UNIVERSIDADE FEDERAL DE SERGIPE

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Virtualização de Funções de Rede em Nuvem para Instituições Públicas

Dissertação de Mestrado

Lucio da Silva Gama Junior

São Cristóvão – Sergipe

2017

UNIVERSIDADE FEDERAL DE SERGIPE

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Lucio da Silva Gama Junior

Virtualização de Funções de Rede em Nuvem para Instituições Públicas

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Orientador(a): Prof.ª Dr.ª Edilayne Meneses Salgueiro
Coorientador(a): Prof. Dr. Ricardo José Paiva de Britto Salgueiro

São Cristóvão – Sergipe

2017

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL UNIVERSIDADE FEDERAL DE SERGIPE

Gama Junior, Lúcio da Silva
M527a Virtualização de funções de rede em nuvem para instituições públicas / Lúcio da Silva Gama Junior; orientador Edilayne Meneses Salgueiro. – São Cristóvão, 2017.
86 f. : il.

Dissertação (mestrado em Ciências da computação) – Universidade Federal de Sergipe, 2017.

1. Internet - Programas de computador. 2. Internet das coisas. 3. Redes de computação. 4. Computação em nuvem. I. Salgueiro, Edilayne Meneses, orient. II. Título.

CDU: 004.738.5

Lucio da Silva Gama Junior

Virtualização de Funções de Rede em Nuvem para Instituições Públicas

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Trabalho aprovado. São Cristóvão – Sergipe, 28 de agosto de 2017:

Prof.ª Dr.ª Edilayne Meneses Salgueiro
Orientador

Prof. Dr. Ricardo José Paiva de Britto Salgueiro
Coorientador

Prof. Dr. Tarcísio da Rocha
Convidado

Prof. Dr. Paulo Romero Martins Maciel
Universidade Federal de Pernambuco (UFPE)

São Cristóvão – Sergipe
2017

Dedico este trabalho a Deus, aos meus pais, irmãos, esposa, filha e a toda a minha família e meus amigos que, com muito carinho e apoio, me fortaleceram para que eu chegasse até esta etapa da minha vida.

Agradecimentos

Agradeço primeiramente a DEUS, por ter me dado a permissão de chegar até aqui, e por toda a força concedida na concretização desse sonho.

À Professora Drª. Edilayne Meneses Salgueiro, orientadora desta dissertação, pela colaboração, paciência, pelo seu incentivo, disponibilidade e apoio que sempre demonstrou. Aqui lhe exprimo a minha gratidão.

Ao co-orientador Professor Drº. Ricardo José Paiva de Britto Salgueiro, pela sua disponibilidade nos trabalhos de campo, pelo seu incentivo, pela sua disponibilidade e igualmente pelo seu apoio na elaboração deste trabalho.

À Universidade Federal de Sergipe (UFS) por me proporcionar um aperfeiçoamento gratuito e de qualidade.

Ao Grupo de Pesquisa em Redes e Computação Distribuída da UFS pelo cessão do equipamentos imprescindíveis para a realização desta dissertação.

Aos colegas de mestrado Itauan Silva Eduão Ferreira e Wesley Oliveira pela paciência e disponibilidade, pelos ensinamentos e trocas de conhecimentos.

Ao Instituto Federal de Sergipe (IFS) por ter permitido a minha participação nesse programa de mestrado.

Aos professores do IFS/Campus Itabaiana pelo companheirismo e apoio recebido.

Aos meus pais, Lucio e Yedda, minha esposa Pollyana e minha filha Giovanna pelo apoio incondicional que me deram ao longo da elaboração deste trabalho. Obrigado pelo amor!

À minha família, meus irmãos Glauber e Ivana, meus sogros Paulo e Selma, meus sobrinhos Andressa, Rafael, Guilherme e Pablo Henrique, meus cunhados Rose, Pablo e Aline que sempre se orgulharam de mim e confiaram em meu trabalho.

Aos meus amigos pelos incentivos e apoios.

À Fapitec, pelo apoio concedido que foi de fundamental importância para o desenvolvimento deste trabalho.

Sou um pouco de todos que conheci,
um pouco dos lugares que fui,
um pouco das saudades que deixei
e sou muito das coisas que gostei.
(Antoine de Saint-Exupéry)

Resumo

Virtualização de funções de rede (NFV) é uma nova tendência na rede, onde as funções de rede estão passando de dispositivos de *hardware* personalizados para implementações de *software* em máquinas virtuais hospedadas em *hardware* comuns. Isso se deve ao fato que em uma rede tradicional, todas as funções são implementadas em *hardware* e *software* específicos, que empregam tecnologia proprietária, de difícil interoperabilidade, configuração e operação, tornado-a inadequada para atender as crescentes demandas da Internet, que impulsionam a transformação da sua infraestrutura. NFV e computação em nuvem são tecnologias complementares que oferecem novas soluções para projetar, construir e operar redes, possibilitando a coexistência de múltiplas redes, agora virtuais, sobre uma mesma infraestrutura física. Apoiada nos benefícios que a computação em nuvem proporciona, as redes podem ser organizadas de forma mais flexível, com programabilidade e dinamismo de suas configurações. O objetivo deste trabalho é propor uma estratégia de virtualizar funções de rede em um ambiente de nuvem, permitindo assim que organizações, especialmente públicas, foco desse trabalho, possam aproveitar os paradigmas que o NFV e a computação em nuvem, podem provocar na infraestrutura de seus *Data Centers*. Para tanto, foi construído um ambiente de experimentação em nuvem a fim prover a realização de testes de rede de computadores evitando assim que esses testes em uma rede ativa e em produção, comprometessem o seu funcionamento normal. Neste trabalho, estão detalhadas a implantação de uma plataforma de computação em nuvem de código aberto, o *OpenStack*, como ela pode ser utilizada para implantar o NFV e como ambos podem contribuir na criação de soluções aptas a atender as demandas originadas de infraestruturas de rede tradicional. Para validar o ambiente de experimentação, foi realizado um estudo de caso da migração de uma rede tradicional para uma rede NFV utilizando os protocolos IPv4 e IPv6 em um ambiente de rede de instituições públicas. Os resultados alcançados permitiram constatar o quanto o ambiente criado é uma ferramenta importante para a realização de experimentos que permitam aos administradores de redes planejar, por exemplo, o período de transição e coexistência desses protocolos, possibilitando que essa mudança nas redes de computadores possa ocorrer de forma mais suave.

Palavras-chave: Virtualização da Funções de Rede (NFV), Computação em Nuvem, Desempenho, OpenStack.

Abstract

Network Function Virtualization (NFV) is a new trend in the network, where network functions are shifting from custom hardware devices to software deployments in virtual machines hosted on common hardware. This is due to the fact that in a traditional network, all functions are implemented in specific hardware and software, employing proprietary technology, difficult interoperability, configuration and operation, rendering it inadequate to meet the increasing demands of the Internet, which transformation of its infrastructure. NFV and cloud computing are complementary technologies that offer new solutions for designing, building and operating networks, allowing the coexistence of multiple, now virtual, networks on the same physical infrastructure. Backed by the benefits of cloud computing, networks can be organized more flexibly, with programmability and dynamism in their configurations. The objective of this work is to propose a strategy to virtualize network functions in a cloud environment, thus allowing organizations, especially public, to focus on this work, to take advantage of the paradigms that NFV and cloud computing can provoke in the infrastructure of their Data Centers. In order to do so, a cloud experimentation environment was built in order to provide computer network testing, thus preventing those tests in an active network and in production, compromising its normal operation. In this work, we detail the deployment of an open source cloud computing platform, OpenStack, how it can be used to implement the NFV, and how both can contribute to the creation of solutions able to meet the demands of traditional network infrastructures . To validate the experiment environment, a case study was carried out of the migration of a traditional network to an NFV network using the IPv4 and IPv6 protocols in a network environment of public institutions. The results obtained showed that the environment created is an important tool for conducting experiments that allow network administrators to plan, for example, the transition period and coexistence of these protocols, allowing that this change in computer networks can occur way.

Keywords: Network Functions Virtualization (NFV), Cloud Computing, Performance, OpenStack.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

API	*Application Programming Interface*
APNIC	*Asia-Pacific Network Information Center*
ARIN	*American Registry for Internet Numbers*
BD	Banco de Dados
BIND	*Berkeley Internet Name Daemon*
CAPEX	*Capital Expenditures*
CEO	*Chief Executive Officer*
CF	Constituição Federal
CIDR	*Classless Internet Domain Routing*
CORE	*Common Open Research Emulator*
vCPU	*virtual Central Processing Unit*
CTCC	*Centre Tecnològic Telecomunicacions Catalunya*
DC	*Datacenter*
DHCP	*Dynamic Host Configuration Protocol*
DNS	*Domain Name System*
DPI	*Deep Packet Inspection*
ELAN	*Experimental LAboratory in computer Networks*
EMGETIS	Empresa Sergipana de Tecnologia da Informação
ETSI	*European Telecommunications Standards Institute*
GENI	*Global Environment for Network Innovations*
GMPLS	*Generalized Multi Protocol Label Switching*
GNU	*General Public License*
GPRCom	Grupo de Pesquisa em Redes e Computação Distribuída
IaaS	*Infrastructure as a Service*

IANA	*Internet Assigned Numbers Authority*
IBM	*International Business Machines*
IDS	*Intrusion Detection System*
IMUNES	*Integrated Multi-protocol Network Emulator/Simulator*
IP	*Internet Protocol*
IPS	*Intrusion Prevention System*
ISG NFV	*Industry Specification Group for Network Functions Virtualization*
LACNIC	*Latin American and Caribbean IP Address Regional Registry*
MBPS	*Megabits per Second*
MS	*Millisecond*
NASA	*National Aeronautics and Space Administration*
NAT	*Network Address Translation*
NBAPI	*Northbound Application Programming Interface*
NIC.br	Núcleo de Informação e Coordenação do Ponto BR
NIST	*National Institute of Standards and Technology*
NFV	*Network Functions Virtualization*
NFVI	*Network Functions Virtualization Infrastructure*
NV	*Network Virtualization*
OPEX	*Operational Expenditure*
ONF	*Open Networking Foundation*
PaaS	*Platform as a Service*
QoS	*Quality of Service*
RFC	*Request for Comments*
RIPE NCC	*Réseaux IP Européens Network Coordination Center*
RIR	*Regional Internet Registry*
SaaS	*Software as a Service*

SBAPI	*Southbound Application Programming Interface*
SDN	*Software-Defined Networking*
SF	*Service Function*
TI	Tecnologia da Informação
TCP/IP	*Transmission Control Protocol/Internet Protocol*
UDP	*User Datagram Protocol*
UML	*User Mode Linux*
VLAN	*Virtual Local Area Network*
VM	*Virtual Machine*
VMM	*Virtual Machine Monitor*
VN	*Virtual Network*
VNF	*Virtualized Network Functions*
VON	*Virtual Optical Networks*
VPN	*Virtual Private Network*
XML	*eXtensible Markup Language*

Sumário

1 Introdução

O conceito de virtualização, seja de serviços, aplicativos ou de servidores já existe há muito tempo e, hoje, é uma realidade dentro do ambiente de Tecnologia da Informação (TI) das organizações, sejam elas de qualquer porte, de natureza pública ou privada. É uma poderosa ferramenta para utilização, gerência e monitoramento de recursos computacionais (CARNEIRO et al., 2016). Nos *Data Centers* (DC) das organizações atuais, o termo "virtual" está associado a um computador que não existe fisicamente, ou seja, uma "máquina virtual".

Na década de 1980, a popularização de plataformas de *hardware* baratas como o PC, fez a virtualização perder importância. No entanto, o aumento do poder computacional dos atuais processadores, a disseminação de sistemas distribuídos e a onipresença das redes de computadores causaram, por várias razões, o ressurgimento da virtualização (LAUREANO; MAZIERO, 2008).

Atualmente, dispositivos computacionais móveis como *tablets* e *smart phones* e a evolução dos *softwares* desenvolvidos, tornaram muito difícil imaginar um sistema computacional que não seja conectado em rede. Essa conectividade, fez com que as TIs das organizações, independentemente do seu tamanho, passassem a criar estruturas de DC que comportassem uma maior quantidade de equipamentos com alto poder de processamento e características heterogêneas, voltadas para o atendimento de necessidades específicas de seus clientes e funcionários (DANIELS, 2009).

Passado algum tempo, essas estruturas de DC tornaram-se grandes, complexas, caras e muitas vezes ineficientes gerando nos departamentos de TI, uma grande pressão por mais eficiência da infraestrutura e melhores resultados, oportunidade em que a virtualização mostrou seus benefícios: maximização dos recursos, múltiplos sistemas e integração orçamentária (ZHANG; CHENG; BOUTABA, 2010). Como exemplo, pode-se citar as empresas operadoras de telecomunicações que, diante do desafio de tornar sua infraestrutura mais adequada às mudanças no comportamento dos clientes, com maior eficiência operacional (OPEX - *Operational*

Expenditure), não significasse um elevado aumento de seus investimentos (CAPEX - *Capital Expenditures*) num curto prazo.

Além disso, à medida que esses sistemas computacionais se tornaram mais comuns, a demanda por serviços de rede de computador aumentou, especialmente a Internet, que vivenciou uma explosão de crescimento, fazendo com que sua arquitetura tradicional fosse repensada para mitigar os impactos das dificuldades de desempenho, segurança, escalabilidade e mobilidade que surgiram e não haviam sido previstas em sua criação.

Dadas essas circunstâncias, ao longo dos últimos anos três abordagens tem despertado interesse tanto no âmbito acadêmico quanto no empresarial. São elas:

1. A possibilidade de desenvolver diferentes arquiteturas de rede sobre uma mesma infraestrutura física;
2. O acesso a recursos computacionais através da Internet de forma prática, escalável, sob demanda e com pagamento baseado no uso;
3. A substituição das funções que tradicionalmente são executadas em um *hardware* especializado, com *software* proprietário, para serem executadas em servidores comuns.

Essas abordagens são amplamente referidas, respectivamente, como:

1. **Virtualização de Redes** (NV - *Network Virtualization*) (CHOWDHURY; BOUTABA, 2010)
   A virtualização de redes permite a existência de numerosos ambientes de rede diferentes sobre a mesma infraestrutura física, e portanto, outorga uma grande flexibilidade para a personalização das topologias e recursos virtuais das redes criadas sobre a estrutura física.
2. **Computação em Nuvem** (*Cloud Computing*) (MELL; GRANCE, 2011)
   A computação em nuvem se baseia nos avanços das técnicas de virtualização que possibilitaram desacoplar o ambiente de *software* do ambiente de *hardware*. Em um ambiente virtualizado, um servidor é uma máquina virtual (VM – *Virtual Machine*), que pode ser movida e copiada de um servidor físico (*hardware*) para outro.
3. **Virtualização de Funções de Rede** (NFV - *Network Functions Virtualization*) (CHIOSI et al., 2012)
   NFV consiste em aplicar o conceito de virtualização nas diversas funções da rede, substituindo o *hardware* especializado por VMs, que podem executar em *hardware* genérico de prateleira, como alternativa ao uso de dispositivos de rede e segurança customizados, ou infraestrutura na nuvem.

A adoção dessas técnicas têm criado grandes possibilidades de atingir a flexibilidade da rede com custos menores em relação ao cenário de rede atual (MONTELEONE; PAGLIERANI, 2013), haja vista permitir aproveitar os benefícios demonstrados pela tecnologia da virtualização no âmbito de TI como: a adoção de servidores convencionais, a consolidação e a simplificação da infraestrutura.

Além disso, como NFV traz a ideia de que é possível otimizar a infraestrutura a partir da criação, em *hardware* específico ou em servidores-padrão, de máquinas virtuais capazes de cumprir funções até então realizadas por equipamentos dedicados, possibilita tornar as redes mais ágeis e flexíveis reduzindo os custos materiais e de operação (MAURICIO et al., 2017).

Esse trabalho, aborda a tecnologia de virtualização das funções de rede no domínio da computação em nuvem para atender as necessidades de organizações utilizarem os recursos de TI para disponibilizar sistemas que permitam a prestação de serviços públicos em áreas como educação, saúde, segurança e planejamento urbano, e avalia o resultado da combinação dessas tecnologias como a consolidação e simplificação da infraestrutura com soluções mais econômicas e sustentáveis, proporcionando a opção de se trabalhar na administração da rede com diferentes fabricantes, reduzindo tanto o custo de capital (CAPEX) quanto o custo operacional (OPEX).

1.1 Problema

O sucesso da Internet é inegável, mas sua arquitetura atual além de grandiosa e muito complexa sofreu, ao longo dos anos, muitas extensões e modificações ou ”remendos”, para incluir novas funcionalidades ao seu projeto inicial (FARIAS et al., 2011). Os autores ainda citam alguns dos remendos mais conhecidos: *Classless Internet Domain Routing* (CIDR), *Network Address Translation* (NAT), Serviços Integrados (*Intserv*) e Serviços Diferenciados (*Diffserv*). Mesmo assim, ainda possui uma série de deficiências e limitações que fazem com que diversas expectativas não sejam atendidas.

Nesse contexto, a arquitetura é tida como ”ossificada” (CHOWDHURY; BOUTABA, 2010) ao não permitir grandes modificações no núcleo da rede, restringindo sua ação para atender os requisitos atuais e futuros das aplicações. Por outro lado, em face os avanços na área de tecnologia de informação e comunicação, há a necessidade de expandir a infraestrutura dos sistemas computacionais, bem como assegurar que estejam aptos a suportar mecanismos de gerenciamento, monitoramento e segurança capazes de unificar e homogeneizar os processos que levem ao seu bom funcionamento.

Contudo, mecanismos tradicionais de configuração e operação não são escaláveis e flexíveis para lidar com essas tendências além de não permitirem a interoperabilidade através de seus protocolos padrões da indústria onde, na maioria dos casos, a configuração, operação e manutenção são realizadas através de métodos e ferramentas próprios de cada fabricante, obri-

gando as organizações a desenvolver capacidades muito específicas internamente ou contratarem serviços profissionais externos.

Isso tudo se torna ainda mais aparente no ambiente da administração pública, cujas características de suas instituições, com seus princípios, controles e processos específicos de alocação de recursos para aquisição de ativos, tornam ainda mais complexa a tarefa de administrar uma infraestrutura heterogênea de TI.

1.2 Hipótese

A construção de um ambiente de experimentação utilizando computação em nuvem para prover serviços de NFV deverá fornecer melhores subsídios aos administradores de redes de instituições públicas para os novos desafios que a infraestrutura requer.

O desenvolvimento de novos serviços de redes que não comprometam o tráfego de produção, unificação das funcionalidades dos planos de controle e gerência, especialização de redes para diferentes aplicações e clientes (virtualização de redes) e maior facilidade no gerenciamento da rede, devem ter um significativo ganho de desempenho frente a uma solução tradicional.

1.3 Objetivos

Consoante com a questão apresentada, este trabalho tem como objetivo geral, propor a uma estratégia de virtualização de funções de rede dentro do contexto de uma infraestrutura de computação em nuvem a fim de proporcionar um cenário de rede mais flexível, minimizando a dependência de restrições de *hardware* para instituições públicas.

Para atingir tal objetivo, foram estabelecidos os seguintes objetivos específicos:

- Construção de um ambiente de experimentação;
- Demonstrar o desempenho que se obtém com uma rede NFV em comparação a uma rede tradicional.

1.4 Estrutura do Documento

Este Capítulo apresentou a problemática que motiva esta dissertação, a hipótese e as questões de partida envolvidas, os objetivos e as principais metas deste trabalho. O restante do documento está estruturado em Capítulos, da seguinte forma: no Capítulo 2 apresenta–se o referencial teórico, com alguns conceitos necessários para o melhor entendimento do trabalho, tais como: Virtualização de Funções de Rede, Computação em Nuvem e Esgotamento do IPv4. No Capítulo 3 é apresentada uma revisão sistemática e os trabalhos relacionados ao

contexto desta dissertação. O Capítulo 4 é dedicado à descrição do ambiente de experimentação designadamente a arquitetura geral, a descrição dos principais componentes e a sua interligação. No Capítulo 5 descreve-se o desenvolvimento realizado, nomeando as principais escolhas tomadas e descrevendo os processos mais importantes da aplicação para virtualização de funções de rede. No Capítulo 6 são tratados dois Estudos de Caso para validar a solução proposta. Por fim, no Capítulo 7 é dada a conclusão sobre o trabalho realizado, as dificuldades encontradas e as sugestões de trabalhos futuros.

2

Contextualização

Neste capítulo são apresentados aspectos teóricos relacionados a virtualização de funções de rede, computação em nuvem, impactos da virtualização de funções de rede em nuvem nas instituições públicas, esgotamento do IPv4, gerenciamento de redes de computadores e as ferramentas utilizadas nos experimentos desta dissertação.

2.1 Virtualização de Funções de Rede

Virtualização não é uma expressão nova em tecnologia. Apareceu na década de 1960 por intermédio da empresa IBM (*International Business Machines*) com a finalidade de obter um maior aproveitamento dos grandes e caros *mainframes*. Cada equipamento possuía seu sistema operacional padrão e era necessária uma solução que permitisse a utilização de *softwares* legados das empresas (DOUGLIS; KRIEGER, 2013). Em 1974, Popek e Goldberg (1972) apresentaram em sua dissertação na universidade de *Harvard* a base teórica da arquitetura para sistemas computacionais virtuais.

A virtualização refere-se ao processo de criação de uma versão virtual de algo. Para Douglis e Krieger (2013) , virtualização é uma tecnologia baseada em *software* que funciona dividindo um recurso de *hardware* em partes, chamadas de máquinas virtuais (VMs - *Virtual Machines*), que podem ser utilizadas para fins distintos, tornando possível executar vários sistemas operacionais e aplicativos no mesmo servidor, ao mesmo tempo. Uma máquina virtual é uma abstração em *software* de uma máquina real.

O conceito de virtualização se estende de maneira similar às redes de computadores. A virtualização de redes, que poderia ser simplesmente definida como uma rede física que passa a ser capaz de abrigar várias redes virtuais (VNs - *Virtual Networks*), passou a ser uma das mais marcantes tecnologias nessa nova ideia da Internet do futuro (CHOWDHURY; BOUTABA, 2010).

Ao se referirem ao conceito de virtualização de redes, Luizelli e outros (2014) afirmam ser um mecanismo que permite a coexistência de múltiplas VNs heterogêneas, compartilhando recursos de uma mesma infraestrutura física. Essas VNs podem apresentar arquiteturas, protocolos e topologias independentes das do substrato de rede na qual serão instanciadas.

Na arquitetura de virtualização de redes, uma VN é composta por uma série de nodos virtuais unidos por enlaces virtuais. Cada VN é formada por um subconjunto dos recursos da rede física. Wen e outros (2013) definem que um nodo pode ser qualquer equipamento de rede, na maioria das vezes um *switch* ou um roteador. Um enlace é uma conexão física ou lógica entre dois nós na rede.

Em seus trabalhos, Chowdhury e Boutaba (2009) e (2010) , definem que múltiplas redes lógicas coexistentes podem ser classificadas em quatro classes principais:

- VLAN (*Virtual Local Area Network* ou Rede Local Virtual)
  Um grupo de *hosts* com um interesse comum que são logicamente reunidos sob um único domínio de *broadcast*, independentemente da sua conectividade física.

- VPN (*Virtual Private Network* ou Rede Privada Virtual)
  Uma rede dedicada a conectar vários *sites* usando túneis privados e seguros através de redes de comunicação compartilhadas ou públicas como a Internet.

- Redes Ativas e Programáveis
  Presente na maioria dos projetos nesta área levando o conceito de redes coexistentes através de programação, apesar de não poderem ser consideradas como exemplos diretos de virtualização de rede.

- Rede Sobreposta (*Overlay Network*)
  Uma rede virtual que cria uma topologia virtual no topo da topologia física de outra rede. Nós em uma rede sobreposta estão ligados através de ligações virtuais que correspondem a caminhos na rede subjacente.

Já as tecnologias de rede de código aberto fizeram surgir o conceito de *Software-Defined Networking* (SDN), com a proposta de ser uma arquitetura dinâmica, gerenciável, adaptável e de baixo custo (STANFORD, 2014).

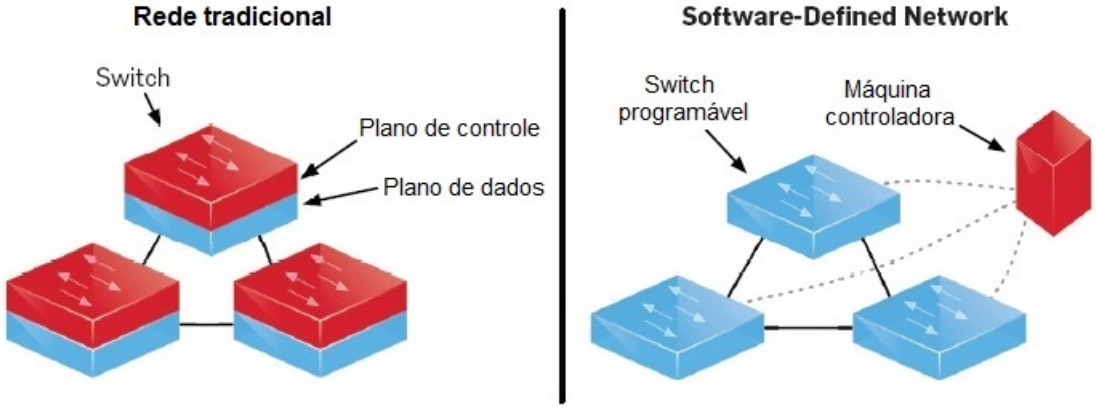
Para Kreutz e outros (2015), SDN é um paradigma de rede que foi proposto para dar esperança de mudar as limitações de infraestrutura das rede atuais. Primeiro, pela quebra da integração vertical, separando a lógica de controle da rede (o plano de controle) dos roteadores e *switches* subjacentes que encaminham o tráfego (o plano de dados). Em segundo lugar, pela separação dos planos de controle e de dados, *switches* de rede tornam-se dispositivos (ou elementos) de encaminhamento simples e a lógica de controle é implementada em um controlador de rede (também chamado sistema operacional de rede ou hypervisor de rede (*Network Hypervisor*))

logicamente centralizado, simplificando a aplicação de políticas e (re)configuração e evolução da rede.

