retira o rótulo MPLS e encaminha o pacote fora do domínio MPLS.

É interessante notar que os critérios de ligação entre rótulo e pacotes associados podem ser variados, devido a que também poderiam ser adicionados critérios com direções origem, protocolo, portas ou campos ToS (*Type of Service*) entre outros critérios simultaneamente. Cada FEc (*Forward Equivalence class*) tem ligado um rótulo, que define o grupo de pacotes a ser encaminhado, utilizando o mesmo LSP. Adicionalmente se menciona que os LSP são definidos de forma unidirecional, tendo que ser estabelecidos dois LSP (um para cada direção) para proporcionar uma comunicação bidirecional através da rede MPLS.

O protocolo MPLS não funciona isoladamente, requere de um protocolo de encaminhamento interno para seu funcionamento. Inicialmente o protocolo de encaminhamento como o OSPF ou ISIS construi as regras de encaminhamento interno para atingir todas as redes destino. Com esta informação disponibilizada nas tabelas de encaminhamento, o protocolo MPLS estabelece ligações entre entradas da tabela e rótulos MPLS. Utilizando o protocolo LDP (*Label Distribution Protocol*) estas ligações entre rótulos são informadas a todos os roteadores vizinhos. Como exemplo, na Figura 2.11 o roteador LSR1 informa ao roteador LER1 que a FEc associada ao prefixo 10.1.1.0/24 está

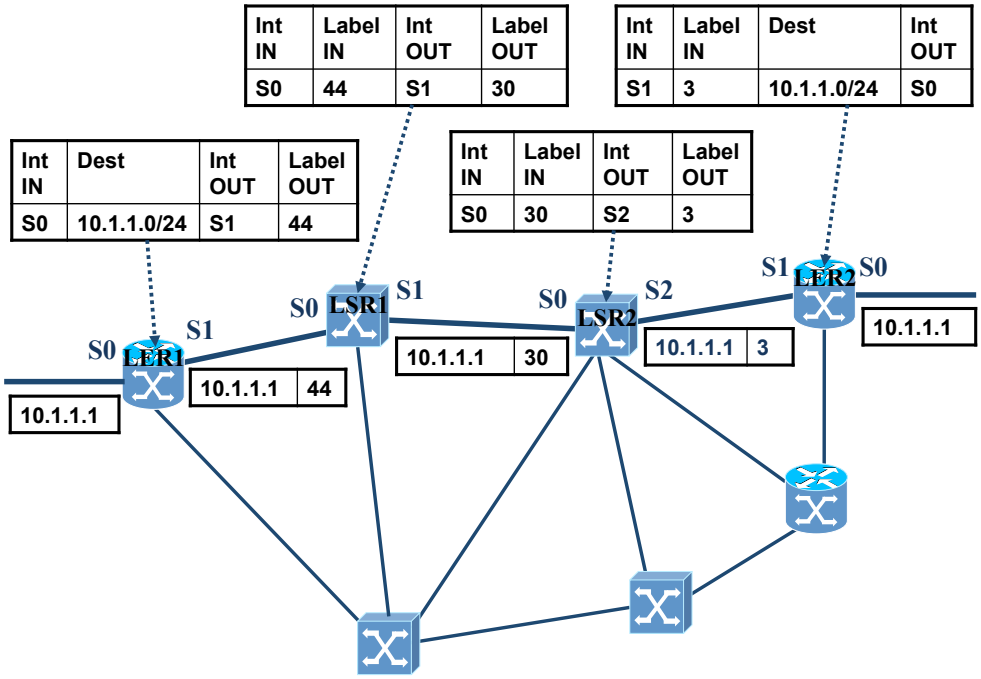

Figura 2.11: construção de um LSP em uma rede MPLS

ligada localmente ao rótulo 44. Com esta informação o roteador LER1 sabe que para enviar um pacote ao roteador LSR1 utilizando esse LSP (definido pela FEc associada ao prefixo 10.1.1.0/24) tem que ser adicionado o rótulo MPLS 44. Analogamente o roteador LSR2 informa ao LSR1 que tem que colocar o rótulo 30 para encaminhar os pacotes por esse LSP. com isto o roteador LSR1 sabe que os pacotes entrantes pela interface S0 com rótulo 44, têm que ser enviados pela interface S1 com rótulo 30.

2.5.2 Engenharia de Tráfego

Existem duas alternativas nas redes de MPLS para estabelecer os LSP. Uma delas é utilizando mecanismos *hop by hop* independentes, através do protocolo LDP previamente mencionado. Outra alternativa é a construção do que é denominado ER-LSP (*Explicitly-Routed LSP*), caminho explicitamente estabelecido indicando cada nó do trajeto fim a fim.

Para a construção de estes caminhos (ER-LSP) os protocolos mais utilizados são o RSVP-TE (*Resource Reservation Protocol - Traffic Engineering*) e o cR-LDP (*constraint-based Routing Label Distribution Protocol*). Estes protocolos têm a possibilidade de manipular caminhos utilizando parâmetros predeterminados de qualidade de serviço ou classe de serviço, que permitem construir o LSP ótimo para um tipo de tráfego. Estes caminhos podem ser estabelecidos estritamente indicando todos os nós que devem ser atravessados, ou de forma flexível indicando só alguns nós intervenientes do caminho fim a fim.

Dentro das técnicas adicionais de engenharia de tráfego se encontram os mecanismos de res-

tauração imediata. Estas técnicas permitem estabelecer um caminho titular e seu secundário para fluxos específicos. Em condições normais, o tráfego utiliza sempre o caminho titular para seu encaminhamento, mas no caso de queda de um nó ou enlace interveniente no caminho titular (ver Figura 2.12), o tráfego é imediatamente enviado pelo caminho secundário (restaurando o serviço instantáneamente). É importante ressaltar que a configuração prévia do caminho secundário diminui sensívelmente os tempos de restabelecimento, reduzindo ao mínimo as possibilidades de perdas de pacotes no processo de transição.

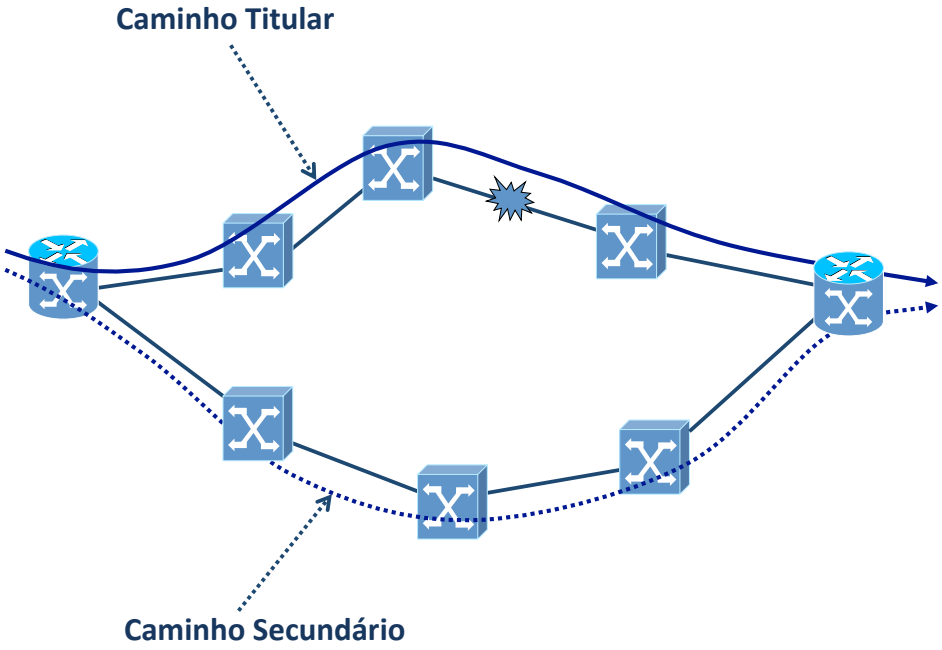

Figura 2.12: Mecanismo de restauração ante queda de nó ou enlaces do caminho titular

Finalmente se menciona que existem diferentes técnicas de proteção de caminhos. Estas podem proteger todo o caminho (todo o LSP), podem ser para a proteção ante a queda de um nó específico, ou para a proteção de quedas de enlaces particulares. Informação adicional pode ser encontrada em textos como: MPLS Fundamentals [21] e Traffic Engineering with MPLS [22].

2.5.3 MPLS VPN

É uma família de métodos que utilizam a fortaleza de MPLS para criar serviços VPN. Proporciona mecanismos de engenharia de rede flexíveis para o transporte de diversos tipos de tráfego utilizando a rede MPLS. Existem três tipos de VPN MPLS: VPN *point-to-point*, VPN de camada dois (VPLS - *Virtual Private LAN Services*) e VPN de camada 3 (VPRN - *Virtual Private Routed Network*). Na Figura 2.13 se ilustram duas destas alternativas.

As MPLS VPN *point-to-point*, fornecem conectividade camada 2 fim a fim entre dois locais. Desde a perspectiva do cliente, é oferecido um serviço de conectividade fim a fim entre dois locais,

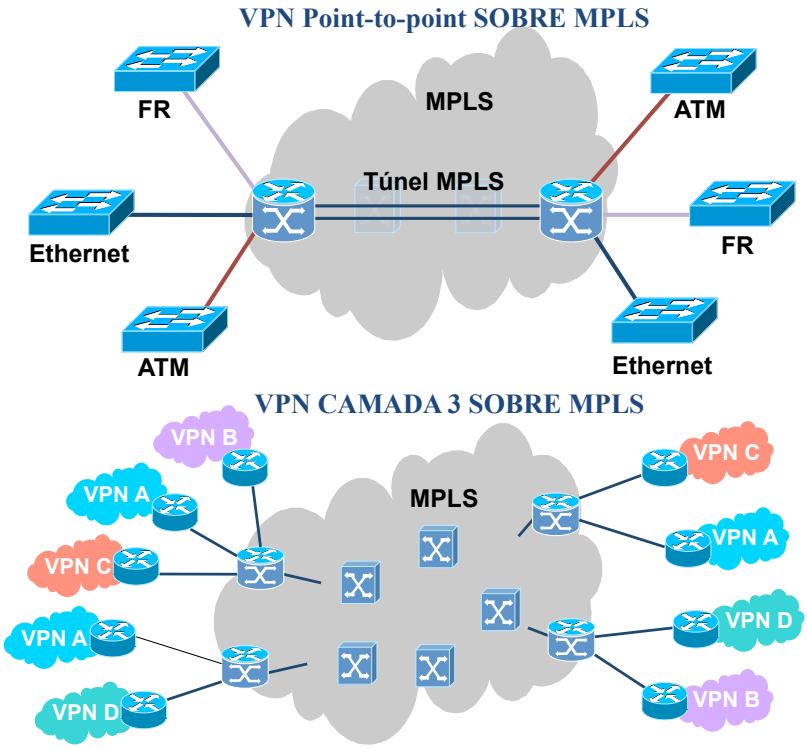

Figura 2.13: Representação de VPN MPLS *point-to-point* e VPN MPLS de camada 3 (VPRN)

que podem ser de diferentes tipos como *Ethernet*, ATM, TDM ou *Frame Relay*. Este método mantém o transporte intermédio utilizando a rede MPLS totalmente transparente para o cliente. Para isto é realizado um caminho MPLS virtual dentro do domínio MPLS que transporta o protocolo de camada 2 entre os dois pontos da rede MPLS próximos aos destinos a serem interconectados.

Por outra parte, as MPLS VPN do tipo VPLS, proporcionam interconectividade entre vários pontos simultáneamente. Desde a perspectiva do cliente a conectividade proporcionada por esta alternativa é equivalente a proporcionar ao cliente um grande "*switch*"com uma porta em cada um dos locais nos que o cliente requer de interconectividade. Para poder cumprir com este objetivo a rede MPLS implementa a habilidade de transportar a informação de VLAN dentro do domínio MPLS (entre os locais de cada cliente).

As VPRN utilizam VRF (*Virtual Routing and Forwarding*) de camada 3 para segmentar as tabelas de encaminhamento para cada cliente. Com isto, é possível encaminhar o tráfego entre locais do cliente com independência dos endereços IP utilizados por outros clientes. A seguir é mostrado um exemplo geral do funcionamento, em particular indicando o caso para VPN camada 3, mas o conceito de adição de duplo *stacks* MPLS é um conceito utilizado em todos os tipos de VPN MPLS.

Na Figura 2.14 é ilustrada uma VPN camada 3, a qual interconecta dois locais do cliente A (VPN A) e dois locais do cliente B (VPN B). Os *routers* denominados Rx (com x entre 1 e 5) são de clientes e estão fora do domínio MPLS. Nessa figura estão representados dois pacotes que ingressam à rede MPLS pelo LER 1, um proveniente da VPN A e outro proveniente da VPN B. O LER 1 mantém tabelas de encaminhamento isoladas por VPN, construidas com informação proveniente

de todos os LER que intervêm nas VPN. Com esta informação o LER1 conhece o rótulo MPLS que tem que colocar para encaminhar o pacote dentro da rede MPLS até o LER destino (rótulo externo de cor verde), e o rótulo adicional que tem que ser colocado para diferenciar a VPN destino quando o pacote MPLS chega ao LER 2 (rótulo azul e rosa para a VPN A e VPN B respectivamente, ver Figura 2.14).

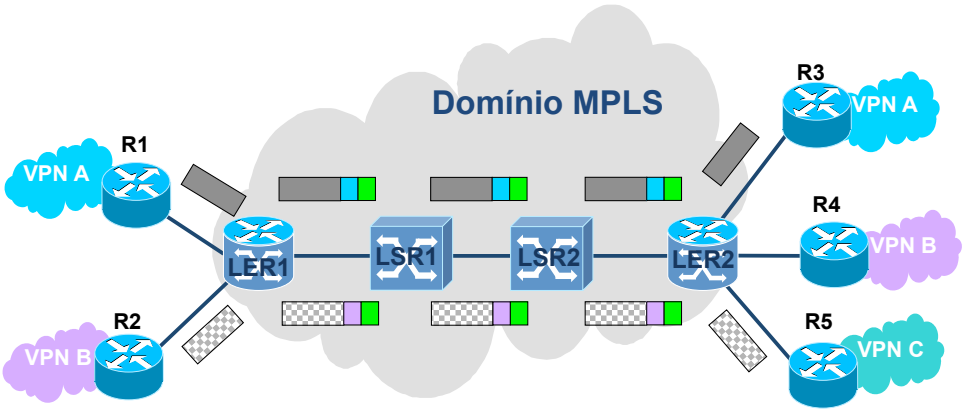

Figura 2.14: Exemplo de funcionamento de MPLS VPN camada 3 (VPRN)

Com esta metodologia e a adição de dois rótulos MPLS, é possível a construção de VPN facilmente gerenciáveis e muito flexíveis, que proporcionam o devido isolamento entre VPN (ainda quando o espaço de endereços utilizados por cada VPN é o mesmo). Isto é possível devido a que a informação de encaminhamento intercambiada entre os LER sempre contém um identificador de VPN que faz única a combinação de este indicador e o prefixo IP. Por mais informação sobre MPLS VPN se recomenda a leitura dos textos MPLS Fundamentals [21] e MPLS e VPN Architectures [23].

2.6 Rede móvel 4G

A rede móvel pode ser considerada composta por dois tipos de entidades, físicas e lógicas. Dentro das entidades físicas temos os *routers*, *switches* e enlaces físicos (interligados por várias topologias possíveis) encarregados de proporcionar interconectividade e transporte às entidades lógicas da rede móvel. Por outro lado, as entidades lógicas são as que implementam as funcionalidades específicas das redes móveis como o processo de *attachment* dos UE, gerenciamento da mobilidade, transporte de dados, mensagens de controle, etc. Algumas das entidades lógicas são descritas a seguir:

- eNodeB/eNB: proporciona a interface de rádio aos UE, e desempenha a função de gerenciamento dos recursos radiais para a rede LTE (*Long-Term Evolution*). Isto inclui o controle dos rádio *bearers*, controle de admissão, e o planejamento dos recursos de rádio (*Uplink* e *Downlink*) para cada UE individualmente. Também tem a importante função de dar suporte à compressão do cabeçalho IP e à encriptação do plano de dados. O eNodeB se conecta ao

EPc (*Evolved Packet core*) via a interface S1, onde a interface de controle ÃĽ denominada S1-MME (conecta ao eNodeB com o MME) e a interface para o plano de dados (tráfego) é denominada S1-U (conecta lógicamente a eNodeB com o S-GW).

- MME (*Mobility Management Entity*): Desde a perspectiva do núcleo da rede móvel, o MME é a entidade mais importante no que refere ao plano de controle para a rede de acesso. Quando uma UE se conecta à rede móvel por primeira vez, ou no processo de *handover*, o MME é o encarregado de escolher um S-GW para o UE. Também é o responsável pelos procedimentos de *tracking* e *paging* para as UE, e pela ativação e desativação dos *bearers* do lado da UE. Esta entidade se interconecta lógicamente com o HSS para proporcionar autenticação aos usuários finais.

- S-GW (*Serving Gateway*): é o responsável por terminar todas as conexões contra o E-UTRAN. cada UE conectado ao EPS (*Evolved Packet System*) está associado com um único S-GW. Este é um dos elementos encarregados de encaminhar o tráfego das UE e está localizado lógicamente entre o eNodeB e o P-GW. Entretanto, podem ser unificados o S-GW com o P-GW formando um equipamento denominado P/S-GW nas redes móveis.

- P-GW (*PDN Gateway*): proporciona a conectividade aos UE com uma PDN (*Packet Data Network*), proporcionando entre outros parâmetros, o endereço IPv4 ou o prefixo IPv6. é a porta de entrada e saída do tráfego dentro do domínio da rede móvel. Em sua função como *gateway*, o P-GW realiza a inspeção e o filtrado de pacotes para cada usuário. Também realiza o controle de admissão, o de largura de banda utilizada pelos UE, e ações relativas a QoS como o marcado de pacotes adicionando prioridade no DScP (*Diferentiated Services code Point*). O P-GW é o dispositivo central para o controle de QoS, quem ativa, desativa e modifica os EPS-*bearers* e realiza o mapeamento de tráfego para EPS-*bearers* específicos (por mais informação sobre EPS-*bearer* ver a seção 2.6.1).

- PCRF (*Policy and Charging Rules Function*): é o elemento lógico da rede móvel designado para determinar as regras a serem aplicadas aos serviços multimídia em tempo real. Este equipamento é informado de uma sinalização em processo, e solicita informação adicional do perfil do usuário e de método de faturamento. Com esta informação decide a política a ser aplicada e solicita sua implementação ao P-GW. Com este procedimento podem ser oferecidos serviços multimídia com requisitos específicos.

Outra possível subdivisão da rede móvel consiste em: rede de acesso móvel, formada pelos terminais móveis (UE - *User Equipment*) e os eNodeB, que proporcionam o acesso por rádio aos UE; o *backhaul*, que está composto por todos os *switches* que permitem agrupar e encaminhar o tráfego proveniente da rede de acesso móvel; e o núcleo da rede móvel, que agrupa os elementos centrais para o gerenciamento da mobilidade, para o faturamento, para o controle de admissão para a interconectividade e para o gerenciamento de QoS (MME, S/P-GW, HSS, PcRF, etc). Na Figura 2.15 é mostrada uma implementação topológica de arquitetura 4G que permite visualizar a subdivisão previamente mencionada.

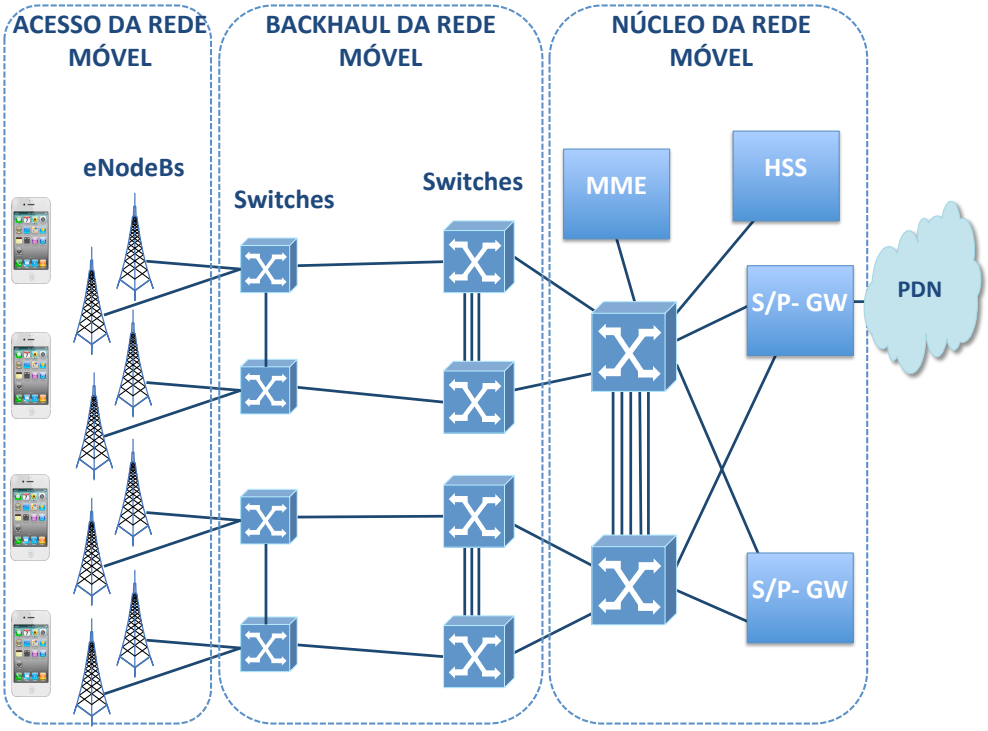

Figura 2.15: Exemplo de arquitetura de rede 4G subdividida em acesso, *backhaul* e núcleo móvel

Toda a arquitetura 4G, a diferença das arquiteturas móveis anteriores, está pensada para proporcionar todos seus serviços baseados totalmente na rede IP, reduzindo custos e simplificando funcionalidades, mas também dificultando o gerenciamento do QoS por se tratar de uma rede baseada em comutação de pacotes (*best effort*).

É importante notar que para qualquer rede móvel, o gerenciamento da mobilidade é um dos principais desafios, particularmente devido ao fato de que o *stack* IP não foi criado pensando na mobilidade. Isto se deve a que o endereço IP cumpre com a função de identificador, mas também com a função de localizador. Para resolver este problema, as redes móveis utilizam em seu plano de dados, túneis IP sobre IP, que na modalidade mais utilizada são criados com o protocolo GTP (*GPRS Tunneling Protocol*) entre o eNodeB e o P-GW (ver Figura 2.16). Particularmente é utilizado para isto o GTP-U. Desta forma, os pacotes IP com destino o UE, quando entram na rede móvel, são recebidos pelo P-GW, colocando o pacote dentro de outro pacote, utilizando o protocolo GTP-U. O cabeçalho GTP-U adicionado contém uma nova IP destino, que permite atravessar a rede móvel e atingir o eNodeB (conectado ao UE). Finalmente o eNodeB retira o cabeçalho GTP-U adicionado e encaminha o pacote original ao UE destino. É importante notar que a modalidade de tunelizado descrita é uma alternativa de mobilidade baseada na rede (os UE não requerem nenhum gerenciamento da mobilidade).

Os túneis GTP-U identificam univocamente os fluxos de tráfego que recebem um mesmo tratamento de QoS entre o P-GW e o eNodeB. O filtros denominados TFT (*Traffic Flow Template*) são colocados nos UE para mapear tráfego para *bearers* específicos, onde os *bearers* são caminhos estabelecidos para prover de QoS (por mais informação dos conceitos de *bearer*, ver seção 2.6.1).

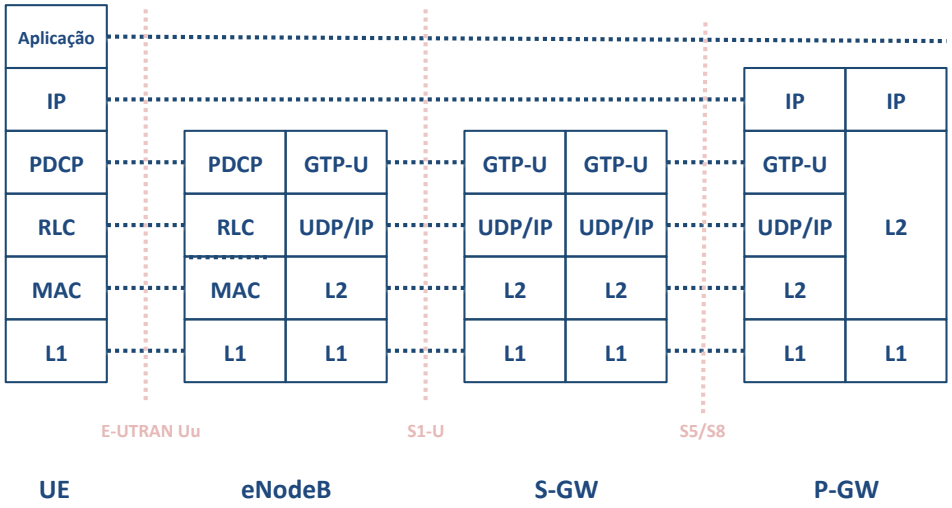

Figura 2.16: *Stack* para o plano de dados 4G, utilizando GTP-U

Os TFT estão compostos por regras de coincidência que permitem classificar o tráfego para um *bearer* particular (ver Figura 2.18). O cabeçalho GTP-U possui um campo denominado TEID (*Tunnel Endpoint Identifier*) que permite identificar univocamente todos os tuneis GTP-U (existe uma relação unívoca entre *bearer* e TEID no trajeto eNodeB e P-GW).

Finalmente se menciona que toda a comunicação realizada no plano de controle entre MMEs, entre MME e S-GW, e entre S-GW e P-GW é realizada utilizando o protocolo GTPv2-c que é encaminhado diretamente sobre UDP (ver Figura 2.17). Esta variante de GTP, a diferença do GTP-U, não requer da utilização de túneis IP, devido a que o destino e origem dos pacotes enviados pelo protocolo GTP-c são entidades lógicas da rede móvel (sempre alcançáveis dentro da rede móvel).

2.6.1 QoS e políticas de controle nas redes móveis 4G

A conectividade das terminais móveis à rede de pacotes de dados (PDN) é realizada através de *PDN connection* que liga a terminal com a rede de núcleo da rede 3GPP (*3rd. Generation Partnership Project*). A PDN *connection* tem várias funções como: proporcionar conectividade IP, provisionamento de largura de banda, gerenciamento de QoS e mecanismos para o faturamento. Dentro do EPc ou do núcleo das redes 4G, as *PDN connections* são construidas através de EPS-*bearers* entre a terminal móvel (UE) e o P-GW.

As PDN *connections* suportam três possibilidades de alocação de IP ao UE: só IPv4, só IPv6, ou os dois tipos de endereços IP simultáneamente. Esta alocação é realizada durante o procedimento

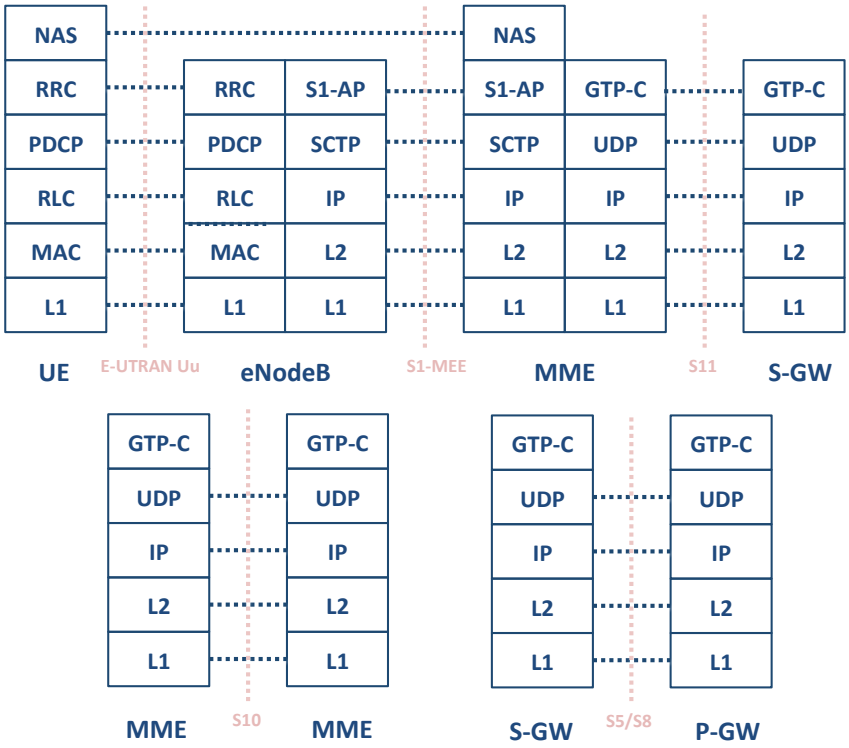

Figura 2.17: *Stack* para o plano de controle 4G utilizando GTP-c (UE–eNodeB, eNodeB–MME, MME–S-GW, MME–MME e S-GW–P-GW)

de *attachment* do UE (*PDN context activation*), ou imediatamente depois, utilizando DHcPv4 (*Dynamic Host configuration Protocol* v4) para a alternativa IPv4 ou *stateless IPv6 address autoconfiguration* para IPv6.

Uma UE pode ter associada mais de uma *PDN connection*, e cada uma delas pode estar constituída por varias EPS-*bearers*. Uma *default bearer* é sempre assignada obrigatoriamente para o tráfego geral dessa PDN, e *bearers* específicos opcionais podem ser ativados quando uma aplicação requer de conectividade com caraterísticas de QoS específicas. Os TFT são os que especificam as regras de coincidência para classificar o tráfego e encaminhá-lo pelos apropriados EPS-*bearers*. Quando um novo EPS-*bearer* é criado, também deve ser construido o TFT ligado a este (a Figura 2.18 mostra a relação entre os diferentes conceitos). É importante notar que estes TFT são aplicados no UE para classificar o tráfego no sentido *Uplink* e no P-GW para classificar o tráfego no sentido *Downlink* (ver tabela 2.2). Os TFT estão constituidos por regras (ou filtros de tráfego) que examinam os campos dos pacotes, procurando coincidências. Principalmente estas coincidências estão vinculadas ao endereço IP origem/destino, protocolo de transporte, e às portas origem/destino. Também podem ser referenciados grupos de endereços IP, ToS Ipv4, MPLS-Tc IPv6 ou *flow label* IPv6 entre outros.

Na arquitetura 4G só o P-GW pode ativar, apagar, modificar e decidir as regras de classificação dos EPS-*bearers*, o que mostra uma forte centralização e dependência de toda a arquitetura nos P-GW. Na Figura 2.19 se mostra a terminologia dos *bearers* utilizada para as diferentes partes

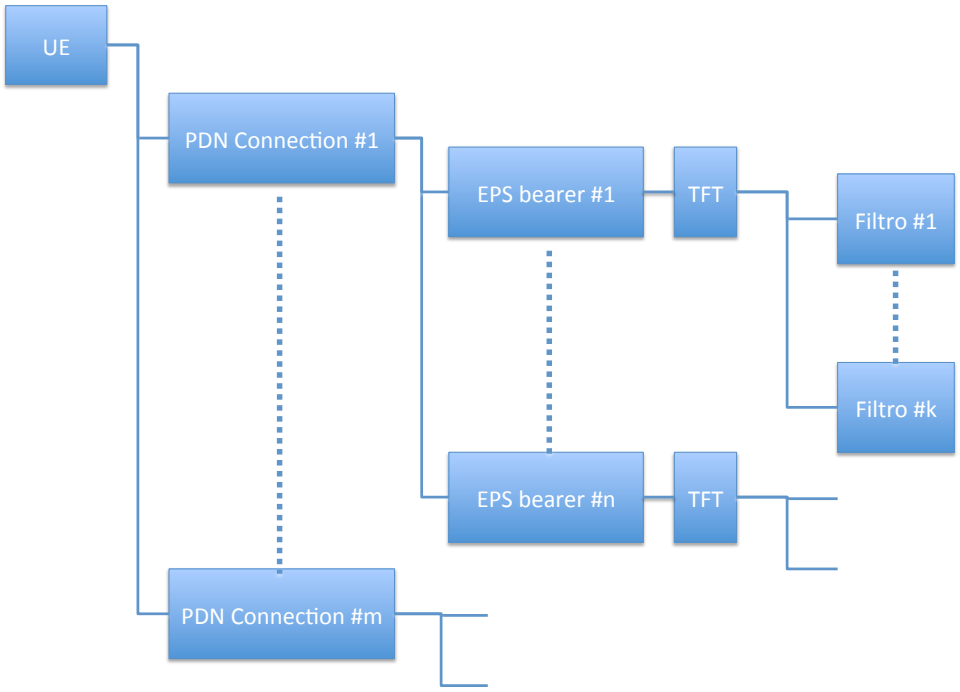

Figura 2.18: Relação entre PDN *connection*, *bearer*, TFT e regras de filtro em um UE

que compõem uma comunicação fim a fim entre um dispositivo UE e um equipamento externo pertencente à Internet. Para criar um fluxo QoS fim a fim tem que ser construido um EPS-*bearer* dentro do domínio 4G, e um *bearer* externo para Internet (este último fora do domínio do operador móvel). Adicionalmente dentro do domínio 4G, o EPS-*bearer* pode se dividir em S5/S8 *bearer* e E-RAB, onde a última está composta de um rádio *bearer* e um S1 *bearer*.

Os EPS-*bearers* possuem dois parâmetros principais associados com ele: o QcI (QoS class Identifier) e o ARP (*Allocation and Retention Priority*). Adicionalmente os EPS-*bearers* também podem ter associada uma largura de banda ligada, mas só isto acontece no caso de se tratar de EPS-*bearers* do tipo GRB (*Guaranteed Rate Bit*). A seguir são descritos cada um desses parâmetros:

QcI (*QoS class Identifier*): O EPS utiliza os conceitos de QoS baseados em classes, vinculando um grupo de características de QoS (como o limiar de admissão, gerenciamento das filas de prioridade, configuração de protocolos de camada de enlace de dado) a um valor de QcI. Estes valores são predefinidos em cada nó, e o valor específico do QcI não representa nenhuma caraterística de QoS por se só, dependem do mapeamento de características de QoS com o valor de QcI escolhido. Entretanto, existem alguns valores de QcI padrões que podem ser utilizados e que identificam distintas características de serviço pré-estabelecidas (ver Tabela.2.1).

ARP (*Allocation and Retention Priority*): Este parâmetro é utilizado para indicar a prioridade de alocação e retenção que um EPS-*bearer* tem. Isto inclui:

- *Priority level*: Solicitações com maior prioridade de estabelecimento e modificação são preferidas em situações de baixos recursos.

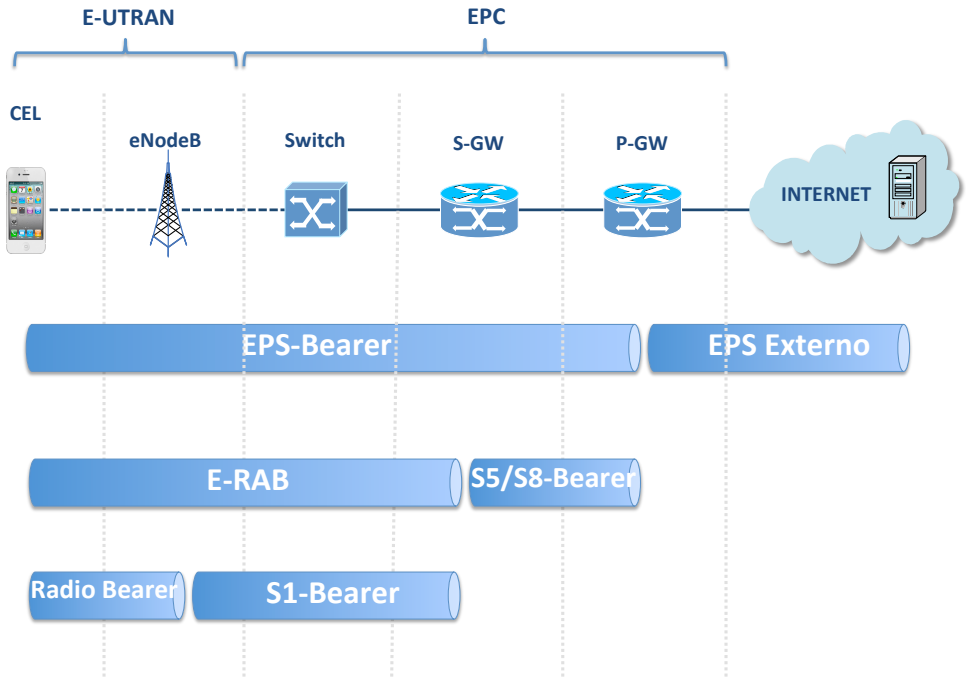

Figura 2.19: Terminologia utilizada para os diferentes *bearers*

- *Preemtion capability*: é um parâmetro lógico que indica que a solicitação de criação de um EPS-*bearer* com este parâmetro em *true* pode apagar outro EPS-*bearer* de prioridade inferior.
- *Preemtion vulnerability*: Se é *true* este EPS-*bearer* pode ser apagado durante o estabelecimento de outro EPS-*bearer* com *priority level* superior.

Como foi previamente mencionado, o EPS diferencia dois tipos de *bearers*, os GBR e os Non-GBR. O primeiro caso trata de serviços onde é melhor não permitir novas conexões que degradar os serviços preexistentes do mesmo tipo (são os casos de VoIP e conversações de vídeo, que se caracterizam por requerer uma largura de banda mínima constante). Para os *EPS-bearer* do tipo GBR é possível definir o parâmetro GBR, que representa a largura mínima de banda garantida e o MBR (*Maximum Bit Rate*), que representa a máxima largura de banda permitida para este *EPS-bearer* (em caso que se disponha de largura de banda ociosa no enlace). Nos casos de Non-GBR *EPS-bearers*, não podem ser definidas larguras de banda mínimas garantidas; no seu lugar, podem ser definidos dois parâmetros de controle de largura de banda: o APN-AMBR (APN-*Aggregate Maximum Bit Rate*) que define a largura de banda máxima permitida de todos os Non-GBR *EPS-bearer* de uma PDN/UE específica, e o UE-AMBR que limita o tráfego de todos os Non-GBR EPS-*bearer* referentes a um UE específico.

As políticas de controle da largura de banda definida pelo parâmetro UE-AMBR no sentido UL(*Uplink*) e DL(*Downlink*) são realizadas pelo eNodeB, que desta forma limita o tráfego máximo de todos os Non-GBR EPS-*bearers* referentes a um UE específico. Por outro lado, o controle referente ao parâmetro APN-AMBR (que define a largura de banda máxima de todos os Non-GBR

Tabela 2.1: correspondência entre tipos de tráfego e QcI padrões

QcI	**Tipo de *Bearer***	**Priori-dade**	**Atraso pocotes**	**Perda pacotes**	**Exemplo de uso**
1	GBR	2	100 ms	10^{-2}	chamadas de VoIP
2	GBR	4	150 ms	10^{-3}	chamadas de vídeo
3	GBR	3	50 ms	10^{-3}	Jogos online (tempo real)
4	GBR	5	300 ms	10^{-6}	*Streaming* de vídeo
5	Non-GBR	1	100 ms	10^{-6}	Sinalização IMS
6	Non-GBR	6	300 ms	10^{-6}	Serviços baseados em TcP como: chat, ftp, e-mail, web, etc
7	Non-GBR	7	100 ms	10^{-3}	voice, vídeo, jogos interativos
8 e 9	Non-GBR	8 e 9	300 ms	10^{-6}	Vídeo, serviços baseados em TcP como: chat, ftp, e-mail, web, etc

EPS-*bearer* de uma UE dirigidos a uma PDN específica) é controlado no P-GW em ambas direções. Finalmente o GBR e o MBR dos GBR EPS-*bearers* é controlado pelo P-GW no sentido DL e pelo eNodeB no sentido UL.

Finalmente se menciona que o HSS (*Home Subscriber Server*) é quem define cada PDN *subscription context*, EPS-*subscriber QoS profile*, contém o nível dos parâmetros de QoS para o *default bearer* (QcI e ARP) e o valor do APN-AMBR. Adicionalmente, no caso de não ser especificados valores de ARP particulares para a criação de *bearers* específicos, o P-GW utilizará o ARP do *default bearer*. Para uma maior compreensão, e como sumarização dos aspectos previamente mencionados, a tabela 2.2 indica as entidades encarregadas do controle de cada um dos parâmetros ligados aos EPS-*bearer*.

2.6.2 Política de controle e cobrança nas redes 4G

O Pcc (*Policy and charging control*) é uma arquitetura de controle centralizada para proporcionar serviços com o tratamento de QoS apropriado, método de cobrança e largura de banda. O Pcc possibilita o controle tanto dos serviços IMS (*IP Multimedia Subsystem*) quanto dos non-IMS.

