Redes Definidas por Software e Funções de Rede Virtualizadas: Um Estudo com Mininet

Rayres dos Santos Dias [1], Flávio Pereira da Silva [1] e Daniel dos Anjos Costa [1]

[1] Instituto Federal da Bahia (IFBA) – Santo Amaro – BA – Brasil

rayressdias28@gmail.com, flavio.pereira@ifba.edu.br e daniel.anjos@ifba.edu.br

Abstract. *With the growing demand for more flexible networks, traditional solutions become limited and costly. Network Function Virtualization (NFV) and Software-Defined Networking (SDN) emerge as innovative alternatives. NFV executes virtualized network functions on common servers, while SDN centralizes control via software. This work explores these technologies using the Mininet emulator for simulations. Two topologies were tested with three distinct scenarios, focusing on connectivity and performance. The results showed low latency and high reliability. The study concludes that Mininet is effective for SDN experiments. The combination of SDN, NFV, and emulators allows for more efficient and economical network management.*

Resumo. *Com a crescente demanda por redes mais flexíveis, soluções tradicionais tornam-se limitadas e de altos custos. A Virtualização de Funções de Rede (NFV) e as Redes Definidas por Software (SDN) surgem como alternativas inovadoras. A NFV executa funções de rede virtualizadas em servidores comuns, enquanto a SDN centraliza o controle via software. Este trabalho explora essas tecnologias utilizando o emulador Mininet para simulações. Foram testadas duas topologias com três cenários distintos com foco em conectividade e desempenho. Os resultados mostraram baixa latência e confiabilidade. O estudo conclui que o Mininet é eficaz para experimentos com SDN. A combinação entre SDN, NFV e emuladores permite uma gestão de redes mais eficiente e econômica.*

1. Introdução

O rápido avanço das tecnologias da informação tem gerado uma demanda cada vez maior por redes mais adaptáveis, eficientes e escaláveis. As redes tradicionais, que dependem de hardware e configurações manuais, lidam com dificuldades operacionais, elevados custos e obstáculos à expansão e à adaptação frente a novas exigências, como as trazidas pela Internet das Coisas (IoT), computação em nuvem e redes 5G.

Diante destas dificuldades, surgem duas tecnologias inovadoras: a NFV (*Network Function Virtualization*) e a SDN (*Software-Defined Networking*). A NFV propõe a migração das funções de rede, antes executadas por dispositivos físicos, para ambientes virtualizados, tornando possível sua implementação em servidores comuns. O modelo SDN, por outro lado, introduz uma separação entre os planos de controle e de dados da rede, promovendo uma administração centralizada e programável por meio de protocolos como o *OpenFlow*. A combinação dessas tecnologias viabiliza um novo paradigma para as redes modernas, permitindo adaptações rápidas às demandas do ambiente e ao tráfego.

Para testar e validar essas tecnologias, o uso de plataformas de emulação se mostra essencial. O *Mininet* destaca-se como uma ferramenta eficiente e de baixo custo, permitindo a simulação das topologias complexas e a análise da latência, confiabilidade e desempenho, sem a necessidade do uso de equipamentos físicos dedicados. Com isso, o presente artigo tem como objetivo analisar o comportamento de redes baseadas em SDN e NFV por meio de cenários distintos simulados com a ajuda do *Mininet*, com o intuito de avaliar sua viabilidade em ambientes reais.

2. Fundamentação Teórica

Este capítulo apresenta os fundamentos teóricos que fundamentam e contextualizam o presente trabalho. Através da revisão e análise crítica da literatura especializada, busca-se estabelecer os pilares conceituais essenciais para a compreensão do tema, aprofundando os conhecimentos sobre os conceitos-chave e as abordagens que subsidiam a pesquisa e o desenvolvimento aqui propostos.

2.1 *Network Function Virtualization* (NFV)

A Virtualização de Funções de Rede (NFV) é uma abordagem que visa a migração das funções de rede, tradicionalmente implementadas em dispositivos dedicados, para uma infraestrutura baseada em *hardware* comum [Gama Junior, 2017]. A NFV surgiu como resposta à pouca flexibilidade e aos altos custos das infraestruturas tradicionais baseadas em *hardware*. Desta forma, em vez de depender de um dispositivo físico, as funções de redes são virtualizadas e executadas em servidores físicos comuns, que não foram inicialmente concebidos para executar um serviço específico [Mendes, 2019].