Em resumo, SDN separa o *hardware* (plano de dados) do *software* (plano de controle), passando para o controlador a decisão de encaminhar ou descartar pacotes que serão repassados aos dispositivos (ou elementos) de rede para tão somente executar, possibilitando assim a configuração automatizada da rede.

A Figura 1 ilustra a comparação entre arquiteturas de rede tradicional e SDN. Na tradicional, cada dispositivo de rede, tais como *switch* e roteador, possui o plano de dados e o plano de controle (KREUTZ et al., 2015). Na rede SDN, o sistema responsável pelos protocolos e pelas tomadas de decisão que resultam na confecção das tabelas de encaminhamento, o plano de controle é separado dos sistemas adjacentes que encaminham os dados para um determinado destino, o plano de dados. Dessa forma, o plano de controle é centralizado e plano de dados fica no dispositivo.

Figura 1 – Comparação das Arquiteturas de redes Tradicional e SDN



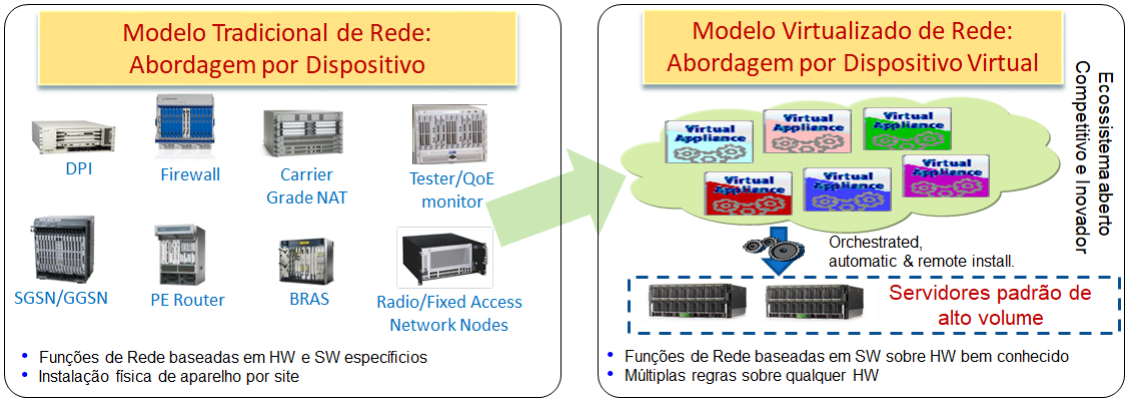
Fonte: (CASADO; FOSTER; GUHA, 2014)

A virtualização de funções de rede (NFV) é apenas um conceito de migrar funções de rede (*Network Function* - NF), executadas normalmente em equipamentos dedicados (*Appliance*) de propósito específico, para um *hardware* genérico, conforme ilustrado na Figura 2.

Por NF (também conhecida como *middleboxes* (CARPENTER; BRIM, 2002)), entende-se com sendo uma determinada funcionalidade em redes de computadores, dentre as quais se destacam: comutação de pacotes (*switches*), roteamento (*router*), segurança (*firewall*), inspeção de pacotes (DPI *Deep Packet Inspection*), *proxy*, DNS (*Domain Name System*), DHCP (*Dynamic Host Configuration Protocol*), balanceamento de carga, além das funções de uma rede de telecomunicações que são muito mais diversas. Para Heideker e Kamienski (2015), essas NFs funcionam de modo conjunto ou isolado e empregam em sua construção tecnologia proprietária tanto no *hardware* quanto no *software*.

As especificações do NFV estão sendo desenvolvidas pelo Instituto Europeu de Normas

Figura 2 – Relação entre Rede Tradicional e Rede Virtualizada



Fonte: (ETSI ISG NFV, 2013)

de Telecomunicações (ETSI - *European Telecommunications Standards Institute*), especificamente pelo ISG NFV (*Industry Specification Group for Network Functions Virtualization*). Seu principal objetivo é transformar a maneira que os operadores projetam suas redes, por meio de tecnologias de virtualização de TI padrão, para consolidar equipamentos de rede em servidores padrão de mercado de alto volume, *switches* e dispositivos de armazenamento, que poderiam ser localizados em centros de dados, nós da rede ou nas instalações do usuário final (CHIOSI et al., 2012).

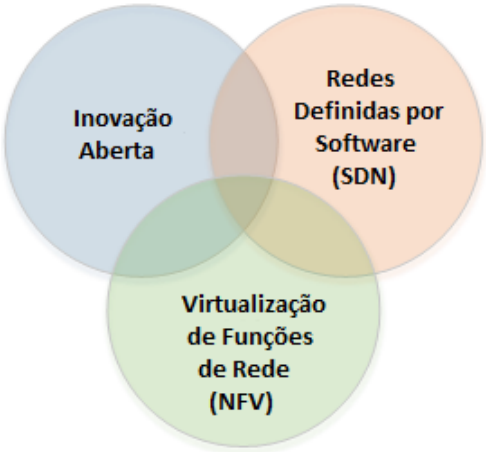
Interessante notar que, embora os conceitos de NFV e SDN sejam considerados altamente complementares conforme indicado na Figura 3, eles não dependem um do outro (CHIOSI et al., 2012). Em vez disso, ambas as abordagens podem ser combinadas para promover a inovação no contexto da rede. Enquanto o NFV trata da implementação dos serviços de rede como *software*, que pode ser executado em *hardware* de servidores de padrão comercial, o SDN com sua capacidade para abstrair e, programaticamente, controlar os recursos da rede, desempenha um papel importante na orquestração do NFV ((NUNES et al., 2014); (HAKIRI et al., 2014) e (MATIAS et al., 2015)).

A sinergia entre estes conceitos tem levado a ONF (*Open Networking Foundation*) e a ETSI a trabalharem em conjunto, com o objetivo comum de evoluir ambas as abordagens e proporcionar um ambiente estruturado para o seu desenvolvimento.

Na Figura 4, são apresentados os componentes de arquitetura padrão do NFV. Nela estão identificados os principais domínios, que são: Função de Rede Virtualizada (VNF - *Virtualized Network Functions*), Infraestrutura NFV (NFVI - *Network Functions Virtualization Infrastructure*) e o Gerenciamento e Orquestração NFV (ETSI ISG NFV, 2013).

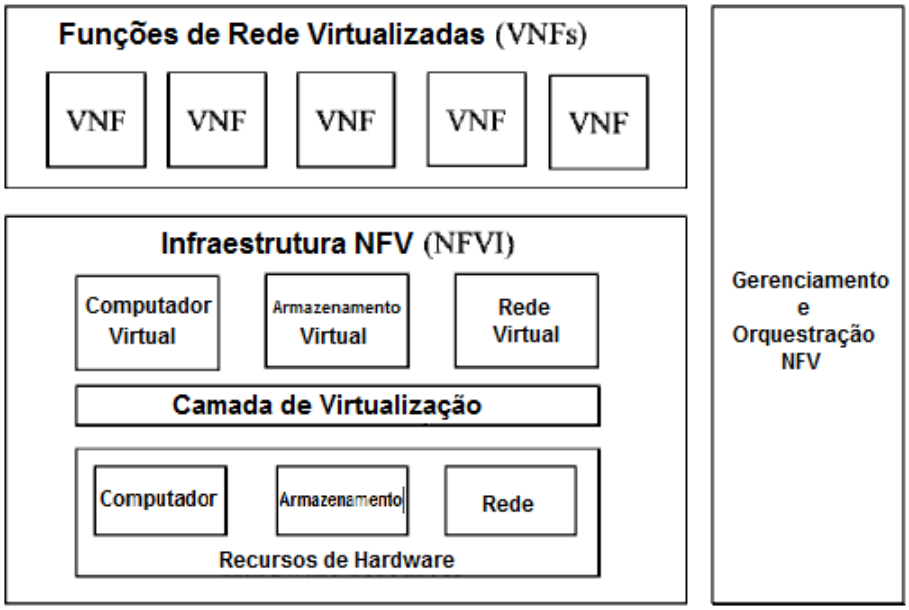
VNF é a implementação em *software* de uma função de rede com a capacidade de rodar sobre a infraestrutura NFV. A NVFI inclui a diversidade de recursos físicos e como estes podem ser virtualizados. O Gerenciamento e Orquestração (*Orchestrator*) é uma entidade centralizada, com visão completa da rede para gerenciar o plano de controle e implantar VNFs (ETSI ISG

Figura 3 – Relação entre SDN e NFV



Fonte: (CHIOSI et al., 2012)

Figura 4 – Arquitetura Padrão do NFV



Fonte (ETSI ISG NFV, 2013)

NFV, 2013).

A tecnologia NFV tira proveito de serviços de IaaS para formar a infraestrutura de virtualização de função de rede (NFVI).

2.2 Encadeamento de Funções de Serviço

NFV propõe que todas as funções executadas por nós de rede sejam definidas de forma virtualizável, de modo que todas as operações de rede possam ser classificadas em blocos de construção separados que podem ser encadeados. Cada um desses blocos de construção

representaria uma função de rede virtualizada (VNF) (ETSI ISG NFV, 2013). Desta forma, os serviços de NFV são providos por uma sequência de VNFs, denominada Encadeamento de Funções de Serviço (*Service Function Chaining*, SFC) [(SAHHAF et al., 2015) e (HERRERA; BOTERO, 2016)].

SFC é definida pela RFC 7665:2015 (HALPERN; PIGNATARO, 2015) como um conjunto ordenado de abstrações de Funções de Serviços (*Service Function* - SF) que devem ser aplicadas aos fluxos de pacotes selecionados como resultado de uma classificação. Em suma, são funções virtuais do tipo Balanceamento de Carga, *Firewall*, *IDS/IPS (Intrusion Detection System/Intrusion Prevention System)*, NAT, *Quality of Service- QoS*, *Proxy*, *DPI* etc (HALPERN; PIGNATARO, 2015).

Ainda de acordo com a RFC 7665:2015 (HALPERN; PIGNATARO, 2015), os principais componentes da SFC são o classificador, o plano de controle e os encaminhadores de funções de serviço. O classificador, 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.

Na abordagem NFV, os serviços são implementados ao encadear adequadamente os VNFs que podem ser distribuídos ao longo do NFVI.

2.3 Computação em Nuvem

Apesar de divergências quanto ao criador da expressão ”computação em nuvem”, para Aymerich, Fenu e Surcis (2008) o termo foi utilizado pela primeira vez em 2006, pelo *CEO* (*Chief Executive Officer*) do Google, Eric Schmidt, para se referir à computação empregando os recursos da Internet.

O NIST (*National Institute of Standards and Technology*), entidade de padronização ligada ao governo norte americano, definiu Computação em Nuvem como um estilo de computação no qual os recursos computacionais (servidores, redes, serviços, sistemas de armazenamento e aplicações) são fornecidos aos clientes através da Internet.

Em outras palavras, a computação em nuvem é uma solução em que um conjunto compartilhado de recursos computacionais são fornecidos aos usuários sem a necessidade de uma infraestrutura presente e podem ser rapidamente alocados e liberados com mínimo esforço ou interação com o provedor de serviços (MELL; GRANCE, 2011). Os clientes (inquilinos) de uma plataforma de nuvem utilizam os recursos desse conjunto.

De acordo com o NIST (MELL; GRANCE, 2011), as principais características de computação em nuvem, são serviço sob demanda, acesso em banda larga, compartilhamento de

recursos, rápida elasticidade e bilhetagem. Descreve também que a computação em nuvem é composta por três modelos de serviço e quatro modelos de implantação, detalhados a seguir.

2.3.1 Modelos de Serviços

Na computação em nuvem os recursos são distribuídos na forma de serviços e para o NIST (MELL; GRANCE, 2011), três modelos são reconhecidos:

- SaaS - *Software as a Service* ou Software como Serviço;
- PaaS - *Platform as a Service* ou Plataforma como Serviço;
- IaaS - *Infrastructure as a Service* ou Infraestrutura como Serviço.

No modelo SaaS, um aplicativo que é mantido e gerenciado pelo fornecedor é oferecido aos clientes através da Internet com controle limitado sobre as configurações de software. A PaaS fornece a estrutura de hospedagem para a implantação de aplicativos específicos do inquilino, que precisam ser compatíveis com a plataforma do provedor. O modelo de IaaS oferece recursos básicos de computação como armazenamento, conectividade de rede, processamento e outros elementos. Os recursos podem ser virtual ou físico, com diferentes níveis de isolamento entre inquilinos (por exemplo, diferentes grupos de recursos).

2.3.2 Modelos de Implantação

Quanto ao modelo de implantação que significa como a nuvem é fornecida para as empresas e para os usuários, o NIST (MELL; GRANCE, 2011) classifica a computação em nuvem em quatro categorias de modelo:

- Privada - serve uma única organização podendo ser utilizado pela organização ou por um terceiro e pode existir no cliente ou fora do seu próprio *Data Center*;
- Pública - serve o público em geral e pode ser fornecida para uma variedade de organizações;
- Comunitária - presta serviços a um grupo de organizações com objetivos semelhantes;
- Híbrida - quando dois ou mais modelos de implantação são compostos para a entrega de serviços em nuvem.

2.3.3 Plataformas para Computação em Nuvem

Uma plataforma para computação em nuvem pode ser definida como um sistema integrado para a criação e o gerenciamento de serviços em nuvens privadas, públicas e híbridas;

combinado com a virtualização de servidores, armazenamento, rede e segurança através de uma abordagem centralizada para automatizar o ciclo de vida dos serviços na infraestrutura de nuvem.

Algumas plataformas de computação em nuvem disponíveis são: *VMware vRealize Suite*, *Openstack*, *OpenNebula*, *Cloudstack*, *Eucalyptus* e *Windows Azure*, porém nem todas são de código aberto (*open-source*) e o custo com licenças, dependendo do tamanho da infraestrutura, pode exceder milhões de reais.

Openstack, *Cloudstack* e *Eucalyptus* são ferramentas de código aberto e concorrem entre si para se tornarem a plataforma padrão, sendo o *Openstack*, a mais promissoras de todas. Grandes empresas de tecnologia, como HP, IBM, RedHat, VMware, Cisco, Dell, EMC, Yahoo, etc.; fazem parte do consórcio que financiam e colaboram com o desenvolvimento do *Openstack*, muitas delas possuem uma versão licenciada com funcionalidades desenvolvidas e/ou adaptadas para seus produtos e soluções.

2.4 Impactos da Virtualização de Funções de Rede em Nuvem nas Instituições Públicas

O setor público é composto por diversas instituições que afetam a vida das pessoas. Estas instituições, que são criadas e mantidas por todos os entes que integram a federação brasileira (União, Estados, Distrito Federal e Municípios), incluem organizações políticas e estruturas que determinam e implementam leis, provêm serviços sociais e públicos básicos. Compõem, ainda, um sistema que atua em áreas como assistência social, finanças, transportes, justiça, educação e saúde, isto é, essenciais para a sociedade (PALUDO, 2010).

As instituições públicas fazem parte da Administração Pública, um modelo de gestão de empresas e instituições públicas e governamentais que tem como finalidade principal desempenhar toda a atividade administrativa do Estado visando a satisfação das necessidade coletivas (MEIRELLES; FILHO, 2016).

Segundo Costin (2010), a estrutura que separa o Brasil em três níveis de governo (Federal, Estadual e Municipal), chamada de Estado federativo é a base da administração pública no país. Outro aspecto levantado pela autora é que com a Constituição Federal (CF) de 1988 (FEDERAL, 1988), a Administração Pública recebeu tratamento em capítulo próprio e em seu Art. 37, instituiu para todos os poderes da União os princípios da legalidade, impessoalidade, moralidade, publicidade e eficiência além de separá-la em dois tipos: a Administração Direta e a Indireta. A Administração Direta é composta por governos, ministérios, secretarias e a Indireta é formada por autarquias, fundações, empresas públicas e sociedades de economia mista (COSTIN, 2010).

A administração pública também impõe às instituições públicas certas particularidades:

- Os recursos são obtidos através dos impostos, taxas e contribuições recolhidos dos contribuintes;
- O controle é do Estado, ou seja, controle político;
- A tomada de decisões são baseadas em políticas públicas, geralmente mais lentas;
- O ordenamento jurídico é regido pelo princípio da legalidade e do direito público (constitucional e administrativo), de forma que só é permitido fazer aquilo que está previsto na lei, sendo assim vedado tudo aquilo que não houver previsão legal.

O funcionamento das instituições públicas ajustadas aos projetos e às políticas governamentais vigentes, faz com que o seu orçamento, seu direcionamento de verbas e seu tema de atuação estejam perfeitamente alinhados. Dessa forma, seus recursos ficam limitados, não permitindo atender a todas as necessidades, sendo por isso necessário estabelecer prioridades. Dadas essas características, verifica-se, cada vez mais, que as organizações públicas ampliam sua dependência em relação aos serviços de TI para satisfazer seus objetivos corporativos e atender às necessidades estratégicas das organizações (LARROCHA et al., 2010).

A garantia da correta aplicação de recursos públicos, o estímulo a proteção de informações críticas e a cooperação para que as instituições públicas atinjam seus objetivos organizacionais e sociais com dispêndios toleráveis, influenciam, inclusive, o desenvolvimento, implantação e utilização dos sistemas de informação, cuja importância para esse tipo de organização tem crescido nos últimos anos, viabilizando sua utilização (BRANDI; MALHEIRO; BAPTISTA, 2017).

Dentro desse contexto, virtualizar funções de rede torna-se uma oportunidade promissora para uma boa gestão do gasto público, uma vez que propicia a redução de desperdícios, permitindo o aumento dos recursos disponíveis para que o Estado possa atender melhor a população nas mais variadas formas, já que a NFV propõe mover as funções de rede de um *hardware* especializado, para módulos de *software* em servidores comuns trazendo o processamento da função de rede para a nuvem.

A validação dessa mudança passa, necessariamente, pela execução de algoritmos e protocolos num ambiente já em plena atividade o que torna muito difícil a realização desses testes sem comprometer o funcionamento da rede atual. Por outro lado, a maioria dos pesquisadores não tem acesso a uma grande infraestrutura de nuvem para testar suas propostas (BENET et al., 2017).

2.5 Esgotamento do IPv4

As redes de computadores, especialmente a Internet, utilizam o protocolo TCP/IP (*Transmission Control Protocol/Internet Protocol*) como padrão de comunicação (COMER, 2016).

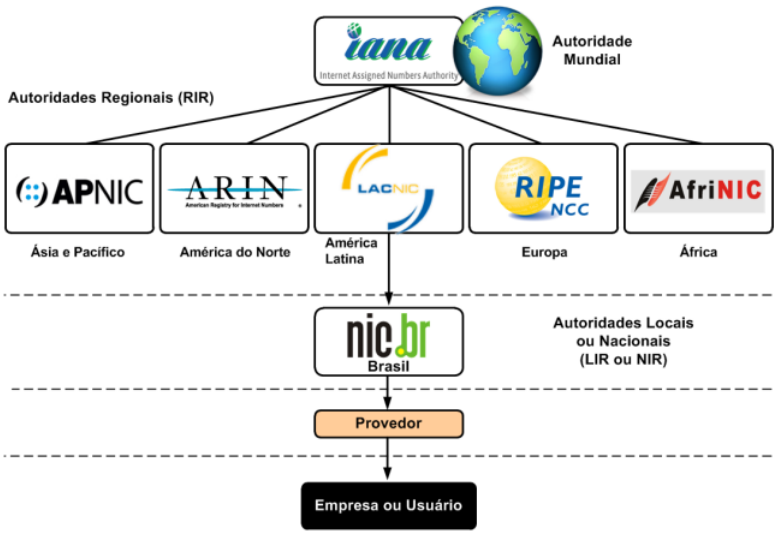
2.5.1 Protocolo IP

O protocolo IP foi projetado para criar ligações entre diferentes redes, possibilitando a intercomunicação entre dispositivos (o que inclui computadores, *smartphones* e afins) nelas presentes. Cada computador na Internet, possui um número único, que serve para identificar o dispositivo na rede, chamado endereço de Internet ou endereço IP (FOROUZAN; FEGAN, 2009).

A distribuição de endereços IP na Internet é controlada, de forma hierárquica, por diversas autoridades, conforme ilustrado na Figura 5. IANA (*Internet Assigned Numbers Authority* – Autoridade para Atribuição de Números da Internet) é a autoridade mundial responsável pela administração global dos registros de endereços IP, repassando-os para as Autoridades Regionais, chamadas RIR's (*Regional Internet Registry*), que administram a distribuição dos endereços nas autoridades abaixo de sua hierarquia localizadas em uma região específica (BRITO, 2013).

Cada RIR serve a um região diferente, distribuindo endereços a cada uma das cinco regiões: África (AfriNIC - *African Network Information Center*), Ásia e Pacífico (APNIC - *Asia-Pacific Network Information Center*), América do Norte (ARIN - *American Registry for Internet Numbers*), América Latina e Caribe (LACNIC - *Latin American and Caribbean IP Address Regional Registry*) e Europa e Oriente Médio (RIPE NCC - *Réseaux IP Européens Network Coordination Center*). No Brasil a Autoridade Local é o Núcleo de Informação e Coordenação do Ponto BR (NIC.br) que, subordinado ao LACNIC, repassa os endereços para os provedores e estes para as empresas ou usuários (BRITO, 2013).

Figura 5 – Autoridades da Internet



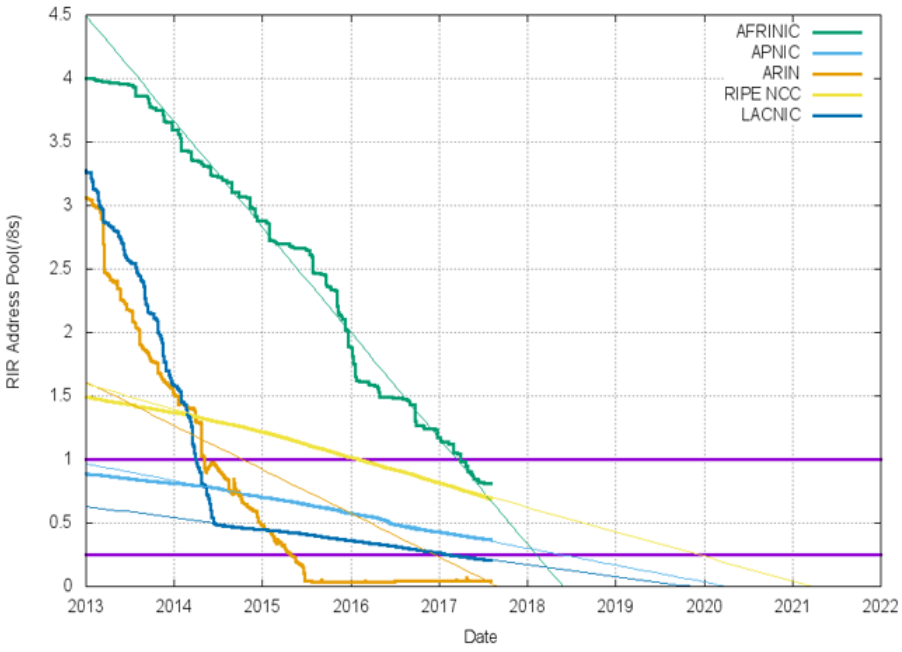
Fonte: (BRITO, 2013)

2.5.2 IPv4

Atualmente, a maior parte dos computadores utiliza o protocolo IPv4 (*Internet Protocol Version 4*) que é definido e especificado na RFC 791 (POSTEL, 1981). No IPv4, os endereços IP são formados por quatro campos numéricos com dígitos decimais separados por pontos (exemplo: 192.168.2.128). Cada um desses campos representa um octeto, o que significa dizer que o endereço IPv4 completo tem 32 bits. O espaço de endereçamento do IPv4 permite 4 294 967 296 endereços (COMER, 2015).

Apesar do sucesso do IPv4, seu projeto original não previu alguns aspectos como: o aumento da tabela de roteamento; problemas relacionados a segurança dos dados transmitidos; prioridade na entrega de determinados tipos de pacotes e o crescimento das redes. Este último em especial, provocou o esgotamento dos endereços IP (COMER, 2015). A Figura 6 ilustra a projeção feita pelo Núcleo de Informação e Coordenação do Ponto br - NIC.br (NIC.br, 2017)

Figura 6 – Previsão de esgotamento de endereços IPv4 nos RIR



Fonte: IPv6.br

No entanto, esse esgotamento não se concretizou devido ao desenvolvimento de uma série de tecnologias que funcionaram como uma solução paliativa para o problema, adiando assim o esgotamento do IPv4 (COMER, 2016).

2.5.3 IPv6

O IPv6 (*Internet Protocol Version 6*) definido na RFC 2460 (DEERING; HINDEN, 1998) veio para resolver vários problemas do IPv4, entre eles, o número limitado de endereços disponíveis. Apresenta também melhorias com relação ao IPv4 em áreas como a auto configuração de

roteamento e de rede. O IPv4 e o IPv6 não são diretamente compatíveis entre si. O IPv6 não foi projetado para ser uma extensão, ou complemento, do IPv4, mas sim, um substituto que resolve o problema do esgotamento de endereços. Embora não interoperem, ambos os protocolos podem funcionar simultaneamente nos mesmos equipamentos e com base nisto a transição foi pensada para ser feita de forma gradual (COMER, 2016).

