



UEPB

**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I - CAMPINA GRANDE
CENTRO DE CIÊNCIA E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

GEOVANE FERNANDES DA SILVA

**A PESQUISA E EXPERIMENTAÇÃO CIENTÍFICA COM REDES DEFINIDAS POR
SOFTWARE EM AMBIENTE DE REDE EMULADO**

**CAMPINA GRANDE - PB
2021**

GEOVANE FERNANDES DA SILVA

**A PESQUISA E EXPERIMENTAÇÃO CIENTÍFICA COM REDES DEFINIDAS POR
SOFTWARE EM AMBIENTE DE REDE EMULADO**

Trabalho de Conclusão de Curso de Graduação em Ciência da Computação da Universidade Estadual da Paraíba, como requisito à obtenção do título de Bacharel em Ciência da Computação.

Orientador: Dr. Wellington Candeia de Araújo

**CAMPINA GRANDE - PB
2021**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

S586p Silva, Geovane Fernandes da.
A pesquisa e experimentação científica com redes definidas por software em ambiente de rede emulado [manuscrito] / Geovane Fernandes da Silva. - 2021.
31 p. : il. colorido.

Digitado.
Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2021.
"Orientação : Prof. Dr. Wellington Candeia de Araújo ,
Coordenação do Curso de Computação - CCT."
1. Redes definidas por software. 2. Pesquisa científica. 3. Redes Wi-Fi. I. Título

21. ed. CDD 004.6

GEOVANE FERNANDES DA SILVA

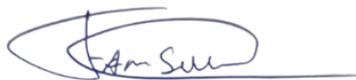
**A Pesquisa e Experimentação Científica com Redes Definidas por Software
em Ambiente de Rede Emulado**

Trabalho de Conclusão de Curso de
Graduação em Ciência da
Computação da Universidade Estadual
da Paraíba, como requisito à obtenção
do título de Bacharel em Ciência da
Computação.

Aprovada em 20 de Outubro de 2021.



Prof. Dr. Wellington Candêia de Araujo (DC - UEPB)
Orientador(a)



Prof. Me. Francisco Anderson Mariano da Silva (CCEA - UEPB)
Examinador(a)



Prof. Vinícius Reuteman Feitoza Alves de Andrade (CCEA - UEPB)
Examinador(a)

À minha família por todo apoio, orações e esforços empreendidos, em especial minha amada esposa e meus filhos, Sophia e Samuel, DEDICO.

“We hope that OpenFlow will gradually catch-on in other universities, increasing the number of networks that support experiments. We hope that a new generation of control software emerges, allowing researchers to reuse controllers and experiments, and build on the work of others (MCKEOWN et al., 2008).

LISTA DE ILUSTRAÇÕES

Figura 1 –	Camadas da arquitetura SDN.....	16
Figura 2 –	Exemplo do Cenário.....	21
Figura 3 –	Exemplo de topologia <i>Spine-Leaf</i>	23
Figura 4 –	Taxa de Transferência (Mb/s) - sem fio.....	23
Figura 5 –	Taxa de Transferência (Mb/s) - com fio.....	24
Figura 6 –	RTT - <i>Round Trip Time</i> (ms) - sem fio.....	25
Figura 7 –	RTT - <i>Round Trip Time</i> (ms) - com fio.....	25
Figura 8 –	<i>Jitter</i> em média (4 <i>Spine</i> , 6 <i>Leaves</i>).....	26

LISTA DE QUADROS

Quadro 1 – Abordagens e componentes de <i>software</i> presentes na literatura....	18
Quadro 2 – Configuração de <i>hardware</i> e <i>software</i>	20

LISTA DE ABREVIATURAS E SIGLAS

ONF	<i>Open Networking Foundation</i>
ONOS	<i>Open Network Operating System</i>
RAN	<i>Radio Access Network</i>
RFC	<i>Request for Comments</i>
SDN	<i>Software Defined Networking</i>
TCP	<i>Transmission Control Protocol</i>
UDP	<i>User Datagram Protocol</i>

LISTA DE SÍMBOLOS

- © Copyright
- ® Marca Registrada

SUMÁRIO

1. INTRODUÇÃO	12
1.1 JUSTIFICATIVA	12
2. REFERENCIAL TEÓRICO	14
2.1 O paradigma SDN	14
2.1.1 O protocolo OpenFlow	14
2.1.2 Arquitetura SDN	15
2.1.3 O Mininet	16
2.1.3.1 - O Mininet-WiFi	16
2.1.4 Os controladores OpenFlow	17
2.1.4.1 - ONOS	17
2.1.4.2 - OpenDayLight	17
2.1.5 Utilitários e ferramentas	17
2.1.5.1 - Iperf	17
2.1.5.2 - Wireshark	17
3. METODOLOGIA	18
3.1 Escolha do cenário	18
3.1.1 Ferramentas auxiliares	19
3.2 Instalação do cenário	20
3.3 Experimentos no Cenário	21
3.3.1 Topologias sugeridas	21
3.3.2 Taxa de transferência	23
3.3.3 Latência	24
4. RESULTADOS E DISCUSSÕES	26
5. CONSIDERAÇÕES FINAIS	27
REFERÊNCIAS	28

A PESQUISA E EXPERIMENTAÇÃO CIENTÍFICA COM REDES DEFINIDAS POR SOFTWARE EM AMBIENTE DE REDE EMULADA

SCIENTIFIC RESEARCH AND EXPERIMENTATION WITH SOFTWARE DEFINED NETWORKS IN A EMULATED NETWORK ENVIRONMENT

Autor Geovane Fernandes¹

RESUMO

Mediante a evolução em comunicações sem fio atualmente marcadas com o advento da 5ª geração de comunicação móvel, 5G, a nível mundial, viabiliza o desenvolvimento de material científico em torno da infraestrutura utilizada e da arquitetura de rede aplicada. Um dos aspectos apresentados nesta evolução é que, o 5G faz uso dos princípios do paradigma de Redes Definidas por Software (do inglês, *Software Defined Networking - SDN*), materializando as vantagens a ele associadas, dessa forma, percebe-se a relevância da pesquisa científica, onde em algum momento as teorias e experimentos tornam-se concretos e parte do nosso cotidiano. Em uma arquitetura que faça uso do paradigma SDN o processo de escolha de quais softwares são necessários para criação de uma rede com seus princípios é árduo, visto a constante atualização dos controladores SDN, necessitando a reavaliação constante do desempenho destes. Com intuito de promover o desenvolvimento de pesquisas científicas na área, que produzam avaliações, testes, protocolos e otimização das tecnologias associadas, propõe-se um cenário de execução de redes definidas por software composto por softwares *open-source*, com o controlador ONOS (*Open Network Operating System*) e o emulador de rede Mininet-WiFi, estas são partes principais juntamente com o protocolo OpenFlow a fim de permitir um cenário flexível, seguro e escalável. De modo análogo busca-se reafirmar a viabilidade de pesquisas em SDN no cenário proposto por intermédio de experimentos a realizar-se, com base nas recentes avaliações apresentadas na literatura fazendo o uso do Iperf para geração do tráfego de rede e a ferramenta de análise Wireshark.

Palavras-chave: Redes definidas por software. Pesquisa Científica. Redes Wi-Fi.

¹ Graduando em Ciências da Computação - Universidade Estadual da Paraíba (UEPB) - geovane.silva@aluno.uepb.edu.br

ABSTRACT

Through the evolution in wireless communications currently marked with the advent of the 5th generation of mobile communication, 5G, worldwide, it enables the development of scientific material around the infrastructure used and the applied network architecture. One of the aspects presented in this evolution is that 5G makes use of the principles of the Software Defined Networking (SDN) paradigm, materializing the advantages associated with it, thus, the relevance of the research is perceived. scientific, where at some point the theories and experiments become concrete and part of our daily lives. In an architecture that uses the SDN paradigm, the process of choosing which software is necessary to create a network with its principles is arduous, given the constant updating of SDN controllers, requiring constant reassessment of their performance. In order to promote the development of scientific research in the area, which produce evaluations, tests, protocols and optimization of associated technologies, it is proposed a scenario of execution of networks defined by software composed of open-source software, with the ONOS controller (Open Network Operating System) and the Mininet-WiFi network emulator, these are key parts together with the OpenFlow protocol to enable a flexible, secure and scalable scenario. Similarly, it seeks to reaffirm the feasibility of research in SDN in the proposed scenario through experiments to be carried out, based on recent assessments presented in the literature using Iperf to generate network traffic and the Wireshark analysis tool .

Keywords: Software defined networks. scientific research. Wi-Fi networks.

1 INTRODUÇÃO

Este presente trabalho tem como objetivo geral expor um estudo de caso que faça o uso do paradigma de Redes Definidas por *Software* (do inglês, *Software Defined Networking - SDN*) a fim de fomentar o desenvolvimento de material científico nesta área do conhecimento, como também explorar *softwares* necessários para desenvolvimento de um cenário base de implantação, desenvolvimento e experimentação de pesquisas científicas em redes definidas por *software*.

