Encadeamento de serviços em rede: uma nova abordagem para provisionamento dinâmico de serviços

Tales Anaximandro do Bonfim Visgueira [1]
Ricardo Gomes Queiroz [1]
Christian Esteve Rothenberg [2]

Resumo: A expansão tecnológica na Internet tem motivado pesquisadores a buscar técnicas no intuito de flexibilizar o comportamento das redes, bem como minimizar custos de operação e prover serviços de forma dinâmica para adequação às novas necessidades tecnológicas. Neste contexto, este artigo apresenta o conceito do encadeamento dinâmico de serviços em redes de computadores, do inglês *Network Service Chaining (NSC)*, exemplificado usando uma implementação protótipo baseada no controlador OpenDayLight.

***Palavras-chave*:** Encadeamento de Serviços em Redes. OpenDayLight. Redes Definidas por Software.

Abstract: *The technological expansion on the Internet has motivated researchers to seek techniques in order to make more flexible the network behavior, and minimize operating costs and provide services to dynamically conform to the new technological requirements. In this context, this paper the concept of Network Service Chaining (NSC) exemplified through a prototype implementation based on the OpenDayLight controller.*

***Keywords*:** *Network Service Chaining. OpenDayLight. Software Defined Networking*

1 INTRODUÇÃO

A crescente demanda na utilização dos meios de telecomunicações tem proporcionado uma explosão de novas soluções tecnológicas (e.g., Telefonia Móvel, Virtualização de Máquinas, Computação em Nuvem e a revolucionária Internet das Coisas), e devido ao desenvolvimento dessas novas soluções, há a previsão de existir mais de 50 bilhões de dispositivos conectados na Internet até 2020 [1]. Durante a realização de pesquisas para evolução da infraestrutura das redes, os pesquisadores identificaram a existência de problemas que tornam difícil a realização de mudanças no núcleo das redes, como a existência de uma grande quantidade de protocolos de comunicação, desenvolvidos por fabricantes de dispositivos diferentes (protocolos proprietários) e a existência de "caixas pretas", implementações verticalmente integradas em plataformas *software* e *hardware* também conhecidos como *middlebox* [2].

Associados a estes problemas a arquitetura atual das redes de computadores, projetada na década de 80 (há mais de 30 anos) e estruturada na comutação de pacotes com a arquitetura TCP/IP, consiste em uma estrutura de rede calcificada ou ossificada (engessada), uma analogia ao processo de envelhecimento das cartilagens dos ossos dos seres vivos quando atingem o seu nível de amadurecimento máximo, e por essa razão, dificultam a implementação de novas soluções tecnológicas [3]. Adicionalmente aos problemas comentados, em ambientes corporativos como *data centers*, núcleos de Computação em Nuvem e soluções para a Internet das Coisas, existe a necessidade de disponibilizar infraestrutura ou serviços de rede para atender às demandas dos usuários, onde normalmente são atendidas com a utilização de Máquinas Virtuais, do inglês *Virtual Marchine (VM)*. A evolução para este novo ambiente virtual proporcionou o surgimento de uma nova geração de redes inteligentes e flexíveis baseadas em software e com uma abordagem revolucionária.

Segundo McKeown et al. (2008), o paradigma das Redes Definidas por Software, do inglês *Software Defined Networking (SDN)*, surgiu como uma abordagem tecnológica visando promover a inovação das redes de

[1]Pós-Graduação Lato Sensu em Redes de Computadores - Faculdade Santo Agostinho (FSA) - Teresina (PI) - Brasil
{talesvisgueira,rgqueiroz@gmail.com}
[2]Departamento de Engenharia, Computação e Automação Industrial - Unicamp - Campinas (SP) - Brasil
{chesteve@dca.fee.unicamp.br }

computadores. O principio básico do paradigma SDN é a capacidade de programar os elementos da rede através de uma interface programática de software (API), que desacopla o plano de dados do plano de controle. Outra tecnologia emergente é a virtualização de funções de redes, do inglês *Network Function Virtualization (NFV)*, que desacopla a função executada por um sistema (*software*) de sua parte física (*hardware*). Apesar dessas inovações trazerem avanços na gerência das redes, existe ainda a necessidade de escalonar os serviços de forma dinâmica e flexível. Atualmente, esses serviços são implantados de forma estática e dependem de configurações fixas que tornam a abordagem inflexível e difícil de adaptar-se às mudanças de requisitos.

Dentro do contexto acima, o problema desta pesquisa consiste em tornar o provisionamento dos serviços nas redes de computadores mais escaláveis e flexíveis. A hipótese em estudo é que a integração dos conceitos de gerenciamento centralizado e programável, somados à capacidade de virtualização dos dispositivos existentes nas tecnologias SDN e NFV, podem otimizar a forma de provisionamento dos serviços. Nesta abordagem, o objetivo principal é implementar o conceito de Encadeamento de Serviços em Redes (NSC), com a finalidade de prover serviços de forma dinâmica através da técnica de Cadeia de Funções de Serviços, do inglês *Service Function Chains (SFC)*.

Os objetivos específicos serão desenvolvidos com os seguintes passos: i) analisar e apresentar os novos conceitos da próxima geração de redes (e.g., SDN, NFV e SFC) aos administradores não familiarizados com essas tecnologias; ii) implementar o estado da arte no provimento de serviços com a técnica SFC, disponibilizada pelo controlador SDN de código aberto OpenDayLight (ODL) e; iii) avaliar a implementação prova de conceito de NSC em um ambiente de experimental baseado no emulador Mininet. A justificativa para a realização deste trabalho baseia-se na necessidade de explorar a tecnologia de NSH no contexto de SDN e NFV como paradigmas na área de redes para disponibilizar, de forma otimizada, políticas e serviços em diferentes ambientes de redes de computadores para atender a crescente demanda de clientes e soluções tecnológicas.

A organização deste trabalho está estruturada da seguinte forma: inicialmente na seção 2 é realizada a revisão teórica referente as tecnologias SDN e NFV, dando sequência, na seção 3 é apresentada a arquitetura do Encadeamento de Serviços em Rede (NSC) para posteriormente, na seção 4 ser discutido o processo de implementação do provisionamento dinâmico de serviços com o controlador OpenDayLight e por fim, na seção 5 são apresentadas as conclusões obtidas no experimento e os aspectos para implantação do conceito NSC em novos ambientes.

2 REFERENCIAL TEÓRICO

Atualmente existem duas abordagens com a finalidade de decidir o "Futuro das Redes". A primeira, denominada **Evolucionária**, argumenta que a arquitetura TCP/IP, elemento chave no modelo das redes atuais tem funcionado satisfatoriamente, não sendo recomendável uma ruptura completa, apenas a evolução do próprio protocolo. A segunda abordagem tem uma visão mais **Revolucionária** e defende a criação de uma nova rede, deve-se abandonar o modelo atual e desenvolver uma arquitetura inteiramente nova [5]. Essa nova ideia de recomeçar do zero é conhecida entre os pesquisadores como *Clean-Slate* (algo como "Tela em Branco"), ou arquitetura disruptiva [6]. No entanto, apesar de existirem ideias distintas, há um consenso entre os pesquisadores em relação à necessidade de realizar mudanças estruturais nas redes para torná-las mais eficientes e flexíveis.

