UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DEPARTAMENTO ACADÊMICO DE INFORMÁTICA

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

FÁBIO DE SOUZA SILVA

QUALIDADE DE SERVIÇO EM UMA REDE WIFI DEFINIDA POR SOFTWARE

PONTA GROSSA

2022

FÁBIO DE SOUZA SILVA

QUALIDADE DE SERVIÇO EM UMA REDE WIFI DEFINIDA POR SOFTWARE

Quality of Service on a WiFi Network Defined by Software

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Augusto Foronda

PONTA GROSSA

2022

FÁBIO DE SOUZA SILVA

QUALIDADE DE SERVIÇO EM REDE WIFI DEFINIDA POR SOFTWARE

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 18 de maio de 2022

_____

Prof. Dr. Augusto Foronda
Doutorado
Universidade Tecnológica Federal do Paraná

_____

Prof. MSc. Geraldo Ranthum
Mestrado
Universidade Tecnológica Federal do Paraná

_____

Prof. Dr. Lourival Aparecido de Gois
Doutorado
Universidade Tecnológica Federal do Paraná

PONTA GROSSA

2022

Dedico este trabalho à minha família.

AGRADECIMENTOS

Agradeço aos meus pais, pois foram os principais pilares nesta trajetória.

Aos amigos que fiz durante o período de graduação.

Ao meu orientador Prof. Dr. Augusto Foronda, que me guiou neste trabalho.

Gostaria de deixar registrado também, o meu reconhecimento aos professores do curso que me passaram seus conhecimentos e foram um ponto de referência importante para minha futura carreira.

Enfim, a todos os que por algum motivo contribuíram para a realização desta pesquisa.

RESUMO

Qualidade de Serviço (QoS) é uma forma de gerenciamento que possibilita ao usuário da rede priorizar tráfegos de dados. Com a crescente demanda de conexões sem fio, começou-se a pensar em uma nova forma de arquitetura para estas redes, que possa ser eficiente e menos custosa do que as arquiteturas tradicionais e que viabilize QoS. Surge então a arquitetura de redes definidas por software e por consequência redes sem fio definidas por software. Redes definidas por software, do inglês *Software defined networking* (SDN), são redes que tem uma arquitetura onde controladores, com base em software, regem o tráfego de dados. Assim como uma SDN, a rede WiFi definida por software, do inglês *Software defined wireless networking* (SDWN) também usa controladores para gerenciar o tráfego de dados, mas com a diferença de que a rede não é cabeada. Uma das características de uma rede SDWN é facilitar a configuração de QoS na rede através do controlador. Este trabalho tem como objetivo coletar informações de uma SDWN com QoS através do ambiente de simulação Mininet-WiFi, e analisar os dados obtidos para relatar o funcionamento desta tecnologia.

Palavras-chave**:** Redes. WiFi. Arquitetura. SDN. QoS.

ABSTRACT

Quality of Service (QoS) is a form of management that allows the network user to prioritize data traffic. With the growing demand for wireless connectivity, people started to think about a new form of architecture for these networks, which can be efficient and less costly than traditional architectures and which would enable QoS. Then the software defined network (SDN) architecture emerges and consequently the software defined wireless networks (SDWN). Software defined networks are networks that have an architecture where software-based controllers conduct the data traffic. As a SDN, the SDWN also uses controllers to manage data traffic, but with the difference that the network is wireless. One of the characteristics of a SDWN is to facilitate the configuration of QoS on the network through the controller. This paper aims to collect information from a SDWN with QoS through the Mininet-WiFi simulation environment, and analyze these data to report the functioning of this technology.

Keywords**:** Networks. Internet. WiFi. Architecture. Software. QoS.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS, SIGLAS

AP	*Access Point*
API	*Application Programming Interface*
ARP	*Address Resolution Protocol*
BIT	*Binary Digit*
CA	*Collision Avoidance*
CAPWAP	*Control and Provisioning of Wireless Access Points*
CLI	*Command-Line Interface*
CSMA	*Carrier Sence Multiple Access*
GB	Gigabytes
GHZ	Giga-Hertz
ICMP	*Internet Control Message Protocol*
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
IETF	*Internet Engineering Task Force*
IP	*Internet Protocol*
ISM	*Industrial, Scientific and Medical*
LWAPP	*Lightweight Access Point Protocol*
MHZ	Mega-Hertz
PA	Ponto de Acesso
QoS	*Quality of Service*
SDN	*Software Defined Network*
SDWN	*Software Defined Wireless Network*
TCP	*Transmission Control Protocol*
WLAN	*Wireless Local Area Network*

SUMÁRIO

1 INTRODUÇÃO

Uma rede é um conjunto de dispositivos ou nós conectados por enlaces de comunicação. Um nó pode ser um computador, uma impressora ou outro dispositivo de envio e/ou recepção de dados, que estejam conectados a outros nós da rede. As redes de comunicação podem ser: redes cabeadas e WiFi (FOROUZAN, 2008).

Uma rede WiFi tem a sua comunicação sem fios, ou seja, *wireless*. Desta forma, a informação é transmitida na forma de ondas de rádio através do ar. Os dispositivos desta rede estão equipados com uma placa de rede apropriada que contém um transmissor /receptor de rádio que converte pulsos digitais em ondas eletromagnéticas e vice-versa. Essas ondas de rádio são transmitidas em forma de *broadcast* numa dada frequência e qualquer dispositivo sintonizado aquela frequência, num determinado raio, pode receber a mensagem transmitida (XIA, 2014).

As redes sem fio estão presentes em praticamente todos os lugares, isso se deve ao número crescente de dispositivos móveis em circulação. O tráfego móvel representará quase 20% do tráfego IP global em 2022, quase 113 vezes mais que o tráfego móvel global gerado em 2012. No Brasil, o tráfego móvel representará 21% do total do tráfego IP em 2022, em 2017 representava 7% (CISCO, 2020).

Porém, para administrar as redes WiFi cada vez mais sobrecarregadas com um número crescente de tráfego de dados e cada vez mais dispositivos de Internet implantados, está cada vez mais custoso e difícil para as empresas que gerenciam essas redes. Uma das causas é que cada empresa que desenvolve dispositivos w*ireless* tem um padrão diferente das outras, em termos de arquitetura dos dispositivos e manutenção.

O gerenciamento de dispositivos de rede é penoso para os administradores de rede, que sofrem pela falta de interoperabilidade entre as soluções dos diversos fabricantes e por não conseguirem soluções sob medida para suas necessidades (MOURA, 2015). A configuração dos equipamentos de rede e sua instalação requerem técnicos especializados e altamente treinados,

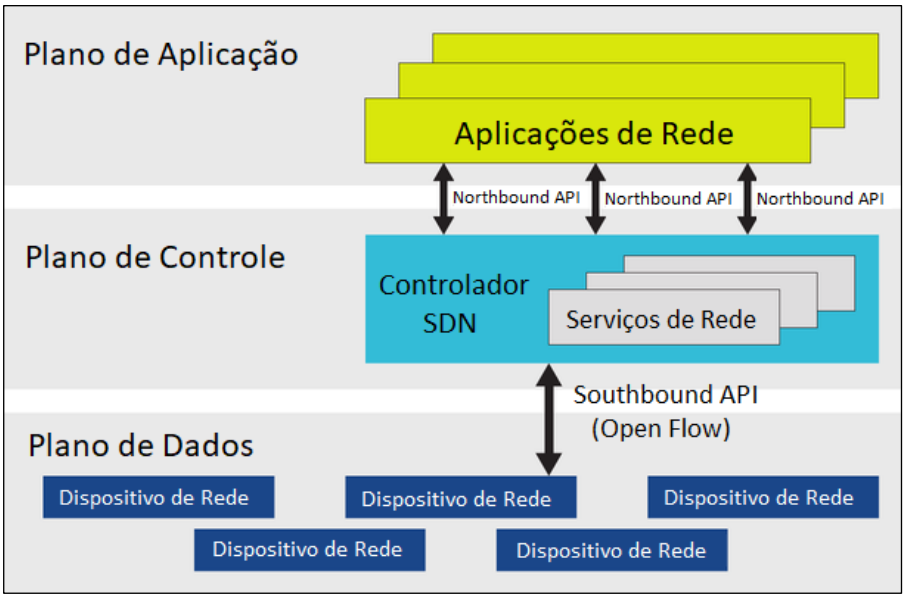
mesmo assim obtêm-se muitas vezes, desempenho moderado em função da complexidade da tarefa (SEZER et al. 2013).