Ainda de acordo com esse contexto Teixeira et al., (2018) complementa que, quando um cliente faz a solicitação de uma nova função de rede, os provedores podem de imediato configurar e ativar VMs (Virtual Machine - Máquinas Virtuais) que realizam essas funções. Isso acaba tornando possível uma maior agilidade no processo de implementação, possibilitando que novas funções de redes sejam disponibilizadas em semanas, em vez de meses, que é o período necessário para a implantação de hardwares tradicionais.

A adoção da NFV proporciona diversos benefícios, como rapidez no processo de implementação de serviços, diminuição de custos operacionais, flexibilidade no gerenciamento de recursos e possibilidade de ajustar ou migrar funções de forma remota [Teixeira et al., 2018; Torbes, 2018]. Como exemplo, serviços como IPTV (*Internet Protocol Television*) e vídeo sob demanda se beneficiam da escalabilidade e disponibilidade oferecidas por essa tecnologia [Marchesan, 2018]. Além disso, a NFV é também utilizada em *data centers*, permitindo a composição de cadeias de serviços otimizadas, dando origem a ambientes de computação em nuvem altamente escaláveis e flexíveis [Souza, 2020].

2.1.1 Arquitetura da NFV

O modelo apresentado pelo ETSI (*European Telecommunications Standards Institute*) organiza a arquitetura da NFV em três componentes principais, conforme apresentado na Figura 01: 1) a Infraestrutura de Virtualização de Funções de Rede (*Network Function Virtualization Infrastructure*, NFVI), 2) as Funções de Rede Virtualizadas (VNFs) e 3) o Gerenciamento e Orquestração (NFV *Management and Orchestration*, NFV MANO) [Rosa et al., 2014; Torbes, 2018].

A NFVI é constituída por servidores de prateleira, dispositivos de armazenamento e equipamentos de rede, como *links* e *switches*, que são encarregados pelo encaminhamento dos pacotes. As funções realizadas pelos recursos físicos são atraídas pelos recursos virtuais, essa abstração só é viabilizada devido a uma camada de virtualização, baseada em um hypervisor, que permite que o software das VFNs utilize a infraestrutura virtualizada subjacente. Desta maneira, recursos de processamento e armazenamento conseguem ser representados por uma ou mais máquinas virtuais, ao passo que a rede virtual é constituída por *links* e *nodos* virtuais [Torbes, 2018].

VNFs é a implementação em software de funções de rede que podem rodar sobre a infraestrutura NFV [Gama Junior, 2017]. As VNFs executam funções comuns de dispositivos de rede, como servidores DHCP (*Dynamic Host Configuration Protocol*), roteadores e *gateways*. Essas funções podem ser centralizadas em uma máquina virtual (VM - *Virtual Machine*) ou distribuídas em múltiplas máquinas virtuais interconectadas logicamente [Torbes, 2018; Rosa et al., 2014].

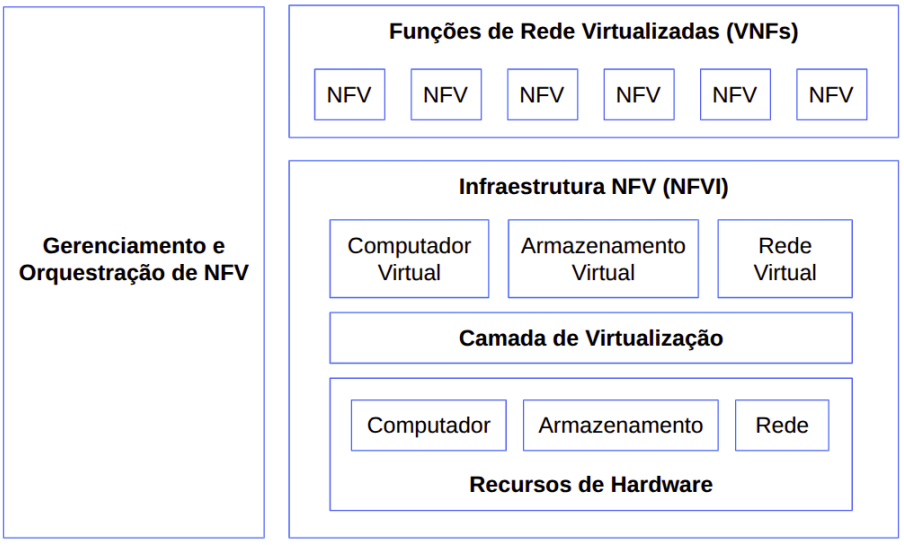

Figura 1 - Arquitetura Padrão do NFV. Fonte: Adaptado de Torbes, 2018.