O IPv6 é constituído por 128 bits em vez dos 32 do IPv4. A quantidade de endereços disponíveis pode chegar a 340.282.366.920.938.463.463.374.607.431.768.211.456. Isso evita que os endereços disponíveis se esgotem rapidamente como também a necessidade do uso de técnicas como NAT (FOROUZAN; FEGAN, 2009).

Em IPv6, os endereços passam a ser representados por números hexadecimais de 16 bits, separados por ":" (é indiferente representar as letras com maiúsculas ou minúsculas) e algumas abreviações são possíveis, como a omissão de zeros à esquerda e a representação de um conjunto contínuo de zeros por "::". Além disso, são escritos em oito grupos de quatro dígitos hexadecimais (exemplo: 2001:0db8:85b3:1319:8c2e:0370:7344) (FOROUZAN; FEGAN, 2009).

2.5.4 Transição IPv4/IPv6

A evolução natural da Internet será o IPv6, porém mesmo tendo essa consciência, temos que levar em conta o alto grau de disseminação do IPv4, que aliado a falta de profissionais preparados para trabalhar com IPv6 e o alto custo dos equipamentos para implementação do mesmo, causam uma lentidão a esse processo de transição. Segundo Comer (2015) , a mudança de IPv4 para IPv6 deve recorrer a métodos de transição que funcionem com ambos os protocolos, já que os dois vão continuar em funcionamento durante alguns anos. Alguns deles são:

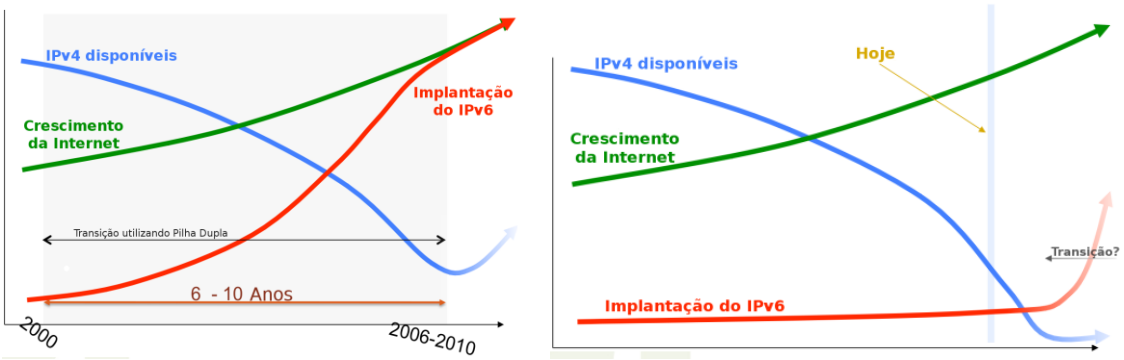
- Pilha dupla ou camada de IP dupla (*Dual Stack*)
  Consiste em implementar e utilizar ambos os protocolos, IPv4 e IPv6 na rede em geral, de maneira gradativa, implicando na coexistência de duas redes em paralelo. Dessa forma, essa estratégia facilita o processo de transição para um ambiente totalmente baseado em IPv6;

- Túneis IPv6 sobre IPv4 (*Tunneling*)
  É uma técnica que permite o tráfego baseado em um protocolo ser transportado por meio de outro protocolo, ou seja, quando pacotes IPv6 forem transportados (tunelados) sob pacotes IPv4;

- Tradução (*Translation*)
  Os mecanismos de tradução permitem que equipamentos que usem IPv4 consigam comunicar com outros que usam IPv6, e vice-versa por meio da conversão dos pacotes.

Prevê-se que ambos os protocolos funcionem lado a lado durante algum tempo, mas a médio ou longo prazo o IPv6 substituirá o IPv4. Na prática, a transição do IPv4 para IPv6 se

Figura 7 – Implantação do IPv6 no Brasil - Plano Ideal x Realidade



(a) Plano Ideal (b) Realidade

Fonte: (NIC.br, 2017)

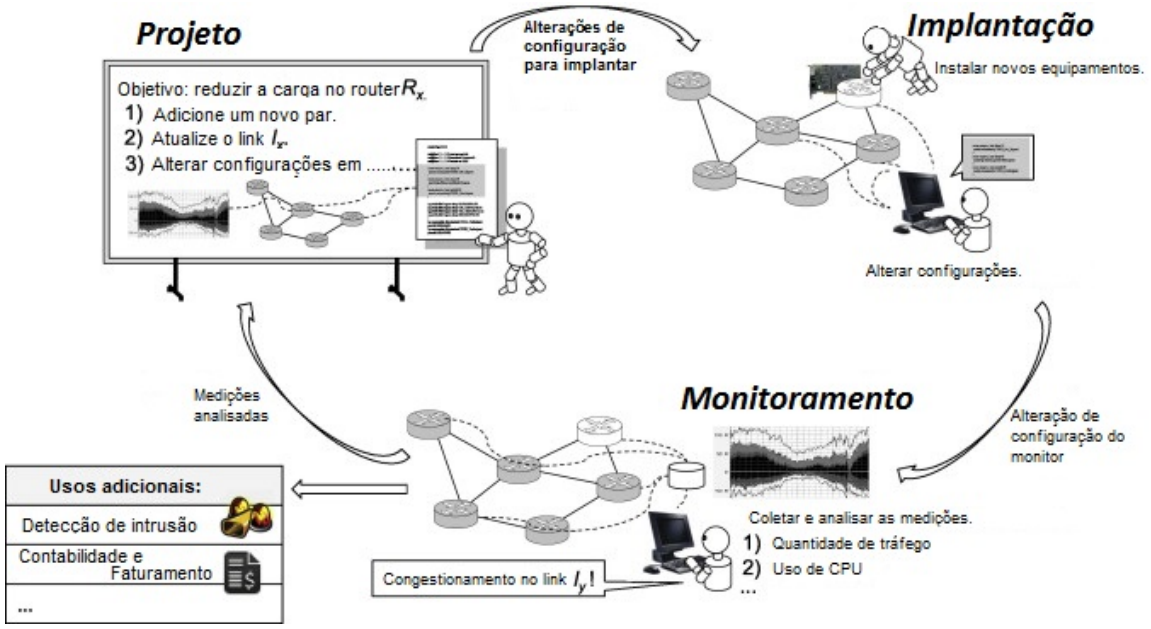
dará gradativamente, até que todos os *hosts* sejam IPv6. Neste caso, o novo protocolo deve ser compatível com a versão anterior, onde *hosts* IPv6 são capazes de se comunicar tanto com *hosts* IPv6 quanto IPv4. O NIC.br (NIC.br, 2017) demonstra o cenário ideal e a realidade da transição de um protocolo para o outro no Brasil conforme ilustrados nas Figuras 7(a) e 7(b).

2.6 Gerenciamento de Redes de Computadores

Em gerenciamento de redes os operadores de redes utilizam duas fontes de dados: medições e configurações (LEE; LEVANTI; KIM, 2014). As medições mostram o comportamento da rede atual e incluem *traces* de pacotes coletados em diferentes pontos da rede. As configurações incluem arquivos de configuração com seus parâmetros como também mapas de topologia física e lógica. Para os autores, o ciclo operacional do gerenciamento de redes contempla três operações: (*i*) monitoramento do comportamento de uma rede, (*ii*) projeto de mudanças de configuração de acordo com os requisitos e (*iii*) implantação de mudanças de configuração e infraestrutura, retratadas na Figura 8. Ainda segundo os autores, as funções básicas de gerenciamento de rede no âmbito de cada grupo de operações, podem assim ser detalhadas:

- Monitoramento
  - Monitorar e solucionar problemas de uma rede
  - Medir o comportamento da rede
  - Identificar padrões de uso e localizar problemas na rede
  - Verificar a precisão das alterações de configuração
- Projeto
  - O projeto da configuração muda de acordo com os requisitos

Figura 8 – Classificação das Operações de Gerenciamento de Rede



Fonte: (LEE; LEVANTI; KIM, 2014)

  - Projetar a comportamento desejado de acordo com os requisitos
  - Ler e compreender as configurações existentes
  - Mapear o comportamento desejado em mudanças de configuração

- Implantação
  - Implantar as mudanças de configuração na rede
  - Entregar as alterações de configuração com segurança
  - Retornar para um estado anterior quando as mudanças não são satisfatórias
  - Certificar-se de que o *software* dos dispositivos de rede recebam *patches* e atualizações

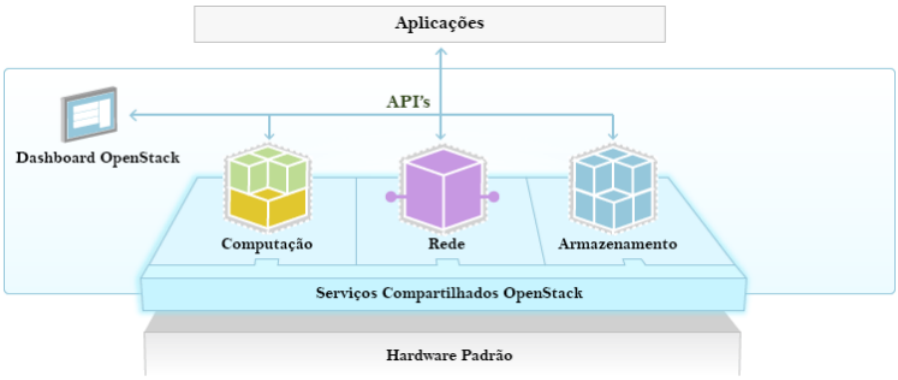
2.7 Ferramentas

Nessa seção são apresentadas as ferramentas utilizadas no trabalho.

2.7.1 Orquestrador de Nuvem OpenStack

O orquestrador executa diversas tarefas de controle da infraestrutura, como a escolha dos servidores hospedeiros de um determinado conjunto de VMs, a criação e a destruição de VMs, a autenticação de usuários e a coleta de estatísticas de uso (SCIAMMARELLA et al., 2016).

Figura 9 – Arquitetura OpenStack



Fonte: (OPENSTACK, 2011)

O *OpenStack*, orquestrador de nuvem criado em julho de 2010 pela *Rackspace Hosting* em conjunto com a NASA (*National Aeronautics and Space Administration*), é uma plataforma de arquitetura modular e flexível que reúne um conjunto de projetos independentes, ou seja, ele não é um software monolítico. Sendo assim, consegue-se utilizar apenas os componentes que desejar (OPENSTACK, 2011).

Utilizando código aberto, essa tecnologia possibilita que uma série de projetos inter-relacionados controlem três conjuntos de recursos, conforme ilustrado na Figura 9.

- Computação
  Responsável por fornecer e gerenciar grandes redes de máquinas virtuais, criando uma plataforma de computação em nuvem redundante e escalável (p.ex. processamento e memória);

- Rede
  Fornece gerenciamento de IP, DNS, DHCP, balanceamento de carga, políticas de *firewall* e gerenciamento de VPN;

- Armazenamento
  Provê funcionalidades de gerenciamento de recursos de armazenamento através de mecanismos de redundância e escalabilidade para uso de *backups*, arquivamento e retenção de dados.

Para gerenciar esses recursos, o *OpenStack* fornece um conjunto de serviços e APIs nativas (*Application Programming Interface*) que permitem a manipulação da nuvem além de proporcionar aos operadores (administradores e usuários) uma *interface* gráfica, chamada *Dashboard*, para acessar, fornecer e automatizar recursos baseados em nuvem possibilitando a esses operadores, interagirem com os recursos do *OpenStack* como também provisionarem seus próprios recursos dentro dos limites estabelecidos (COUTO et al., 2015).

Os projetos para *OpenStack*, juntamente com um vibrante ecossistema de provedores de tecnologia e futuros projetos, fornecem uma estrutura e sistema operacional conectáveis para nuvens públicas e privadas.

2.7.2 Fuel for OpenStack

Uma das principais dificuldades no uso do *OpenStack* é o seu processo de instalação, que pode ser bastante extenso e complexo. Além de impactar a criação de ambientes de produção, isso prejudica ainda mais atividades de desenvolvimento e validação da plataforma, onde é necessário realizar a instalação do software diversas vezes, usando diversas configurações.

O *Fuel* é uma ferramenta de implantação do *OpenStack* criada pela Mirantis Inc. que fornece uma experiência intuitiva orientada por uma interface gráfica web para a implantação e gestão de uma variedade de módulos e plug-ins *Openstack*. Sua finalidade é acelerar processos antes demorados e complexos de implantação dos módulos do *OpenStack* focando em ser aberto e compatível(MIRANTIS, 2014).

O *Fuel* traz a simplicidade de instalação para os usuários, simplificando e acelerando o complexo processo de implementação do *Openstack* que muitas vezes pode gerar erros em sua configuração em grande escala (MIRANTIS, 2014).

2.7.3 Emuladores de Rede de Computadores

Ferramentas gráficas como por exemplo Zenmap (Zenmap, 2017) ou OpenNMS (OpenNMS, 2017) são utilizadas para documentar a organização da topologia física e lógica da rede.

Mas, para o desenvolvimento, estudos, experimentos e testes de novos protocolos em uma rede, existem três métodos que podem ser usados por cientistas para testes de suas aplicações em uma rede de computador especifica, são eles: teste em rede real, técnica de simulação em redes e o método de emulação em rede. Cada método tem seus pontos fortes e fracos no qual podem ser usados em conjuntos para suprir as necessidades uns dos outros (SARI; WIRYA, 2007).

Testes em rede real podem comprometer o bom funcionamento da rede de computadores enquanto que, para a realização de testes com diferentes topologias de rede se faz necessário o uso de diversos tipos de equipamentos de rede que, além do custo de aquisição, necessitam de um espaço físico adequado podendo inviabilizar a realização de experimentos (SARI; WIRYA, 2007).

Uma alternativa ao uso de equipamentos físicos consiste em utilizar *softwares* e ferramentas para emulação e simulação de redes. Existem vários *softwares* com este objetivo, entre os quais podem-se citar *Packet Tracer* (TRACER, 2017), *GNS3* (SIMULATOR, 2012), *NetKit* (PIZZONIA; RIMONDINI, 2008) e *Common Open Research Emulator* (CORE) (AHRENHOLZ et al., 2008).

Todas essas ferramentas são de código aberto (*Open Source*) e especificamente o *Auto-NetKit*, *NetKit* e o *CORE* serviram de base para a realização do estudo de caso no âmbito dessa dissertação e são apresentados a seguir.

O *AutoNetKit* (KNIGHT et al., 2012) é uma ferramenta destinada a automatizar a geração e implementação de configurações em redes emuladas através do *framework* NetKit. Assim, o AutoNetKit permite aos utilizadores do NetKit não só criarem redes maiores como também mais complexas de forma rápida e fácil.

O *NetKit* (PIZZONIA; RIMONDINI, 2008) tem a finalidade de permitir a execução e gerenciamento de experimentos com redes de computadores em um computador simples, possibilitando um baixo custo em relação a compra e o uso de equipamentos de redes físicos. O experimento criado no emulador *NetKit* pode ser iniciado e criado através de *scripts* ou através da linguagem NetMl, que é uma linguagem baseada em XML (*eXtensible Markup Language*) para redes.

Os dispositivos emulados do *Netkit* baseiam-se no *kernel Linux* UML (*User Mode Linux*) que funcionam como roteadores ou computadores, e *switches Ethernet* virtuais (UML *switch*) para interligar as máquinas virtuais (PIZZONIA; RIMONDINI, 2008). As máquinas virtuais criadas no Netkit é um computador completo rodando uma distribuição Debian em modo usuário (RIMONDINI, 2007).

O *Core* (AHRENHOLZ et al., 2008) é uma ferramenta de emulação de rede utilizada em sistemas operacionais *Linux* que possibilita a construção de rede interconectáveis. Baseado no código livre do *Integrated Multi-protocol Network Emulator/Simulator* (IMUNES) da Universidade de Zabred, permite emular computadores, *switch*, roteadores e *links* de redes além de possui suporte a redes *wireless*, *scripts* de mobilidade, IPsec, IPv6, emulação distribuída, controle de roteadores externos Linux, uma API remota e *widgets* gráficos (AHRENHOLZ et al., 2008).

O *Core* consiste de um *daemon* (*backend*) que é responsável por gerenciar as sessões de emulações e uma interface gráfica, o CORE GUI (*frontend*), para desenhar topologias de rede. Essencialmente seu funcionamento baseia-se em conectar nós e redes via interfaces de cada nó, e então o *daemon* responsável pela criação da sessão é executado via interface gráfica CORE GUI. Esses *daemons* usam módulos *Python* que são importados diretamente de *scripts Python* da instalação.

2.7.4 Geração de Tráfego

Para geração de carga em redes de computadores são utilizadas ferramentas de geração de tráfego. Segundo Diniz e Junior (2014), geradores de tráfego permitem que sejam gerados fluxos de dados com características específicas para simular o acesso a uma aplicação, como o tráfego de voz ou vídeo, por exemplo.

Diversas são as ferramentas de medição ativa disponíveis na Internet, tanto de código

aberto como *Clink* (DOWNEY, 1999), *Iperf* (TIRUMALA et al., 2005), *Netperf* (Netperf, 2017), *Pathrate* (DOVROLIS; PRASAD, 2004), quanto pagas como *AppNeta Performance Manager* (APPNETA, 2014).

A escolha recaiu sobre o *Iperf* por permitir a coleta de métricas interessantes (taxa de transferência, *jitter* e taxa de pacotes perdidos) além de possibilitar, como uma de suas principais vantagens, a alteração de parâmetros TCP, tal como o tamanho da janela TCP, exibir relatórios de banda nos modos TCP e UDP e relatórios de *jitter* e perda de pacotes no modo UDP (DINIZ; JUNIOR, 2014).

O *Iperf* é uma ferramenta desenvolvida com código livre e gratuita, do tipo cliente/servidor para medição e geração de tráfego de dados TCP (*Transport Control Protocol*) e UDP (*User Datagram Protocol*).

Trata-se de uma aplicação com duas vertentes, servidor e cliente, que funcionam em complementaridade: cabe ao cliente gerar o tráfego que será recebido pelo servidor, permitindo assim as medições referidas no parágrafo anterior.

Este *software* possui ainda a vantagem de ser possível configurar vários parâmetros, tais como o tipo de tráfego gerado, o tamanho das mensagens enviadas ou a duração da experiência.

A título de exemplo, para correr uma experiência com tráfego UDP, de duração de 60 segundos, com relatórios das estatísticas de 5 em 5 segundos foram utilizadas as seguintes instruções:

- Do lado do servidor: iperf –s –u –i 5
- Do lado do cliente: iperf –c 192.168.0.3 –u –i 5 –t 60

2.7.5 *Firewall*

Um *firewall* é um sistema, ou um conjunto de sistemas, que força uma política de segurança entre uma rede interna segura e uma rede insegura, como a Internet (NETO, 2004).

O *Iptables* é o principal *firewall* para o *Linux*, e de longe o mais utilizado pelos administradores de sistemas. Isto se dá graças a sua portabilidade e confiabilidade, além da facilidade de manutenção e manipulação de seu banco de regra (NETO, 2004).

Firewall em nível de pacotes, o *iptables* funciona baseado em endereços/portas de origem/destino e desempenha suas funções através da comparação de regras, organizadas e armazenadas em tabelas internas, para saber se um pacote tem ou não permissão para adentrar a rede/máquina que está sendo protegida. Em configurações mais restritivas, o pacote é bloqueado e registrado para que o administrador do sistema tenha condições de avaliá-lo posteriormente (FERRAREZI; GROSSI; MARCHI, 2017).

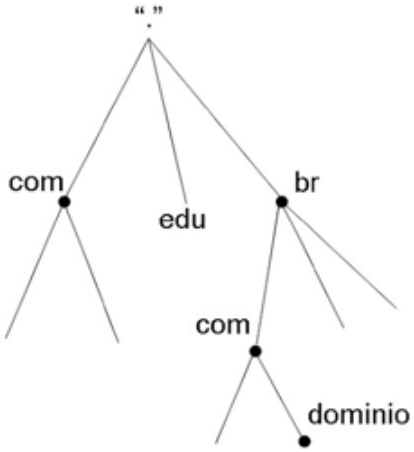
O *iptables* também pode ser usado para modificar e monitorar o tráfego da rede, fazer NAT (*Network Address Translation*), redirecionamento e marcação de pacotes, modificar a prioridade de pacotes que entram e saem do seu sistema, contagem de *bytes*, dividir tráfego entre máquinas e criar proteções contra várias técnicas de ataque (FERRAREZI; GROSSI; MARCHI, 2017).

2.7.6 Sistema de Nomes de Domínio (DNS)

O Sistema de Nomes de Domínio (ou *Domain Name System* - DNS) é responsável por traduzir nomes de domínios existentes para endereços IP usando o protocolo UDP, na porta 53. O DNS lida com o mapeamento entre os nomes do *host*, convenientes para nós, seres humanos, e entre os endereços na Internet, com os quais o computador lida (ALBITZ; LIU, 2002).

O DNS está especificado na RFC 1035 (MOCKAPETRIS, 1987) e é basicamente um banco de dados de informações de *hosts*. O banco de dados distribuído do DNS é indexado por nomes de domínios. Cada nome de domínio é essencialmente apenas um caminho em uma grande árvore invertida, denominada espaço de nomes de domínio. A Figura 10 apresenta um esquema de parte da árvore de domínios da Internet.

Figura 10 – Árvore de domínio DNS



Fonte: (COSTA, 2006)

A árvore de domínios do DNS é dividida em zonas. Cada zona pode conter informações de outras zonas e de *hosts*.

O DNS é implementado por meio de uma aplicação cliente-servidor. O cliente é o *resolver*, que contém um conjunto de rotinas em uma implementação de TCP/IP, que permite a consulta a um servidor, e um servidor geralmente é o programa BIND ou uma implementação específica de um servidor de DNS. Os servidores DNS são configurados para possuir autoridade sobre uma ou mais zonas (COSTA, 2006).

As especificações do DNS definem três tipos de servidores:

- *servidor primário (mestre)*: obtém os dados acerca das zonas sobre as quais ele tem autoridade, a partir de arquivos previamente configurados. Os servidores primários são autorizados, isto é, possuem informações completas e atualizadas a respeito de seus domínios (ALBITZ; LIU, 2002).

- *servidor secundário (escravo)*: obtém os dados acerca de suas zonas de autoridade a partir de outros servidores que possuem autoridade sobre essas zonas (ALBITZ; LIU, 2002).

- *servidor de cache*: definido como um servidor que guarda em cache consultas que já foram anteriormente solicitadas, de forma a melhorar a performance da resolução de nomes (ALBITZ; LIU, 2002).

Além desses servidores, outro menos utilizado é o *forwarder* que remete a solicitação para outros servidores de nomes (COSTA, 2006).

O servidor de nomes utilizado na grande maioria dos servidores da Internet é o BIND (*Berkeley Internet Name Domain*), provendo uma estável e robusta arquitetura sobre a qual as organizações podem construir sua estrutura de nomes (COSTA, 2006).

O BIND é um software de código aberto que permite publicar suas informações do DNS na Internet e resolver consultas de DNS para seus usuários (ALBITZ; LIU, 2002).

No capítulo seguinte é apresentada a revisão bibliográfica sobre o tema Virtualização de Funções de Rede em Nuvem.

3

Trabalhos Relacionados

Este capítulo é dedicado à apresentação de uma revisão sistemática da literatura realizada sobre o tema Virtualização de Funções de Rede em Nuvem, especificamente buscando identificar respostas para as questões propostas além de apresentar trabalhos relevantes já publicados com aplicações em ambientes de experimentação (*testbeds*), tanto em cenários reais como em ambientes simulados, que apresentam relação com o tema desenvolvido nessa dissertação.

Este capítulo está organizado da seguinte forma: a Seção 3.1 apresenta o estado da arte em relação às formas de virtualização de funções de rede utilizadas em ambientes de computação em nuvem que sejam mais propícias a atender as necessidades de uma administração de redes utilizar os recursos de TI, dividido na Subseção 3.1.1, que descreve o método utilizado na pesquisa, e na Subseção 3.1.2, que apresenta a análise dos resultados. Na Seção 3.2, são apresentados outros trabalhos relacionados que contribuíram para a tomada de decisões e na Seção 3.3 são apresentadas as considerações finais.

3.1 Revisão Sistemática da Literatura

Mesmo com iniciativas diferentes, NFV e Computação em Nuvem procuram basicamente através da abstração, dissociar as funções de rede do *hardware* específico e o resultado disso é que o *software* e a função que realiza, não estarão mais submetidos a um *hardware* dedicado e caro. Essa combinação também oferece a consolidação e simplificação da infraestrutura com soluções mais econômicas e sustentáveis, proporcionando a opção de se administrar uma rede com equipamentos de diferentes fabricantes.

3.1.1 Método da Revisão

Esse mapeamento sistemático tem como objetivo estudar e mapear o estado da arte em relação às formas de virtualização de funções de rede utilizadas em ambientes de computação

em nuvem que sejam mais propícias a atender as necessidades de uma administração de redes utilizar os recursos de TI.

Ao adaptarem o método de revisões sistemáticas utilizado na medicina e nas ciências sociais para guiar a construção de revisões em diversos tópicos da engenharia de software, (KITCHENHAM, 2004) definiram um mapeamento sistemático da literatura (*SLM*, *Systematic Literature Mapping*) como um meio de identificar, avaliar e interpretar todas as pesquisas disponíveis relevantes para a questão de pesquisa específica ou área temática ou fenômeno de interesse.