Temos como objetivos específicos promover a temática de SDN mediante a apresentação de um cenário básico de rede, que permita viabilizar o desenvolvimento de aplicações e soluções de rede, bem como, incrementar possíveis adaptações por intermédio de funcionalidades extras. Busca-se delimitar o cenário porém sem que impossibilite um leque de opções a serem seguidas, para isso, prioriza-se o uso de *softwares* de código-fonte aberto (*open-source*), até pelo fato do custo-benefício associado. De forma similar é optado utilizar um emulador de rede que cubra o máximo de requerimentos básicos e abranja redes sem fio (*wireless*) e fixas (*wired*), com isso, é utilizado o Mininet-WiFi (FONTES; ROTHENBERG, 2019) a fim de simular dispositivos de rede, bem como gerenciá-los. Em conjunto com o Mininet-wifi utiliza-se o protocolo OpenFlow para prover a conexão com o controlador OpenFlow, oportunamente escolhido o ONOS (*Open Network Operating System*) devido a amplitude de cenários que este pode abranger.

A estrutura emuladora de rede permite a criação de um ambiente seguro (SECCI, 2019), escalável (MAMUSHIANE, 2018); (JOURNAL, 2021), com fidelidade ao mundo real e reproduzível (FONTES; ROTHENBERG, 2019), o qual será submetido a experimentos de desempenho, latência e submetido a topologias.

1.1 JUSTIFICATIVA

Mediante a forte divulgação de trabalhos na área de redes definidas por *software* nos últimos anos mesmo assim torna-se imprescindível a abordagem de formas diversas por intermédio de publicações científicas e em cenários distintos. É precisamente necessária a atualização dos *softwares* e componentes envolvidos, pois muitos são descontinuados ou substituídos, como por exemplo o *Floodlight*, tendo sua última versão publicada em 2016 conforme repositório (FLOODLIGHT, 2021), que mesmo substituído por derivados e novas gerações frequentemente mantidas, foi por anos um dos principais controladores, demonstrando excelentes resultados no geral em comparação com outros como POX e NOX (PADILHA, 2017) e ainda é utilizado em aplicações específicas e personalizadas.

Dentre novas tecnologias emergentes, o padrão IMT-2020 ou 5G (ITU-R, 2015) proporcionou o crescimento dos estudos envolvendo o paradigma SDN aplicado na arquitetura de rádio (*SDN-based RAN*) (ZAIDI, et al. 2017), esta quinta geração oferece o suporte nativo na divisão dos fluxos de controle, de dados e o fatiamento (*slicing*) da rede, que permite a otimização dos recursos de acordo com os critérios dos serviços, tais características são premissas ao desenvolvimento do OpenFlow (OPENFLOW, 2015) elencadas por McKeown et al. (2008). 5G tornou-se a aplicação mais atual em redes definidas por *software* que impactará diretamente nossa vida cotidiana.

A indústria já está a absorver essas mudanças, a rede passa a ser dinâmica no ponto de vista de se adequar a cada ambiente ou serviço de acordo com a programação a ela associada. É possível perceber que grandes empresas possuem cada dia mais casos de sucesso na área e estão atentas às novas tendências, como exposto no Relatório anual da Cisco, padrões que fazem uso da abordagem SDN, computação de borda, redes baseadas em intenções e entre outros (CISCO, 2020).

Vislumbrando o futuro, temos a certeza que o apresentado por McKeown et al. (2008), de propor um novo recurso de comutação que ajudasse a ampliar a programabilidade nos ambientes de redes universitárias, foi apenas o primeiro passo a transferir inteligência à rede, permitindo que o comportamento seja autoprogramável visto as demandas dos serviços cada vez mais exigentes. Como exposto por Golash:

Com a crescente demanda por requisitos de largura de banda e largura de banda elástica, a rede deve ter uma capacidade de redimensionar o pool de recursos eficientemente para que a utilização máxima da largura de banda existente ocorra. (GOLASH, 2015, p. 03, tradução nossa)²

Em suma, é inegável a importância de pesquisas em redes programáveis, sejam elas voltadas para indústria, casas inteligentes (*Smart Home*), cidades inteligentes (*Smart Cities*) ou qualquer outra área de aplicação. Em paralelo, o OpenFlow ainda se destaca como protocolo de comunicação segundo Padilha (2017), dada a importância do controlador OpenFlow, este fornece uma plataforma de operação para diversas aplicações de controle e gerenciamento, é fundamental compreender o desempenho e o comportamento deste software para a operação em uma SDN (PADILHA, 2017). O autor salienta a importância desenvolvimento em pesquisas e testes de desempenho, a fim de levar os benefícios de redes definidas por software as operadoras comerciais e conseqüentemente ao usuário final, porém ele ainda elenca um dos principais empecilhos para uma aplicação efetiva da nova arquitetura:

Entretanto, pouco se sabe sobre a estabilidade e desempenho dessas plataformas de controle. Desse modo, escolher o controlador correto para um dado cenário de rede exige uma análise aprofundada sobre esses quesitos nos diversos modelos existentes (PADILHA, 2017).

Este ponto levantado por Padilha é o objeto de estudo deste trabalho, pois sabe-se que existem diversos possíveis cenários em SDN, onde podem ser aplicadas um conjunto de ferramentas específicas, entretanto torna-se uma tarefa dispendiosa a busca pela melhor solução dado um ambiente proposto, este ponto de fragilidade da escolha do controlador dar-se pelo fato deste estar logicamente centralizado e ser o "Cérebro" da rede. Tal dificuldade na escolha do conjunto de *softwares* para compor o desenvolvimento de pesquisas na área necessita ser contornada, e este processo se dá na constante criação de material científico avaliativo, comparativo e quantitativo.

O presente trabalho está estruturado da seguinte forma: na Seção 2 tem-se o Referencial Teórico com uma breve explanação sobre SDN e das ferramentas propostas de acordo com a literatura; na Seção 3 trata-se da Metodologia utilizada

² With increasing demand for bandwidth and elastic bandwidth requirements, the network should have an ability to resize resource pool efficiently so that maximum utilization of existing bandwidth takes place. (GOLASH, 2015, p. 03, trecho original)

para desenvolvimento da pesquisa; na Seção 4 apresenta-se o Resultado e Discussões, nesta será realizada uma visão geral dos objetivos alcançados e manifestar as possíveis aplicações do cenário; por fim, têm-se as Considerações Finais do trabalho.

2 REFERENCIAL TEÓRICO

A estrutura de redes conhecida por internet vem sofrendo mudanças desde sua concepção, novas demandas foram surgindo, novas tecnologias foram desenvolvidas com intuito de suprir requisitos associados a tais demandas e dessa forma foi-se evoluindo a visão que se tinha de redes de computadores. Passou-se de uma visão de *intranet* para constituição de uma rede mundial de computadores, não obstante, limitações de recursos acompanham o crescimento e desenvolvimento das redes, desde o endereçamento como o que se tem no protocolo IPV4, esgotadas as reservas em 2020 aqui no Brasil (IPV6, 2021), aos presentes desafios com altas de demandas relacionadas à serviços *web* XaaS (*X as a Service*, qualquer coisa como um serviço), processo de evolução exposto por Golash (2015), estes são serviços nativamente locais passam a ser baseados em nuvem, distribuídos e carregam consigo prioridades de disponibilidade, eficiência, segurança e escalabilidade. Golash ainda nos traz que, a atual arquitetura de rede utilizada foi desenvolvida a princípio apenas para conexão básica entre computadores ou *e-mails* por exemplo, sofre com a pressão de suportar a carga de diversos novos serviços baseados em nuvem.

2.1 O paradigma SDN

É consenso que qualquer que seja o novo modelo de Internet aceito para o futuro, ele deverá ser flexível para suportar as mudanças de requisitos das próximas gerações de aplicações (Ferreira, 2016). Essa flexibilidade é preceito de um paradigma de arquitetura de rede conhecido por Redes Definidas por Software, este por sua vez abre uma série de possíveis cenários de aplicações vislumbrando um futuro de redes onde não apenas computadores, como também, qualquer coisa pode estar conectada, esta é a IoT (Internet of Things ou internet das coisas). As redes de computadores encontram-se neste processo de mudança e adaptação, de acordo com o relatório anual da internet da CISCO (CISCO, 2020), traz-se a perspectiva das novas tendências em tecnologia para redes, bem como números estimados em 2023 de dispositivos conectados sendo maior que 3 dispositivos por pessoa, somando cerca de 29,3 bilhões, com uma taxa anual de crescimento em 10%, este é um número bastante expressivo quando se trata de arquitetura de redes e faz-se necessário a adoção de soluções tal como em redes definidas por software para contornar o desafio de gerir uma infraestrutura tão robusta necessária a este cenário futuro.

2.1.1 O protocolo OpenFlow

Em 2008, um grupo de pesquisadores da universidade de Stanford em colaboração com pesquisadores de outras universidades propuseram um protocolo de comunicação entre um dispositivo central de rede mencionado como sendo o controlador e qualquer *switch* com suporte à comunicação, este protocolo denominado OpenFlow (MCKEOWN et al., 2008) seria amplamente utilizado em

pesquisas na área desde então e se tornaria uma das principais ramificações do paradigma de redes definidas por software, tomando como base a visão de Ferreira em 2016:

OpenFlow é uma materialização de SDN e é uma contribuição fundamental para a comunidade de pesquisa no âmbito da Internet do Futuro, permitindo o uso e a avaliação de mecanismos para inovar tanto o controle de rede quanto o transporte de dados(FERREIRA, 2016).