Por fim, o NFV MANO é responsável pela gerência e orquestração do ciclo de vida das VNFs, coordenando os recursos físicos e virtuais da infraestrutura. Ele também provê interfaces com sistemas tradicionais de gestão, como OSS/BSS (*Operational Support Systems/Business Support Systems*), viabilizando a possível utilização de redes híbridas [Torbes, 2018].

2.2 Software Defined Networking (SDN)

A complexidade crescente das redes tradicionais, compostas por dispositivos variados e configurações manuais, torna a gestão e escalabilidade um desafio [Teixeira, 2018]. Visando superar esses obstáculos, surgiram as Redes Definidas por Software (SDN), que baseia-se na separação entre os planos de controle e o plano de dados, promovendo um gerenciamento centralizado, programável e eficiente [Mendes, 2019].

Na SDN, o plano de controle que é responsável pelas decisões de roteamento é centralizado em um controlador central, enquanto o plano de dados somente encaminha pacotes de acordo com as instruções [Mendes, 2019]. Essa arquitetura permite flexibilidade, automação e uma visão global da rede [Zamuner; Martins, 2023].

De forma diferente das redes tradicionais, em que cada nó gerencia seu próprio controle, a SDN centraliza a lógica em um único controlador. Essa tecnologia minimiza a complexidade operacional e simplifica a configuração de serviços, além de possibilitar respostas rápidas a mudanças na topologia ou no tráfego [Teixeira, 2018].

2.2.1 Arquitetura da SDN

A arquitetura lógica da SDN é estruturada por três principais camadas: Aplicação, Controle e Infraestrutura [Rezende, 2016]. Na camada de Aplicação estão as aplicações de rede, como

balanceadores de carga e *firewalls*. A camada de Controle centraliza a lógica da rede por meio de um controlador SDN, que executa decisões com base em políticas definidas. A camada de Infraestrutura executa essas decisões, encaminhando pacotes conforme as instruções recebidas [Rezende, 2016; Carneiro, 2021].

As interações entre essas camadas são realizadas por APIs (*Application Programming Interface*). A API da Borda Norte (*northbound*) liga o controlador às aplicações, enquanto a API da Borda Sul (*southbound*) conecta o controlador à camada de infraestrutura [Rezende, 2016]. A Figura 02 ilustra a visão lógica da arquitetura SND.

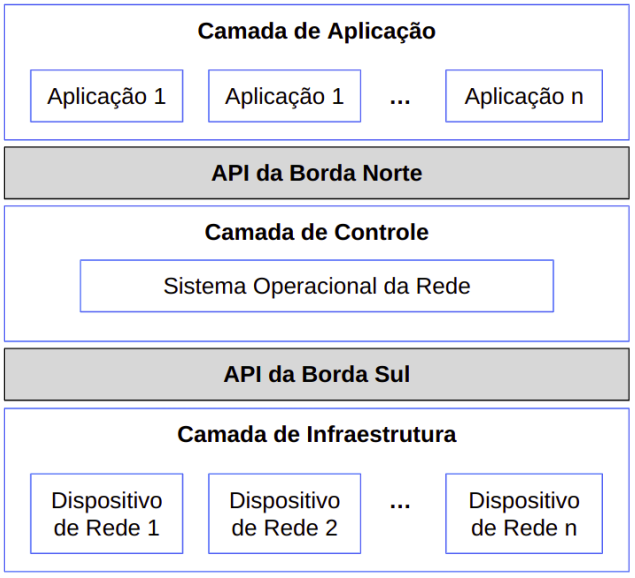

Figura 2 - Visão lógica da Arquitetura SDN. Fonte: Adaptado de Rezende, 2016.

Segundo Kurose (2021), o controlador SDN pode ser dividido em três componentes: (1) uma camada de comunicação com os dispositivos, (2) uma camada de gerenciamento de estado da rede e (3) uma interface com as aplicações de controle. Isso permite ao controlador configurar de maneira dinâmica a rede, detectar eventos e automatizar operações, promovendo uma gestão centralizada e adaptável [Vago, 2017].

2.2.2 Protocolo *OpenFlow*

O *OpenFlow* é o protocolo mais utilizado para comunicação entre controladores SDN e dispositivos da infraestrutura. Desenvolvido pela Universidade de Stanford, ele define uma interface padrão que possibilita que o controlador programe de forma direta as tabelas de fluxo dos *switches* [Chinelate, 2016].

Essas tabelas contêm regras que determinam como os pacotes devem ser tratados. Quando um pacote não corresponde a nenhuma regra existente, ele é encaminhado ao controlador, que então define a ação apropriada. Entre as ações possíveis estão o encaminhamento para uma porta específica, o envio ao controlador ou o descarte do pacote [Andreoli et al., 2017].