Dentro do contexto do Pcc, o termo *bearer* é um conceito mais geral, que identifica os fluxos IP que são tratados com as mesmas características. Todo o tráfego que é transportado pelo mesmo *bearer* recebe o mesmo tratamento referente a QoS. O Pcc adiciona uma maior precisão de gerenciamento adicionando um novo conceito denominado *service session* onde múltiples *service sessions* podem ser transportados por um mesmo *bearer*, mas podem ser aplicados mecanismos de QoS e cobrança particulares a cada *service session*.

A arquitetura geral do Pcc no EPS é mostrada nar Figura 2.20 e está constituida por diferentes

Tabela 2.2: controles/ações realizadas e entidades responsáveis

	UE	**eNodeB**	**Rede de transporte móvel**	**P-GW**
Filtragem de pacotes	X (*Uplink*)			X (*Downlink*)
Admissão de GBR/ARP		X		X
ARP *preemtion*		X		X
Gerenciamento de filas	X	X		
Controle de largura de banda (GBR, MBR)		X (*Uplink*)		X (*Downlink*)
Controle de largura de banda (non-GBR/UE-AMBR)		X (*Uplink*)		X (*Downlink*)
Controle de largura de banda (Non-GBR/PDN-AMBR)	X			X
Uplink + Downlink Scheduling		X		

entidades interligadas, descritas a seguir:

- AF (*Application Function*): Esta entidade interatúa com os serviços que requerem um Pcc dinâmico. Os AF são colocados em pontos estratégicos de forma que a sinalização dos serviços atravessem ou terminem em um AF. Desta forma, o AF tem consciência do início, mudança ou finalização de serviços que requerem de um tratamento de QoS específico, e o AF informa estes acontecimentos ao PcRF utilizando a interface Rx. Adicionalmente o AF pode se subscrever a eventos que acontecem no plano de tráfego (eventos detectados pelo PcEF-*Policy and charging Enforcement Funcion* ou o BBERF-*Bearer Binding and Event Reporting Function*) e neste caso é o PcRF quem informará de estes eventos ao AF quando estes aconteçam.

- SPR/UDR (*Subscriber Profile Repository / User Data Repository*): Representam duas alternativas de base de dados para guardar informações de subscrição específicas que determinam as políticas que devem ser aplicadas a cada usuário.

- OcS (*Online charging System*): é o sistema de gerenciamento de créditos para o sistema de faturamento pré-pago.

- OFCS (*Offline Charging System*): é utilizado para a cobrança dos serviços *off-line*. Este recebe os eventos de cobrança desde o PCEF e gera os cDR (*charging Data Record*) transferidos ao sistema de faturamento.

- PCRF (*Policy and Charging Rule Function*): é a entidade central para o controle de políticas

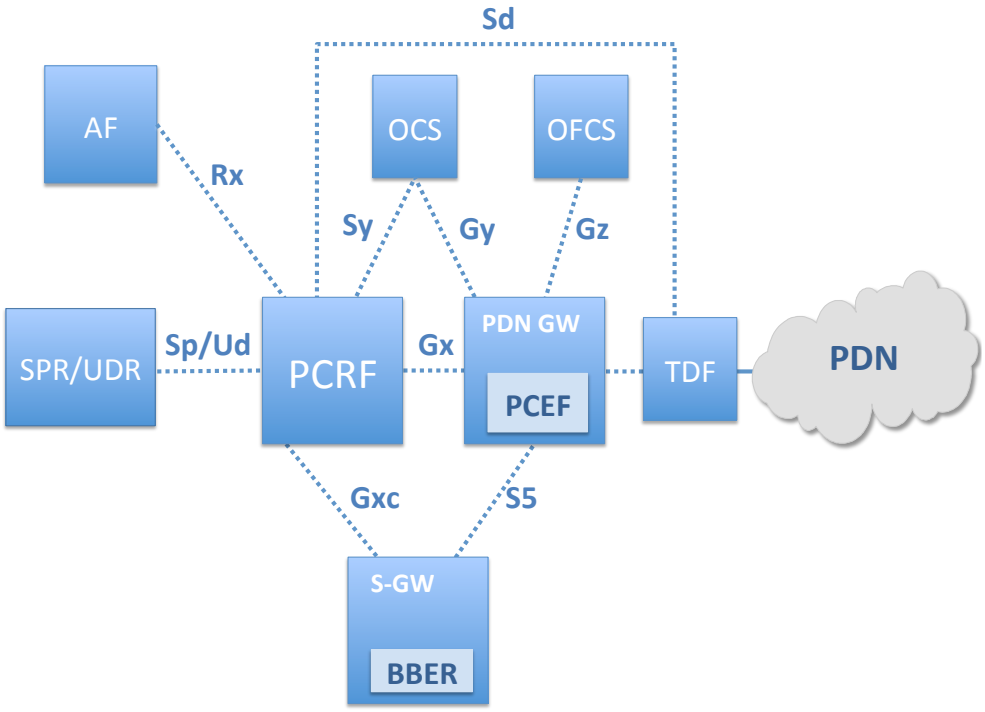

Figura 2.20: Arquitetura de controle Pcc

do Pcc. Este recebe informação de seção pela interface Rx, informação da rede de acesso através da interface Gx e informação de subscrição pela interface SP/UD (ver Figura 2.20). Com estas informações decide as políticas a serem aplicadas. Estas decisões são enviadas ao PcEF, o encarregado de sua aplicação.

- PCEF (*Policy and Charging Enforcement Function*): é o responsável pela execução das decisões tomadas pelo PcRF (como permissão de acesso, limitação da largura de banda, etc.). O PcEF também é o encarregado de realizar medidas de tráfego; reporta o uso de recursos ao OFcS e interatua como o OcS para o gerenciamento do crédito. No 3GPP *release* 11 o PcEF também pode implementar detecção de aplicações e controle (ADc - *Application Detection and control*).

- TDF (*Traffic Detection Function*): é uma entidade que fornece a funcionalidade ADc, utilizando para isto a inspeção de pacotes e realizando um reporte das aplicações detectadas ao PcRF.

Dentro do contexto do PCC, a política que este aplica se refere às funções de permissão de acesso e de controle de QoS. A decisão de permissão de acesso é efetuada pelo PcRF, e a execução é realizada pelo PCEF. O controle de QoS também é realizado pelo PcRF, que tem a informação do QoS *class* e a largura de banda a ser aplicada ao fluxo específico. A implementação da política de QoS é efetuada pelo PCEF, quem controlará a largura de banda e colocará o fluxo em um *bearer* apropriado.

O PcRF é a entidade central do Pcc e responsável das decisões a serem tomadas, as que se

baseiam em:

- Configurações realizadas pelo operador no PcRF, para definir políticas de serviços
- Informação de subscrição recebida pelo SPR
- Informação referente a serviços enviada pelo AF
- Informação do TDF ou o PcEF referente a aplicações detectadas
- Informação recebida desde o sistema de faturamento, referente aos créditos disponíveis
- Informação da rede de acesso indicando o tipo de tecnologia utilizada.

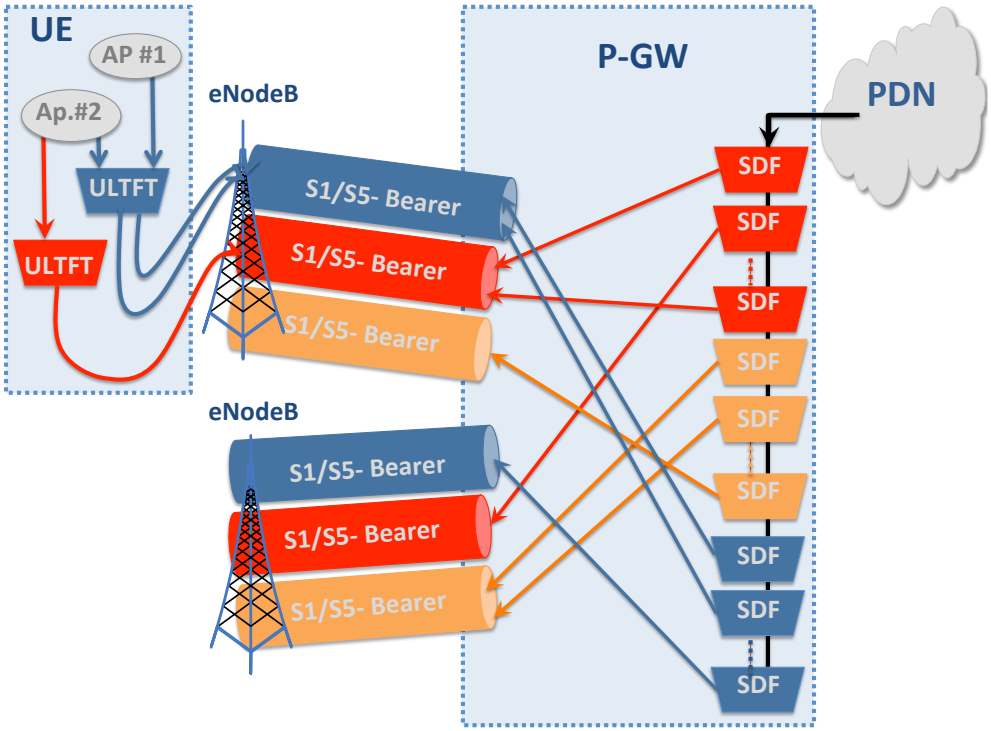

Figura 2.21: conceptualização de SDF, EPS-*bearer* e TFT

As decisões tomadas pelo PcRF são entregadas na forma de "*PCC rules*", e também formando parte de um subgrupo de informação denominada "*QoS rules*". As *Pcc rules* contêm informação em forma de *template* denominada SDF (*Service Data Flow*), e a todos aqueles pacotes que coincidam com o filtro especificado em um SDF *template* particular lhes é atribuído esse SDF. O SDF *template* contém a descrição do fluxo (IP origem e destino, protocolo, porta origem/destino e outros parâmetros), o estado de porta de acesso (*open/closed*) e a informação relativa à política de QoS e faturamento do SDF. No que refere à informação de QoS, a mesma contém o QcI, MBR, GBR e ARP, mas se bem estes têm a mesma nomenclatura que foi definida para os *bearers*, os fluxos definidos pelo SDF *template* têm maior precisão, de forma que vários fluxo definidos por diferentes Pcc *rules* podem utilizar uma mesma *bearer*. A Figura 2.21 ilustra a relação dos SDF

colocados no P-GW (PCEF) com o EPS-*bearers* e com os TFT aplicados no UE, utilizando para sua visualizaçõ, a topologia previamente escolhida.

A seguir é descrito o funcionamento do PCC para o estabelecimento de um novo serviço (ver Figura 2.22):

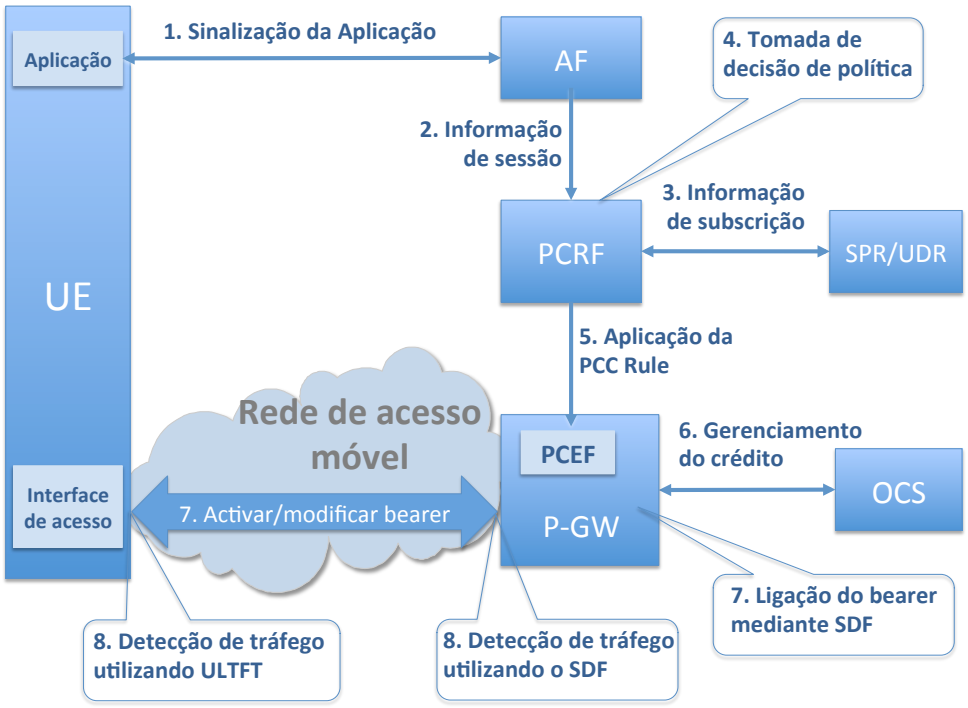

Figura 2.22: Procedimento de ativação de um *bearer* com requisitos de QoS na arquitetura Pcc

1. O usuário inicia um serviço, por exemplo uma chamada de VoIP por IMS, e executa a sinalização da seção da aplicação fim a fim, que é interceptada pelo AF. No caso de IMS, a sinalização da aplicação utiliza o protocolo SIP (*Session Initiation Protocol*) e a descrição do serviço é proporcionada pela própria sinalização do protocolo (no protocolo SIP, o SDP - *Session Description Protocol*, é utilizado para descrever a seção).

2. O AF, baseado na descrição do serviço obtida, envia ao PCRF a informação relativa ao serviço utilizando a interface Rx. Esta contém a informação relativa a QoS (tipo de serviço, largura de banda requerida) e a informação requerida para identificar o fluxo IP (IP origem e destino, protocolo, porta origem e destino, etc).

3. O PCRF solicita informação relacionada à subscrição do usuário particular ao SPR ou UDR. Este passo pode não ser necessário se a informação já foi solicitada previamente para este usuário.

4. O PCRF, baseado na informação da seção, as políticas definidas pelo operador referente ao serviço, a informação de subscrição e outros dados, define a política a aplicar construindo para isso as Pcc *rules*.

5. As PCC *rules* são enviadas desde o PcRF ao PcEF, quem executa as ações definidas nas Pcc *rules*.

6. Se as PCC *rules* especificam que o faturamento tem que ser *online*, o PcEF tem que se contatar com o OcS via a interface Gy, com a finalidade de solicitar o crédito disponível (o método de medição do crédito também é definido na Pcc *rule*).

7. O P-GW (particularmente o PCEF) instala as PCC *rules* e vincula estas com um *bearer* apropriado. Esta ação pode requerer da criação de um novo *bearer* ou modificar ou utilizar um já existente.

8. Finalmente os dados para este serviço são transportados pela rede utilizando o *bearer* apropriado, e o PCEF utiliza o filtro do SDF *templates* para identificar o fluxo específico para este serviço.

2.7 *Softwares* utilizados para a construção dos protótipos

Para a realização dos protótipos desenvolvidos no capítulo 5, foram utilizados os programas Mininet para a construção da topologia, OpenvSwitch como *software* dos clientes *OpenFlow*, e Ryu para o Controlador. Nas seguintes subseções se descreve cada um deles.

2.7.1 Mininet

Mininet [24] [25] é um emulador de rede capaz de criar uma topologia virtual com *hosts*, *switches*, Controladores e enlaces virtuais. Mininet executa um *software* de rede padrão de Linux e seus *switches* executam OpenFlow.

Este emulador de rede é um projeto *open source* que permite criar uma rede de provas completa sobre um só computador, possibilitando assim a pesquisa, o desenvolvimento, a aprendizagem e a criação de protótipos e testes, sem nenhum custo. Adicionalmente, Mininet otimiza o comportamento do sistema anfitrião, ainda quando é utilizado para grandes topologias. Estas topologias podem ser compostas por um grande número de *hosts*, *switches OpenFlow*, enlaces virtuais e Controladores.

As redes Mininet executam um código real, incluindo aplicações de redes Unix/Linux padrão, assim como o *kernel* de Linux e a camada de rede. É por isso que Mininet proporciona muito mais que um simples simulador, devido a que um projeto ou código desenvolvido e testado em Mininet, seja para um Controlador, *switches* OpenFlow ou *hosts*, pode ser colocado em um sistema real praticamente sem a necessidade de realizar mudanças. Mininet combina muitas das melhores características de emuladores, redes experimentais em *hardware* e simuladores.

Na comparação com abordagens baseadas em virtualização completa do sistema, Mininet tem as vantagens indicadas a seguir:

- inicia mais rápido (só em segundos);

- possibilita a criação de topologias com uma grande dimensão, pudendo atingir centenas de *hosts* e *switches* interligados;
- fornece uma maior utilização da largura de banda;
- é mais fácil de ser instalado.

No caso em que é comparado com redes experimentais em *hardware*, Mininet tem as vantagens indicadas a seguir:

- é gratuito e está sempre disponível;
- possibilita que as topologias sejam reconfiguradas de uma forma muito rápida.

Finalmente, se a comparação é realizada com simuladores, Mininet tem as vantagens indicadas a seguir:

- executa o código real, sem modificações, incluindo o código de aplicativo, o código do núcleo (*kernel*) do sistema operacional e o código do plano de controle (o código do Controlador *OpenFlow* e o código de OpenvSwitch);
- construi topologias que se podem conectar fácilmente ás redes reais;
- tem possibilidade muito grande de crescimento.

A Figura 2.23 faz uma descrição em forma simplificada da criação por Mininet de uma topologia básica, formada por dois *hosts*, um *switch OpenFlow* que os interconecta e um Controlador. Inicialmente, o lançador de Mininet cria dois processos *bash* que representam o *Host1* e o *Host 2*, cada um deles com seu próprio espaço de nomes. Seguidamente, são criadas duas duplas de *Ethernet* virtuais, e são designados nomes do *name space*. Posteriormente, é criado um *switch OpenFlow* para interconectar os *hosts* e finalmente é criado um Controlador *OpenFlow* com o canal de controle estabelecido.

É importante notar que o Controlador que está incluido dentro do Mininet é o Controlador NOX, só com suporte para OpenFlow 1.0. Por esta razão, para a criação de nosso protótipo, foi necessário instalar o Controlador externo Ryu, com suporte para *OpenFlow* 1.3 e superior. adicionalmente, o software utilizado por Mininet para a virtualizacão dos *switches* só implementa a versão 1.0 de *OpenFlow*, pelo qual também foi modificado para o protótipo, utilizando OpenvSwitch com suporte para *OpenFlow* 1.3.

2.7.2 OpenvSwitch

Open vSwitch [26] [27] é um software de *switch* multicamada construido para ambientes virtuais que utiliza a licença de código aberto apache 2.0. O objetivo de Open vSwitch é implementar uma

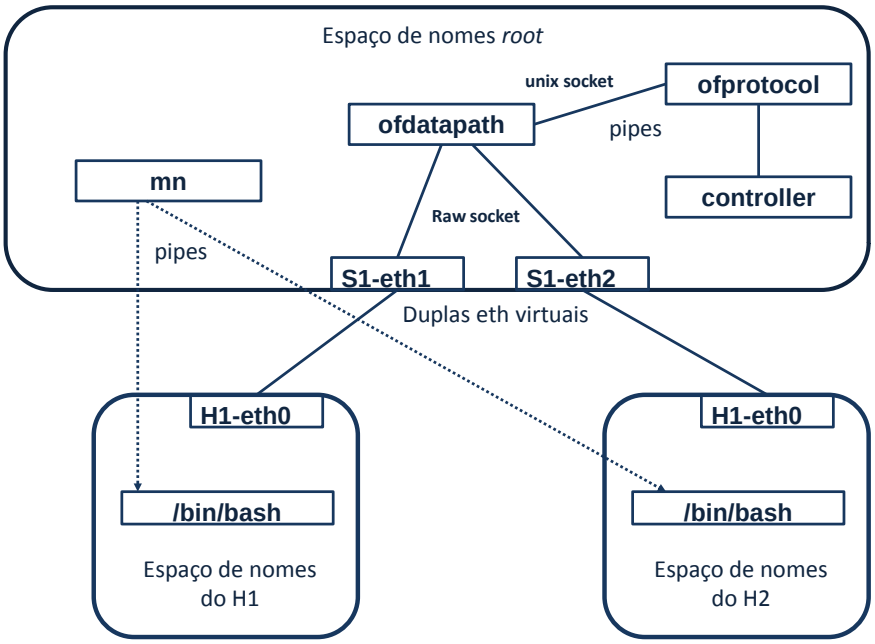

Figura 2.23: arquitetura exemplo de Mininet

plataforma para *switches* de qualidade que seja capaz de dar suporte a interfaces de gerenciamento padrão e ser aberto para as funções de controle e roteamento programáveis.

Este *software* reside dentro do *hypervisor* ou domínio de gerenciamento e fornece conectividade entre as máquinas virtuais e as interfaces físicas. A arquitetura geral de Open vSwitch pode ser observada na Figura 2.24.

Open vSwitch fornece uma interface local de gerenciamento através da qual a camada de virtualização pode manipular a configuração topológica. Isto inclui a criação de *switches* (vários *switches* virtuais podem ser criados em um único *host* físico), gerenciamento de conectividade das interfaces virtuais (para cada interface virtual é adicionada uma porta lógica em correspondência) e gerenciamento da conectividade com as interfaces físicas.

Adicionalmente o OpenvSwitch fornece uma interface que permite a configuração dos *switches* virtuais, similar ás interfaces de gerenciamento dos *switches* físicos. Também possibilita a manipulação remota do plano de dados de seus *switches* virtuais (tabelas de fluxos), especificando como processar os pacotes com base nos cabeçalhos de camadas 2, 3 e 4. A interface para a manipulação das tabelas é implementada por meio do protocolo OpenFlow.

O código é escrito em plataforma C independente, e é fácil de ser transportado para outros ambientes. Dentro das características mais importantes que Open vSwitch dá suporte estão:

- visibilidade em comunicações entre máquinas virtuais;
- modelo padrão IEEE 802.1Q;

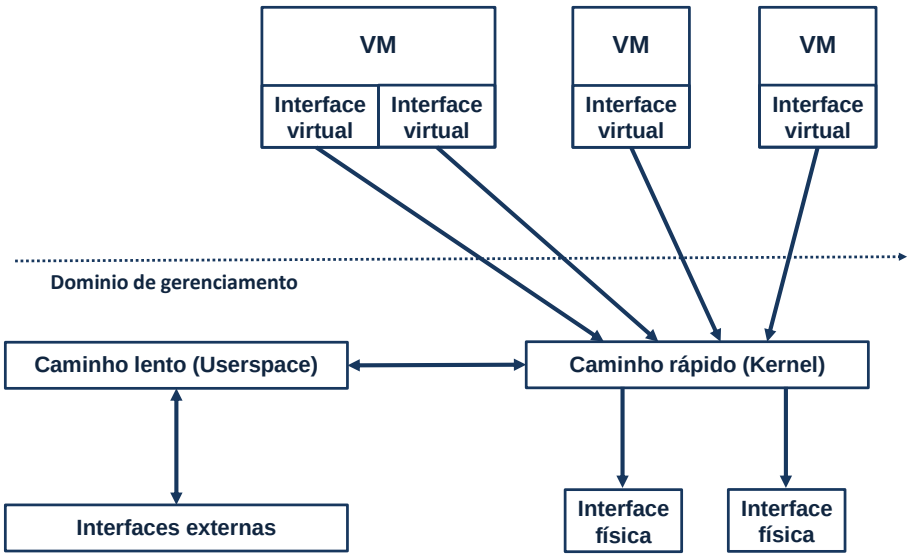

Figura 2.24: arquitetura de Open vSwitch

- STP(*Spanning Tree Protocol*);
- precisão para o controle de QoS;
- controle de tráfego por cada interface de máquina virtual;
- suporte a *OpenFlow* em suas diferentes versões;
- IPv6 (Internet Protocol v6);
- linha de processamento com múltiples tabelas com mecanismos de cachê de fluxos.

2.7.3 Controlador Ryu

O Controlador Ryu [28] é um componente criado para o contexto das redes definidas por *software*. Todo seu código está disponível gratuitamente com a licença de apache 2.0 e está completamente escrito na linguagem de programação *Python*, facilitando assim, a criação de novas aplicações de controle (Figura 2.25).

Ryu fornece componentes de *software* com aPI muito bem definidas, o que facilita a criação de novas formas de gerenciamento de redes e aplicações de controle aos desenvolvedores. Ryu dá suporte a vários protocolos para o gerenciamento dos equipamentos de rede como Netconf, OF-config, *OpenFlow*. Particularmente, para este último, dá suporte ás versões 1.0, 1.2 e 1.3, característica essencial para sua utilização no protótipo desenvolvido neste trabalho.

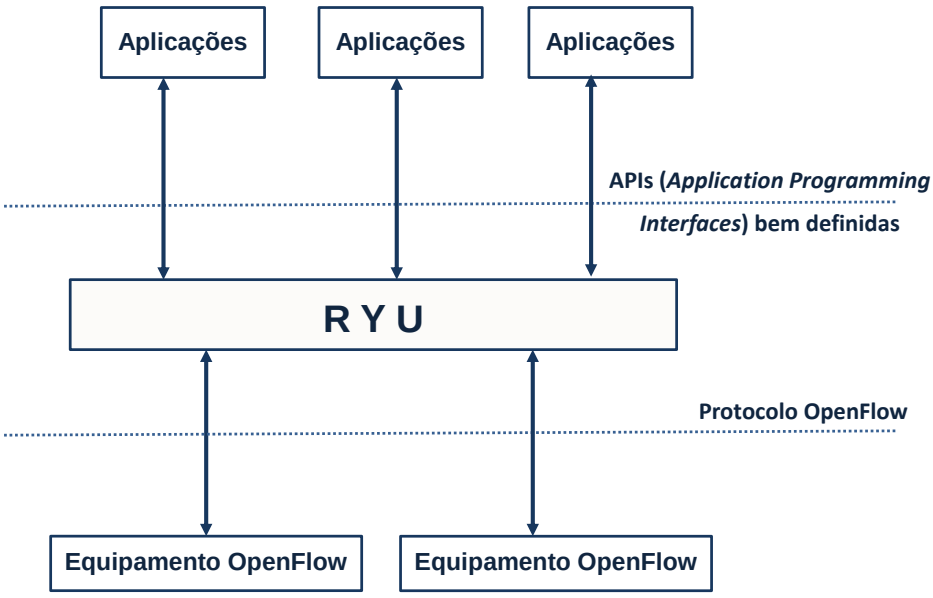

Figura 2.25: Controlador Ryu

Capítulo 3

Proposta de arquitetura para o domínio de transporte

3.1 Introdução

Apesar do amplo desenvolvimento e implantações nas redes de dados utilizando SDN, ainda não é incomum a discussão sobre qual é o método mais apropriado de gerenciar o plano de controle nas redes de dados. A questão central consiste em definir as virtudes e as desvantagens de utilizar o controle de rede distribuído comprovado e implementado hoje na indústria (implementado, por exemplo com OSPF – *Open Shortest Path First* e ISIS – *Intermediate System to Intermediate System protocols*) e o controle centralizado baseado em SDN [29, 30, 31] que possibilita uma ampla programabilidade da rede.

Amplamente utilizado por muitos anos nas redes de dados, o controle distribuído é uma escolha robusta que permite que as redes de dados se adaptem às mudanças topológicas sem requerer a existência de um nó com funções de controle especiais. O processamento encontra-se distribuído entre todos os nós da rede, os quais são responsáveis por calcular os melhores caminhos para todos os destinos a partir de sua própria perspectiva. Apesar de sua ampla adoção, as redes IP tradicionais são relativamente estáticas, complexas e difíceis de serem gerenciadas, necessitando de mecanismos complexos para manter as informações da rede sincronizadas e consistentes entre os nós [32, 33, 34, 35, 36]. Além disso, se o MPLS é adicionado ao controle distribuído juntamente com as soluções de engenharia de tráfego e VPN (Virtual Private Network), a complexidade e os requerimentos de troca de informações entre os nós aumenta ainda mais. Além disso, os equipamentos de rede vêm de um número reduzido de fabricantes que mantêm o código-fonte inacessível e com algumas funcionalidades proprietárias. Este fato dificulta a interoperabilidade entre equipamentos, torna a inovação mais lenta e condicionada aos interesses do fabricante e reduz as possibilidades de engenharia de tráfego.

Em uma arquitetura de rede baseada em um controle lógicamente centralizado, é utilizado um elemento central, o Controlador, que tem a visão global da rede e utiliza um protocolo padrão (por exemplo, *OpenFlow*) para gerenciar cada nó da rede. Como consequência, o plano de dados é im-

plementado utilizando elementos de rede mais simples, cujo objetivo é essencialmente encaminhar pacotes com base nas informações de rede indicadas pelo Controlador. Essa arquitetura é flexível, programável, com nós de rede padrão e oferece possibilidades ilimitadas de engenharia de tráfego. Com esta opção, os nós são facilmente substituíveis e a inovação é mais rápida e ilimitada em diversas áreas como: engenharia de tráfego [4, 5, 6, 37], computação em nuvem [38, 39], gerenciamento de mobilidade [40, 41, 42], e varias outras, gerando particular interesse na academia e operadores de rede. No entanto, são desvantagens do controle centralizado a excessiva dependência de rede do Controlador, a grande quantidade de informações que o Controlador deve processar e o atraso adicional na criação de novos fluxos com requerimentos específicos, já que o pacote deve aguardar a resposta do Controlador para encaminhar os pacotes [43]. Tais desvantagens reduzem a robustez e limitam a escalabilidade da rede.

Diversos trabalhos tem sido propostos para lidar com as desvantagens do controle centralizado. Como exemplo, [44, 45, 46, 47] propoem o uso de um grupo de Controladores distribuídos geograficamente, mas trabalhando lógicamente como uma só entidade com a visão global da rede. Uma classificação de diferentes implementações que utilizam Controladores SDN distribuídos pode ser encontrada em [48]. A distribuição geográfica dos Controladores aumenta a robustez da rede, distribui a carga e incrementa a escalabilidade da solução. No entanto, o uso de vários Controladores sempre envolve uma compensação entre a consistência e a disponibilidade da rede, e incrementa os riscos de *loops* ou *black holes* [49]. Além disso, a comunicação adicional requerida entre os Controladores pode incrementar o atraso na criação de novos fluxos.

Por outra parte, em [50, 51, 52] propõem a adição de nós SDN específicos em pontos estratégicos de uma rede distribuída padrão, para assim aprimorar as possibilidades de engenharia de tráfego e melhorar o balanceamento de carga. Para isso, o Controlador deve intercambiar mensagens de protocolos de encaminhamento distribuído entre os *routers* e nós SDN. Esta alternativa requere que o Controlador implemente protocolos de encaminhamentos distribuídos padrão, tornando esta alternativa complexa de ser implantada. Além disso, os nós com protocolos distribuídos padrão são mais limitados em termos de recursos de engenharia de tráfego quando comparados com os nós puramente SDN. Em [53, 54] é realizado um levantamento de diferentes alternativas híbridas, e em [49] os autores apresentam os problemas que podem ocorrer, como *loops* ou *black hole*, quando vários protocolos de controle coexistem no mesmo equipamento. Particularmente, é analisado o uso simultâneo de um protocolo de controle distribuído e centralizado.

Neste trabalho, propõe-se uma abordagem diferente para o controle de rede com SDN. Em vez de se concentrar na redundância do Controlador, o foco é melhorar o gerenciamento e a robustez da rede [17] para reduzir as fraquezas do plano de controle centralizado. A arquitetura de rede SDN-*OpenFlow* proposta implementa uma nova aplicação de controle de rede que é capaz de gerenciar o tráfego com e sem requisitos de QoS e reduzir substancialmente a dependência da rede no Controlador. Além disso, a arquitetura proposta diminui o número de mensagens que o Controlador deve processar e elimina os tempos de espera requeridos para o encaminhamento de novos fluxos.

Adicionalmente, é incorporado nos testes um mecanismo para impedir a duplicação de men-

sagens ao Controlador (PIDAM – *Packet-in Duplication Avoid Mechanism*) [55] e é incorporada na arquitetura o plano de dados MPLS. Além disso, a arquitetura pode ser implementada usando equipamentos puramente OpenFlow ou equipamentos híbridos, sendo este último definido como um equipamento que implementa o plano de controle distribuído padrão e o protocolo OpenFlow simultâneamente. A implementação da arquitetura proposta com equipamentos híbrido cria uma alternativa que incorpora as fortalezas do controles distribuídos e do centralizado. Além disso, a alternativa com equipamentos híbridos fornece um cenário de migração adequado se quer passar de um controle distribuído a um controle centralizado baseado em *OpenFlow*.

3.2 Arquitetura de transporte proposta

Este capítulo apresenta uma proposta de arquitetura para a implementação do domínio de transporte do SP. Esta arquitetura é implementada utilizando a tecnologia SDN-*OpenFlow*, e é construída para que seja capaz de porporcionar serviços de transporte diferenciados segundo os requisitos do tráfego a ser transportado. Os serviços de transporte oferecidos permitem a interconectividade entre os diferentes domínios do SP assim como entre clientes de grande porte diretamente conectados à rede de transporte. Adicionalmente, a rede de transporte oferece interconexão à Internet, permitindo o encaminhamento do tráfego para e desde um destino fora do SP.

A arquitetura de rede de transporte de SP desenvolvida neste capítulo é construída utilizando o protocolo *OpenFlow* e seu plano de dados implementado com MPLS [7]. A tecnologia MPLS proporciona uma correspondência perfeita entre fluxo e rótulo MPLS, e permite que as decisões de encaminhamento de tráfego sejam mais simples e mais rápidas. A utilização de MPLS proporciona critérios de agrupamentos de tráfego com maior precisão, flexibilidade e abrangência ao mesmo tempo, permitindo mediante a utilização de um simples rótulo MPLS, representar um grupo de endereços IP disjuntos (origem/destino), protocolos, portas (origem/destino), e outros amplos critérios simultâneamente, que permitem classificar o tráfego para o encaminhamento de forma otimizada.

3.2.1 Lógica inicial

O fato de que *OpenFlow* permita a definição e o controle de fluxos individuais incrementa a precisão e as possibilidades de gerenciamento. Contudo, esta manipulação individual também leva a um maior processamento, atrasos no estabelecimento de novos fluxos, e carga no Controlador. É importante destacar que cada fluxo adicional requere a configuração da uma regra correspondente nas tabelas de fluxos de todos os equipamentos intervenientes no encaminhamento fim a fim dentro da rede de transporte, requerendo o intercâmbio de várias mensagens com o Controlador. Por esta razão, é importante diferenciar e definir as características dos diferentes tipos de tráfego na rede, analisando qual tráfego requere um tratamento fluxo a fluxo e qual requere de um tratamento geral, sempre tendo o objetivo de diminuir a carga no Controlador.

Na Figura 3.1 é mostrado um exemplo topológico de rede de transporte que facilita a visualização de conceitos desenvolvidos neste capítulo, mas a arquitetura desenvolvida não se restringe a este caso particular. Como foi previamente estabelecido, a arquitetura proposta no presente trabalho é implementada utilizando a tecnologia SDN-*OpenFlow*, e seu plano de dados está construido mediante a utilização de MPLS. Dentro do domínio SDN-*OpenFlow*-MPLS (que constitui a proposta de arquitetura de rede de transporte do SP) são utilizadas nomenclaturas para os equipamentos que têm relação com a utilizada no domínio MPLS (ver seção 2.5). é assim que são definidos dois tipos de equipamentos: os equipamentos de borda denominados OF-LER (que têm interfaces dentro e fora do domínio SDN-*OpenFlow*-MPLS) encarregados de receber o tráfego de fora do domínio MPLS, e que baseados nas caraterísticas do tráfego, colocam o cabeçalho MPLS para seu transporte dentro do domínio. Também os OF-LER são equipamentos de saída da rede SDN-*OpenFlow*-MPLS, encarregados de remover o cabeçalho MPLS e encaminhar o tráfego pela interface de saída apropriada. Por outra parte, os equipamentos internos são denominados OF-LSR (que possuem todas suas interfaces dentro do domínio SDN-*OpenFlow*-MPLS) e baseiam as decisões de encaminhamento só mediante a inspeção do rótulo MPLS (ver Figura 3.1). Na nomenclatura dos equipamentos foi adicionado o prefixo OF- que indica que estes equipamentos são implementados utilizando *OpenFlow*.

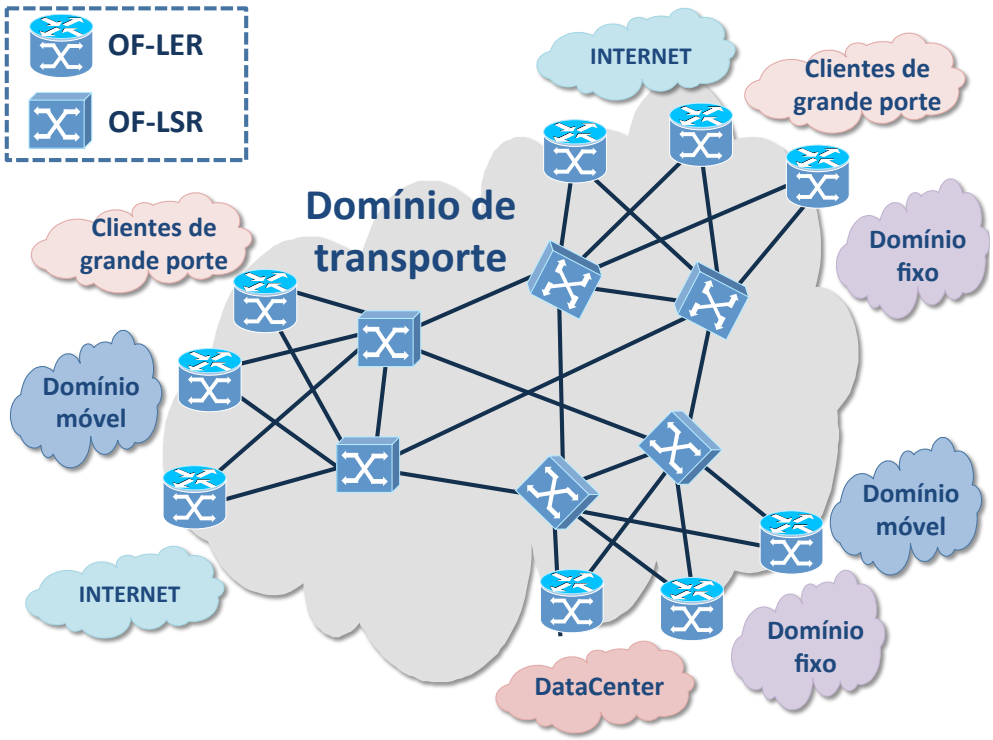

Figura 3.1: Exemplo de topologia de rede de transporte para a arquitetura proposta

Os pacotes que chegam aos equipamentos da rede de transporte (SDN-*OpenFlow*-MPLS) inicialmente são processados pelo plano de dados *OpenFlow*, onde são enviados à tabela inicial indexada como 0 (ver Figura 3.2). Esta tabela é criada inicialmente pelo Controlador em cada um dos equipamentos da rede de transporte, e não requere ser modificada por mudanças topológicas na rede (a tabela tem informação de serviços e processamento, não de topologia). A tabela 0 está constituída

por entradas de fluxo gerais iniciais que permitem classificar o tráfego utilizando o menor número de regras possíveis (como o IP-DSCP ou o MPLS-TC, ou grupo de endereços IP entre outros critérios). Em função desta classificação os pacotes são enviados a uma lógica de gerenciamento apropriada que permite cumprir os requisitos específicos do tráfego.

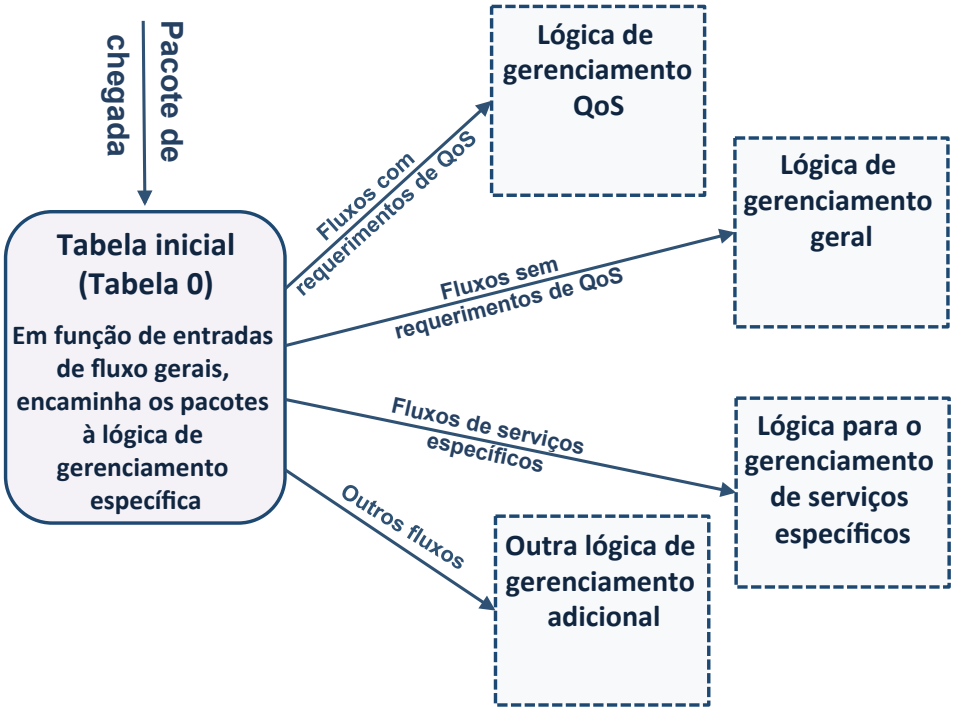

Figura 3.2: Lógica inicial de processamento

No presente trabalho são desenvolvidos três tipos de lógicas de gerenciamento:

- Lógica de gerenciamento geral: esta lógica é a encarregada de encaminhar todo o tráfego sem requisitos de QoS, utilizando entradas de fluxo gerais. Com esta lógica é possível o encaminhamento do tráfego pela rede de transporte para todos os destinos do SP.
- Lógica para o gerenciamento de QoS: permite encaminhar massivamente fluxos com requisitos de QoS fim a fim dentro do domínio de transporte. São exemplos de serviços com requisitos de QoS os fluxos de VoIP, *streaming* de vídeo, TV digital, entre outros.
- Lógica para o gerenciamento de serviços específicos: esta lógica permite cumprir com requisitos de serviços específicos de forma otimizada. Esta técnica não é aplicada para o tratamento de requisitos de QoS massivamente, só é utilizada para casos de interconectividade específica entre dois ou mais clientes da rede de transporte. São definidos como clientes da rede de transporte, outros domínios do SP (domínio móvel, fixo ou DataCenter) assim como os clientes de grande porte interconectados ao domínio de transporte diretamente.