Diante deste problema, foi preciso repensar o modelo de rede que é usado hoje. Então surgiu uma alternativa a esse modelo, que é a rede definida por software (SDN). Uma rede SDN passa a ter um controlador externo (antes associado ao plano de dados) via software. Esse controlador irá gerenciar o fluxo de dados usando APIs que fornecem uma interface entre o plano de controle programável e o plano de dados, bem como uma interface entre as aplicações/ serviços e o plano de controle. Isso traz flexibilidade para as redes de Internet e também torna a manutenção mais fácil e barata (MOURA, 2015).

A arquitetura SDN usa um protocolo de comunicação padrão para estabelecer a ligação entre a camada de controle e a de encaminhamento de dados, conhecido como *OpenFlow,* que é a chave para o gerenciamento de diversos dispositivos de forma única (ANTONIO, 2014).

A Figura 1 ilustra a arquitetura de uma SDN.

Figura 1 - Arquitetura de uma SDN



Fonte: DUQUE (2012)

Dessa forma a arquitetura SDN torna a administração e gerenciamento de redes cabeadas e WiFi mais fácil e consequentemente menos custosa para ser mantida operante. Uma das necessidades de gerenciamento e configuração em uma rede são os algoritmos de QoS que podem priorizar tráfego de dados da rede. QoS é fundamental para que determinados usuários tenham seus

serviços garantidos através da rede, por exemplo, com QoS é possível que todo tráfego de dado de um certo IP da rede pode ser priorizado em relação a outros, esse IP pode ser o de uma Smart Tv que faz *streaming* de filmes por exemplo.

Este trabalho visa investigar e configurar as opções de algoritmos de qualidade de serviço em uma rede WiFi definida por software (SDWN) para que administradores de rede possam usar este serviço caso seja necessário.

1.1 Objetivos

O presente trabalho define os objetivos como objetivos gerais, estabelecidos na subseção 1.1.1 e objetivos específicos, na subseção 1.1.2.

1.1.1 Objetivo Geral

Analisar uma rede SDWN para prover qualidade de serviço.

1.1.2 Objetivos Específicos

Para alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

- Encontrar na bibliografia as instruções de uso do simulador Mininet-WiFi, assim como o entendimento das SDWN;
- Implementar no simulador Mininet-WiFi uma rede SDWN;
- Testar o desempenho da rede implementada;
- Analisar as opções de QoS na rede SDWN;
- Configurar QoS na rede SDWN;
- Analisar os resultados obtidos na rede SDWN com QoS.

1.2 Justificativa

Nota-se que as redes de computadores estão cada vez mais difíceis e custosas de serem administradas. As SDWN são uma alternativa ao modelo de redes que são usadas atualmente, tornando as redes de Internet mais acessíveis e flexíveis, facilitando o uso de tecnologias com QoS.

O estudo da implementação de uma SDWN com QoS ajudará pessoas interessadas no assunto a identificarem suas vantagens ou desvantagens e o funcionamento de fato dessas tecnologias.

2 REFERENCIAL TEÓRICO

Esta seção irá descrever os conceitos teóricos do tema, para compreensão detalhada do que vai ser trabalhado a partir do capítulo 3.

A subseção 2.1 irá detalhar os conceitos básicos sobre a tecnologia de redes sem fio WiFi, assim como um resumo sobre sua origem e suas peculiaridades. A subseção 2.2, aborda os detalhes da nova arquitetura de SDN para que na subseção 2.3 fique mais clara a explicação dos conceitos sobre a arquitetura de SDWN, e então na subseção 2.4 serão detalhadas as funcionalidades do simulador usado no trabalho, o simulador Mininet-WiFi. E na subseção 2.5 serão detalhados QoS e depois na subseção 2.6 será detalhado QoS em SDWN.

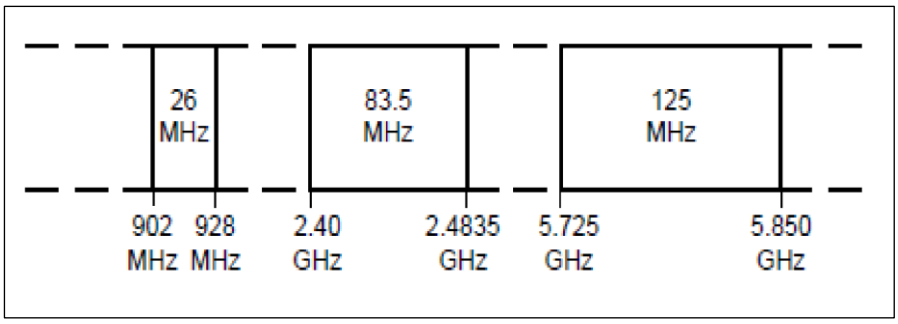
2.1 WiFi

Wireless Fidelity (WiFi) é uma tecnologia para redes sem fio baseada no IEEE 802.11. O termo WiFi vem de *Wireless Fidelity* (Fidelidade sem fio), alusivo ao termo Hi-Fi (*High Fidelity*), usado pela indústria fonográfica nos anos 50. Com o uso do WiFi é possível a transmissão de dados da Internet sem o uso de conexões físicas entre dispositivos, pois, todos os dados são transmitidos através de ondas de rádio que se propagam pelo ar em uma determinada frequência, que eventualmente são agrupadas em bandas, existem outras tecnologias que fazem conexão sem fio, um exemplo é o Bluetooth, que se resume a distâncias curtas.

No Brasil existe o termo “largura de banda” que é comumente associado a velocidade de conexão, porém na prática a largura de banda se refere a quantidade de frequências utilizadas ao mesmo tempo durante uma conexão, possibilitando maior capacidade no tráfego de dados (TANENBAUM, 2003). A frequência do sinal WiFi se mantém no agrupamento de frequências chamado ISM (*Industrial, Scientific, and medical*), que ao contrário de outros agrupamentos não requer licença de autoridades reguladoras, nesse agrupamento existem alguns exemplos conhecidos, um deles é a frequência do sinal do controle remoto (LUCA, 2010). A Figura 2 mostra o espectro de frequências de 900 a 5850 MHz, e também a largura de banda, que é de 26 MHz,

83.5 MHz e 125 MHz para as bandas de 900 MHz, 2400 MHz e 5800 MHz respectivamente.

Figura 2 - Banda ISM



Fonte: LUCA (2019)

Estabelecido pelo *Institute of Eletrical and Eletronics Engineers* (IEEE), o IEEE 802.11 é o padrão de comunicações para redes sem fio mais aceito mundialmente. A tecnologia WiFi, como é mais comumente conhecida, é a tecnologia para *Wireless Local Area Network* (WLAN) baseada no IEEE 802.11, sendo um *trademark* da WiFi Alliance. Os motivos para a larga aceitação desse padrão são diversos, destacando-se, principalmente, a relação custo-desempenho (FONTES et al. 2019). Existem outros padrões 802.11, como por exemplo 802.11g, 802.11n e etc., todos operam principalmente em duas frequências: 2.4 GHz ou 5GHz, a 802.11 opera em 2.4 GHz.

A Tabela 1 compara diferentes padrões de 802.11, apresentando algumas de suas características básicas.