No contexto das abordagens evolutivas, foram desenvolvidas soluções emergenciais (remendos) para minimizar os efeitos ocasionados pela crescente expansão da arquitetura das redes, destacando-se as seguintes: i) *Classless Internet Domain Routing (CIDR)*, uma solução definida para estender o endereçamento IPv4; ii) *Network Address Translation (NAT)*, uma solução de interconexão entre redes privadas e públicas; iii) *Dynamic Host Configuration Protocol (DHCP)*, um serviço para distribuição de endereços IP e informações adicionais da rede (e.g., máscara de sub-rede, gateway, endereço de DNS etc). Já no contexto revolucionário, a nova geração de IP conhecida como IPv6 e definida na RFC 1550:1993 como um novo protocolo para transporte de dados, iniciou o processo de mudanças *Clean-Slate*. Uma de suas principais características inovadora é o aumento do endereçamento IP de 2^{32} para 2^{128}, o equivalente a 340 undecilhões de novos endereços. Outras soluções revolucionárias também estão em fase de implantação (e.g., redes programáveis e funções de rede virtualizadas) [5].

Na implementação dessas novas soluções, os pesquisadores sempre esbarravam em dificuldades de realizar mudanças nas redes, pelo fato dos trabalhos de pesquisas permanecerem em projetos de laboratórios e as redes atu-

ais serem constituídas por dispositivos proprietários (roteadores/*switches*) contendo componentes internos também chamados de *middlebox* ou "Caixas Pretas" normalmente implementados por fornecedores distintos (e.g., Cisco, Hewlett-Packard, Juniper, NEC), que tornam as mudança difíceis de serem testadas em grande escala.

A Força Tarefa de Engenheiros da Internet, do inglês *Internet Engineering Task Force (IETF)*, define na RFC-3224:2002, o termo *Middlebox* como qualquer dispositivo intermediário que executa funções específicas no encaminhamento de datagramas entre um host de origem e o destino [7]. Esses componentes desempenham funções bem específicas como Firewall, NAT, Proxy, Sistemas de Detecção de Instrução (IDS/IPS), *Load Balancer*, etc. Embora estas ferramentas ofereçam um ponto de vista central de administração, operam no nível de protocolos e necessitam de engenheiros de redes para configurar políticas definidas em alto nível para comandos de baixo nível. Esta forma de operação, diminui a inovação e aumenta os custos operacionais de gestão das redes.

2.1 Origem das redes programáveis

Com a finalidade de tornar as redes atuais mais flexíveis e gerenciáveis, com base na analogia da relativa facilidade de programar os computadores pessoais, os pesquisadores propuseram soluções que contribuíram na formação do conceito de redes programáveis, onde destacam-se as seguintes: Redes Ativas, NetFPGA, ForCES, SANE, Ethane e o protocolo OpenFlow.

- **Redes Ativas**, do inglês *Active Networking*, foi uma solução proposta em 1990 como uma abordagem que utiliza uma interface de programação (API) para expor os recursos de redes (e.g., filas de processamento, armazenamento e pacotes), apoiados em aplicações com funcionalidade personalizada para aplicar regras em um subconjunto de pacotes que passam através do nó, semelhante a um sistema operacional de rede em cada nó [8]. Esta abordagem foi considerada inadequada, pois havia o entendimento que a simplicidade no núcleo das redes era fundamental para o sucesso da Internet. Neste sentido, as Redes Ativas foram uma das primeiras abordagens *Clean-Slate* que exploravam os serviços prestados pela pilha tradicional da Internet.

- A solução **NetFPGA** foi iniciada no final de 2001 na Universidade de Stanford, como uma plataforma aberta que combina memórias (SRAM e DRAM), processadores de sinais e interfaces de redes Ethernet (1 Gbps ou 10 Gbps) em uma placa padrão PCI, denominada *Field-Programmable Gate Array - FPGA*. Sua finalidade é permitir aos desenvolvedores programar o FPGA para implementar novos mecanismos, protocolos e arquiteturas. A NetFPGA possibilita modificar os pacotes em trânsito e controlar o encaminhamento do tráfego de rede. Segundo [9], o projeto NetFPGA foi motivado pela necessidade de criar ferramentas para ensinar sistemas de rede de computadores em nível de graduação e pós-graduação.

- A solução **ForCES**, do inglês *Forwarding and Control Element Separation* é considerada a primeira iniciativa desenvolvida para separação do elemento de controle e encaminhamento, sua proposta foi apresentada inicialmente pela IETF com a RFC 3654 em 2003. A ideia era redefinir a arquitetura interna dos dispositivos de rede, separando os elementos de controle e encaminhamento [10]. No entanto, o dispositivo de rede ainda é representado como uma entidade única.

- O projeto **SANE**, do inglês *Secure Architecture for the Networked Enterprise (SANE)* foi concebido como uma arquitetura segura para redes empresariais em 2006, sua finalidade era fornecer uma única camada de proteção para redes, localizada entre as camadas Ethernet e de Rede, similar ao que ocorre com as *Virtual Local Area Network (VLANs)*. Nessa abordagem rotas são construídas por um controlador de domínio logicamente centralizado, que concede acesso usando um ponto de vista global, permitindo implementar políticas de uma forma independente de topologia [11]. Em contraste com as políticas de redes existentes, que baseiam-se em regras nos firewalls e outros middleboxes com dependências implícitas sobre como a topologia das redes estão implantadas .

- O projeto **Ethane** (sucessor do SANE) estabelece uma solução logicamente centralizada a nível de fluxo para o controle de acesso em redes empresariais. Sua principal característica era reduzir as interrupções às tabelas de fluxo nos dispositivos de comutação de dados. Essas tabelas são preenchidas por um controlador central, que utilizava políticas de segurança de alto nível. O projeto foi implementado inicialmente no departamento de ciência da computação de Stanford em 2007, uma preparação do terreno para a criação do protocolo OpenFlow [12].

- O protocolo **OpenFlow** foi desenvolvido em 2008 como uma solução baseada na abordagem *Clean-Slate*, considerada a que mais teve sucesso na implementação da separação dos planos de controle e dados. Sua principal finalidade é controlar o fluxo de dados, definir as rotas que os pacotes vão seguir e o processamento que devem ser submetidos.

Todas essas soluções colaboraram para o surgimento de um novo paradigma de flexibilização e otimização, chamado de **Redes Definidas por Software (SDN)** [8]. O termo SDN teve sua proposta inicial na publicação do artigo de Greene (2009), que relatou os trabalhos dos pesquisadores Nick McKeown, Martin Casado e outros membros da Universidade de Stanford, motivados pela incapacidade de mexer nos dispositivos de encaminhamento, e por este motivo, desenvolveram um novo padrão que acessa as tabelas de fluxos e modificam o comportamento de encaminhamento dos pacotes da rede [8]. Atualmente, o órgão responsável para evoluir o conceito SDN é a fundação *Open Networking Foundation - ONF*, estruturada em vários grupos de trabalhos (GTs) e focados em promover às áreas de Extensibilidade, Transporte Óptico, Rede Móvel e sem Fio. A IETF também definiu o conceito SDN, através da RFC-71409:2014, como um conjunto de técnicas utilizadas para facilitar o projeto, entrega e gestão de serviços de rede de uma maneira determinista, dinâmica e escalável [13].