2.3 Emuladores de Rede e *Mininet*

A emulação de redes permite simular o comportamento de sistemas reais em ambientes controlados, oferecendo suporte à pesquisa, desenvolvimento e teste de protocolos e topologias. Ao contrário da simulação, a emulação executa softwares reais, permitindo maior

precisão nos resultados, além de controle individual de parâmetros como latência e largura de banda [Carissimi, 2008; Kropotoff, 2002].

O *Mininet* é um emulador eficiente desenvolvido para facilitar a realização de experimentos com Redes Definidas por Software (SDN). O *Mininet* permite a criação de redes completas com *hosts*, *switches OpenFlow*, *links* e controladores em um único computador, utilizando recursos como máquinas virtuais ou instalações em *Linux*. Com o uso do protocolo *OpenFlow*, o *Mininet* viabiliza testes realistas e em larga escala, sendo possível simular até 4096 hosts [Garcia, 2016].

3. Metodologia

Este trabalho adota uma abordagem mista, com pesquisa bibliográfica e experimental. A parte teórica fundamentou-se na revisão de obras acadêmicas sobre SDN e NFV. Já a pesquisa prática foi realizada através de simulações com o emulador *Mininet*, que permitiram observar o desempenho de redes SDN em topologias distintas.

Foram testados três cenários diferentes, variando em número de nós e *switches*, a fim da análise de conectividade, latência e largura de banda. A abordagem qualitativa possibilitou examinar os resultados observando o comportamento da rede em diferentes cenários, considerando também a conformidade com os parâmetros do Acordo de Nível de Serviço (SLA) estabelecidos.

Os cenários foram executados em ambiente virtual, por meio do emulador Mininet instalado em um computador com sistema operacional Linux. Não houve o uso de equipamentos físicos ou conexões externas reais. O Mininet possibilita a criação de redes completas com *switches*, *hosts* e controladores de maneira virtualizada, oferecendo um controle maior dos parâmetros como latência e largura de banda. Dessa maneira, os testes de latência, *throughput* e conectividade foram realizados em um ambiente controlado e replicável.

Essa metodologia permitiu uma avaliação prática da aplicação de SDN e NFV, contribuindo para compreender seus benefícios e limitações em ambientes de rede simulados.

4. Resultados e Discussões

Nesta seção, são apresentados e analisados os resultados das simulações realizadas no emulador *Mininet*, com o intuito de avaliar o desempenho de redes baseadas em SDN e NFV em cenários distintos, com base em parâmetros como conectividade, latência e largura de banda.

4.1 Cenário I

No primeiro cenário, foi emulada uma rede SDN constituída por dois *hosts* (*h1* e *h2*), um *switch openflow* (*s1*) e um controlado (*c0*) que utiliza o protocolo *OpenFlow*, utilizando o comando *"sudo mn"* no *Mininet*. A conectividade foi testada com os comandos *"ping"*, os testes demonstraram uma latência inicial mais alta no primeiro pacote enviado de *h1* para *h2* (Figura 3), comportamento esperado das redes SDN, onde o primeiro fluxo precisa da atuação do controlador para definir e instalar a regra de encaminhamento no *switch*. Depois dessa configuração, os pacotes seguintes foram encaminhados de forma direta, com latências muito menores.

Ao executar o comando "*h1 ping -c 10 h2*", os resultados mostraram latência média de 0.120 ms, sem perda de pacotes e valores permanecem bem abaixo do limite de 100 ms estabelecido pelo SLA (Figura 4). Esse comportamento comprova a eficiência da arquitetura SDN mesmo em redes pequenas e os resultados também reforçam a funcionalidade do *Mininet* como uma ferramenta prática e confiável para simulação de redes programáveis.

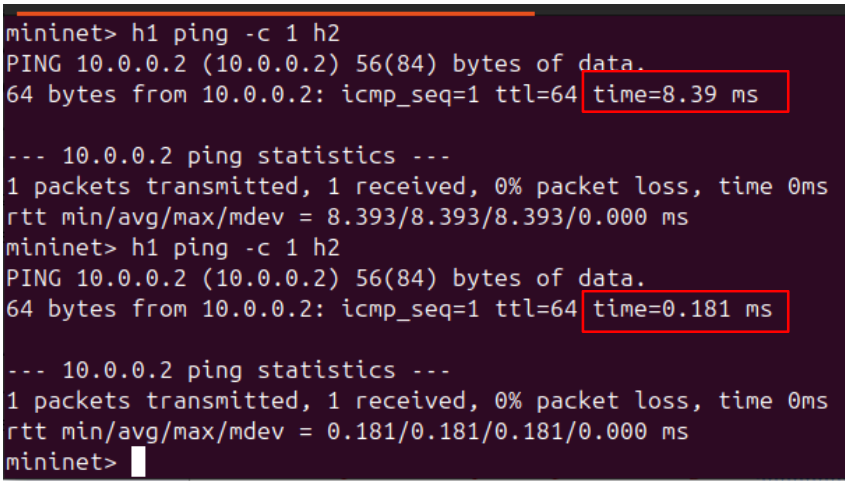

Figura 3 - Execução dos dois comandos ping entre os hosts h1 e h2.