Cada uma de estas lógicas de gerenciamento é implementada por uma tabela ou grupo de tabelas de fluxos interligadas dentro da linha de processamento de *OpenFlow* em cada um dos

equipamentos. Como será expressado ao longo do capítulo, cada lógica de gerenciamento específica varia em função do tipo de equipamento (OF-LER ou OF-LSR) e da localização de este na rede. Dependendo dessa localização e do fluxo particular a ser transportado, o OF-LER pode ser de saída ou de entrada para ao domínio de transporte. Nas seguintes seções são desenvolvidas as lógicas previamente descritas.

3.2.2 Lógicas de gerenciamento geral

Esta lógica está formada por uma tabela ou grupo de tabelas constituidas por entradas de fluxo gerais (não fluxo a fluxo) com a informação de encaminhamento para atingir todas as redes destino dentro do domínio do SP (similar ao que acontece nas redes atuais com protocolos de encaminhamento internos como OSPF e combinado com LDP). Estas tabelas são construidas proativamente pelo Controlador (antes da chegada dos fluxos) em cada um dos nós que compõem a rede de transporte, e são dinámicamente atualizadas quando uma mudança topológica acontece. Para isto, o Controlador deve possuir a informação dos nós da rede, suas interconexões (topologia da rede de transporte) e o conhecimento de todas as redes dentro do SP interconetadas aos OF-LER. Adicionalmente o Controlador deve ser informado pelos clientes *OpenFlow* (OF-LER e OF-LSR) de qualquer mudança topológica para atualizar as tabelas de fluxo gerais de todos os equipamentos da rede de transporte.

Com o conhecimento topológico e o conhecimento de todas as redes conetadas aos OF-LER fora do domínio de transporte, o Controlador executa o algoritmo de roteamento conveniente para descobrir como atingir todas as redes destino a partir de cada um dos nós. O resultado de este processamento é equivalente ao obtido pelos protocolos de roteamento atuais (as tabelas de roteamento). Seguidamente, o Controlador estabelece ligações dos rótulos MPLS com as redes destino em cada um dos nós, constituindo o denominado na terminología MPLS como FEC (ver seção 2.5). Com esta informação calculada e construida, o Controlador preenche as tabelas de fluxo gerais para todos os equipamentos do domínio SDN-*OpenFlow*-MPLS.

Estas entradas de fluxo na tabela de fluxos gerais devem incluir os campos de coincidência (*matching*), as ações de modificação, adição ou remoção de rótulos MPLS, assim como a porta de saída correspondente, entre outras ações. Nos casos em que o algoritmo de roteamento executado pelo Controlador encontre mais de um caminho ótimo, opcionalmente, ambos poderão ser considerados e adicionados nas tabelas correspondentes. Para isto, é possível configurar mediante *OpenFlow*, *Group Tables* [2], que permitem balancear o tráfego e assim melhorar a distribuição da carga na rede.

É importante ressaltar que o algoritmo de roteamento só é conhecido e executado no Controlador, responsável pela construção das regras de encaminhamento para todos os destinos em cada um dos nós. Nesta arquitetura, os equipamentos de rede não precisam trocar informação de roteamento nem de mudança topológica entre eles, não sendo necessário aguardar tempos de espera que assegurem a propagação da informação de forma certa na rede. Quando acontece a queda de um enlace ou nó, é suficiente que outro dos nós diretamente conetado a esta queda, perceba a falha e a informe imediatamente ao Controlador. Com esta informação, o Controlador calcula a nova

topologia, e realiza as mudanças nas tabelas de fluxos gerais para assim obter a nova convergência. Esta metodologia não requere da utilização de complexos algorítmos de propagação de informação e roteamento como OSPF-TE (*OSPF Traffic Engineering*) ou ISIS-TE (*ISIS-Traffic Engineering*), nem de distribuição de rótulos como LDP, o que diminui sensívelmente o processamento requerido nos equipamentos de rede. Entretanto, as tabelas construidas pelo Controlador são similares às obtidas com estes protocolos.

Com esta lógica de gerenciamento geral implementada, quando um pacote sem requisitos de QoS entra na rede de transporte por um OF-LER de entrada, e tem como destino uma rede do SP, o domínio de transporte o encaminha utilizando as entradas de fluxo gerais até a saída do domínio mais próximo ao destino final. Para isto o pacote é inicialmente processado pela tabela 0 do OF-LER de entrada, que envia o pacote para a tabela de fluxos geral (dentro da lógica de gerenciamento geral). Nesta tabela, encontra coincidência com uma entrada de fluxo geral que indica que tem que ser adicionado o cabeçalho MPLS (ação do tipo *Push*-MPLS) com o rótulo requerido para o encaminhamento de este pacote dentro do domínio de transporte, e envia o pacote pela porta física correspondente (ação do tipo *output*). Uma vez no OF-LSR o pacote MPLS é examinado pela tabela 0, que o encaminha à tabela de gerenciamento geral. Nesta tabela o rótulo MPLS encontra coincidência com uma entrada de fluxo, que opcionalmente muda o rótulo e o encaminha pela porta de saída apropriada. Um procedimento análogo acontece em todos os OF-LSR do caminho até chegar ao OF-LER de saída. Finalmente no OF-LER, o pacote é encaminhado à tabela de gerenciamento geral, mas neste caso, encontra coincidência na entrada de fluxo, que indica que o rótulo MPLS tem que ser removido e o pacote encaminhado pela porta física apropriada fora do domínio de transporte.

3.2.3 Lógica de gerenciamento para QoS

Quando um pacote que requere QoS entra na rede, esta lógica cria um caminho fim a fim dentro do domínio de transporte (caminho criado reativamente ao fluxo QoS entrante) que permite satisfazer os requisitos de QoS do fluxo específico. Como foi previamente mencionado, para diferenciar quando um pacote entra na rede, devem ser identificadas suas fronteiras, e com isso, os equipamentos da borda OF-LER (os que têm interfaces externas e internas ao domínio de transporte). Desta forma, se um pacote entra ao domínio de transporte utilizando uma interface externa o OF-LER de entrada. Entretanto, se o pacote utiliza uma interface interna de um OF-LER para entrar neste equipamento, trata-se de um OF-LER de saída, ou é um OF-LER que está sendo utilizado como OF-LSR para seu encaminhamento dentro da rede.

Em termos gerais, a lógica de gerenciamento de QoS funciona da forma descrita a seguir. Quando um pacote com requisitos de QoS chega a um OF-LER de entrada, a tabela 0 envia o pacote à lógica de gerenciamento de QoS (tabela de fluxos QoS), que envia o pacote ao Controlador e simultáneamente envia o pacote à tabela de fluxos geral para o seu encaminhamento imediato. O Controlador calcula o caminho ótimo para satisfazer os requisitos do fluxo QoS, e configura as respectivas entradas de fluxo nas tabelas de fluxos QoS de todos os equipamentos intervenientes no caminho ótimo calculado (os sucessivos pacotes do fluxo QoS, agora utilizam o caminho ótimo

criado). Para compreender em detalhe como funciona esta lógica, a seguir são descritas as diferentes tipos de entradas de fluxo que compõem a tabela de fluxos QoS:

- Entradas de fluxo QoS gerais (com interface de entrada externa): São pro-ativamente criadas pelo Controlador e colocadas na tabela de fluxos QoS (da lógica de gerenciamento QoS) nos OF-LER. Estas entradas de fluxo são construidas para encontrar coincidências com pacotes com requisitos de QoS se estes utilizam uma interface externa (é um OF-LER de entrada ao domínio de transporte). Esta entrada de fluxo QoS geral executa duas tarefas simultâneamente: encaminha o pacote QoS à tabela de fluxos gerais para que esta encaminhe imediatamente o pacote; e envia uma mensagem *Packet-in* ao Controlador para que configure um caminho ótimo. Estas entradas de fluxo são utilizadas enquanto o pacote ainda não tenha uma entrada de fluxo QoS específica configurada na tabela de fluxos QoS. É importante ressaltar que os OF-LSR não requerem de este tipo de entradas de fluxo.

- Entrada de fluxo QoS *default*: é proativamente construída e colocada na tabela de fluxos QoS (da lógica de gerenciamento QoS) nos OF-LER e nos OF-LSR. Esta é a entrada de fluxo mais geral, e construida para encontrar coincidência com tráfego QoS, ainda sem uma entrada de fluxo QoS específica configurada (adicionalmente é utilizada caso não coincida com uma entrada de fluxo QoS geral). A entrada de fluxo QoS *default* é utilizada para encaminhar o tráfego QoS à tabela de fluxos gerais para o encaminhamento imediato do tráfego.

- Entradas de fluxo QoS específicas: Estas entradas de fluxo são criadas (com prioridade) para encontrar coincidência com fluxos QoS específicos e colocadas nas tabelas de fluxo QoS de todos os equipamentos envolvidos no caminho com QoS ótimo. Inicialmente, a tabela de fluxos QoS não tem entradas de fluxo QoS específicas configuradas, sendo estas criadas reativamente pelo Controlador quando um novo pacote com requisitos de QoS chega ao domínio de transporte.

Quando um novo pacote QoS entra na rede por um OF-LER, este é encaminhado à tabela de fluxos QoS e inicialmente processado por uma regra QoS geral com interface de entrada externa. Esta regra encaminha o pacote por duas linhas de processamento simultâneamente (ver Figuras 3.2 e 3.3, detalhadas a seguir:

1. Primeiramente, a entrada de fluxo QoS com interface de entrada externa na tabela QoS envia um pedido de criação de novo fluxo ao Controlador (*packet-in*) contendo uma cópia total ou parcial do pacote. Para isso, o cliente *OpenFlow* tem a instrução de execução imediata do tipo ***Apply-actions*** e dentro desta uma ação do tipo ***output*** à porta reservada ***Controller*** [2]. Com esta informação, o Controlador calculará o caminho ótimo que permita satisfazer os requisitos QoS do fluxo fim a fim dentro do domínio de transporte. Para isto, o Controlador pode utilizar algoritmos de *Dijkstra* com restrições, similares às utilizadas por ISIS-TE ou OSPF-TE, ou podem ser criados protocolos de encaminhamento novos específicos para cada tipo de QoS (por exemplo, tráfego em tempo real sem perdas ou tráfego em tempo real

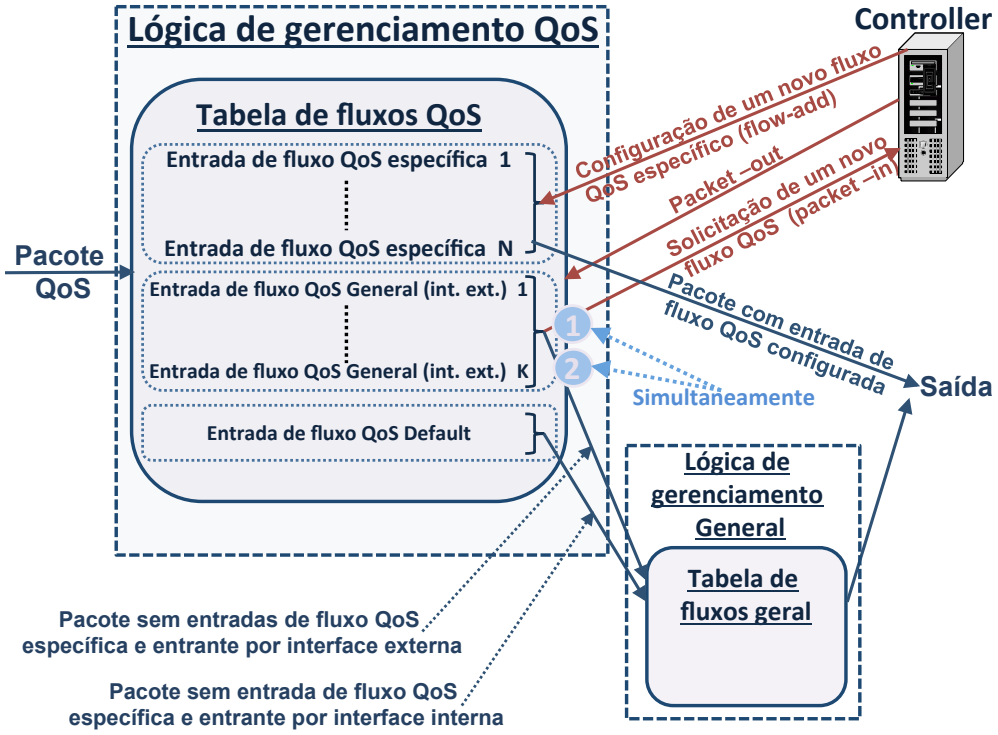

Figura 3.3: Lógica de gerenciamento QoS no OF-LER

com possibilidade de perda [4]). Com o caminho ótimo calculado, o Controlador configura uma entrada de fluxo QoS específica para este novo fluxo na tabela de QoS de cada um dos equipamentos intervenientes no caminho ótimo obtido (fluxo construido no sentido oposto ao encaminhamento). É importante mencionar que todas as entradas de fluxo QoS específicas são criadas com prioridade (para que sejam processadas antes das entradas de fluxo QoS gerais e da entrada de fluxo QoS *default*) e com um *timeout* apropriado (para que sejam apagados automáticamente depois de um período de inatividade). Adicionalmente, *OpenFlow* permite que as entradas de fluxo QoS específicas sejam ligadas a uma política de enfileiramento (com a ação *set-queue*) e este tráfego pode ser encaminhado pela interface de saída como prioritário. Finalmente o Controlador envia uma mensagem *packet-out* sem ação ao OF-LER, indicando que o pacote que gerou a consulta tem que ser apagado (o pacote já foi encaminhado pela segunda linha de procesamento, descrita no ponto 2 a seguir).

2. Simultâneamente às ações realizadas no ponto anterior, o pacote continua o processamento estabelecido pela entrada de fluxo QoS geral com portas de entrada externas (no OF-LER). Esta entrada de fluxo tem definida a instrução do tipo ***Goto-Table*** [2], que indica que o pacote tem que continuar seu processamento na tabela de fluxo geral (dentro da lógica de gerenciamento geral ver Figuras 3.2 e 3.3). Desta forma, o pacote QoS é processado e encaminhado imediatamente como um fluxo sem requisitos de qualidade de serviço, não tendo que aguardar a resposta do Controlador.

Quando o pacote chega a um OF-LSR (durante o período de tempo que o Controlador ainda não estabeleceu as entradas de fluxo QoS específicas nos equipamentos envolvidos no caminho ótimo),

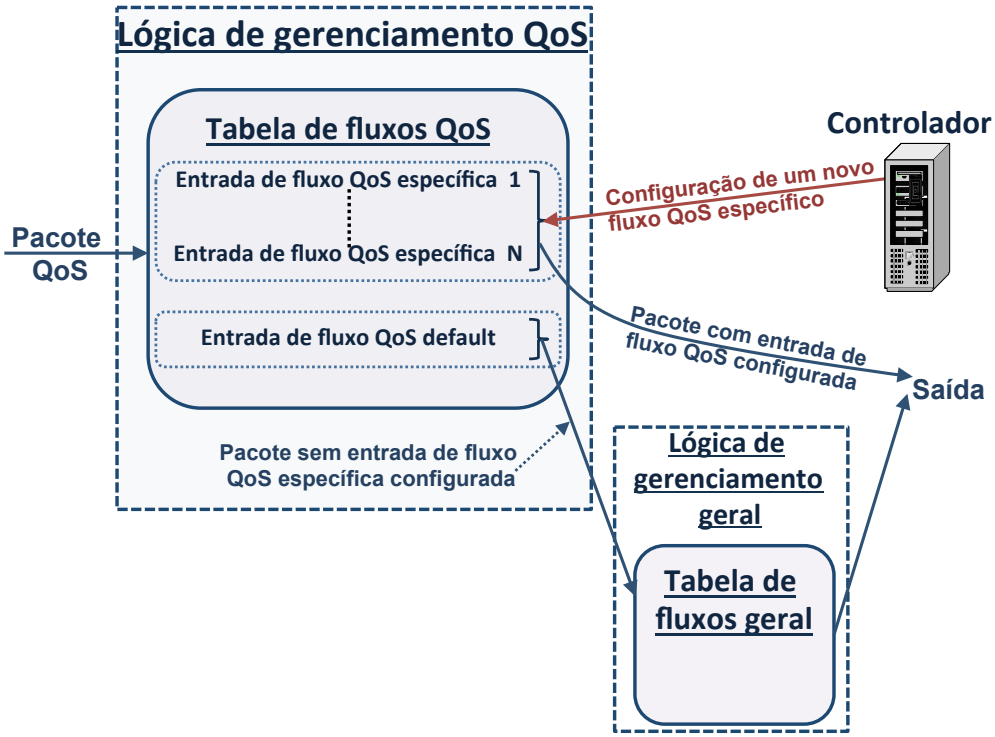

Figura 3.4: Lógica de gerenciamento QoS no OF-LSR

o pacote encontra coincidência com a entrada de fluxo QoS *default* (ver Figura 3.4). Esta entrada de fluxo indica ao pacote que tem que ser encaminhado à tabela de gerenciamento geral (instrução do tipo *Goto-Table*) para que este seja encaminhado imediatamente utilizando as entradas de fluxo gerais sem requisitos de QoS (os OF-LSR não enviam o *Packet-in* ao Controlador). Quando finalmente o pacote chegar ao OF-LER de saída, este também encontra coincidência com a entrada de fluxo QoS *default* (neste caso não se trata de OF-LER de entrada) que encaminha o pacote à tabela de fluxo geral. Finalmente esta tabela encaminha o pacote sem requisitos de QoS fora do domínio de transporte do SP.

É importante ressaltar que os OF-LSR e o OF-LER de saída não requerem do envio de uma mensagem *Packet-in* ao Controlador, devido a que o OF-LER de entrada já tem enviado a mensagem *Packet-in* ao Controlador indicando a chegada de um novo fluxo com requisitos de QoS. Além disso, quando o Controlador calcula e configura o caminho QoS específico, este é criado em todos os equipamentos envolvidos no caminho (incluindo os OF-LSR e o OF-LER de saída). Esta metodologia é muito importante, devido a que impede que mensagens *Packet-in* redundantes cheguem ao Controlador. Adicionalmente como só o primeiro pacote de um novo fluxo com requisitos de QoS é enviado ao Controlador (ver Figuras 3.5 e 3.6), a carga por processamento no Controlador é significativamente menor comparado com uma arquitetura *OpenFlow* padrão (os fluxos sem requisitos de QoS não são enviados ao Controlador na arquitetura proposta).

Por fim, quando o Controlador configura as entradas de fluxo QoS específicas (na tabelas QoS) em todos os equipamentos envolvidos no caminho QoS ótimo, os seguintes pacotes do mesmo fluxo QoS são encaminhados utilizando as entradas de fluxo QoS específicas. Adicionalmente para

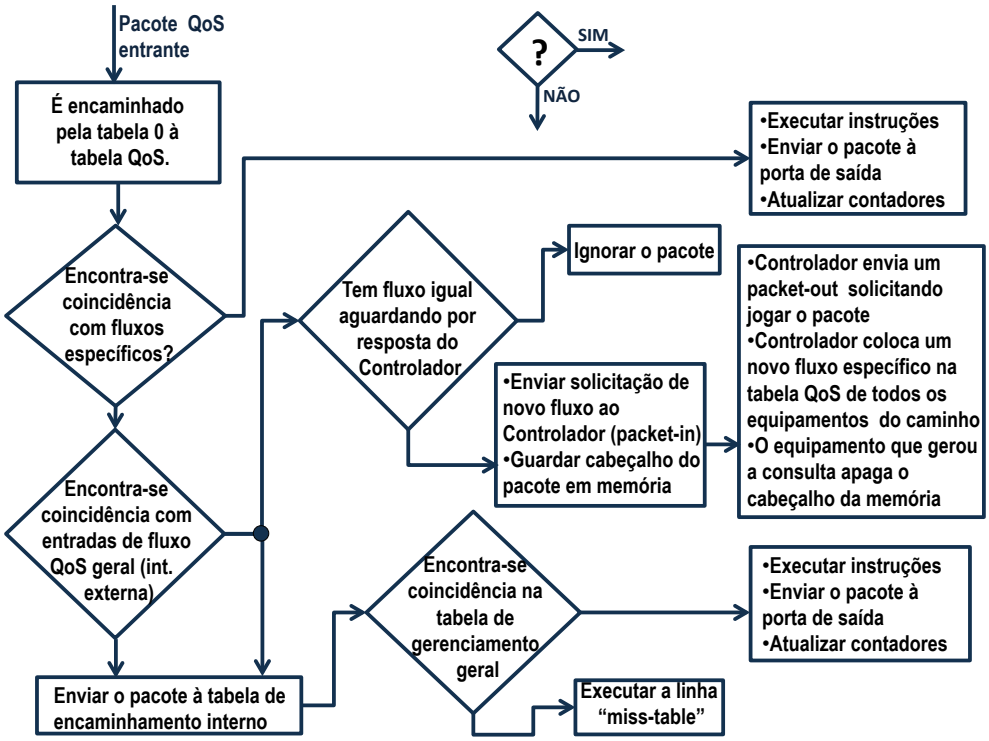

Figura 3.5: Diagrama para a lógica de gerenciamento QoS no OF-LER

diminuir o possível *jitter*, é criado um caminho QoS específico único sem balanceamento de carga entre duas alternativas. Se a característica de serviço o requere, pode ser criado também um caminho alternativo de *backup*, que pode ser utilizado em caso de que o caminho ótimo apresente inconvenientes.

3.2.3.1 Considerações importantes da arquitetura

Referente à construção dos fluxos com requisitos de QoS específicos, pode ser realizada uma otimização importante. Quando o Controlador calcula o melhor caminho específico para um fluxo com requisito de QoS, este pode examinar se já existe um caminho com os mesmos requisitos de QoS que esteja utilizando o mesmo trajeto dentro da rede de transporte (os mesmos OF-LER e os mesmos OF-LSR). Neste caso, pode ser utilizado o mesmo LSP (para mais informação referente a LSP, ver seção 2.5) do caminho previamente construido, adaptando se fosse requerido, a largura de banda garantida para este LSP fim a fim dentro do domínio de transporte. Se este é o caso, o Controlador coloca uma nova entrada de fluxo QoS específica no OF-LER de entrada, que indica que para os pacotes coincidentes deve ser adicionado o mesmo rótulo MPLS, e enviado pela mesma porta física que o fluxo previamente configurado. No equipamento de borda de saída, o Controlador também deve adicionar uma nova entrada de fluxo que retira o rótulo MPLS e que encaminha o pacote pelo destino certo, mas nos OF-LSR não é requerida a adição de uma nova entrada de fluxo, pode ser utilizada a mesma entrada de fluxo do caminho prévio (só no caso em que fosse requerido, o Controlador deve adaptar as garantias de QoS para as entradas de fluxo QoS específicas nos OF-LSP intermédios para que sejam incorporados e garantidos os requisitos QoS do novo fluxo).

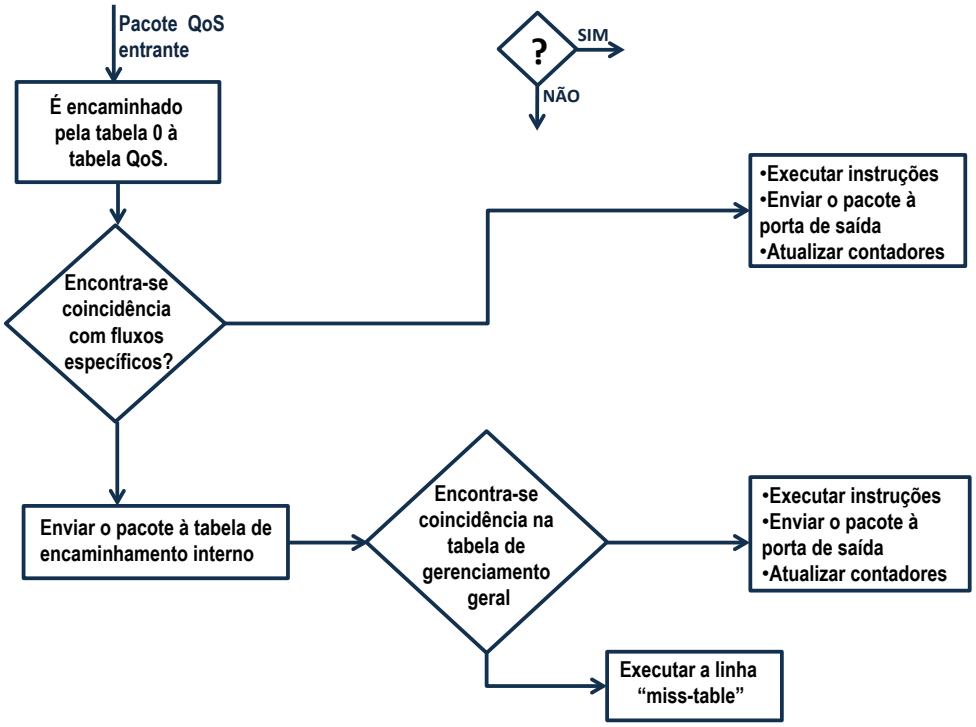

Figura 3.6: Diagrama para a lógica de gerenciamento QoS no OF-LSR

Com esta otimização é possível diminuir sensívelmente o número de entradas de fluxo requeridas nos LSR, devido a que estes equipamentos só tem que possuir uma entrada de fluxo (na tabela de fluxo QoS) para cada combinação da terna: tipo de tráfego, OF-LER origem e OF-LER destino. É importante notar que os fluxos com requisitos de QoS transportados por os mesmos LSP internos, têm o mesmo tratamento de QoS e não devem ser priorizados uns com respeito aos outros.

Adicionalmente se ressalta que na arquitetura proposta, o Controlador não adiciona tempos de espera para os primeiros pacotes de cada fluxo. Esta afirmação se baseia no fato de que os pacotes do mesmo fluxo sempre são encaminhados inicialmente pelas entradas de fluxo gerais e paralelamente processados pelo Controlador. Posteriormente, quando a tabela de fluxos QoS tiver a entrada de fluxo QoS específica preenchida, os seguintes pacotes do mesmo fluxo coincidirão com esta nova entrada e encaminharão o fluxo pelo caminho ótimo.

Se destaca também como uma das características mais importantes desta lógica, a significativa redução da dependência da rede do Controlador. Na arquitetura proposta, quando acontece uma queda do Controlador ou este apresenta tempos de resposta elevados, os novos fluxos QoS sempre continuam sendo encaminhados pela lógica de gerenciamento geral. Este fato garante um aspecto de robustez indispensável para a rede de transporte de um SP.

Além disso, para qualquer implementação de *OpenFlow* [56] é importante a existência de mecanismos nos nós de rede que evitem que pacotes sucessivos do mesmo fluxo gerem as mesmas consultas ao Controlador enquanto se aguarda a resposta deste, evitando-se sobrecargas desnecessárias. Na arquitetura *OpenFlow* padrão, a implementação disto é complexa, dado que ela requere que todos os sucessivos pacotes do mesmo fluxo (no período de tempo que se aguarda pela resposta

do Controlador) sejam armazenados na memória dos *switches*; quando o *packet-out* do primeiro pacote chegar, tem que ser executada a ação que o *packet-out* indique em todos os pacotes que aguardam em memória. Na arquitetura proposta, a lógica para evitar que sucessivos pacotes do mesmo fluxo gerem as mesmas consultas ao Controlador resulta mais fácil de implementar, dado que não há a necessidade de manter nenhuma informação dos sucessivos pacotes que chegam ao nó durante o periodo de tempo no qual se aguarda resposta do Controlador. Isto se deve a que quando o *packet-out* chega, os mencionados sucessivos pacotes já teriam sido encaminhados pela tabela de fluxos geral. Na arquitetura proposta, só deve ser mantida uma lista com a informaḍão dos cabeçalho do primeiro pacote de cada um dos fluxos que aguarda resposta do Controlador e deve ser comparado o cabeçalho do pacote entrante, com os cabeçalhos dos pacotes armazenados. Desta forma, é possível conhecer se este é o primeiro pacote de um fluxo QoS ou se trata-se de um pacote fluxo que aguarda resposta (ver Figura 3.5).

O mecanismo que implementa a lógica para evitar o encaminhamento de mensagens p*acket-in* duplicadas é denominado em este trabalho como PIDAM, e é implementado modificando o procedimento de envío *packet-in* nos *OpenVSwitches* [55]. Nos testes desenvolvidos na seção 5.1.7 podem ser observados os resultados adicionais da sua incorporação na arquitetura proposta.

Menciona-se também que, segundo o resultado obtido em [57], na caracterização do tráfego para dispositivos móveis, a probabilidade de encontrar um pacote pertencente a um novo fluxo em um cliente *OpenFlow* pode atingir 4%. Se este resultado é projetado ao tráfego em geral, isto implica que para o caso de *OpenFlow* padrão, 4% do tráfego total pode ser encaminhado ao Controlador. No caso de uma rede de transporte do um SP, este número elevado de consultas representa um desafio muito grande para um Controlador, o que adicionalmente requere do aumento dos recursos da rede para dar suporte ao incremento de largura de banda. Nesta arquitetura a carga do Controlador diminui considerávelmente em relação a uma arquitetura *OpenFlow* padrão pelas considerações descritas a seguir:

- o tráfego sem QoS não é encaminhado ao Controlador; ele é encaminhado pela tabela de fluxos gerais proativamente configurada;
- só o primeiro pacote de cada fluxo QoS é enviado ao Controlador, reduzindo ainda mais a carga;
- quando chega um novo pacote QoS, só os OF-LER de entrada ao domínio de transporte geram uma consulta ao Controlador.

3.2.4 Lógica para o gerenciamento de serviços específicos

Esta lógica é construida para oferecer serviços específicos a clientes da rede de transporte. Lembra-se que são clientes da rede de transporte os domínios fixos, os domínios móveis, os Data-Center e os clientes de grande porte diretamente conectados à rede de transporte. Diferentemente às técnicas anteriormente descritas, esta alternativa é completamente predefinida em forma de serviço acordado entre o cliente e a rede de transporte. É um caso particular de estes tipos de serviços

a realização de uma VPN entre clientes de grande porte, ou a interconexão entre dois domínios utilizando caminhos com garantias de QoS específicas.

Esta lógica é configurada só nos equipamentos intervenientes no serviço requerido, e não são lógicas de aplicação genérica em todos os equipamentos da rede. Concretamente, se é requerida uma interconectividade específica entre dois clientes de grande porte, só são configurados os OF-LER que têm interface com os clientes que contrataram o serviço e os OF-LSR que proporcionam o caminho requerido.

Na configuração de estes serviços pode ser definida uma grande variedade de critérios que podem ser utilizados para diferenciar o tráfego que requere do serviço contratado. Exemplo, podem ser configuradas regras para que todo tráfego proveniente do cliente seja encaminhado pelo serviço de transporte contratado, ou se pode ser mais específico definindo um grupo de endereços IP origem que podem utilizar o serviço, ou pode existir um acordo prévio entre as partes que estipule qual rótulo MPLS tem que ser colocado pelo cliente para utilizar tal ou qual serviço.

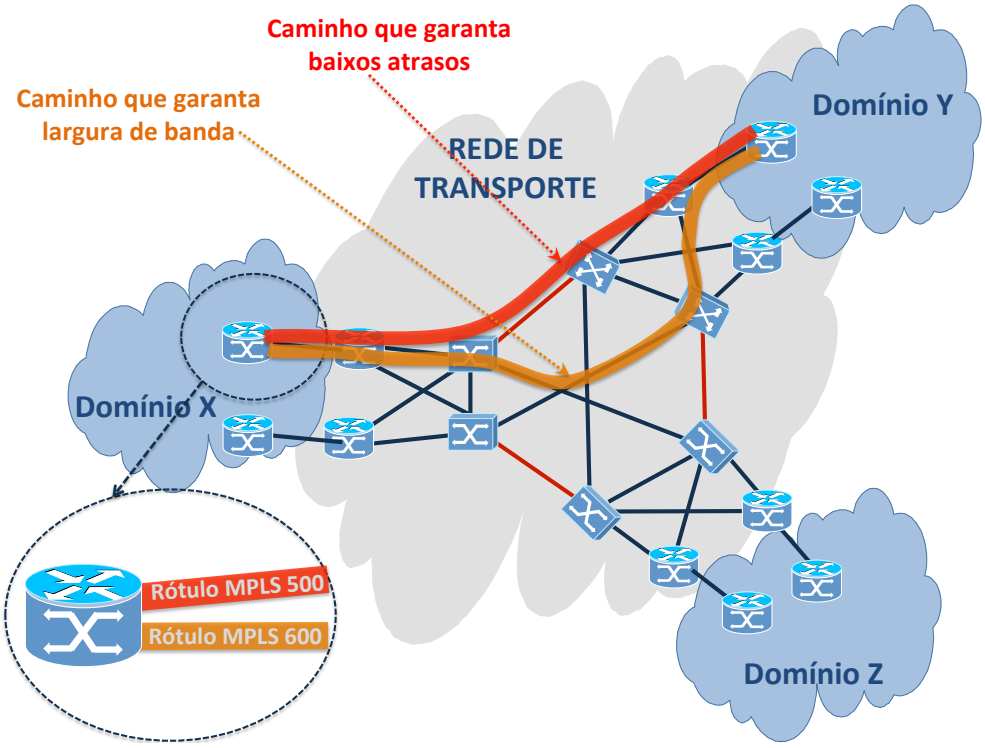

Figura 3.7: Exemplo de aplicação de lógica para gerenciamento de serviços específicos

Na Figura 3.7 é ilustrado o caso particular onde a rede de transporte proporciona serviços diferenciados entre os domínio X e Y. Neste caso existe um acordo prévio entre os clientes e a rede de transporte, o qual estabelece que o domínio X envia pacotes com rótulo MPLS 500, o tráfego é transportado por um caminho que garante reduzidos atrasos, mas se é colocado o rótulo MPLS 600, o tráfego é encaminhado até o domínio Y por um caminho que garante largura de banda. Para implementar este serviço, o Controlador tem que adicionar no OF-LER conectado ao domínio X, as entradas de fluxo na tabela 0 que permitam encaminhar o tráfego pelo caminho acordado. Adicionalmente tem que ser configurados todos os equipamentos intervenientes no caminho até o

destino. Reciprocamente deve ser aplicada a mesma lógica para o tráfego na outra direção.

Esses serviços são conceitualmente semelhantes aos serviços Lan to Lan oferecidos hoje. A diferença é que, nesse caso, os caminhos são identificados por tags MPLS e em os serviços Lan to Lan são identificados pela identificação de VLAN.

É importante notar que no exemplo previamente mencionado, é o cliente (com a adição do rótulo MPLS) quem escolhe o serviço apropriado. A rede de transporte só deve cumprir com o acordo preestabelecido. Para uma maior compreensão das alternativas de serviços que podem ser criados com esta lógica, se recomenda a leitura da seção 4.5, que contém um exemplo de aplicação de esta interconetividade entre domínios móveis do SP.

3.3 Arquitetura de transporte com equipamentos híbridos

3.3.1 Conceito de equipamento híbrido

Um elemento de rede *OpenFlow* híbrido é um equipamento de rede que incorpora as funcionalidades de *OpenFlow* e ao mesmo tempo mantém as capacidades do plano de controle distribuído padrão, conforme ilustrado na Figura 3.8. Estes equipamentos possuem dois tipos de processamento para o envio de pacotes: o padrão que utiliza uma tabela de encaminhamento construida com a informação obtida do plano de controle distribuído clássico, isto é, informação obtida por protocolos como OSPF, ISIS, LDP, RSVP e BGP implementados em cada equipamento de rede, e um segundo tipo de processamento construído e gerenciado pelo Controlador, composta por um grupo de tabelas *OpenFlow* interligadas mediante a instrução ***Goto-Table***.

Atualmente existem várias opções de implementação de *switches* híbridos [58] [59], algumas delas já se encontram implementadas em equipamentos de rede. Estas opções tentam proporcionar alternativas de processamento que permitam distinguir quando um pacote de entrada tem que ser processado pelo controle distribuído ou pelo controle *OpenFlow*. Entre as opções de implementação de *switches* híbridos, encontram-se as listadas a seguir:

- Que cada porta do equipamento de rede seja ligada a um único tipo de controle (controle *OpenFlow* ou controle distribuído atual).
- Que cada VLAN seja ligada a um tipo único de controle, melhorando a utilização das portas disponíveis no equipamento.
- Que exista uma ACL (*Access List*) inicial que permita distinguir qual pacote tem que ser processado por um tipo de controle e qual pelo outro tipo de controle.
- Que o controle *OpenFlow* seja o primeiro caminho de procura, mas se não existir coincidência com uma linha de *OpenFlow*, continuar o processamento pelo controle distribuído atual.
- Utilizar a funcionalidade ainda não implementada, mas já considerada na especificação de *OpenFlow* 1.5 (porta reservada ***Normal***) que permite que qualquer entrada de fluxo dentro

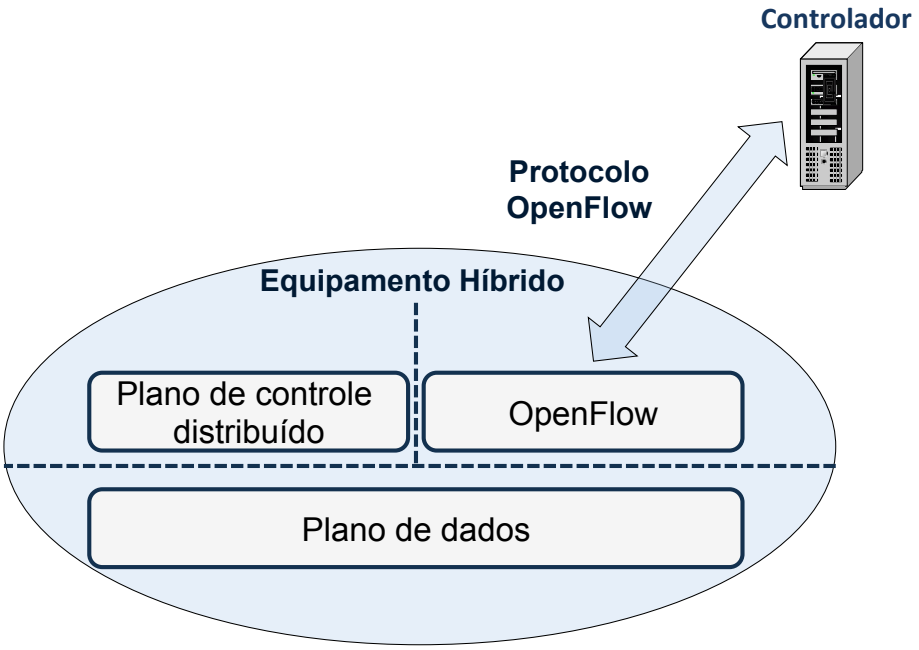

Figura 3.8: Equipamento *OpenFlow* híbrido

das tabelas que compõem a linha de processamento *OpenFlow* encaminhe o pacote ao plano de controle distribuido padrão. Para esta finalidade, o *OpenFlow* permite definir entradas de fluxo com instruções do tipo ***Apply-actions*** ou ***Write-actions*** e com a ação ***output*** à porta reservada ***Normal*** [2]. Desta forma,o pacote coincidente nesta regra é encaminhado internamente ao controle distribuído clássico.

Pela potencialidade que a última das alternativas proporciona e pelo fato de que na arquitetura proposta os equipamentos de rede não são somente elementos de camada de enlace de dados, a última alternativa foi a escolhida para a proposta de arquitetura para o domínio de transporte com equipamentos híbridos. Nesta proposta, todos os pacotes são inicialmente processados pelo plano de controle SDN *OpenFlow* e é este quem decide se encaminhar os pacotes ao controle distribuído clássico (utilizando a funcionalidade previamente descrita) ou encaminha os pacotes a outra tabela *OpenFlow*.