Este trabalho de mapeamento sistemático da literatura seguirá as diretrizes definidas por Petersen e outros (2008) , compostas por quatro etapas principais:

1. Definição das questões de pesquisa;
2. Busca e seleção dos estudos relevantes;
3. Extração de dados;
4. Análise dos resultados.

Nas subseções seguintes estão descritos, de forma detalhada, o processo de busca e seleção dos estudos primários.

3.1.1.1 Questões de Pesquisa:

Neste mapeamento definiu-se as seguintes questões de pesquisa que guiaram a condução do trabalho:

Q1) Como se deu a evolução histórica das publicações sobre o uso de NFV em Cloud Computing?
Q2) Qual(is) função(ões) de rede são mais virtualizadas?

3.1.1.2 Estratégias de Busca e de Seleção

A busca ocorreu de forma manual tomando como referência a estratégia composta de três elementos: as fontes onde se realizaria a busca, quais os idiomas desejados para o trabalho e quais palavras-chaves a serem aplicadas na busca.

- Fontes de busca: bases de dados na área de computação: *ACM Digital Library* (*ACM*), *IEEE Xplore* (*IEEE*), *Science Direct* (*SD*), *Google Scholar* (*GS*) e a Biblioteca Digital Brasileira de Computação (BDBComp). Para o uso sem restrições de *download* em algumas bases foi utilizado o portal de periódicos da CAPES (http://www.periodicos.capes.gov.br).

- Idioma dos trabalhos: Inglês, por ser universalmente aceita para trabalhos científicos; Português, que permitisse obter publicações nacionais além do fato de ser a língua nativa dos revisores.

- Palavras-chaves: em inglês "*nfv*", "*network function virtualization*", "*cloud*" e "*performance*". Durante as pesquisas, percebeu-se que a palavra-chave derivada "*network function virtualisation*" era frequente nos artigos analisados sendo portanto inserida ao termo de busca. Em português, foram adotadas "*nfv*", "*virtualizacao de funcao de rede*", "*computacao em nuvem*" e "*desempenho*" e para obter um maior resultado foram inseridas também as palavras "*cloud*" e "*performance*".

Na execução da busca, de acordo com a estratégia estabelecida, foram utilizadas as ferramentas de filtragem de cada base visando considerar somente o título, resumo e palavras-chave dos artigos, excluindo trabalhos que claramente eram irrelevantes para as questões investigadas. Entretanto, na base BDBComp foi realizada uma pesquisa simples, uma vez que não havia mecanismo de busca avançado disponível.

A partir das palavras-chaves apresentadas, foram elaboradas algumas *strings* de busca que foram submetidas às fontes de busca relatadas anteriormente:

Em Inglês: (("nfv" OR "*network function virtuali**") AND ("*cloud*") AND (*"performance"*)).

Em Português: (("nfv" OR "virtualizacao de func* de rede") AND ("computacao em nuvem" OR "cloud") AND ("desempenho" OR "*performance*")).

Com base nestes termos de busca, em julho de 2016 foram realizadas as consultas nas bases indicadas, retornando um total de 338 artigos. Os resultados das buscas estão apresentados na Tabela 1 onde se é possível verificar a quantidade de artigos obtidos das consultas com os termos em inglês e português.

Tabela 1 – Resultados das buscas nas bases de dados utilizando os termos de busca

Bases de dados	**Inglês**	**Português**
ACM	38	–
IEEE Xplore	161	–
Science Direct	72	–
Scopus	62	–
Google Scholar	–	5
BDBComp	–	0
TOTAL	**338**	

Na pesquisa em inglês, *IEEE Xplore* foi a base que apresentou o maior número de resultados com 161 classificados, que corresponde a aproximadamente 47,6% do total do estudos

coletados. Na pesquisa em português, *Google Scholar* foi a base que apresentou o maior número de resultados com 5 classificados, que corresponde a aproximadamente 1,54% do total coletado.

Após a execução da busca, o próximo passo foi a filtragem dos artigos encontrados com base nos critérios de seleção.

3.1.1.3 Critérios de Seleção

Como a aplicação das expressões de busca é restrita ao aspecto sintático, a seleção preliminar, com o uso da expressão de busca, não garante que todo o material coletado seja útil no contexto da pesquisa. Desta forma, os resumos (*abstracts*) das publicações retornadas foram lidos e analisados seguindo os critérios de inclusão e exclusão definidos a seguir, verificando se a publicação propõe ou descreve como sua principal contribuição o uso de NFV.

Com o objetivo de automatizar, ao máximo, as tarefas envolvidas durante a execução deste mapeamento, foi utilizada como suporte a ferramenta START (*State of the Art through Systematic Review*) (ZAMBONI et al., 2010).

Foram utilizados os seguintes critérios de inclusão de estudos:

1. Está disponível na web;
2. Apresenta textos completos dos estudos em formato eletrônico;
3. Contém as principais palavras-chave;
4. Contém alguma experiência com relação à virtualização de rede aplicada em *cloud*.

Quanto aos critérios de exclusão, definiu-se:

1. Idioma diferente do inglês ou português;
2. Mesmo autor e tema (duplicados);
3. Publicações anteriores a 2012;
4. Não disponibiliza o resumo (*abstracts*);
5. Não disponibiliza texto completo para leitura;
6. Não aborda o tema virtualização de rede em relação a *cloud*.

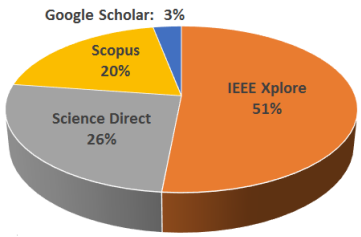
Após a aplicação dos critérios de inclusão e de exclusão dos artigos encontrados, foram recuperados, das bases adotadas nesse mapeamento, 66 trabalhos relacionados ao tema proposto, conforme demonstrada na Tabela 2. Nota-se um alto índice de redução no número total de artigos, já que apenas 19,5% deles foram qualificados pelos critérios de seleção.

Tabela 2 – Resultados das buscas nas bases de dados e da aplicação dos critérios de seleção

Bases de dados	**Resultados das buscas**	**Aplicação dos Critérios de Seleção**
ACM	38	0
IEEE Xplore	161	34
Science Direct	72	17
Scopus	62	13
Google Scholar	5	2
BDBComp	0	0
TOTAL	**338**	**66**

A Figura 11 ilustra a distribuição da contribuição de cada base onde se observa também a grande representatividade da base *IEEE Xplore*, com 51,5% (34/66) do total de trabalhos selecionados. Como as bases *ACM* e BDBComp não apresentaram resultados, não estão demonstradas na figura.

Figura 11 – Contribuição de cada base para o total de estudos primários selecionados



O próximo passo, foi a leitura dos trabalhos que passaram por todas as etapas de seleção.

3.1.2 Análise de Resultado da Revisão

Nesta seção, são apresentados os resultados da análise dos estudos primários que respondem as questões de pesquisa deste mapeamento.

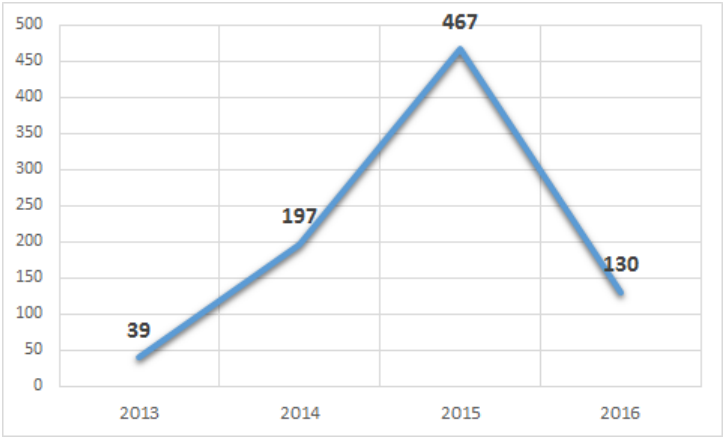
Q1) Como se deu a evolução histórica das publicações sobre o uso de NFV em Cloud Computing?

Nessa questão, que tem aspecto histórico, busca-se a evolução das publicações sobre NFV aplicada a ambientes de Computação em Nuvem (*Cloud Computing*). A Figura 12 ilustra o crescimento no número de publicações ao longo dos anos.

Foram publicados 833 trabalhos, demonstrando assim o aumento do interesse da comunidade científica na temática virtualização de funções de rede em nuvem.

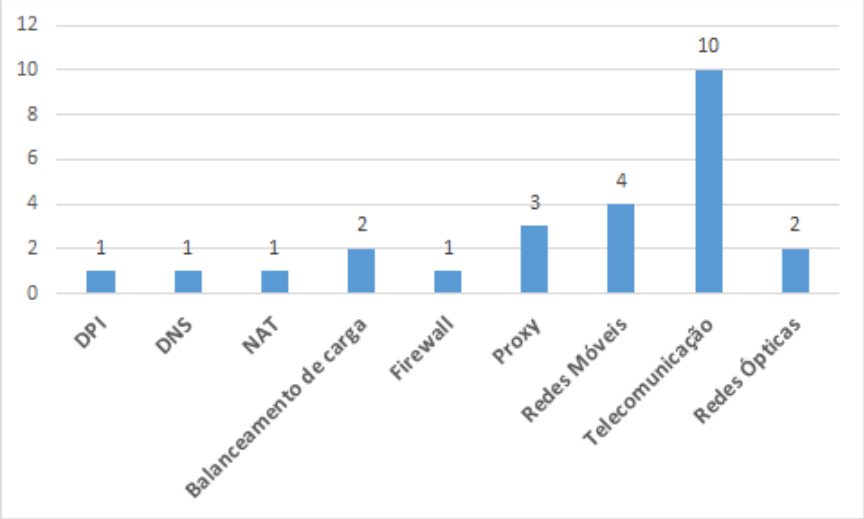
Q2) Quais as funções de rede mais virtualizadas num ambiente de computação em nuvem (Cloud Computing)?

Figura 12 – Quantidade de publicações por ano (2013 a 2016)



Essa questão de pesquisa visa traçar tendências que apontem as funções de rede que são mais virtualizadas nos trabalhos selecionados. A Figura 13 ilustra o resultado que indica que funções empregadas nas redes de telecomunicação está na ponta da escolha.

Figura 13 – Funções de rede mais virtualizadas



A tabela 3 lista os artigos escolhidos que foram utilizados para realizar a contagem das funções de rede nos mais citadas nos trabalhos selecionados nesse mapeamento.

3.2 Áreas correlatas

Esta seção é dedicada à apresentação de outros trabalhos relevantes já publicados acerca da temática da virtualização de funções de rede em nuvem com aplicações em ambientes de experimentação (*testbeds*), tanto em cenários reais como em ambientes simulados, que apresentam relação com o tema desenvolvido nessa dissertação.

Tabela 3 – Referência dos artigos ilustrados na Figura 13

Função usada	**Referência**
DPI	(GE et al., 2014)
DNS	(HEIDEKER; KAMIENSKI, 2015)
NAT	(HEIDEKER; KAMIENSKI, 2016)
Firewalls	(DECUSATIS; MUELLER, 2014)
Balanceamento de Carga	(SOARES; SARGENTO, 2015); (TRAJANO; FERNANDEZ, 2015)
Redes Móveis	(CHERKAOUI; ROSENBERG, 2013); (AHMAD et al., 2016); (TALEB; KSENTINI; KOBBANE, 2014); (NAUDTS et al., 2016a)
Redes Óticas	(CANNISTRA et al., 2014); (VILALTA et al., 2016)
Redes de Telecomunicação	(NAUDTS et al., 2016b); (MANZALINI; CRESPI, 2016); (SKöLDSTRöM et al., 2014); (AGYAPONG et al., 2014); (GARAY et al., 2016); (TALEB; KSENTINI; SERICOLA, 2016); (KSENTINI; TALEB; MESSAOUDI, 2014); (RAJAN, 2016); (SOARES et al., 2015); (ZHIQUN et al., 2013)

3.2.1 Ambiente de Experimentação de Nuvem

OpenStackEmu (BENET et al., 2017) é um projeto que visa fornecer um ambiente de experimentação de nuvem personalizável e controlado que permita a realização de pesquisas em nuvem, de rede em nuvem ou calcular a infraestrutura necessária para testar seus algoritmos e protocolos em larga escala.

Segundo os autores, com o OpenStackEmu os experimentadores podem estudar o impacto de novas estratégias de migração de VM no desempenho de aplicativos reais em uma rede de larga escala emulada intra ou inter-DC, o impacto das novas funcionalidade de roteamento ou controle de tráfego usando o SDN na comunicação VM a VM ou o tráfego de gerenciamento de nuvem ou o impacto de diferentes topologias de DC no desempenho da aplicação na nuvem.

A arquitetura do OpenStackEmu combina a infraestrutura do *OpenStack* (OPENSTACK, 2011) com o emulador de rede *Common Open Research Emulator* (CORE) (AHRENHOLZ et al., 2008) e um controlador SDN, o OpenDaylight (Opendaylight, 2017). No nó emulador que recebeu o CORE, foram criadas as VMs que simularam a emulação de rede em larga escala proposta pelos autores. A conexão dos nós físicos do *OpenStack* com a rede emulada no CORE se deu através da interface TUN/TAP do CORE. A partir daí, foi injetado tráfego das VMS criadas no *OpenStack*, para as VMs dentro da rede emulada no CORE utilizando *DC Traffic Generator* (DCT^{2}Gen) (WETTE; KARL, 2014).

Com essa configuração, os autores propuseram outro cenário de demonstração do OpenStackEmu, avaliando o desempenho de aplicativos reais que são executados dentro das VMs *OpenStack*. Como um exemplo de uma aplicação em tempo real, foi configurado um serviço de vídeo sob demanda em uma VM, um servidor web em outra VM e um servidor *proxy* numa terceira VM, todas em nós de computação diferentes dentro da nuvem *OpenStack*. Dessa forma

foi possível avaliar o tráfego VM-to-VM percorrendo a topologia emulada, uma vez que as VMs estão localizadas em diferentes CNs, além de também avaliar os impactos que a migração de VM entre os nós poderia causar.

O trabalho de Vilalta e outros (2015) apresenta a rede de transporte multi-domínio e a plataforma de computação em nuvem SDN/NFV do ambiente de experimentação *Adrenaline*, instalado no CTTC (*Centre Tecnològic Telecomunicacions Catalunya*) em Castelldefels (Barcelona, Espanha). Nesse ambiente, os serviços NVF são fornecidos graças a uma orquestração integrada de recursos de TI e de rede, a fim de prover conectividade de rede intra/inter *Data Centers* e implantar VMs utilizando o sistema de computação em nuvem *OpenStack*.

Os autores apresentam detalhes da implementação da plataforma de computação em nuvem na qual foi utilizada o *OpenStack* versão Havana em cinco servidores físicos cada um com 2 processadores Intel Xeon E5-2420, 32 GB de RAM e 2 TB de disco. Um servidor é dedicado ao controlador *OpenStack* e os outros quatro atuam como *OpenStack Compute Hosts* para instanciação de VM. Além disso, na conexão intra-*data center* são utilizados quatro *switchs* OpenFlow com várias placas de interface de rede 1Gb (NICs) executando o OpenVSwitch (OVS), e a conexão inter-*datacenter* é fornecida através de uma rede óptica com GMPLS (*Generalized Multi Protocol Label Switching*).

Ainda no mesmo trabalho, os autores propuseram dois casos de uso para validar a viabilidade da arquitetura proposta: o primeiro deles trata da virtualização da função de rede PCE (*Path Computation Element*), que é responsável pelo mecanismo de cálculo de rota para criação de caminhos na rede de transporte. O segundo caso de uso é a implantação de controladores SDN virtuais em cima de redes virtuais.

O ambiente de experimentação *Adrenaline* também é citado nos trabalhos de Vilalta e outros (2016) e Muñoz e outros (2017). No primeiro trabalho, os autores propõem a combinação de virtualização de rede óptica e virtualização de funções de rede (NFV) para a implantação de redes ópticas virtuais (VON - *Virtual Optical Networks*) controladas sob demanda por *OpenFlow*. No segundo trabalho, a arquitetura do ambiente *Adrenaline* é composta pela infraestrutura *Cloud*/*Fog* (nuvem na borda da rede) (STOLFO; SALEM; KEROMYTIS, 2012), as redes intra-DC, as redes heterogêneas multidomínios *Wireless*/Óptica e os planos de controle/gerenciamento para atividades experimentais relacionadas aos serviços 5G, a quinta geração de tecnologia de redes móveis.

Outro trabalho sobre ambiente de experimentação em nuvem para NFV é o de Chou e outros (2016), no qual foi projetado e implementado um *testbed* SDN para fornecer um ambiente de laboratório virtual para avaliar as implementações SDN/NFV. O modelo apresentado é dividido em três camadas: a primeira é um centro de gestão integrada, responsável por enviar as instruções para os módulos de gerenciamento de recursos gerenciamento e *slice*. A segunda é a camada de gerenciamento e a terceira camada é a infraestrutura de recursos, tendo *OpenStack* como fornecedor de recursos de computação para estabelecer um ambiente de rede virtual.

O experimento apresentado no trabalho, utilizou uma abordagem de escolha do caminho mais curto, para selecionar o melhor uso do recurso de rede virtual, aumentando a utilização da rede no *testbed*, que provocou uma redução do atraso de transmissão de 3.21ms para 0.46ms, equilibrando o consumo de largura de banda entre os serviços de rede virtual, melhorando o desempenho da rede.

3.2.2 Geração de Tráfego

Naik e outros (2016) em seu artigo descrevem o NFVPerf, uma ferramenta de monitoramento de desempenho e detecção de gargalo para NFV. O NFVPerf funciona monitorando passivamente o tráfego através do gráfico de encaminhamento do VNF, calculando os *throughputs* e atrasos da camada de aplicação e identificando gargalos de desempenho com base em uma degradação dessas métricas. O design do NFVPerf é genérico o suficiente para trabalhar em uma variedade de VNFs.

Os autores utilizaram o *Iperf* para enviar e receber pacotes UDP entre as VMs no máximo possível e usar o NFVPerf para capturar e analisar os pacotes. O *script* de captura executado foi construído em *Python* e funcionou em dois modos: com DPI (*Deep Packet Inspection* - inspeção profunda de pacotes) para emular pacotes de análise com um VNF real, e sem DPI, para quantificar a sobrecarga da base. O cliente e o servidor *Iperf* funcionaram em uma VM de núcleo único, em uma mesma máquina física, bem como em máquinas físicas diferentes. Naik e outros (2016) concluem que a medição *on-line* do NFVPerf não adiciona muito erro em comparação com a captura de pacotes *offline*.

No trabalho de Ma e outros (2015) são estudados os efeitos de como implementar de forma eficiente *middleboxes* (VNFs) como máquinas virtuais (VMs) para alcançar o balanceamento de carga usando uma abordagem de rede definida por software (SDN) considerando os efeitos de mudança de tráfego de diferentes desses *middleboxes*. Para validar o projeto, os algoritmos propostos foram implementados em um protótipo de sistema com o controlador SDN de código aberto *Floodlight* e a plataforma de emulação *Mininet*. Além disso, foram realizadas simulações em *ns-3* para avaliação de desempenho em redes de grande escala e para a geração de tráfego, foi utilizado o *Iperf* nos *hosts* em modo UDP.

Akhtar e outros (2016), descrevem um caso de uso para gerenciamento de NFV usando SDN e a teoria de controle. Para isso utilizaram a arquitetura de gerenciamento do RINA (a Arquitetura de Inter-redes Recursiva limpa) para gerenciar as instâncias da Função de Rede Virtual (VNF) no ambiente de experimentação GENI (Global Environment for Network Innovations). Foi implantado o *Snort*, um Sistema de Detecção de Intrusão (IDS) como VNF. A topologia de rede possui *hosts* de origem e de destino, IDSs múltiplos, um *VSwitch* aberto (OVS) e um controlador *OpenFlow*.

Um aplicativo de gerenciamento distribuído executado no RINA mede o estado das

instâncias do VNF e comunica essas informações para um controlador que, em seguida, fornece informações de balanceamento de carga para o controlador *OpenFlow*. O último controlador, por sua vez, atualiza as regras de encaminhamento do fluxo de tráfego no *switch* OVS, balanceando a carga nas instâncias do VNF. O tráfego TCP foi gerado usando a aplicação *Iperf*, variando o número de instâncias do iPerf para alterar a quantidade de carga de tráfego nas instâncias do VNF.

Com isso, os autores demonstram os benefícios de usar essa abordagem de balanceamento de carga com a teoria de controle e a arquitetura de gerenciamento RINA em ambientes virtualizados para gerenciamento de NFV ilustrando também que o GENI pode suportar facilmente uma ampla gama de experimentos relacionados com SDN e NFV.

Callegati e outros (2016), apresentam algumas ideias sobre como uma plataforma de computação em nuvem de código aberto, como o *OpenStack*, implementa a virtualização de rede multi-inquilino e como ela pode ser usada para implantar o NFV, enfocando em particular os problemas de desempenho de encaminhamento de pacotes. Para este propósito, é apresentado um conjunto de experiências que se referem a uma série de cenários inspirados na computação em nuvem e nos paradigmas do NFV, considerando os cenários de um único e de multi-inquilinos. A partir dos resultados da avaliação foi possível destacar potencialidades e limitações de execução do NFV no *OpenStack*.

3.2.3 Migração de IPv4 para IPv6

Baseado nos experimentos de Barayuga e Yu (2014) que em seu trabalho realizaram testes em uma rede experimental utilizando as tecnologias NAT44 (IPv4), NAT64 e IPV6 no contexto de uma migração para IPv6 utilizando o IPERF como gerador de carga em modo UDP com diferentes conjuntos de tamanho de carga e nível de concorrência.

Em outro trabalho, Barayuga e Yu (2015) repetiram os mesmos testes porém utilizando o IPERF em modo TCP com as métricas tempo de transferência, vazão e taxa de transferência.

Os trabalhos correlatos analisados nesta seção podem ser classificados de acordo com os critérios de ambiente de experimentação, orquestrador de nuvem, emulador de rede e gerador de tráfego utilizados, conforme apresentado na Tabela 4.

3.3 Considerações Finais

A realização deste mapeamento sistemático possibilitou uma análise dos estudos encontrados e o melhor conhecimento sobre como a virtualização de funções de rede em um ambiente de computação em nuvem pode ajudar a reduzir custos aquisitivos e operacionais nas organizações sejam elas, públicas ou privadas.

A condução do processo de mapeamento através de um protocolo de busca e seleção de

Tabela 4 – Comparação dos Trabalhos Correlatos

Autores	NFV	Ambiente de Experimentação	Orquestrador de Nuvem	Emulador de Rede	Gerador de Tráfego
(BENET et al., 2017)	Sim	-	*OpenStack*	CORE	DCT^2Gen
(VILALTA et al., 2015)	Sim	*Adrenaline*	*OpenStack*	-	-
(VILALTA et al., 2016)	Sim	*Adrenaline*	*OpenStack*	-	-
(MUñOZ et al., 2017)	Sim	*Adrenaline*	*OpenStack*	-	-
(CHOU et al., 2016)	Sim	Próprio	*OpenStack*	-	-
(CALLEGATI; CERRONI; CONTOLI, 2016)	Sim	Próprio	*OpenStack*	-	*Iperf*
(NAIK; SHAW; VUTUKURU, 2016)	Sim	Não utiliza	Não utiliza	*Mininet*	*Iperf*
(MA; MEDINA; PAN, 2015)	Sim	Não utiliza	Não utiliza	*Mininet*	*Iperf*
(AKHTAR; MATTA; WANG, 2016)	Sim	GENI	-	-	Iperf
(BARAYUGA; YU, 2014)	Não	-	-	-	Iperf
(BARAYUGA; YU, 2015)	Não	-	-	-	Iperf

estudos foi o método adotado nesse trabalho. Os trabalhos utilizados nessa análise são recentes, haja vista o próprio tema ser novo na academia. A definição dos termos de busca que foram executados nas bases de dados citada nesse trabalho, trouxeram um resultado de 338 trabalhos relacionados ao tema proposto, sendo que 66 deles foram selecionados para leitura.

A partir dessas informações pode-se concluir que, para uma busca mais efetiva, as *strings* devem ser analisadas e elaboradas novamente, outros sinônimos e algumas palavras-chaves mais específicas poderiam ter sido inseridas na *string* de busca. A inclusão de outras fontes de busca poderiam enriquecer ainda mais os resultados alcançados. O uso de uma ferramenta permitiu reduzir o tempo necessário para a realização das etapas, permitindo responder às questões da pesquisa aqui levantadas de forma mais ágil e confiante.

Os resultados da revisão demonstram o grande interesse da academia em buscar soluções para virtualizar uma grande parte das funções de rede utilizadas pelas operadoras de telecomunicação, mas sem esquecer sua aplicação em redes privadas.

Com os resultados e análises obtidos nesse mapeamento, acredita-se que esta pesquisa apresenta resultados relevantes à academia e aos empreendedores, mas percebe-se também que muito trabalho ainda pode ser feito pois várias expectativas ainda não foram alcançadas, sendo necessário trabalhar em pesquisas a fim de desenvolver outras funcionalidades em diferentes contextos.

Finalmente, com os demais trabalhos relacionados pode-se então obter os conhecimentos necessários para a implantação de virtualização de funções rede, incluindo ferramentas utilizadas pela academia, como o emulador conectado a nuvem, a migração de protocolos IPv4 para o IPv6

e a geração de tráfego. Por serem recentes, indicam a atualidade do assunto demonstrando o limite atual de desenvolvimento do NFV.