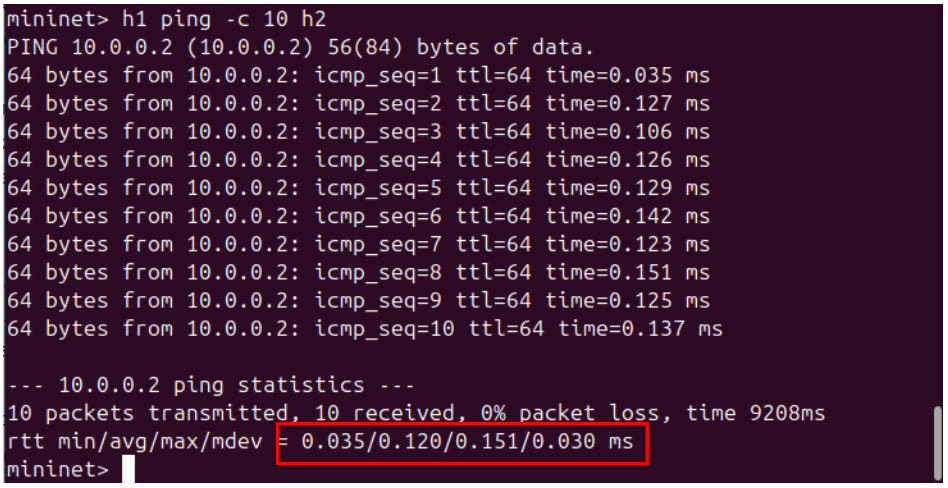

Figura 4 - Saída do terminal para o teste de 10 pings seguidos entre os hosts h1 e h2 no Cenário I.

4.2 Cenário II

No segundo cenário, foi emulada uma rede SDN mais complexa, composta por 16 *hosts*, 5 *switches* e um controlador, estruturados em uma topologia em árvore com profundidade 2 e fator de ramificação 4. A verificação da conectividade foi realizada com o comando *"pingall"*, totalizando 240 testes entre os hosts (Figura 5). Todos os pacotes foram entregues com sucesso, comprovando o funcionamento correto da topologia e a eficiência do gerenciamento centralizado promovido pelo controlador SDN.

Para análise de desempenho, foi realizado um teste de latência entre os hosts h1 e h16. Os resultados indicaram uma latência mínima de 0,058 ms e máxima de 0,311 ms sem perda de pacotes. Esses resultados estão bem abaixo do limite de 100 ms definido pelo SLA,

evidenciando que mesmo sendo uma topologia mais complexa, a rede apresentou desempenho e comunicação eficiente.