Prévio à descrição da arquitetura híbrida, se resume que Como sumario, os equipamentos *OpenFlow* que implementam a versão 1.5 (e algumas versões anteriores) podem configurar em qualquer uma de suas entradas de fluxo: que um pacote seja enviado ao Controlador, que um pacote seja enviado a outra tabela, que o pacote seja encaminhado por uma porta fisica específica, que seja enviado ao controle distribuido padrão, que seja encaminhado mediante uma ação de grupo, ou realizar várias destas alternativas simultâneamente (combinando instruções do tipo ***Apply-actions***, ***Write-actions*** e ***Goto-Table***.

3.3.2 Proposta de arquitetura para domínio de transporte do SP com equipamentos híbridos

A arquitetura hibrida proposta combina equipamentos híbridos com a proposta de arquitetura da seção 3.2. Desta forma, é possível desenvolver uma arquitetura alternativa que programe utilizando *OpenFlow* a Tabela 0 e a lógica de gerenciamento QoS. Por outra parte, esta arquitetura implementa a lógica de gerenciamento geral com o controle distribuido clássico (ver Figura 3.9). As lógicas restantes de gerenciamento poderão ser implementadas opcionalmente pelo controle centralizado *OpenFlow* ou pelo controle distribuído padrão.

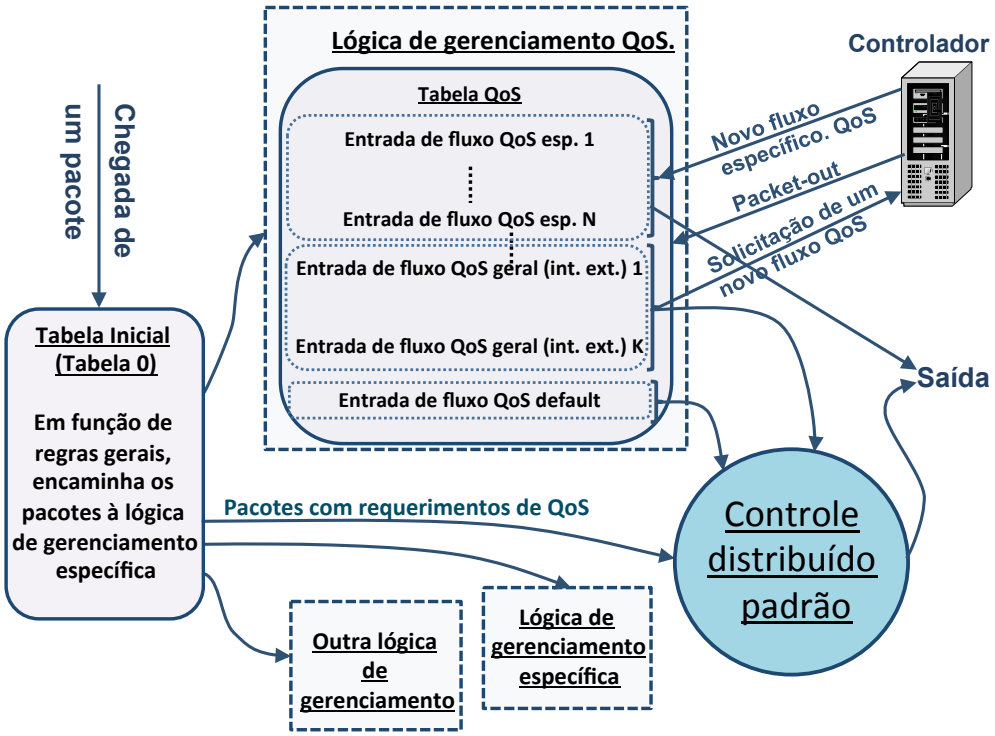

Figura 3.9: Arquitetura de controle com equipamentos híbridos

Novamente nesta arquitetura, o controle implementado nos OF-LER deve ser diferenciado do controle implementado nos OF-LSR. Esta diferenciação se deve a que só os OF-LER híbridos de entrada enviam a solicitação de criação de novo fluxo QoS ao Controlador.

Com a arquitetura previamente mencionada, o tráfego com requisitos de QoS tem o comportamento descrito a seguir e ilustrado na Figura 3.10. Quando um pacote com requisitos de QoS pertencente a um novo fluxo chega a um OF-LER híbrido por interfaces de entrada externas (pacote entrante ao domínio de transporte do SP), este é inicialmente processado pela tabela 0 e é encaminhado à tabela de fluxos QoS. Como ainda não existe uma entrada de fluxo QoS específica configurada para o novo fluxo, o pacote é encaminhado por uma entrada de fluxo QoS geral que encaminha o pacote simultâneamente ao Controlador e neste caso ao controle distribuído padrão. Desta forma, enquanto não exista a entrada de fluxo QoS específica na tabela de fluxos QoS, os sucessivos pacotes do novo fluxo QoS serão encaminhados pelo controle distribuído padrão. Quando a entrada de fluxo QoS específica estiver disponível, os sucessivos pacotes serão encaminhados por

esta entrada.

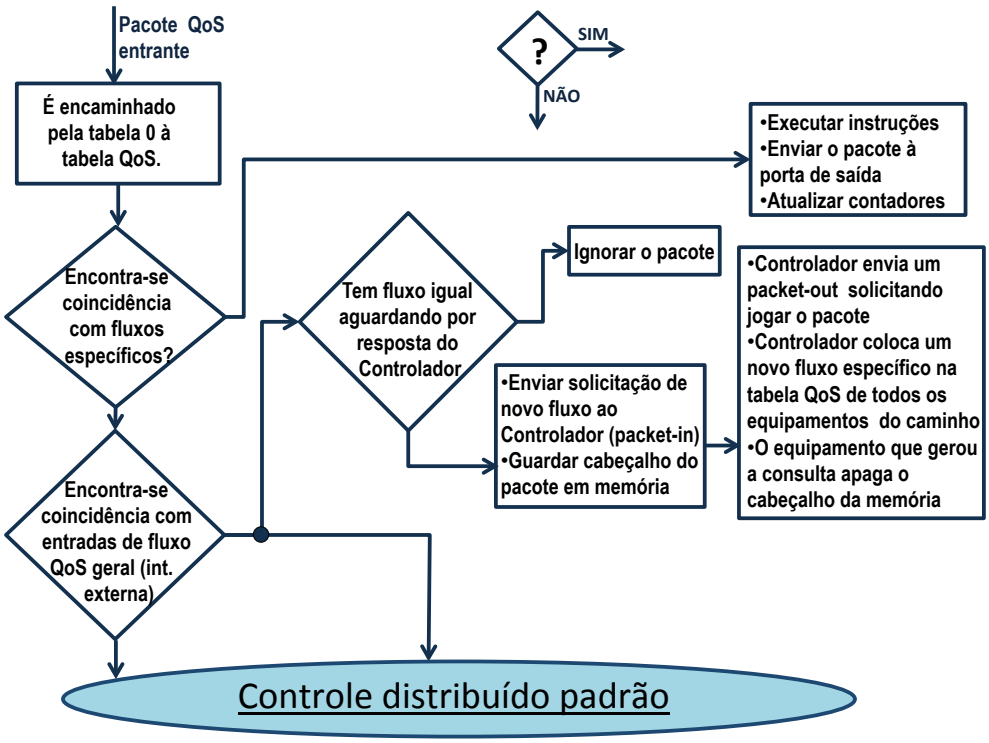

Figura 3.10: Diagrama para a lógica de gerenciamento QoS em equipamentos híbridos OF-LSR

3.3.3 Considerações adicionais

Esta arquitetura é considerada na comparativa de arquiteturas apresentada na seção 5.1.8 e poderia ter menor resistência a ser utilizada em um SP. Esta arquitetura incrementa as possibilidades de engenharia de tráfego por meio da incorporação de *OpenFlow*, mantendo ao mesmo tempo a robustez das redes atuais. Na arquitetura híbrida, no caso de queda do Controlador, os novos pacotes continuam sendo encaminhados mediante o controle distribuído padrão (esta topologia é robusta ainda no caso de mudança topológica durante a queda do Controlador).

Finalmente menciona-se que esta arquitetura pode ser de muita utilidade em um processo de migração que tenha como objetivo mudar a rede de transporte de um SP implementada com o controle distribuido padrão, para uma rede com o controle centralizado *OpenFlow* como a desenvolvida na seção 3.2. Neste caso, se ressalta que as diferentes lógicas de gerenciamento podem ser migradas independentemente uma das outras e por etapas, proporcionando um processo de migração confiável e controlado.

Capítulo 4

Proposta de arquitetura para o domínio móvel

4.1 Introdução

Nas últimas décadas, o crescimento no número de dispositivos móveis tem incrementado de forma exponencial e por tanto o seu tráfego. Por outra parte, a diversidade de aplicativos móveis tais como VoiP (Voice over IP), navegação web, realidade virtual, TV móvel, jogos online, etc., têm expandido a diversidade de requisitos de QoS do tráfego, o que tem derivado em que os operadores devam melhorar seus serviços. A geração 5G de redes móveis está emergindo com o propósito de satisfazer tais demandas, necessitando cumprir com as mais elevadas características de tráfego como mínima latência, comunicação ultra-confiável, elevadas taxas de transmissões de dados e alta experiência na mobilidade dos usuários [60].

A tecnologia celular 4G existente desenvolvida por 3GPP (*Third Generation Partnership Project*) [61, 62] tem melhorado significativamente as características das gerações celulares prévias. Porém, a tecnologia 4G ainda possui uma série de limitações. As entidades lógica da arquitetura móvel 4G são baseadas na configuração de *hardware* propietário e implantado de uma forma estática e ineficiente em termos de custo e ciclo de inovação da rede. O controle e o plano de dados estão rebocados no SGW (*Serving Gateway*) e no PGW (*Packet Data Network Gateway*), o que incrementa a complexidade do gerenciamento da rede e limita a escalabilidade da solução. Adicionalmente, o plano de dados da rede móvil 4G é centralizado, de forma que o tráfego até e desde os UE (*User Equipment*) sempre deve ser enviado através do PGW, o que pode ser um caminho ineficiente quando os UEs envolvidos na comunicação estão próximos uns com os outros.

Para mitigar as fraquezas da arquitetura 4G e para alcançar os objetivos da tecnologia celular 5G, este trabalho como outros prévios [63, 64, 65] implementa NFV na arquitetura de rede móvel. A tecnologia NFV [66, 67] permite a virtualização das principais entidades de arquitetura móvel, o que é implementado como uma camada de software colocada sobre *hardware* standard menos oneroso ou sendo executada sobre um ambiente de computação em nuvem. A NFV oferece uma alternativa flexível que pode se adaptar às diferentes variações na demanda e fornecer crescimentos

nos aplicativos de forma distribuída, o que incrementa a robustez das arquiteturas de redes móveis.

Outra alternativa ampliamente sugerida e também aplicada no presente trabalho é a incorporação de conceitos de SDN [68, 69, 35]. A arquitetura SDN movimenta o plano de controle das entidades móveis para dispositivos centrais chamados Controladores, os quais têm uma visão global da rede e são responsáveis do gerenciamento do plano de dados das redes móveis. A tecnologia SDN oferece uma interface standard aberta para a comunicação entre o Controlador e o dispositivo de plano de dados, incrementando a flexibilidade e a programabilidade da rede, simplificando os elementos do plano de dados [70]. A pesar do mencionado anteriormente, são desvantagens do controle de rede centralizado, a excesiva dependência de cada nó respeito do Controlador, a grande quantidade de informação que o Controlador deve processar e o atraso adicional na criação do fluxo QoS específico, devido a que o pacote deve aguardar a resposta do controlador antes de ser encaminhado [43].

Utilizando os conceitos de SDN e NFV, este trabalho propõe uma nova forma de abordar a detecção de tráfego QoS e seu gerenciamento [18], dois dos requerimentos principais da próxima geração de redes 5G [71]. Se o tráfego com requerimentos de QoS é detectado de forma mais rápida e eficiente, este poderá ser processado rápidamente. Por tanto, com um tratamento de QoS especializado, pode ser criado um fluxo de QoS ótimo, fim a fim, que cumpra tais características. Os resultados mostram que a lógica proposta é flexível e programavel, aplicada muito próximo do lugar onde o tráfego é gerado (próximo dos UEs), e que herda as possibilidades de engenharia de tráfego das arquiteturas SDN.

Os resultados quantitativos mostram uma arquitetura SDN que reduz os problemas relacionados aos elevados tempos de resposta do Controlador, e que é capaz de enviar o tráfego QoS sobre demanda de forma imediata. Adicionalmente, são implementados dois níveis para a detecção de tráfego QoS, um na fronteira de rede móvel nos eNB (*Evolved NodeB*) ou gNB (*Next Generation NodeB*) e a outra em elementos especializados para esse fim, o que gera uma ampla flexibilidade e escalabilidade à solução.

Adicionalmente serão propostas alternativas que permitam garantir caraterísticas de QoS para a comunicação fim a fim entre equipamentos que pertencem a diferentes domínios móveis de um mesmo SP, comunicados por um domínio de transporte. Isto é realizado mantendo a independência entre os diferentes domínios, minimizando no que for possível, as necessidades de interações entre eles. Finalmente são indicadas alternativas que incrementam a robustez da rede, mecanismos para a recuperação do tráfego ante quedas de nós ou enlaces, e procedimentos para o gerenciamento da mobilidade que permitam manter os serviços eficientemente durante o processo de *handover*.

Nas seguintes seções é detalhada a arquitetura móvel proposta, começãndo com uma descrição geral dos elementos constitutivos, e as carateristicas gerais de arqutitetura. Seguidamente são aprofundados os mecanismos de detecção de serviços que têm requerimentos de QoS, os procedimentos de criação de *bearers* com requisitos de QoS, e os procedimentos de *attachment*. Finalmente são desenvolvidos aspectos relacionados à mobilidade dentro e entre domínios móveis de um mesmo SP, e para garantir requerimentos de QoS fim a fim.

4.2 Considerações gerais da arquitetura e trabalhos relacionados

A próxima geração de redes móveis gerará aprofundadas modificações na arquitetura e ao mesmo tempo deverá manter a interoperabilidade com as gerações celulares anteriores. As redes 5G estão evoluindo de forma de incorporar a tecnología NFV, movimentando o controle na fronteira da rede móvel e incorporando as capacidades SDN para incrementar a separação entre os planos de controle e de dados [72].

Este artigo propõe uma arquitetura que aplica SDN para obter uma clara separação entre o plano de usuário e o de controle dos elementos da arquitetura móvel PGW e SGW [73, 74, 75, 76]. O plano de usuário do SGW e do PGW é implementado por *switches OpenFlow* simples [77] (chamados OF-*Switch* e OF-GW respectivamente) e o plano de controle de esses elementos é implementado de forma separada em elementos lógicos chamados PGW-C (*PDN Gateway Control*). O Controlador é responsável do gerenciamento do plano de dados de todos os OF-GW e os elementos físicos móveis como *routers* e *switches* (implementados como OF-Switches), utilizados para transportar os pacotes através do *core* e do *backhaul* móvel (ver Figura 4.1). Esta separação é implementada na proposta de rede 5G [60, 78, 79], onde o plano de usuário do PGW é implementado em um novo elemento denominado SMF (*Service Management Function*).

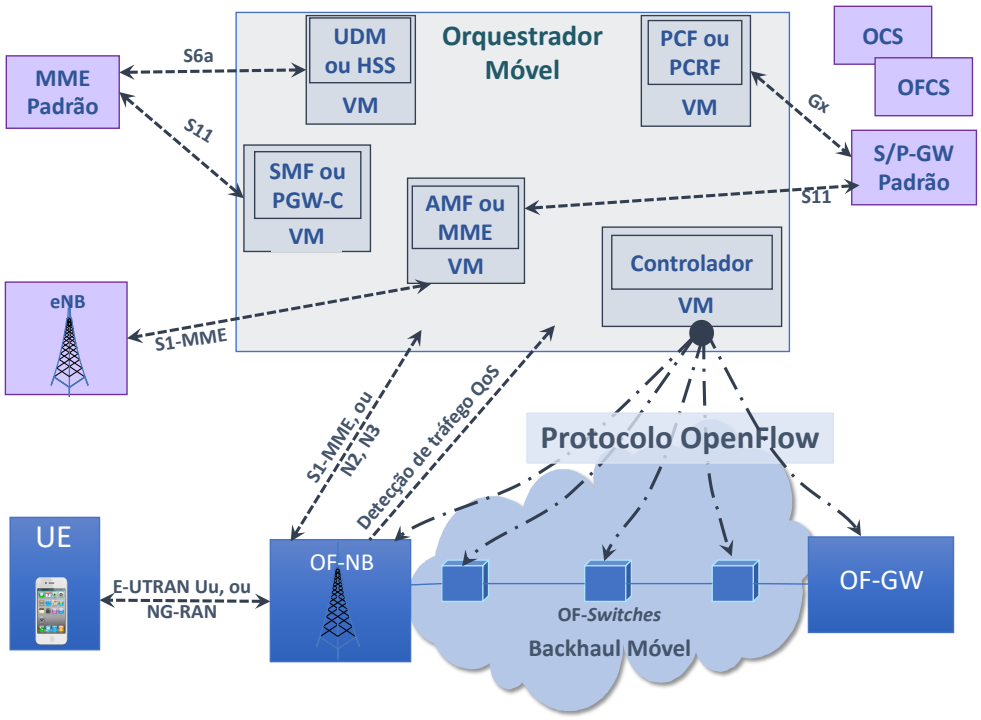

Figura 4.1: Arquitetura proposta indicando opcionalmente a nomenclatura das entidades 4G ou 5G.

Como é proposto em [80, 81, 82, 83] e na arquitetura 5G apresentada recientemente, este trabalho incorpora conceitos de NFV de forma de incrementar a escalabilidade e a adaptabilidade a demandas variáveis. A eleição aqui consiste na implementação do plano de controle dos elementos móveis da arquitetura 4G (PGW-C, MME - *Mobility Management Entity*, PCRF - *Policy and*

Charging Rules Function, HSS - *Home Subscriver Server*) ou da arquitetura 5G [60, 79] (SMF, PCF - *Policy Control Function*, AMF - *Access and Mobility Management Function*, e UDM âĂŞ *Unified Data Management*) e o Controlador como VNFs (*Virtual Network Function*) sobre um ambiente de nuvem. Por outra parte, os OF-Swiches, OF-GW e as estações base (4G eNB ou 5G gNB) são implementadas como elementos físicos distribuídos sobre o domínio móvel (ver Figura 4.1).

Os eNB da arquitetura 4G ou gNB da 5G são elementos chave na proposta do presente trabalho. Estes elementos mantêm as interfaces de acesso e de controle, mas ao mesmo tempo incorporam as funcionalidades de *OpenFlow* (estes elementos modificados são chamados OF-NB). Na proposta, o Controlador é responsável pela criação de todas as entradas de fluxo requeridas para ligar o tráfego das interfaces de rádio do OF-NB e as interfaces de *backhaul* de esse mesmo elemento. Na Figura 1 também se demonstra que os OF-NB podem opcionalmente implementar diferentes tipos de interfaces do acesso móvel [84, 85, 86] como 4G E-UTRAN (*Evolved UMTS Terrestrial Radio Access Network*) ou 5G NG-RAN (*Next Generation Radio Access Network*). Este trabalho está focado sobre a rede móvil de *core* e de *backhaul*, sendo capaz de se adaptar a diferentes tipos de redes de acesso.

A arquitetura móvel utiliza o MPLS para implementar o plano de dados. O protocolo MPLS proporciona uma excelente correspondência entre fluxo de tráfego e *bearers* com rótulos, melhorando o encaminhamento e incrementando as possibilidades de engenharia de tráfego [70]. Porém, a implementação do plano de dados utilizando MPLS não é obrigatória, de forma de que outras alternativas também possam ser utilizadas como 802.1Q ou 802.1AD [87, 88], não gerando mudanças significativas na arquitetura proposta. É importante ressaltar que o MPLS tem 20 bits para identificar rótulos enquanto o 802.1AD somente utiliza 12 bits para a identificação de VLAN (*Virtual Local Area Network*), por tanto, o MPLS incrementa o número de possíveis identificadores. Adicionalmente, o protocolo MPLS foi pensado para transportar tráfego de diferentes tecnologias de camada 2, o que representa uma característica útil em redes heterogêneas.

A implementação do plano de dados através de SDN gera mudanças significativas como o fato de que não é mais requerido o uso do protocolo GTP-U (GPRS *Tunneling Protocolo User plane*) para a mobilidade [73, 87, 88, 89]. A Figura 4.2 indica as mudanças no *stack* de protocolos da rede LTE móvil se é implementado o SDN junto com o MPLS. Como pode ser observado, o *radio-Bearer* entre o OF-NB e o UE se mantém sem mudanças, enquanto o S1-Bearer e o S5/S8-Bearer se combinam em um novo *Bearer* chamado OF-Bearer (criado e gerenciado pelo Controlador). O EPS-Bearer (na terminologia EPS) ou o QoS-Flow (na terminologia 5GS [60] é a concatenação entre o *Radio Bearer* e o OF-Bearer, responsável por fornecer a comunicação fim a fim sobre o domínio da rede móvel com as características de QoS apropriadas. Por simplificação, este trabalho denomina os termos EPS-Bearer e QoS-Flow como EPS-Bearer.

Em vez de adicionar um *header* GTP-U com o TEID (*Tunnel Endpoint Identifier*) apropriado para alcançar o UE destino e identificar o EPS-Bearer a ser utilizado, este trabalho adiciona dois rótulos MPLS para obter esses mesmos objetivos. O encaminhamento utilizando o EPS-Bearer se obtém através da inspeção dos dois rótulos MPLS, onde a interior identifica o UE/Radio Bearer

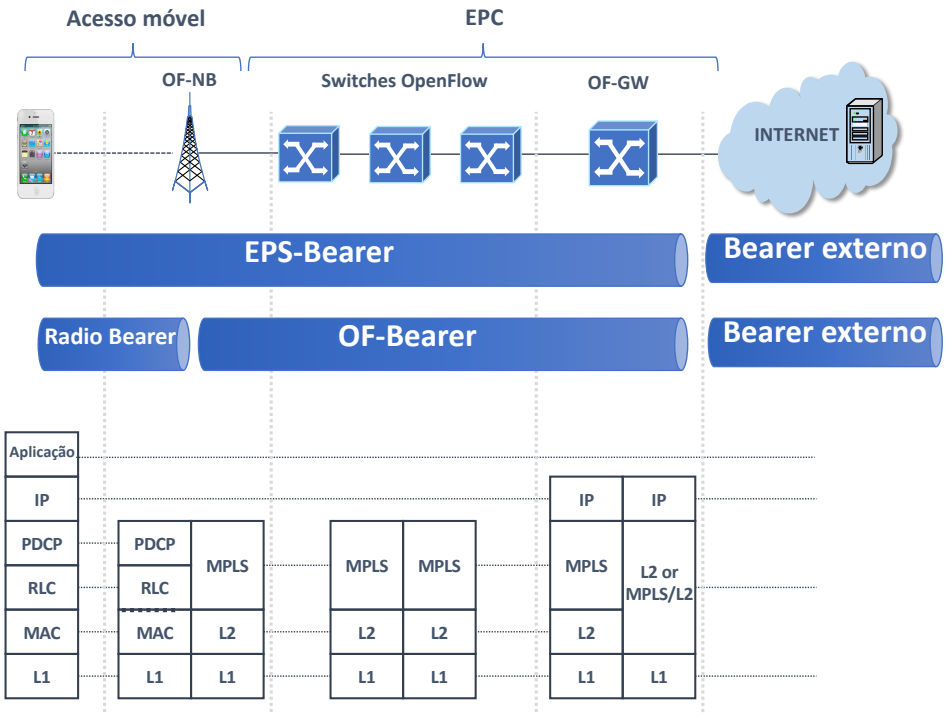

Figura 4.2: Relação entre *EPS-Bearer*/QoS-Flow e *Radio Bearer* com o OF-*Bearer* proposto, e mudanças no *Stack* da rede móvel quando MPLS é utilizado.

e a exterior identifica o OF-Bearer a ser utilizado. A Figura 4.3 expõe o uso de rótulos MPLS aplicados ao EPS-Bearer por default (OF-Bearer por default 2).

As atuais redes móveis propõem o uso de elementos chamados AF para detectar o tráfego com requisitos de QoS. Estes elementos devem ser colocados em lugares estratégicos para que o tráfego a ser analisado os atravesse, o que reduz a flexibilidade da solução; enquanto os AF devem ser desenhados para detectar requisitos de QoS para tráfego de um tipo específico. Na lógica proposta, a função de detecção dos requisitos de QoS é distribuída em todos os OF-NBs, e é colocada de forma flexível e programável considerando diferentes perfis de usuário. Quando esta política detecta tráfego com requisitos de QoS, este é endereçado a uma nova entidade (localizada em lugares centrais como centros de dados) que extraem os requisitos de QoS, método que incrementa as possibilidades de detecção de tráfego e engenharia de tráfego.

A diferença das arquiteturas SDN tradicionais que devem aguardar pela resposta do Controlador para encaminhar o tráfego ou utilizam entradas de fluxo pré-configuradas para encaminhar os pacotes, este trabalho propõe uma lógica combinada que permite encaminhar os pacotes imediatamente utilizando regras gerais, e ao mesmo tempo encaminha os pacotes com requisitos de QoS para sua inspeção em búsqueda de suas necessidades de QoS. Como será descrito nas seções seguintes, este método incrementa a velocidade de encaminhamento, e ao mesmo tempo, reduz o impacto durante tempos de resposta elevados do Controlador.

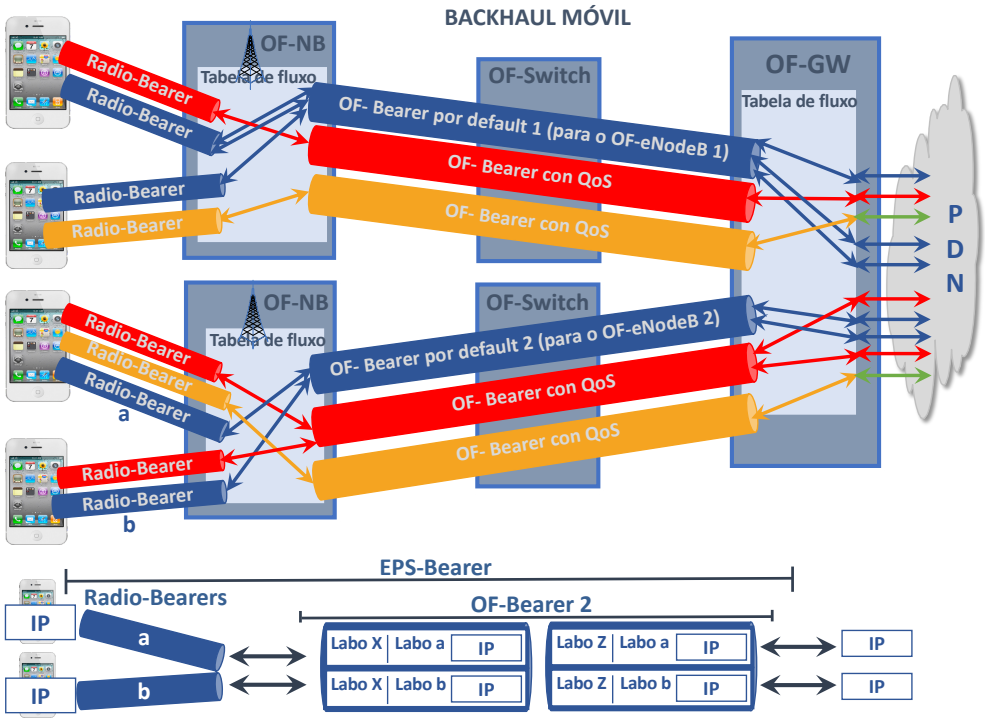

Figura 4.3: Representação de : Rótulos MPLS, EPS-*Bearer*, *Radio Bearer*, e OF-*Bearer*.

4.3 Arquitetura proposta

4.3.1 Descrição geral

A proposta cria duas lógicas especializadas para o encaminhamento de tráfego. Uma para tráfego sem requerimentos QoS e outra para tráfego com QoS. O primeiro mantém um elemento central no plano de dados chamado OF-GW e todo o tráfego tem que ser encaminhado através de OF-GW para alcançar o destino. As características chave para a lógica de encaminhar tráfego sem requerimentos de QoS são o envío mais rápido, políticas de encaminhamento mais simples e robustez de rede. Por outra parte, o tráfego com requisitos de QoS é encaminhado utilizando EPS-Bearers específicas criadas para cumprir os requerimentos de tráfego QoS como latência ultra-baixa, comunicação ultra-confiável, alta velocidade de comunicação ou outros.

A arquitetura proposta utiliza os dois modos de estabelecimento de fluxos *OpenFlow* [90]: o método proativo onde o caminho é estabelecido de forma adiantada e o modo reativo onde o Controlador escuta os *switches* para configurar fluxos sob demanda, ambos os métodos com suas características particulares. Quando o tráfego é recebido, o método proativo não requer ação do Controlador. O tráfego é imediatamente encaminhado utilizando regras pré-configuradas, mas os pacotes se enviam utilizando critérios gerais. Por outra parte, o método reativo é perfeito para a criação de caminhos com requerimentos de QoS específico sob demanda, mas requer maior processamento do Controlador e se deve aguardar pela resposta dele para encaminhar o tráfego. Ambos os métodos são amplamente utilizados na literatura acadêmica SDN [90], mas este trabalho propõe outra opção onde o tráfego com requisitos de QoS é processado utilizando ambos os métodos

ao mesmo tempo.

A idéia geral é resumida da seguinte forma: quando um UE é *attached* no OF-NB, este inicialmente tem uma EPS-Bearer padrão, utilizada para encaminhar todo o tráfego sem requisitos de QoS ao OF-GW. Quando o UE envia um pacote com requisitos de QoS, o OF-NB recebe o pacote e o envia simultâneamente por duas linhas de processamento. Uma de essas linhas utiliza a EPS-*Bearer* por *default* para encaminhar imediatamente o pacote como tráfego sem QoS (processamento utilizando o método proativo), e a outra linha de processamento simultâneo é a encarregada de detectar o tráfego com requisitos de QoS enviando uma cópia do pacote a um novo elemento chamado SDD (*Service Detector Device*) responsável de extrair os requisitos de QoS específicos. Esta informação é enviada à entidade gerenciadora (tipo PCRF na arquitetura 4G ou PCF na 5G), a qual executa um procedimento para a criação de um EPS-Bearer com QoS específicos para atender esse tráfego (processamento utilizando o método reativo). Com o EPS-Bearer com QoS específico criado, os seguintes pacotes do mesmo fluxo se encaminhan utilizando-o.

4.3.2 Configuração inicial

Inicialmente, a rede móvel conhece os diferentes tipos de serviços que oferecer como tráfego IMS para VoIP, tráfego IMS para vídeo, conectividade com servidores de jogos online, TV móvel, comunicação com mínimos atrasos, etc. (todas com seus requerimentos de QoS específicos). Cada cliente (UE) indica seus requerimentos de serviço, com suas caraterísticas de QoS, e com isto é associado pela operadora móvel um perfil de usuário específico. Existirão um grupo de perfis de usuário, e cada UE estará associado com um deles (esta associação é mantida numa base de dados por o provedor de seriviço).

Inicialmente, o Controlador conhece a topologia da rede móvel, e tem a capacidade de configurar entradas de fluxo em todos os OF-*switches*, OF-NBs, e OF-GW, pelo que o Controlador realiza as ações proativas iniciais indicadas a seguir:

- Cria entradas de fluxo *OpenFlow* em todos os equipamentos para que possam alcançar os OF-GW e para que os OF-GW possam alcançar outros equipamentos, outros OF-GW, os servidores de aplicação e a saída da rede móvel;
- Para que possam ser encaminhado o tráfego sem requerimentos de QoS, o Controlador cria os OF-Bearer por default entre cada OF-NB e seu correspondente OF-GW. Todos os OF-NB têm que ter como mínimo uma OF-*Bearer* por *default* para alcançar um OF-GW;
- Para o transporte dos serviços com requerimentos de QoS massivos são criadas OF-*Bearer* com QoS entre os OF-NBs e o OF-GWs apropriado;
- Finalmente são criadas em cada OF-NB todas as tabelas de detecção de tráfego para todos os perfis de usuários. Será depois no processo de *attachement* que será ligado o UE a uma de estas tabelas de detecção de tráfego (correspondente ao seu perfil de usuário).

Os OF-*Bearers* podem opcionalmente ser implementados com uma política de gerenciamento

de filas para que possam ser priorizados os tráfegos de diferentes tipos de OF-*Bearers*. Também os OF-*Bearers* poderão ter reservas de largura de banda, caminhos com atrasos mínimos, mecanismos para recuperação imediata ante quedas, ou outras alternativas. Todos os OF-*Bearers* serão univocamente identificados pelo rótulo MPLS exterior, o qual pode trocar salto a salto quando o tráfego atravessa a rede MPLS móvel (mudará como em qualquer implementação de rede MPLS). Este rótulo MPLS exterior junto ao rótulo MPLS interior identificam univocamente o EPS-*Bearer* a ser utilizado por um UE. Por outra parte, se os EPS-*Bearers* de diferentes UE têm os mesmos requerimentos de QoS, estes poderão utilizar o mesmo OF-*Bearer* com QoS (ver Figura 4.3.

4.3.3 Lógica em OF-NB e de detecção de tráfego

Prévio ao processo de criação de *bearers* específicos, devem ser desenvolvidos mecanismos que permitam detectar as necessidades de requisitos de QoS da forma mais transparente possível para os UE. Com esta finalidade, as redes atuais propõem a incorporação de elementos lógicos denominados AF localizados em pontos estratégicos para a detecção de processos de sinalização. Um exemplo de isto é a colocação de um AF no P-CSCF para a detecção da sinalização SIP de uma chamada IMS. Uma vez que o AF detecta as mensagens de sinalização e extrai os requisitos de QoS de estas mensagens, o AF envia esta informação ao PCRF, equipamento encarregado de realizar a aplicação das políticas nas redes móveis atuais.

Mas os serviços que podem ser detectados com os AF têm limitações e requerem ser colocados em pontos específicos que não sempre é possível colocar. É requerida uma alternativa para a detecção do tráfego com requisitos de QoS mais flexíveis e com políticas dinâmicas e facilmente gerenciáveis. Por outra parte, é desejável que estes mecanismos não requeiram da interação dos UE, para que estes não tenham que implementar novos protocolos e processamentos dificilmente extensíveis a todos os UE. Adicionalmente, estes mecanismos têm que evitar a adição de atrasos que podem afetar a qualidade da comunicação fim a fim, o que representa um desafio para qualquer implementação que utilize SDN-*OpenFlow*.

No presente trabalho é proposta a criação de uma lógica de processamento *OpenFlow* nos OF-NB para a detecção do tráfego com requisitos de QoS. Esta lógica é construída durante o processo de *attachment* inicial pelo Controlador, e pode ser adaptada por ele quando seja requerido. Na Figura 4.4 é indicada a forma como os OF-NB processam todo o tráfego *Uplink*, o qual inclui o tráfego para os serviços com requisitos específicos de QoS, o tráfego para o encaminhamento do tráfego sem requisitos de QoS, e a detecção do tráfego com requisitos de QoS.

A lógica de detecção de tráfego é pensada para substituir elementos da arquitetura 3GPP como o AF e a TDF (*Traffic Detection Function*) de uma forma flexível e programável, e adicionalmente tem a finalidade de mover a análise do tráfego e os contadores de facturação mais próximos aos UEs.

Todos os OF-NB têm inicialmente pré-configuradas todas as tabelas de detecção de tráfego, e durante o processo de *attachment* cada UE é vinculado a uma de essas tabelas de detecção dependendo do perfil de usuário. O perfil de usuário a ser utilizado pode se obter durante o

processo de *attachment* consultando uma base de dados apropriada, utilizando o identificador QCI (QoS *Class Identifier*) proporcionado pelo HSS, ou consultando uma nova base de dados da arquitectura 5G.

A Figura 4.4 representa as diferentes tabelas *OpenFlow* que compõem a lógica de encaminhamento no sentido *uplink* nos OF-NB. A referida figura indica M tabelas de detecção de tráfego para M diferentes perfis de usuários, mas é detalhada somente a tabela correspondente ao perfil de usuário Y. Também é exposto como o UE x lhe é asignado a tabela de detecção de tráfego correspondente ao perfil de usuario Y, vinculação que se consegue através da entrada de fluxo por *default* para o UE x (ver tabela 0).

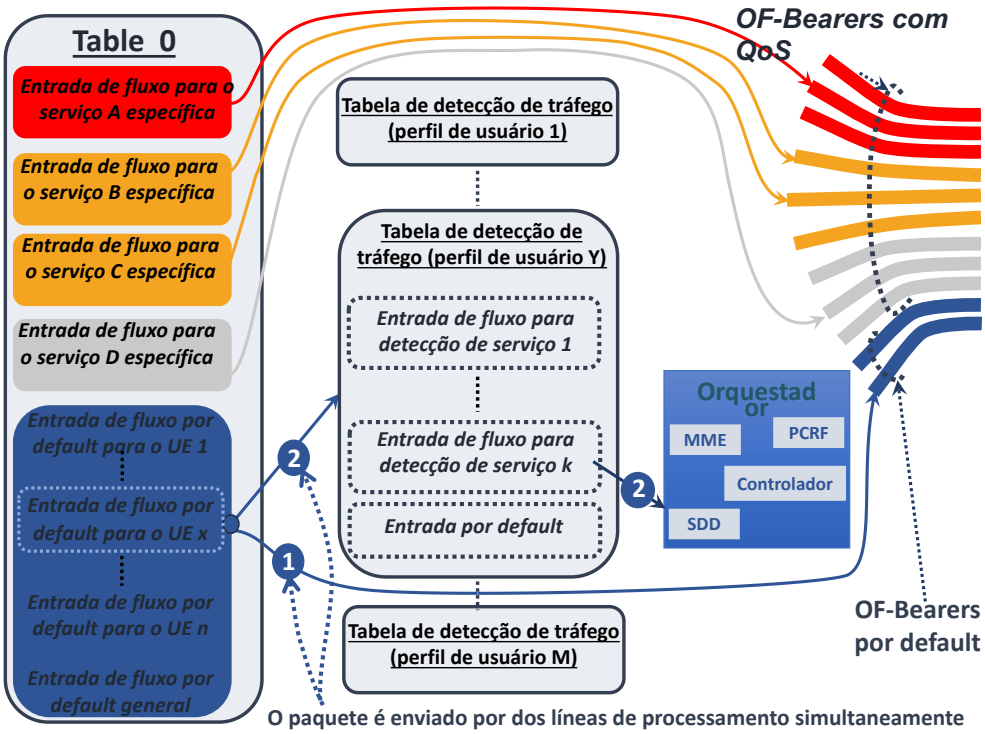

Figura 4.4: Lógica de detecção de tràfego no OF-NB.

A tabela 0 proposta, implementada em OF-NB, está constituída por três tipos de entradas de fluxo:

- Entrada de fluxo para serviço específica: Esta é utilizada para encaminhar tráfego de um UE que possui requerimentos de QoS particulares. Para isso é utilizado um OF-*Bearer* com QoS específico, criado sob demanda.
- Entrada de fluxo por *default* para o UE: Cada UE *attached* tem sua própria entrada de fluxo por *default*, utilizada como a rota *default* para encaminhar todo tráfego originado no UE. Esta entrada de fluxo é criada durante o processo de *attachment* e é a responsável por vincular o *Radio-Bearer* por *default* com a OF-Bearer por default (o que cria a EPS-*Bearer* por *default* para o UE, (ver Figuras 4.2 e 4.3). Adicionalmente, este tipo de entrada de fluxo é responsável por vincular o tráfego do UE com a tabela de detecção de tráfego correspondente

ao seu perfil de usuário.

- Entrada de fluxo por *default* geral: é responsável de descartar todo o tráfego que seja recebido desde uma origem móvel inválida.

Antes de que o UE x se encontre *attached*, o OF-NB não possui entradas de fluxo para serviços específicos nem entrada de fluxo por *default* para o UE x (esta última é criada durante o processo de *attachment)*. O critério para encontrar coincidência de tráfego que utiliza a entrada de fluxo por *default* para o UE x está baseado na IPv-4 origem, o prefixo IPv-6 origem ou a *Radio Bearer* por *default*, os quais inequívocamente identificam o UE que originou o tráfego. Se o tráfego desde o UE x não encontra coincidências com entradas de fluxo de serviço específicas, este será sempre encaminhado pela entrada de fluxo por *default* para o UE x.