Tabela 1 - Comparação técnica entre diversos padrões IEEE 802.11

Protocolo	Freq. (GHz)	Largura de Banda (MHz)	Cobertura Aproximada Interna	Cobertura Aproximada Externa
802.11	2.4	20	20 m	100 m
802.11a	3.7/5	20	35 m	120 m
802.11b	2.4	20	35 m	140 m
802.11g	2.4	20	38 m	140 m
802.11n	2.4/5	20 - 40	70 m	250 m
802.11ac	5	20/40/80/160	35 m	n/d
802.11ad	60	2,160	60 m	100 m
802.11ay	60	8000	60 m	1000 m

Fonte: FONTES et al. (2019)

2.1.1 Redes Independentes (AD-HOC) e Infraestruturada

Redes móveis podem ser classificadas de duas formas: infraestruturada e independentes (ad-hoc). Rede infraestruturada é quando a comunicação do host móvel se dá sempre com um AP. Mesmo uma comunicação entre dois hosts móveis, que estão a uma distância que permitiria uma eventual comunicação direta, esta deve se dar através do AP. Em redes ad-hoc a comunicação é diretamente entre os hosts móveis, se o destino não estiver ao alcance, requisita-se o serviço de outros hosts móveis vizinhos. Normalmente o host fixo não é considerado, ou é como sendo mais um host móvel (CÂMARA, 1998).

Vantagens de redes ad-hoc sobre redes infraestruturadas:

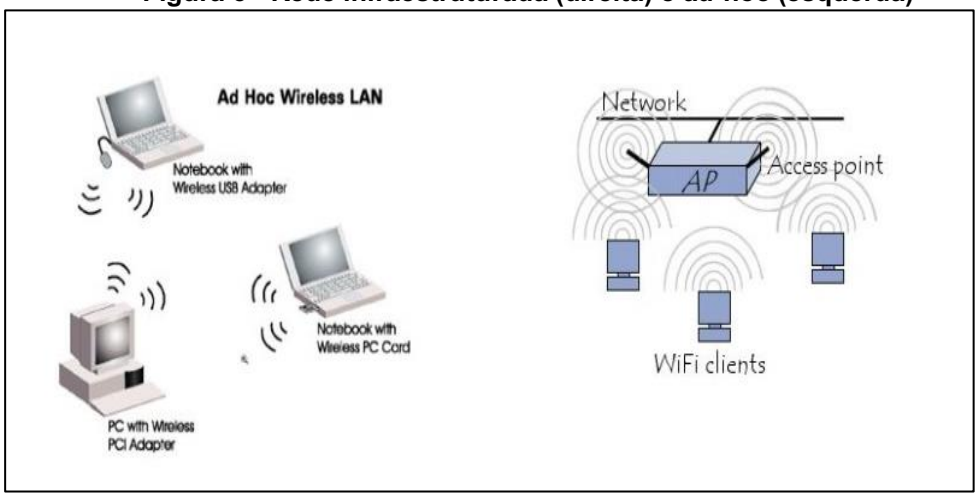
- Redes ad-hoc podem ser instaladas rapidamente em ambientes sem uma infraestrutura prévia;
- Estações com problemas de funcionamento e ou desligadas podem ser rápida e facilmente contornadas, ao contrário de redes fixas ou redes infraestruturadas, se a falha ocorrer no AP;
- Quando duas estações estão a distância em que podem se "ouvir", se têm um canal de comunicação. O que não acontece em redes infraestruturadas.

Desvantagens de redes ad-hoc sobre redes infraestruturadas:

- Banda passante bem menor que em redes infraestruturadas;
- Taxa de erros muito maior;
- Topologia muda constantemente, ao contrário de redes infraestruradas que tem pouca ou nenhuma alteração em um curto espaço de tempo.

A Figura 3 mostra uma representação gráfica de uma rede infraestruturada a esquerda e uma rede ad-hoc a direita.

Figura 3 - Rede infraestruturada (direita) e ad-hoc (esquerda)



Fonte: NETO et al. (2018)

2.1.2 Protocolo CSMA/CA

O CSMA/CA (*Carrier-sense multiple access with collision avoidance*) é uma sigla que significa Múltiplo Acesso por Detecção de Portadora, sendo que o CA é o mecanismo que evita a colisão de pacotes. O CSMA/CA está por trás do protocolo de comunicação, tanto da frequência 2.4 GHz quanto na de 5 GHz. Faz parte do protocolo 802.11, que serve para certificar que o cliente consiga se comunicar com o AP sem que existam colisão de dados. Lembrando que as colisões acontecem quando dois clientes transmitem informações ao mesmo tempo para um único AP (INTELBRAS, 2018).

Em resumo, o CSMA-CA tem as seguintes características:

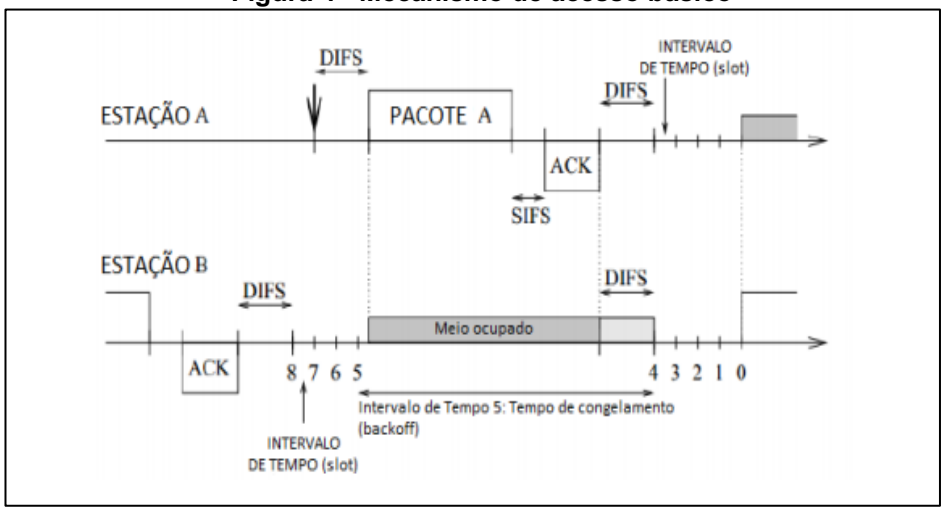
- É desenvolvido para evitar colisões de pacotes de dados;
- Exige que as estações se “escutem”;
- Organiza a transmissão de dados;
- Garante uma informação confiável da informação.

No CSMA/CA, para que um nó possa transmitir dados, primeiramente é verificado se o canal está livre por um período de tempo distribuído. Uma vez ocorrida uma colisão, espera-se por um período de *backoff* (espera) aleatório para tentar transmitir novamente.

Um método padrão para envio de pacotes de dados, denominado de acesso básico, é uma técnica de *handshake* bidirecional. Ela é caracterizada pela transmissão de um bit de confirmação imediata, enviado pela estação destino para a estação fonte, após recepção bem sucedida de um pacote de dados enviado pela estação fonte.

Neste cenário, ao final da transmissão de um pacote, a estação B aguarda por um período e escolhe um *backoff* igual a oito antes de transmitir seu próximo pacote. Supondo que o primeiro pacote da estação A chegue no intervalo indicado pela seta, ele só será transmitido após um período. É possível inferir ainda que o pacote de dados da estação A é transmitido durante o período de congelamento do *backoff* da estação B e isso se deve ao fato da estação B ter detectado que o canal se encontrava ocupado. Contudo, ao verificar que o canal ficou ocioso, o *backoff* voltou a ser decrementado. Considerando que o pacote transmitido pela estação A chegou no tempo previsto, após um período, a estação B transmite seu próximo pacote, conforme pode ser visto na Figura 4 (BIANCHI, 2000).