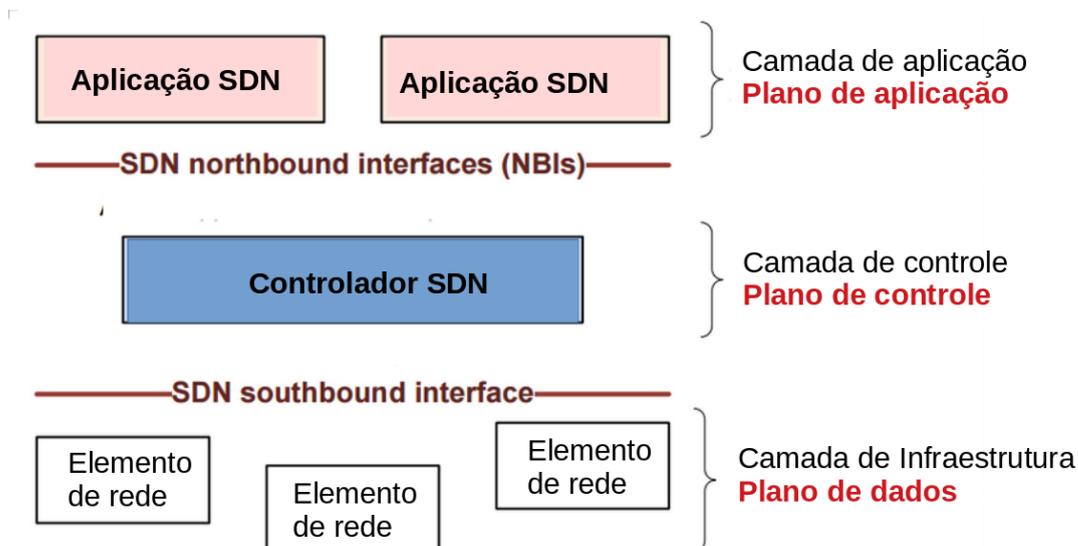
Neste trecho, Ferreira expressa a visão que se tinha do protocolo em 2016, onde em sua versão 1.5.1 (OPENFLOW, 2015), demonstrava estar sempre com o foco na internet do futuro, e a partir dele tornou-se possível o desenvolvimento e ou implantação de outros protocolos de comunicação com as interfaces sul, tais como NETCONF (Network Configuration), OF-CONFIG (OpenFlow Management and Configuration), OVSDB (OVS Database Management Protocol), BGP (Border Gateway Protocol), XMPP (Extensible Messaging and Presence Protocol), entre outros protocolos (ENNE, 2020).

2.1.2 Arquitetura SDN

A arquitetura em redes definidas por software tem como base três camadas básicas de acordo com a *Open Networking Foundation - ONF* (TR-502, 2014) :

- **Camada de aplicação:** Envia requerimentos de rede em direção ao plano do controlador por meio de interfaces norte, conhecidas por *North Bound Interfaces - NBIs* ou *northAPIs*);
- **Camada de controle:** (plano de controle) Intermedia a comunicação entre o plano de aplicação com o plano de dados, recebe e traduz os requisitos de rede vindos das aplicações e realizando o controle de baixo nível na camada de infraestrutura;
- **Camada de infraestrutura:** (plano de dados) composta por elementos de rede (switches, roteadores IP, etc.), que expõem seus recursos para a camada de controle (plano de controle) por meio de interfaces sul do controlador (*South Bound Interfaces - southAPIs*), recebem e executam instruções da camada de controle conduzindo pacotes por meio do caminho de dados.

Figura 1 - Camadas da arquitetura SDN



Fonte: (TR-502,2014. adaptada)

Além das camadas detalhadas na Figura 1, SDN é baseada em três princípios arquiteturais e na definição de interfaces abertas. Os princípios são os seguintes: Separação do plano de controle e do plano de dados; Controle centralizado logicamente; Serviços de rede programáveis. Neste trabalho buscou-se propor ferramentas de *software* que atendessem esses princípios e fossem amplamente utilizadas atualmente.

2.1.3 O Mininet

Esta ferramenta possibilita o desenvolvimento de ambientes protótipos de rede com o uso do paradigma de redes definidas por *software*, com ela cria-se uma rede virtual no entanto realista pois possui componentes reais, isso tudo com a praticidade de execução em apenas uma máquina, que pode ser uma máquina robusta para simular redes de maior porte ou apenas um computador pessoal, com configurações básicas (MININET, 2021). Permite a criação dos nós da rede por intermédio de linha de comando ou com auxílio da linguagem de programação Python.

2.1.3.1 - O Mininet-WiFi

Uma ramificação do Mininet foi desenvolvida por Fontes e Esteve Rothenberg (2019), a qual incrementa funcionalidade à ferramenta inicial, adicionou-se a possibilidade de montar uma rede virtual sem fio completa com base nos drivers sem fio Linux e no *driver* de simulação *wireless* 80211_hwsim. Ao incrementar a ferramenta, o autor adicionou e modificou classes e *scripts* a fim de garantir as propriedades sem fio, mas continua permitindo a simulação de todos recursos antes presentes no *software* base. É necessário salientar que o Mininet-WiFi tem como objetivo de ser um emulador de alta fidelidade, sendo o mais fiel possível ao meio emulado fazendo o uso do emulador sem fio Wmediumd, permitindo a reprodução de condições reais de redes sem fio (FONTES; ROTHENBERG, 2019).

2.1.4 Os controladores OpenFlow

Em 2008 as funcionalidades básicas, conexão e comportamento que deveriam fazer parte de um *switch* OpenFlow foram definidas por McKeown et al. (2008), este é um marco histórico, onde foi possível projetar perspectivas para um tratamento de rede completamente diferente, que proporciona a rede ser adaptável a requisitos e tecnologias que venham a surgir, por intermédio de um protocolo que no decorrer de tantas pesquisas mostrou-se ser um dos principais meios de comunicação em uma rede definida por *software*.

2.1.4.1 - ONOS

Com a intenção de criar um controlador SDN de código aberto para construção de soluções em SDN/ NFV, foi desenvolvido o Open Network Operating System, tendo *release* inicial em dezembro de 2014, teve o desenvolvimento liderado pela *Open Network Lab* (ON.LAB) e empresas privadas. Com foco em atender demandas de redes de provedores de internet, este suporta configuração em tempo real, eliminando a necessidade de protocolos de controle de roteamento, movendo a inteligência para um controlador em nuvem. Esta é uma das principais características que o diferencia dos demais, o tratamento descentralizado, distribuído, fornecido em conjunto com outras soluções que compõem a infraestrutura de rede em nuvem sugerida pela ONF - *Open Networking Foundation*, atual mantenedora da solução.

2.1.4.2 - OpenDayLight

Mais conhecido por ODL, o OpenDayLight trata-se de um projeto de *software* aberto custeado em conjunto por vários fabricantes de equipamentos de rede. É uma plataforma aberta modular para personalizar e automatizar redes de qualquer tamanho e escala. Foi projetado desde o início como uma base para soluções comerciais que atendem a uma variedade de casos de uso em ambientes de rede existentes. Mantido pela *Linux Foundation* e assim como o ONOS foi desenvolvido em Java e possui o Apache Karaf como componente básico (OPENDAYLIGHT, 2021).

2.1.5 Utilitários e ferramentas

2.1.5.1 - Iperf

É uma ferramenta em linha de comando que gera um fluxo de rede de acordo com o protocolo especificado, para obter medições ativas da largura de banda máxima alcançável em redes IP (*Internet Protocol* ou Protocolo da Internet), possui suporte a protocolos (TCP, UDP, SCTP com IPv4 e IPv6). Para cada teste realizado, este relata a largura de banda, perda de pacotes, *jitter* e etc, de acordo com configurações e protocolos pré-definidos (IPERF, 2021).

2.1.5.2 - Wireshark

Esta ferramenta possibilita a captura e análise do fluxo de rede, com análise aprofundada do protocolo de rede utilizado. Inicialmente desenvolvido por Gerald Combs em 1998, hoje é mantido por contribuições de uma comunidade de especialistas ao redor do mundo. É um *software* multiplataforma, provê a captura em tempo real e análise *offline*, possui uma grande quantidade de filtros de exibição,

utilitários de análise de protocolos específicos e tem suporte de descryptografia para muitos protocolos, incluindo IPsec, ISAKMP, Kerberos, SNMPv3, SSL / TLS, WEP e WPA / WPA2 (WIRESHARK, 2021).

3 METODOLOGIA

Com o intuito de obter referências bibliográficas e sumarizar estudos primários, é realizado um mapeamento sistemático da temática, onde por intermédio de fontes de busca como o IEEE Xplore, Springerlink, ACM Digital Library e ResearchGate lista-se trabalhos recentes contendo avaliações entre controladores e ou cenários em redes definidas por *software*. O mapeamento é concebido no período de 2016 à agosto de 2021, obtendo 150 estudos, observa-se como critérios de exclusão a ordem de relevância dos 50 estudos de cada fonte, o contexto deve ser avaliativo e ou comparativo e o estudo deve ser público. Aplicados os critérios, observou-se 21 estudos base, que serão utilizados dado a sua ordem de relevância.