2.2 Redes Definidas por Software (SDN)

A arquitetura clássica de SDN é dividida em três planos: plano de aplicação, controle e dados. No plano de aplicação encontram-se os aplicativos (softwares especialistas) que podem receber eventos de redes e atuar de forma proativa ou reativa para oferecer serviços ou configurações personalizadas. No plano de controle, fica o controlador central, também conhecido como Servidor Operacional de Rede, do inglês *Network Operating System (NOS)*. Já o plano de dados é constituído por dispositivos (Roteadores ou Switchs) que realizam a função de encaminhar os dados. A Figura 1 ilustra um modelo da arquitetural SDN.

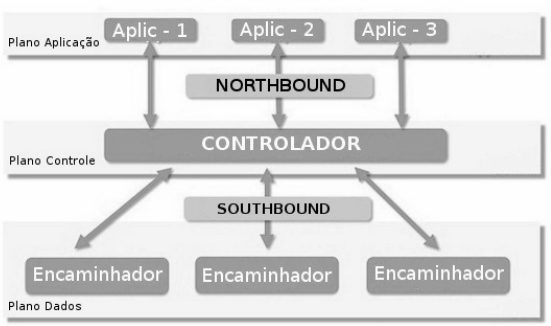

Figura 1: Adaptação do modelo da arquitetura SDN
Fonte Intel Digital Labs, 2014

Aplicações SDN são soluções a nível de software implementas em alguma linguagem de programação (e.g,. C, C++, Java, Python, etc.) com a finalidade de processar as mensagens enviadas pelo plano de dados ao controlador e repassada às aplicações através de uma API disponibilizada pelo próprio controlador SDN. Essa característica possibilita uma capacidade programável para tomada de decisão sobre quais ações devem ser realizadas pelo dispositivo de encaminhamento, por este motivo as aplicações podem monitorar o estado dos fluxos de dados e permitir alteração no comportamento da rede.

A Northbound API (interface para o norte, em português), representa uma interface de comunicação entre o plano de controle e às aplicações SDN para permitir o gerenciamento sobre as funções de rede. A função principal da API é traduzir os requisitos das aplicações de gerenciamento em instruções de baixo nível para os dispositivos da rede e transmitir estatísticas geradas nos dispositivos de encaminhamento e processadas pelo controlador para às aplicações SDN. Essa interface abstrai os eventos e conjuntos de instruções de baixo nível dos dispositivos de comutação para possibilitar às aplicações acopladas ao controlador decidir quais ações de encaminhamento, encapsulamento ou descarte (drop) de pacotes serão realizadas. Normalmente a API é implementada por um dos protocolos a seguir: Flow Management Language (FML), Procera, Frenetic, RCP ou REST.

A Interface Southbound (interface para o sul, em português), representa o elo de ligação entre o plano de controle e os elementos de encaminhamento de dados, sendo, portanto, o elemento crucial para separar claramente a funcionalidade de controle e plano de dados. Essa API está fortemente ligada aos elementos de encaminhamento da infraestrutura física ou virtual, podendo ser implementada pelos protocolos OpenFlow e ForCES, Protocol-Oblivious Forwarding (POF), Path Computation Element (PCE) e Interface to the Routing System (IRS)). Suas principais funções são i) gerar eventos com mensagens sobre o comportamento da rede (e.g., mudança de porta ou enlace); ii) enviar estatísticas do fluxo de dados e características da rede e; iii) enviar pacotes do fluxo de dados para o controlador com a finalidade de decidir qual o comportamento de encaminhamento.

Segundo Costa (2013), o elemento central na arquitetura SDN é o controlador, responsável por disponibilizar um ambiente programático de controle onde os desenvolvedores podem ter acesso aos eventos gerados por uma interface de rede e gerar comandos para controlar o fluxo de dados nos dispositivos de encaminhamento através de suas interfaces programáticas NorthBound ou SouthBound [2]. As interfaces programáticas (API) possibilitam a implementação de políticas baseados em níveis de abstrações maiores, com a centralização da inteligência e lógica de roteamento no plano de controle onde estão armazenadas as tabelas de Bases de Informações de Roteamento, do inglês *Routing Information Base (RIB)*. Essas informações centralizadas permitem a realização da comparação dos fluxos de dados e a inclusão ou remoção de regras nas tabelas de encaminhamento, denominadas de *Forwarding Information Base (FIB)* localizadas nos equipamentos de comutação [14]. O primeiro controlador construido para o contexto SDN foi o NOX que motivou a implementação de vários outros controladores, a Tabela 1 lista os principais controladores de código livre disponíveis na comunidade atualmente.

Tabela 1: Principais controladores SDN open source.

NOME	ANO	LINGUAGEM	PLATAFORMA
NOX	2008	C++	Linux
POX	2010	Python	Windows, Mac e Linux
Maestro	2010	Java	Windows, Mac e Linux
Beacon	2010	Java	Windows, Mac, Linux e Android
RYU	2011	Python	Windows, Mac, Linux
FloodLight	2011	Java	Windows, Mac, Linux
OpenContail	2013	Python	Windows, Mac, Linux
OpenDayLight	2014	Java	Linux
ONOS	2014	Java	Windows, Mac, Linux

Outro importante componente da arquitetura SDN é o Agente, um *firmware* (software embutido no hardware) que deve ser instalado nos equipamentos de encaminhamento (EE) para habilitar a arquitetura SDN. A instalação desse software, possibilita aos EE's utilizarem um protocolo de comunicação para intermediar as mensagens entre os planos de dados e controle. As tabelas de fluxo de dados existentes nos EE's são formadas por estruturas com um conjunto de entradas (regras), junto a campos de cabeçalhos (*header*), instruções (ações) e campos estatísticos (número de bytes, duração do fluxo etc), definidos para manusear os pacotes do tráfego de dados [15], conforme Figura 2.

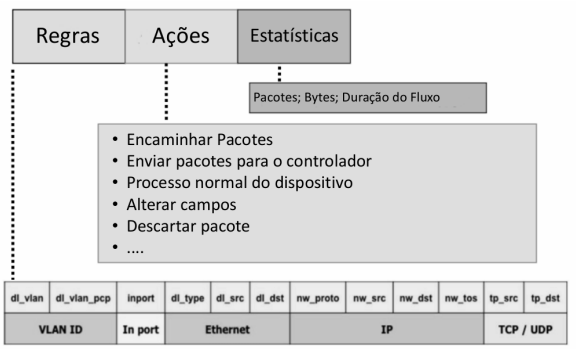

Figura 2: Definição de um fluxo na arquitetura OpenFlow
Fonte SBRC Tutorial OpenFlow, 2010

A definição de fluxo é representada pelo conjunto de ações que formam uma entrada da tabela de fluxos, como ilustrado na Figura 3. Os EE's quando em operação, analisam cada pacote que chega e comparam os cabeçalhos dos pacotes com as entradas das tabela de fluxos. Caso seja encontrado um casamento, considera-se que o pacote pertence àquele fluxo, então são aplicadas as ações existentes ou se por acaso, nenhuma regra for encontrada, o pacote é encaminhado para o controlador. A comunicação entre os EE (comutadores) e o controlador é determinada por um dos protocolos de comunicação da Southbound, que definem as regras que serão aplicadas aos fluxos nos dispositivos de comutação. Essa comunicação ocorre em um canal seguro de rede e pode empregar criptografia (SSL ou TLS) para garantir o sigilo no tráfego dos dados.

PORT	VLAN	MAC Src	MAC Dst	Eth Type	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Ação
*	*	*	00:1f ...	*	*	*	*	*	*	Porta 6
p3	vl1	00:20 ...	00:1f ...	0800	1.2.3.4	5.6.7.8	4	1765	80	Porta 6
*	*	*	*	*	*	*	*	*	22	Drop
*	*	*	*	*	*	5.6.7.8	*	*	*	Porta 6
*	vl1	00:1f. ..	*	*	*	*	*	*		Portas 6,7,8

Figura 3: Tabela de fluxo da arquitetura OpenFlow.