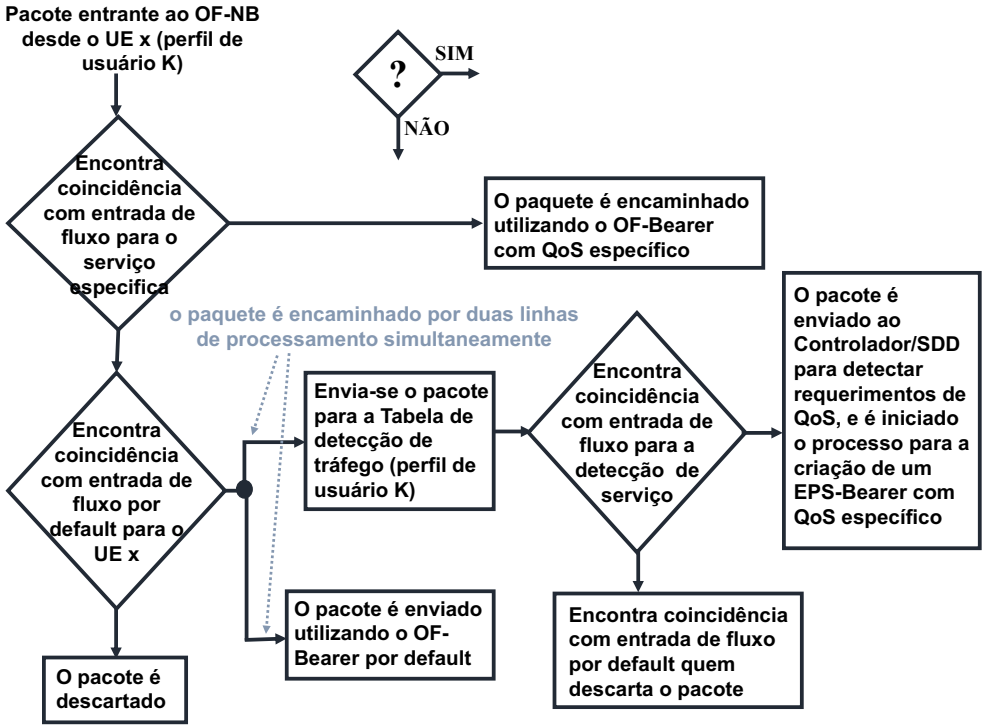

Figura 4.5: Diagrama de fluxo para a lógica de detecção de tráfego no OF-NB.

A lógica proposta no OF-NB se detalha na Figura 4.4 enquanto o diagrama de fluxo é descrito na Figura 4.5. Quando o tráfego no sentido *uplink* encontra coincidência na entrada de fluxo por *default* para o UE x, o tráfego é enviado por duas linhas de processamento em simultáneo, sendo una proativa que encaminha o pacote de forma imediata como tráfego sem requisitos QoS (utilizando o OF-*Bearer* por *default*) e a outra linha de processamento envia o pacote à tabela de detecção de tráfego do perfil do usuário Y. A seguir são detalhadas ambas as linhas de processamento:

1. A entrada de fluxo por *default* para o UE x tem a instrução do tipo *apply-action* e dentro de esta, duas ações Push-MPLS responsáveis por identificar inequívocamente o EPS-*Bearer* por *default* a utilizar para o envío do tráfego. O rótulo MPLS mais interno, identifica o

UE/*Radio Bearer* por *default*, enquanto o rótulo externo identifica a OF-*Bearer* por *default*. Adicionalmente, a instrução do tipo *apply-action* tem a ação *output* indicando a porta de saída apropriada, a qual envia uma cópia do pacote utilizando o OF-Bearer padrão (comportamento padrão de uma ação *output* de uma instrução de tipo *apply-action* [77]).

2. Simultâneamente ao indicado no punto 1, a entrada de fluxo por *default* para o UE x tem a instrução do tipo *goto table*, a qual envia o pacote à tabela de detecção do tráfego correspondente ao perfil de usuário Y. Esta tabela consiste em um grupo de entradas de fluxo responsáveis por detectar tráfego com requisitos específicos de QoS, de acordo ao perfil do usuário do UE x. Quando o tráfego encontra coincidências com uma entrada de fluxo para a detecção de um serviço nessa tabela, a entrada indica que devem ser adicionados dois rótulos MPLS adequados para encaminhar o pacote ao SDD. Por outra parte, se não se encontram coincidências com uma entrada de fluxo de detecção de serviço específica, o tráfego é descartado sem executar nenhuma ação, devido a que os pacotes que atinge essa instância já foram encaminhado como tráfego sem requisitos de QoS (como foi indicado no passo 1).

Se o pacote é enviado ao SDD, este é processado na búsqueda das características que identificam o serviço como tipo de multimédia, direção IP origem, direção IP destino, protocolo, porta utilizada, tipo de *encoder*, requisitos de QoS, etc. Seguidamente, o SDD envia a informação do serviço ao elemento móvel PCRF da arquitetura 4G ou PCF da arquitetura 5G, o qual inicia o procedimento para a criação de un EPS-Bearer com QoS específico. É importante notar que o OF-Bearer com QoS e o radio-Bearer com QoS são interconectados utilizando entradas de fluxo de serviço específicas (ver Figura 4), o que cria o EPS-Bearer com QoS.

4.3.4 Criação de EPS-Bearers com requerimentos de QoS

A Figura 4.6 descreve o procedimento para a criação de EPS-*Bearers* específicos considerando um UE 4G (um procedimento similar pode ser definido utilizando uma tecnologia de acesso 5G). Como será mostrado, para este procedimento não é requerido realizar mudanças nas terminais móveis 4G, porém, o procedimento de criação de EPS-*Bearer* com QoS proposto requer significativas mudanças sé é comparado com o procedimento utilizado hoje na arquitetura PCC (*Policy Control and Charging*). A seguir é descrito o procedimento proposto.

1. Inicialmente, o tráfego com requerimentos de QoS é encaminhado simultaneamente utilizando o OF-*Bearer* por *default*, e é enviado à tabela de detecção de tráfego, que encaminha o tráfego para o SDD. Logo, o SDD processa os pacotes é para determinar as caraterísticas e necessidades do serviço. Esta informação é enviada para o PCRF (ou para um novo elemento com essa funcionalidade), o que debe definir as políticas a serem aplicadas.

2. Se é requerida informação adicional, o PCRF pode solicitar esta informação a outro elemento móvel. Como exemplo, pode ser realizada uma solicitação de informação de crédito disponível ao OCS (*Online Charging System*) o que determinará o tempo ou quantidade de tráfego que pode ser transmitida para esse serviço.

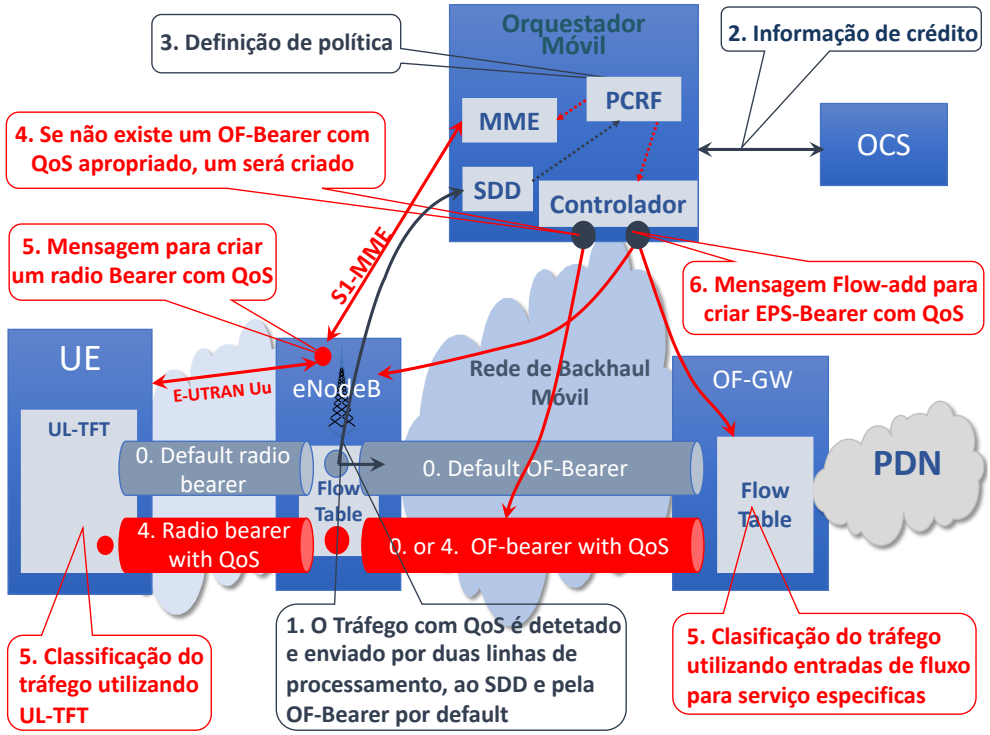

Figura 4.6: Procedimento para a criação de EPS-Bearer com QoS.

3. Com a informação da característica do tráfego, do perfil de usuário, de o credito disponível, o PCRF define a política a ser aplicada (as características que deve ter o EPS-*Bearer* a ser criado). Finalmente, á politica é comunicada ao MME e ao Controlador.

4. Se já existe um OF-Bearer com as características de QoS apropriadas para esse tráfego, este será utilizado. Por outro lado, se ainda não existe um OF-Bearer com essas caraterísticas, o Controlador criará um novo OF-Bearer que cumpra com os requerimentos de QoS. Para isso, o Controlador define o rótulo MPLS externo apropriado e cria as entradas de fluxo em cada um dos *switches OpenFlow* envolvidos no caminho com QoS fim a fim.

5. A seguir são indicados os passos para a criação de *Radio Brearer* com QoS:

   (a) O MME envia uma mensagem de solicitação 3GPP padrão do tipo *bearer setup request* (utilizando a interface padrão S1-MEE) ao OF-NB indicando a necessidade de criação de um *Radio Bearer* apropriado.

   (b) O OF-NB envia o correspondente *RRC* (*Radio Resource Control*) *connection reconfiguration* ao UE (procedimento padrão utilizando a interface E-UTRAN-Uu) indicando que deve ser criado um *Radio Bearer* com QoS. Esta mensagem tem a informação de requerimentos de QoS associada e tem os filtros TFT (*Traffic Flow Template*) a ser aplicados no UE. Estes filtros incorporam as regras para classificar o tráfego no sentido *uplink* e para encaminhar ele por um novo *Radio Bearer* com QoS.

   (c) O UE responde ao OF-NB com uma mensagem padrão *RRC connection reconfiguration complete*.

(d) O OF-NB responde ao MME como uma mensagem padrão *bearer setup response*, completando a etapa de criação do *Radio Bearer*. No final de este passo o tráfego continua sendo encaminhado utilizando o OF-*Bearer* por *default*.

6. Como é descrito a seguir, o *Radio Bearer* com QoS e o OF-*Bearer* com QoS são ligados para criar um EPS-*Bearer* com QoS..

   (a) O Controlador define a entrada de fluxo necessária no sentido *downlink* (a configurar no OF-NB) para ligar o tráfego que vem desde o OF=Bearer com QoS com o *Radio Bearer* com QoS agora criado. Neste passo fica definido o rótulo MPLS interno, o que identifica o UE/*Radio Bearer*.

   (b) O Controlador define a entrada de fluxo no sentido *downlink* que deve ser aplicado no OF-GW para encaminhar o tráfego do serviço com QoS através do OF-*Bearer* com QoS (com destino o UE). As entradas de fluxo são similares às entradas de fluxo para o serviço específicas presentes no OF-NB (ver Figura 4.4). Adicionalmente os critérios de coincidência são baseados nas características do serviço e o endereço IP destino do UE, e a entrada de fluxo colocada no OF-GW deve adicionar os dois rótulos MPLS apropriados. Em este passo, o tráfego no sentido *downlik* pode já ser encaminhado utilizando o novo EPS-*Bearer* com QoS.

   (c) Se não foram previamente configuradas, o Controlador cria as entradas de fluxo específicas (no OF-GW) para encaminhar o tráfego para o serviço com QoS no sentido *uplink*.

   (d) Finalmente, como pode ser observado nas Figuras 4.4 e 4.6, o Controlador cria as entradas de fluxo para o serviço específicas no OF-NB, de forma de ligar o tráfego que usa o *Radio Bearer* com QoS com o OF-*Bearer* com QoS. Esta entrada de fluxo tem seu critério de coincidência baseado nas caraterísticas do serviço e do endereço IP origem. Em este passo o tráfego nos sentidos uplink e downlink já podem ser encaminhados, devido a que o EPS-*Bearer* com QoS fica estabelecido nas duas direções.

As entradas de fluxo são criadas com uma prioridade *OpenFlow* adequada, a qual determina a ordem em que elas são processadas. Particularmente, as entradas de fluxo para serviços específicas sempre têm maior prioridade que as entradas de fluxo por *default* para o UE. Por tanto, se uma entrada de fluxo para o serviço específica existe, o tráfego será processado utilizando esta entrada (ver Figura 4.6).

É importante notar que os OF-GW são *switches OpenFlow* padrão situados na borda da rede móvel ou conectados aos servidores de aplicação. A posição na rede onde eles são colocados faz que eles sejam apropriados para terminação de OF-Bearer por *default*, e em vários casos para a terminação de OF-*Bearers* com QoS. Os OF-Bearers com QoS podem ser criados diretamente entre qualquer par de switches *OpenFlow*, sendo um exemplo de este conceito o que mostra um OF-*Bearer* criado diretamente entre dois OF-NB (representado na Figura 5.10).

4.3.5 Considerações adicionais

Devido a que o tráfego pode ser encaminhado diretamente entre dois OF-NB sem requerer o passo por um OF-GW, os contadores de tráfego e de tempo de duração de fluxos devem ser movimentados à borda da rede de acesso móvel. Particularmente para isto podem ser utilizados os contadores das entradas de fluxo para serviço específicas e os contadores das entradas de fluxo por *default* para os EU ver Figura 4.4. Se durante a criação de um EPS-*Bearer* com QoS, a consulta a uma base de dados (como a OCS) informa que o EPS-*Bearer* tem que ser criado com x minutos de crédito, a entrada de fluxo para o serviço específica deve ser criada com um *idle-timeout* de x minutos indicados. Por outra parte, se o credito é relacionado a uma quantidade de tráfego, a entrada de fluxo para o serviço especifica deve ser examinada periódicamente até que o contador de tráfego de esse fluxo atinga o limite estabelecido.

Durante a operação normal, o tráfego sem requerimentos de QoS é encaminhado utilizando o caminho por *default* previamente configurado, e o Controlador só interatúa se o tráfico tem requerimentos de QoS ou no processo de *attachment*, o que reduz o número das mensagens *OpenFlow* que o Controlador deve processar (diminuindo a carga no Controlador).

Adicionalmente, como a arquitetura proposta sempre encaminha os pacotes imediatamente utilizando o OF-*Bearer* por *default*, esta arquitetura tem menos impacto durante tempos de resposta elevados do Controlador, dado que o tráfego continua sendo encaminhado utilizando o OF-*Bearer* por *default*. Este comportamento incrementa a roubustez da solução se esta é comparada com implementações que utilizam SDN de forma tradicional nas que os pacotes pertencentes ao novo fluxo com QoS devem aguardar a resposta do Controlador para ser encaminhados.

É importante notar que a lógica de detecção de tráfego proposta está composta por dois estagios principais, um implementado de forma distribuída nos OF-NB utilizando a tabela de deteção de tráfego, e o segundo estagio implementado nos SDD utilizando software dedicado para personalizar as caraterísticas de QoS que devem ser detetadas. O primeiro estagio é limitado pelas combinação de critérios de coincidência possíveis por o OpenFlow, e a segundo é programável e ilimitado (portanto a proposta combinada permite critérios de detecção ilimitados). Como um exemplo prático, pode ser considerado un caso onde debe ser identificado um conteúdo especifico de camada de aplicação quando os pacotes são encaminhados a um servidor de aplicaçao específico. Em este caso, a tabela de detecção de tráfego no OF-NB é responsável por encontrar coincidência quando o pacote é encaminhado ao servidor de aplicação a uma porta TCP/UDP específica, e é responsavel por encaminhar esse tráfego detectado a um SDD apropriado. Uma vez que o pacote é recebido no SDD, este realiza uma inspeção no nível de aplicação aprofundada para encontrar os requerimentos.

Por outra parte, para reduzir o processamento nos elementos SDD, estos podem ser implementados de forma distribuída dentro do domínio da rede móvel. De esta forma, os OF-NB podem ser divididos em grupos onde cada grupo é atendido por um SDD que dependerá da locação geográfica. Outra alternativa é criar SDDs específicos para analisar diferentes tipos de tráfego. Em este último caso são os OF-NB os responsáveis pela classificação do tráfego com requerimentos de QoS e seu encaminhamento ao SDD correspondente.

Finalmente, para incrementar a robustez da solução, as funções do Controlador podem ser divididas e colocadas em três Controladores diferentes trabalhando juntos de forma complementaria [91]. Uma possível separação é utilizar um Controlador para a configuração proactiva dos fluxo, sendo este responsável pela criação dos OF-*Bearer* por *default*, pelas tabelas de deteção de tráfego, e pelas regras de encaminhamento gerais. Um segundo Controlador pode ser responsável pelo processo de *attachment*, sendo o encarregado de criar o EPS-*Bearer* por *default* e pela ligação de um UE com a tabela de detecção de tráfego correspondente. Finalmente, um terceiro Controlador pode ser responsável pela criação de EPS-*Bearers* com QoS específicos.

4.3.6 Processo de *attachment* para UE 4G

Para melhor entendimento do procedimento de *attachment* na nova arquitetura é útil previamente compreender o processo de *attachment* de um UE na arquitetura 4G. É por isto que inicialmente são descritos os passos para esse *attachment*, e posteriormente são descritas as mudanças no procedimento para a arquitetura proposta no trabalho. O exemplo escolhido de *attachment* inicial requer a mudança de localização do UE no HSS, devido a uma mudança de MME. Este exemplo foi particularmente selecionado devido a que também permite mostrar a interação entre a arquitetura proposta e a arquitetura 4G. O exemplo pode ser fącilmente modificado para representar o caso em que um usuário se movimenta da rede 4G à arquitetura móvel SDN-*OpenFlow* descrita no trabalho. Informação adicional sobre os procedimentos de *attachment* nas redes 4G pode ser encontrada no [62].

4.3.7 Processo de *attachment* na arquitetura 4G

Na Figura 4.7 é mostrado o processo de *attachment* inicial em uma rede 4G. A seguir são descritas as etapas do processo de attachement em resumo. No exemplo escolhido o GUTI (*Globally Unique Temporary Identity*) é mudado, devido à mudança do MME ao que o UE se encontra ligado:

A. O UE envia um *Attach Request* ao eNB o qual examina o MME ID que é transferido pela camada RRC. Como o eNB não tem um enlace com o MME identificado, o eNB escolhe um novo MME e envia o Attach Request ao novo MME.

B. Como o MME tem mudado, o novo MME utiliza o MME ID do GUTI para achar o MME antigo e recuperar assim a informação de contexto.

C. É realizada a autenticação e são executados os mecanismos de segurança. A informação de identidade também é recuperada neste passo.

D. O novo MME informa ao HSS que o UE tem se movimentado. O HSS guarda a direção do novo MME, e indica ao antigo MME que cancele o contexto com o UE.

E. O *default Bearer* é autorizado pelo PCRF e é estabelecido entre o SGW e o PGW. Muitos parâmetros são intercambiados e considerados neste passo como: IMSI, UE IP, PDN ID (APN), PCC Rule (SDF filter, QCI, ARP, APN-AMBR (UL/DL), etc.

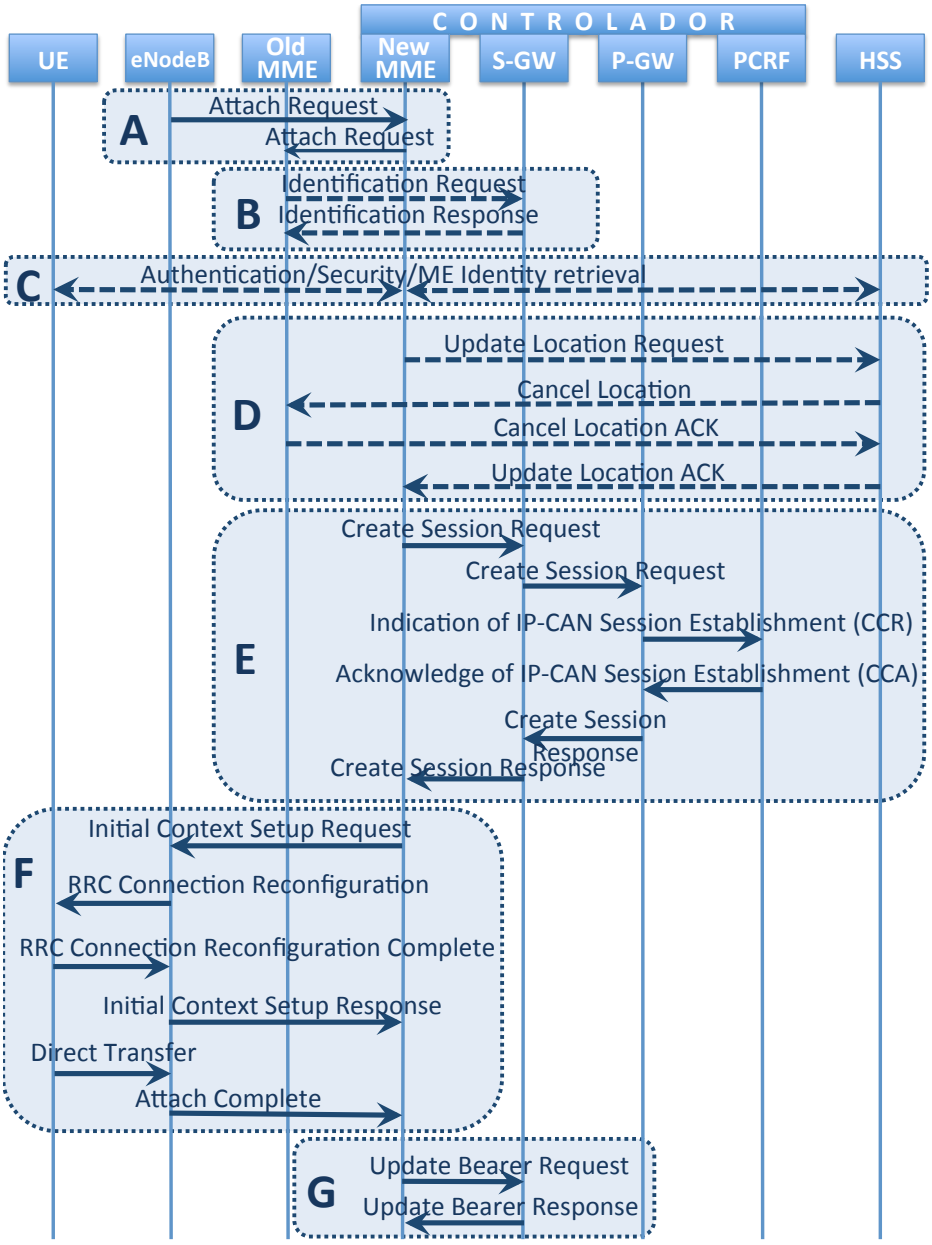

Figura 4.7: Processo de attachment inicial nas redes 4G padrão

F. O *default Bearer* é estabelecido sobre a interface de rádio entre o UE e o eNB. O *Attach Accept* e enviado ao UE dentro da mensagem *RRC Connection Reconfiguration*. Neste procedimento são enviadas todas as informações requeridas ao UE como: GUTI, UE IP, TAI List, APN, EPS-*Bearer* ID, *Authorized QoS Profile* (QCI, APN-AMBR (UL), TFT(UE), etc.

G. O MME informa ao S-GW o TEID (*Tunnel Endpoint Identifier*) do eNB, o que completa o estabelecimento do *default bearer*.

No procedimento previamente descrito foram ressaltados os aspectos mais relevantes para nosso trabalho, mas é recomendável uma leitura aprofundada dos detalhes envolvidos em cada mensagem, assim como os processos internos que os elementos lógicos devem realizar entre cada mensagem [92],[62].

4.3.8 Processo de *attachment* na arquitetura proposta

Na arquitetura SDN-*OpenFlow* proposta tanto o MME, o PCRF quanto o controle do SGW e do PGW se encontram formando parte constitutiva do Controlador. Na Figura 4.7 também foi ilustrada a correspondência de estes elementos lógicos constitutivos com o Controlador para facilitar a visualização das mudanças e as similitudes com a arquitetura proposta. Como pode ser observado em essa figura, só nas etapas E e G os elementos constitutivos do Controlador interagem entre eles, o que provoca significativas mudanças nestas etapas. Entretanto, as outras etapas se mantêm sem significativas alterações, devido a que as interfaces S1-MME, S10, S6a, E-Utran Uu, permanecem invariáveis.

Na Figura 4.8 se ilustra o processo de *attachmet* inicial quando um usuário se conecta à arquitetura proposta (pode ser considerado que anteriormente estava em uma rede 4G padrão). Como pode ser observado, a etapa G não é requerida devido a que envolve a comunicação de dois elementos constitutivos do Controlador. No que refere à etapa F, esta agora forma parte constitutiva da etapa E. Na arquitetura proposta, é o Controlador quem ordena ao OF-NB (mediante a interface padrão S1-MME) o estabelecimento do contexto e a criação dos *Bearers* iniciais. Por outra parte, é considerado que prévio ao início da etapa E, o Controlador dispõe de toda a informação do perfil do usuário, a qual pode ser obtida nas etapas prévias e mediante uma consulta ao DDR ou pode ser modificada a informação contida no valor do QCI para que este contenha a informação do perfil do usuário. Seja qual for o procedimento escolhido para disponibilizar da informação do perfil de usuário ao Controlador, na etapa E o Controlador deve: configurar o OF-GW para que ligue o *Bearer* externo como o OF-*Bearer* por default correspondente (previamente configurado), solicitar ao OF-NB que estabeleça o contexto e crie os Rádios *Bearer* por *default*, e finalmente o Controlador deve fazer a ligação do *Radio Bearer* por *default* com o OF-*Bearer* por *default* formando o EPS-*Bearer* por *default* correspondente.

A seguir são descritos com maior profundidade os passos que constituem a etapa E, devido a que esta contém os aspectos chave da presente arquitetura. Como foi mencionado, prévio a etapa E, o Controlador dispõe de toda a informação do perfil do usuário, e determina outras informações necessárias para que o processo de *attachment* seja efetuado. Entre estas informações se encontram

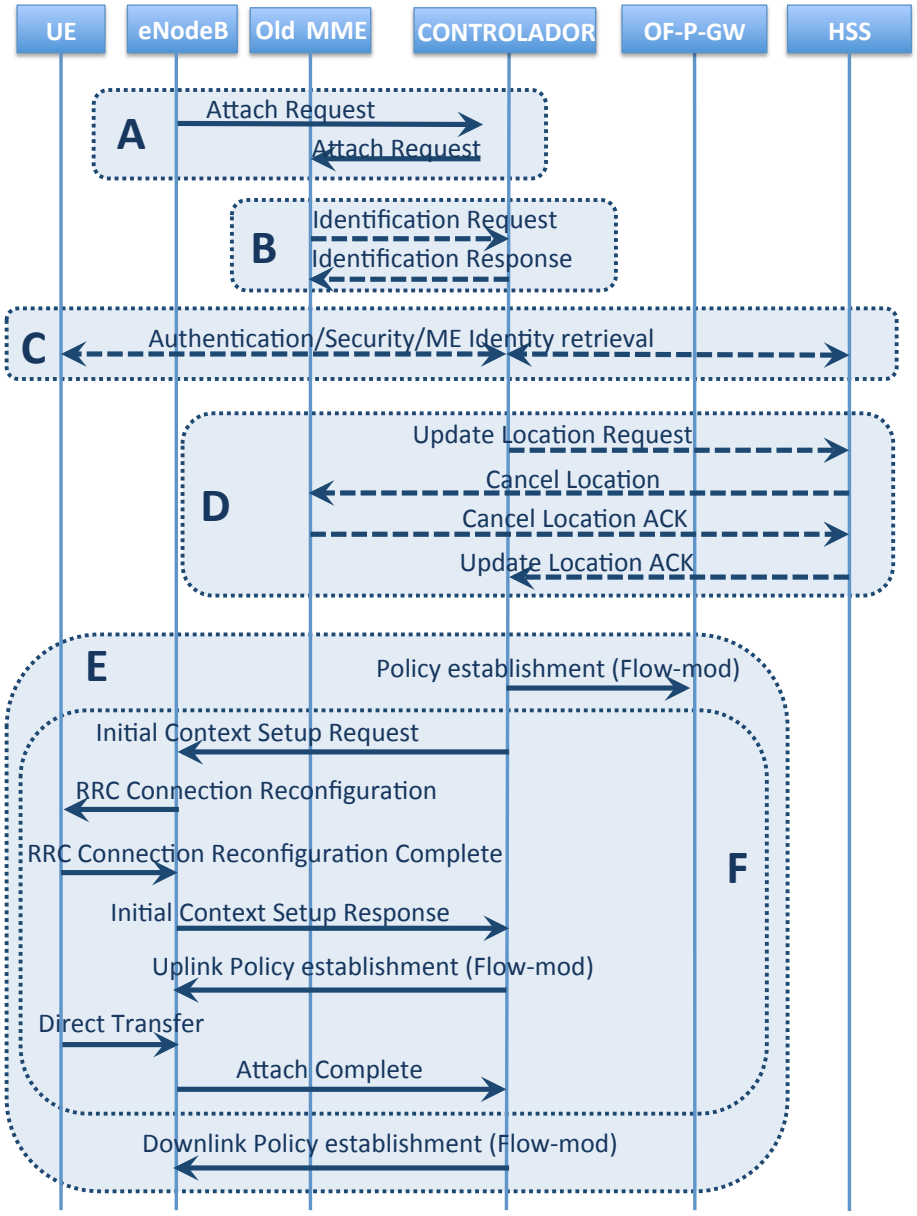

Figura 4.8: Processo de *attachment* inicial na arquitetura proposta

os serviços contratados pelo usuário, PDN ID (APN), QCI, APN-AMBR (UP/DL), TFT, SDF, IP UE, GUTI, *charging type*, etc. Com esta informação são realizas as ações descritas as seguir:

1. O Controlador escolhe o OF-GW (se ainda não foi escolhido)
2. O Controlador cria as regras de tráfego nas tabelas de fluxo do OF-GW que permitam ligar o tráfego *Downlink* com destino o UE, com o apropriado OF-*bearer* sem requisitos específicos de QoS. O OF-*bearer* selecionado foi criado previamente pelo Controlador (este interconecta o OF-GW com o OF-NB ao que o UE se encontra em processo de *attachment*).
3. O Controlador cria as entradas de fluxo no OF-GW que permitem encaminhar o tráfego *Uplink* do UE.
4. O Controlador envia ao OF-NB uma mensagem *Initial Context Setup Request* (pela interface S1-MME) para que o *attachment* seja estabelecido. Esta mensagem contem todos os parâmetros requeridos para a criação do contexto e os *Radio Bearers* iniciais.
5. O OF-NB envia o correspondente *RRC Connection Reconfiguration* ao UE para realizar o *attachment*. Este contém a informação que foi enviada no passo prévio pelo Controlador como: a informação de contexto, *Radio Bearer* inicial, parâmetros de QoS associados, e os filtros TFT que têm que ser incorporados no UE para classificar o tráfego *Uplink* pelo *Radio Bearer* configurado.
6. O Controlador cria as entradas de fluxo nas tabelas do OF-NB para que o tráfego em sentido *Uplink* seja encaminhado ao OF-*Bearer* por *default*. Adicionalmente, é criada a logica (nas tabelas de fluxo do OF-NB) que permite encaminhar o tráfego à tabela de detecção de tráfego correspondente ao peril do usuário.
7. O UE responde ao OF-NB com um *RRC Connection Reconfiguration* Complete, e seguidamente o OF-NB responde ao Controlador com uma mensagem *Initial Context Setup Response*.
8. No sentido *Downlink* do UE são criadas as entradas de fluxo para que o tráfego com destino o UE seja encaminhado pelo *default Radio Bearer* criado.

Finalizado o processo de *attachment*, todo o tráfego *Uplink* é encaminhado pelo *Radio Bearer* por *default* até o OF-NB e logo colocado no OF-*Bearer* por *default* até o OF-GW. Posteriormente o OF-GW encaminha o tráfego até o destino, que em caso de se encontrar fora do domínio móvel e encaminhado pela rede de transporte do SP. No sentido *Downlink*, o tráfego com destino o UE é recebido pelo OF-GW, quem o coloca no OF-*Bearer* sem requisitos de QoS até o OF-NB. Finalmente este o coloca no *Radio Rearer* por *default* criado até o UE destino.

4.4 Processo de *handover*

O processo de *handover* escolhido é da família de métodos de mobilidade controlados pela rede. Neste tipo de métodos todo o processo de mobilidade é realizado pela rede móvel, e os UE não

requerem de uma intervenção ativa no processo. As IPv4 ou prefixos IPv6 são mantidos em todo o processo, sendo que o UE não percebe diferênça entre os dois pontos de *attachment*.

Para entender o *handover* na arquitetura proposta, inicialmente são descritas cinco etapas que estão presentes nos diferentes processos de *handover* das redes 4G. Uma vez compreendidas estas etapas, são descritas as diferenças que cada etapa tem na arquitetura proposta.

1. Necessidade de *handover*: Neste passo o eNB origem é o que determina se é requerido o *handover* baseado no nível de sinal de rádio do UE. Para isto, o eNB se baseia em restrições previamente impostas pela rede (MME), as quais indicam os requisitos para que o *handover* seja realizado. Nesta fase é determinado o eNB destino, o qual pode estar gerenciado pelo mesmo MME que gerencia o eNB origem ou pode ser gerenciado por outro MME. Por outra parte podem ser utilizados os mesmos SGW ou podem ser utilizados diferentes. Seja qual for o tipo de *handover* a ser efetuado, são realizados os passos descritos a seguir:

2. Preparação do *handover*: Nesta fase toda a informação requerida para realizar o *handover* tem que ser comunicada ao eNB destino, e deve ser autorizado o processo. O procedimento de como a informação é enviada, e de quantos elementos lógicos da rede móvel estão envolvidos no proceso depende do tipo de *handover*. Mas no final desta etapa, é criado um canal de comunicação entre os dois eNB envolvidos no processo de *handover* que permite encaminhar o tráfego desde o eNB origem ao eNB destino (durante o *detachmet*). No caso mais simples (no que refere ao canal de comunicação) só os *switches* intermédios e os eNB estão envolvidos, mas em casos mais complexos os dois SGW e outros elementos lógicos da rede móvel também estarão envolvidos.

3. Execução do *handover*: Uma vez acordados todos os aspectos prévios para a realização do *handover*, o eNB origem envia o comando para realizar o *handover* ao UE. Durante o período de tempo que o UE se encontra desconectado dos dois eNodeB envolvidos, o eNodeB origem envia toda a informação que chega a ele (no sentido *Downlink*) ao eNB destino. Isto é realizado utilizando o canal de comunicação entre os eNB estabelecido no passo anterior. Até que o *handover* seja efetivado, o eNB destino armazena em *buffer* todo o tráfego com destino ao UE. Finalizado o *attachment* é estabelecida a comunicação bidirecional entre o UE e o eNB destino, por o que este eNB pode encaminhar o tráfego armazenado.

4. Conclusão do *handover*: No começo de este passo, o tráfego *Uplink* do UE é enviado ao eNB destino, que o envia ao SGW, quem seguidamente o encaminha ao PGW da forma certa. Mas no sentido *Downlink*, o tráfego ainda não se encontra configurado otimamente. Neste caso o tráfego *Downlink* continua sendo enviado ao eNB origem e este o encaminha pelo canal de comunicação criado até o eNB destino, quem finalmente o envia até o UE destino. Corresponde a esta etapa regularizar esta situação, criando para isto um *Bearer* apropriado no sentido *Downlink* entre o PGW e o eNB destino.

4.4.1 Diferenças das etapas com a arquitetura proposta

A etapa mais crítica do processo é a de "Execução do *handover*", devido a que esta requer da interrupção do serviço. É por isto que na rede 4G são aplicadas técnicas que otimizam este procedimento minimizando a interrupção. Como exemplo, no processo de *handover* entre dois eNB gerenciados por o mesmo MME, os eNB realizam toda a etapa de execução do *handover* sem a necessidade de intervenção do MME. Neste trabalho entende-se que grande parte do controle do processo do *handover* tem que ficar nos OF-NB, particularmente no que refere à interação entre os OF-NB e os UE e a interação entre OF-NB.

é por isto que no trabalho ainda mantemos as conexões X1 entre eNodeBs para que a etapa de "Execução do *handover*"não tenha quase mudanças, e a interrupção do serviço continue sendo a mínima possivel. Mas a etapa prévia e a posterior à de "Execução do *handover*"requerem das mudanças que são descritas a seguir.

Na etapa de "Preparação do *handover*"o Controlador sempre tem que ter conhecimento do processo (o OF-NB envia uma mensagem com a informação do *handover* ao Controlador) e é o Controlador que da a ordem ao OF-NB origem de iniciar o processo. Nesta etapa o Controlador tem que configurar o OF-NB destino para que possua as regras de encaminamento do UE em processo de *handover*. Concretamente, o Controlador tem que adicionar a lógica para o encaminhamento e detecção no OF-NB destino que permita encaminhar imediatamente o tráfego em sentido *Uplink* quando o UE se conecte. Para isto tem que ser criada a lógica de encaminhamento descrita na seção 3.2.4, a que identifica o tráfego baseado na IP origem (este endereço não muda) e nas caraterísticas do fluxo, para assim encaminhar o tráfego pelos OF-*bearer* apropriados (depois de ser processado pelas tabelas de detecção se assim correspondesse).

Adicionalmente, na etapa de "Preparação do *handover*", o Controlador tem que criar (se já não se encontra estabelecido) o canal de comunicação entre os dois OF-NB. Este canal pode ser previamente estabelecido (com entradas de fluxo no caminho entre os OF-NB vizinhos envolvidos no *handover*) no caso de que os OF-NB pertençam ao mesmo domínio de controle, mas no caso que os OF-NB pertençam a domínios de controle diferentes, os Controladores de cada domínio devem participar no processo de criação do canal de comunicação entre os OF-NB envolvidos.

Finalmente, o Controlador intervém na etapa de "Conclusão do *handover*"na configuração do caminho *Downlink* ao OF-NB destino certo. Lembra-se que no começo da etapa "Conclusão do *handover*", o caminho *Uplink* se encontra estabelecido corretamente, mas no sentido *Downlink* o tráfego continua sendo encaminhado ao OF-NB origem e posteriormente ao OF-NB destino utilizando o canal de comunicação estabelecido para esta finalidade. Por tanto, o Controlador cria o caminho *Downlik* utilizando os OF-*Bearers* adequados, e o canal de comunicação entre os OF-NB pode deixar de ser utilizado. No caso de que o *handover* seja entre dois domínios móveis diferentes, a criação do caminho *Downlink* ao OF-NB é criada por o Controlador do domínio destino, porem o canal de comunicação é apagado com a interação dos Controladores.

4.5 Interconectividade entre domínios móveis do SP e mobilidade entre domínios

Para cumprir com o objetivo de proporcionar QoS nas comunicações fim a fim, particularmente quando toda a comunicação se encontra dentro do mesmo SP, é preciso incorporar no problema o encaminhamento fora do domínio móvel. É por isto que nesta seção é incorporado o domínio de transporte do SP na análise, e são estudados aspectos referentes ao *Bearer* externo. Para isto são diferenciados dois tipos de comunicações segundo seu destino: tráfego entre diferentes domínios móveis do SP, e tráfego com destino fora do SP. Para isto, é subdividido o *Bearer* externo em dois *Bearers*, o que no trabalho foi denominado *Transport-Bearer* (composto pelo tramo dentro do domínio de transporte do SP), e o *Bearer* externo, que é definido para o tramo fora do SP (não só fora do domínio móvel). Com esta mudança em consideração é construída a Figura 4.9, que permite visualizar a composição dos *Bearers* no caminho fim a fim para dois casos específicos: o caso onde o destino fica fora do SP e o caso onde existe uma comunicação entre dois domínios móveis do mesmo SP (interconectados mediante a rede de transporte).

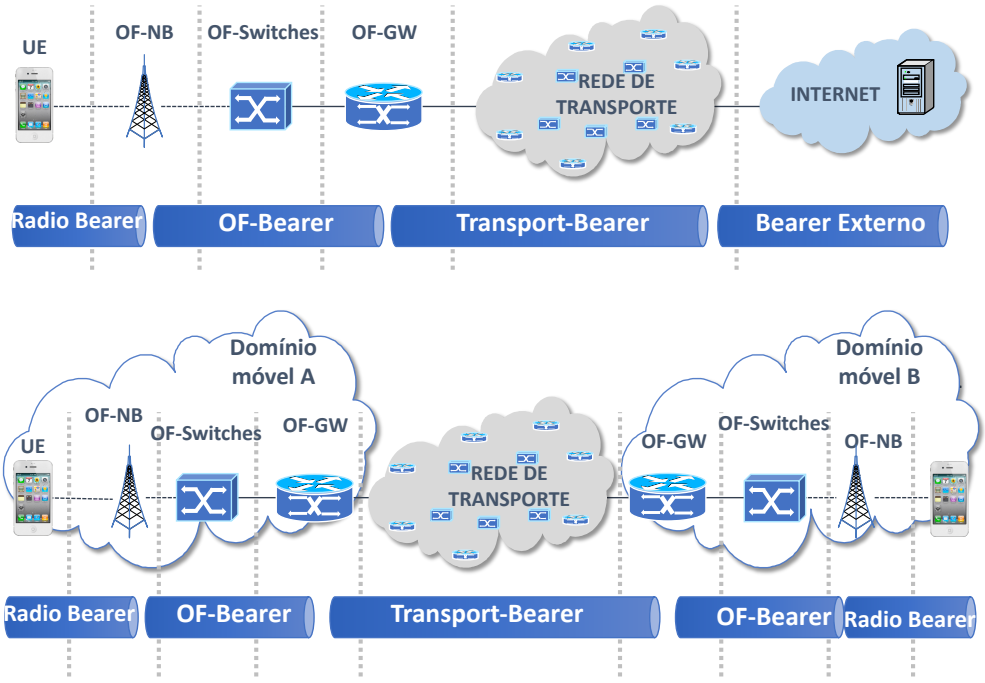

Figura 4.9: Composição dos *Bearers* que integram o caminho fim a fim para dois casos de interconetividade

É importante notar que se é utilizada a rede de transporte construida no Capítulo 3, os requisitos de QoS no *Transport-Bearer* já podem ser automáticamente cumpridos, isto mantendo ainda a independência entre domínios. Neste caso, o único que é requerido é o estabelecimento do campo DSCP apropriado quando a rede móvel envia os pacotes à rede de transporte. Entretanto, se acredita que podem ser realizadas optimizações para o caso em que a comunicação é estabelecida entre os domínios móveis do SP. Estas optimizações incrementam as possibilidades de definição

de serviços específicos, melhoram os mecanismos de restauração, e facilitam o gerenciamento da mobilidade entre domínios.