Figura 4 - Mecanismo de acesso básico



Fonte: Adaptado de BIANCHI (2000)

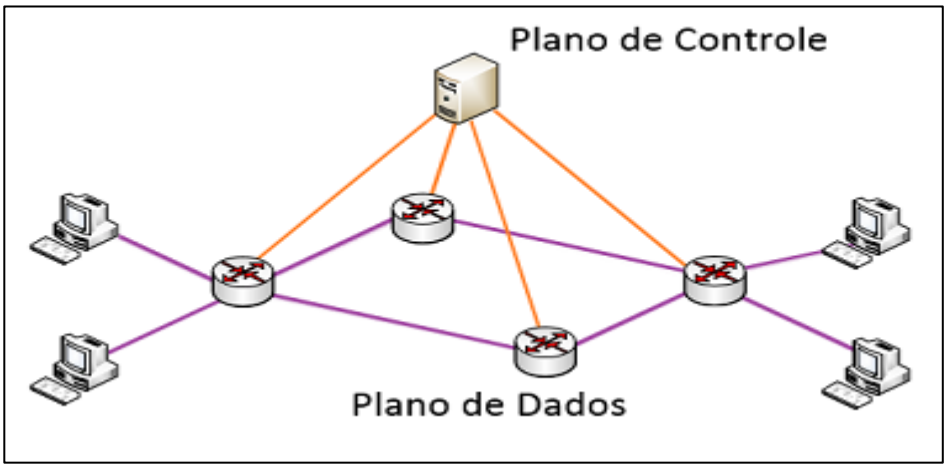
2.2 Redes definidas por software (SDN)

Uma SDN é caracterizada pela existência de encaminhamentos de pacotes de dados de forma personalizada, por meio da separação do plano de controle do plano de dados, centralizando o plano de controle e o gerenciando

via software, tirando essa carga de hardwares comutadores, como por exemplo roteadores, switches, APs e etc., tornando todo o caminho de dados mais fácil de ser administrado e aplicado.

A Figura 5 mostra de maneira simples a arquitetura de uma SDN.

Figura 5 - Arquitetura de uma SDN



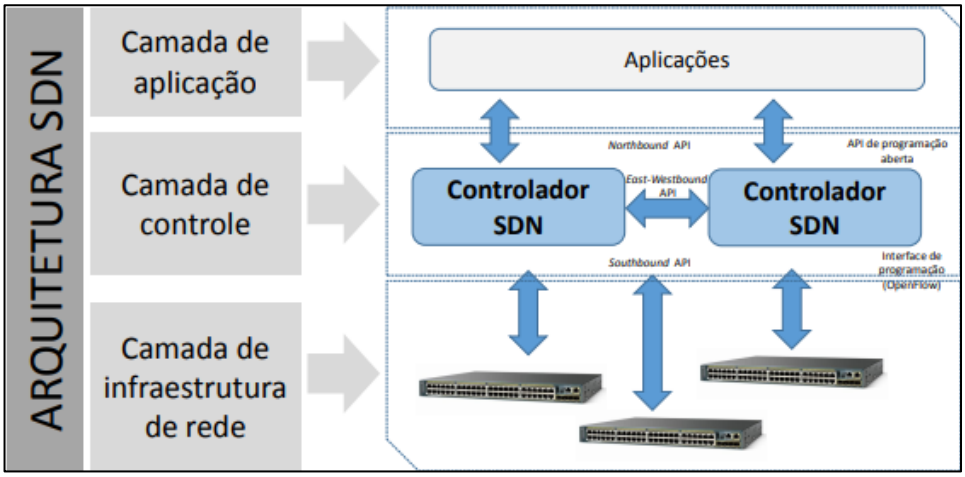
Fonte: LOBATO et al. (2013)

Explicando de maneira análoga o funcionamento de uma SDN, o plano de dados é como uma via para automóveis, onde os automóveis se deslocam de um ponto para outro, assim como os pacotes de dados que transitam no plano de dados em uma rede, e para reger o trânsito de automóveis existem placas, faixas de trânsito, guardas e etc. que ordenam as ruas para que o fluxo de carros ocorra de maneira eficaz, assim é o plano de controle que ordena o fluxo de dados, todas as regras de trânsito são ditadas por gestores, aí pode-se aplicar o software de uma SDN que gera o plano de controle.

Pode-se esquematizar a arquitetura de SDN por três camadas ou planos distintos. Uma camada de aplicação, que consiste nas aplicações de rede que utilizam SDN, uma camada de controle, que fornece controle consolidado da rede e abstrações para a camada de aplicação, e uma camada de dados, onde se encontra a infraestrutura formada pelos equipamentos de rede que encaminham pacotes. Na arquitetura também se encontram APIs ou interfaces que ligam as camadas mais externas com a camada de controle, essas APIs são chamadas de *Southbound* (fronteira sul) e *Northbound* (fronteira norte) (MOURA, 2015).

A Figura 6 mostra a arquitetura de uma SDN com dois controladores e uma interface entre eles, porém esse trabalho foca nas SDN de um controlador, contudo é válido mostrar que existe essa possibilidade.

Figura 6 - Arquitetura de uma SDN detalhada



Fonte: MOURA (2015)

A camada de aplicação, é constituída por aplicações SDN, que são programas que permitem comunicação com o controlador via interfaces *Northbounds*, e nos quais são definidos os requisitos e comportamentos pretendidos para a rede. A camada de controle é constituída por um controlador SDN, responsável pela conversão dos pedidos da camada de aplicação e enviar a informação para a camada de dados. A camada de dados é constituída pelos dispositivos de rede que encaminham o tráfego de dados de acordo com as indicações recebidas do controlador SDN (ANTONIO, 2014).

As interfaces *Northbounds* representam o software que provê interface entre os módulos da plataforma do controlador e as aplicações SDN em execução sobre a plataforma de rede. Estas APIs expõem abstrações do modelo de dados de rede e funcionalidade para uso pelos aplicativos de rede. Normalmente são APIs de código aberto. Como o controlador precisa se comunicar com a infraestrutura de rede, certos protocolos são necessários para controlar e gerenciar estes dispositivos, isso é feito na *Southbound*, as tabelas de encaminhamento dos comutadores são, assim, controladas remotamente pelo controlador (MOURA, 2015).

O *OpenFlow* é a primeira interface de comunicação padrão definida para estabelecer a ligação entre as camadas de controle e encaminhamento da

arquitetura SDN. O *OpenFlow* permite o acesso direto e manipulação dos planos de encaminhamento dos dispositivos de rede, tais como switches e roteadores, tanto físicos como virtualizados. Nenhum outro protocolo padrão permite mover o controle da rede por parte dos switches de rede para um software de controle lógico centralizado.

O protocolo *OpenFlow* é implementado nas interfaces dos dispositivos de rede da infraestrutura e no software da camada de controle de uma SDN. O *OpenFlow* usa o conceito de fluxos para identificar o trafego da rede com base em regras predefinidas que podem ser programadas dinamicamente ou por estatísticas, via software da camada de controle do SDN. Este conceito permite ao administrador da rede definir o modo como o tráfego irá circular pela rede, tendo por base os padrões de uso, aplicações e recursos de serviços de nuvem.

Como o *OpenFlow* permite programar a rede com base em fluxos, uma arquitetura SDN baseada neste protocolo disponibiliza um controle extremamente granular, permitindo à rede adaptar-se em tempo real às exigências das aplicações, utilizadores e sessões. O encaminhamento IP atualmente utilizado não permite este tipo de controle (ANTONIO, 2014).

Um grande trunfo da arquitetura *OpenFlow* é a flexibilidade que ela oferece para se programar de forma independente o tratamento de cada fluxo observado, do ponto de vista de como o mesmo deve ou não ser encaminhado pela rede. Basicamente, o padrão *OpenFlow* determina como um fluxo pode ser definido, as ações que podem ser realizadas para cada pacote pertencente a um fluxo e o protocolo de comunicação entre comutador e controlador, utilizado para realizar alterações dessas definições e ações. A união de uma definição de fluxo e um conjunto de ações forma uma entrada da tabela de fluxos *OpenFlow* [MCKEOWN et al. 2008].