Dentro do contexto da SDN, o controlador realiza a verificação de sua tabela de fluxo e a notificação das aplicações acopladas ao controlador através da geração de eventos, que podem resultar na criação de uma nova entrada no comutador, fazendo assim com que os próximos pacotes, contendo o mesmo cabeçalho, sejam encaminhados pelo próprio comutador. Isto evita o reenvio dos pacotes para o controlador novamente, resultando em ganho de desempenho. Existem cinco ações básicas associadas a cada entrada na tabela de fluxos de um comutador programável: i) encaminhar os pacotes deste fluxo para uma determinada porta (ou portas); ii) encapsular e transmitir pacotes deste fluxo para um controlador e; iii) descartar pacotes deste fluxo; iv) alterar o conteúdo dos campos e; v) executar o processo normal do EE. Estas ações são realizadas através de trocas de mensagens de controle entre o controlador e o switch, de forma assíncronas e simétricas.

2.3 OpenDaylight (ODL)

O ODL é um projeto desenvolvido e apoiado pela *The Linux Foundation*, cujo objetivo é acelerar adaptação da tecnologia SDN através de uma plataforma modular e extensível. Essa organização é apoiada por várias empresas que comprometeram-se em prover ajuda economica e recursos técnicos ao desenvolvimento da plataforma (e.g., Arista Networks, Big Switch Networks, Brocade, Cisco, Citrix, Ericsson, HP, IBM, Juniper Networks, Microsoft, NEC, Nuage Networks, PLUMgrid, Red Hat and VMware). Além dos motivos comentados, o controlador ODL utiliza em sua arquitetura o *framework Open Services Gateway Initiative (OSGi)*, uma solução que permite a implantação de aplicações no formato de plugins (módulos) e disponibiliza a funcionalidade de encadeamento de serviço (SFC) através da API Northbound com o protocolo REST, acrônimo de *Representational State Transfer (REST)*, uma arquitetura para construir web services escaláveis através do protocolo HTTP. Essas características, possibilitam operadores de rede criar, atualizar e excluir cadeias de serviços, bem como especificar a troca de metadados entre os nós de um caminho de serviço através do uso de uma interface web.

Adicionalmente às características já comentadas, o controlador implementa uma camada de abstração de serviço, do inglês *Service Abstration Layer (SAL)*, que permite a comunicação entre os plugins através de mensagens de exportação que ignora o papel das APIs *Southbound* e *Northbound*, baseando-se na definição de plugins consumidor e provedor. Esta alteração significa que cada plugin dentro do ODL pode ser visto como um fornecedor ou consumidor, podendo trocar informações apenas pelo fluxo de mensagens entre os encaixes envolvidos. Os provedores são plugins que expõem recursos para aplicações e outros plugins através da sua API *Northbound*, os consumidores são componentes que fazem uso dos recursos fornecidos por um ou mais provedores.

A funcionalidade principal da SAL é descobrir como um serviço solicitado vai ser executado, independentemente do protocolo subjacente usado entre o controlador e os dispositivos de rede. Neste sentido, a realização do

controle dos dispositivos em seu domínio passa a saber apenas quais dispositivos estão na rede, suas capacidades e forma de acessibilidade, cujas informações são coletadas, armazenadas e gerenciadas pelo gerenciador de topologia. Com esta abordagem, os módulos da API Southbound ficam ligados dinamicamente na camada da SAL, possibilitando suporte a vários protocolos (e.g., OpenFlow 1.0, OpenFlow 1.3, OpenVSwitch, NetConf com Yang e BGP-LS). Essa arquitetura apresenta duas diferentes abordagens: *API-Driven SAL (AD-SAL)* e o *Model-Driven SAL (MD-SAL)*.

A abordagem AD-SAL realiza ações reativas através do recebimento de eventos da rede, às aplicações são acopladas ao controlador com o framework OSGi e o comportamento padrão é Stateless (não guarda estado). Já a abordagem MD-SAL possui uma programação proativa, sem a possibilidade de recebimento direto de eventos da rede e possui um modelo agnóstico que suporta apenas alguns dispositivos e modelos. Além disso o modelo de dados de estruturas das cadeias de servicos são definidas em arquivos e manipulados por linguagens de modelagem como por exemplo a linguagem YANG. Esta linguagem define a sintaxe e semântica na modelagem da camada de conteúdo sendo compatível com o modelo *Structure of Management Information (SMIv2)*.

2.4 Virtualização de Funções de Rede (NFV)

O grupo que lidera as especificacoes NFV é o *European Telecommunications Standards Institute (ETSI)*, através do grupo de trabalho *Industry Specifications Group (ISG)*, que trabalha para mover os serviços dos servidores para equipamentos comuns ou virtualizados. As motivações para a implantação dessa tecnologia são: i) reduzir os custos dos equipamentos e o consumo de energia, através da redução da quantidade de equipamento físicos para a disponibilização de serviços; ii) aumentar a velocidade de instanciação e disponibilização de novos serviços e; iii) virtualização dos recursos da rede de modo a atender um maior número de clientes.

Essa tecnologia permite criar funções de redes virtualizadas, do inglês *Virtualization Network Function (VNF)*, uma combinação de serviços dentro de um servidor de uso geral que disponibiliza funções de rede, do inglês *Network Function (NF)*. O termo NF, representa um conjunto de blocos funcionais dentro de uma infraestrutura de rede com comportamento bem definido [12]. A RFC-7665:2015 define o termo NF, como um nó de rede que tem a função de realizar o tratamento específico de pacotes recebidos (e.g., *Firewall, Dynamic Host Configuration Protocol (DHCP), Network Address Translation (NAT), Intrusion Detection System (IDS), Deep Packet Inspection DPI* etc).

As VNF, são implementadas dentro de uma Infraestrutura de Virtualização de Funções de Rede (NFVI), que inclui uma diversidade de recursos físicos para suportar a execução das NF. A camada de gerenciamento e Orquestração das NFV, do inglês *Network Function Virtualization Management and Orchestration (NFV-MANO)* é responsável por controlar o ciclo de vida e gestão física dos recursos. A arquitetura do NFVI possui três domínios: i) **Computacional**, inclui recursos de computação para VNFs através da camada de virtualização armazenados em storages (e.g., NAS, SAN); ii) **Virtualização**, camada que abstrai os recursos de hardware e dissocia o VNF do hardware. Normalmente, este tipo de funcionalidade é fornecida por hypervisors. iii) **Infra-estrutura de rede**, um domínio que inclui tanto os recursos de redes físicos e virtuais para interligar os recursos de computação dentro da NFVI, visualizar Figura 4.

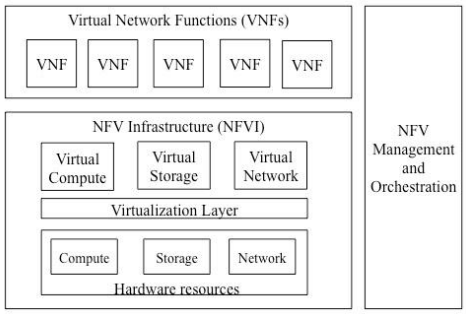

Figura 4: Arquitetura Network Function Virtualization (NFV)
Fonte ETSI ISG 4G Americas, 2014.