4.5.1 Otimização da interconexão entre domínios móveis do SP

Para cumprir com o objetivo, é aprofundado o conceito de que a rede de tranporte brinda serviços aos outros domínios de rede, e neste caso deve ser criado um serviço de transporte otimizado. Com esta finalidade são criados caminhos com requisitos de serviços diferenciados na rede de transporte que intercomuniquem os domínios móveis do SP (os OF-GW dos diferentes domínios). Para isto, são utilizados os conceitos de tunelização MPLS, e é realizado um acordo prévio entre os domínios móveis e o domínio de transporte para coordenar os rótulos que têm que ser colocados para os diferentes serviços (para o transporte dos diferentes tipos de tráfego). Na Figura 4.10 é mostrado um exemplo de como podem ser definidos alguns dos diferentes caminhos entre os OF-GW para dois tipos de tráfego. É importante notar que são os OF-GW os que conhecem as características dos serviços móveis a serem transportados, e por tanto são estes equipamentos os que colocam o rótulo MPLS que define o tipo de serviço de transporte e o OF-GW destino (definem o *Transport-Bearer* apropriado).

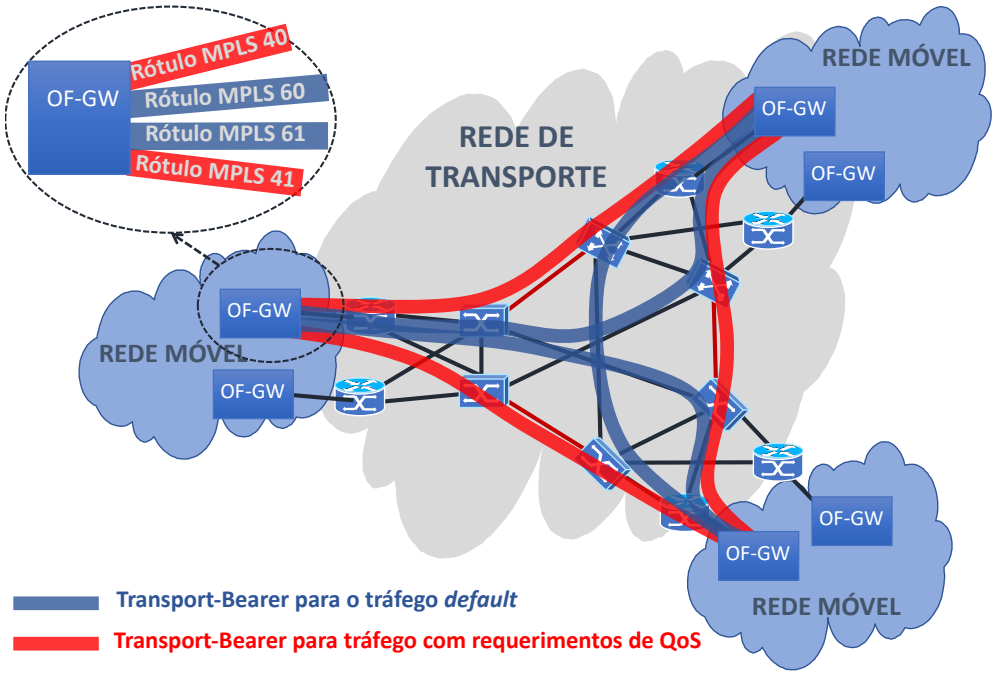

Figura 4.10: Definição de caminhos com requisitos de serviços de transporte diferenciados para a interconexão de domínios móveis do SP

Desta forma, a rede de transporte pode predefinir serviços para os domínios móveis de forma de brindar a priorização adequada, largura de banda, e mecanismo de restauração adequados para cada tipo de tráfego. Adicionalmente, este mecanismo permite conservar independência entre os diferentes domínios, devido a que só são requeridos acordos iniciais para definir os rótulos MPLS

para cada serviço. No transcurso da operação não é requerido nenhum tipo de interação entre os domínios móveis e o domínio de transporte. Desta forma, é adicionado um novo mecanismo para brindar QoS entre dois domínios móveis, o que permite garantir os requisitos de QoS fim a fim.

É importante notar que nesta alternativa, os domínios da rede móvel devem possuir a informação de onde se encontra o UE destino (dentro do domínio próprio ou dentro de outro domínio móvel do mesmo SP). Para isto, cada domínio móvel dispõe de seus prefixos IP que são utilizados para proporcionar os endereços IPv4 ou os prefixos IPv6 aos UE no decorrer do *attachment* inicial. Esta informação é conhecida pelos Controladores de cada domínio, sendo estes os responsáveis de inicialmente colocar as regras gerais nos OF-GW para alcançar os outros domínios utilizando os *Transport-Bearers* sem requisitos de QoS. Quando uma comunicação com requisitos de QoS (exemplo: IMS de VoIP) é estabelecida entre UEs de diferentes domínios móveis, no processo de criação de este *Bearer* específico tem que ser adicionada uma ligação entre o OF-*Bearer* com requisitos de QoS e o *Transport-Bearer* com os mesmos requisitos. Para isto, os respectivos Controladores de cada domínio colocam entradas de fluxo (nos respectivos OF-GW de seu domínio) que permitem encaminhar o tráfego ao respectivo UE destino utilizando o *Transport-Bearer* com requisitos de QoS apropriado.

4.5.2 Mobilidade entre domínios

A informação de encaminhamento para o tráfego sem requisitos de QoS que identifica a IP destino com o domínio móvel destino é relativamente estática. No trabalho, esta informação só muda quando os UEs se movimentam desde o seu próprio domínio móvel para outro domínio móvel do SP. Neste caso, os Controladores dos domínios móveis devem ser comunicados da mudança acontecida para manter a alcançabilidade a este UE particular.

Dentro de um domínio móvel, o fato de ter um UE com uma IP de outro domínio não requer de considerações especiais para que a IP possa ser alcançada localmente. As entradas de fluxo nos OF-GW já possuem regras específicas para atingir cada UE que se encontra no domínio. Para isto, as regras consideram a IPv4 ou o prefixo IPv6 específicos de cada UE, seja esta assignada dentro do domínio ou adquirida em outro domínio. É importante destacar que ainda quando a IP pertence aos prefixos IP definidos para esse domínio, estas entradas específicas nos OF-GW utilizam a IP destino como um dos campos de coincidência para encaminhar o tráfego pelo OF-*Bearer* apropriado até o OF-NB (ao que o UE se encontra conectado). É assim que quando um UE entra em outro domínio, têm que ser criadas iguais tipos de regras às utilizadas para o encaminhamento dos UEs de seu próprio domínio.

A principal diferença no encaminhamento quando o UE está localizado em outro domínio móvel do SP acontece no sentido *Downlink* para o tráfego que vem desde a rede de transporte do SP. Na proposta realizada no trabalho, a rede de transporte não deve conhecer informação de onde se encontra localizada a IP de um UE particular (nem deve conhecer os prefixos correspondentes a cada domínio móvel). O único que o domínio de transporte deve conhecer é quais prefixos IP pertencem a os domínios móveis e quais não. Neste caso, o tráfego entrante à rede de transporte (desde um domínio não móvel ou desde a própria Internet) com destino algum domínio móvel do SP é enca-

minhado ao domínio móvel mais próximo (ao OF-GW mais perto) utilizando encaminhamento por anycast. Uma vez em esse domínio, como os respectivos Controladores previamente configuraram nos OF-GW de seu dominio todas as regras de encaminhamento para todos os outros domínios móveis o tráfego é encaminhado aodestino utilizando o *Transport-Bearer* apropriado. Claramente, se o destino final já fosse o domínio móvel mais próximo o tráfego é encaminhado localmente utilizando o OF-*Bearer* e o *Radio Bearer* apropriado.

Este procedimento é possível devido a que os Controladores têram a informação de correspondência entre domínio móvel e prefixos IP, e conhecem os UE de seu domɲnio que se encontram fora do domínio (tem essa lista de IPs). Na Figura 4.11 se representa o caminho que tem que percorrer o tráfego desde Internet até o UE destino quando o UE não se encontra no domínio móvel mais próximo à entrada do tráfego à rede de transporte (caso tráfego sem requerimentos de QoS utilizando Transport-Bearer azul). Também é representado o caso em que o tráfego entrante tem como destino um UE que se encontra dentro do domínio móvel mais próximo (neste caso se representa um caminho com requisitos de QoS).

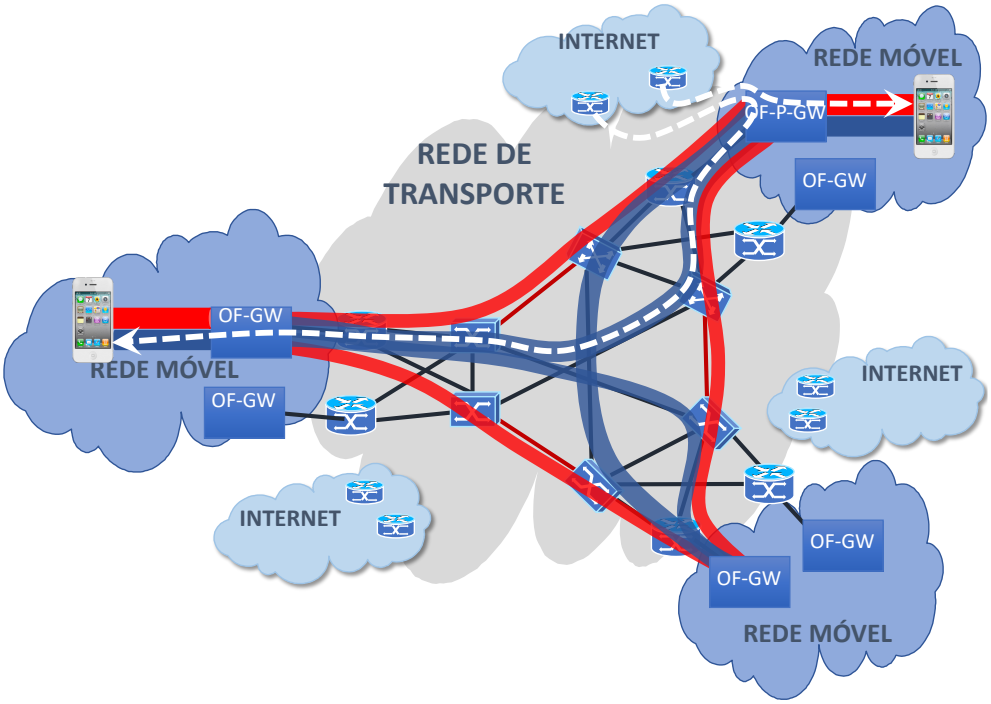

Figura 4.11: Encaminhamento do tráfego proveniente de Internet com destino domínios móveis do SP

É importante notar que esta alternativa evita quase completamente o problema da triangulação do tráfego nas redes móveis, onde o encaminhamento ao UE sempre tem que atravessar o domínio móvel que proporcionou a IP em primeira instância. Neste caso só é requerido o encaminhamento ao domínio móvel mais próximo, que seguidamente envia o tráfego ao domínio móvel destino utilizando os *Transport-Bearers* que garantem seus requerimentos. Adicionalmente, se reitera que não são requeridas informações de encaminhamento específicas dentro da rede de transporte; só é necessário conhecer se o endereço pertence a algum domínio móvel ou não. A informação de encaminhamento

dentro dos domínios móveis só deve ser conhecida pelos Controladores, que configuram as entradas de fluxo nos OF-GW para cumprir com o objetivo.

Para que a rede de transporte possa realizar o encaminhamento à rede móvel mais próxima existem diferentes alternativas. Se a rede de transporte é implementada com o controle distribuído padrão, os equipamentos de borda (OF-LER) da rede de transporte conetados a redes móveis (aos OF-GW) têm que redistribuir todos os prefixos móveis dentro da rede de transporte. Desta forma, é informado a todos os OF-LER da rede de transporte que os preficos móveis são atingidos por os OF-LER conetados aos domínios móveis. São as métricas do protocolo de encaminhamento que escolhem o domínio móvel mais próximo (conceito de *anycast* [93]). Neste caso se o domínio móvel mais próximo cair, o protocolo de encaminhamento escolhe o segundo domínio móvel mais perto para encaminhar o tráfego. No caso de que a rede de transporte seja implementada mediante SDN-*OpenFlow*, o Controlador é quem deve implementar esta lógica nos equipamentos de borda de entrada da rede de transporte.

No que refere à metodologia para o intercâmbio de informação entre os Controladores dos diferentes domínios móveis (para comunicar os UE que se encontram fora de seu próprio domínio) existem duas posibilidades. Uma delas é que o Controlador que recebe em seu domínio a um UE de outro domínio informe a todos os outros esta mudança. Outra posibilidade é que a mudança seja informada a dois elementos centrais que são responsáveis pela distribuição da informação para os outros Controladores. A segunda alternativa permite maior escalabilidade, mas depende da dimensão dos domínios móveis para determinar qual é a escolha apropriada. Na segunda alternativa é mencionada a utlização de dois elementos centrais para ter uma solução mais robusta, devido a que no caso de falhas em um deles, a informação continua sendo distribuída pelo outro.

4.6 Otimização para o encaminhamento com requisito de QoS nas comunicações internas a um domínio móvel

Nesta seção é realizada uma otimização do encaminhamento para o caso em que a origem e o destino do tráfego se encontre dentro do mesmo domínio móvel. Particularmente é considerada a otimização na construção dos *Bearers* com requisitos de QoS necessário para garantir os requisitos fim a fim na comunicação entre os UEs do mesmo domínio móvel.

No funcionamento descrito nas seções prévias, o tráfego com e sem requisitos de QoS é inicialmente encaminhado até o OF-GW apropriado utilizando os OF-*Bearers* correspondentes para cada tipo de tráfego. Para os casos onde o destino se encontra fora do domínio móvel, ou nos casos sem requisitos de QoS, este encaminhamento pode ser considerado apropriado, mas em casos que a comunicação requere QoS entre dois destinos do mesmo domínio móvel (exemplo, chamada de IMS de VoIP entre dois UE do domínio) podem ser realizadas otimizações que liberam recursos desnecessários e baixam o atraso fim a fim. Como pode ser observado na Figura 4.12, se não é realizada nenhuma otimização, quando um *Bearer* com requisitos de QoS é construído, este segue o caminho ótimo até o OF-GW apropriado, utilizando para isto o OF-*Bearer* com requisitos de QoS previamente configurado, e posteriormente é encaminhado pelo OF-*Bearer* com requisitos de

QoS até o UE destino.

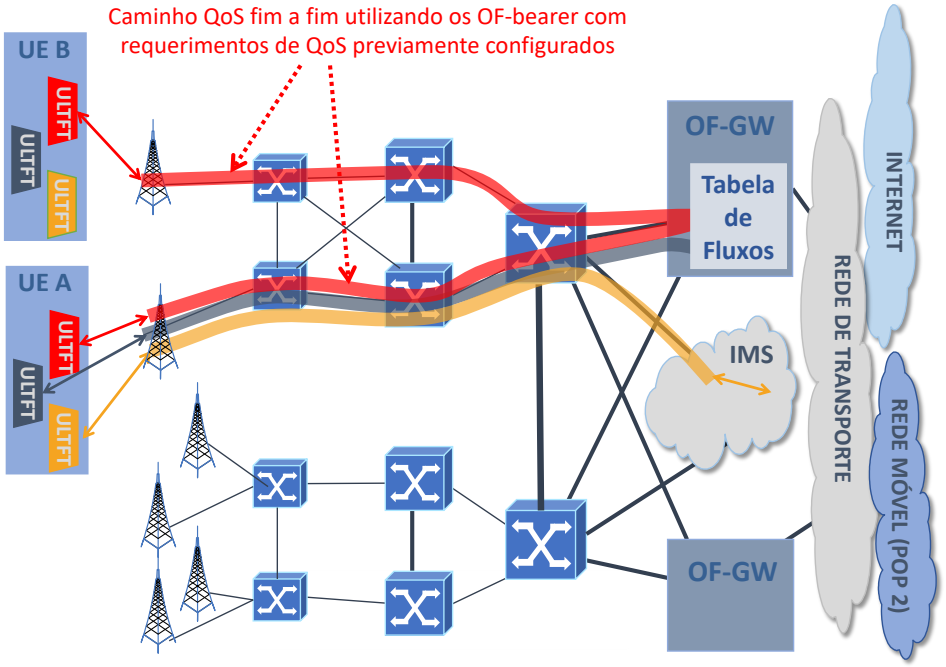

Figura 4.12: Caminho com requisito de QoS entre dois UE de um domínio móvel utilizando os OF-*Bearers* com requisitos de QoS previamente configurados

Para otimizar os caminhos utilizados para tipos de tráfego com requisitos de QoS dentro de um domínio móvel (como o tráfego de VoIP entre UE do domínio móvel), é necessário modificar o procedimento de criação dos *Bearer* com QoS. Para isto, quando o OF-NB realiza a detecção de sinalização, ele deve encaminhando os pacotes imediatamente pelo OF-*Bearer* sem requierimentos de QoS e deve enviar uma copia dos poacotes SDD ou ao Controlador. Entretanto, se o SDD/Controlador detecta que o tráfego tem requisitos de QoS e a origem e o destino estão dentro do domínio móvel, este pode construir um caminho ótimo na comunicação fim a fim sem a necessidade de utilizar um OF-GW. No caso concreto da Figura 4.13, o Controlador detecta que o caminho ótimo entre o OF-NB A e o OF-NB B não deve utilizar os OF-*Bearers* com requisitos de QoS previamente configurados, devido a que existe um caminho ótimo mais curto como o indicado na mesma Figura.

É importante notar que para a construção do caminho ótimo com requisitos de QoS, o Controlador deve criar as entradas de fluxo correspondentes em todos os equipamentos envolvidos no caminho fim a fim, e em ambas as direções (com os rótulos MPLS correspondentes). Adicionalmente, as entradas de fluxo devem ser criadas na direção oposta ao sentido do tráfego, sendo neste caso construído em última instância o *Radio Bearer* com requisitos de QoS e a entrada de fluxo que liga este *Radio Bearer* com o caminho ótimo criado. Desta forma, quando um novo *Radio bearer* com requisitos de QoS é utilizado, o caminho fim a fim já se encontra construído, evitando assim possíveis problemas no encaminhamento.

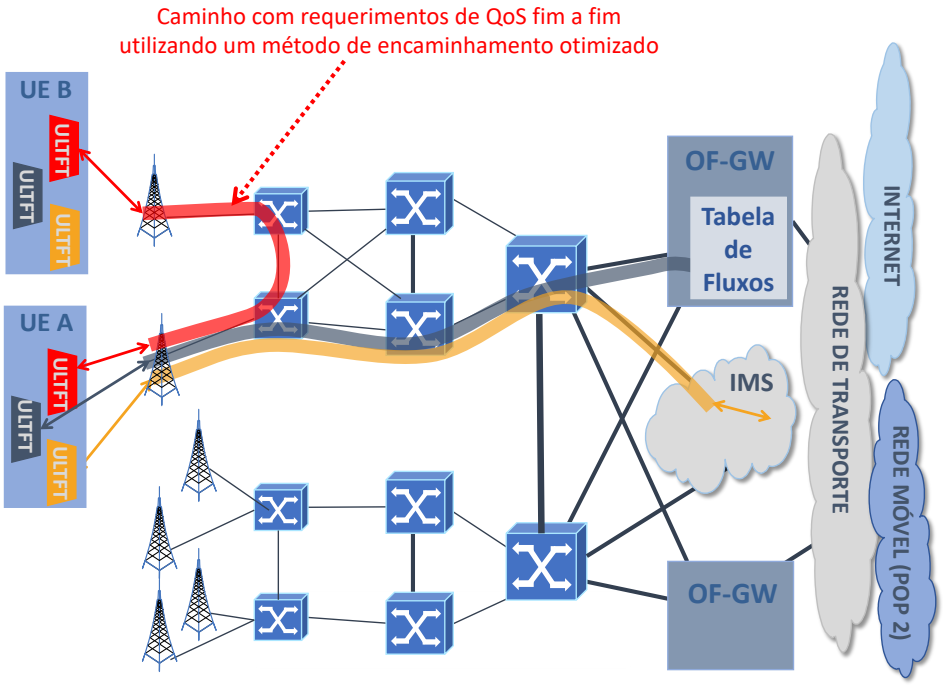

Figura 4.13: Caminho com requisito de QoS fim a fim interno ao domínio utilizando o método otimizado proposto

Se é comparado o novo caminho criado com requerimentos de QoS utilizando o método otimizado com o caminho que é construido utilizando os OF-*Bearers* com requisitos de QoS proativamente configurados, pode-se observar que:

- é requerido um menor número de equipamentos na construção do caminho;
- são utilizados menos recursos da rede para o encaminhamento. Particularmente se são requeridas garantias de largura de banda, esta reserva deve ser realizada em um menor número de equipamentos;
- o caminho obtido apresenta menores atrasos fim a fim;
- é diminuída a carga de processamento nos OF-GW, devido a que estes não têm que processar os novos *bearers* com requisitos de QoS criados com este método.

Capítulo 5

Protótipos para o domínio de transporte e para o domínio móvel de um SP

5.1 Protótipo para domínio de transporte de um provedor de serviço

5.1.1 Descrição geral

Para mostrar os benefícios que a arquitetura proposta proporciona, foi construido um protótipo com a arquitetura mostrada na Figura 5.1. Neste, todos os equipamentos implementam *OpenFlow* e MPLS em seu plano de dados. O protótipo está composto por três POP (*Point of Presence*), cada um com dois OF-LER ou equipamentos de núcleo (denominados SCxx), dois OF-LSR ou equipamentos de borda (denominados SBxx) e três *hosts*. Adicionalmente, os SCxx encontram-se conectados a outros três equipamentos de núcleo e aos SBxx de seu próprio POP.

Para a construção da topologia, foram utilizados os *softwares* descritos na seção 2.7, todos com suporte para OpenFlow 1.3 ou superior:

- Mininet [24] [25], capaz de criar uma topologia virtual completa, com centenas de *hosts* e *switches* interligados em um só computador. Adicionalmente, Mininet combina muitas das melhores características de emuladores, redes experimentais em *hardware* e simuladores;
- o Open vSwitch [27] [26]: software de *switch* multicamada construido para ambientes virtuais e com licença de código aberto apache 2.0;
- o Controlador Ryu [28]: Componente de *software* criado para o contexto das redes definidas por *software*, com a licença de código aberto apache 2.0. Este tem aPI (*application Programming Interface*) muito bem definidas, o que facilita a criação de novas formas de gerenciamento de redes e aplicações de controle para os desenvolvedores.

Os pacotes de tráfego gerados pelos *hosts* com endereço IP 10.x.1.2 (x variando entre 1 e 3, de acordo com o respectivo POP) são pacotes com requisitos de QoS que possuem o campo DSCP

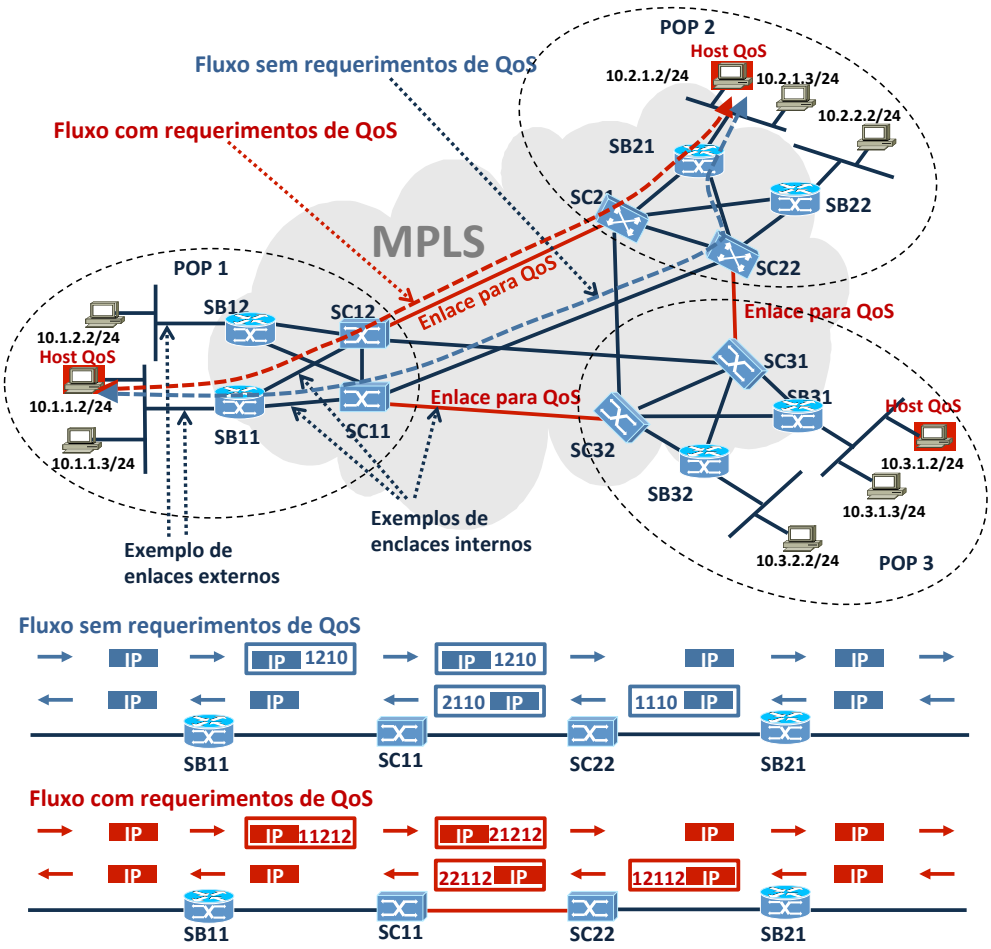

Figura 5.1: Topologia do protótipo OpenFlow-MPLS para domínio de transporte. Destacam-se fluxos com e sem requerimentos de QoS

estabelecido em 5 (prioritário). Quando um fluxo com requisitos de QoS específico é criado, este tráfego é encaminhado entre os diferentes POP através dos enlaces ilustrados como **Enlaces para QoS** na Figura 5.1. Por outro lado, o tráfego sem requisitos de QoS é encaminhado através dos outros enlaces. Todo o tráfego é encaminhado dentro do domínio de transporte utilizando MPLS, para isto os OF-LER de entrada (SBxx) adicionam o rótulo MPLS e mapeam o campo DSCP do pacote IP ao campo TC do cabeçalho MPLS. Com isto é mantida a diferenciação do tráfego durante o encaminhamento dentro do domínio de transporte (o que facilita as regras de classificação). Os SCxx encaminham o tráfego em função de este rótulo e do campo TC até o SBxx de saída, que finalmente encaminha o tráfego fora do domínio de transporte. No protótipo foi utilizada a técnica denominada PHP (*Penultimate Hop Popping*), que consiste em que o último dos SCxx (último OF-LSR) do caminho é o encarregado de remover o rótulo MPLS para diminuir o processamento requerido no SBxx de saída.

As entradas de fluxo dentro das tabelas são criadas com prioridade 32.768 (em caso de não especificar outro valor), e processadas na ordem que foram inseridas. Se for requerido que algumas das entradas de fluxo sejam examinadas antes de outras entradas, as primeiras devem ser estabelecidas com maior prioridade. No protótipo, as entradas de fluxo QoS específicas são criadas com

prioridade 45.000 para essa finalidade.

5.1.2 Construção das regras gerais

Todos os equipamentos contêm uma tabela de fluxos inicial indexada como 0, que encaminha o tráfego a uma lógica de gerenciamento específica. a tabela de fluxos geral é implementada pela tabela 10, e a tabela de fluxos QoS é implementada pela tabela 5. A numeração foi escolhida considerando o fato de que as entradas de fluxo dentro de tabelas com numeração inferior podem enviar o tráfego às tabelas de numeração superior, mas o inverso não é possível. Em nossa arquitetura, as entradas de fluxo na tabela de fluxos QoS devem enviar os pacotes à tabela de fluxos geral.

A seguir são descritas as regras que o Controlador adiciona inicialmente de forma proativa para implementar a lógica definida nas diferentes tabelas:

- **Tabela 0:**
  - Nos SBxx são adicionadas duas entradas de fluxo: uma para instruir que os pacotes entrantes marcados com DSCP = 0 sejam enviados à tabela 10 (tabela de fluxos geral), e outra para que os pacotes marcados com DSCP = 5 (que corresponde ao ToS = 20) sejam enviados à tabela 5 (tabela de fluxos QoS).
  - Nos SCxx são adicionadas duas entradas de fluxo: uma para instruir que os pacotes MPLS marcados com TC = 0 sejam enviados à tabela 10, e outra para que os pacotes MPLS marcados com TC = 5 sejam enviados à tabela 5 (tabela de fluxos QoS).
- **Tabela 10:**
  - Esta tabela contém entradas de fluxo gerais para encaminhar os pacotes a todas as redes destino (tabela similar à criada pelos protocolos de roteamento interno atuais). a tabela 10 encaminha independentemente do DSCP e do TC estabelecido.
- **Tabela 5:**
  - Nos SBxx são inseridas entradas de fluxo QoS gerais que só geram coincidência quando o pacote entra por uma interface externa (OF-LER de entrada). Esta regra envia o pacote por duas vias: ao Controlador (instrução) ***apply-actions***, porta reservada ***Controller***) e à tabela 10 (***Goto-Table 10***) para o encaminhamento imediato.
  - Também nos SBxx é inserida a entrada de fluxo QoS *default*, que no caso de não ter coincidência com as entradas de fluxos QoS gerais (ou com entradas de fluxo QoS específicas), envia o pacote somente para a tabela 10.
  - Nos SCxx, somente é adicionada a entrada de fluxo QoS *default*. A entrada indica que os pacotes sejam encaminhados à tabela 10 até que o Controlador adicione as entradas de fluxo QoS específicas (solicitação realizada pelo SBxx de entrada para esse fluxo).

5.1.3 Modo de operação e regras de QoS específicas

A seguir, é descrito o modo de funcionamento:

- Os fluxos sem QoS (DSCP = 0 ou TC = 0) são processados em cada equipamento inicialmente pela tabela 0, que os envia à tabela 10 para o seu encaminhamento, conforme ilustrado na Figura 5.2, opção 2 (OP. 2).

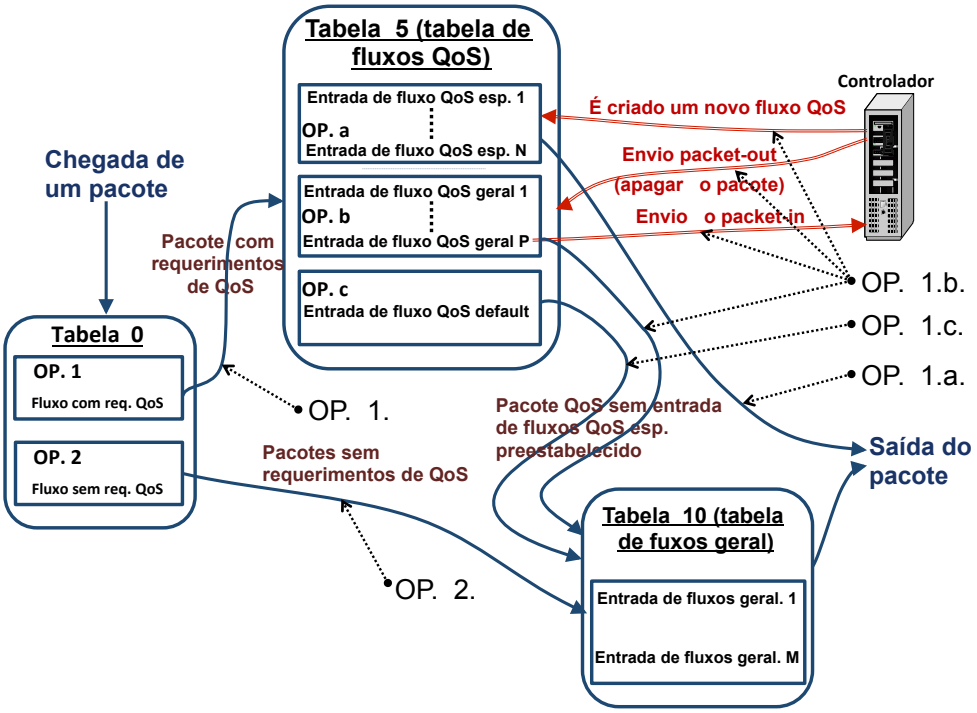

Figura 5.2: arquitetura do protótipo no caso de um SBxx (OF-LER)

- Os fluxos com requisitos de QoS (DSCP = 5 ou TC = 5) são processados em cada equipamento pela tabela 0, que os envia à tabela 5, conforme ilustrado na Figura 5.2, opção 1 (OP. 1).

- Nos equipamentos SBxx, quando ainda não existem entradas de fluxo QoS específicas para o pacote QoS entrante, e houver coincidência com entradas de fluxo QoS gerais (caso que seja um SBxx de entrada ao domínio, ver Figura 5.2, op. 1.b), o SBxx envia uma mensagem *packet-in* ao Controlador e encaminha imediatamente o pacote pela tabela 10. O Controlador recebe a mensagem *packet-in*, calcula o caminho ótimo e cria as entradas de fluxo QoS específicas com *idle-timeout* apropriado (para que sejam apagadas automáticamente quando as entradas de fluxo QoS específicas não sejam utilizadas) e prioridade 45.000 nas tabelas 5 de todos os equipamentos intervenientes no caminho fim a fim (as entradas de fluxo são criadas em sentido oposto ao sentido do fluxo). Adicionalmente, o Controlador envia um *packet-out* como resposta ao SBxx que realizou a solicitação, indicando que não deve encminhar o pacote que gerou a solicitação (ele já foi encaminhado pela tabela 10).

- Nos equipamentos SBxx, quando não existir ainda entrada de fluxo QoS específica configu-

rada, e não houver coincidência com entrada de fluxos QoS geral (caso de SBxx de saída ver Figura 5.2, OP. 1.c.), o SBxx envia o pacote só à tabela 10.

- Nos equipamentos SCxx, quando ainda não existirem fluxos QoS específicos, o pacote é sempre encaminhado à tabela 10.
- Finalmente em qualquer equipamento, no caso de existir entrada de fluxo QoS específica para o pacote entrante (Figura 5.2, OP. 1.a.), o pacote é encaminhado por essa entrada (caminho ótimo).

5.1.4 Cenários de testes

A Figura 5.3 ilustra as tabelas de fluxo reais dos equipamentos SB11, SC11 e SC12 (com suas entradas de fluxo reagrupadas para facilitar a visualização). Estas foram obtidas na realização do teste descrito na seção 5.1.5, e utilizadas nesta seção para mostrar exemplos precisos da forma como são implementados os diferentes tipos de entradas de fluxo em um caso real.

- Exemplos de entradas de fluxo iniciais (tabela 0): No nó SB11 as entradas de fluxo 1 e 2 da Figura 5.3, a decisão de envio é realizada segundo o campo DSCP. No nó SC11 as entradas de fluxo 1 e 2 da Figura 5.3, a decisão é realizada utilizando o campo TC-DSCP.
- Exemplos de entradas de fluxos gerais (tabela 10): No nó SB11 a entrada 4 da Figura 5.3, o cabeçalho MPLS é adicionado. No nó SC11 a entrada de fluxo 3 da Figura 5.3, o rótulo MPLS é trocado de 1.210 para 2.210. No nó SC11, a entrada de fluxo 2 da Figura 5.3, é realizado o POP do cabeçalho MPLS e enviado ao SB11. É importante ressaltar que o SC11 é o penúltimo *hop* no domínio MPLS quando o fluxo tem como destino a rede 10.1.1.0/2 (no protótipo é utilizada a técnica PHP como foi indicado na seção 5.1.1, por isto é o SC11 quem remove o cabeçalho MPLS).
- Exemplos de entradas de fluxos QoS gerais com interface de entrada externa (tabela 5): No nó SB11, na entrada de fluxo 3 da Figura 5.3, o pacote é encaminhado ao Controlador (*actions* = *CONTROLLER*) e enviado à tabela 10 simultâneamente (onde o cabeçalho MPLS é adicionado e encaminhado imediatamente). Este fluxo é considerado entrante ao domínio devido a que utiliza a interface de entrada externa 3 (indicada como in_ port = 3 na entrada de fluxo) que interconecta o SB11 com o *host* com endereco IP 10.1.1.2 (origem do tráfego). A entrada é utilizada para o tráfego com requisitos de QoS desde o *host* 10.1.1.2 para o *host* 10.2.1.2 durante o periodo de tempo que não existe uma entrada de fluxo QoS específica para esse tráfego (é utilizado o caminho indicado como **Fluxo sem requierimentos de QoS** da Figura 5.1).
- Exemplos de entradas de fluxo QoS *default* (tabela 5): No nó SB11, a entrada de fluxo 5, tabela 5 da Figura 5.3, é o caso de um equipamento SB11 de saída. No nó SC11, a entrada de fluxo 1, tabela 5 da Figura 5.3 (equipamento interno), e o caso de entrada de fluxo QoS *default* que encaminha à tabela 10 utilizando como critério de coincidência somente o TC = 5.

- Exemplos de entrada de fluxo QoS específicas (tabela 5): No nó SB11, a entrada de fluxo 1, tabela 5 da Figura 5.3, representa uma entrada QoS específica criada dinámicamente pelo Controlador, que utiliza o caminho ótimo **Enlace com requisitos de QoS** (o cabeçalho MPLS é adicionado). No nó SC12, a entrada de fluxo 1, tabela 5 da Figura 5.3 representa uma entrada de fluxo QoS específica em um equipamento interno (o cabeçalho MPLS é trocado salto a salto).

Por razões de implementação, todos os testes foram realizados com equipamentos puramente *OpenFlow*, mas os resultados são exatamente os mesmos que os obtidos se a tabela 10 dos equipamentos fosse implementada com o controle distribuído atual (arquitetura híbrida). Em esse caso, só é requerida a utilização de equipamentos híbridos que permitam substituir todas as instruções do tipo *Goto-Table*=10 pelas instruções do tipo *Write-action* onde o *action-set* é definido com uma ação do tipo *output* à porta reservada *Normal*.

5.1.5 Teste para a simulação de uma queda do Controlador

Este teste foi iniciado com todas as entradas de fluxo QoS específicas apagadas e com todas as entradas proativas previamente configuradas. Nesta situação inicial, é simulada uma queda do Controlador, e durante esta queda um fluxo com requerimentos de QoS de ida e volta foi gerado entre os *hosts* com endereços IP 10.1.1.2 e 10.2.1.2. Durante a queda, o Controlador não pode criar a entrada de fluxo específica e o trajeto utilizado é o denominado **Fluxo sem requisitos de QoS** (ver Figura 5.1). Mais tarde o Controlador é restabelecido e uns segundos depois o fluxo com requerimentos de QoS é detido, finalizando assim o teste. Quando o Controlador é restabelecido, este cria as entradas de fluxo QoS específicas em cada equipamento do caminho ótimo (em ambas as direções) e o tráfego é reencaminhado utilizando o caminho QoS específico (denominado **Fluxo com requerimentos de QoS** na Figura 5.1). O resultado do teste pode ser observado nas tabelas de fluxo reais dos equipamentos SB11, SC11 e SC12 (Figura 5.3). Seguidamente é descrito este exemplo com a profundidade requerida para sua total compreensão.

Neste exemplo, 80 pacotes QoS (todos com DSCP = 5) são gerados (40 na direção SB11⟶SB21 e 40 na direção SB21⟶SB11), os quais encontram coincidência com a entrada de fluxo 1, da tabela 0 da Figura 5.3, que envia o pacote à tabela 5. Dos 80 pacotes que compõem o teste, 43 são enviados durante a queda do Controlador (22 na direção SB11⟶SB21, e 21 na direção SB21⟶SB11). Os outros 37 pacotes restantes (18 na direção SB11⟶SB21, e 19 na direção SB21⟶SB11) são enviados na segunda parte do teste, quando o Controlador se encontra restabelecido.