4

Ambiente de Experimentação

4.1 Introdução

As emulações apresentadas neste trabalho foram realizadas no ambiente do Laboratório Experimental em Redes de Computadores (ELAN - *Experimental LAboratory in computer Networks*) disponibilizado pelo Grupo de Pesquisa em Redes e Computação Distribuída (GPR-Com, 2006) do Departamento de Computação da Universidade Federal de Sergipe (UFS).

O ELAN tem como um dos seus principais papéis, permitir a execução de simulações e serviços dos pesquisadores envolvidos. Para tanto, a infraestrutura provida é voltada para experimentos e simulações de rede e sistemas distribuídos possibilitando aos pesquisadores um ambiente propício e adequado para a realização de testes.

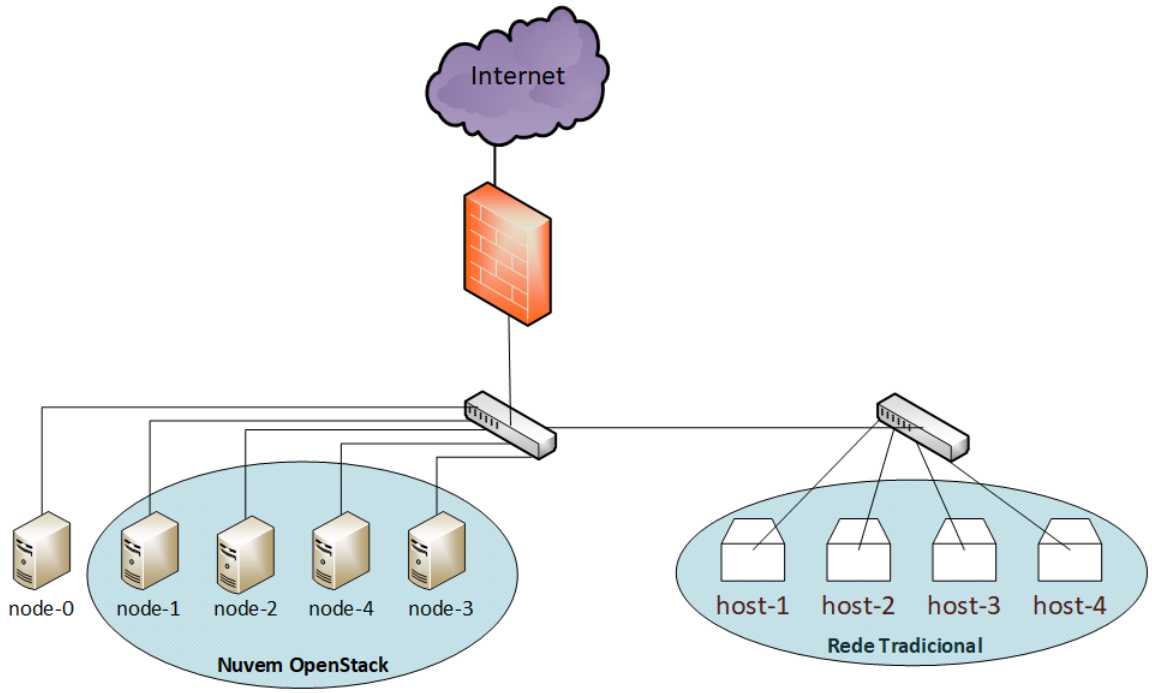
A montagem de toda a infraestrutura do ELAN deu-se início em janeiro de 2017, período no qual todo o processo de aquisição e recebimento dos equipamentos foi concluído. Em meados desse mesmo mês, após completado o serviço de cabeamento estruturado, foi realizada a configuração dos equipamentos de conexão à Internet como *firewall* e roteador *Wireless*.

Em uma outra etapa, procedeu-se a implantação do ambiente de nuvem do ELAN, encerrando com a configuração e conexão dos *hosts Linux*. Todo o processo de montagem, instalação e configuração foi concluído na primeira quinzena de abril de 2017, estando então plenamente disponível para a realização dos testes. Mais detalhes sobre a topologia do ELAN, estão descritos nas Seções 4.2 e 4.3.

4.2 Topologia Física

A Figura 14 apresenta a topologia física do ELAN demonstrando a distribuição dos diversos equipamentos na infraestrutura do laboratório.

Figura 14 – Topologia Física do ELAN



Na Tabela 5 estão listadas as características dos equipamentos que formam o ambiente de Rede NFV em nuvem e de Rede Tradicional do laboratório do ELAN.

Tabela 5 – Características dos Equipamentos de Rede NFV e Tradicional do ELAN

Nome	**Descrição**
node-0	Processador Intel Core 2 Duo 2.80 GHz, memória RAM 4 GB, 500 GB de disco rígido e duas interfaces de rede Gigabit (10/100/1000).
node-1	Processador Intel Xeon 1.90 GHz, memória RAM 16 GB, 1 TB de disco rígido e quatro interfaces de rede Gigabit (10/100/1000).
node-2	Processador Intel Xeon 2.80 GHz, memória RAM 48 GB DDR4, 1 TB de disco rígido e quatro interfaces de rede Gigabit (10/100/1000).
node-3	Processador Intel Xeon 1.90 GHz, memória RAM 8 GB, 1 TB de disco rígido e quatro interfaces de rede Gigabit (10/100/1000).
node-4	Processador Intel Xeon 2.40 GHz, memória RAM 32 GB, 1 TB de disco rígido e quatro interfaces de rede Gigabit (10/100/1000).
host-1	Processador Intel Core i3 3.20 GHz, memória RAM 4 GB, 500 GB de disco rígido e duas interfaces de rede Gigabit (10/100/1000).
host-2	Processador Intel Core 2 Duo 2.80 GHz, memória RAM 4 GB, 500 GB de disco rígido e duas interfaces de rede Gigabit (10/100/1000).
host-3	Processador Intel Core i3 3.20 GHz, memória RAM 4 GB, 500 GB de disco rígido e duas interfaces de rede Gigabit (10/100/1000).
host-4	Processador Intel Core 2 Duo 2.80 GHz, memória RAM 4 GB, 500 GB de disco rígido e duas interfaces de rede Gigabit (10/100/1000).

Para a interconexão dos equipamentos são utilizados *switches* que operam nas camadas 2 (enlace) e 3 (rede) do modelo OSI (*Open System Interconnection*), também conheci-

dos como *Layer-2* e *Layer-3*. Todos os *switches* são gerenciáveis, possuem 24 portas RJ-45 10/100/1000BASE-T e 4 portas 1000/10000 SFP+ (*Small Form-factor Pluggable*) além de suporte à Openflow.

A arquitetura de nuvem adotada baseia-se na plataforma *OpenStack* com uma infraestrutura que consiste de quatro servidores, denominados *node-1*, *node-2*, *node-3* e *node-4*. A versão do *OpenStack* instalada é a *Release Newton 2016.2.10.0* que dá suporte a SFC e é implementada sob o sistema operacional *Ubuntu 16.04*. A tecnologia *OpenStack* consiste de uma série de projetos interrelacionados que oferecem vários componentes como solução completa de IaaS entretanto, a sua instalação através de pacotes individualizados pode consumir muito esforço e tempo.

O processo de instalação do sistema operacional (Ubuntu 16.04), módulos *Openstack* e parametrização base, foram executados através da ferramenta *Fuel* já que todos os equipamentos apresentados na Tabela 5 atendem aos requisitos de *hardware* para a instalação do *OpenStack* utilizando-se o*Fuel* e o *node-0* foi o equipamento escolhido para receber a instalação da versão 10.1 do *Fuel*.

Com o *Fuel* instalado e atualizado, o próximo passo foi configurar o ambiente de nuvem utilizando então a interface gráfica gerada pelo *Fuel*. Nessa configuração, foram definidos quais recursos (computação, rede e armazenamento) do *OpenStack* foram atribuídos a cada *node*, a topologia de rede da nuvem e a forma como os dados serão armazenados nos servidores.

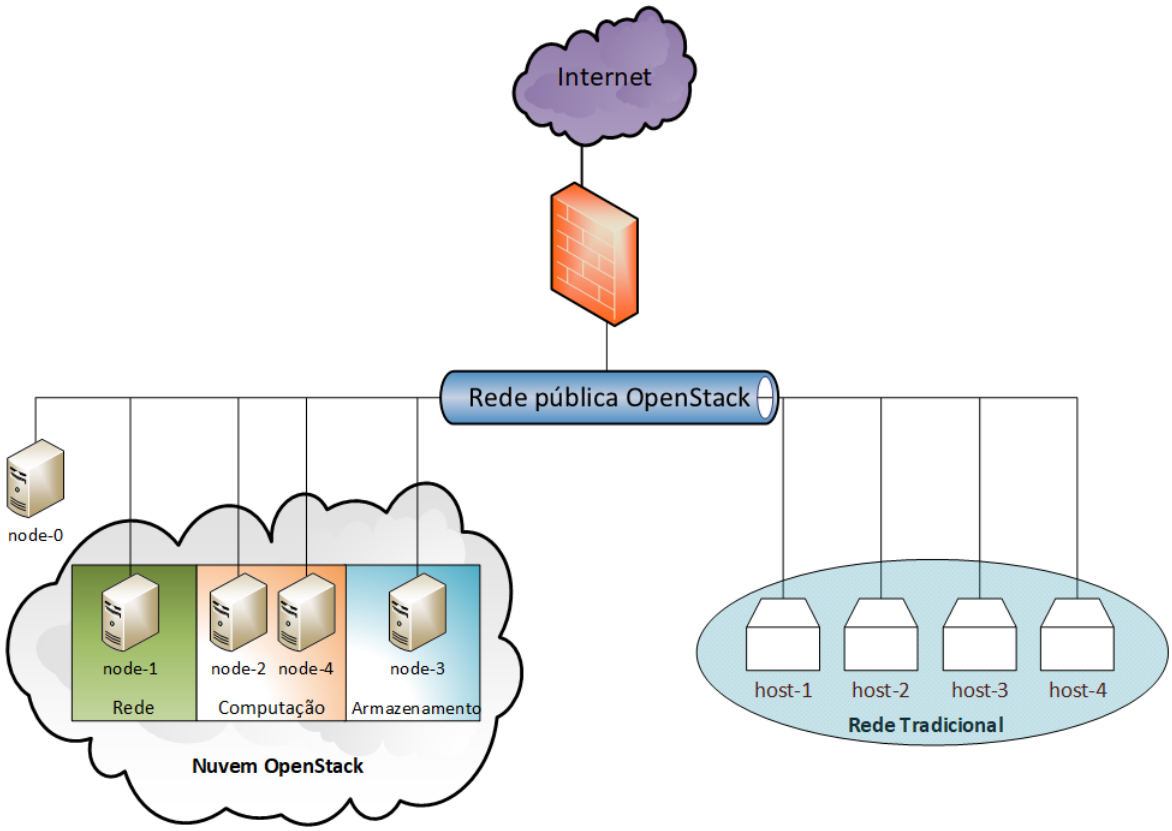
Os recursos gerenciados pelo *OpenStack* foram distribuídos entre os servidores, da seguinte forma:

- Controlador (Rede): *node-1*
- Computação: *node-2* e *node-4*
- Armazenamento e Telemetria: *node-3*

Os equipamentos *node-1*, *node-2*, *node-3* e *node-4* possuem cada um, quatro placas de rede que foram interligadas, cada uma, aos *switches* do ELAN. Nesses *switches*, foi necessário realizar a configuração de *vlan's* a fim de adequar a rede física aos requisitos de instalação da nuvem *OpenStack* já que foi adotada a topologia de segmentação por VLAN, onde uma VLAN é atribuída a cada inquilino da nuvem possibilitando assim a separação do tráfego de rede entre os grupos de rede do *OpenStack* conforme listados abaixo:

- Rede administrativa: permite a conexão do demais *nodes* da infraestrutura possibilitando a instalação da nuvem *OpenStack*.
- Rede pública: permite a conexão da nuvem *OpenStack* através de roteamento global da infraestrutura para ter acesso a rede interna e a Internet.

Figura 15 – Topologia Lógica do ELAN



- Rede de gerenciamento: possibilita acessar a infraestrutura de gerenciamento do *OpenStack*.
- Armazenamento e Redes privadas (VLANs): isolar de outras redes, não sendo roteável.

Concluída essa preparação, foi possível então realizar a instalação dos componentes do *OpenStack* nos servidores (*node-1*, *node-2*, *node-3* e *node-4*) e assim poder ter a nuvem *OpenStack* disponível para os experimentos onde, através da interface gráfica (*dashboard*) Horizon é possível acessar, provisionar e automatizar recursos da nuvem.

Quanto a arquitetura de rede tradicional, foram utilizados quatro *hosts Linux*. Um *host Linux* do ELAN é um computador com duas ou mais *interfaces* de rede que implementam funções de rede e tem capacidade de processamento (cpu, memória) similares, conforme especificado na Tabela 5.

Cada *host Linux* na rede tradicional recebeu uma denominação: *host-1*, *host-2*, *host-3* e *host-4* e estão interligados através de um *switch* que está conectado à VLAN pública do *OpenStack*, ou seja, tem conectividade com a nuvem. Em todos os quatro *hosts Linux* foi utilizado o sistema operacional *Ubuntu 16.04*.

4.3 Topologia Lógica

O ambiente está dividido em duas grandes áreas: uma estrutura de nuvem e outra de rede tradicional. Quando o experimento exigir uma estrutura em nuvem, utilizam-se instâncias (como são chamadas as máquinas virtuais - VMs em ambiente de nuvem) que são criadas em projetos específicos e são oferecidas a cada solicitação de utilização da infraestrutura. Encerrada a necessidade, essas instâncias criadas podem ser simplesmente removidas da plataforma. Nessa infraestrutura de nuvem podem ser realizados os experimentos de NFV.

Por outro lado, quando o experimento exigir uma estrutura tradicional, o ELAN possui um conjunto de computadores, os *hosts Linux* citados na Seção 4.2, aptos a atender as necessidades dos testes.

Além disso a infraestrutura computacional do laboratório, ilustrada na Figura 15, permite ainda a execução de testes não somente em cada ambiente isoladamente mas também entre eles, já que a conexão entre esses ambientes possibilita a realização dos experimentos de rede tradicional para rede em nuvem e vice-versa.

5

Virtualização de Funções de Rede

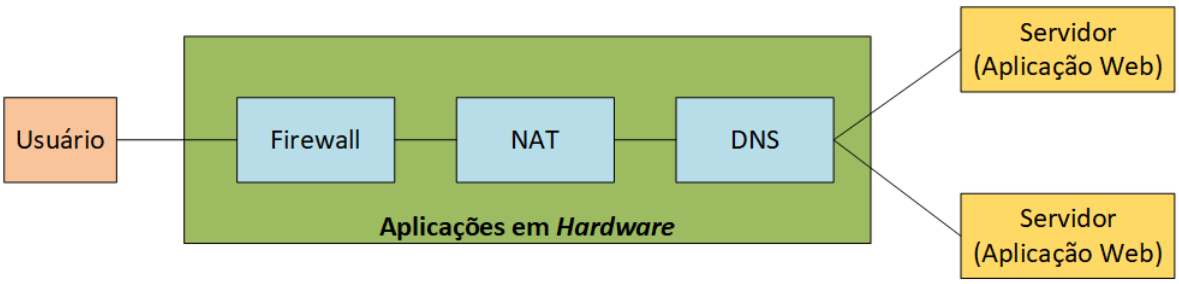
Neste capítulo, são descritos como foi implementado o cenário de virtualização de funções de rede.

A virtualização de funções de rede (*Network Functions Virtualization* - NFV) refere-se a virtualização dos serviços de rede para substituir os dispositivos de *hardware* dedicados e de altos custos, como roteadores e *firewalls*, por dispositivos baseados em *software*, executados como máquinas virtuais (VMs). Quando implementado corretamente, permite diminuir a quantidade de hardware proprietário necessária para operar serviços de rede.

O objetivo da NFV é desacoplar as funções de rede de dispositivos de *hardware* dedicados e as transferir para serem hospedadas em VMs, consolidando várias funções em um único servidor físico. Uma vez que as funções de rede estão sob o controle de um hipervisor, os serviços que exigem *hardware* dedicado podem ser executados em servidores x86 padrão.

Na Figura 16, está representada uma configuração típica de um *firewall*, um NAT, um DNS e vários servidores que hospedam o mesmo aplicativo *web* para alta disponibilidade. O *firewall*, o NAT e o DNS são um dispositivo de *hardware* real. Este é um exemplo típico de uma arquitetura que não é NFV.

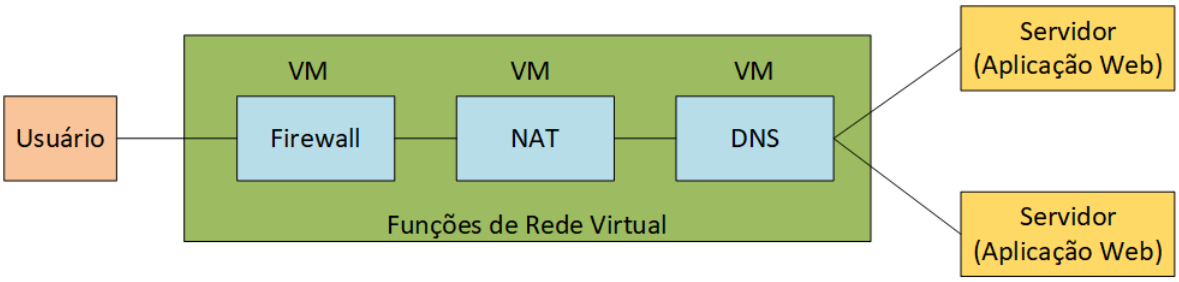
Figura 16 – Diagrama de Rede Tradicional



Na Figura 17, foi substituído o *firewall* de *hardware*, o NAT e o DNS por um *software* executado em máquinas virtuais (VM). Esta implementação de *software* de um *firewall* ou

qualquer outra função de rede é chamada *Virtual Network Function* (VNF). O conceito/arquitetura de mover o software de um dispositivo de hardware dedicado para uma máquina virtual é chamado de Virtualização de Funções de Rede (NFV).

Figura 17 – Diagrama de Rede NFV



A Proposta

O primeiro passo para virtualizar as funções de rede proposta nesta dissertação, foi providenciar uma arquitetura para a nuvem, que combina o seu gerenciamento e a sua orquestração para oferecer novos serviços virtuais distribuídos pela rede. Para isso foi utilizada a infraestrutura do ambiente de experimentação do ELAN descrita no Capítulo 4, o *OpenStack*.

No modelo proposto por essa dissertação, o foco do NFV foi garantir que as novas VNFs desempenhem tão bem quanto as funções de rede física que estão substituindo. Do ponto de vista técnico, este é um passo importante, uma vez que estas VNFs são as peças fundamentais para a adoção da NFV e da virtualização da rede.

Para o fim pretendido, as questões operacionais relacionadas com NFV são simples. As VNFs são alocados na infraestrutura de nuvem do ELAN. Neste nível mais básico, as VNFs não exigem uma orquestração complexa, e muitos dos sistemas implantados são gerenciados usando sistemas simples de gerenciamento de elementos, de gerenciamento em nuvem ou gerenciadores VNF.

Também nesse modelo proposto é possível implementar uma função NFV sem qualquer orquestração complexa, por causa da simplicidade e da natureza estática das VNFs. Nesse caso é basicamente a replicação da rede física em uma versão virtual. E por ser pequeno o número de elementos de rede, os gerenciadores de elementos e VNF são suficientes. Mas, à medida que aumentem o número de VNFs e a complexidade da rede, surge a necessidade de um orquestrador NFV dedicado.

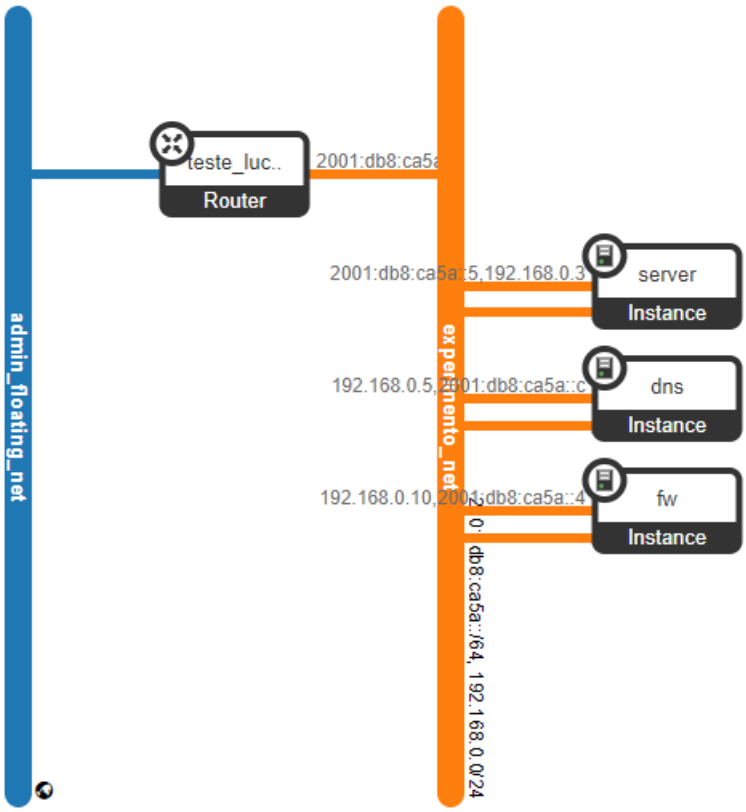
Levando-se em conta os estudos apresentados no Capítulo 3, foram escolhidas as seguintes funções de rede: *firewall*, NAT e DNS. A proposta dessa dissertação é virtualizar as funções de rede em um ambiente de nuvem apresentando uma estratégia baseada na utilização de programas de código aberto.

Tendo o orquestrador de nuvem *OpenStack* e nele instalado o *plugin networking-sfc*, utiliza-se o Painel (*dashboard*) *Horizon* através do qual são criadas, terminadas, reiniciadas e

gerenciadas as instâncias.

Através do Painel *Horizon* do *OpenStack* foram criadas três instâncias (VMs). Na criação de cada instância, são selecionados os recursos desejados, tais como: quantidade de vCPUs (*virtual Central Processing Unit* - CPU virtual alocada a VM, também conhecido como processador virtual), memória principal, disco e interfaces de rede (NIC - *Network Interface Card*). Cada instância, para receber as VNFs *firewall*, NAT e DNS, foi criada com 1 vCPU, 1 GB de memória, 50 GB de disco, 2 interfaces de rede e sistema operacional Ubuntu 16.04, conforme ilustrado na Figura 18.

Figura 18 – Topologia de Rede no OpenStack

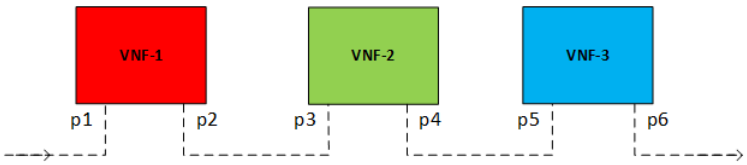


Instâncias criadas, passou-se então para a configuração do encadeamento de funções de serviço (*Service Function Chains*, ou SFC) definindo primeiramente a Cadeia de Portas (*Port Chain*), ou seja a definição da sequencia de funções de serviço e um conjunto de classificadores de fluxo, para especificar os fluxos de tráfego classificados para entrar na estrutura.

Como cada função de rede no modelo proposto tem um par de portas, a primeira porta no par de portas é a porta de entrada da função de serviço e a segunda porta é a porta de saída da função de serviço, ficando a sequência assim definida: [{'p1':'p2'}, {'p3':'p4'}, {'p5':'p6'}], conforme apresentado na Figura 19.

Os classificadores de fluxo são usados para selecionar o tráfego que pode acessar a cadeia. O tráfego que corresponda a qualquer classificador de fluxo será direcionado para a primeira

Figura 19 – Topologia do Encadeamento de Funções de Serviço



porta da cadeia. O classificador de fluxo será um módulo independente genérico e poderá ser usado por outros projetos como FW, QOS, etc.

5.1 Metodologia

No contexto do gerenciamento de redes de computadores, este trabalho está relacionado às fases de monitoramento e projeto de redes.

Na fase de Monitoramento, baseado nos dois tipos de fontes de dados, foram definidas as seguintes ações:

- Medição - A fim de obter informações do sistema existente são necessárias medições de tráfego; escolha de pontos de gargalo da rede; coleta, tratamento e análise de dados. Muitas vezes os dados de monitoramento não são acessíveis porque pode ser um serviço terceirizado.
- Configuração de serviços - Dados de monitoramento de serviços dos *Datacenters* podem não estar disponíveis, principalmente quando envolve dados dos servidores de seus inquilinos. O tráfego específico de aplicações pode não ser permitido, mas é possível estimar características do tráfego a partir dos dados de capacidade dos servidores e tipo de aplicação predominante.

Na fase de Projeto, baseado nas medições realizadas, foram identificadas as potenciais funções de rede a serem virtualizadas.

A técnica de avaliação utilizada nesse trabalho foi baseada na metodologia de Jain (1991), seguindo os seguintes passos:

1. Definição do sistema
   Empresa pública do Estado de Sergipe, a Emgetis (Empresa Sergipana de Tecnologia da Informação) é responsável por cuidar da Governança da área de Tecnologia da Informação e da Comunicação (TIC) e prestar serviços corporativos no âmbito do Governo Estadual, estando vinculada à Secretaria de Estado do Planejamento, Orçamento e Gestão (Seplag), conforme estabelecido em seu estatuto (Emgetis, 2017).