Mediante o desenvolvimento de pesquisas em redes definidas por *software* nos últimos anos, busca-se realizar um levantamento de quais controladores e componentes seriam mais adequados atualmente para impulsionar pesquisas no neste campo, visto que o controlador é um ponto de fragilidade em uma arquitetura de rede centralizada e os componentes de *software* envolvidos são responsáveis pelo desempenho final na execução de uma pesquisa, experimento, "test bed".

Após levantamento dos estudos relevantes na área, busca-se realizar um estudo de caso com base nas sugestões dos autores quais ferramentas eram mais utilizadas e dentre estas as mais bem avaliadas de acordo com as métricas apresentadas em cada estudo, porém levando em consideração o tempo de resposta, taxa de transferência e latência, características estas, que compõem parte da padronização desenvolvida no estudo apresentado por Nascimento (2017), em conjunto com a RFC 8456 (VENGAINATHAN, et al., 2018), RFC 7426 (Haleplidis, et al., 2015) e de acordo com a natureza dos experimentos realizados em estudos presentes na literatura.

3.1 Escolha do cenário

Com base na análise empreendida sob os estudos anteriores, possibilita-se elencar os componentes de *software* recentemente utilizados em experimentos, testes de *benchmarking* e avaliações vistos no Quadro 1:

Quadro 1 - Abordagens e componentes de software presentes na literatura

Estudo	Controlador	Ferramentas	Métricas
(Ferreira, 2016)	ONOS, OpenDayLight	TestON e Mininet	Requisitos de redes Carrier Grade
(Padilha, 2017)	POX, Ryu, Floodlight	Cbench, OFCBenchmark, OFCProbe, HCProbe, Collectl e Mininet	Perda de pacotes, Latência, Fluxos por segundo, Utilização de CPU e Memória, Desempenho, Segurança
(Mamushiane et al., 2018)	OpenDayLight, ONOS, Ryu, FloodLight	Cbench	Latência e Taxa de Transferência, Desempenho(Performance agregada), Escalabilidade

(Goswami, 2019)	ONOS	Docker, Iperf, Mininet	Taxa de Transferência
(Istam et al., 2019)	POX, ONOS, Ryu, FloodLight	Mininet-WiFi, Iperf	Delay, Jitter, Taxa de Transferência
(Zhu et al., 2019)	NOX, POX, Floodlight, ODL, ONOS, Ryu, OpenMUL, Beacon e Maestro.	CBench, PktBlaster e OFNet.	Latência, Taxa de transferência, RTT(Round Trip Time), utilização da CPU e Fluxos por segundo
(Journal, 2021)	OpenDayLight e ONOS	Mininet, Iperf	Jitter, Latência, Delay, Taxa de transferência

Fonte: Autoria própria, 2021.

Realizando-se análise dos *softwares* expostos na Quadro 1 juntamente ao apresentado por Zhu et al. (2019), percebe-se que nos últimos anos os cenários montados tendem a tomar novos componentes de *softwares*, principalmente quando se trata de uma das partes principais de uma arquitetura SDN, o controlador, destacam-se o ONOS e o OpenDaylight (ODL), estes são controladores recentemente lançados comparando-os com os demais, em 2014 mantido pela Open Networking Foundation (ONF) e 2013 mantido pela Linux Foundation respectivamente. De acordo com características inerentes ao desenvolvimento de pesquisas científicas tais como: a deficiência de espaços físicos adequados, a insuficiência de financiamento industrial, o direcionamento de verbas públicas em incentivo à pesquisa e em decorrência destes últimos, a incapacidade de adquirir recursos de *hardware* e *software* específicos.

Perante o antes exposto, adota-se em nossa solução o ONOS, pois este é um *software* de código-fonte aberto atendendo a requisitos antes elencados, construído para soluções SDN/NFV, flexível à implantação de novos serviços de rede dinâmicos, extensível, distribuído (ONOS, 2021) e com suporte a categoria de redes carrier grade (Ferreira, 2016). Em conjunto com o ONOS observou-se a utilização do emulador de rede Mininet, onde no cenário proposto iremos utilizar o Mininet-WiFi, por este contemplar recursos de rede sem fio e todas características antes expostas na Seção 2. Ao adotar o uso do Mininet temos intrínseco a este o Open Vswitch que é um switch virtual multicamadas fornecendo suporte ao protocolo OpenFlow e recursos idênticos à um *switch ethernet*. Estes recursos de *software* a serem utilizados são amplamente mantidos por comunidades de *software* de código-fonte aberto, têm capacidade de gerir cenários comerciais e exaltam a necessidade de desenvolvimento com baixo custo presente em projetos de pesquisa.

3.1.1 Ferramentas auxiliares

Na Quadro 1 são expostas diversas ferramentas de *software* utilizadas como auxílio no desenvolvimento das pesquisas, algumas destas têm como objetivo a execução de testes de *benchmark*, tais como o CBench, OFNet, OFCProbe, TestON, PktBlaster e Iperf, amplamente utilizadas em outras pesquisas. Dentre estes, como apoio à esta pesquisa, será escolhido o Iperf pois não existe a pretensão em realizar avaliação entre controladores e sim validar requisitos mínimos presentes na literatura em aplicação no cenário a ser sugerido. Como *software* extra para compor o cenário de experimentos, utiliza-se o Wireshark, uma ferramenta completa na análise de fluxos de redes.

O Mininet aparece em grande parte do Quadro 1, observada assim sua relevância no desenvolvimento de estudos na área, o presente trabalho fará o uso do Mininet-WiFi, visto que, este tem os mesmos recursos apresentados pelo Mininet incrementados da possibilidade do uso em redes *wireless* conforme apresentado na Seção 2.

3.2 Instalação do cenário

Com intuito de proporcionar um ambiente flexível e de baixo custo utiliza-se uma configuração de *hardware* básica, pois o cenário a ser apresentado não requer forte estrutura computacional, demonstrando-se assim que a criação de aplicativos com intuito prover soluções específicas de uma rede como por exemplo balanceadores de carga, *firewalls* e gerenciadores de fluxo é uma tarefa possível independente dos meios físicos dispostos. Faz-se uso de distribuições GNU/Linux, pois estas são plataformas base para desenvolvimento de muitos *softwares* de código aberto, como partes principais do cenário têm-se o ONOS, que foi compilado e instalado na Máquina 1 no sistema nativo e via Docker, na mesma máquina foi instalado o Mininet-WiFi e como máquina virtual, arquivo .ova, disponível no site³. A Máquina 2 servirá como apoio na execução do cenário. No Quadro 2 apresenta-se a configuração de *hardware* e *software* a ser utilizada:

Quadro 2 - Configuração de hardware e software

Recursos Físicos	CPU	RAM	Placa de Rede	
Máquina 1	Intel Core i3-4005U	8 GB	Realtek RTL8111/8168/8411 Gigabit Ethernet	Realtek RTL8723BE PCIe
Máquina 2	Intel Core i3-6006U	4 GB	Realtek PCIe GBE Family Gigabit Ethernet	Qualcomm QCA9377
Recursos de Software	Detalhes		Sistema Operacional	Kernel
Java Virtual Machine	OpenJDK 64-Bit Server VM ver. 11.0.8+10-LTS		Debian GNU/Linux 10 (buster) - Máquina 1	4.19.0-17-amd64 x86_64
Karaf	ver. 4.2.9		Windows 10.1 - Máquina 2	
Mininet-Wifi	ver. 2.4.3		Lubuntu 20.04 - Máquina Virtual	5.11 - x86_64
Docker	ver. 20.10.8			
ONOS	ver. 2.7.0			
Imagem do ONOS	https://hub.docker.com/r/onosproject/onos			
Iperf	ver. 3.6 e ver. 2.0.12			
Wireshark	ver. 2.6.20			

Fonte: Autoria própria, 2021.