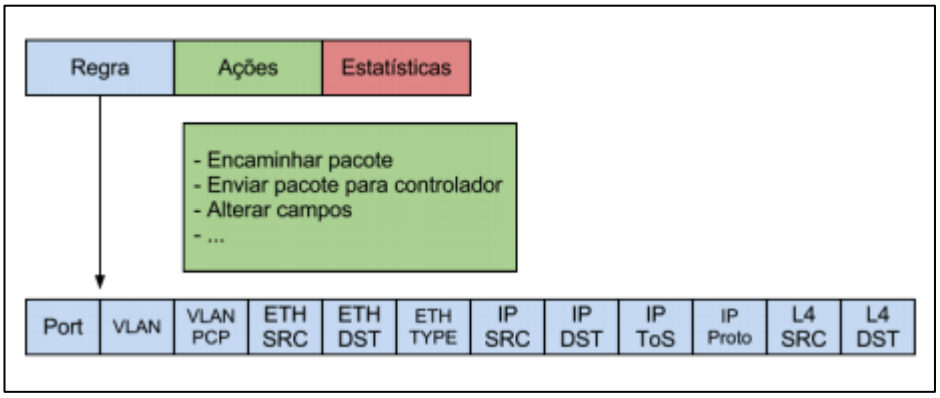
Em um *switch OpenFlow*, cada entrada na tabela de fluxos pode ser implementada como um padrão de bits representado em uma memória TCAM (*Ternary Content-Addressable Memory*). Nesse tipo de memória, bits podem ser representados como zero, um ou “não importa” (*don’t care*), indicando que ambos os valores são aceitáveis naquela posição. Como o padrão e programado a partir do plano de controle, fluxos podem ser definidos da forma escolhida pelo controlador (p.ex., todos os pacotes enviados a partir do endereço físico A para

o endereço físico B, ou todos os pacotes TCP enviados da máquina com endereço IP X para o porto 80 da máquina com endereço IP Y).

A Figura 7 apresenta uma visão geral de uma entrada da tabela *OpenFlow*. Cada pacote que chega a um comutador *OpenFlow* é comparado com cada entrada dessa tabela; caso uma correspondência seja encontrada, considera-se que o pacote pertence aquele fluxo e aplica-se as ações relacionadas a esse fluxo. Caso uma correspondência não seja encontrada, o pacote é encaminhado para o controlador para ser processado — o que pode resultar na criação de uma nova entrada para aquele fluxo. Além das ações, a arquitetura prevê a manutenção de três contadores por fluxo: pacotes, bytes trafegados e duração do fluxo.

Esses contadores são implementados para cada entrada da tabela de fluxos e podem ser acessados pelo controlador através do protocolo (GUEDES, 2014).

Figura 7 - Exemplo de uma entrada na tabela de fluxos OpenFlow



Fonte: GUEDES (2014)

2.3 Redes WiFi definidas por software (SDWN)

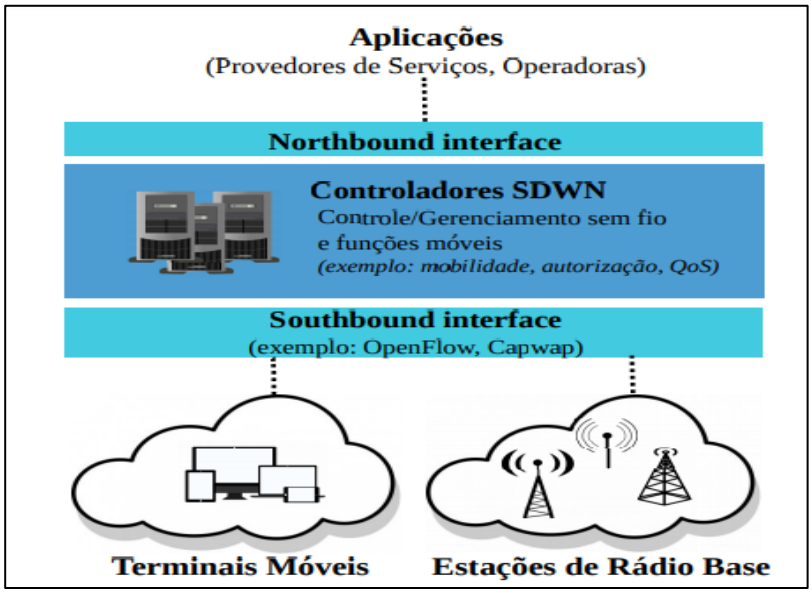
A distribuição de sinal de internet via redes sem fio tem suas peculiaridades em se tratando de SDWN, uma delas é que a necessidade de se ter um controlador central em uma rede sem fio, se deve ao grande número de protocolos de distribuição de rede proprietários, que torna menos simplificado o gerenciamento de uma rede sem fio. A ideia central de uma SDWN é a abertura de um protocolo padrão aberto que vai aumentar a robustez ao gerir uma rede

wireless. Com isso, irá facilitar por exemplo a seleção de melhores canais de transmissão.

SDWN consiste de uma abordagem que permite o controle centralizado da rede através de aplicações que não precisam estar necessariamente localizadas em PAs. Assim, regras definidas através de aplicações mais comumente conhecidas como controladores são quem ditam o comportamento da rede. Os princípios de SDWN são muito similares aqueles definidos para SDN, que consiste na separação do plano de controle e o plano de dados (FONTES et al. 2019).

A Figura 8 mostra a arquitetura genérica de uma SDWN e suas peculiaridades em relação a SDN, como na arquitetura de SDN temos a camada de aplicação, de controle e de dados, cada uma contendo as suas interfaces, sendo elas *Northbound e Southbound*, a diferença com a arquitetura SDN é que o controlador é específico para gerenciamento sem fio e funções móveis, também se tem as estações de transmissão de rádio, que é uma adição na camada de dados que tem na arquitetura SDWN.

Figura 8 - Arquitetura de uma SDWN



Fonte: FONTES et al. (2019)

Em suma, para interfaces *Southbound*, o protocolo mais usado é o *OpenFlow,* porém existem outras soluções como CAPWAP (*Control And*

Provisioning of Wireless Access Points), LWAPP (*Lightweight Access Point Protocol*) e etc.

O LWAPP é um projeto de protocolo da força-tarefa de engenharia da Internet (IETF) que define as mensagens de controle para configuração e autenticação de caminho e operações de tempo de execução. O LWAPP também define o mecanismo de tunelamento para o tráfego de dados. O CAPWAP, que é baseado no LWAPP, é um protocolo padrão e interoperável que permite que uma controladora gerencie uma coleção de PAs wireless. Os PAs com LWAPP habilitado podem descobrir e ingressar em um controlador CAPWAP. A conversão para um controlador CAPWAP é simples. Por exemplo, o processo de descoberta da controladora e o processo de download de firmware quando você usa o CAPWAP são os mesmos de quando você usa o LWAPP. A uma exceção é para as implantações da camada 2, que não são aceitas pelo CAPWAP.

Pode-se implantar controladoras CAPWAP e controladoras LWAPP na mesma rede. O software com CAPWAP habilitado permite que os PAs ingressem ou em um controlador que execute CAPWAP ou LWAPP (FONTES et al. 2019).

2.4 Qualidade de Serviço (QoS)

No contexto de redes em geral, QoS é muito importante para aqueles usuários que desejam que certos tráfegos da rede sejam priorizados em relação a outros, de forma abstrata e no contexto de SDWN, um usuário pode determinar que tráfegos de rede como chamadas de vídeo e jogos online, por exemplo, tenham uma porcentagem maior de velocidade de internet sobre qualquer outro tipo de tráfego que possa atravessar a rede.

Pode-se, então, classificar QoS de acordo com o nível de garantia oferecido (KAMIENSKI, 2000):

- QoS baseado em reserva de recursos, ou rígido, que oferece garantias para cada fluxo individualmente. Esse tipo de QoS é mais complexo (e caro) de implementar.
- QoS baseado em priorização, ou flexível, onde as garantias são para grupos, ou agregação de fluxos. Nesse caso, cada fluxo individual não possui garantias.