É importante notar que o pacote 23 na direção SB11⟶SB21, e o pacote 22 na direção SB21⟶SB11 são os primeiros pacotes que têm resposta do Controlador, que cria as respectivas entradas de fluxo QoS específicas em todos os equipamentos envolvidos no caminho ótimo (um caminho QoS ótimo por cada direção). Entretanto, esses dois pacotes que geram a primeiras consultas exitosas ao Controlador, uma para cada direção, ainda continuam sendo processados pelas entradas de fluxo QoS gerais com interface de entrada externa no SB11 e no SB21, respectivamente (em esse preciso momento a entrada de fluxo QoS específica continua sem ser criada pelo

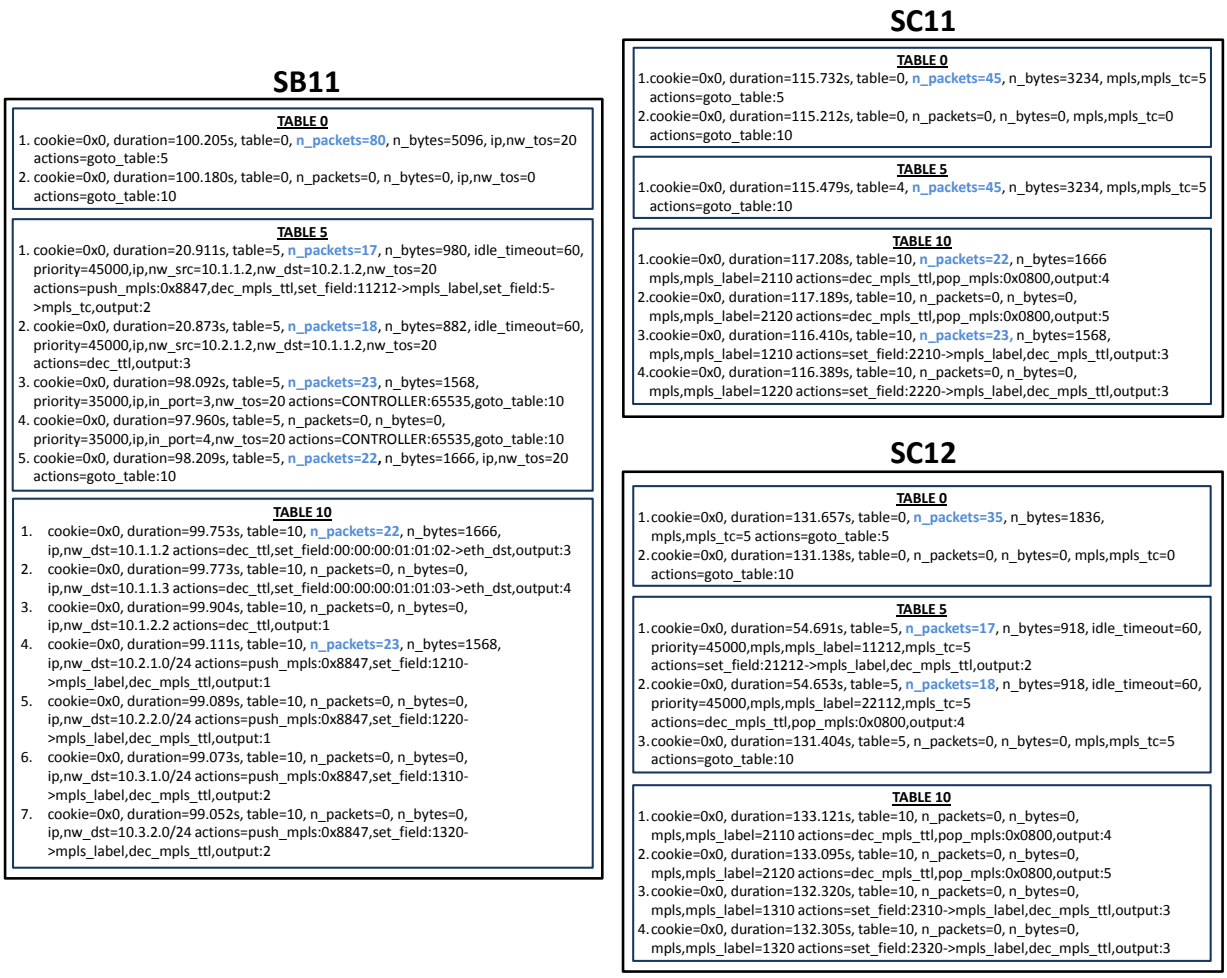

Figura 5.3: Tabelas de fluxo para os equipamentos SB11, SC11 e SC12 reagrupadas para facilitar a visualização

Controlador).

Como resultado do anteriormente descrito, os primeiros 23 pacotes na direção SB11⟶SB21 são processados pela entrada de fluxo QoS geral com interface de entrada externa (Figura 5.3, tabela 5, entrada de fluxo 3). Esta entrada de fluxo QoS geral envia o pacote simultâneamente por duas linhas de processamento ao Controlador (*actions* = *CONTROLLER*) e à tabela 10 (*goto_ table*:10), onde encontra coincidência com a entrada de fluxo geral (entrada de fluxo 4, da tabela 10 da Figura 5.3) e são executadas as ações descritas a seguir: o cabeçalho MPLS é colocado com rótulo MPLS 1.210, o IP-DSCP é mapeado ao MPLS-TC, e o pacote é enviado ao SC11. No SC11 a tabela 0 envia o pacote até a tabela 5 e esta até a tabela 10 (ver Figura 5.3, entrada de fluxo 3, da tabela 10), onde o rótulo MPLS é trocado pelo 2.210 e os pacotes são enviados ao SC22. Finalmente o SC22 envia o pacote até o SB21 que envia o pacote até o *host* destino.

Por outro lado, os 22 primeros pacotes na direção SB21⟶SB11 são processados pelo SB21 de forma similar à que foi descrita no parágrafo anterior, e envia o pacote ao SC22 (indicado como fluxo sem requisitos de QoS na Figura 5.1), quem a sua vez envia o pacote ao SC11. Como o SC11 é o penúltimo *hop* dentro do domínio MPLS, o pacote encontra coincidência com a entrada de fluxo 1, da tablela 10, da Figura 5.3, e remove o rótulo MPLS e envia o pacote ao SB11. Finalmente, no SB11 o pacote é processado pela entrada de fluxo QoS *default* (Figura 5.3, tabela 5, entrada de fluxo 5), devido a que o SB11 é um equipamento de saída para o fluxo na direção SB21⟶SB11. Esta entrada de fluxo só envia o pacote à tabela 10, onde finalmente encontra coincidência com a entrada de fluxo 1, da Figura 5.3, e envia o pacote ao *host* destino.

Os restantes 17 pacotes na direção SB11⟶SB21, e os 18 pacotes na direção SB21⟶SB11 já têm configurada a entrada de fluxo QoS específica pelo Controlador, em cada uma das tabelas de fluxos QoS dos equipamentos envolvidos no caminho ótimo. Assim, os pacotes são diretamente encaminhados utilizando estas entradas de fluxo QoS específicas (ver **fluxo com requerimentos de QoS** indicado na Figura 5.1).

No SB11 na direção ⟶SB21, o pacote é inicialmente processado pelo SB11 e agora encontra coincidência com a entrada de fluxo 1, tabela 5 da Figura 5.3, onde são realizadas as ações descritas a seguir: o cabeçalho MPLS é inserido com rótulo 11.212, o MPLS-TC é mapeado desde o campo IP-DSCP, e o pacote é enviado ao SC12 diretamente por esta entrada. No SC12 a tabela 0 envia o pacote à tabela 5 (ver Figura 5.3, entrada de fluxo 1, tabela 5), trocando o rótulo MPLS ao 21.212 e envia o pacote ao SC21. O SC21 envia o pacote ao SB21, quem finalmente envia o pacote ao *host* destino (encaminhado como fluxo com requisitos de QoS na Figura 5.1).

Finalmente, na direção SB21⟶SB11, os últimos 18 pacotes finais são inicialmente processados pelo SB21 de forma similar à descrita no parágrafo anterior, e são enviados ao SC21, quem envia o pacote ao SC12. Como o SC12 é o penúltimo *hop* no domínio MPLS, o pacote encontra coincidência com a entrada de fluxo 2, da tabela 5 Figura 5.3, quem remove o cabeçalho MPLS, e o pacote é enviado ao SB11. Finalmente no SB11 o pacote encontra coincidência com a entrada de fluxo com requisito de QoS (*flow entry* 2, da tabela 5 da Figura5.3) e o pacote é enviado ao *host* destino.

O test confima a robustez da arquitetura SDN-OpenFlow-MPLS proposta, e comprova como o pacote continua sendo encaminhado ainda com uma queda do Controlador. Adicionalmente,

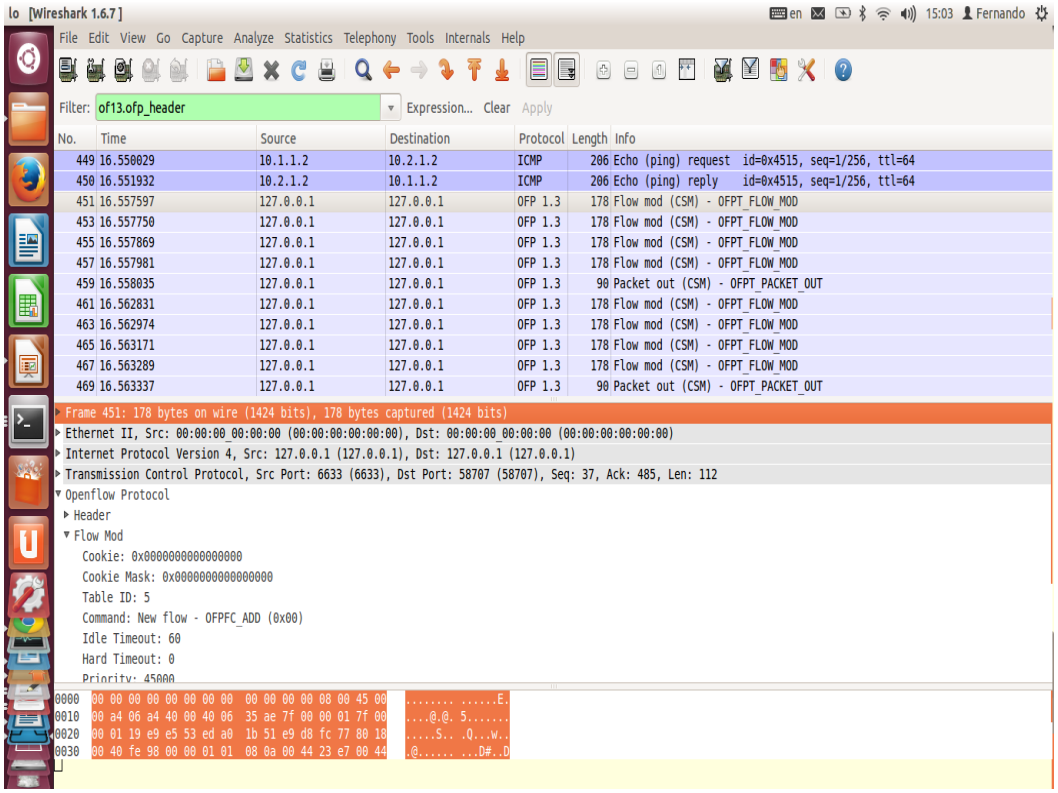

Figura 5.4: Mensagens trocadas na criação de um novo fluxo QoS específico

quando o Controlador é restabelecido, as entradas de fluxo QoS específicas são criadas, e o fluxo é reencaminhado por estas entradas sem perdas no processo. Finalmente, é confirmado o imediato encaminhamento de todos os pacotes ainda durante uma queda do Controlador e é confirmado o importante fato de que o Controlador não adiciona tempos de espera na criação de novos fluxos.

Finalmente a Figura 5.4 mostra uma janela do analisador de protocolo *Wireshark* com uma captura realizada, onde podem ser observadas duas mensagens *Packet-in*, que representam a solicitação de um novo fluxo em ambas as direções: uma no sentido de ida, gerada pelo equipamento SB11, e outra no sentido de volta, gerada pelo equipamento SB21 (Figura 5.1). Posteriormente são mostradas quatro mensagens *Flow-mod*, que são as mensagens utilizadas pelo Controlador para adicionar ou alterar fluxos. Estas mensagens são enviadas a cada um dos equipamentos que se encontram dentro do caminho ótimo de ida, para assim configurar o fluxo QoS (no exemplo: SB11, SC12, SC21 e SB21. Figura 5.1).

Posteriormente, é enviada uma mensagem *Packet-out* pelo Controlador para indicar que o equipamento que gerou a consulta (SB11) tem que ignorar o pacote, dado que este já foi encaminhado imediatamente por uma entrada de fluxo geral. A seguir, são enviadas outras quatro mensagens pelo Controlador, necessárias para configurar o fluxo específico QoS para o caminho de volta (SB21, SC21, SC12 e SB11). Finalmente, é enviada uma mensagem *Packet-out* pelo Controlador ao equipamento que gerou a consulta de volta (SB21), indicando-lhe que o pacote tem que ser ignorado, devido as que este também, neste caso, foi encaminhado por uma entrada de fluxo geral.

5.1.6 Teste para a simulação de tempos de resposta elevados do Controlador e problemas de sobrecarga na rede

Com a finalidade de obter resultados quantitativos, é implementada uma arquitetura de rede de transporte do SP na mesma topología mas, neste caso utilizando equipamentos com *OpenFlow* padrão, tendo sido comparado seu comportamento com a arquitetura do domínio de transporte proposta no Capítulo 3. Com ambas as arquiteturas implementadas foram simulados diversos problemas que podem existir como:

- atraso por enfileiramento no Controlador.
- sobrecarga e tempos de resposta elevados do Controlador.
- requisitos de retransmissões TCP nas mensagens *packet-in*.
- atrasos no caminho utilizado pela mensagem *packet-in* (devido a problemas nos links ou nós envolvidos).

Os testes consistem na adição de atrasos variáveis para emular os problemas mencionados anteriormente, e medir o tempo de ida e volta dos pacotes entre os *hosts* com endereços IP 10.1.1.2 e IP 10.2.1.2 (ver Figura 5.1). É importante notar que os dois primeiros problemas mencionados afetam o tempo de ida do pacote e de sua resposta, mas o terceiro e quarto dos problemas mencionados poderiam afetar só o caminho da mensagem *packet-in* (se a resposta é encaminhada por uma ruta que não utiliza os nós ou enlaces afetados). Nestes testes são consideradas as duas alternativas.

Os testes são realizados com atrasos adicionados no rango de 0 ms a 100 ms e com incrementos de 10 ms. Todos os testes são realizados para a arquitetura proposta para rede de transporte de SP, e para a arquitetura utilizando *OpenFlow* padrão. Assim mesmo, foram considerados os dois tipos de atrasos, tanto no que afeta à mensagem *packet-in* quanto o que afeta à mensagem *packet-in* e sua resposta. Os resultados são mostrados na Figura 5.5, e como poderia ser previsível, a arquitetura proposta para a rede de transporte tem um comportamento totalmente independente em relação ao atraso adicionado em todos os casos.

Outra consideração importante é que ainda com zero adição de atrasos, a arquitetura utilizando *OpenFlow* padrão, apresenta um atraso total de 19 ms comparada com a arquitetura proposta no trabalho (ver Figura 5.5). Este atraso representa dois tempos de espera pela resposta do Controlador às mensagens *packet-in* (uma em cada direção). Estes 19 ms são inferiores ao caso real devido a que no protótipo utilizado, os nós, os links e o Controlador são implementados no mesmo computador. Se é considerada a presença de uma rede entre o Controlador e os nós, os 19 ms do gráfico são incrementados sensívelmente, deslocando os gráficos do caso de arquitetura com *OpenFlow* padrão para cima, incrementando ainda mais os atrasos com relação à arquitetura proposta de rede de transporte para SP.

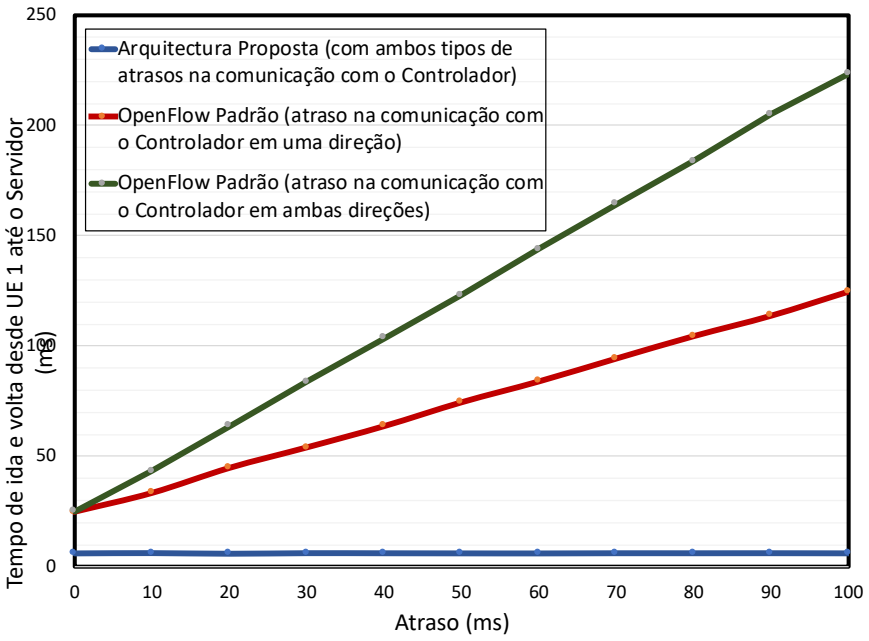

Figura 5.5: Atrasos totais no encaminhamento de pacotes de ida e volta para a arquitetura de rede de transporte proposta e para a arquitetura implementada utilizando *OpenFlow* padrão, considerando atrasos na comunicação

5.1.7 Teste com e sem mecanismos PIDAM

O mecanismo PIDAM foi implementado no protótipo [55] para melhorar ainda mais o comportamento da proposta da arquitetura. Este mecanismo é implementado nos *switches* emphOpenFlow para que só o primeiro pacote de um novo fluxo com requerimentos de QoS gere uma mensagem *packet-in* para o Controlador. Para isto são modificados os *switches* emphOpenFlow para que esses mantenham uma lista interna de todos os fluxos que aguardam resposta do Controlador. Sempre que chegar um novo pacote com requerimentos de QoS a um *switch*, este compara suas características com as dos pacotes da lista, e só gera um *packet-in* se não encontra coincidência (o que reduz ainda mais o processamento do mesmo).

O PIDAM foi implementado em *OpenVSwitches* modificando o procedimento de geração do *packet-in*. Assim, foi criada uma lista com o cabeçalho de todos os pacotes que geraram uma mensagem *packet-in* e que ainda aguardam a resposta do Controlador. Sempre, antes de gerar uma mensagem *packet-in*, o cabeçalho do pacote é comparado com os cabeçalhos do pacote armazenado na lista, e o *packet-in* é enviado somente se não é encontrada nenhuma coincidência.

O teste consiste em cinco estágios, nos quais pacotes com QoS são enviados desde todos os *hosts* com QoS (h11, h21 e h31) até todos os *hosts* com QoS (h11, h21, h31) simultâneamente:

- Estágio 1, pacote com carga útil de 64 bytes
- Estágio 2, pacote com carga útil de 128 bytes,

- Estágio 3, pacote com carga útil de 256 bytes,
- Estágio 4, pacote com carga útil de 1512 bytes,
- Estágio 5, pacote com carga útil de 5000 bytes.

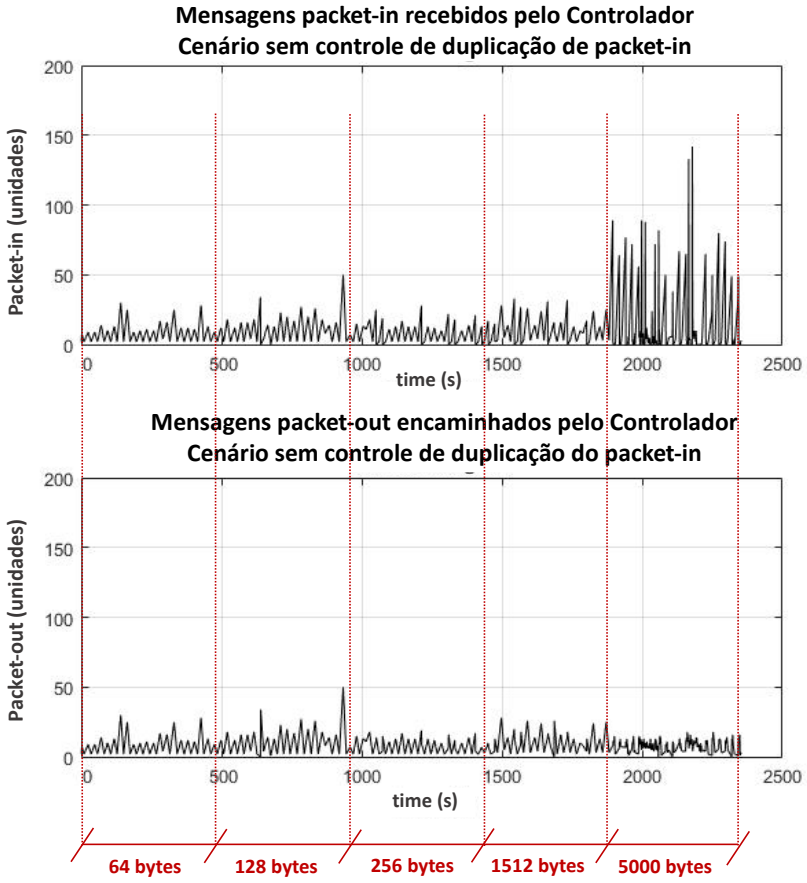

Figura 5.6: Mensagens *packet-in* e *packet-out* intercambiadas utilizando a arquitetura proposta (sem o controle de duplicação de Packet-in PIDAM).

Cada estágio consiste na execução de vinte etapas, e em cada uma, todos os *hosts* com QoS enviam 1.000 pacotes ICMP (com separação de 10 ms entre eles) para os outros *hosts* com QoS. Cada vez que uma etapa é concluída, as entradas de fluxo QoS específicas são apagadas e se passa à seguinte etapa. Foi decidido que sejam vinte etapas repetidas em cada estágio para melhorar a visualização dos resultados. Em seguida, é mostrada a rotina de execução no *host* h11 durante o estágio 1.

$i = 0$
while $((20 > i))$
do
ping 10.2.1.1 -*s* 64 -*i* 0,01 -*c* 1000
ping 10.3.1.2 -*s* 64 -*i* 0,01 -*c* 1000

execute ClearFlowTable
let i++

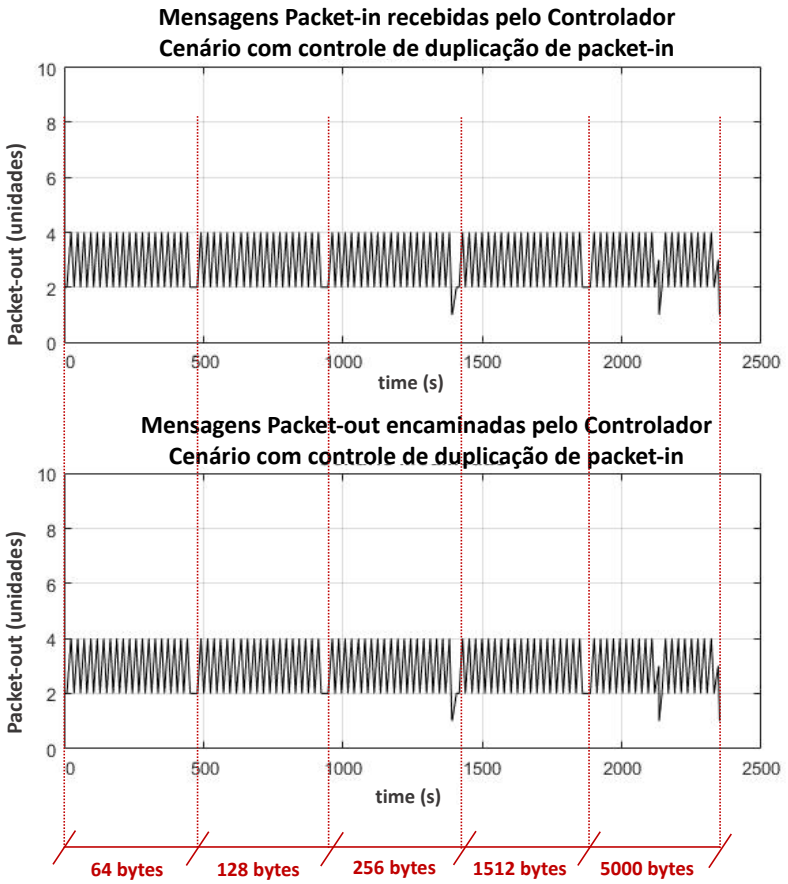

Figura 5.7: Mensagens *packet-in* e *packet-out* intercambiadas utilizando a arquitetura proposta (com o controle de duplicação de packet-in PIDAM).

Foi realizada uma comparativa do desempenho da arquitetura proposta com e sem o mecanismo PIDAM. Em ambos os casos, os pacotes com QoS não precisam esperar a resposta do Controlador para ser encaminhados. Ainda antes de que o Controlador crie a entrada de fluxo QoS específica, o pacote com requerimentos de QoS é enviado por duas linhas de processamento simultâneamente.

As Figuras 5.6 e 5.7 mostram os *packet-in* e *packet-out* intercambiados durante os diferentes estágios de teste com e sem os mecanismos PIDAM. Como pode ser visto, os diferentes comportamentos são particularmente observados durante o estágio 5. Se os pacotes de QoS têm 5000 bytes de carga útil, ele é fragmentado em quatro pacotes consecutivos (muito próximos uns dos outros), e cada fragmento gera uma mensagem *packet-in* incrementando o processamento do Controlador. Como pode ser observado na Figura 5.6, no estágio 5 quando o mecanismo PIDAM não é usado, nem todas as mensagens *packet-in* têm uma resposta *packet-out* do Controlador.

Adicionalmente, a Figura 5.8 mostra que o tempo de resposta do Controlador aumenta significativamente no estágio 5 quando o mecanismo PIDAM não é usado. Ao incrementar o tempo

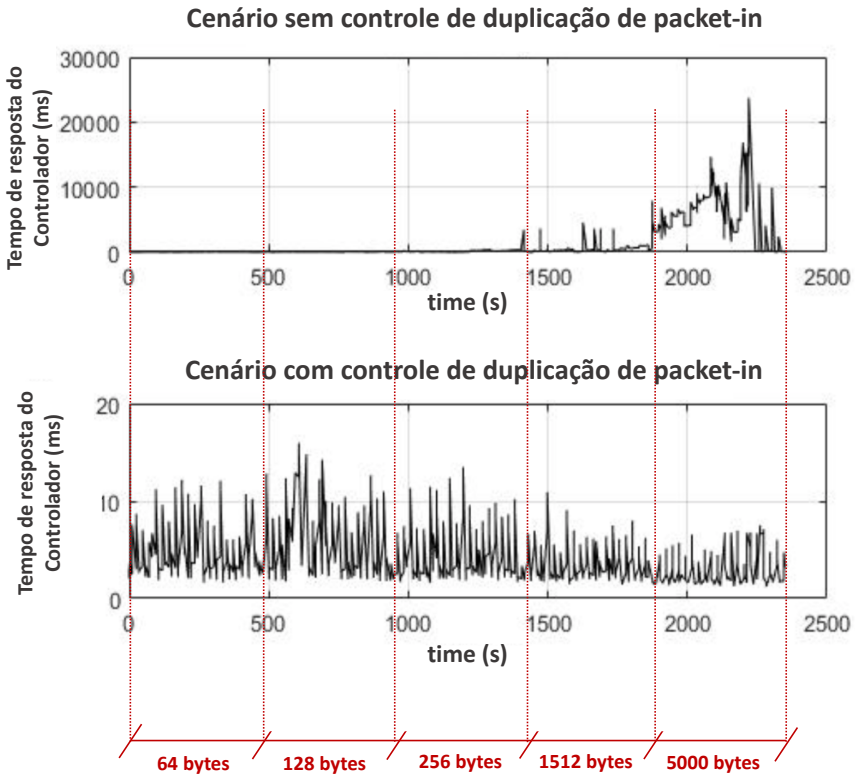

Figura 5.8: Tempo de resposta do Controlador para a arquitetura proposta com e sem o mecanismo PIDAM

de resposta, mais mensagens *packet-in* são enviadas para o Controlador, devido a que este tarda mais tempo na criação da entrada de fluxo QoS específica. Esse problema é resolvido quando o mecanismo PIDAM é usado, porque somente o primeiro pacote de cada novo fluxo de QoS gera uma solicitação *packet-in*. Em nosso exemplo, apenas seis pacotes em cada etapa de teste geram um *packet-in*, um para cada fluxo de QoS.

É importante ressaltar que se a arquitetura proposta sem o mecanismo PIDAM for usada, mesmo na fase 5, os pacotes continuam sendo enviados imediatamente usando a lógica geral de gerenciamento. Por outro lado, se durante o estágio 5 fosse utilizada a arquitetura padrão *OpenFlow* haveria um importante problema no comportamento devido a que os pacotes não seriam encaminhados (se deveria aguardar tempos de resposta do Controlador muito elevados).

5.1.8 Comparativo das arquiteturas para o domínio de provedor de serviço

Através da análise do Capítulo 3 e dos resultados obtidos no presente capítulo é construída a tabela 5.1 que mostra uma comparativa entre quatro alternativas de implementação da rede de transporte de um SP: as redes atuais, as redes implementadas com *OpenFlow* padrão, a arquitetura proposta com equipamentos híbridos (seção 3.3) e a arquitetura proposta com equipamentos

Tabela 5.1: Comparativo das arquiteturas para rede de transporte de SP

	Rede atual	***OpenFlow* padrão**	**Proposta Híbrida**	**Proposta MPLS-*OpenFlow***
Resposta ao encaminhamento	Imediata	Intermédia	Imediata	Imediata
Possibilidades de engenharia de tráfego	Boa	Muito Boa	Muito Boa	Muito Boa
Jitter para tráfego QoS	Meio	Baixo	Baixo	Baixo
Comportamento no caso de quedas do Controlador	N/A	Ruim	Ótimo	Muito Bom
Comportamento ante atrasos na resposta do Controlador	N/A	Ruim	Ótimo	Ótimo
Carga no Controlador	N/A	Elevada	Intermédia	Intermédia
Complexidade nos protocolos de roteamento	Muito alta	Intermédia	Muito alta	Intermédia
Carga por processamento nos equipamentos de rede	Elevada	Baixa	Elevada	Baixa

OpenFlow (seção 3.2).

Para a arquitetura de rede de transporte *OpenFlow* proposta, destaca-se:

- a robustez da rede ante a queda do Controlador;
- a robustez da rede devido a atrasos na resposta do Controlador;
- a diminuição da carga exigida ao Controlador em razão de:
  - redução das mensagens que são enviadas ao Controlador para seu processamento (só tráfego com requisitos de QoS);
  - só o primeiro pacote de cada fluxo QoS realiza uma consulta ao Controlador;
  - só os OF-LER enviam mensagens *packet-in* solicitando a criação de um novo fluxo ao Controlador;
- a rápida resposta ao encaminhamento de pacotes de novos fluxos sem adicão de atrasos;
- a eliminação do plano de controle distribuido nos equipamentos.

Para a arquitetura de rede de transporte proposta com equipamentos híbridos, destaca-se:

- a robustez da rede devido a atrasos na resposta do Controlador;
- a robustez da rede ante a queda do Controlador;

- a robustez da rede ante mudanças topológicas durante uma queda do Controlador;
- a diminuição da carga exigida ao Controlador em razão de:
  - redução das mensagens enviadas ao Controlador para seu processamento (tráfego com QoS);
  - só o primeiro pacote de cada fluxo QoS realiza uma consulta ao Controlador;
- a rápida resposta ao encaminhamento de pacotes pertencentes a novos fluxos.

5.2 Protótipo para domínio móvel de um provedor de serviço

5.2.1 Implementação e descrição geral

Como prova de conceito, foi implementado o protótipo mostrado na Figura 5.9. Este está constituído por nove elementos móveis implementados como *switches OpenFlow*. Dois deles representam OF-GW, dois representam OF-NB; e os sete restantes representam *switches* de *backhaul* (SW1 - SW7). adicionalmente, 4 VMs (*Virtual Machines*) são criadas, três representando UEs e a quarta representando um servidor que fornece serviços para UEs que requerem tratamento especial de QoS.

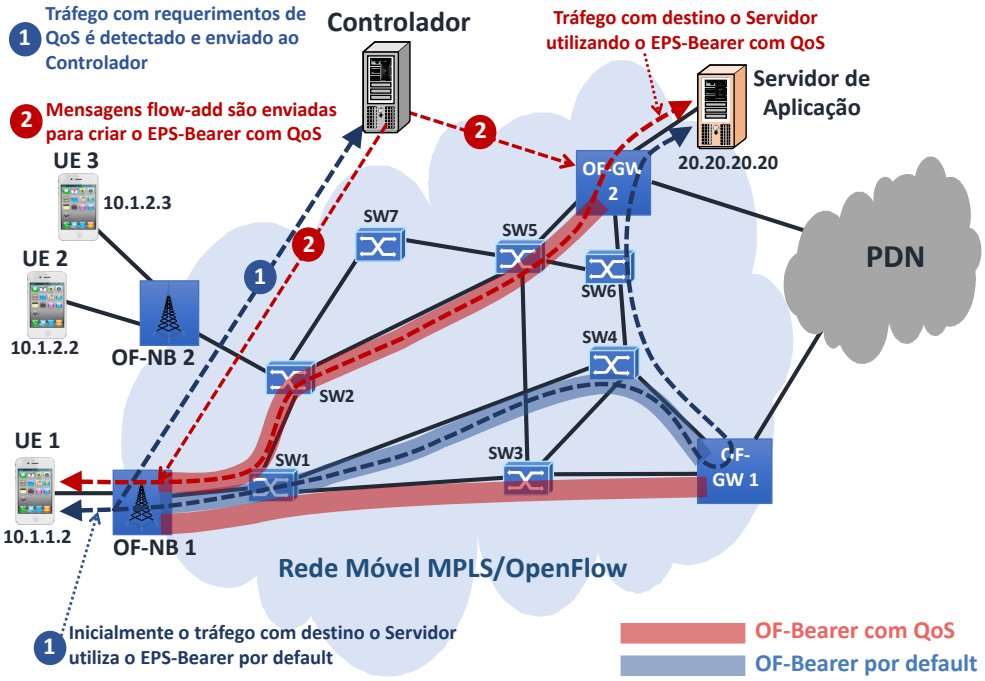

Figura 5.9: Protótipo: Exemplo 1, criação de EPS-Bearer com QoS ao servidor de aplicação

Para a construção da topologia é utilizado Mininet, enquanto para a implementação dos OF-NB e *switches* é utilizado Open vSwitch, e como Controlador foi escolhido Ryu (todos com suporte do *OpenFlow 1.3* ou superior).

O protótipo implementa a lógica de detecção de tráfego e a criação do EPS-Bearer com QoS. Além disso, a arquitetura é implementada para o *backhaul* e o núcleo da rede móvel (a implementação do acesso móvel está além do escopo deste trabalho). Por fim, o Controlador é implementado como o orquestrador, incorporando também as funcionalidades de elementos como PCRF e SDD.

Como resumo, o protótipo mostra as características da solução indicadas a seguir:

- a flexível na operação da lógica de detecção proposta, fornecendo exemplos de entradas de fluxo que a compõem.
- a criação proativa de regras de encaminhamento gerais que fazem possível o envio imediato do tráfego utilizando EPS-Bearer por *default* (sem a interação do Controlador).
- a simultaneidade da solução no que refere ao roteamento do tráfego e da inspeção do tráfego em procura de requerimentos de QoS. Em particular, é descrito um exemplo de entrada de fluxo, que envia o tráfego com requerimentos de QoS por duas linhas de processamento..
- é indicada a criação pelo Controlador de um EPS-Bearer com QoS, e como o tráfego com esses requerimentos é reencaminhado usando-o (são fornecidos exemplos de entrada de fluxo para serviço específica).
- a proposta usa MPLS e o protocolo GTP-U não é necessário.
- é mostrado como todos os elementos móveis do plano de dados (como OF-GW, OF-NB e *switches*) são implementados utilizando *OpenFlow*, pelo que qualquer um deles pode ser configurado para a função que seja requerida. Isto faz também que os OF-*Bearers* criados possam ser implementados desde qualquer um de estes elementos construindo caminhos ótimos (ver Figura 5.10).

Conforme descrito nas seções anteriores, o Controlador cria inicialmente as entradas de fluxo geral para todos os OF-GW, de forma que possam alcançar os outros elementos móveis, os servidores de aplicação, e a saída da rede móvel (PDN). Logo, o Controlador cria o OF-*Bearer* por *default* que permitem que os OF-NB possam alcançar o correspondente OF-GW. Também são criados os OF-*Bearers* com QoS que podem ser utilizados para serviços massivos entre os OF-NB e OF-GW (usando os rótulos MPLS apropriados). Finalmente, são construídas as tabelas de detecção de tráfego para todos os perfis de usuários em todos os OF-NBs.

Os exemplos indicados nas seguintes seções começam com os UEs *attached* aos OF-NB e com os EPS-*Bearers* por *default* já criado. No caso do UE 1, ambas são criadas, tanto a entrada de fluxo por *default* para o UE 1 no OF-NB 1 (para o sentido *uplink*) quanto a entrada de fluxo por *default* para o UE 1 no OF-GW (para o sentido *downlink*). Estas entradas de fluxo são as responsáveis por ligar o *Radio Bearer* por *default* do UE 1 com o OF-*Bearer* por *default*, construindo assim o EPS-*Bearer* por *default* para o UE 1.

5.2.2 Detecção de tráfego e criação de EPS-Bearer com QoS ao servidor de aplicação

A Figura 5.9 mostra o OF-*Bearer* por *default* entre o OF-NB 1 e OF-GW 1 (utilizando os *switches* SW1 e SW4) e dois OF-*Bearers* com QoS entre o OF-NB 1 e os dois OF-GW (para *serviços* massivos com requisitos de QoS proativamente criados). O OF-*Bearer* com QoS entre OF-NB 1 e OF-GW 2 representa um caminho ótimo com atrasos mínimos para que OF-NB 1 possa alcançar o servidor de aplicação.

Inicialmente, o tráfego desde o UE 1 para o servidor de aplicação encontra coincidência com a entrada de fluxo para o UE 1 por *default* e segue inicialmente o caminho indicado na Figura 5.9. Um exemplo desse tipo de entrada de fluxo é mostrado a seguir:

*cookie=0x0, duration=35.568s, table=0, n_packets=0, n_bytes=0, ip,**nw_src=10.1.1.2***
*actions=**push_mpls:0x8847, set_field:1010102->mpls_label, push_mpls:0x8847,***
set_field:100001->mpls_label**, dec_mpls_ttl,output:1, **goto_table:15

Como pode ser observado, a entrada de fluxo envia o pacote por duas linhas de processamento: uma inserindo dois rótulos MPLS e enviando uma cópia desse pacote através da porta de saída 1, o que indica que debe ser utilizado o OF-*Bearer* por *default* para alcançar o servidor de aplicação (ver Figura 5.9); e a outra linha de processamento usa a ação *goto-table*, a qual encaminha o pacote à tabela 15, tabela de detecção de tráfego correspondente ao perfil do UE 1.

Quando o pacote chega à tabela de detecção de tráfego, ele encontra correspondência com uma entrada de fluxo para a detecção do serviço que utiliza como campo de coincidência o endereço IP destino do servidor de aplicação (IP 20.20.20.20):

*cookie=0x0, duration=42.918s, table=15, n_packets=0, n_bytes=0, **priority=3500**,*
*ip,nw_**dst=20.20.20.20 actions=CONTROLLER**:65535*

Como el SDD é implantado no Controlador, a entrada de fluxo para a detecção do serviço tem indicada a ação do tipo *output Controller*, o que encaminha o tráfego para o Controlador para a inspeção dos requerimentos de específicos.

O Controlador processa a mensagem *packet-in*, e determina que o caminho ótimo para este tráfego é utilizar o OF-*Bearer* con QoS entre o OF-NB 1 e o OF-GW 2. Seguidamente, o Controlador cria no OF-NB 1 e no OF-GW 2 as entradas de fluxo para o serviço específico, construindo assim o EPS-*Bearer* com QoS requerido (ver Figura 5.9). Finalmente, o tráfego com requisitos de QoS desde o OF-NB 1 e com destino o servidor de aplicação utilizará o caminho SW1 - SW2 -SW5 criado para este tipo de tráfego.