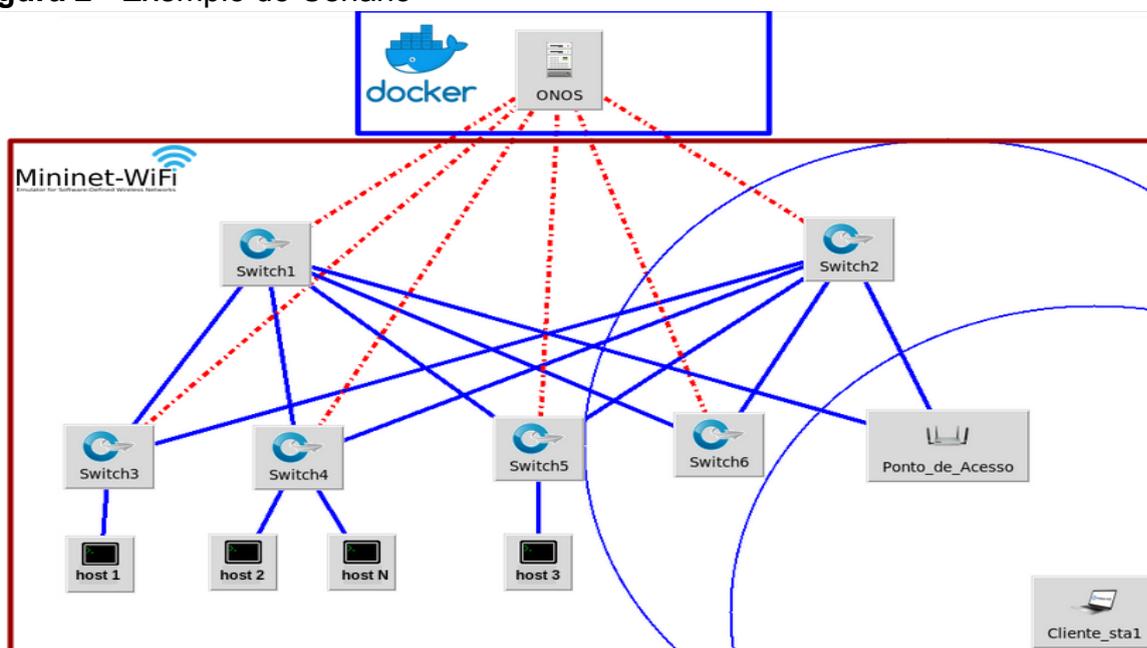
³ <https://hub.docker.com/r/sdnhub/onos>

A configuração de *hardware* disposta no Quadro 2, não deve ser observada como um ponto de insuficiência da rede a ser emulada, tomamos esta como um desafio a ser contornado pelos recursos de tolerância a falhas e otimização a recursos de *hardware* presentes na estrutura de *software* principal.

3.3 Experimentos no Cenário

Com base nos estudos analisados até o momento é possível ser desenvolvida uma sequência básica de experimentos com intuito de pôr à prova o cenário proposto, validando-o e possibilitando o desenvolvimento de novas avaliações e aplicações SDN. Na figura 2 abaixo, é exposto um exemplo do cenário a ser utilizado:

Figura 2 - Exemplo do Cenário



Fonte: Autoria própria, 2021.

Com a ascensão de novas tecnologias de virtualização é possível simular redes, bem como componentes robustos, observando técnicas de virtualização utilizadas por (Goswami, 2019), neste cenário é utilizado o Docker com intuito de executar o controlador da rede em *container*, conseqüentemente tornando-o mais estável e com uma reinicialização rápida, visto os possíveis erros gerados por sobrecarga do *hardware* e reinícios para manter recursos sempre em estado inicial a cada execução de um novo experimento. Neste exemplo são apresentados elementos básicos mistos em uma topologia *Spine-Leaf*, com o controlador, ONOS, com seis *switches* OpenFlow, estes possuindo *hosts* e um ponto de acesso *WiFi* com um cliente. O cenário permite diversas possibilidades de ambientes de testes e experimentos, visto as características heterogêneas de redes físicas diversas.

Nesta subseção iremos expor sugestões de topologias, seguido dos dados, características e dificuldades encontradas nos testes que são realizados.

3.3.1 Topologias sugeridas

Algumas topologias de rede são candidatas a compor a sequência de testes:

- Linear

De modo que utiliza-se o mininet-WiFi como emulador de rede, temos a topologia Linear como padrão, na execução sem atributos. Como por exemplo:

```
# mn --wifi
```

Neste comando será executado todos atributos de configuração padrão, gerando uma rede com nós alinhados em um canal de infraestrutura de ponta a ponta, possuindo 1 *switch* OpenFlow (Ponto de acesso - OVSSwitch), 2 *hosts* (Estação - cliente).

- Árvore

Topologias em árvore ou hierárquica baseia-se na divisão da rede em níveis hierárquicos de sub-redes, uma das principais vantagens isto é a inserção de duas ou mais topologias distintas, obtendo assim vantagens de cada uma delas. um exemplo de simulação de topologia em árvore tem-se no comando:

```
# mn --controller=remote,ip=,port=6653 --topo=tree,depth=2,fanout=4
```

Neste comando é gerada uma rede com dois níveis de profundidade e dispersão de quatro nós (*hosts*) por nó folha de infraestrutura.

- Mesh

Esta topologia permite a conexão de cada nó de infraestrutura da rede, com objetivo de formar uma rede com vários possíveis caminhos o cliente pode estar conectado com qualquer nó. Essas características podem influenciar na resiliência, escalabilidade e robustez.

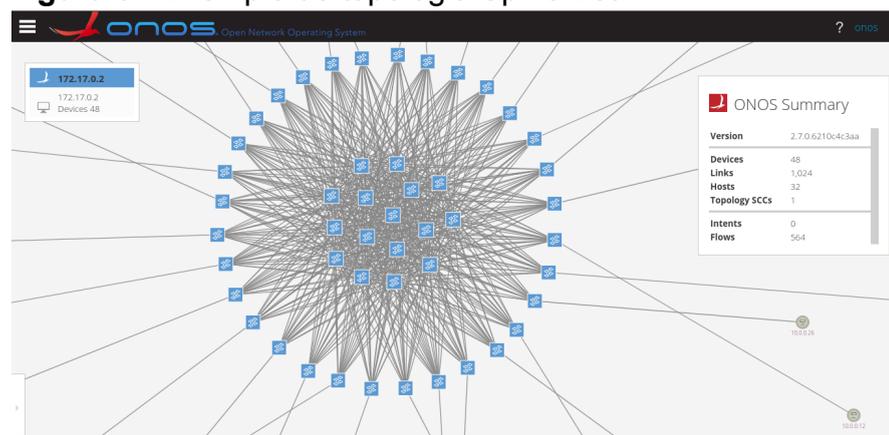
Não existem argumentos específicos para gerar uma rede deste tipo com o Mininet-WiFi, entretanto existem exemplos no repositório da própria distribuição que possuem o esboço básico para se criar versões personalizadas, onde a topologia pode ser montada por meio da linguagem Python (PYTHON, 2021).

- Spine-Leaf

A utilização é aplicável em *data centers*, sendo observado a disposição dos nós em níveis, normalmente dois, a camada de *Leaf* que refere-se a camada de acesso e a camada *Spine* é responsável pela interconexão de todos switches *Leaf*. Esta conexão entre as camadas se dá por uma rede em malha, onde cada nó da camada inferior (acesso) tem *links* com todos *switches* da camada superior (agregação). Tornando a carga de tráfego distribuída, latência previsível, largura de banda variável.

Por padrão não é possível gerar esta topologia diretamente na linha de comandos do Mininet, desta forma utiliza-se a mesma abordagem da topologia em mesh antes apresentada, no entanto com intuito de fornecer mais ferramentas auxiliares realiza-se o uso da ferramenta Mininet Topologies (MININET-TOPOS, 2016) e do utilitário gráfico MiniEdit, um *script* python disponível nos exemplos do Mininet (FONTES; ROTHENBERG, 2019).

Figura 3 - Exemplo de topologia Spine-Leaf



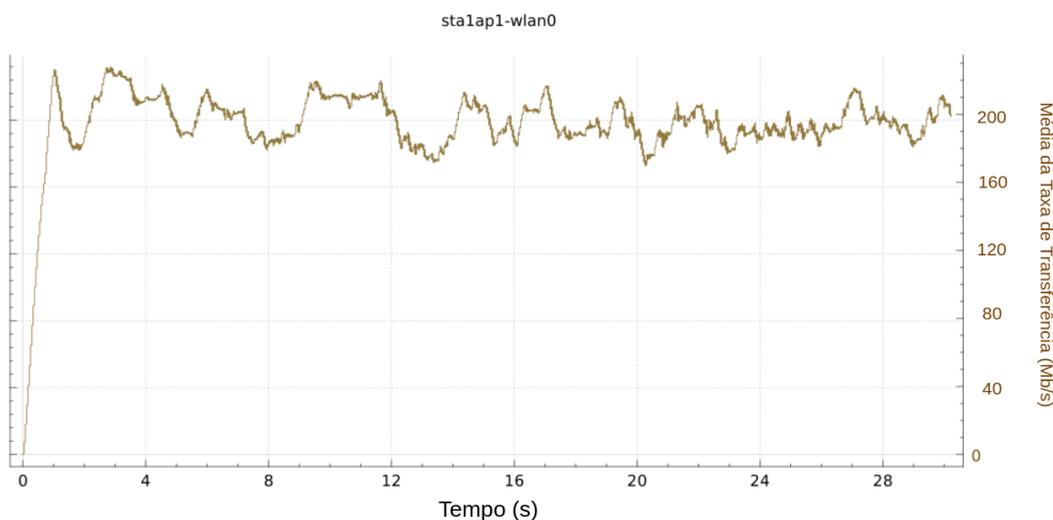
Fonte: Autoria própria, 2021.