2.5 Trabalhos relacionados

John et al. (2013) abordaram em seu artigo as vantagens da adaptação do conceito NSC e as direções de pesquisa para o encadeamento de serviço. Detalham também os principais aspectos a serem considerados durante o ciclo de vida típico de uma cadeia de serviços, no contexto das redes de telecomunicações [4]

Os pesquisadores Li and Chen (2015) apresentaram uma investigação aprofundada sobre o desenvolvimento de NFV e sob a arquitetura NFV-Definida por Software, com ênfase no encadeamento de serviço como uma aplicação. Primeiro, introduziram a arquitetura NFV definida por software como o estado da arte e as relações atuais entre NFV e SDN. Com isso, fornecem uma visão histórica do envolvimento de middlebox com NFV, além de apresentar os desafios mais significativos e soluções relevantes para a NFV [16].

Brown (2015) apresentou uma avaliação sobre o porque cadeias de serviço são úteis e como operadoras podem habilitar a criação de serviços dinamicamente com a utilização de equipamentos dedicados (appliance) e funções de redes virtualizadas. Em sua análise ele aplicou várias opções de implementação de cadeias de serviço, abordando especificamente o papel do classificador, opções de cabeçalho do pacote e a importância dos metadados [17].

Ruckert (2015) implementou uma prova de conceito com uma abordagem de encadeamento de serviço totalmente baseada em software para cadeias de serviço. O protótipo é executado no padrão hardware e inclui um conjunto de funções de rede como exemplo para mostrar a exequibilidade da abordagem. Além disso, ele utiliza uma API de alto nível para integrar um operador e um portal de auto-serviço ao cliente [18].

3 ENCADEAMENTO DE SERVIÇOS EM REDE (NSC)

Hoje a disponibilização de serviços nas redes de computadores exige a configuração e instanciação de funções de rede manualmente, essa demanda envolve a intervenção humana e por este motivo está propenso a ocorrência de erros e a uma maior carga de trabalho pela grande quantidade de dispositivos que necessitam ser configurados. A iniciativa de integrar as tecnologias SDN e NFV permite aproveitar o que há de melhor em ambas arquiteturas para configuração e reconfiguração a rede através de um controlador central, bem como a utilização de toda a capacidade de virtualização de máquinas e serviços, como pode ser visto na Figura 5.

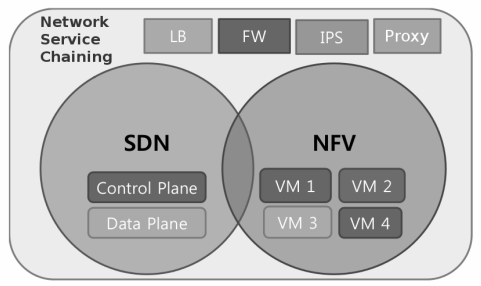

Figura 5: Integração das arquiteturas SDN e NFV para o conceito NSC

A integração das tecnologias SDN e NFV possibilita a utilização de uma nova abordagem conhecida como Encadeamento de Serviços em Rede, do inglês *Network Service Chaining (NSC)*, que implementa uma abstração em nível de plano de serviço ao lado do plano de controle para direcionar o fluxo de dados dinamicamente através dos gráficos de serviços criados. Segundo John et al. (2013), o encadeamento dinâmico de serviço é definido como um processo de entrega contínua de serviços com base em associações de função de rede. Neste contexto, a entrega contínua significa uma orquestração dinâmica das funções de rede para a implantação automática de melhorias na eficiência operacional, recuperação de falhas, bem como na capacidade da realização de testes [4].

3.1 Cadeias de Função de Serviço (SFC)

Como discutido nas seções 2.2 e 2.3, a integração das soluções existentes na nova geração das redes (SDN e NFV), habilita a criação de uma nova técnica para criação de Cadeias de Funções de Serviço, do inglês *Service Function Chains (SFC)*, com a finalidade de criar topologias em um gráfico de caminhos contendo funções de serviços em rede, como pode ser visto na Figura 6.

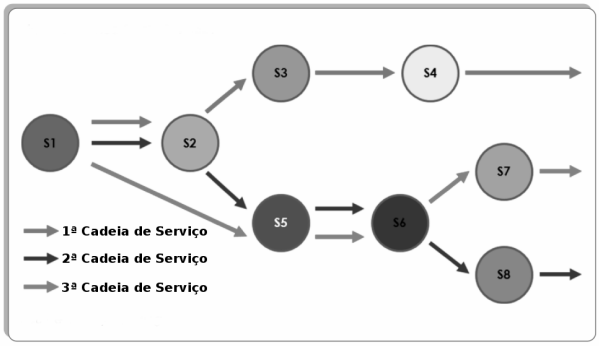

Figura 6: Exemplo de gráfico com funções de serviço
Fonte Heavy Reading, 2014

A RFC-7665:2015 define SFC como um conjunto ordenado de abstrações de Funções de Serviços (SF) que devem ser aplicadas aos fluxos de pacotes selecionados como resultado de uma classificação. Em suma, são funções virtuais do tipo LoadBalancer, Firewall, IDS/IPS, NAT, QoS, Proxy, DPI etc [19].

Cada uma dessas SFs podem ser realizadas através de um serviço dedicado e são atravessadas em uma dada ordem (1). A conectividade entre as SFs podem ser representadas por um gráfico conhecido como Gráfico de Função de Rede, do inglês *Network Function Graphic (NFG)*, sendo definido em uma formulação matemática de decomposição dos serviços, representada por um mapeamento de cada NF em um conjunto de NFGs (2).

$$Classificador \longrightarrow SFF \longrightarrow S1 \longrightarrow S2 \longrightarrow S3 \longrightarrow S4 \tag{1}$$

$$NFi \longrightarrow \{NFGi1, NFGi2, ...\} \tag{2}$$

Os principais componentes da SFC são o classificador, o plano de controle e os encaminhadores de funções de serviço. O classificador, do inglês *Classifier*, possui habilidade para identificar e classificar o tráfego antes de encaminhar para processamento pelo gráfico de serviço. O plano de controle é responsável por definir a topologia, políticas e o caminho do fluxo pelo gráfico de serviço e os Encaminhadores de Funções de Serviço, do inglês *Service Function Forwarder (SFF)*, tem a função de determinar o destino do tráfego.

Nessa arquitetura um Controlador SDN (*Controller*) gerencia o Classificador e os SFF através do Plano de Controle para atualizar as tabelas de fluxo dos EE. Neste sentido, o classificador (*Classifier*) é colocado no início do tráfego e define as cadeia de serviços com a inclusão de uma etiqueta no cabeçalho do pacote chamado de Cabeçalho de Serviço de Rede, do inglês *Network Service Header (NSH)*, essa etiqueta determina o tipo de serviço a ser aplicado ao fluxo. O "Serviço"diz respeito ao tipo de SF e à ordem através da qual o fluxo passa por elas, neste sentido a implementação de cadeias de serviços pode ser realizada a partir de um ou mais serviço das camadas 4 a 7 do modelo OSI.