   A Emgetis possui em suas instalações um DC que hospeda diversos servidores e outros dispositivos tais como armazenamento de dados (*storages*) e ativos de rede (*switches* e

roteadores) e de telecomunicação. No DC da Emgetis estão armazenadas informações das diversas secretarias e órgãos da administração pública estadual, como dados da Secretaria de Estado da Segurança Pública (SSP), informações relacionadas com o trabalho da Secretaria de Estado da Saúde (SES), os sistemas Compras Net e da Folha de Pagamento da Secretaria de Estado da Administração (Sead), o correio eletrônico Expresso, o ambiente de produção do Detran, entre outros. A Emgetis ainda presta serviço de configuração até a borda das entidades públicas e como provedora, deve oferecer serviços em IPv6 nos próximos anos.

Para projetar a migração é necessário implantar funções de rede em um ambiente controlado pois qualquer alteração na topologia de uma rede em funcionamento, implica necessariamente em alterações na configuração de equipamentos de rede envolvidos no funcionamento dessa rede. Por outro lado, algumas alterações à topologia da rede podem ser automaticamente descobertas pelos outros equipamentos existentes na rede, podendo, em alguns casos, traduzir-se numa atualização automática da configuração de cada um dos equipamentos.

Como sugestão de configuração de cenário em nuvem foi necessário obter dados de monitoramento da rede em estudo para a caracterização do tráfego e dos serviços disponíveis na rede operacional. Dados sobre servidores (especificação - processador, memória e disco rígido, função que realiza, quantidade), roteadores e computadores (especificação, quantidade) foram coletados em todas as secretarias e órgãos da administração pública do Estado de Sergipe, inclusive a localização geográfica de cada uma dessas organizações. Além dessas informações, foram coletados também os dados de tráfego dos enlaces que interligam essas entidades à Emgetis, sendo que uma parte desses enlaces é mantida e administrada pela própria Emgetis e outra parte por uma operadora de telecomunicações.

Todos esses dados foram agrupados em um banco de dados (BD) de onde parte deles foi extraída para utilização nesse estudo de caso. Foram selecionados dados de tráfego mensal de rede das instituições públicas ligadas à rede da Emgetis. Esses dados extraídos, conforme demonstrado na Tabela 6, mostram o comportamento real da rede da Emgetis ao longo de um mês e foram utilizados como carga de trabalho nesse estudo de caso.

2. Modelo de topologia física e lógica

   No desenvolvimento do estudo de caso a geração das topologias físicas e lógicas foi realizada de forma semi-automática através de *scripts* de configuração.

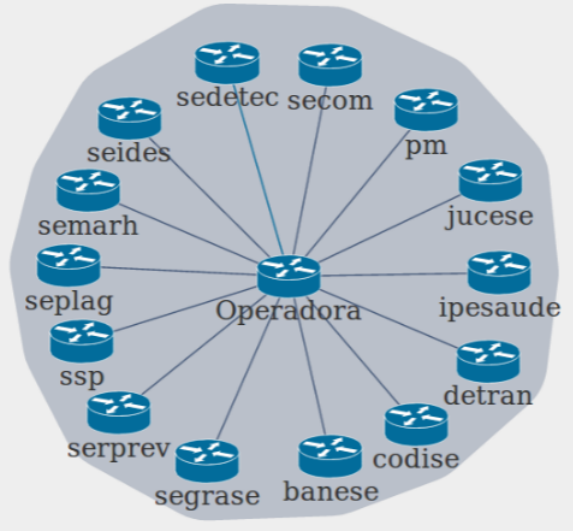
   Inicialmente foi adotado o conjunto de ferramentas AutoNetKit (KNIGHT et al., 2012) e NetKit (PIZZONIA; RIMONDINI, 2008). O AutoNetKit seria o responsável pela semi-automatização da topologia já que, com os dados extraídos do BD foi possível construir através de *scripts* baseados em XML, um modelo de topologia, conforme ilustrado na Figura 20.

Tabela 6 – Tráfego Mensal da Rede da Emgetis

Número da VM	**Entidade Pública**	**Tráfego Médio**	**Tráfego Máximo**
1	Banco do Estado de Sergipe (BANESE)	249,52 Kbps	51,4 Mbps
2	Companhia de Desenvolvimento Econômico de Sergipe (CODISE)	75,73 Kbps	8,76 Mbps
3	Departamento Estadual de Trânsito (DETRAN)	3,05 Mbps	92,37 Mbps
4	Instituto de Promoção e de Assistência à Saúde de Servidores do Estado de Sergipe (IPESAUDE)	314,84 Kbps	14,83 Mbps
5	Junta Comercial do Estado de Sergipe (JUCESE)	7,52 Mbps	282,45 Mbps
6	Polícia Militar do Estado de Sergipe (PM)	127,41 Kbps	22,1 Mbps
7	Secretaria de Estado da Comunicação Social (SECOM)	228,3 Kbps	54,79 Mbps
8	Secretaria de Estado do Desenvolvimento Econômico, Ciência e Tecnologia (SEDETEC)	60,02 Kbps	2 Mbps
9	Secretaria de Estado da Inclusão, Assistência e do Desenvolvimento Social (SEIDES)	102,76 Kbps	18,27 Mbps
10	Secretaria de Estado do Meio Ambiente e dos Recursos Hídricos (SEMARH)	30,53 Kbps	18,21 Mbps
11	Secretaria de Estado do Planejamento, Orçamento e Gestão (SEPLAG)	340,15 Kbps	22,49 Mbps
12	Secretaria de Estado da Segurança Pública (SSP)	219,4 Kbps	73,03 Mbps
13	Sergipe Previdência (SERPREV)	567,61 Kbps	98,24 Mbps
14	Serviços Gráficos de Sergipe (SEGRASE)	49,55 Kbps	7,53 Mbps

Fonte: Autoria própria.

Figura 20 – Topologia de rede criada no AutoNetKit



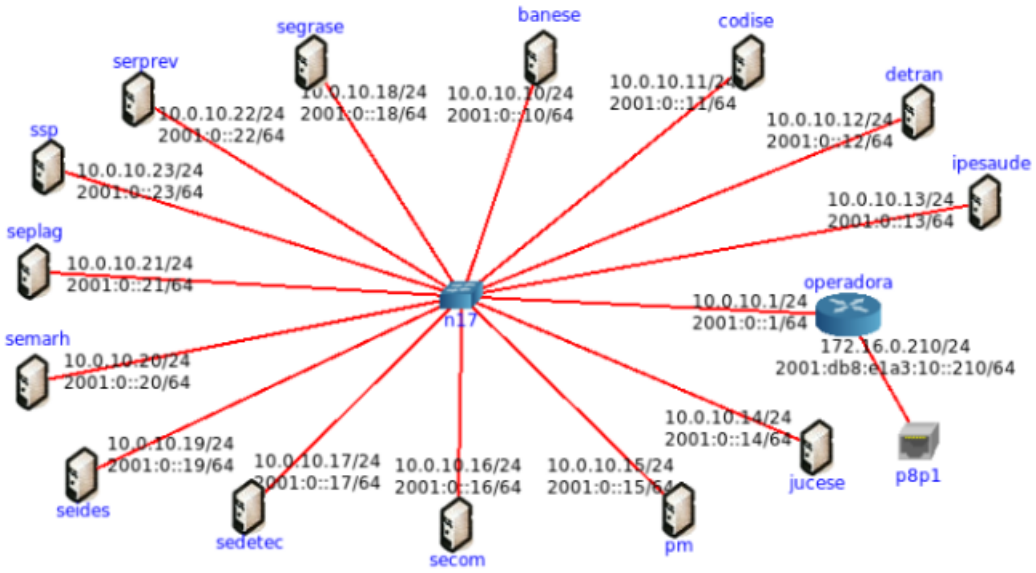
Fonte: Autoria própria

Com a topologia criada pelo AutoNetKit a criação do ambiente de emulação no NetKit, ou seja, a estrutura de máquinas virtuais dentro do emulador já estaria pronta, agilizando assim o andamento do experimento.

Entretanto, com o avançar dos testes foi identificada uma limitação no NetKit que exigiu a mudança de emulador. A limitação diz respeito a conexão das VMs dentro do emulador com *hosts* externos ao emulador utilizando puramente o protocolo IPv6, inviabilizando assim a sua utilização nesse experimento.

Por essa razão, foi adotado como emulador de rede a ferramenta CORE, já que esse programa possui suporte nativo a IPv6 necessário para os testes. A Figura 21 ilustra a topologia criada no emulador apresentando os endereços IPv4 e IPv6 adotados nos testes.

Figura 21 – Topologia de rede criada no CORE



Fonte: Autoria própria

3. Métricas

   Selecionar os critérios para se avaliar e comparar o desempenho dos serviços, medidas que representem a qualidade, eficácia ou eficiência do sistema e que possam ser obtidas nas variações avaliadas do sistema a fim de comparar os resultados. As métricas são em geral de três tipos: desempenho, confiabilidade e disponibilidade. Para cada serviço oferecido pelo sistema, um número variado de métricas de desempenho, confiabilidade e disponibilidade podem ser aplicados. Para a escolha das métricas, deve-se considerar as que possuem pequena variabilidade, evitar redundância (eliminar similares) e escolher um conjunto completo que avalie todas as possíveis saídas do sistema.

   Levando-se em conta a maturidade das métricas existentes conforme evidenciado em (BARAYUGA; YU, 2014) utilizaram em seu trabalho as métricas tempo de transferência,

vazão, taxa de transferência, *jitter* e perda de pacotes. Nos testes realizados nesse trabalho, foram utilizadas as métricas: vazão, dados transferidos, *Jitter* e perda de pacotes.

4. Fatores de estudo
   Os fatores e parâmetros utilizados nesse trabalho foram: os tipos e protocolos de rede; os tráfegos de rede de cada entidade pública da rede da Emgetis e as funções de rede virtualizadas. Os tipos de rede são Tradicional e NFV; os protocolos IPv4 (NAT44), IPv6 e NAT64; tráfegos médio e máximos são os valores apresentados na Tabela 6; concorrência é a quantidade de VMs que enviaram tráfego em cada teste e as funções de rede virtualizadas foram: *firewall*, NAT e DNS. Essas três funções foram selecionadas devido as necessidades de encadeamento de serviços. Nesse caso, um serviço de vídeo exige que seu tráfego passe por *cache*, *firewall* e NAT; O serviço HTTP exige que seu tráfego passe por *firewall* e NAT; O serviço criptografado exige que seu tráfego passe pelo NAT. O serviço de resolução de nomes na Internet exige que seu tráfego passe pelo *firewall*, NAT e DNS (DING et al., 2015).

5. Geração de carga
   Diz respeito a seleção de uma carga de trabalho que proporcione ao serviço realizar a tarefa, considerando que a carga funcione igualmente em todos as variações do sistema, não privilegiando o desempenho de um em relação ao outro.

   Neste trabalho foi utilizada a ferramenta *Iperf* versão 2.0.5.

   A geração de carga foi realizada através da ferramenta *Iperf* em modo UDP. Cada instância do *Iperf* gera um fluxo e foram utilizadas catorze instâncias de *Iperf* para geração de tráfego. Foram variados o número de instâncias do *Iperf* para alterar a quantidade de carga de tráfego.

6. Técnicas de Avaliação

   - Medição - é possível instanciar em ambiente de nuvem, VNFs para realizar medição de um determinado serviço em ambiente de experimentação para não atrapalhar a rede operacional.
   - Emulação - é possível realizar configurações reais em ambiente emulado controlado, com intuito de testar opções de parâmetros, antes de implantar na rede operacional.
   - Simulação - *software* de simulação de uso geral e específicos podem ser utilizados para tornar mais próximas da realidade.

   Como técnica de avaliação foi utilizada a medição.

6

Estudo de Caso

6.1 Migração de Rede Tradicional para Rede NFV utilizando os protocolos IPv4 e IPv6

Para testar o ambiente proposto, foi construído um estudo de caso que permitiu executar o experimento de migração de uma rede Tradicional para uma Rede NFV utilizando os protocolos IPv4 e IPv6. Nos dois cenários, o estudo de caso visou a realização de três testes do funcionamento dos protocolos: (*i*) em uma rede puramente IPv4, (*ii*) em uma rede puramente IPv6 e (*iii*) no terceiro o mecanismo de transição IPv4/IPv6 (NAT64).

Para representar as organizações apresentadas na Tabela 6 nos cenários de rede Tradicional e de rede NFV, foi utilizado um emulador de rede. Nesse emulador, instalado em um dos *hosts Linux* do ELAN, foram criadas quinze VMs, sendo uma para cada entidade pública da referida tabela, fazendo um total de catorze VMs e mais uma representando a operadora de telecomunicações.

Neste trabalho, foi utilizado como parâmetro de vazão do *Iperf* os dados de monitoramento cedidos pela Emgetis, aqui referenciados como dados de tráfego médio e de tráfego máximo, apresentados na Tabela 6.

Para cada tipo de rede (Tradicional e NFV), protocolo (IPv4, IPv6 e NAT64) e tráfego (médio e máximo), foram executados catorze testes, que correspondem a quantidade de entidades públicas apresentadas na Tabela 6.

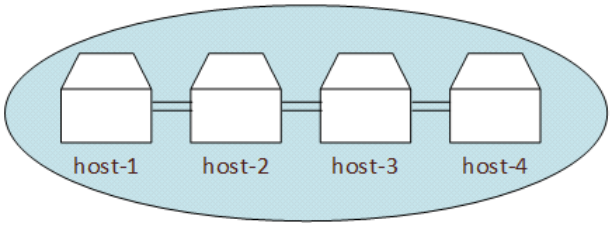
Como exemplo, pode-se citar que para rede NFV, com protocolo IPv4 e dados de tráfego médio, foi executado no *host* onde está instalado o emulador de rede *CORE*, uma VM que representa uma entidade pública da Tabela 6. Nessa VM, o *Iperf* em modo cliente foi utilizado para enviar tráfego na rede para o servidor *Iperf* (localizado em outro *host* da rede quando o cenário for de rede Tradicional ou em uma instância quando o cenário for de rede NFV),

utilizando no parâmetro de vazão do *Iperf* o valor correspondente ao tráfego médio dessa entidade na Tabela 6. Na segunda execução do testes, ainda para rede NFV, protocolo IPv4 e tráfego médio foram utilizadas duas VMs, cada uma representando uma entidade pública da Tabela 6, com o *Iperf* em modo cliente em cada uma delas enviando tráfego na rede para um único servidor *Iperf*, utilizando no parâmetro de vazão do *Iperf* o valor correspondente ao tráfego médio de cada uma dessas entidades extraídos da Tabela 6 e assim sucessivamente até a décima quarta execução onde as catorze VMs enviaram tráfego para o servidor *Iperf*. O mesmo procedimento se repetiu para o tráfego máximo, para os protocolos IPv4, IPv6 e NAT64 para os cenário de redes Tradicional e rede NFV.

6.1.1 Cenário I - Rede Tradicional

Essa Seção descreve os procedimentos realizados nos testes do estudo de caso de migração de IPv4 para IPv6 proposto nessa dissertação para o cenário de rede Tradicional. Para a execução dos testes com os diferentes protocolos (IPv4, IPv6 e NAT64), foi necessário utilizar os seguintes *hosts* do ELAN: *host-1*, *host-2*, *host-3* e *host-4*. Estes *hosts*, representam a infraestrutura de rede tradicional de um DC, ou seja, fora da nuvem, conforme apresentado na Figura 22. Em todos esses *hosts*, o sistema operacional utilizado foi o Ubuntu 16.04.

Figura 22 – Topologia Física da Rede Tradicional



Fonte: Autoria própria

No *host-1*, foi instalada a ferramenta CORE utilizada como emulador de rede onde foram criadas as quinze VMs (catorze VMs representando as organizações públicas listada na tabela que 6 e uma representando a operadora de telecomunicações) da rede da Emgetis.

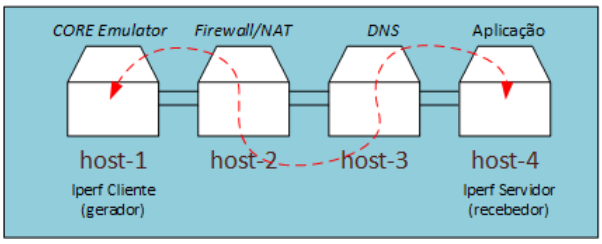
O *host-2* representa as funções de rede *firewall* e NAT da rede do DC, através da aplicação *Iptables* (COULSON, 2003) que foi instalado nesse *host*.

O *host-3* representa a função de rede DNS da rede do DC. Para isso, foi necessário instalar a aplicação BIND. Foi configurado como servidor DNS autoritativo convertendo respostas com registros do tipo A (IPv4) como também do tipo AAAA (IPv6).

O *host-4* representa um servidor de aplicação genérico utilizado como servidor do *Iperf*.

Nesse cenário, cada VM do emulador CORE, instalado no *host-1*, será um cliente do programa *Iperf*, que enviará tráfego para o servidor de *Iperf* localizado no *host-4*, passando pelos *host-2* e *host-3*, conforme ilustrado na Figura 23.

Figura 23 – Topologia Lógica da Rede Tradicional



Fonte: Autoria própria

O primeiro experimento deste cenário, chamado Teste 1A, refere-se à transmissão de pacotes em uma rede puramente IPv4, onde todos os *hosts* e as VMs do emulador foram configurados apenas com endereços IPv4. No segundo experimento, Teste 2A, a transmissão de pacotes se dá em uma rede puramente IPv6, ou seja, todos os *hosts* e as VMs do emulador estarão configurados apenas com endereços IPv6. Já o terceiro experimento, denominado Teste 3A, considera a ideia de usar o NAT64 como um mecanismo de transição adequado para substituir as redes NAT44 atuais. Nesse caso, todas as interfaces de rede das VMs do emulador, do *host-1* e uma das interfaces do *host-2*, estarão com endereços IPv6. Nos demais *hosts*, *host-3* e *host-4* além da segunda interface de rede do *host-2* estarão com endereço IPv4. O *host-2* é quem irá efetuar a tradução de IPv6 para IPv4 utilizando para isso o programa *Tayga* (LUTCHANSKY, 2011) juntamente com o *Iptables*.

Para efetuar as medições nos testes experimentais, foi necessário configurar o *software Iperf* com os seguintes parâmetros:

- No modo Servidor: *iperf -s -u*
  -s - indicando que é o servidor;
  -u - indicando que o teste seria com o protocolo UDP.

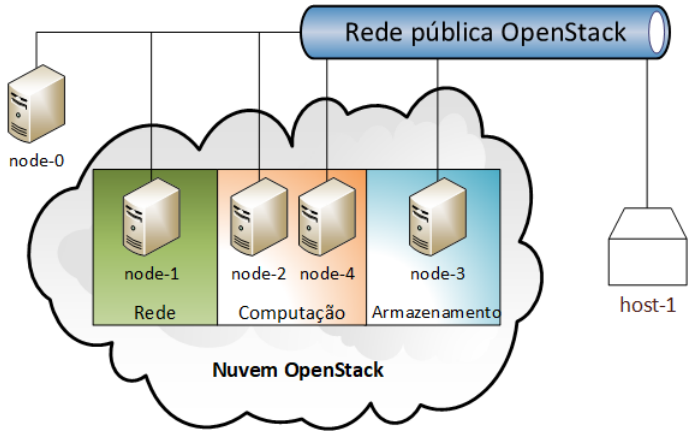
- No modo Cliente: *iperf -c <endereço_do_servidor> -u -f m -b <valor > -t 120*
  -c <endereço_do_servidor> - indicando que é o cliente do *Iperf*. Como endereço foi utilizado o nome cadastrado no servidor DNS;
  -u - indicando que o teste seria com o protocolo UDP;
  -f m - indicando que o formato de saída foi Mbits;
  -b <banda_utilizada> - indicando a banda a ser utilizada. Nesse caso foram adotados os valores de tráfego médio e de tráfego máximo mostrados na Tabela 6;
  -t <tempo> - no qual foi adotado o valor 120 segundos;
  -V - utilizado em conexões IPv6.

Destaque-se que os *hosts* utilizados nesse experimento estavam exclusivamente disponíveis para esse fim.

6.1.2 Cenário II - Rede NFV

Essa Seção descreve os procedimentos realizados nos testes do estudo de caso proposto nessa dissertação para o cenário de rede NFV. Para a execução dos testes com os diferentes protocolos (IPv4, IPv6 e NAT64) foi utilizado o mesmo *host-1* do cenário de rede Tradicional, onde já estavam configuradas as VMs dentro do emulador CORE, representando assim as organizações públicas da rede da Emgetis, conforme ilustrado na Figura 24.

Figura 24 – Topologia Física de Rede NFV no OpenStack



Fonte: autoria própria

No ambiente de nuvem, foram criadas três instâncias (*VNF-1*, *VNF-2* e *VNF-3*) representando, cada qual, uma função de rede virtualizada (VNF). O termo instância será utilizado nesse cenário para diferenciar a VM criada no ambiente de nuvem de uma VM criada em outros ambientes. Cada instância possui a seguinte configuração: 1 CPU virtual (vCPU), memória principal (RAM) de 2 GB e disco de 20 GB.

A instância *VNF-1* representa as funções de rede *firewall* e NAT da rede do DC, através da aplicação *Iptables* (COULSON, 2003) que foi instalado nessa instância.

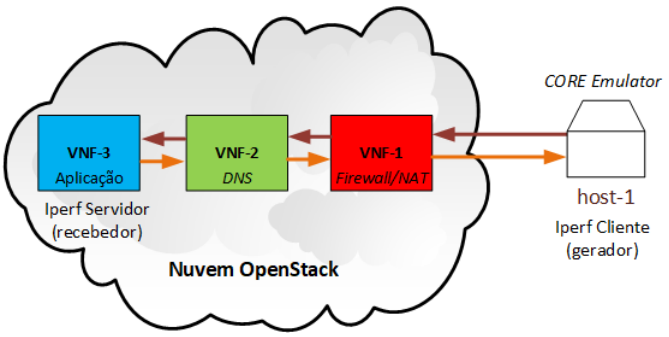
A *VNF-2* representa a função de rede DNS da rede do DC. Nessa instância, foi instalada a aplicação BIND (*Berkeley Internet Name Domain*), configurada como servidor DNS autoritativo convertendo respostas com registros do tipo A (IPv4) como também do tipo AAAA (IPv6).

A instância *VNF-3* representa um servidor de aplicação genérico utilizado como servidor do *Iperf*.

Nesse cenário de Rede NFV, cada VM do emulador CORE, instalado no *host-1*, será um cliente do programa *Iperf*, que enviará tráfego para o servidor *Iperf* localizado na instância *VNF-3*, passando pelas instâncias *VNF-1* e *VNF-2*, conforme ilustrado na Figura 25.

Os serviços de NFV são oferecidos por uma sequência de VNFs, formando uma SFC (*Service Function Chaining*). Para tanto, as VNFs foram ordenadas considerando a sequência de funcionamento dos serviços, ou seja *firewall*-DNS-aplicação. Os enlaces virtuais que ligam as

Figura 25 – Topologia Lógica de Rede NFV no OpenStack



Fonte: autoria própria

VNFs indicam o sentido que a SFC percorre atravessando cada VNF.

O primeiro experimento deste cenário, chamado Teste 1B, refere-se a transmissão de pacotes em uma rede puramente IPv4, onde o *host-1*, as VMs do emulador e as instâncias da nuvem estarão configurados apenas com endereços IPv4. No segundo experimento, Teste 2B, a transmissão de pacotes se dá em uma rede puramente IPv6, ou seja, o *host-1*, as VMs do emulador e as instâncias da nuvem estarão configurados apenas com endereços IPv6. Já no terceiro experimento, denominado Teste 3B, para NAT64 todas as interfaces de rede das VMs no emulador CORE, do *host-1* e da *VNF-1* estarão com endereços IPv6. As demais, *VNF-2* e *VNF-3* estarão com endereço IPv4. A *VNF-1* é quem irá efetuar a tradução de IPv6 para IPv4 utilizando para isso o programa Tayga (LUTCHANSKY, 2011) juntamente com o *Iptables*.

O *Iperf*, assim como no cenário de rede Tradicional, foi utilizado para efetuar as medições nos testes experimentais utilizando os mesmos parâmetros descritos na Seção 6.1.1.

Ao realizar os testes, foi identificado que a nuvem OpenStack utiliza o método de Pilha Dupla (*Dual Stack*) o que comprometeria a realização do Teste 3B já que o NAT é um serviço inerente à própria infraestrutura de rede do OpenStack.

Cabe ressaltar que no momento da execução dos testes o ambiente estava disponível exclusivamente para esse fim.

6.1.3 Resultados

Nessa Seção são apresentados os resultados obtidos nos experimentos realizados de tráfego médio e máximo nos dois cenários, de rede tradicional e de rede NFV com os protocolos IPv4, IPv6 e NAT64. As medições de (*i*) vazão (em Mbps), (*ii*) dados transferidos (em MBytes), (*iii*) média de *jitter* das conexões (em ms - milissegundo) e (*iv*) média de perdas de pacotes (em %), variam de acordo com a quantidade de VMs que enviam tráfego pelo *Iperf*, consideradas em cada experimento.