O principal objetivo de QoS é fornecer prioridade em relação aos parâmetros de QoS, incluindo, mas não limitado a (KARAKUS, 2017):

- Largura de banda: é a média ou a quantidade de entrega de pacotes que é possível fazer ao final de um período de tempo;
- Atraso: é o tempo entre o envio de uma mensagem de uma determinada origem e a recepção desta mensagem ao destino;
- Jitter: é a variação no atraso de pacotes em determinado tempo;
- Perda de pacotes: é a porcentagem de pacotes enviados que não chegaram ao destino, devido a algum problema.

2.5 QoS em redes SDWN

O conceito de QoS se enquadra perfeitamente na questão de regras *OpenFlow* e SDWN, pois são essas regras, como criação de fluxos, que irão priorizar tráfegos em relação a outros. No contexto de QoS essas regras são tratadas com o nome de *meter table* ou tabela medidor.

Uma rede SDN pode prover QoS através de duas ferramentas: Medidores (*Meters*) e Filas (*Queues*). Medidor é um mecanismo usado na entrada da rede para limitar a taxa de dados e fila é usada na saída da rede onde podem ser configuradas várias filas para separar tráfego (BOLEY, 2016).

A tabela medidora foi introduzida no *OpenFlow* 1.3 como um novo recurso para prover QoS. Ela permite monitorar a taxa de ingresso de um fluxo e, em seguida, executa as operações com base na taxa do fluxo. O medidor é conectado às entradas de fluxo e é uma propriedade de uma porta de switch. Uma tabela de medidores é composta por um identificador medidor, bandas de medidor e contador. Quando a taxa do fluxo é maior do que a taxa da banda especificada, a operação especificada no tipo de banda será realizada para o fluxo. Existem dois tipos de operação de banda que definem como os pacotes são processados: “drop” que descarta pacotes que excedem o especificado na taxa da banda e é parecido com o min_rate de uma fila, e também se tem o “dscp remark”, que aumenta a precedência do campo DSCP (*Differentiated Services Code Point*) no cabeçalho IP do pacote. O DSCP se refere a um conjunto de valores de QoS e estes valores podem ser utilizados para priorizar o tráfego,

onde terá três níveis de prioridade e quatro classes de serviço (PRADEEP, 2017).

A tabela medidora é um complemento da fila, sendo que a fila *OpenFlow* pode garantir uma taxa mínima para um fluxo usando modelagem de tráfego e policiamento em switches, enquanto que com a tabela medidor não é possível fazer isso (KRISHNA, 2016).

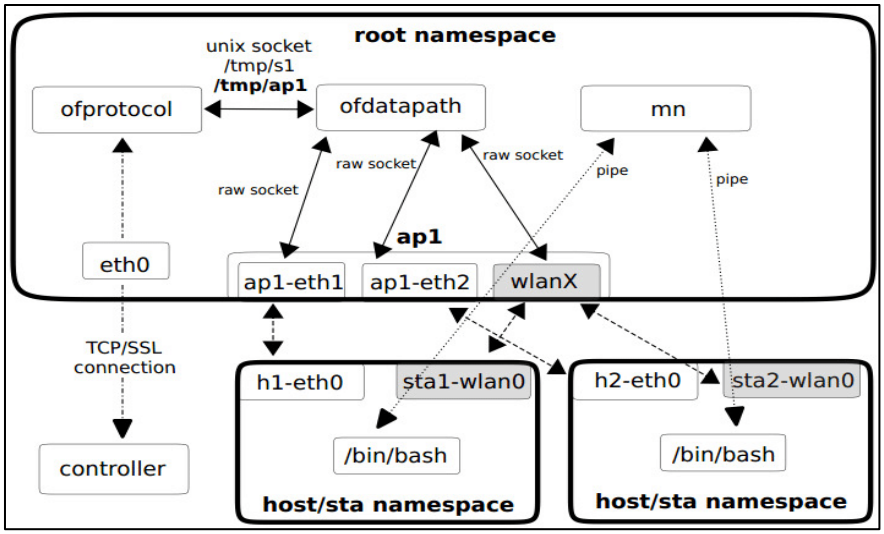
2.6 Mininet-WiFi

Para realização desse trabalho, foi usada como ferramenta de simulação da SDWN, o simulador Mininet-WiFi desenvolvido na Unicamp-SP, usado para simulação e testes na área de redes sem fio. Esse simulador é uma extensão do simulador Mininet, que é um simulador bastante conhecido na área de redes (FONTES et al. 2019).

No Mininet-WiFi é possível emular e virtualizar PAs, hosts, switches e controladores, por isso consequentemente o Mininet-WiFi é bastante viável para a implementação da SDWN, que é o objetivo desse trabalho.

A base do processo de virtualização de uma rede sem fio no Mininet-WiFi constitui dos processos executados em Linux Network Namespaces e placas de redes virtuais, esse processo gera uma simulação de um computador real com todas as propriedades de rede que uma máquina real possui. A Figura 9 mostra a arquitetura base do simulador.

Figura 9 - Arquitetura do simulador Mininet-WiFi



Fonte: FONTES et al. (2019)

3 DESENVOLVIMENTO

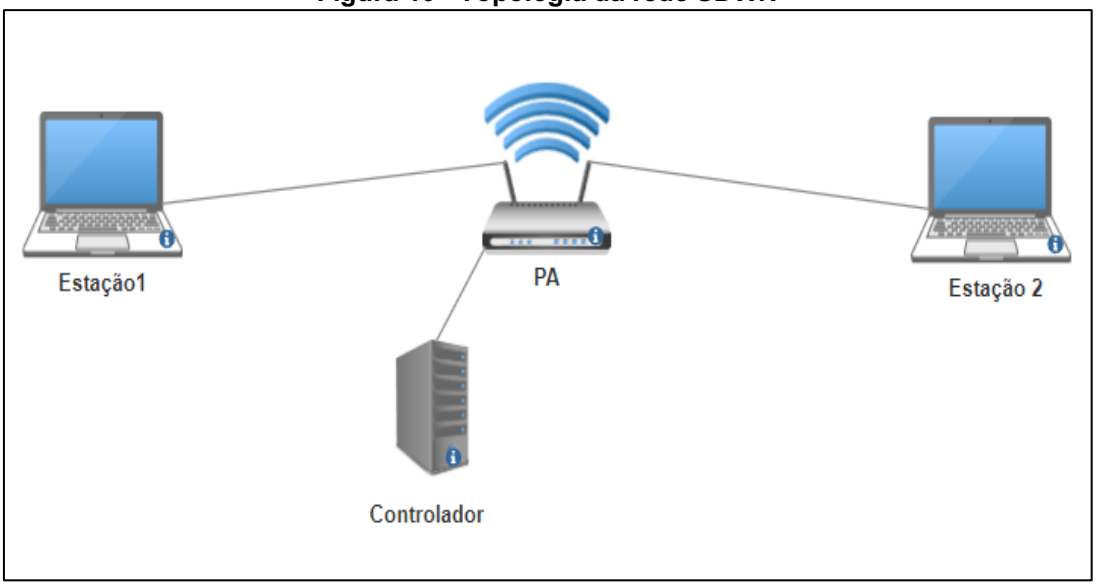
Neste capítulo é mostrado o funcionamento de uma rede SDWN em conjunto com o protocolo *OpenFlow* através do simulador Mininet-WiFi e de outras ferramentas de rede encontradas no sistema Linux, assim como um caso de uso de QoS para redes SDWN.

3.1 Preparação do ambiente de simulação

A realização da simulação da rede SDWN no mininet-WiFi foi feita no sistema operacional Ubuntu 18.04 LTS rodando em máquina virtual. O computador usado foi um Acer Aspire 3 com processador AMD Ryzen 5 3500 U with Radeon Vega Mobile Gfx 2.10 Ghz e 8 GB de memória RAM. A versão do Mininet-WiFi usada foi a 2.6.