3.3.2 Taxa de transferência

Toma-se como base o estudo realizado por Islam et al. (2019), que realizou avaliação e comparação de quatro controladores OpenFlow e se destacou por implantar nas configurações de testes o conceito de redes sem fio definidas por *software* (*Software Defined Wireless Network - SDWN*), por intermédio do Mininet-WiFi. Em conjunto com fundamentos para realização de testes de benchmarking, como por exemplo a quantidade de execuções dos experimentos e as topologias sugeridas na RFC 8456 (VENGAINATHAN, 2018). Contudo esta subseção visa corroborar com estudos antes citados e proporcionar o desenvolvimento de novos experimentos, avaliações e desenvolvimento de aplicações.

Para auxiliar o desenvolvimento dos experimentos foram utilizadas as seguintes ferramentas: Iperf para gerar o fluxo de dados TCP - *Transmission Control Protocol* e UDP - *User Datagram Protocol* entre os nós envolvidos; Wireshark para capturar o fluxo de pacotes e posteriormente realizar o isolamento do protocolo desejado e permitir o esboço do gráfico de desempenho desejado.

Figura 4 - Taxa de transferência (Mb/s) - sem fio

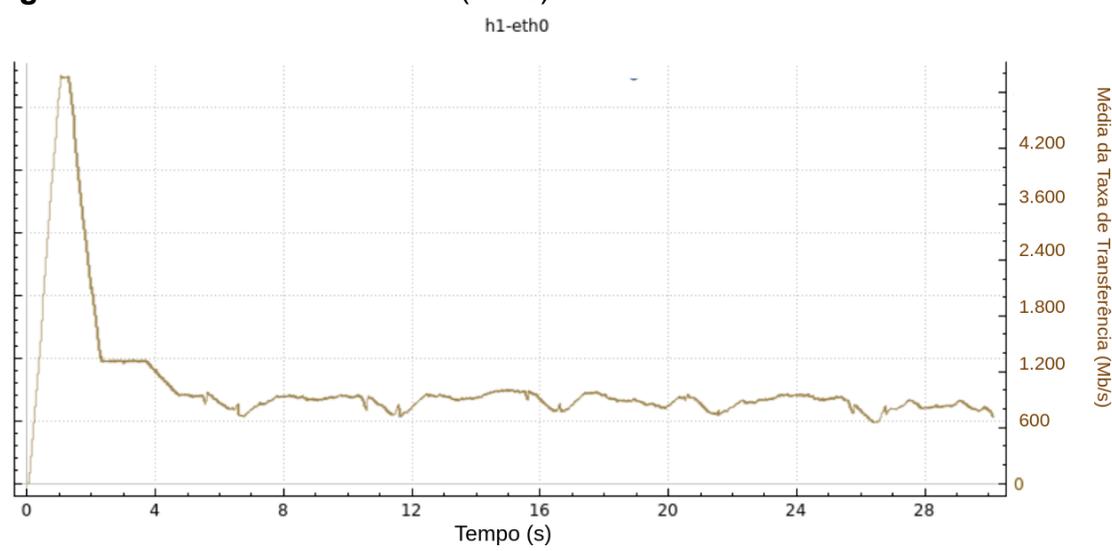


Fonte: Autoria própria, 2021.

De acordo com a Figura 4, que mostra o resultado da média da taxa de transferência obtida no cenário proposto com uma topologia, linear, semelhante à apresentada no estudo de Islam et al. (2019) de forma otimizada, visto que, para obter o desempenho real é necessário incrementar a largura de banda com valores acima da velocidade máxima teórica suportada pelo dispositivo. Onde na avaliação o autor limitou a largura de banda à 10Mb/s, e mediante a configuração de *hardware* apresentada poderia ser melhor explorada, conforme o Quadro 2, com 500Mb/s, fluxo de dados TCP, 4 *switches* com 2 *hosts* cada de acordo com a Figura 4, alcançando cerca de 200Mb/s em média, isso mediante a parâmetros do protocolo IEEE 802.11n.

Com o intuito de salientar as possibilidades oferecidas pelo cenário é realizado o mesmo experimento supracitado utilizando parâmetros presentes na RFC 8456, VENGAINATHAN (2018), que é executado sob topologia Spine-Leaf e conexão com fio a fim de obter o máximo de desempenho da rede emulada, observando taxa de transferência, *delay* e *jitter* associados.

Figura 5 - Taxa de transferência (Mb/s) - com fio



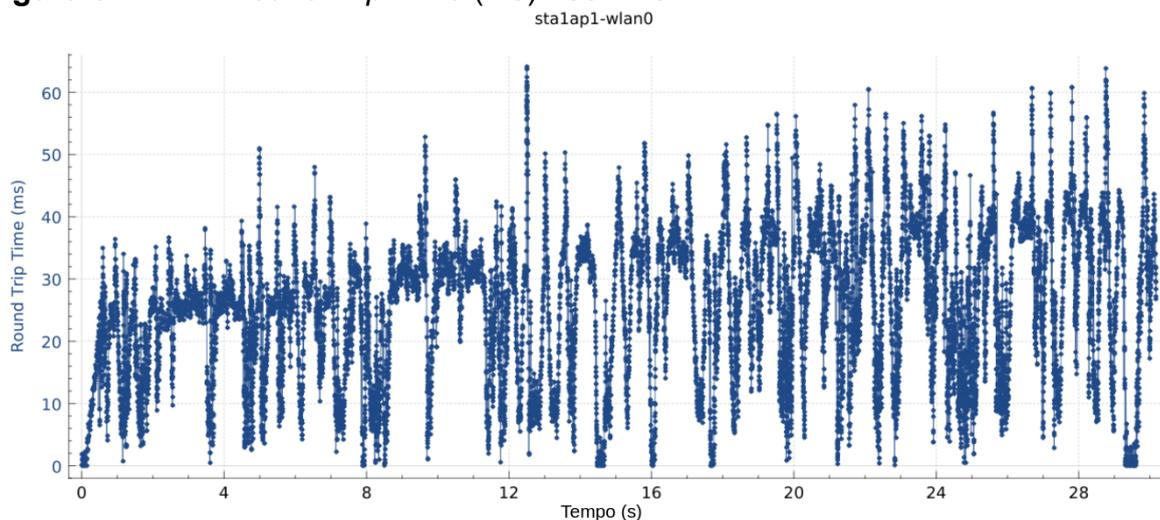
Fonte: Autoria própria, 2021.

Conforme Figura 5, apresenta o resultado médio dos experimentos realizados de taxa de transferência com largura de banda livre em um fluxo de dados TCP, em uma rede com topologia spine 16 - leaves 32 e 32 hosts cada leaf conforme Figura 3, obtendo taxa de transferência em cerca de 600Mb/s

3.3.3 Latência

Considerando em primeiro momento um ambiente sem fio, realizam-se experimentos de *delay* de pacotes em análise com auxílio do Wireshark, do fluxo de taxa de transferência sob 200 Mb/s da Figura 4, gerado a fim de definir a largura de banda suportada pela máquina em rede sem fio, os resultados podem ser observados na Figura 6. Fazendo uma breve análise é possível perceber a variação associada ao tipo de comunicação, no entanto ainda obtendo bons resultados de RTT (*Round Trip Time*), tempo necessário entre o envio de um pacote e a recepção de confirmação de entrega do mesmo, em média cerca de 30 ms.

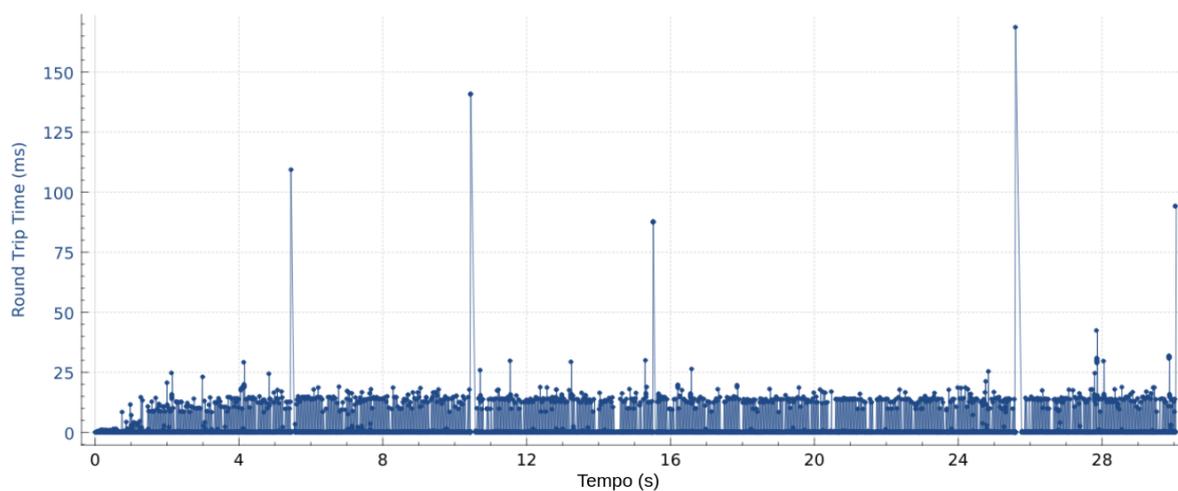
Figura 6 - RTT - Round Trip Time (ms) - sem fio



Fonte: Autoria própria, 2021.