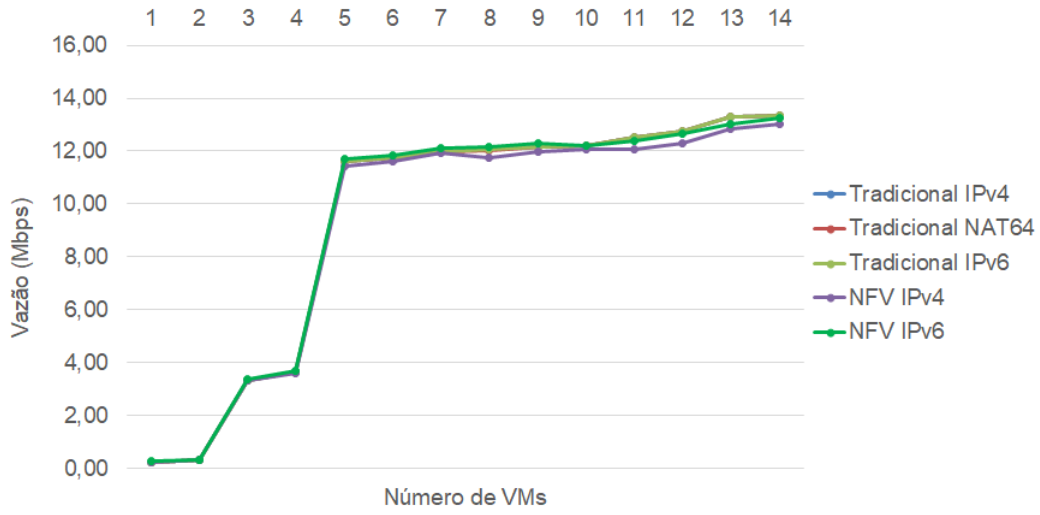
Os resultados são apresentados por ordem das medições *i*, *ii*, *iii* e *iv*, para os tráfegos

médio e máximo nos dois cenários desse estudo de caso, o de rede Tradicional e o de rede NFV. Cada ponto demonstrado na linha do gráfico representa a quantidade de VMs que geram carga através do *Iperf*. As descrições dos gráficos dos resultados podem ser vistas a seguir com suas respectivas considerações.

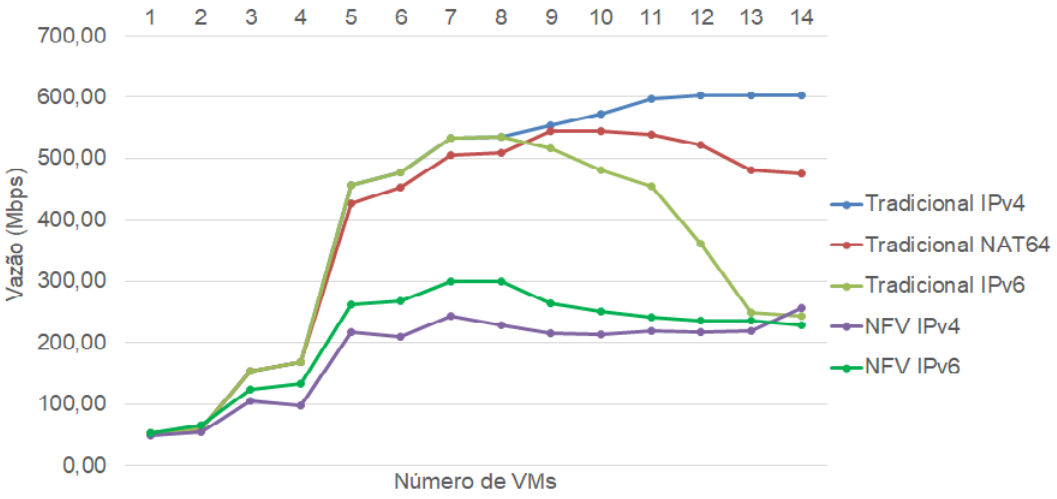
(*i*) **Vazão**

Nos gráficos presentes na Figura 26 têm-se os resultados dos testes de vazão. Percebe-se na Figura 26(a) que não há uma diferença expressiva entre os tipos de protocolos (IPv4 e IPv6) nos cenários de rede testados (Tradicional e NFV) com dados de tráfego médio.

Figura 26 – Vazão em tráfegos médio (a) e máximo (b) nas redes Tradicional e NFV



(a) Tráfego médio



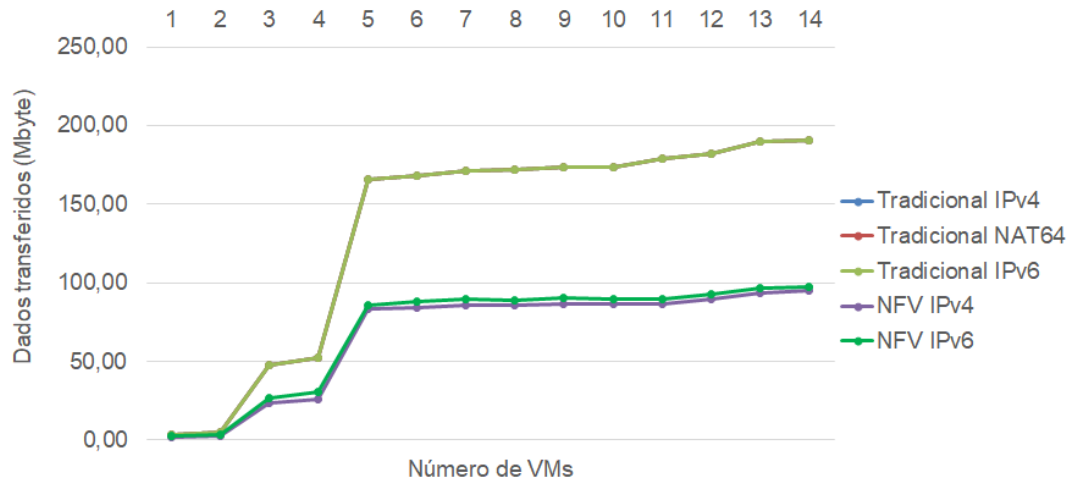
(b) Tráfego máximo

Na Figura 26(b) é perceptível o aumento da vazão quando utilizados os dados de tráfego máximo. Neste caso, a rede Tradicional com protocolo IPv4 variou de 52 Mbps a 603,08 Mbps, com protocolo IPv6 de 52 Mbps a 535,50 Mbps e com protocolo NAT64 de 52 Mbps a 545,01

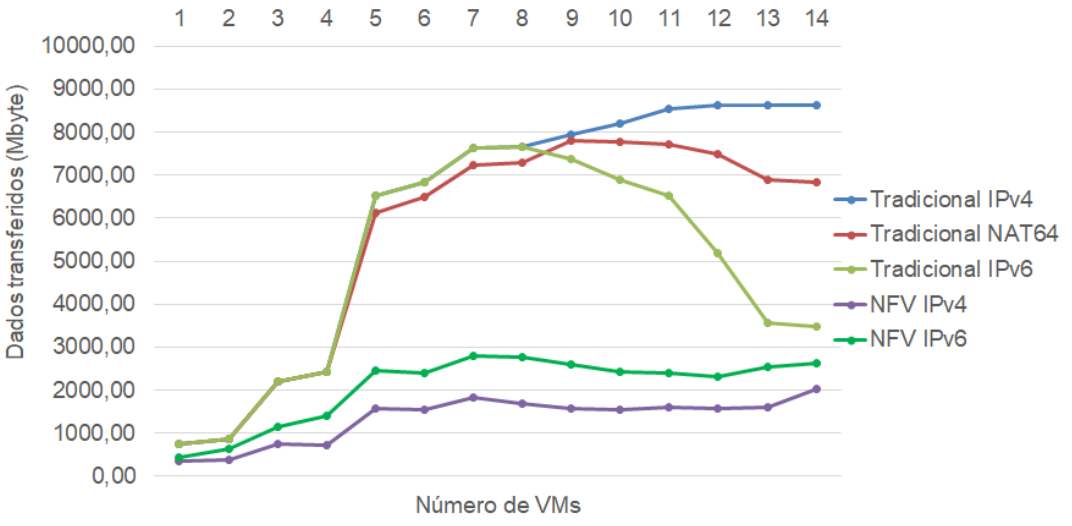
Mbps. Na rede NFV a variação foi de 49,60 Mbps a 256,89 Mbps com protocolo IPv4 e de 52,65 Mbps a 299,94 Mbps com protocolo IPv6, resultados inferiores ao da rede Tradicional.

Importante destacar que a partir de oito VMs gerando tráfego, o protocolo IPv6 na rede Tradicional tanto quanto na rede NFV, apresenta uma trajetória descendente no gráfico. Esses resultados apresentam-se condizentes quando relacionados à métrica perda de pacotes demostrada na Figura 29(b) já que nos testes realizados utilizou-se o DNS, que faz uso do protocolo UDP para tradução de endereços, e o *Iperf* configurado para o protocolo UDP. O protocolo UDP, originalmente não garante a entrega dos pacotes pois os dados são transmitidos apenas uma vez e os pacotes que cheguem corrompidos são simplesmente descartados, sem que o emissor sequer saiba do problema. Além disso, para Dutta e outros (2014), a transmissão de pacotes é sempre maior em IPv6 em comparação com o IPv4, devido ao tamanho do cabeçalho IPv6 (40 *bytes*) ser maior do que o cabeçalho IPv4 (20 *bytes*) .

Figura 27 – Dados transferidos em tráfegos médio (a) e máximo (b) nas redes Tradicional e NFV



(a) Tráfego médio



(b) Tráfego máximo

(*ii*) **Dados Transferidos**

A Figura 27 apresenta os resultados dos testes de dados transferidos. A quantidade de dados transferidos quando utilizados os valores de tráfego médio da Tabela 6, demonstram que no cenário de rede Tradicional todos são equivalentes, independentemente do protocolo utilizado, da mesma forma que no cenário de rede NFV porém, nesse caso os valores são inferiores aos do cenário de rede Tradicional, conforme ilustrado na Figura 27(a). Os dados transferidos no cenário de rede Tradicional ocorreram no intervalo de 3,56 *Mbytes* a 190,78 *Mbytes* para todos os protocolos testados enquanto que, no cenário de rede NFV para o protocolo IPv4 o intervalo foi de 1,78 *Mbytes* a 94,72 *Mbytes* e para o procolo IPv6 de 2,98 *Mbytes* a 97,99 *Mbytes*.

A Figura 27(b), ilustra o gráfico dos dados transferidos com tráfego máximo. No cenário de rede Tradicional, os dados transferidos utilizando o protocolo IPv4 ficaram na intervalo de 744 *Mbytes* a 8626,90 *Mbytes*. Para os protocolos IPv6 e NAT64 foram, respectivamente, 744 *Mbytes* a 7801,10 *Mbytes* e 744 *Mbytes* a 7658,60 *Mbytes*. No cenário de rede NFV, os dados transferidos utilizando os protocolos IPv4 e IPv6, respectivamente, foram de 355 *Mbytes* a 2044,44 *Mbytes* e 439 *Mbytes* a 2789,94 *Mbytes*, demonstrando que esse comportamento se assemelha aos resultados obtidos para vazão, ilustrados na Figura 26(b), ou seja, a perda de pacotes, demostrada na Figura 29(b), influenciou esse resultado da mesma forma, já que o gráfico apresenta uma trajetória descendente para o protocolo IPv6 a partir de oito VMs enviando tráfego.

(*iii*) ***Jitter***

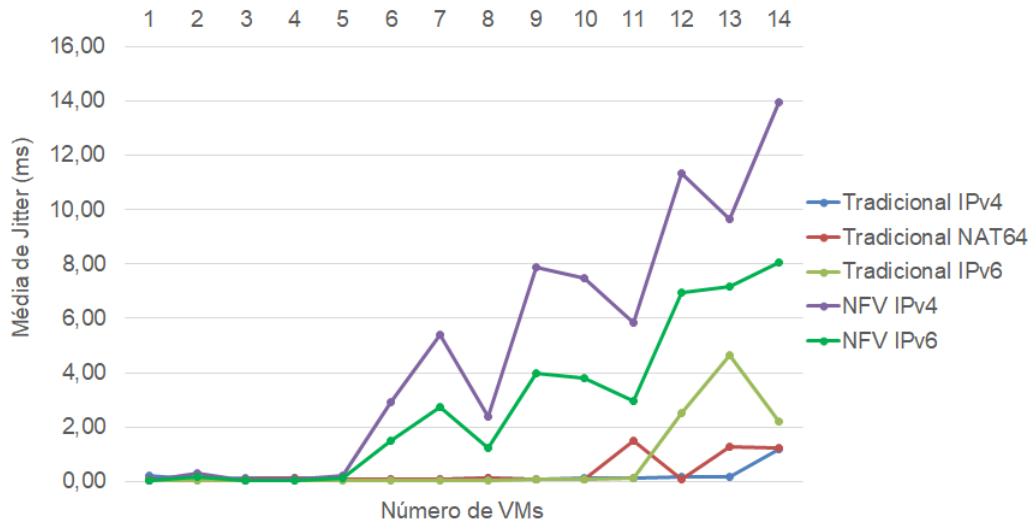
Os gráficos de média de *jitter* são apresentados na Figura 28. Na Figura 28(a) encontra-se o gráfico para tráfego médio onde é possível observar que para Rede NFV com tráfego médio e protocolo IPv4 aconteceu uma variação de 0,12 ms a 0,46 ms e de 0,07 ms a 0,38 ms para o protocolo IPv6. Para esses mesmos protocolos, mas na rede Tradicional, os valores foram 0,03 ms a 0,08 ms e 0,01 ms a 0,05 ms, respectivamente. Para NAT64 o valor foi de 0,02 ms a 0,08 ms. Percebe-se então que o desempenho do *jitter* na rede NFV foi menor do que na rede Tradicional.

Para tráfego máximo a variação do *jitter* na rede Tradicional com os protocolos IPv4, IPv6 e NAT64 foi de, respectivamente, 0,08 ms a 1,18 ms; 0,02 ms a 4,66 ms e 0,08 ms a 1,51 ms. Já na rede NFV a variação foi de 0,04 ms a 13,93 ms para o protocolo IPv4 e de 0,03 ms a 8,07 ms para o protocolo IPv6, conforme gráfico na apresentado na Figura 28(b).

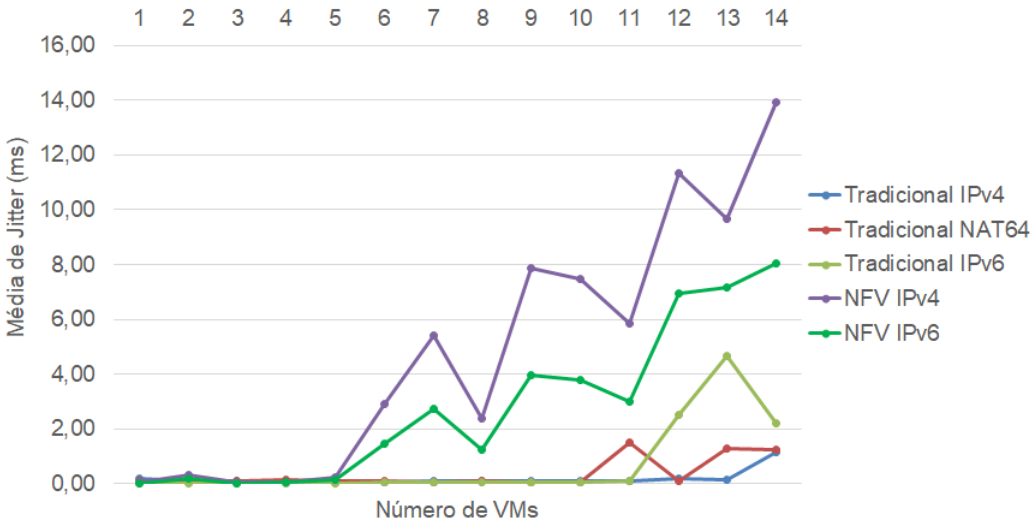
(*iv*) **Perda de Pacotes**

A Figura 29 apresenta os gráficos dos resultados dos testes de perda. Percebe-se na Figura 29(a) que não há uma diferença expressiva entre os tipos de protocolos (IPv4 e IPv6) nos cenários de rede testados (Tradicional e NFV) com dados de tráfego médio. Em todos os testes realizados em rede Tradicional o resultado da perda foi zero. Mas, para rede NFV a perda teve início quando foram utilizadas cinco VMs gerando tráfego atingindo o ápice em 0,71% para IPv4 e 0,71% para IPv6, quando então as catorze VMs estavam gerando tráfego.

Figura 28 – *Jitter* em tráfegos médio (a) e máximo (b) nas redes Tradicional e NFV



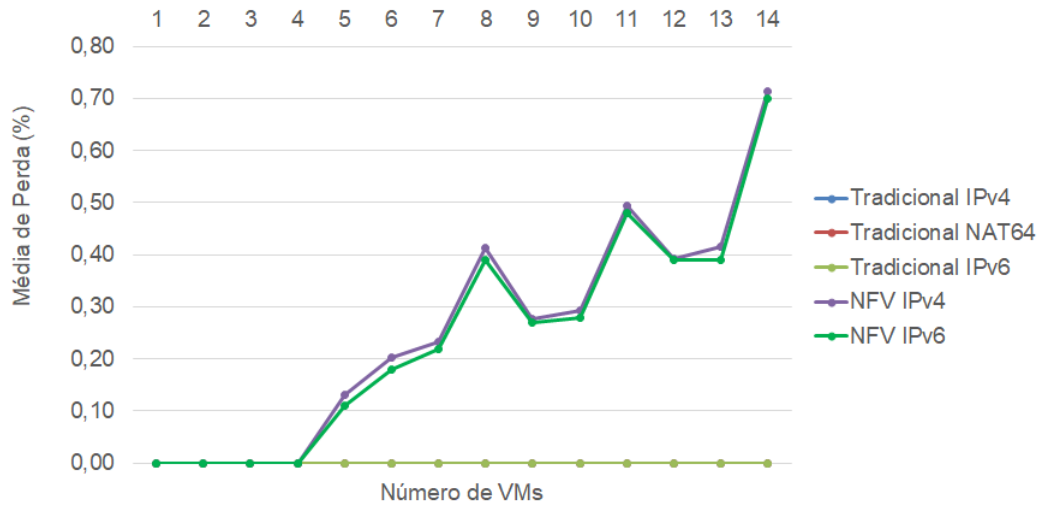
(a) Tráfego médio



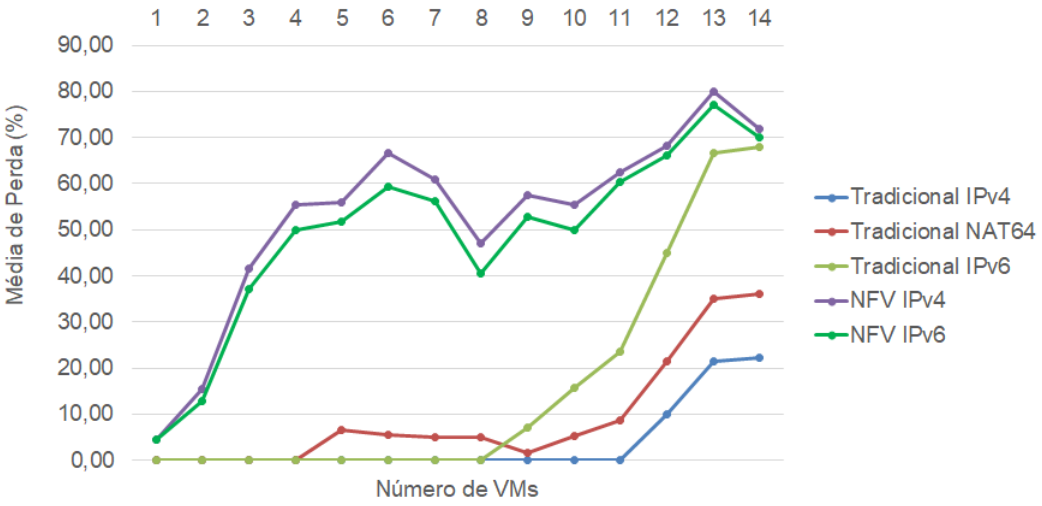
(b) Tráfego máximo

Para tráfego máximo, demonstrado no gráfico da Figura 29(b) é possível observar que a média de perda na rede Tradicional de todos os protocolos testados apresenta-se zerada até quatro VMs em execução gerando tráfego através do *Iperf*, permanecendo assim zerado até a posição seis VMs para o protocolo IPv6 e a posição onze VMs para o protocolo IPv4. Neste último, a média de perda variou de 9,98% a 22,14% e em IPv6 a variação foi de 0,01% a 67,86%. O protocolo NAT64 na rede tradicional gerou perdas quando cinco VMs geravam tráfego, e sua variação foi de 1,64% a 36%. Já para rede NFV já houve perda ainda com a primeira VM executa, tanto para o protocolo IPv4 quanto para o protocolo IPv6, sendo que a média de perda do IPv4 ligeiramente superior a do IPv6, pois a variação foi de 4,60% a 80,08% em IPv4 e de 4,6% a 77% em IPv6.

Figura 29 – Perda de pacotes em tráfegos médio (a) e máximo (b) nas redes Tradicional e NFV



(a) Tráfego médio



(b) Tráfego máximo

6.1.4 Considerações Finais

As simulações, com os cenários propostos, possibilitaram comparar o comportamento de dois tipos de abordagem de rede com a variação dos protocolos, gerando resultados semelhantes ao de um ambiente real.

As medições de vazão, dados transferidos, *jitter* e perda de pacotes refletiram o comportamento do modelo conforme a versão do protocolo.

Com a descrição do ambiente de experimentação, dos cenários do estudo de caso proposto e da análise dos resultados foi possível chegar a conclusão de que as diferenças observadas entre as abordagens de rede demonstram que não há impactos negativos em se fazer uso de funções de rede virtualizadas, ao contrário disso, possibilita que se tenha os benefícios da nuvem e da virtualização, sem comprometer a qualidade dos serviços disponibilizados.

7

Conclusão

O principal resultado deste trabalho é que uma plataforma de computação em nuvem de código aberto, como o *OpenStack*, pode ser efetivamente adotada para implantar o NFV em redes de *data center* em substituição às caras infraestruturas proprietárias ou até mesmo ambientes legados de centrais de telecomunicações. No entanto, esta solução apresenta algumas limitações para o desempenho da rede que não estão simplesmente relacionadas com a capacidade máxima de hospedagem do *hardware*, mas também com a arquitetura de rede virtual implementada pelo *OpenStack*.

Ao longo deste trabalho, foi possível, estudar as tecnologias envolvidas na Virtualização de Funções de Rede e Computação em Nuvem, para através desse estudo, fazer um levantamento dos principais direcionadores de requisitos, incluindo vantagens, desvantagens e atributos de qualidades.

Embora os benefícios do NFV, como a redução de custos e maior agilidade sejam bem compreendidos, ainda existem dúvidas sobre se uma implementação de *software* pode corresponder ao alto desempenho que os *appliances* de *hardware* oferecem.

Devido a isso, por meio de experimentos, procurou-se evidenciar como uma abordagem de NFV pode contribuir para que organizações públicas possam alavancar a infraestrutura de seus *Data Centers* permitindo identificar as vantagens preconizadas pelo ETSI para a tecnologia de NFV, mesmo que de forma experimental: versatilidade, alocação dinâmica de recursos e as consequentes reduções de custo, aliadas as vantagens que a *Cloud Computing* traz: rápida elasticidade, disponibilidade e economia.

Em comparação com o NFV, as caixas de *hardware* exigem manutenção parcial, atualização de *software* subjacente e instalações manuais. Um benefício aparente dos VNFs está em seu tempo para a implantação. O tempo para inicializar uma máquina virtual e executar algumas linhas de código para configurar um VNF não é comparável com o pedido de uma caixa semelhante a vendedores. Essa diferença se torna significativa ao lidar com um conjunto

conectado de VNFs. A conexão de várias caixas usando fios e *switches* é comparativamente mais demorada do que o código de execução para conectar VNFs na forma desejada.

Em termos de resultados a infraestrutura de NFV como *IaaS* no cenário proposto, mostrou-se eficaz e uma alternativa à adoção de *appliances* no tratamento do tráfego, além de permitir a expansão dessa infraestrutura e a implementação de novas funções de rede para suprir outras demandas. Os resultados apurados foram satisfatórios, corroborando a proposta dessa dissertação, mas que incitam a necessidade de mais ajustes no ambiente.

Apesar dos resultados apresentarem uma certa desvantagem da abordagem NFV em relação a abordagem Tradicional de rede, há de se reforçar as vantagens do NFV. Os resultados obtidos pelas medições de vazão, dados transferidos, *jitter* e perda de pacotes refletiram o comportamento do modelo conforme a versão do protocolo, mostrando que a partir de dados de tráfego médio e máximos, coletados de um ambiente real de uma instituição pública que possui um *Data Center*, pudessem ser utilizados nos experimentos como carga de trabalho. Em um dos testes foi gerada carga na rede experimental com dados coletados de tráfego médio que produziu valores de vazão e perda equiparados nas duas abordagens. Em contrapartida, foi possível identificar também que quando foram utilizados os dados de tráfego máximo nos cenários montados no estudo de caso a abordagem de Rede Tradicional se comportou melhor.

Desta forma, na situação proposta, indica-se que de fato virtualizar as funções de rede em ambiente de nuvem pode proporcionar às instituições públicas um ganho de escalabilidade e mobilidade na utilização de suas infraestruturas de *Data Center*, fazendo melhor uso dos recursos da nuvem. Seu aproveitamento se torna mais evidente em cenários onde houver uma sobrecarga de processamento e quantidade de clientes, haja vista, ao instanciar uma função de rede para a nuvem pode-se mais facilmente alterar suas configurações, como por exemplo: quantidade de processadores, tamanho de memória ou até mesmo a replicação de cópias dessa instância. Em qualquer caso, tais limitações devem ser cuidadosamente levadas em consideração para qualquer atividade de engenharia na área de redes virtuais.