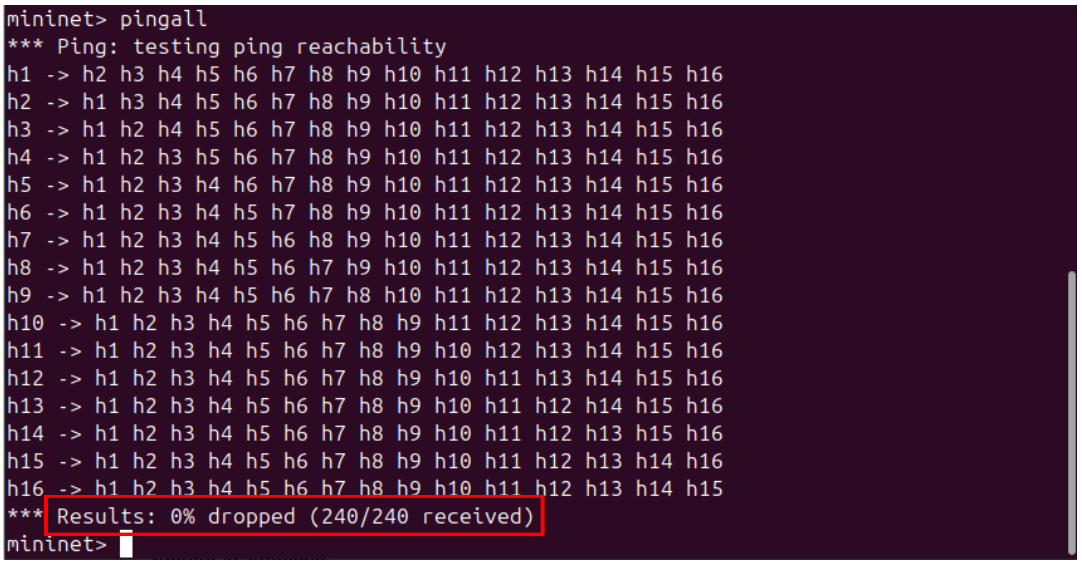

Figura 5 - Teste de comunicação entre todos os hosts da rede.

Adicionalmente, foi realizado um teste de largura de banda utilizando a ferramenta Iperf, utilizando o protocolo TCP (*Transmission Control Protocol*) entre h1 e h16. O *throughput* registrado foi de 24,0 Gbps em 10 segundos,o que totalizou 27,9 GB transferidos (Figura 6). De maneira geral, os testes neste cenário mostram que, mesmo com aumento da complexidade e número de dispositivos, a rede continua com alto desempenho, baixa latência e boas taxas de transferência.

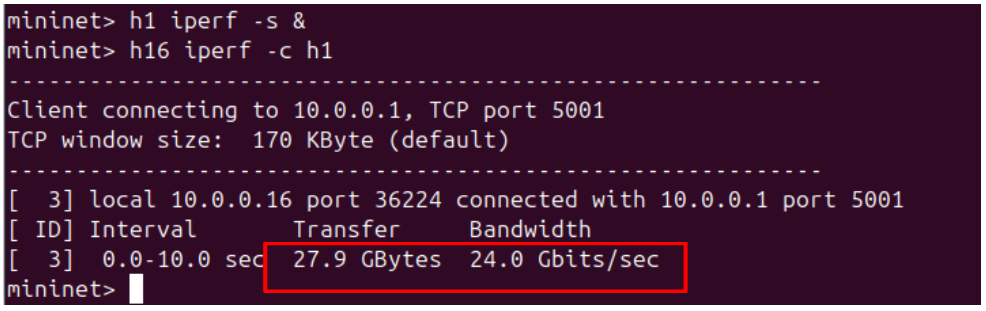

Figura 6 - Resultado do teste de largura de banda entre dois hosts.

4.3 Cenário III

No terceiro cenário, foi utilizada a mesma topologia do Cenário I para simular a comunicação em nível de aplicação, utilizando o protocolo HTTP, entre dois hosts. O host h1 foi configurado como servidor HTTP por meio do comando "python -m http.server 80", enquanto o host h2 realizou uma requisição utilizando "*h2 wget -O - h1*" a esse servidor. A resposta retornada pelo servidor indicou sucesso o status 200 OK, confirmou a comunicação entre os hosts e a transferência de um arquivo HTML simples, demonstrando que a rede emulada suporta aplicações baseadas em protocolos de camada superior (Figura 7).

O experimento valida que o ambiente SDN configurado no *Mininet* permite a comunicação entre os hosts de forma eficiente, com o controlador atuando na instalação das

regras de fluxo. Além de confirmar a conectividade, o teste evidencia que a arquitetura SDN permite o funcionamento de serviços em camadas superiores, sem precisar de configurações manuais nos *switches*, o que reforça sua aplicabilidade prática e a flexibilidade das redes programáveis.