As cadeias de serviços são identificadas por um identificador (ID) único e sua estrutura é alocada em um plano de serviço para posterior consulta pelos SFF. Novas funções de serviço, baseadas em políticas e metadados de negócio existente no plano de controle podem ser adicionadas, removidas ou alteradas quando necessário, basta simplesmente mudar a etiqueta para obter uma nova ramificação da cadeia de acordo com os resultados do processamento da função de serviço [17]. Os SFF representam instâncias virtuais colocados na rede para analisar a etiqueta e determinar o destino do fluxo. Sua finalidade é fornecer serviço de transporte às cadeias de serviços. Neste ponto, os pacotes do fluxos de dados são encapsulado e enviados através de uma rede virtual dedicada a esse conjunto de funções de serviço. A Figura 7 exibe o fluxo da SFC.

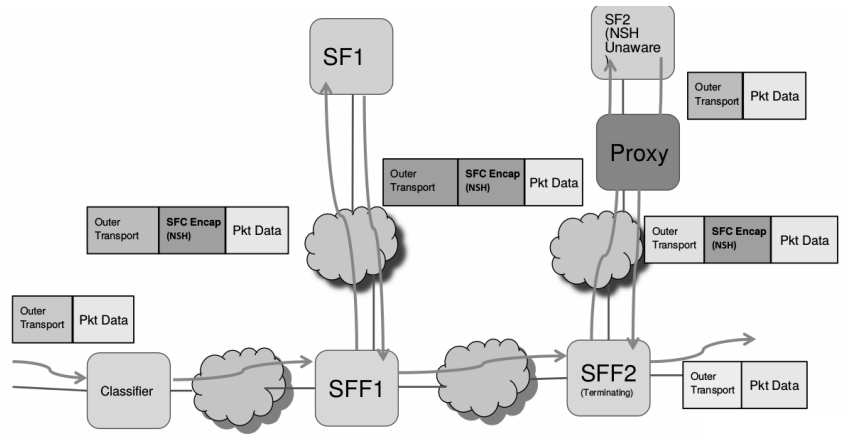

Figura 7: Encadeamento de Funções de Serviço (SFC)
Fonte OpenDayLight.org, 2015

3.2 Encapsulamento NSH

O encaminhamento dos pacotes pelo gráfico de funções de serviço é realizado com a inclusão de uma etiqueta no cabeçalho do pacote que identifica os tipos de funções de serviço e a ordem que os pacotes são aplicadas ao fluxo, sendo os pacotes transferidos com base nesta etiqueta. No documento-rascunho DRAF-SFC-NSH-04, de 13 de março de 2016, é proposto o protocolo NSH como a forma padrão de encapsulação da técnica SFC, o documento descreve com detalhe o formato do cabeçalho a ser inserido nos pacotes ou quadros do fluxo de dados para encapsular caminhos de Funções de Serviço, além de detalhar o mecanismo para troca de metadados ao longo dos serviços instanciados. A documentação apresenta a criação de um plano de serviço com configurações e topologias de caminhos de serviço independentes para transportar os pacotes encaminhados sem alterar a topologia da rede principal [20].

Dessa forma a formatação dos campos do protocolo NSH foram divididas em 3 seções: i) base do cabeçalho, fornece informações sobre o cabeçalho do serviço e o protocolo de carga útil (payload); ii) cabeçalho de caminho do serviço, representa a identificação e localização dentro do caminho de serviço e; iii) cabeçalhos de contexto, carregam informação dos metadados e do comprimento variável das informações codificadas, como pode ser visto na Figura 8.

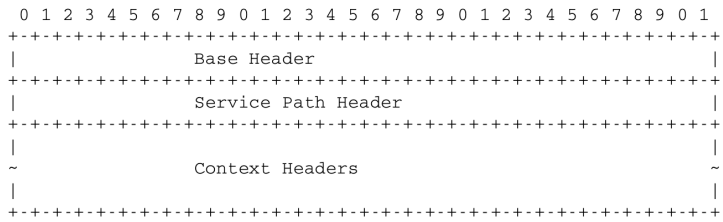

Figura 8: Seções do Network Service Header (NSH),
Fonte: IETF Draf-sfc-nsh-04, 2016

- O base do cabeçalho possui o tamanho padrão de 04 (quatro) bytes, o que corresponde a 32 bits contendo os seguintes campos: i) 02 bits para a versão do protocolo; ii) 08 bits com flags opcionais, do tipo TLVs.; iii) 06 bits para representar o tamanho total do cabeçalho NSH; iv) 08 bits representando o tipo de domínio mandatorio; v) 08 bits com a indicação do próximo protocolo.

- O cabeçalho de caminho do serviço também possui o tamanho de 04 (quatro) bytes com os seguintes campos: i) 24 bits com o identificador para seleção de caminhos de serviço (SPI); ii) 08 bits para armazenar o index do serviço.

- Os cabeçalhos de contexto possui tamanho variável dependendo do valor do campo MD-type da seção do cabeçalho base: i) se o valor for 0x1, o formato terá 4 seções de 32 bits cada contendo informações dos contextos e; ii) se o valor for 0x2, o tamanho do cabeçalho será variável podendo ser zero ou mais cabeçalhos.

Os 02 primeiros bits do cabeçalho base identificam a versão do cabeçalho, os 08 bits de Flags (sinalizadores) seguintes são usados para definir as alterações de cabeçalho ou comportamento de análise. Até o momento, dois sinalizadores estão definidos e os seis restantes estão reservados: i) o bit O: quando definido representa que o pacote é tipo de operação, administração e gestão (OAM); ii) o bit C: indica a presença de um metadados crítico TLV. Os próximos 06 bits representam o tamanho total do cabeçalho NSH (Length).

O campo MD Type é composto por 08 bits e define o tipo de cabeçalho de contexto que armazena a seção de Cabeçalho de Contexto. Se possuir o valor 0x1 o cabeçalho de contexto será configurado com 4 cabeçalhos de 04 bytes cada. Caso o valor seja 0x2, nenhum ou cabeçalho de contexto de tamanho variado pode ser adicionado. O último campo do cabeçalho base é o próximo protocolo que também possui 08 bits com a indicação do tipo do próximo protocolo do pacote original. Existem 03 tipos definidos como 0x1 para IPV4, 0x2 para IPv6 e 0x3 para Ethernet, a Figura 9 abaixo ilustra o formado do NSH.

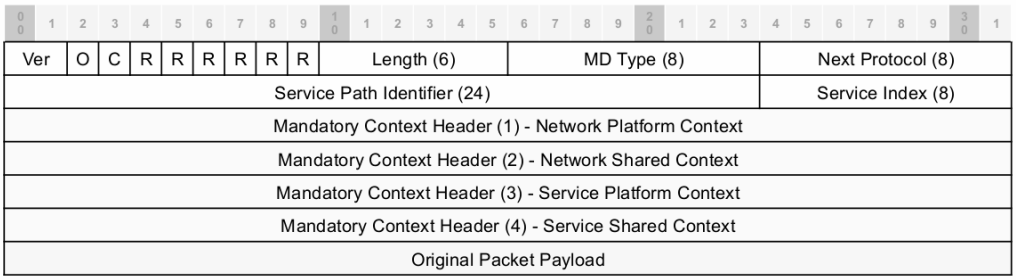

Figura 9: Formato detalhado do Network Service Header (NSH),
Fonte: Cisco Corporation, 2016

Na seção do caminho do serviço o campo Identificador de Caminho do Serviço, do inglês *Service Path Identifier (SPI)*, possui 24 bits e identifica o caminho que interconecta a função de serviço. Este prover um nível de abstração seguindo os nós para selecionar o protocolo de transporte de rede apropriado e a técnica de encaminhamento. O próximo campo desta seção é o Índice de Serviço (SI) com 08 bits, que trata da localização do pacote dentro do caminho de serviço, e quando este é atravessado o SI é decrementado. A combinação do SPI com o SI prover uma clara visibilidade de onde o pacote está dentro do caminho de serviço.