7.1 Limitações do estudo

Durante o desenvolvimento desse trabalho, as limitações de funcionamento em rede IPv6 do Netkit, emulador de redes escolhido inicialmente para a realização dos testes, como também a implementação do IPv6 no ambiente de nuvem do laboratório comprometeram o cronograma.

7.2 Desafios Superados

Por se tratar de uma arquitetura relativamente recente, NFV apresentou-se com um desafio para implementar. A construção de um ambiente de nuvem para avaliação assim como o de uma rede tradicional necessitavam de *hardwares* em uma quantidade que se mostrasse

possível executar os experimentos. Para nossa satisfação ainda em meados de janeiro de 2017 todos os equipamentos já estavam disponíveis no laboratório.

A tarefa de montar e personalizar a nuvem *OpenStack* também provocou a ampliação dos estudos e análise de algumas ferramentas para a instalação da nuvem, optando-se pelo *Fuel* devido suas simplicidade de instalação para os usuários.

Paralelamente à nuvem implantada buscou-se identificar um emulador de redes e montar um ambiente para testes de rede tradicional a fim de confrontar os resultados com uma rede NFV montada na nuvem. Como emulador de rede a escolha recaiu sobre o CORE, visto possuir suporte nativo a IPv6.

7.3 Trabalhos Futuros

Aspira-se em trabalhos futuros a realização de medições/simulações em outros cenários e em variados volumes de tráfego. Além disso, outras funções de rede podem ser implementadas suprindo novas demandas desse cenário, criando inclusive serviços de encadeamento mais complexos além de verificar a implementação de funções de rede virtualizadas em ambientes de alta disponibilidade (HA).

Agradecimentos

Agradecemos à FAPITEC – Fundação de Apoio à Pesquisa e Inovação Tecnológica do Estado de Sergipe pelos recursos de fomento a este trabalho e à Emgetis – Empresa Sergipana de Tecnologia da Informação pela parceria e disponibilização das informações técnicas.

Referências

AGYAPONG, P. K. et al. Design considerations for a 5g network architecture. *IEEE Communications Magazine*, v. 52, n. 11, p. 65–75, Nov 2014. ISSN 0163-6804. Citado na página 47.

AHMAD, I. et al. New concepts for traffic, resource and mobility management in software-defined mobile networks. In: *2016 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*. [S.l.: s.n.], 2016. p. 1–8. Citado na página 47.

AHRENHOLZ, J. et al. Core: A real-time network emulator. In: *MILCOM 2008 - 2008 IEEE Military Communications Conference*. [S.l.: s.n.], 2008. p. 1–7. ISSN 2155-7578. Citado 3 vezes nas páginas 36, 37 e 47.

AKHTAR, N.; MATTA, I.; WANG, Y. Managing nfv using sdn and control theory. In: *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. [S.l.: s.n.], 2016. p. 1113–1118. Citado na página 51.

ALBITZ, P.; LIU, C. *DNS et Bind*. [S.l.]: O'Reilly Media, Inc., 2002. Citado 2 vezes nas páginas 39 e 40.

APPNETA. *AppNeta Performance Manager*. 2014. <https://www.appneta.com/>. Citado na página 38.

AYMERICH, F. M.; FENU, G.; SURCIS, S. An approach to a cloud computing network. In: IEEE. *Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the*. [S.l.], 2008. p. 113–118. Nenhuma citação no texto.

BARAYUGA, V. J. D.; YU, W. E. S. Study of packet level udp performance of nat44, nat64 and ipv6 using iperf in the context of ipv6 migration. In: *2014 International Conference on IT Convergence and Security (ICITCS)*. [S.l.: s.n.], 2014. p. 1–6. Citado 2 vezes nas páginas 51 e 64.

BARAYUGA, V. J. D.; YU, W. E. S. Packet level tcp performance of nat44, nat64 and ipv6 using iperf in the context of ipv6 migration. In: *2015 5th International Conference on IT Convergence and Security (ICITCS)*. [S.l.: s.n.], 2015. p. 1–3. Citado na página 51.

BENET, C. H. et al. Openstackemu x2014; a cloud testbed combining network emulation with openstack and sdn. In: *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*. [S.l.: s.n.], 2017. p. 566–568. Citado 3 vezes nas páginas 29, 47 e 51.

BRANDI, L.; MALHEIRO, A.; BAPTISTA, G. D. S. Governança de tecnologia da informação: Investigação em administração pública municipal. In: . [S.l.: s.n.], 2017. Citado na página 29.

BRITO, S. H. B. *IPv6-O novo protocolo da Internet*. [S.l.]: Novatec Editora, 2013. Citado na página 30.

CALLEGATI, F.; CERRONI, W.; CONTOLI, C. Virtual networking performance in openstack platform for network function virtualization. *Journal of Electrical and Computer Engineering-JECE*, Hindawi Publishing Corp., New York, NY, United States, v. 2016, abr. 2016.

ISSN 2090-0147. Disponível em: <http://dx.doi.org/10.1155/2016/5249421>. Citado na página 51.

CANNISTRA, R. et al. Enabling autonomic provisioning in sdn cloud networks with nfv service chaining. In: *Optical Fiber Communications Conference and Exhibition (OFC), 2014*. [S.l.: s.n.], 2014. p. 1–3. Citado na página 47.

CARNEIRO, J. B. L. et al. Análise da tecnologia de virtualização de servidores em um data center como fator para obtenção de ganhos de produtividade. *Revista ESPACIOS| Vol. 37 (Nº 27) Año 2016*, 2016. Citado na página 16.

CARPENTER, B.; BRIM, S. *Middleboxes: Taxonomy and Issues*. [S.l.], 2002. <http://www.rfc-editor.org/rfc/rfc3234.txt>. Disponível em: <http://www.rfc-editor.org/rfc/rfc3234.txt>. Citado na página 23.

CASADO, M.; FOSTER, N.; GUHA, A. Abstractions for software-defined networks. *Commun. ACM*, ACM, New York, NY, USA, v. 57, n. 10, p. 86–95, set. 2014. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/2661061.2661063>. Citado na página 23.

CHERKAOUI, O.; ROSENBERG, C. Keynote 1: Wireless controlled and managed from the cloud. In: IEEE. *Ad Hoc Networking Workshop (MED-HOC-NET), 2013 12th Annual Mediterranean*. [S.l.], 2013. p. 1–2. Citado na página 47.

CHIOSI, M. et al. Network functions virtualisation: An introduction, benefits, enablers, challenges and call for action. In: *SDN and OpenFlow World Congress*. [S.l.: s.n.], 2012. p. 22–24. ISBN 0890-8044. Citado 3 vezes nas páginas 17, 24 e 25.

CHOU, L. D. et al. The novel sdn testbed with virtual network functions placement. In: *2016 International Conference on Software Networking (ICSN)*. [S.l.: s.n.], 2016. p. 1–5. Citado na página 51.

CHOWDHURY, N.; BOUTABA, R. Network virtualization: state of the art and research challenges. *Communications Magazine, IEEE*, v. 47, n. 7, p. 20–26, July 2009. ISSN 0163-6804. Nenhuma citação no texto.

CHOWDHURY, N. M. K.; BOUTABA, R. A survey of network virtualization. *Computer Networks*, v. 54, n. 5, p. 862 – 876, 2010. ISSN 1389-1286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128609003387>. Citado 3 vezes nas páginas 17, 18 e 21.

COMER, D. *Interligação de Redes com TCP/IP – Vol. 1: Princípios, Protocolos e Arquitetura*. Elsevier Brasil, 2015. ISBN 9788535278644. Disponível em: <https://books.google.com.br/books?id=F1_jBwAAQBAJ>. Citado na página 31.

COMER, D. *Redes de Computadores e Internet - 6.ed.:*. [s.n.], 2016. ISBN 9788582603734. Disponível em: <https://books.google.com.br/books?id=1nwdDAAAQBAJ>. Citado 3 vezes nas páginas 29, 31 e 32.

COSTA, D. *DNS - Um Guia para Administradores de Redes*. [S.l.]: BRASPORT, 2006. ISBN 9788574522920. Citado 2 vezes nas páginas 39 e 40.

COSTIN, C. *Administração Pública*. Elsevier Brasil, 2010. ISBN 9788535211191. Disponível em: <https://books.google.com.br/books?id=HGhlY-EltX8C>. Citado na página 28.

COULSON, D. *Network Security Iptables*. [S.l.]: Apr, 2003. Citado 2 vezes nas páginas 67 e 69.

COUTO, R. de S. et al. Gt-pid: Uma nuvem iaas universitária geograficamente distribuída. 2015. Citado na página 35.

DANIELS, J. Server virtualization architecture and implementation. *Crossroads*, ACM, New York, NY, USA, v. 16, n. 1, p. 8–12, set. 2009. ISSN 1528-4972. Disponível em: <http://doi.acm.org/10.1145/1618588.1618592>. Citado na página 16.

DECUSATIS, C.; MUELLER, P. Virtual firewall performance as a waypoint on a software defined overlay network. In: *High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC,CSS,ICESS), 2014 IEEE Intl Conf on*. [S.l.: s.n.], 2014. p. 819–822. Citado na página 47.

DEERING, S.; HINDEN, R. *RFC 2460: Internet Protocol*. [S.l.]: Version, 1998. Citado na página 31.

DING, W. et al. Openscaas: an open service chain as a service platform toward the integration of sdn and nfv. *IEEE Network*, v. 29, n. 3, p. 30–35, May 2015. ISSN 0890-8044. Citado na página 65.

DINIZ, P. H.; JUNIOR, N. A. Ferramenta iperf: geração e medição de tráfego tcp e udp. *NOTAS TÉCNICAS*, v. 4, n. 2, 2014. Citado na página 38.

DOUGLIS, F.; KRIEGER, O. Virtualization. *Internet Computing, IEEE*, v. 17, n. 2, p. 6–9, March 2013. ISSN 1089-7801. Citado na página 21.

DOVROLIS, C.; PRASAD, R. *Pathrate*. 2004. Citado na página 38.

DOWNEY, A. B. Clink: a tool for estimating internet link characteristics. *Wellesley College, Online at: http://rocky. wellesley. edu/downey/clink*, 1999. Citado na página 38.

DUTTA, N. et al. Article: Analysis of packet transmission overhead of ipv4 and ipv6 through simulation. *IJCA Proceedings on National Conference cum Workshop on Bioinformatics and Computational Biology*, NCWBCB, n. 2, p. 20–24, May 2014. Full text available. Nenhuma citação no texto.

Emgetis. *Estatuto Social da Emgetis*. 2017. <http://187.17.2.102/emgetis2/wp-content/uploads/2017/01/estatuto-2.pdf>. Acessado em 13-06-2017. Citado na página 61.

ETSI ISG NFV. Network Functions Virtualisation (NFV); Architectural Framework. v. 1, p. 1–21, 2013. Citado 3 vezes nas páginas 24, 25 e 26.

FARIAS, F. N. et al. Pesquisa experimental para a internet do futuro: Uma proposta utilizando virtualização e o frame-work openflow. *XXIX Simpósio de Redes de Computadores e Sistemas Distribuídos-SBRC*, 2011. Citado na página 18.

FEDERAL, S. Constituição da república federativa do brasil. *Brasília: Senado*, 1988. Citado na página 28.

FERRAREZI, G.; GROSSI, I. D.; MARCHI, K. Firewall iptables e exemplo de implementação no ambiente corporativo. 09 2017. Citado 2 vezes nas páginas 38 e 39.

FOROUZAN, B.; FEGAN, S. *Protocolo TCP/IP - 3.ed.:*. McGraw Hill Brasil, 2009. ISBN 9788563308689. Disponível em: <https://books.google.com.br/books?id=fNvIgp3kkyQC>. Citado 2 vezes nas páginas 30 e 32.

GARAY, J. et al. Service description in the nfv revolution: Trends, challenges and a way forward. *IEEE Communications Magazine*, v. 54, n. 3, p. 68–74, March 2016. ISSN 0163-6804. Citado na página 47.

GE, J. et al. Experimenting adaptive services in sea-cloud innovation environment. In: *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*. [S.l.: s.n.], 2014. p. 137–138. Citado na página 47.

GPRCom. *Grupo de Pesquisa em Redes e Computação Distribuída*. 2006. <http://gprcom.ufs.br/pagina/16209>. Acessado em 01-05-2017. Citado na página 53.

HAKIRI, A. et al. Software-defined networking: Challenges and research opportunities for future internet. *Computer Networks*, Elsevier, v. 75, p. 453–471, 2014. Citado na página 24.

HALPERN, J.; PIGNATARO, C. *Service Function Chaining (SFC) Architecture*. [S.l.], 2015. Citado na página 26.

HEIDEKER, A.; KAMIENSKI, C. Funções de Rede Virtualizadas em Plataforma de Computação em Nuvem para Cidades Inteligentes. *XIII Workshop em Clouds e Aplicações - WCGA*, v. 0, p. 43–56, 2015. Citado na página 47.

HEIDEKER, A.; KAMIENSKI, C. Gerenciamento flexível de infraestrutura de acesso público à internet com nfv. In: *34 Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos – SBRC 2016*. [S.l.: s.n.], 2016. Citado na página 47.

HERRERA, J. G.; BOTERO, J. F. Resource allocation in nfv: A comprehensive survey. *IEEE Transactions on Network and Service Management*, IEEE, v. 13, n. 3, p. 518–532, 2016. Citado na página 26.

POSTEL, J. (Ed.). *RFC 791 Internet Protocol - DARPA Inernet Programm, Protocol Specification*. [S.l.], 1981. Citado na página 31.

JAIN, R. K. The art of computer systems performance analysis: techniques for experimental design, measurement, simulation and modeling. Wiley, 1991. Nenhuma citação no texto.

KITCHENHAM, B. Procedures for performing systematic reviews. *Keele, UK, Keele University 33.2004 (2004): 1-26*, Citeseer, 2004. Citado na página 42.

KNIGHT, S. et al. Autonetkit: simplifying large scale, open-source network experimentation. In: ACM. *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*. [S.l.], 2012. p. 97–98. Citado 2 vezes nas páginas 37 e 62.

KREUTZ, D. et al. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, v. 103, n. 1, p. 14–76, Jan 2015. ISSN 0018-9219. Citado na página 23.

KSENTINI, A.; TALEB, T.; MESSAOUDI, F. A lisp-based implementation of follow me cloud. *IEEE Access*, v. 2, p. 1340–1347, 2014. ISSN 2169-3536. Citado na página 47.

LARROCHA, E. R. et al. Filling the gap of information security management inside itil x00ae;: Proposals for posgraduate students. In: *IEEE EDUCON 2010 Conference*. [S.l.: s.n.], 2010. p. 907–912. ISSN 2165-9559. Citado na página 29.

LAUREANO, M. A. P.; MAZIERO, C. A. Virtualização: Conceitos e aplicações em segurança. *Livro-Texto de Minicursos SBSeg*, p. 1–50, 2008. Citado na página 16.

LEE, S.; LEVANTI, K.; KIM, H. S. Network monitoring: Present and future. *Computer Networks*, v. 65, p. 84 – 98, 2014. ISSN 1389-1286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S138912861400111X>. Citado 2 vezes nas páginas 33 e 34.

LUIZELLI, M. et al. Reconnecting partitions on physical infrastructures: Towards an expansion strategy for efficient virtual network embedding. In: *Computer Networks and Distributed Systems (SBRC), 2014 Brazilian Symposium on*. [S.l.: s.n.], 2014. p. 335–343. Nenhuma citação no texto.

LUTCHANSKY, N. *Tayga-nat64 for linux. Tayga*. 2011. Citado 2 vezes nas páginas 68 e 70.

MA, W.; MEDINA, C.; PAN, D. Traffic-aware placement of nfv middleboxes. In: *2015 IEEE Global Communications Conference (GLOBECOM)*. [S.l.: s.n.], 2015. p. 1–6. Citado na página 51.

MANZALINI, A.; CRESPI, N. An edge operating system enabling anything-as -a-service. *IEEE Communications Magazine*, v. 54, n. 3, p. 62–67, March 2016. ISSN 0163-6804. Citado na página 47.

MATIAS, J. et al. Toward an sdn-enabled nfv architecture. *IEEE Communications Magazine*, v. 53, n. 4, p. 187–193, April 2015. ISSN 0163-6804. Citado na página 24.

MAURICIO, L. A. et al. Uma arquitetura de virtualização de funções de rede para proteção automática e eficiente contra ataques. 2017. Citado na página 18.

MEIRELLES, H.; FILHO, J. *Direito administrativo brasileiro*. Malheiros Editores, 2016. ISBN 9788539203192. Disponível em: <https://books.google.com.br/books?id=qCFutQAACAAJ>. Citado na página 28.

MELL, P.; GRANCE, T. The nist definition of cloud computing. Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology Gaithersburg, 2011. Citado 3 vezes nas páginas 17, 26 e 27.

MIRANTIS. *Mirantis OpenStack Planning Guide*. 2014. <https://www.mirantis.com/software/openstack/fuel/>. Acessado em 18-07-2017. Citado na página 36.

MOCKAPETRIS, P. *RFC 1035 Domain Names - Implementation and Specification*. [S.l.], 1987. Disponível em: <http://tools.ietf.org/html/rfc1035>. Citado na página 39.

MONTELEONE, G.; PAGLIERANI, P. Session border controller virtualization towards "service-defined"networks based on nfv and sdn. In: *Future Networks and Services (SDN4FNS), 2013 IEEE SDN for*. [S.l.: s.n.], 2013. p. 1–7. Citado na página 18.

MUñOZ, R. et al. The adrenaline testbed: An sdn/nfv packet/optical transport network and edge/core cloud platform for end-to-end 5g and iot services. In: *2017 European Conference on Networks and Communications (EuCNC)*. [S.l.: s.n.], 2017. p. 1–5. Citado na página 51.

NAIK, P.; SHAW, D. K.; VUTUKURU, M. Nfvperf: Online performance monitoring and bottleneck detection for nfv. In: *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. [S.l.: s.n.], 2016. p. 154–160. Citado na página 51.

NAUDTS, B. et al. How can a mobile service provider reduce costs with software-defined networking? *International Journal of Network Management*, Wiley Online Library, v. 26, n. 1, p. 56–72, 2016. Citado na página 47.

NAUDTS, B. et al. Deploying sdn and nfv at the speed of innovation: toward a new bond between standards development organizations, industry fora, and open-source software projects. *IEEE Communications Magazine*, v. 54, n. 3, p. 46–53, March 2016. ISSN 0163-6804. Citado na página 47.

NETO, U. *Dominando Linux Firewall Iptables*. [S.l.]: Ciência Moderna, 2004. ISBN 8573933208. Citado na página 38.

Netperf. *The Netperf Homepage*. 2017. <https://github.com/HewlettPackard/netperf>. Acessado em 03-06-2017. Citado na página 38.

NIC.br. *Núcleo de Informação e Coordenação do Ponto br*. 2017. <http://www.nic.br>. Acessado em 20-07-2017. Citado 2 vezes nas páginas 31 e 33.

NUNES, B. et al. A survey of software-defined networking: Past, present, and future of programmable networks. *Communications Surveys Tutorials, IEEE*, v. 16, n. 3, p. 1617–1634, Third 2014. ISSN 1553-877X. Citado na página 24.

Opendaylight. 2017. Disponível em: <https://www.opendaylight.org/>. Citado na página 47.

OpenNMS. *OpenNMS*. 2017. <https://www.opennms.org/en>. Acessado em 13-05-2017. Citado na página 36.

OPENSTACK. OpenStack : An Overview OpenStack : The Projects In Detail. *Image (Rochester, N.Y.)*, 2011. Citado 2 vezes nas páginas 35 e 47.

PALUDO, A. *Administração Pública*. Elsevier Editora, 2010. ISBN 9788535238754. Disponível em: <https://books.google.com.br/books?id=gKOoQghSLzYC>. Citado na página 28.

PETERSEN, K. et al. Systematic mapping studies in software engineering. In: *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*. Swinton, UK, UK: British Computer Society, 2008. (EASE'08), p. 68–77. Disponível em: <http://dl.acm.org/citation.cfm?id=2227115.2227123>. Nenhuma citação no texto.

PIZZONIA, M.; RIMONDINI, M. Netkit: easy emulation of complex networks on inexpensive hardware. In: ICST (INSTITUTE FOR COMPUTER SCIENCES, SOCIAL-INFORMATICS AND TELECOMMUNICATIONS ENGINEERING). *Proceedings of the 4th International Conference on Testbeds and research infrastructures for the development of networks & communities*. [S.l.], 2008. p. 7. Citado 3 vezes nas páginas 36, 37 e 62.

POPEK, G. J.; GOLDBERG, R. P. Formal requirements for virtualizable third generation architectures. *Commun. ACM*, ACM, New York, NY, USA, v. 17, n. 7, p. 412–421, jul. 1974. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/361011.361073>. Nenhuma citação no texto.

RAJAN, D. Common platform architecture for network function virtualization deployments. In: *2016 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*. [S.l.: s.n.], 2016. p. 73–78. Citado na página 47.

RIMONDINI, M. Emulation of computer networks with netkit. 2007. Citado na página 37.

SAHHAF, S. et al. Network service chaining with optimized network function embedding supporting service decompositions. *Computer Networks*, Elsevier, v. 93, p. 492–505, 2015. Citado na página 26.

SARI, R. F.; WIRYA, P. L. P. Performance analysis of session initiation protocol on emulation network using nist net. In: *The 9th International Conference on Advanced Communication Technology*. [S.l.: s.n.], 2007. v. 1, p. 506–510. ISSN 1738-9445. Citado na página 36.

SCIAMMARELLA, T. et al. Uma análise do tráfego de controle de uma nuvem iaas geodistribuıda. 2016. Citado na página 34.

SIMULATOR, G. G. N. Disponível em:< http://www.gns3.net/>. *Acesso em: junho*, 2012. Citado na página 36.

SKöLDSTRöM, P. et al. Towards unified programmability of cloud and carrier infrastructure. In: *2014 Third European Workshop on Software Defined Networks*. [S.l.: s.n.], 2014. p. 55–60. ISSN 2379-0350. Citado na página 47.

SOARES, J. et al. Toward a telco cloud environment for service functions. *IEEE Communications Magazine*, v. 53, n. 2, p. 98–106, Feb 2015. ISSN 0163-6804. Citado na página 47.

SOARES, J.; SARGENTO, S. Optimizing the embedding of virtualized cloud network infrastructures across multiple domains. In: *2015 IEEE International Conference on Communications (ICC)*. [S.l.: s.n.], 2015. p. 442–447. ISSN 1550-3607. Citado na página 47.

STANFORD, U. *Clean Slate: A interdisciplinary Research Program*. 2014. Disponível em: <http://cleanslate.stanford.edu/>. Citado na página 22.

STOLFO, S. J.; SALEM, M. B.; KEROMYTIS, A. D. Fog computing: Mitigating insider data theft attacks in the cloud. In: *2012 IEEE Symposium on Security and Privacy Workshops*. [S.l.: s.n.], 2012. p. 125–128. Citado na página 48.

TALEB, T.; KSENTINI, A.; KOBBANE, A. Lightweight mobile core networks for machine type communications. *IEEE Access*, v. 2, p. 1128–1137, 2014. ISSN 2169-3536. Citado na página 47.

TALEB, T.; KSENTINI, A.; SERICOLA, B. On service resilience in cloud-native 5g mobile systems. *IEEE Journal on Selected Areas in Communications*, v. 34, n. 3, p. 483–496, March 2016. ISSN 0733-8716. Citado na página 47.

TIRUMALA, A. et al. Iperf: The tcp/udp bandwidth measurement tool. *htt p://dast. nlanr. net/Projects*, 2005. Citado na página 38.

TRACER, P. *Packet Tracer*. 2017. <https://www.netacad.com/courses/packet-tracer-download/>. Citado na página 36.

TRAJANO, A. F. R.; FERNANDEZ, M. P. Two-phase load balancing of in-memory key-value storages through nfv and sdn. In: *2015 IEEE Symposium on Computers and Communication (ISCC)*. [S.l.: s.n.], 2015. p. 409–414. Citado na página 47.

VILALTA, R. et al. The sdn/nfv cloud computing platform and transport network of the adrenaline testbed. In: *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*. [S.l.: s.n.], 2015. p. 1–5. Citado na página 51.

VILALTA, R. et al. Multitenant transport networks with sdn/nfv. *Journal of Lightwave Technology*, v. 34, n. 6, p. 1509–1515, March 2016. ISSN 0733-8724. Citado 2 vezes nas páginas 47 e 51.

WEN, H.; TIWARY, P. K.; LE-NGOC, T. Wireless Virtualization. p. 5–11, 2013. Disponível em: <http://link.springer.com/10.1007/978-3-319-01291-9>. Nenhuma citação no texto.

WETTE, P.; KARL, H. Dct${ˆ2}$gen: A versatile TCP traffic generator for data centers. *CoRR*, abs/1409.2246, 2014. Disponível em: <http://arxiv.org/abs/1409.2246>. Citado na página 47.

ZAMBONI, A. B. et al. StArt uma ferramenta computacional de apoio à revisão sistemática. In: *Brazilian Conference on Software: Theory and Practice - Tools session*. UFBA: [s.n.], 2010. Citado na página 44.

Zenmap. *Zenmap - Official cross-platform Nmap Security Scanner GUI*. 2017. <https://nmap.org/zenmap/>. Acessado em 13-05-2017. Citado na página 36.

ZHANG, Q.; CHENG, L.; BOUTABA, R. Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, Springer, v. 1, n. 1, p. 7–18, 2010. Citado na página 16.

ZHIQUN, X. et al. Emerging of telco cloud. *China Communications*, v. 10, n. 6, p. 79–85, June 2013. ISSN 1673-5447. Citado na página 47.