A topologia usada na simulação consiste em duas estações conectadas a um PA através de um meio sem fio, e também um controlador SDN conectado ao PA, como mostra a Figura 10.

Figura 10 - Topologia da rede SDWN



Fonte: Autoria própria (2022)

Após a inicialização do Mininet-WiFi com essa topologia o ambiente de simulação está pronto para as análises com a rede SDWN.

3.1.1 Configuração do Ponto de Acesso e Fluxos

Para iniciar os testes foi necessário instruir o PA sobre o que fazer com os pacotes de dados que tem a mesma interface de rede como entrada e saída de dados, pois no protocolo *OpenFlow* o controlador demanda que se descarte qualquer pacote de dados que se comportam dessa forma, pois o protocolo foi criado para ser usado em redes cabeadas, e em uma rede sem fio todos os pacotes de dados entram e saem pela mesma interface, a interface sem fio, assim seria impossível ter comunicação entre a estação 1 e a estação 2 se não fosse feita essa modificação na configuração do simulador.

Para instruir o PA sobre as ações tomadas quando houverem pacotes de dados que entram e saem pela mesma interface de rede, é necessário criar fluxos que permitem fazer o PA agir da forma requerida. Criando esses fluxos, o PA não irá requerer instruções do controlador sobre o que fazer com os pacotes de dados, pois as regras já serão estaticamente estabelecidas.

Fluxos são regras que podem ser estaticamente criadas para que os componentes de uma rede se comportem de uma certa forma, esse processo é bastante similar à inserção de rotas estáticas em roteadores quando existem protocolos de roteamento em uso. No caso dos testes, foi necessário instruir ao PA que todo pacote ARP e ICMP que entrar pela interface de rede, devem sair por ela mesma. Para adicionar os fluxos ao PA, foram utilizados os comandos mostrados na Figura 11, na CLI do simulador.

Figura 11 - Adição de Fluxos

```
sh ovs-ofctl add-flow ap1 "priority=0,arp,in_port=1, actions=output:in_port“

sh ovs-ofctl add-flow ap1 "priority=0,icmp,in_port=1, actions=output:in_port"
```

Fonte: Autoria própria (2022)

Onde o sh é uma palavra que sempre deve ser utilizada dentro da CLI do Mininet-WiFi quando for necessário utilizar um comando do sistema operacional

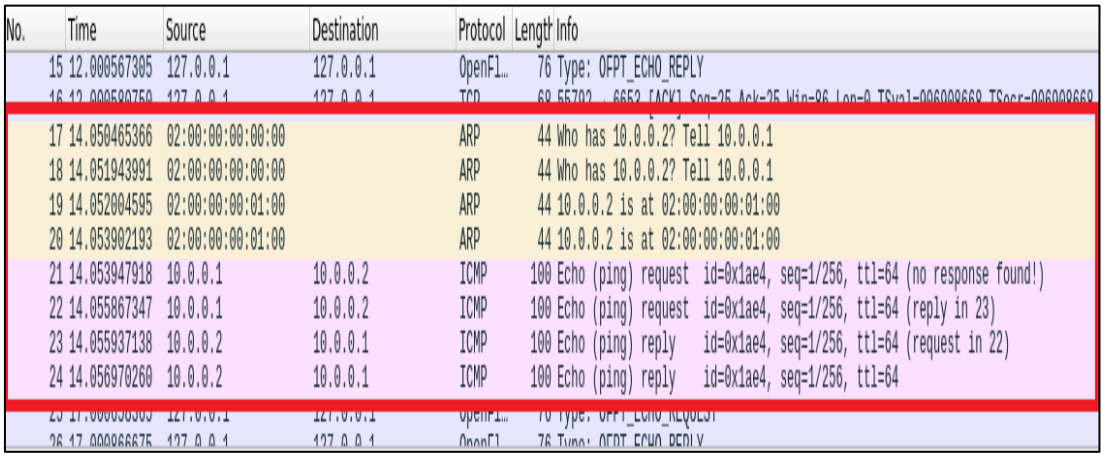
dentro da interface do simulador, que é o comando ovs-ofctl, que pode ser usado tanto para visualizar ou adicionar fluxos. Na simulação os comandos ovs-ofcvtl estão adicionando fluxos, que instruem o PA cujo nome na rede é ap1 que todo pacote arp e icmp que entrar pela interface 1 devem sair por ela mesma.

3.2 Simulação e análise de uma rede SDWN com Openflow

Além do simulador Mininet-WiFi, foi usada a ferramenta *Wireshark*[1], que possibilitou verificar as mensagens envolvidas na rede sem fio.

Com todos os requisitos preparados, foi feito um ping da estação 1 para a estação 2, a Figura 12 mostra as mensagens registradas nesse processo.

Figura 12 - Mensagens da rede SDWN



No.	Time	Source	Destination	Protocol	Length	Info
15	12.000567305	127.0.0.1	127.0.0.1	OpenFl…	76	Type: OFPT_ECHO_REPLY
17	14.050465366	02:00:00:00:00:00		ARP	44	Who has 10.0.0.2? Tell 10.0.0.1
18	14.051943991	02:00:00:00:00:00		ARP	44	Who has 10.0.0.2? Tell 10.0.0.1
19	14.052004595	02:00:00:00:01:00		ARP	44	10.0.0.2 is at 02:00:00:00:01:00
20	14.053902193	02:00:00:00:01:00		ARP	44	10.0.0.2 is at 02:00:00:00:01:00
21	14.053947918	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0x1ae4, seq=1/256, ttl=64 (no response found!)
22	14.055867347	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0x1ae4, seq=1/256, ttl=64 (reply in 23)
23	14.055937138	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0x1ae4, seq=1/256, ttl=64 (request in 22)
24	14.056970260	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0x1ae4, seq=1/256, ttl=64

Fonte: Autoria própria (2022)

Na Figura 12, na sequência 17 pode-se identificar as mensagens ARP sendo enviadas pela estação 1 com identificação 10.0.0.1 procurando a estação 2 com identificação 10.0.0.2 nas portas do PA, a seguir podem ser visto as mensagens ICMP sendo trocadas entre estação 1 e estação 2.

No entanto, não se vê a participação do protocolo *OpenFlow*. Como dito anteriormente, o PA não envia mensagem ao controlador pois já haviam sido estabelecidas as regras de comunicação quando foram criados os fluxos que instruíam o PA a encaminhar as mensagens na interface de rede.

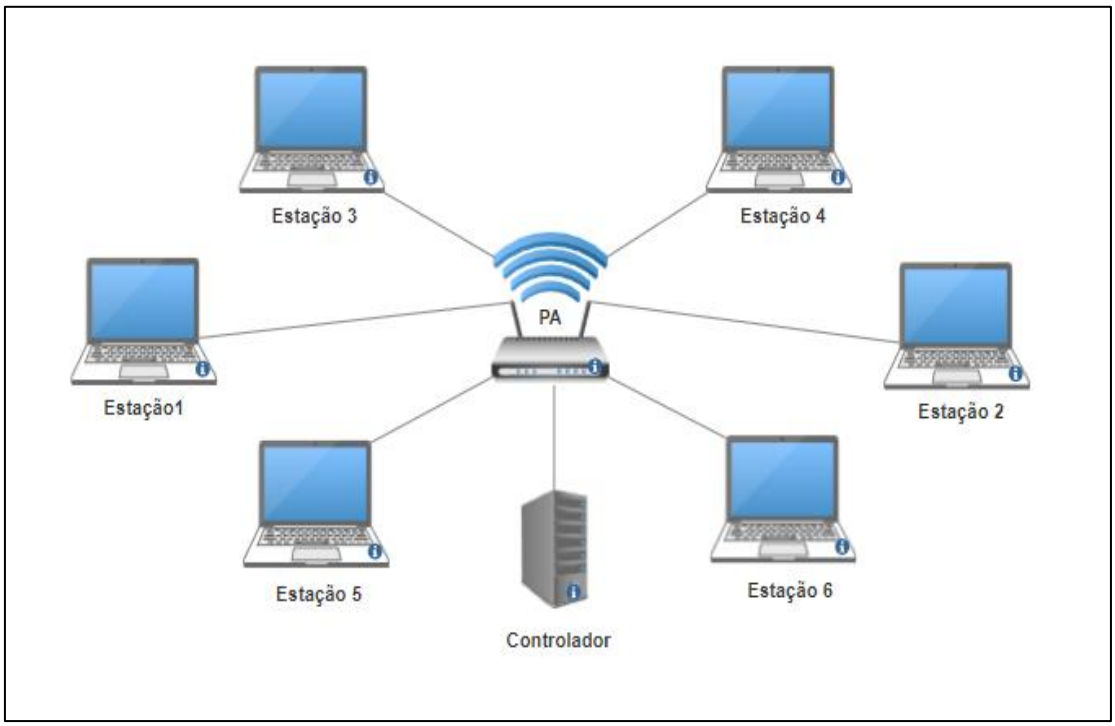
1 https://www.wireshark.org/

Contudo o controlador, está presente nesta rede, e pode ser usado para outras funções como o controle do QoS da rede.