Os cabeçalhos de contextos são constituídos de 16 bytes (128 bits) obrigatórios, definidos como campos de cabeçalhos de contexto para armazenamento de informações de metadados entre os nós de classificação, funções de serviço e os nós da rede. A informação de classificação é exportada dentro do cabeçalho NSH para prover simplicidade de implementação, políticas de funções de serviço e informações de contexto. Os metadados refletem o resultado de uma classificação externa ou antecedente, essa informação de classificação é então provida via NSH por campos do contexto participante da política de decisão local contendo contexto da plataforma e compartilhamento de rede, além do contexto de plataforma e compartilhamento do serviço.

Definida a estrutura do cabeçalho NSH, o próximo passo é a realização do transporte dentro do contexto do plano de serviço, nessa ação pode ser usado qualquer método padrão da indústria que implemente a virtualização de rede, como o *Virtual Extensible LAN (VxLAN)*, Ethernet, *Generic Protocol Extension (GPE)*, *Generic Routing Encapsulation (GRE)* ou a combinação de mais de uma técnica (VxLAN-GPE).

4 IMPLEMENTAÇÃO DO PROTÓTIPO DE ENCADEAMENTO DE SERVIÇO

Nesta seção será demonstrada a implementação do conceito de encadeamento dinâmico de funções de serviço em redes de computadores através da emulação de um cenário virtual como caso de teste. Os softwares selecionados para a implementação foram: a distribuição ODL-SFC e o emulador de rede MiniNet, ferramentas

que possibilitam criar, interagir e customizar protótipos de forma rápida e programável, além de permitirem a realização de testes para depuração e resoluções de problemas.

A distribuição selecionada do projeto ODL é a Beryllium SR2 versão 0.42, com suporte a SFC. Essa distribuição é forneceida dentro de um container chamado de "Karaf", um projeto da fundação Apache baseado em um dos framework OSGi (Equinox ou Felix) para prover características adicionais aos containers. O pacote de distribuição do ODL vem inicialmente com o núcleo principal do controlador e novas características são instaladas sob demanda dentro do container como plugins extensíveis. O processo de instalação das características necessárias para o expererimento deste trabalho consiste na inclusão dos seguintes módulos do projeto ODL: old-restconf; old-mdsal-clustering; old-dlux-core; old-dlux-node; old-dlux-yangui; old-dlux-yangvisualizer; old-l2switch-switch; old-sfc-sb-restt e; old-sfc-ovs.

O Mininet foi utilizado para emular os links, hosts, switches e controladores, reproduzindo processos que executam em espaços de nomes da rede (network namespaces), tendo como principais características: i) fornecer uma maneira simples para a realização de testes em redes com o OpenFlow; ii) permitir a realização múltiplas pesquisas de forma independente; iii) realizar teste em uma topologia grande e complexa, sem a necessidade de uma rede física; iv) incluir ferramentas para depurar e executar testes em toda a rede e v) suportar inúmeras topologias.

O caso de teste para avaliação foi modelado inicialmente com a feramenta CORE e posteriomente implementada no emudador Mininet. A topologia da rede é constituída de equipamentos existentes em três órgãos administrativos na cidade de Teresina, interligados por uma rede metropolitana sem fio, baseada no padrão IEEE 802.16, do inglês *Worldwide Interoperability for Microwave Access (WiMAX)* e conectados a uma rede de fibra óptica no ponto de presença da operadora (Intranet), que estabelece um canal isolado até o ponto de presença na cidade de Fortaleza, como pode ser visualizado na Figura 10.

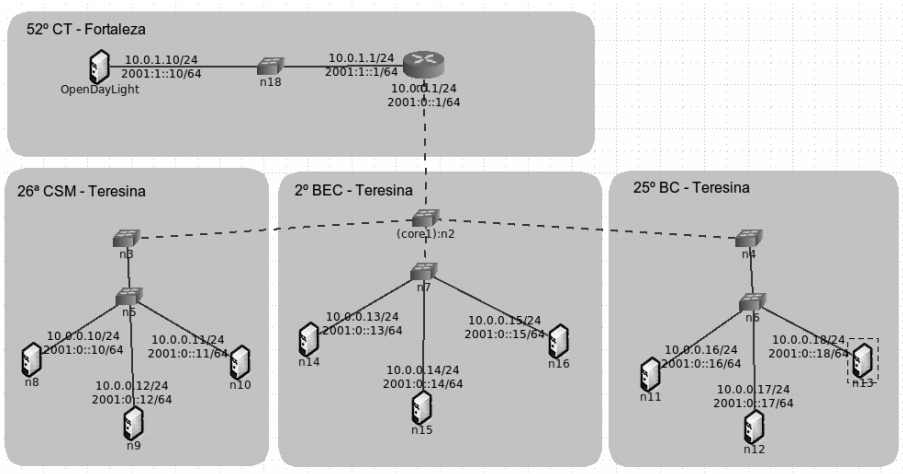

Figura 10: Contexto do caso de teste, autoria própria

A finalidade do caso de teste consiste em criar cadeias de serviços remotamente e de forma dinâmica nos dispositivos de encaminhamento de dados das unidades envovidas, e assim, direcionar o fluxo de dados aos diferentes tipos de serviços de acordo com as requisições das estações clientes existentes ao longo da rede, principalmente ao fluxo destinado aos servidores de conteúdo. As principais ações a serem implementadas são: i) realizar a comutação dos dados nas unidades remotas a partir de políticas e regras demandadas do controlador central; ii) realizar uma análise profunda do tráfego de dados para identificar *malwares* e ações de hackers e iii) configurar remotamente os equipamentos e serviços para adequação as técnicas de transição do protocolo IPv4 para o IPv6.

O processo de implementação do encadeamento dos serviços através da distribuição ODL-SFC se inicia com a instanciação do controlador e da interface web de gerenciamento e o próximo passo consiste na inicialização do agente SFC, uma aplicação desenvolvida em Phyton que tem a funcionalidade de receber requisições do controlador e criar, dentro do plano de dados, às Funções de Serviço (SF) e os Encaminhadores de Funções de Serviço (SFF), bem como estabelecer uma relação entre os Caminhos de Funções de Serviços (SFP) e os Caminhos de

Serviços Renderizados (RSP). As informações para criação dos componentes são enviados para o controlador ODL através da solução JSON que transportar os campos de configurações com seus respectivos dados até a próxima camada. A Figura 11 ilustra as cadeias de serviços criadas a partir das funções de serviços existentes.