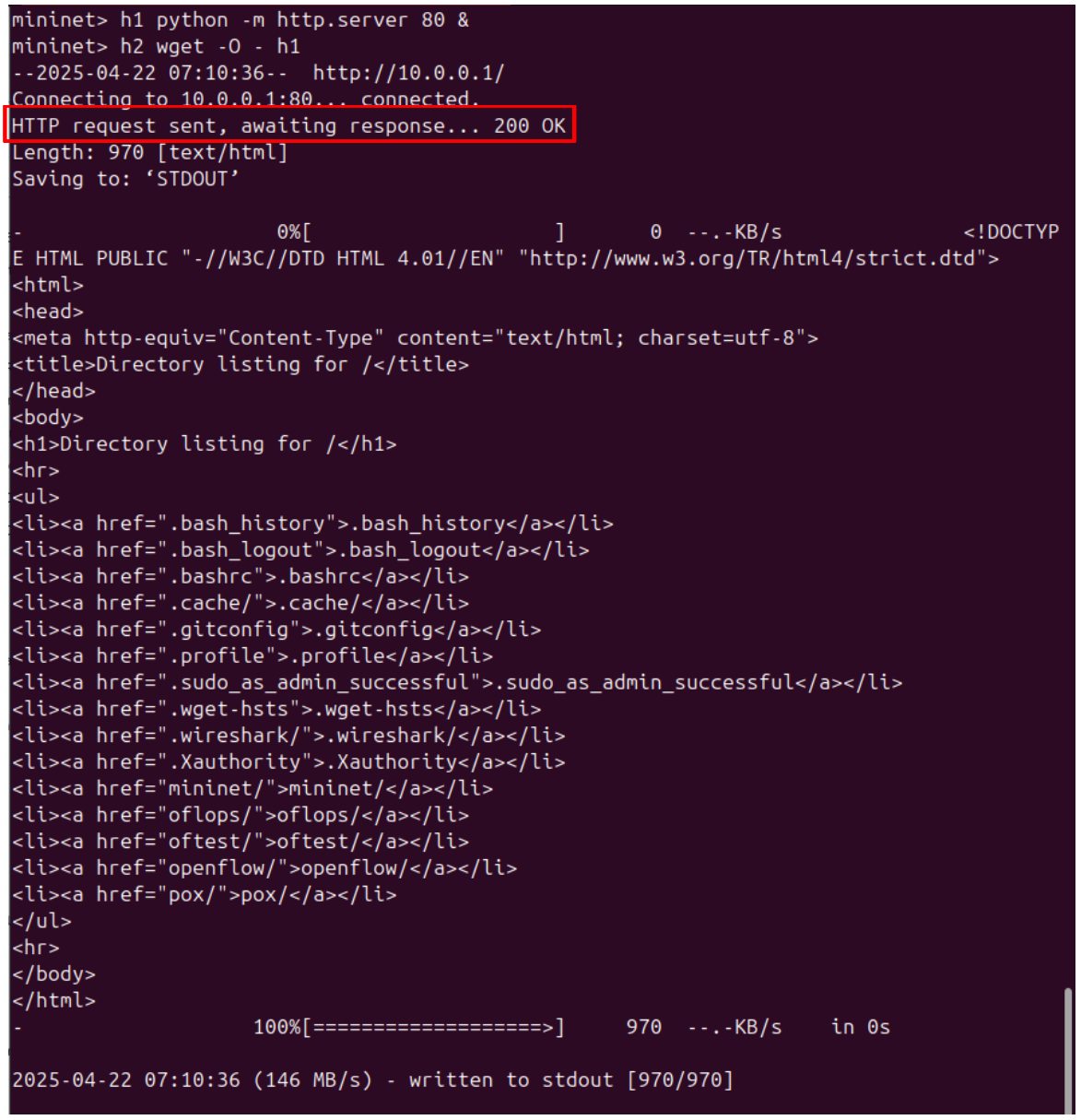

Figura 7 - Simulação de um servidor Web no Mininet.

5. Conclusão

O estudo apresentou as tecnologias de Virtualização de Funções de Rede (NFV) e Redes Definidas por Software (SDN), evidenciando como as duas tecnologias têm impactado o desenvolvimento, a operação e a escalabilidade das redes modernas. A análise destacou que, ao separar as funções de rede do hardware proprietário e centralizar o controle da rede por meio do software, a implementação dessas tecnologias possibilita maior flexibilidade, agilidade e redução dos custos operacionais.

Nesse contexto, destaca-se também a importância do Mininet como um recurso fundamental para simular redes SDN de maneira prática e de baixo custo. Com a possibilidade de criação de topologias virtuais compostas por elementos como *switches*, controladores e hosts, o Mininet se mostra essencial no teste de soluções baseadas em SDN. Ao permitir simulações sem necessidade de hardware dedicados, o uso do Mininet reduz os custos e a complexidade envolvidos na experimentação com hardware real, tornando mais rápida a inovação e possibilitando que pesquisadores e profissionais analisem o comportamento de redes definidas por software em ambientes controlados.

Os resultados das simulações realizadas demonstraram que as redes SDN, quando configuradas de maneira correta, asseguram estabilidade na comunicação, com latências adequadas, nenhum registro de perdas de pacotes e eficiência na transmissão de dados. A comparação entre os diferentes cenários evidenciou a escalabilidade e flexibilidade proporcionadas pelo uso de SDN e NFV, reforçando sua importância para a evolução das infraestruturas de rede modernas.

Dentre as principais contribuições deste estudo, destaca-se a utilidade do Mininet como uma plataforma viável para simular e testar arquiteturas de rede modernas. O estudo evidencia a importância de avançar no desenvolvimento de novas soluções voltadas à melhoria da orquestração de funções de rede e controle eficiente do tráfego.