De modo análogo ao experimento anterior, realiza-se a reanálise dos pacotes resultantes do fluxo de taxa de transferência sob 600 Mb/s exposto acima na Figura 5, os resultados são estáveis e na média de 20 ms conforme Figura 7.

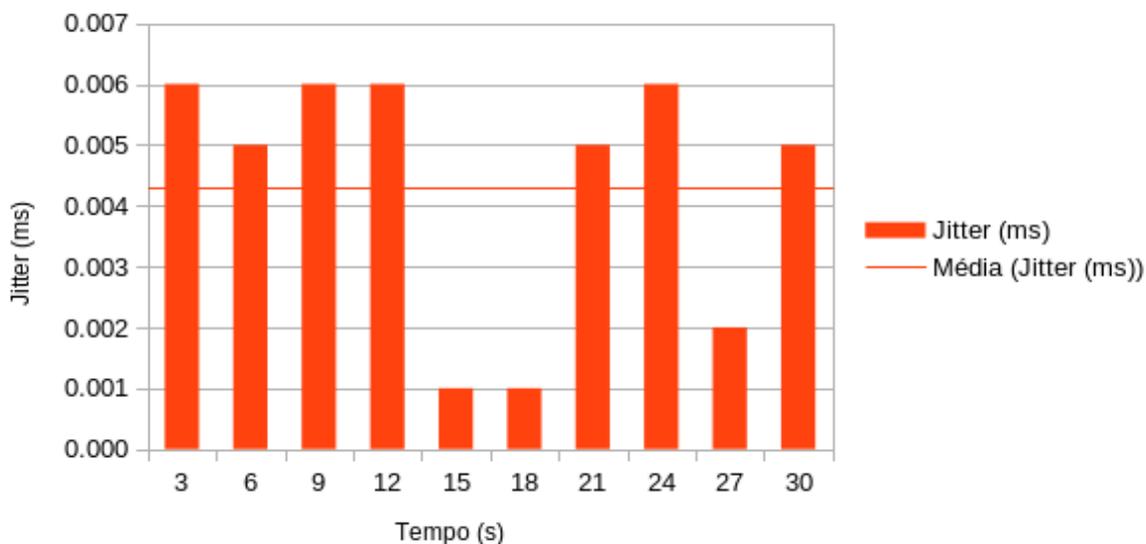
Figura 7 - RTT - Round Trip Time (ms) - com fio



Fonte: Autoria própria, 2021.

Com auxílio do Wireshark realiza-se a filtragem dos pacotes observando valores de atraso entre eles, ou seja o *Jitter*, para isto utilizou-se um fluxo diferente dos demais experimentos realizados, com UDP com largura de banda de 1000 Mb/s, 4 *spines* e 6 *leaves* com 1 *host* cada. Nota-se que os valores aferidos estão dentro do limite mínimo determinado como meta em 2011 (Resolução ANATEL N° 574, 2011), de 50 ms em no mínimo de 95% das medições, pela Agência Nacional de Telecomunicações (Anatel). Este não é um valor limite satisfatório, pois o esperado seria sempre o mais próximo de zero, no entanto valores abaixo de 20 ms são mais aceitáveis. Na Figura 8 fica nítido que o ambiente controlado influencia no valor baixo aferido, entretanto temos uma noção básica do comportamento da rede, onde podemos explorar características específicas de cada tipo de conexão e buscar otimizar deficiências da rede.

Figura 8 - Jitter em média (4 Spine, 6 Leaves)



Fonte: Autoria própria, 2021.

4 RESULTADOS E DISCUSSÕES

Mediante uma arquitetura de rede tradicional percebe-se como os equipamentos de *hardware* de telecomunicações são relevantes para mantê-la em funcionamento, pois comparando-os com a abordagem de *software defined networking* tem-se a flexibilidade implantada diretamente na rede tradicional, onde a decisão não é fornecida pelo equipamento de *hardware* especialista e sim por uma entidade de *software* com limites físicos variáveis que atua como o cérebro da rede. No cenário proposto na seção anterior, tem-se a implantação desta estrutura por intermédio de uma rede emulada, que se faz tão necessária no desenvolvimento de pesquisas na área dado o alto custo dos equipamentos de rede que suportam nativamente a conexão com controladores SDN.

Visto o que a literatura traz, pode-se inferir que os modelos de avaliação em ambientes SDN em sua maioria não seguem um padrão para montagem das configurações de teste, desta forma faz-se necessário seguir orientações específicas como foi abordado por SILVA et al. (2018), onde tomou-se como base a RFC 8456, realizando uma análise comparativa entre cinco dos principais controladores de arquitetura centralizada, POX, Ryu, NOX, Floodlight e Beacon, estudos como este reforçam a compreensão dos limites de cada controlador, sugerem quais aplicações e configurações cada um é capaz de contornar melhor.

O processo de mudança das estruturas básicas de rede para uma arquitetura definida por *software* segue um caminho extenso, no caso deste trabalho foi possível observar os desafios além das vantagens expostas. O fato do ambiente de experimentos não possuir equipamentos dedicados ou até mesmo *hardware* mais robusto nas máquinas, afeta diretamente os resultados finais, desta forma nos experimentos realizados, tiveram que ser normalizados os valores teto das configurações utilizadas com intuito de não afetar o desempenho geral, pois a quantidade de recursos de *hardware* utilizado não eram o bastante para trabalhar com testes intensos e de estresse. Não obstante, foi possível explorar o cenário

dentro dos limites, caso em que não afeta o objetivo em fornecer um cenário possível, *open source*, escalável e flexível, por exemplo, conseguiu-se simular uma rede robusta usada nos experimentos em topologia *spine-leaf*, 16 *spines*, 32 *leaves* e 32 *hosts*; resultando em 1024 links de rede.

Com os experimentos empreendidos sob o cenário com base nos estudos abordados, foi possível observar valores de RTT em torno de 30 ms, valor este que comparado com valores obtidos em estudos como de Islam et al. (2019) que obteve cerca de 100 ms em média de *delay*, o qual afere o atraso entre o envio e recepção do pacote, nota-se então que em uma mesma configuração de ambiente alcançou-se melhoria no tratamento desse tempo, possivelmente associado à atualizações constantes realizadas nos últimos anos. Comportamento análogo é demonstrado nos estudos expostos pelo Journal Jrte (2021) e por Mamushiane et al. (2018), onde o ONOS obteve excelentes valores em termos de taxa de transferência, *delay* e *jitter* pelo Journal Jrte, confirmando a otimização realizada no *software*.

Faz-se necessário salientar o potencial em análises de fluxos por parte do Wireshark, este se mostrou fundamental para a obtenção dos resultados desta pesquisa, por exemplo, em um cenário de alto fluxo de dados como o realizado na Figura 5 obteve-se análise de 88.000 pacotes/segundo.

No geral se obteve resultados satisfatórios em relação a estudos anteriores, infere-se que o ONOS e o OpenDayLight estão em processo de evolução constante e são os principais controladores *open-source* disponíveis com desempenho exemplar, modulares e capacidade *carrier grade*. Salienta-se ainda que, os estudos de avaliação citados fazem uso dos controladores em modo centralizado e estes estão disponíveis em modo *cluster*, distribuído, para que assim a capacidade de cada possa ser integralmente mensurada, este é um objeto de estudos a serem desenvolvidos.

5 CONSIDERAÇÕES FINAIS

De acordo com os resultados obtidos na Seção 3, observamos que o ambiente SDN pode ser trabalhado independente das instalações físicas presentes, este é um alento aos projetos de pesquisas que venham a surgir com intuito de contornar os desafios associados a esta arquitetura de rede que tantas vezes foi citada na literatura como sendo a “arquitetura do futuro”, hoje, é uma arquitetura atual e é possível conhecer diversos casos de sucesso à ela atrelados. Entretanto este processo de mudança está sendo realizado de forma minuciosa e acredita-se que esta mudança acarretará uma arquitetura híbrida, pois em algumas demandas específicas ainda faz-se necessário uma infraestrutura específica, com isso as redes globais, sejam privadas ou públicas, estão evoluindo para atender requisitos emergentes (GTER 46, 2018).

O desenvolvimento de novos protocolos, soluções e otimizações dependem exclusivamente da tratabilidade por parte da comunidade científica, para que assim as pesquisas, experimentos e testes ultrapassem as fronteiras acadêmicas definitivamente e sejam amplamente usufruídas, este é um dos principais objetivos deste trabalho, de fomentar o desenvolvimento de material científico no âmbito de redes definidas por *software*. Todavia, a escolha do cenário de execução: ONOS, Mininet-WiFi, OpenVSwitch, OpenFlow, Iperf3, Wireshark; é apenas uma das possíveis formas de compor um cenário de testes e experimentos que seja seguro, pois os softwares envolvidos são frequentemente mantidos por comunidades open

source, atual, pois de acordo com o que a literatura sugere estes estão em uso, são amplamente testados e têm um futuro promissor, principalmente no caso do ONOS, que possui uma capacidade *Carrier Grade* (Ferreira, 2016) e tem propriedades a ser implantado tanto de forma distribuída quanto centralizada, expondo assim a capacidade de compor soluções em nuvem e juntamente com as diversas soluções de plataformas distribuídas, soluções completas desde a borda da rede ao core e com suporte à 5G, disponíveis na Open Networking Foundation (ONF, 2021).