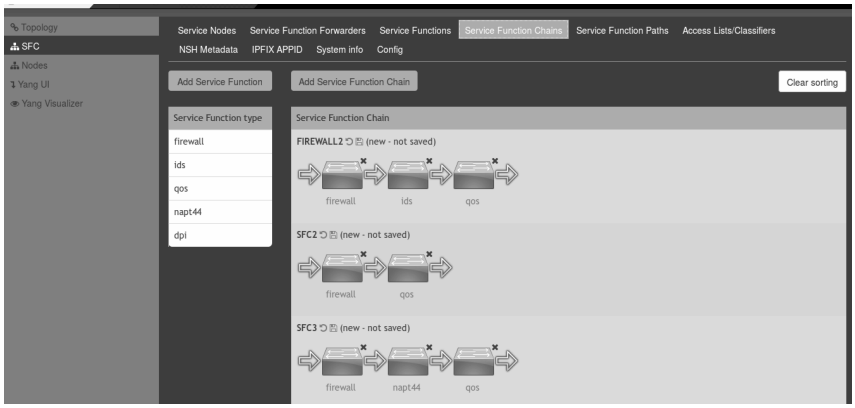

Figura 11: Interface de gerenciamento da Cadeia de Serviço, autoria própria

Depois de criadas às SF e os SFF é necessário criar uma relação ordenada de encadeamento entre as SF chamadas de cadeias de serviços, que poderão ser associadas a um plano de serviço em um SFF para efetivamente estabelecer o caminho de encadeamento entre as funções de rede e o fluxo de dados. A verificação do processo de encadeamento dos serviços na rede foi simulada com a inicialização de uma aplicação cliente no contexto do emulador Mininet e a entrada do comando (3). O fluxo produzido pelos pacotes passando pela cadeia de serviços é registrado nos logs do agente SFC, como pode ser visualizado na Figura 12.

$$> sff\_client.py\ --remote-sff-ip\ 10.0.1.4\ --remote-sff-port\ 4789\ --sfp-id\ 1 \qquad (3)$$

$INFO$:$common.services$:$SFFServer$:$Receivedapacketfrom$:('10.0.1.4',5000)
$INFO$:$common.services$:$SFFServer$:$Processingpacketfrom$:('10.0.1.4',5000)
$INFO$:$common.services$:$SFFServer$:$Sendingpacketsto$:('10.0.1.4',40001)
$INFO$:$common.services$:$firewallservicereceivedpacketfrom$('10.0.1.4',4789):
$INFO$:$common.services$:$firewall$:$Processingreceivedpacket$
$INFO$:$common.services$:$firewall$:$Sendingpacketsto$('10.0.1.4',4789)
$INFO$:$common.services$:$SFFServer$:$ListeningforNSHpackets...$
$INFO$:$common.services$:$idsservicereceivedpacketfrom$('10.0.1.4',4789):
$INFO$:$common.services$:ids:$Processingreceivedpacket$
$INFO$:$common.services$:ids:$Sendingpacketsto$('10.0.1.4',4789)
...
$INFO$:$common.services$:$SFFServer$:$Finishedprocessingpacketfrom$:('10.0.1.4',5000)
$INFO$:$common.services$:$SFFServer$:$Endofpath$

Figura 12: Captura do fluxo de encadeamento no agente SFC, autoria própria

5 CONCLUSÃO

Frente as soluções tradicionais para disponibilização de serviços nas redes de telecomunicações, os resultados obtidos durante os experimentos de integração das soluções SDN e NFV com o controlador ODL-SFC, demonstram uma maior abstração no gerenciamento e uma diminuição da complexidade de configuração dos dispositivos com ganhos significativos no tempo de disponibilização dos serviços. Essa agilidade melhora o gerenciamento em dois aspectos: primeiro, cadeias de serviços podem ser facilmente criadas a partir de funções de serviços existentes e em segundo lugar, os recursos de redes usados para entregar uma SF podem ser alocados

dinamicamente durante sua execução. Neste sentido, o emprego dessa nova abordagem possibilita uma forma centralizada e dinâmica para minimizar os custos de operação e otimizar o provimento de serviços em ambientes de *data centers*, *cloud computing* e *Internet Service Provider (ISP)*.

Como trabalho futuro é proposta a realização de estudos para aplicação da solução NSC nos dispositivos na ponta da linha das redes, conhecidos como equipamentos dentro das instalações do cliente, do inglês *Customer Provided Equipment (CPE)*, que atualmente também realizam funções de rede (e.g., roteamento, firewall, DHCP, NAT e comutação) semelhantes aos ambientes citados acima. Esta nova abordagem, objetiva adicionar ou atualizar as função de rede nos CPE's remotos, principalmente no período atual de transição dos protocolos IPv4 para o IPv6, onde existe a necessidade em aplicar técnicas de transição (e.g., pilha dupla, túneis e tradução) para manter o acesso aos serviços nos dois protocolos simultaneamente.

Agradecimentos

Aos professores da especialização em redes de computadores da Faculdade Santo Agostinho (FSA), Alberto Viegas, Amélia Acácia de Miranda Batista, Humberto Caetano Cardoso da Silva, Márcio de Melo Souza, Paulo Perris e Ricardo Almeida, por toda dedicação e motivação na transmissão do conhecimento para a formação profissional de seus alunos. E aos meus orientadores, Ricardo Gomes Queiroz e Christian Esteve Rothenberg, pela paciência, confiança e empenho que aplicaram na realização deste trabalho.

Referências

[1] SIMõES, G. *A transformação da cloud e as inovações trazidas pela IoT mudam o cenário do setor de telecomunicações*. [S.l.], 2016.

[2] COSTA, L. R. Openflow e o paradigma de redes definidas por software. *Monografia de Graduação ? Instituto de Ciências Exatas ? Departamento de Ciência da Computação - Universidade de Brasília.*, Universidade de Brasília, May 2013.

[3] MCKEOWN, N. et al. Openflow: Enabling innovation in campus networks. 2008.

[4] JOHN, W. et al. Research directions in network service chaining. 2013.

[5] FARIAS, F. N. N. et al. Pesquisa experimental para a internet do futuro: Uma proposta utilizando virtualização e o framework openflow. *XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC 2011*, SBRC 2011, v. 35, n. 15, p. 7, 2011.

[6] ZHANG, H. Clean slate design approach to networking research. May 2005.

[7] CARPENTER, B. *Middleboxes: Taxonomy and Issues*. [S.l.], 2002.

[8] FEAMSTER, N.; REXFORD, J.; ZEGURA, E. The road to sdn: An intellectual history of programmable networks. *Queue*, v. 11, n. 12, p. 20:20?20:40, Dec 2013.

[9] WATSON, G.; MCKEOWN, N.; CASADO, M. The netfpga project. 2007.

[10] ASTUTO, B. N. et al. A survey of software-defined networking: Past, present, and future of programmable networks. 2013.

[11] CASADO, M. et al. Sane: A protection architecture for enterprise networks. 2006.

[12] GORANSSON, P. *Software Defined Networks A Comprehensive Approach*. 1th. ed. [S.l.]: Elsevier, 2014.

[13] BOUCADAIR, M.; JACQUENET, C. *Software-Defined Networking: A Perspective from within a Service Provider Environment*. [S.l.], 2014, 1-56 p.

[14] NADEU, T. D.; GREY, K. *SDN: Software Defined Networking*. 1th. ed. [S.l.]: O'Reilly, 2013.

[15] SANCHEZ, A.; SZARKOWICZ, K. G. *MPLS in the SDN Era*. 1th. ed. [S.l.]: O'Reilly, 2015.

[16] LI, Y.; CHEN, M. Software-defined network function virtualization: A survey. *IEEE Access*, September 2015.

[17] BROWN, G. Serive chaining in carrier networks. 2015.

[18] RUCKERT, J. et al. Demo: Software-defined network service chaining. 2015.

[19] HALPERN, J.; PIGNATARO, C. *Service Function Chaining (SFC) Architecture*. [S.l.], 2015.

[20] QUINNG, P.; ELZUR, U. *Network Service Header (NSH) Context Header Allocation (Network Security)*. [S.l.], 2016.