*cookie=0x0, duration=85.369s, table=0, n_packets=0, n_bytes=0, **priority=4000**,*
*ip,nw_**src=10.1.1.2**, ip,nw_**dst=20.20.20.20** actions=**push_mpls:0x8847,***

set_field:510102– >mpls_label, push_mpls:0x8847, set_field:50102– >mpls_label, *dec_mpls_ttl,* ***output:1***

O critério de coincidência que utiliza esta última entrada de fluxo considera tanto o endereço IP origem do UE1 quanto o endereço de destino do servidor (outros critérios mais restritivos também podem ser considerados). Além disso, a entrada de fluxo adiciona dois rótulos MPLs para identificar unívocamente o EPS-*Bearer* com QoS. O rótulo interior 510102 identifica o UE 1 / *Radio Bearer* com QoS, e o rótulo externo 50102 identifica o OF-*Bearer* com QoS. Por fim, é importante observar que as entradas de fluxo para o serviço específico têm maior prioridade (4000) que os outros tipos de entradas de fluxo, por tanto, esse tipo de tráfego sempre utilizará a entrada específica de tráfego para o encaminhamento.

A flexibilidade desta implementação permite criar EPS-Bearers específicos que podem permitir comunicações diretas com um atraso mínimo entre os UEs. Como um exemplo, pode ser criado um EPS-Bearer com QoS específico para a comunicação entre o UE 1 e o UE 2, que só requere utilizar o caminho SW 1 e SW2 (ver Figura 5.10).

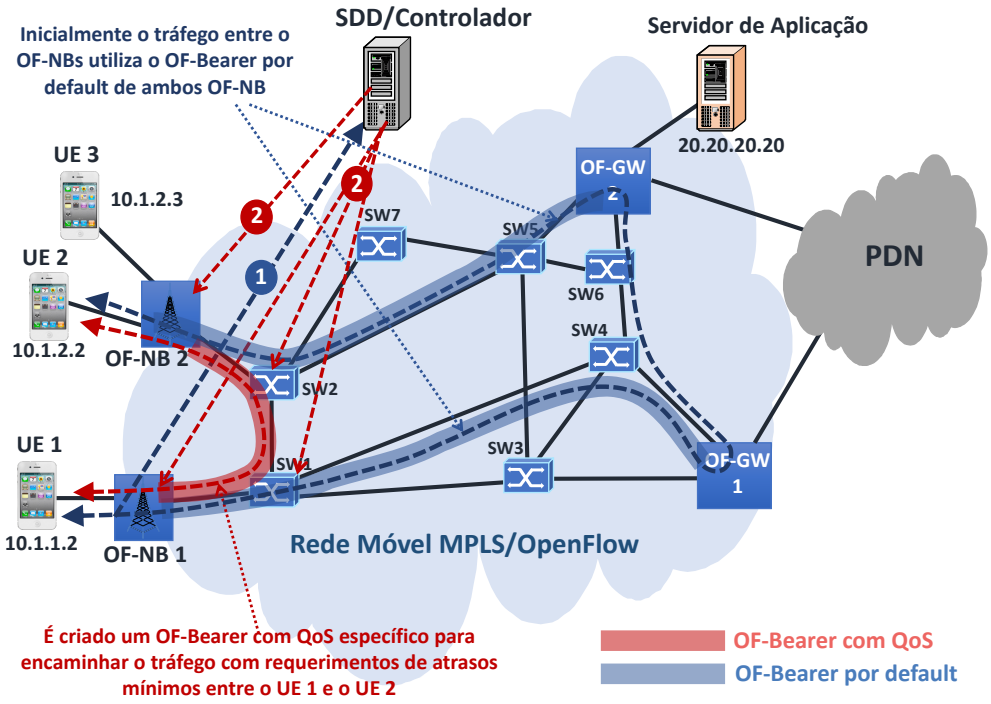

Figura 5.10: Protótipo: Exemplo 2, criação de EPS-*Bearer* com mínimo atraso entre o UE 1 e o UE 2

5.2.3 Comparativo de estratégias de detecção de tráfego

Em seguida, é feita uma comparação qualitativa de diferentes estratégias usadas para detectar tráfego com requisitos de QoS em redes móveis. As duas primeiras usam estratégias da arquitetura EPS atual; a terceira é baseada em NFV onde os elementos lógicos móveis são implementados de

forma centralizada em centro de dados; e a última é implementada utilizando a lógica de detecção de tráfego de QoS proposta.

Tabela 5.2: Comparativa de estratégias de métodos para a detecção de tráfego com requerimentos de QoS

	Mecanismo AF	**Mecanismo TDF**	**DataCenter centralizado**	**Mecanismo proposto**
Robustez	Baixa	Baixa	Boa	Muito boa
Escalabilidade	Boa	Baixa	Muito boa	Muito boa
Processamento de QoS	Distribuído	Centralizado	Centralizado	Distribuído/ Centralizado
Possibilidades de detecção do tráfego	Boa	Boa	Muito boa	Muito boa
Adição de retraso na detecção de tráfego com QoS	Baixa	Baixa	Intermédia	Baixa

- Mecanismo baseado na técnica de detecção de AF: Refere-se ao mecanismo das redes 4G que usam dispositivos de detecção chamados AF, que são colocados em pontos estratégicos da rede. Esses pontos estratégicos de localização são escolhidos de forma que o tráfego a ser inspecionado tenha que atravessar o AF. Quando os requerimentos de QoS são detectados pelo AF, estes são enviados para um dispositivo central (como o PCRF) responsável pela implementação da política de QoS. Os dispositivos AF geralmente são colocados dentro ou antes dos servidores de aplicação e são especializados em detecção de tráfegos específicos. Essa alternativa diminui a flexibilidade e as possibilidades de programação, pois outros tipos de tráfego podem ser roteados usando caminhos diferentes. Por outra parte, os AF podem ser situados de forma distribuída, e este mecanismo gera tempos de espera adicionais no processo de detecção de QoS baixos.

- Mecanismo baseado em filtros TDF: Este método é implementado colocando elementos de detecção dentro de elementos que centralizam o plano de dados como o PGW na arquitetura EPS atual. Como todo o tráfego passa pelo PGW, esse método poderia analisar todos os tipos de tráfego, mas uma inspeção aprofundada dos pacotes poderia comprometer o desempenho do PGW. Nesta alternativa, a grande quantidade de tráfego que deve ser processado e a natureza centralizadora da solução pode impactar significativamente na escalabilidade e robustez da solução.

- Mecanismos de inspeção centralizados em *DataCenters*: Vários trabalhos propuseram [13] a implementação dos elementos do plano de dados e de controle da rede móvel como VNF, implementadas como VM em *hardware* padrão, colocados em centros de dados. Nesta alternativa, todo o tráfego é encaminhado para um centro de dados e é processado em procura dos requisitos de QoS. Esta alternativa gera um atraso adicional devido à necessidade do

encaminhamento do tráfego para o *DataCenter* e requere de um processamento significativo para a inspecção de tráfego. Por outro lado, como pode ser desenvolvido num ambiente de computação em nuvem, isto incrementa a robustez e confiabilidade da solução, mesmo ainda quando é implementado de forma centralizada num DataCenter, a afirmação deve-se a que o ambiente de nuvem é muito elástico e capaz de se adaptar às variações do tráfego utilizando grupos de servidores.

- Mecanismo de detecção proposto: Essa alternativa é capaz de encaminhar os pacotes com requerimentos de QoS imediatamente como tráfego sem requerimentos, e ao mesmo tempo, envia uma cópia de pacote com QoS ao elemento SDD centralizado para que sejam detectados requerimentos de QoS (não é adicionado tempo de espera no encaminhamento inicial). Esta opção incrementa a robustez da solução dado que o tráfego com QoS pode ser encaminhado ainda quando o Controlador tem tempos de resposta elevados ou durante o não funcionamento do SDD (encaminhado como tráfego sem QoS). A escalabilidade da solução deve-se a que a funcionalidade de detecção encontra-se distribuída em todos os OF-NB e em os SDD implementados. Alem disso, esta alternativa incrementa as possibilidades de detecção de QoS, combinando a detecção de primeiro nível nos OF-NB com a de segundo nível nos SDD, este último é um elemento programável capaz de detectar todos os tipos de tráfego que foram requeridos. Finalmente, esta solução de detecção pode ser considerada mais genérica que a alternativa anterior, dado que a tabela de detecção de tráfego (no OF-NB) para o UE pode ser modificada (ainda em tempo real) para que todo o tráfego seja encaminhado para u SDD ou um centro de dados (como a alternativa de detecção anterior).

Um resumo qualitativo das diferentes alternativas de detecção de tráfego pode ser encontrada na tabela 5.2.

Tabela 5.3: Comparativa de arquiteturas de redes móveis.

	arquitetura EPC	**OpenFlow Padrão**	**arquitetura Proposta**
Tempo de resposta ao envio	Imediato	Intermédio	Imediato
Possibiliades de engenharia de tráfego	Boas	Muito boas	Muito boas
Requerimento do protocolo GTP-U	Requerida	Requerida	Não requerida
Flexibilidade para a deteção do tráfego com requirimentos de QoS	Boa	Boa	Muito boa
Tratamento específico para o tráfego com QoS	Imediato	elevado	elevado
Processamento requerido no Controlador	N/a	Muito elevado	Intermédia

5.2.4 Comparativo de arquiteturas móveis

Na tabela 5.3 é mostada uma comparativa entre a arquitetura móvel implementada utilizando a arquitetura EPS das redes 4G atuais; utilizando uma implementação com conceitos centralizados de SDN *OpenFlow* padrão onde o tráfego com requerimentos de QoS deve aguardar a resposta do Controlador para encaminhar os pacotes; e utilizando a arquitetura proposta.

5.2.5 Simulação de tempo de resposta elevados do Controlador e sobrecarga na rede

Com a finalidade de obter resultados comparativos qualitativos foi implementado também sobre o protótipo, o controle *OpenFlow* padrão, comparando o comportamento referente ao encaminhamento de tráfego com requerimentos de QoS para as duas arquiteturas: a arquitetura proposta e a implementada com o controle *OpenFlow* padrão. é importante notar que a alternativa com *OpenFlow* padrão implementada utiliza métodos de estabelecimento de caminhos proativos para a configuração inicial e para as regras de encaminhamento de tráfego sem QoS, e utiliza métodos reativos para estabelecer caminhos para tráfego com requerimentos de QoS. Por tanto, quando pacotes com requerimentos de QoS chegam na arquitetura padrão, uma mensagem *packet-in* será gerada pelo OF-NB, e este aguardará a resposta do Controlador para encaminhar os pacotes com QoS.

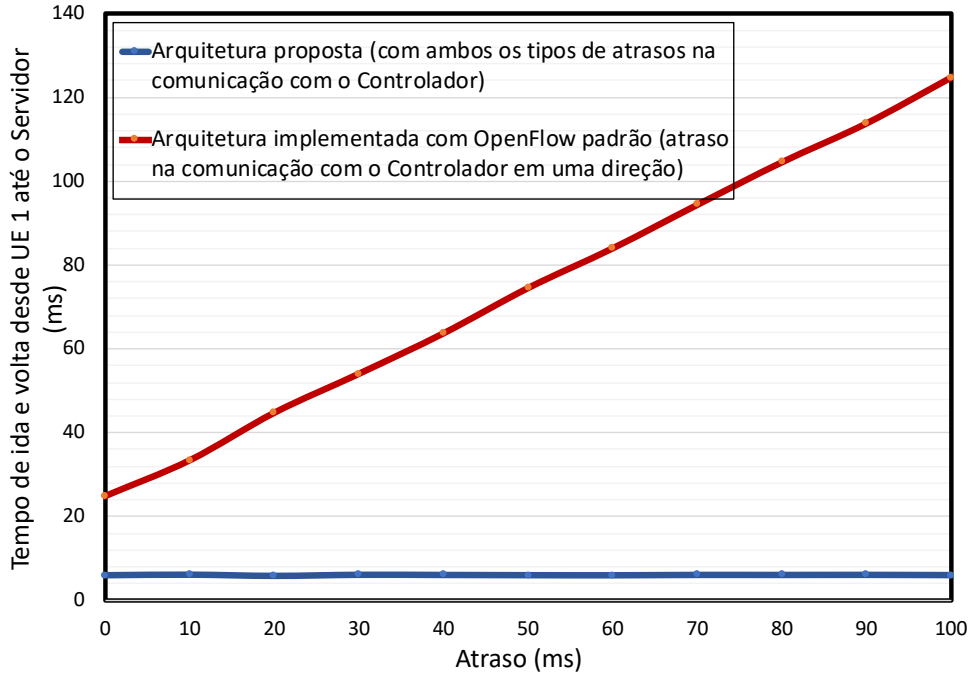

Figura 5.11: Comparativa da arquitetura proposta e a implementada com *OpenFlow* padrão quando o Controlador apresenta tempos de resposta elevados

Com a arquitetura proposta e a implementada com *OpenFlow* padrão, foram emulados proble-

mas na comunicação com o Controlador, e foi comparado o comportamento das duas arquiteturas quando um fluxo de pacotes com requerimentos de QoS específicos é encaminhado desde o UE 1 até o servidor de aplicação. Para o teste foram simulados problemas na comunicação dos OF-NB e OF-GW com o Controlador.

O teste considera que o UE 1 encontra-se *attached*, e todas os OF-*Bearers* com QoS encontram-se apagados (todas as entradas de fluxo para serviços específicas são apagadas). Em esta situação é gerado tráfego com requerimentos de QoS de ida e volta desde o UE 1 até o servidor de aplicação com IP 20.20.20.20 (ver Figura 5.9), e é medido o tempo de ida e volta. O teste é realizado tanto para a arquitetura proposta quanto para a arquitetura com *OpenFlow* padrão, e o teste é repetido incrementando o atraso na comunicação entre o OF-NB 1 e OF-GW 1 com o Controlador. O atraso adicional introduzido encontra-se no rango desde 0 ms até 100 ms, com incrementos de 10 ms.

Os resultados são indicados na Figura 5.11. Como pode ser observado, o tempo de ida e volta entre o UE 1 e o servidor de aplicação não é afetado quando é utilizada a arquitetura proposta. Isto deve-se a que os pacotes com requerimentos de QoS são sempre imediatamente encaminhados sem a necessidade de aguardar a resposta do Controlador. Os pacotes são inicialmente encaminhados utilizando o OF-*Bearer* por *default* pré-configurado durante o *attachment* do UE. Por outra parte, se é utilizada a arquitetura implementada com *OpenFlow* padrão, o tempo de ida e volta é linearmente afetado na medida que é incrementado o atraso na resposta do Controlador.

Capítulo 6

Conclusões

Os resultados deste trabalho podem ser divididos em duas partes, uma com relação à arquitetura proposta para rede de transporte de SP, e a outra para a arquitetura proposta para rede móvel de SP. Cada uma delas com suas características, objetivos e requisitos específicos, sendo necessário o estudo separado dos resultados obtidos para cada uma das arquiteturas propostas.

6.1 Conclusões da proposta de arquitetura para a rede de transporte do SP

.

Durante o transcurso do trabalho foram construídas e verificadas (por médio de testes no protótipo) as diferentes caraterísticas que a arquitetura proposta SDN-*OpenFlow*-MPLS apresenta. Esta permite melhorar diversos pontos fracos que as implementações com *OpenFlow* padrão possuem, particularmente aspectos relacionados à falta de robustez da rede devido à forte dependência dela em relação ao Controlador. Verificou-se como a arquitetura proposta continua em funcionamento ainda durante uma queda do Controlador, durante tempos de resposta elevados, ou no caso que aconteçam perdas das mensagens *OpenFlow* devido a falhas nos links ou equipamentos intervenientes. Além disso foi verificado por meio de testes no protótipo que a arquitetura proposta não adiciona tempos de espera na criação de novos fluxos, devido a que os primeiros pacotes de cada fluxo com requisitos QoS são encaminhados imediatamente pela tabela de fluxos geral (e paralelamente processados pelo Controlador). Posteriormente, quando o Controlador cria uma entrada de fluxo QoS específica em cada equipamento interveniente no caminho fim a fim dentro da rede de transporte, o tráfego é reencaminhado sem tempos de espera nem perdas de pacotes no processo.

A arquitetura proposta reduz significativamente a quantidade de mensagens que o Controlador deve processar. Diferentemente das implementações *OpenFlow* padrão (em que todos os novos fluxos são encaminhados ao Controlador), na arquitetura proposta os fluxos sem requisitos de QoS são encaminhados pelas tabelas de fluxos geral construídas proativamente, as quais não geram consultas ao Controlador. Adicionalmente, foi proposto que só os equipamentos de borda de

entrada ao domínio SDN-*OpenFlow*-MPLS (os OF-LER de entrada) enviem a solicitação de novo fluxo com requisitos de QoS ao Controlador. Finalmente é proposta uma lógica de controle nos equipamentos de borda de entrada para que só o primeiro pacote de cada novo fluxo com requisitos de QoS gere uma consulta ao Controlador, reduzindo ainda mais as mensagens *OpenFlow* a serem processadas.

Diferentemente das redes atuais, a arquitetura proposta possibilita a criação de caminhos com requisitos de QoS de forma massiva e automática, sem a intervenção do administrador para a construção de cada um de estes caminhos. Com a utilização da tecnologia SDN-*OpenFlow*-MPLS é possível incrementar as possibilidades de engenharia de tráfego, facilitando o desenvolvimento de novos produtos, e proporcionando novas oportunidades de negócio. A arquitetura proposta utiliza protocolos abertos e padronizados o que incrementa a velocidade das inovações nestas redes, permitindo o envolvimento de todos os diferentes atores do negócio de forma colaborativa (pesquisadores, fabricantes, usuários, etc.). Adicionalmente, a utilização de protocolos abertos e padronizados permite que as inovações não estejam amarradas aos interesses de alguns de estes atores, e impede que a inteligência da rede se encontre oculta em equipamentos proprietários.

Como desvantagem menciona-se a necessidade de incorporar um novo dispositivo Controlador, que ainda para o caso da arquitetura proposta, representa um elemento centralizador que diminui em certa medida a robustez da rede. Concretamente, a arquitetura SDN-*OpenFlow*-MPLS proposta é capaz de dar suporte a quedas do Controlador, e a tempos de resposta elevados de este, mas apresenta inconvenientes se durante uma queda do Controlador acontece uma mudança topológica (em esse caso a topologia não pode ser atualizada).

Alternativamente foi proposta uma arquitetura para a rede de transporte constituída por equipamentos híbridos, que possibilita a construção das diferentes lógicas de gerenciamento opcionalmente mediante o controle *OpenFlow*, ou mediante o controle distribuído atual. Esta arquitetura mantém a característica de robustez das redes de transporte atuais, sendo capaz de realizar mudanças topológicas durante uma queda do Controlador. Além disso, a arquitetura híbrida facilita o processo de migração desde uma arquitetura com o controle distribuído à arquitetura SDN-*OpenFlow*-MPLS com equipamentos puramente *OpenFlow*. Entretanto, é importante notar que a arquitetura híbrida desenvolvida no trabalho tem a principal desvantagem de que cada um dos equipamentos de rede têm que implementar simultâneamente o controle distribuído atual e o controle *OpenFlow*, requerendo assim de um maior processamento quando é comparada com as outras alternativas.

6.2 Conclusões da proposta de arquitetura para a rede móvel do SP

.

A próxima geração de redes móveis 5G está evoluindo para incorporar os conceitos de NFV, movimentar o controle para a fronteira das redes móveis e para incorporar os conceitos das SDN,

incrementando a separação entre o plano de controle e o de dados. Este trabalho segue esta evolução das redes móveis, particularmente focalizandose nas fraquezas e nas fortalezas da rede SDN e incrementando o uso de estas tecnologias. Enquanto a programabilidade é colocada em todos os elementos da rede móvel, é possível criar lógicas mais especializadas.

Este trabalho propõe uma lógica especializada para encaminhar imediatamente o tráfego sem requerimentos QoS utilizando política simples implementadas sobre regras de encaminhamento proativas. Com esta finalidade, o plano de dados da arquitetura móvel proposta continúa utilizando elementos centrais denominado OF-GW, e o tráfego é encaminhado utilizando este elemento. É importante ressaltar que todos os dispositivos que formam parte do plano de controle são implementados como simples *switches OpenFlow*, sendo fácil sua substituição. O plano de dados é distribuído entre mais elementos programaveis, melhorando a robustez da solução, dado que qualquer *switches OpenFlow* pode ser configurado para atuar como um OF-GW.

Adicionalmente, se cria uma lógica especializada para encaminhar o tráfego com requerimentos de QoS utilizando métodos proativos e reativos ao mesmo tempo. A lógica proposta é desenhada para criar um EPS-*Bearer* específico capaz de utilizar um caminho ótimo que satisfaça as características de QoS, e de esse modo com objetivos das redes móvel 5G. Esta lógica está integrada por um método programável e flexível para detectar tráfego com requerimentos QoS, método que é aplicado próximo aos UEs. A arquitetura criada não adiciona atrasos durante a criação de EPS-*Bearer* com QoS nem durante a detecção de tráfego QoS. Inicialmente, os pacotes com requerimentos de QoS são enviados por duas linhas de processados simultaneamente, uma que encaminha imediatamente uma cópia dos pacotes através de OF-*Bearer* por *defalut*; enquanto a outra é responsavel pela detecção dos requerimentos de QoS e a criação de EPS-*Bearers* específicos que permitam cumprir com os requerimentos.

O plano de controle proposto segue os conceitos que estão sendo utilizados nas arquiteturas 5GS, que propõem a implementação das entidades do plano de controle da rede móvel utilizando os conceitos de NFV. Esta tecnologia permite a virtualização das principais entidades móveis, as quais são implementadas como VNF sobre *hardware* padrão de menor custo, ou sobre um ambiente de computação em nuvem. A tecnologia NFV oferece uma alternativa flexível e escalável que pode se adaptar às variações nas demandas de tráfego, assim como fornecer escalabilidade de forma distribuída incrementa a robustez na da rede móvel.

Finalmente, o *backhaul* móvel proposto e a arquitetura de rede móvel de *core* são criados para interagir opcionalmente com UE da futura geração 5G ou UE da atual 4G, utilizando procedimentos de NG-RAN, ou procedimentos padrão de E-UTRAN.

6.3 Limitações do estudo

Se bem a lógica proposta tem boas características enquanto ao encaminhamento imediato do tráfego, é importante notar que esta lógica pode não ser apropriada para tráfegos que por segurança requerem de inspeção antes de ser encaminhados. Em estes casos, o comportamento padrão de *OpenFlow* pode ser mais apropriado, onde os pacotes devem aguardar as análises do Controlador

e da sua resposta antes de ser encaminhados.

Adicionalmente, para os casos de ataques de DoS (*Denial of Service*) ou DDoS (*Distributed Denial of Service*) a construção de caminhos com requerimentos ótimos não é apropriado. O tráfego com destino o endereço atacado deve ser encaminhado para um elemento intermédio localizado num DataCenter para sua inspeção. Este elemento deve examinar os pacotes em profundidade, e os pacotes que compõem o ataque não devem ser encaminhados até o destino final.

6.4 Trabalhos futuros

Como é indicado na Figura 3.2, a tabela 0 é a responsevel pelo encaminhamento do tráfego para uma lógica de encaminhamento específica, como são a lógica de encaminhamento geral ou a lógica de gerenciamento QoS. Mas como é indicado na mesma figura, outro tipo de lógicas também podem ser criadas e podem ser configuradas novas regras na tabela 0 que permitam encaminhar os pacotes para estas lógicas.

Trabalhos futuros podem se focar na criação de lógicas de encaminhamento para tráfegos que requerem um tratamento diferente ao proposto neste trabalho. São exemplos de estas, o encaminhamento de tráfego que requer autorizaço prévia para seu envio, lógicas para a mitigação de ataques de DoS ou lógicas para o encaminhamento do tráfego para outros sistemas autónomos, entre outras.

Por outra parte, deve ser analisado o comportamento da solução para diversidade de aplicações e serviços. Devem ser implementados em detalhe os mecanismos de deteção de tráfego com QoS, os métodos de extração de requerimentos de QoS, e os algoritmos de criação de fluxos para cada aplicação e serviço. Adicionalmente, outras métricas para a comparativa dos resultados devem ser também consideradas.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] MCKEOWN, N. et al. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, ACM, v. 38, n. 2, p. 69–74, 2008.

[2] OpenFlow v.1.5.1 . *Open Networking Foundation. OpenFlow Switch Specification Version.1.5.1.* 2015. Https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.1.pdf.

[3] LIMONCELLI, T. A. Openflow: a radical new idea in networking. *Queue*, ACM, v. 10, n. 6, p. 40, 2012.

[4] EGILMEZ, H. E. et al. Scalable video streaming over OpenFlow networks: An optimization framework for QoS routing. In: IEEE. *Image Processing (ICIP), 2011 18th IEEE International Conference on*. [S.l.], 2011. p. 2241–2244.

[5] DAS, S. et al. Application-aware aggregation and traffic engineering in a converged packet-circuit network. In: OPTICAL SOCIETY OF AMERICA. *National Fiber Optic Engineers Conference*. [S.l.], 2011. p. NThD3.

[6] AGARWAL, S.; KODIALAM, M.; LAKSHMAN, T. Traffic engineering in software defined networks. In: IEEE. *INFOCOM, 2013 Proceedings IEEE*. [S.l.], 2013. p. 2211–2219.

[7] DAS, S. *PAC.C*. Tese (PhD Dissertation) — Stanford University, 2012. `<http://www.openflow.org/wk/index.php/PACC_Thesis>`.

[8] SHIRAZIPOUR, M. et al. Realizing packet-optical integration with SDN and OpenFlow 1.1 extensions. In: IEEE. *Communications (ICC), 2012 IEEE International Conference on*. [S.l.], 2012. p. 6633–6637.

[9] DAS, S. et al. Packet and circuit network convergence with OpenFlow. In: IEEE. *Optical Fiber Communication (OFC), collocated National Fiber Optic Engineers Conference, 2010 Conference on (OFC/NFOEC)*. [S.l.], 2010. p. 1–3.

[10] GIORGETTI, A. et al. Openflow and pce architectures in wavelength switched optical networks. In: IEEE. *Optical Network Design and Modeling (ONDM), 2012 16th International Conference on*. [S.l.], 2012. p. 1–6.

[11] BAKSHI, K. Considerations for software defined networking (sdn): Approaches and use cases. In: IEEE. *Aerospace Conference, 2013 IEEE*. [S.l.], 2013. p. 1–9.

[12] MATIAS, J. et al. An openflow based network virtualization framework for the cloud. In: IEEE. *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*. [S.l.], 2011. p. 672–678.

[13] BIFULCO, R. et al. Scalability of a mobile cloud management system. In: ACM. *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. [S.l.], 2012. p. 17–22.

[14] NAMAL, S. et al. Enabling secure mobility with openflow. In: IEEE. *Future Networks and Services (SDN4FNS), 2013 IEEE SDN for*. [S.l.], 2013. p. 1–5.

[15] Open Networking Foundation. *SDN Product Directory*. Disponível em: <https://www.opennetworking.org/products-listing>.

[16] BENTON, K.; CAMP, L. J.; SMALL, C. Openflow vulnerability assessment. In: ACM. *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*. [S.l.], 2013. p. 151–152.

[17] LOPEZ-RODRIGUEZ, F.; CAMPELO, D. R. A robust SDN network architecture for service providers. In: IEEE. *Global Communications Conference (GLOBECOM), 2014 IEEE*. [S.l.], 2014. p. 1903–1908.

[18] RODRÍGUEZ, F. L. et al. Qos management and flexible traffic detection architecture for 5g mobile networks. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 19, n. 6, p. 1335, 2019.

[19] KEMPF, J. et al. Fostering rapid, crossdomain service innovation in operator networks through service provider sdn. In: IEEE. *Communications (ICC), 2014 IEEE International Conference on*. [S.l.], 2014. p. 3064–3069.

[20] Open Networking Foundation. *Software-Defined Networking (SDN) Definition*. Disponível em: <https://www.opennetworking.org/sdn-resources/sdn-definition>.

[21] GHEIN, L. D. Mpls fundamentals. Cisco Press, 2007.

[22] OSBORNE, E. D.; SIMHA, A. *Traffic engineering with MPLS*. [S.l.]: Cisco Press, 2002.

[23] PEPELNJAK, I.; GUICHARD, J. *MPLS and VPN architectures*. [S.l.]: Cisco Press, 2002.

[24] Open Source Project. *Mininet Web Page*. Disponível em: <https://mininet.org>.

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

[26] PFAFF, B. et al. Extending networking into the virtualization layer. In: *Hotnets*. [S.l.: s.n.], 2009.

[27] Open Source Project. *Open vSwitch Web Page*. Disponível em: <http://openvswitch.org/>.

[28] Open Source Project. *Ryu Web Page*. Disponível em: <http://osrg.github.io/ryu/>.

[29] ZHANG, H.; YAN, J. Performance of SDN routing in comparison with legacy routing protocols. In: IEEE. *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2015 International Conference on*. [S.l.], 2015. p. 491–494.

[30] WAN, T.; ABDOU, A.; OORSCHOT, P. C. van. A framework and comparative analysis of control plane security of SDN and conventional networks. *arXiv preprint arXiv:1703.06992*, 2017.

[31] KHAN, A. A. et al. A convergence time optimization paradigm for OSPF based networks through SDN SPF protocol computer communications and networks (ccn)/delay tolerant networks. In: ACM. *Proceedings of the International Conference on Future Networks and Distributed Systems*. [S.l.], 2017. p. 43.

[32] BENSON, T.; AKELLA, A.; MALTZ, D. A. Unraveling the complexity of network management. In: *NSDI*. [S.l.: s.n.], 2009. p. 335–348.

[33] RAGHAVAN, B. et al. Software-defined internet architecture: decoupling architecture from infrastructure. In: ACM. *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*. [S.l.], 2012. p. 43–48.

[34] GHODSI, A. et al. Intelligent design enables architectural evolution. In: ACM. *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*. [S.l.], 2011. p. 3.

[35] KREUTZ, D. et al. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, IEEE, v. 103, n. 1, p. 14–76, 2015.

[36] KIM, H.; FEAMSTER, N. Improving network management with software defined networking. *IEEE Communications Magazine*, IEEE, v. 51, n. 2, p. 114–119, 2013.

[37] MENDIOLA, A. et al. A survey on the contributions of software-defined networking to traffic engineering. *IEEE Communications Surveys & Tutorials*, IEEE, 2017.

[38] JAIN, R.; PAUL, S. Network virtualization and software defined networking for cloud computing: a survey. *IEEE Communications Magazine*, IEEE, v. 51, n. 11, p. 24–31, 2013.

[39] JARARWEH, Y. et al. Software defined cloud: Survey, system and evaluation. *Future Generation Computer Systems*, Elsevier, v. 58, p. 56–74, 2016.

[40] COSTA-REQUENA, J. Sdn integration in LTE mobile backhaul networks. In: IEEE. *Information Networking (ICOIN), 2014 International Conference on*. [S.l.], 2014. p. 264–269.

[41] NGUYEN, V.-G.; DO, T.-X.; KIM, Y. SDN and virtualization-based LTE mobile network architectures: A comprehensive survey. *Wireless Personal Communications*, Springer, v. 86, n. 3, p. 1401–1438, 2016.

[42] KIM, Y.-h. et al. A SDN-based distributed mobility management in LTE/EPC network. *The Journal of Supercomputing*, Springer, v. 73, n. 7, p. 2919–2933, 2017.

[43] SEZER, S. et al. Are we ready for SDN? implementation challenges for software-defined networks. *IEEE Communications Magazine*, IEEE, v. 51, n. 7, p. 36–43, 2013.

[44] DIXIT, A. et al. Towards an elastic distributed SDN controller. In: ACM. *ACM SIGCOMM Computer Communication Review*. [S.l.], 2013. v. 43, n. 4, p. 7–12.

[45] LEVIN, D. et al. Logically centralized?: state distribution trade-offs in software defined networks. In: ACM. *Proceedings of the first workshop on Hot topics in software defined networks*. [S.l.], 2012. p. 1–6.

[46] TOOTOONCHIAN, A.; GANJALI, Y. Hyperflow: A distributed control plane for OpenFlow. In: USENIX ASSOCIATION. *Proceedings of the 2010 internet network management conference on Research on enterprise networking*. [S.l.], 2010. p. 3–3.

[47] ASLAN, M.; MATRAWY, A. Adaptive consistency for distributed SDN controllers. In: IEEE. *Telecommunications Network Strategy and Planning Symposium (Networks), 2016 17th International*. [S.l.], 2016. p. 150–157.

[48] OKTIAN, Y. E. et al. Distributed SDN controller system: A survey on design choice. *Computer Networks*, Elsevier, v. 121, p. 100–111, 2017.

[49] VISSICCHIO, S. et al. On the co-existence of distributed and centralized routing control-planes. In: IEEE. *Computer Communications (INFOCOM), 2015 IEEE Conference on*. [S.l.], 2015. p. 469–477.

[50] GUO, Y. et al. Traffic engineering in SDN/OSPF hybrid network. In: IEEE. *Network Protocols (ICNP), 2014 IEEE 22nd International Conference on*. [S.l.], 2014. p. 563–568.

[51] CARIA, M.; JUKAN, A.; HOFFMANN, M. SDN partitioning: A centralized control plane for distributed routing protocols. *IEEE Transactions on Network and Service Management*, IEEE, v. 13, n. 3, p. 381–393, 2016.

[52] CARIA, M.; DAS, T.; JUKAN, A. Divide and conquer: Partitioning OSPF networks with SDN. In: IEEE. *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*. [S.l.], 2015. p. 467–474.

[53] AMIN, R.; REISSLEIN, M.; SHAH, N. Hybrid sdn networks: A survey of existing approaches. *IEEE Communications Surveys & Tutorials*, IEEE, 2018.

[54] HUANG, X. et al. A survey of deployment solutions and optimization strategies for hybrid sdn networks. *IEEE Communications Surveys & Tutorials*, IEEE, 2018.

[55] VAGO, J. D. *Uma estratégia para estabelecer fluxos em redes sdn-openflow com redução de carga no controlador*. DissertaÃğÃčo (Mestrado) — Universidade Federal de Pernambuco, 2017.

[56] LUO, T. et al. Enhancing responsiveness and scalability for openflow networks via control-message quenching. In: IEEE. *ICT Convergence (ICTC), 2012 International Conference on*. [S.l.], 2012. p. 348–353.

[57] WAMSER, F. et al. Traffic characterization of a residential wireless Internet access. *Telecommunication Systems*, Springer, v. 48, n. 1-2, p. 5–17, 2011.

[58] Ivan Pepelnjak. *Hybrid OpenFlow, The Brocade Way.* 2012. Disponível em: <http://blog.ipspace.net/2012/06/hybrid-openflow-brocade-way.html>.

[59] Open Networking Foundation. *Outcomes of the Hybrid Working Group.* 2013. Disponível em: <https://www.opennetworking.org/images/stories/downloads/working-groups/summary-hybrid.pdf>.

[60] B15.2.0, G. T. . *3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; System Architecture for the 5G System; Stage 2 (Release 15).* 2018. Disponível em: <http://www.3gpp.org/ftp/Specs/archive/23_series/23.501/>.

[61] 23.401, G. G. T. *Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS) Enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Access, (Release 15).*

[62] OLSSON, M.; MULLIGAN, C. *EPC and 4G Packet Networks: Driving the mobile broadband revolution.* [S.l.]: Academic Press, 2012.

[63] MIJUMBI, R. et al. Network function virtualization: State-of-the-art and research challenges. *IEEE Communications surveys & tutorials*, IEEE, v. 18, n. 1, p. 236–262, 2015.

[64] YI, B. et al. A comprehensive survey of network function virtualization. *Computer Networks*, Elsevier, v. 133, p. 212–262, 2018.

[65] BOURAS, C.; KOLLIA, A.; PAPAZOIS, A. Sdn & nfv in 5g: Advancements and challenges. In: IEEE. *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN).* [S.l.], 2017. p. 107–111.

[66] YANG, G. et al. Flowvirt: Flow rule virtualization for dynamic scalability of programmable network virtualization. In: IEEE COMPUTER SOCIETY. *11th IEEE International Conference on Cloud Computing, CLOUD 2018.* [S.l.], 2018. p. 350–358.

[67] YANG, G. et al. Litevisor: A network hypervisor to support flow aggregation and seamless network reconfiguration for vm migration in virtualized software-defined networks. *IEEE Access*, IEEE, v. 6, p. 65945–65959, 2018.

[68] LIYANAGE, M.; GURTOV, A.; YLIANTTILA, M. *Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture.* [S.l.]: John Wiley & Sons, 2015.

[69] NUNES, B. A. A. et al. A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, IEEE, v. 16, n. 3, p. 1617–1634, 2014.

[70] DAS, S. *PAC. C: A unified control architecture for packet and circuit network convergence.* Tese (Doutorado) — Citeseer, 2012.

[71] ALLIANCE, N. 5g white paper. *Next generation mobile networks, white paper*, p. 1–125, 2015.

[72] NGUYEN, V.-G. et al. Sdn/nfv-based mobile packet core network architectures: A survey. *IEEE Communications Surveys & Tutorials*, IEEE, v. 19, n. 3, p. 1567–1602, 2017.

[73] COSTA-REQUENA, J. Sdn integration in lte mobile backhaul networks. In: IEEE. *Information Networking (ICOIN), 2014 International Conference on*. [S.l.], 2014. p. 264–269.

[74] SAMA, M. R. et al. Enabling network programmability in lte/epc architecture using openflow. In: IEEE. *2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. [S.l.], 2014. p. 389–396.

[75] JAIN, A. et al. A comparison of sdn and nfv for re-designing the lte packet core. In: IEEE. *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. [S.l.], 2016. p. 74–80.

[76] NGUYEN, V.-G.; KIM, Y. Proposal and evaluation of sdn-based mobile packet core networks. *EURASIP Journal on Wireless Communications and Networking*, Springer, v. 2015, n. 1, p. 172, 2015.

[77] FOUNDATION, O. N. *OpenFlow Switch Specification Version 1.5.0.* 2014. Disponível em: <https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.0.noipr.pdf>.

[78] ETSI. *MEC in 5G networks*. 2018. Disponível em: <https://www.etsi.org/images/ files/ETSIWhite Papers /etsi_wp28_mec_in_5G_FINAL.pdf>.

[79] GTI. *GTI 5G Network Architecture WhitePaper V1.0.* 2018. Disponível em: <http://www.gtigroup.org/d/file/Resources/rep/2018-02-22/06608ce6dbe32673ac1611359e11f794.pdf>.

[80] HAWILO, H. et al. Nfv: State of the art, challenges and implementation in next generation mobile networks (vepc). *arXiv preprint arXiv:1409.4149*, 2014.

[81] HIRSCHMAN, B. et al. High-performance evolved packet core signaling and bearer processing on general-purpose processors. *IEEE Network*, IEEE, v. 29, n. 3, p. 6–14, 2015.

[82] SIASI, N.; SULIEMAN, N. I.; GITLIN, R. D. Ultra-reliable nfv-based 5g networks using diversity and network coding. In: IEEE. *2018 IEEE 19th Wireless and Microwave Technology Conference (WAMICON)*. [S.l.], 2018. p. 1–4.

[83] REN, Y. et al. Dynamic auto scaling algorithm (dasa) for 5g mobile networks. In: IEEE. *2016 IEEE Global Communications Conference (GLOBECOM)*. [S.l.], 2016. p. 1–6.

[84] SEXTON, C. et al. 5g: adaptable networks enabled by versatile radio access technologies. *IEEE Communications Surveys & Tutorials*, IEEE, v. 19, n. 2, p. 688–720, 2017.

[85] PARVEZ, I. et al. A survey on low latency towards 5g: Ran, core network and caching solutions. *IEEE Communications Surveys & Tutorials*, IEEE, v. 20, n. 4, p. 3098–3130, 2018.

[86] ZHANG, L. et al. A survey of advanced techniques for spectrum sharing in 5g networks. *IEEE Wireless Communications*, IEEE, v. 24, n. 5, p. 44–51, 2017.

[87] PAGÉ, J.; DRICOT, J.-M. Software-defined networking for low-latency 5g core network. In: IEEE. *2016 International Conference on Military Communications and Information Systems (ICMCIS)*. [S.l.], 2016. p. 1–7.

[88] COSTA-REQUENA, J. et al. Sdn and nfv integration in generalized mobile network architecture. In: IEEE. *2015 European conference on networks and communications (EuCNC)*. [S.l.], 2015. p. 154–158.

[89] KAIPPALLIMALIL, J.; CHAN, H. A. Network virtualization and direct ethernet transport for packet data network connections in 5g wireless. In: IEEE. *2014 IEEE Global Communications Conference*. [S.l.], 2014. p. 1836–1841.

[90] FERNANDEZ, M. P. Comparing openflow controller paradigms scalability: Reactive and proactive. In: IEEE. *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*. [S.l.], 2013. p. 1009–1016.

[91] JIN, X. et al. Covisor: A compositional hypervisor for software-defined networks. In: *12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15)*. [S.l.: s.n.], 2015. p. 87–101.

[92] Netmania. *Call Flow of Initial Attach*. 2014. Http://www.netmanias.com/en/post/techdocs/6102/attach-emm-lte/emm-procedure-1-initial-attach-part-2-call-flow-of-initial-attach.

[93] BASTURK, E. et al. Using network layer anycast for load distribution in the internet. In: CITESEER. *Tech. Rep., IBM TJ Watson Research Center*. [S.l.], 1997.