Como trabalho futuro, pretende-se desenvolver uma avaliação entre os principais controladores SDN disponíveis com capacidade distribuída, tomando como base a RFC 8456 e o estudo realizado por NASCIMENTO (2017), com pretensão em complementar a avaliação quantitativa realizada por SILVA et al. (2018). Com isso, determinar capacidades e possibilidades de uso em IoT.

REFERÊNCIAS

SECCI, Stefano; DIAMANTI, Alessio; SANCHEZ, Jose; *et al.* Security and Performance Comparison of ONOS and ODL controllers. 2019. Disponível em: <https://opennetworking.org/wp-content/uploads/2019/09/ONOSvsODL-report-3.pdf>. Acesso em 20 out. 2021.

CISCO. **Cisco Annual Internet Report (2018-2023)**. 2020. Disponível em: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf>. Acesso em 14 ago. 2021.

ENNE, Antonio José Figueiredo. **Uma Abordagem Abrangente de SDN (Software Defined Networking)**. 2020. Disponível em: http://www.wirelessbrasil.org/antonio_j_f_enne/2020/Art_12/Artigo%2012.pdf. Acesso em 17 set. 2021.

FERREIRA, Caio Cesar. **Análise comparativa de controladores para redes definidas por software de Classe Carrier Grade**. Dissertação (Mestrado) - Universidade Federal de Uberlândia, Uberlândia - MG, 2016. Disponível em: <https://repositorio.ufu.br/handle/123456789/18139>. Acesso em: 20 agosto 2021.

FONTES, Ramon; ROTHENBERG, Christian Esteve. **Emulando Redes sem Fio com Mininet-WiFi**. 1ª Ed., Campinas - SP, 2019.

FLOODLIGHT, Disponível em: <https://github.com/floodlight/floodlight/releases>. Acesso em 13 ago. 2021.

GOLASH, Mukul. **Future Of Networking : SDN and Beyond**. Conference: ICICT-2015. National Journal of Technology, Vol 11, No.1, 2015.

GTER 46: Automação de ACL's na Globo.com. NIC.br. São Paulo, 2018. 1 vídeo (41 min). Publicado pelo canal NICbrvideos. Palestra ministrada por Gustavo Pantuza. Disponível em: <https://www.youtube.com/watch?v=tM8whSZp4rs>. Acesso em: 03 out. 2021.

GOSWAMI, Bhargavi. Experimenting with ONOS Scalability on Software Defined Network. **Journal of Advanced Research in Dynamical and Control Systems**, v. 10, p. 1820–1830, 2019.

HALEPLIDIS, Evangelos; PENTIKOUSIS, Kostas; DENAZIS, Spyros; *et al.* **Software-Defined Networking (SDN): Layers and Architecture Terminology**. [s.l.]: Internet Engineering Task Force, 2015. Disponível em: <https://datatracker.ietf.org/doc/rfc7426>. Acesso em: 21 set. 2021.

IPERF, iPerf - The TCP, UDP and SCTP network bandwidth measurement tool. Disponível em: <https://iperf.fr/>. Acesso em: 25 set. 2021.

ISLAM, Saiful; KHAN, Md; SHORNO, Sanjida; *et al.* **Performance Evaluation of SDN Controllers in Wireless Network**. 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), 2019. Disponível em: https://www.researchgate.net/publication/338071223_Performance_Evaluation_of_SDN_Controllers_in_Wireless_Network . Acesso em: 23 set. 2021.

ITU-R, **IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond**. p. 21, 2015. Disponível em: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-I!!PDF-E.pdf. Acesso em 13 ago. 2021.

IPv6. Disponível em: <http://ipv6.br/>. Acesso em: 11 out. 2021.

JOURNAL, Jrte. **Performance Comparison of ONOS and ODL Controllers on Software Defined Networks Under Different Network Topologies**. v. 2, p. 94–105, 2021. Disponível em: <https://www.jrte.org/2021/07/09/performance-comparison-of-onos-and-odl-controllers-in-software-defined-networks-under-different-network-topologies/>. Acesso em: 1 out. 2021.

MAMUSHIANE, Lusani; LYSKO, Albert; DLAMINI, Sabelo. **A comparative evaluation of the performance of popular SDN controllers**. IEEE 2018 Wireless Days (WD) - Dubai - United Arab Emirates, 2018. Disponível em: https://www.researchgate.net/publication/325294143_A_comparative_evaluation_of_the_performance_of_popular_SDN_controllers . Acesso em 22 set. 2021.

MCKEOWN, Nick; ANDERSON, Tom; BALAKRISHNAN, Hari; *et al.* **OpenFlow: Enabling innovation in campus networks**. Computer Communication Review, v. 38, p. 69–74, 2008. Disponível em: <http://ccr.sigcomm.org/online/files/p69-v38n2n-mckeown.pdf>. Acesso em: 22 set. 2021.

MININET, **Mininet: An Instant Virtual Network on Your Laptop (or Other PC)**. © 2021 - Mininet Project Contributors. Disponível em: <http://mininet.org/>. Acesso em: 25 set. 2021.

MININET-TOPOS, **Mininet Topologies**. Ciena, 2016. Disponível em: <https://github.com/ciena/mininet-topos>. Acesso em: 22 set. 2021.

NASCIMENTO, Tiago da Silva. **Um Catálogo de Métricas para Avaliação da Qualidade de Ambientes Integrados de Computação em Nuvem, Redes Definidas por Software e Virtualização de Funções de Rede**. Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Quixadá - CE, 2017. Disponível em: <http://www.repositorio.ufc.br/handle/riufc/29542>. Acesso em: 22 set. 2021.

ONF, **Open Networking Foundation**. Copyright © 2021 Open Networking Foundation. Disponível em: <https://opennetworking.org/>. Acesso em: 1 out. 2021.

ONOS, **Open Network Operating System - SDN Controller for SDN/NFV Solutions**. Copyright © 2021 Open Networking Foundation. Disponível em: <https://opennetworking.org/onos/>. Acesso em: 03 out. 2021.

OPENDAYLIGHT. Copyright © 2021 OpenDaylight Project The Linux Foundation®. Disponível em: <https://www.opendaylight.org/>. Acesso em: 03 out. 2021.

OPENFLOW, **ONF - OpenFlow Switch Specification**. 2015. Disponível em: <https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf>. Acesso em 13 ago. 2021.

PADILHA, **Cassiano de Mello. Avaliação Experimental de Tecnologias SDN para Implantação em Redes de Produção**. 2017. Trabalho de Conclusão de Curso (graduação) - Universidade de Santa Cruz do Sul, Santa Cruz do Sul - RS, 2017. Disponível em: <http://hdl.handle.net/11624/2134>. Acesso em 10 ago. 2021.

PYTHON, Python.org. Copyright © 2001-2021 Python Software Foundation. Disponível em: <https://www.python.org/>. Acesso em: 1 out. 2021.

Resolução ANATEL N° 574, **Anatel - Resolução n° 574, de 28 de outubro de 2011**. Disponível em: <https://informacoes.anatel.gov.br/legislacao/resolucoes/2011/57-resolucao-574>. Acesso em: 1 out. 2021.

SILVA, Émerson R.; MONTEIRO, Kayo H. C.; DANTAS, Ramide Augusto Sales; *et al.* Avaliação de Controladores Software-Defined Networking Utilizando Metodologia Padrão para Benchmark. *In: Anais do Workshop Pré-IETF (WPIETF)*. [s.l.]: SBC, 2018. Disponível em: <https://sol.sbc.org.br/index.php/wpietf/article/view/3209>. Acesso em: 22 set. 2021.

TR-502, **SDN architecture**, Open Networking Foundation, 2014. Disponível em https://opennetworking.org/wp-content/uploads/2013/02/TR_SDN_ARCH_1.0_06062014.pdf. Acesso em: 1 out. 2021.

VENGAINATHAN, Bhuvaneshwaran; BASIL, Anton; TASSINARI, Mark; *et al.* **Benchmarking Methodology for Software-Defined Networking (SDN) Controller Performance**. [s.l.]: Internet Engineering Task Force, 2018. Disponível em: <https://datatracker.ietf.org/doc/rfc8456>. Acesso em: 21 set. 2021.

WIRESHARK, Wireshark · Go Deep. Disponível em: <https://www.wireshark.org/>. Acesso em: 25 set. 2021.

ZAIDI, Zainab; FRIDERIKOS, Vasilis; YOUSAF, Faqir; *et al.* **Will SDN be part of 5G?** IEEE Communications Surveys & Tutorials, v. PP, 2017. Disponível em: <https://arxiv.org/abs/1708.05096>. Acesso em: 1 out. 2021.

ZHU, Liehuang; KARIM, Md Monjurul; SHARIF, Kashif; *et al.* **SDN Controllers: Benchmarking & Performance Evaluation.** [s.l.: s.n.], 2019. Disponível em: <https://arxiv.org/abs/1902.04491>. Acesso em 1 out. 2021.