Como trabalhos futuros, sugere-se investigar aplicações reais de SDN/NFV em operadoras e data centers, explorar soluções de segurança para proteger funções virtualizadas e integrar essas tecnologias com 5G, edge computing e inteligência artificial, visando redes ainda mais inteligentes, seguras e adaptáveis.

Referências

ANDREOLI, L.; RIGHI, R. da R.; AUBIN, M. R. (2017) “Analisando métodos e oportunidades em redes definidas por software (SDN) para otimizações de tráfego de dados.” Revista Brasileira de Computação Aplicada, v. 9, n. 4, p. 2–14. DOI: https://doi.org/10.5335/rbca.v9i4.6948.

CARISSIMI, A. (2008) “Virtualização: da teoria a soluções.” In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 2008, Rio de Janeiro. Livro Texto dos Minicursos. Rio de Janeiro: SBC. p. 174–199.

CARNEIRO, R. (2021) “Segurança em Redes Definidas por Software baseadas em OpenFlow.” 48 f. Dissertação (Mestrado em Segurança Informática) – Departamento de Ciência da Computação, Faculdade de Ciências.

CHINELATE, L. C. (2016) “Balanceamento de Carga Utilizando Planos de Dados OpenFlow Comerciais.” Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Juiz de Fora.

GAMA JUNIOR, Lúcio da Silva. “Virtualização de funções de rede em nuvem para instituições públicas.” 2017. 86 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Sergipe, São Cristóvão, 2017.

GARCIA, T. O. (2016) “Definição de novas regras para o IDS Snort em redes definidas por software.” 54 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade de Santa Cruz do Sul, Santa Cruz do Sul.

KUROSE, J. F. (2021) “Redes de computadores e a internet: uma abordagem top-down.” 8. ed. São Paulo: Pearson Prentice Hall.

KROPOTOFF, A. B. (2002) “Um emulador paramétrico de conexões fim-a-fim em redes IP.” 116 f. Dissertação (Mestrado) – Universidade Federal do Rio de Janeiro, Rio de Janeiro.

MARCHESAN, G. (2018) “Uma Análise Comparativa entre Paradigmas de Virtualização de Redes.” 114 p. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Santa Maria, Santa Maria.

MENDES, H. F. S. (2019) “Abordagem teórica da aplicação de virtualização de funções de rede na tecnologia de comunicação 5G.” 64 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Telecomunicações) – Universidade Federal Fluminense.

REZENDE, P. H. A. (2016) ”Extensões na arquitetura SDN para o provisionamento de QoS através do monitoramento e uso de múltiplos caminhos.” 130 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Uberlândia, Uberlândia. DOI: http://doi.org/10.14393/ufu.di.2016.59.

ROSA, R. V.; SIQUEIRA, M.; BAREA, E.; MARCONDES, C. A. C.; ROTHENBERG, C. R. E. (2014) “Network Function Virtualization: Perspectivas, Realidades e Desafios.” In: FRAGA, J. S.; SIQUEIRA, F.; MAZIERO, C. A. (Org.). Minicursos SBRC. Porto Alegre: SBC, 2014. p. 1–54.

SOUZA, R. R. (2020) “Um framework inteligente para escalonamento de VNFs em data center.” Tese (Doutorado em Ciência da Computação) – Universidade Federal de Pernambuco, Recife.

TEIXEIRA, C. E. S.; SILVA, L. B.; CYPRIANO, T. M. (2018)“Cidades inteligentes com infraestrutura de comunicação 5G.” Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação) – Centro Universitário do Estado do Pará, Belém.

TEIXEIRA, E. V. (2018)“Virtualização (SDN e NFV) com ênfase em desempenho de rede.” 51 f. Monografia (Especialização em Gestão de Serviços de Telecomunicações) – Universidade Tecnológica Federal do Paraná, Curitiba.

TORBES, A. R. (2018) “NFVis: Um Ambiente para Visualização de Virtualização de Funções de Rede.” 51 f. Monografia (Bacharelado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre.

VAGO, J. D. (2017) “Uma estratégia para estabelecer fluxos em redes SDN-OpenFlow com redução de carga no controlador.” 107 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Pernambuco, Recife.

ZAMUNER, F. D. M.; MARTINS, M. V. (2023) “Estudo de tráfego em redes com tecnologia SDN e NFV.” Trabalho de Conclusão de Curso (Graduação em Engenharia de Telecomunicações) – Universidade Estadual de Campinas – UNICAMP, Limeira.