É importante notar que, como em uma rede sem fio com um PA, o trafego de dados ocorre na mesma interface de rede, os comandos de criação de fluxo são os mesmos para qualquer número de estações que uma topologia possa ter.

Para fins de teste, também foi criada uma topologia similar à da primeira simulação, com a diferença de que para esse teste foram inseridas mais 4 estações, totalizando 6 estações como mostra na Figura 13, posteriormente o comando usado para a criação dos fluxos foi o mesmo usado na primeira simulação.

Figura 13 - Topologia com 6 estações



Fonte: Autoria própria (2022)

Para finalizar o teste foram feitos pings entre as estações 2 e 5, 4 e 1, e 6 e 2 e o resultado dos pings é mostrado na Figura 14 com sucesso.

Figura 14 - Pings entre múltiplas estações, com fluxo estático

```
mininet-wifi> sta2 ping -c1 sta5
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
64 bytes de 10.0.0.5: icmp_seq=1 ttl=64 tempo=4.47 ms

--- 10.0.0.5 estatísticas de ping ---
1 pacotes transmitidos, 1 recebidos, 0% perda de pacote, tempo 0ms
rtt min/avg/max/mdev = 4.471/4.471/4.471/0.000 ms
mininet-wifi> sta4 ping -c1 sta1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes de 10.0.0.1: icmp_seq=1 ttl=64 tempo=3.90 ms

--- 10.0.0.1 estatísticas de ping ---
1 pacotes transmitidos, 1 recebidos, 0% perda de pacote, tempo 0ms
rtt min/avg/max/mdev = 3.902/3.902/3.902/0.000 ms
mininet-wifi> sta6 ping -c1 sta2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes de 10.0.0.2: icmp_seq=1 ttl=64 tempo=3.23 ms

--- 10.0.0.2 estatísticas de ping ---
1 pacotes transmitidos, 1 recebidos, 0% perda de pacote, tempo 0ms
rtt min/avg/max/mdev = 3.231/3.231/3.231/0.000 ms
```

Fonte: Autoria própria (2022)

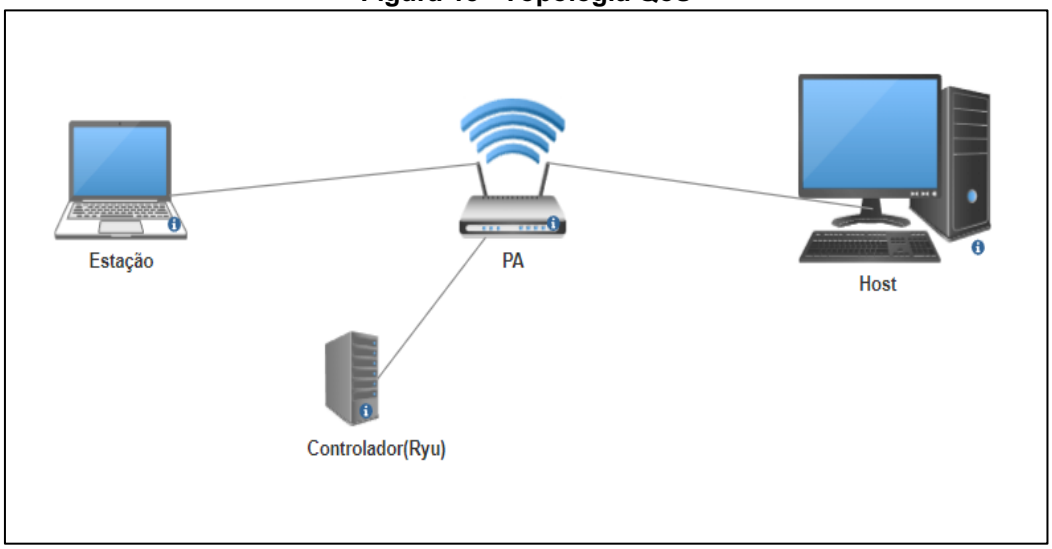
3.3 QoS em redes SDWN

Até então foi mostrado como funciona uma rede SDWN e suas particularidades, agora será abordada a questão de QoS, cujo funcionamento está diretamente ligado a abordagem SDWN e *OpenFlow*.

3.3.1 Configuração de QoS

Nesta subseção é abordada a questão de como regras *OpenFlow* podem ser definidas no contexto de QoS, para tanto, foi feito um teste usando uma topologia simples que possui uma estação, um ponto de acesso e um host, conforme ilustrado na Figura 15.

Figura 15 - Topologia QoS



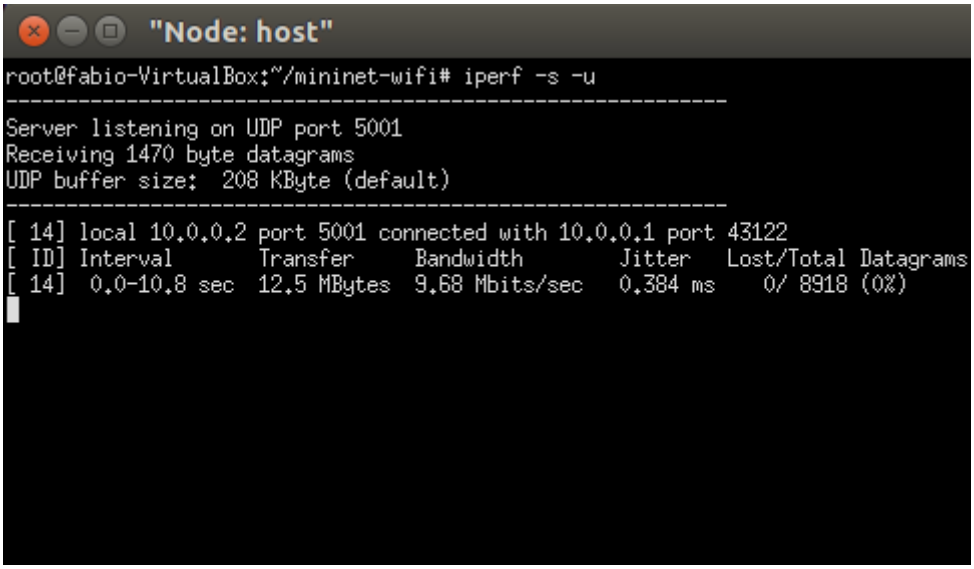
Fonte: Autoria própria (2022)

Esta topologia tem uma largura de banda máxima de 10 Mbps entre os dispositivos e o PA. E o objetivo do teste foi adicionar regras no PA através do controlador que o instruíram a limitar a largura de banda em até 5 Mbps entre a Estação e o Host. Assim estes dispositivos finais teriam esta banda garantida, e o resto da banda (5Mbps) poderia ser usada por outros dispositivos que entrassem na rede.

Para o teste foi usado, em um terminal externo, o controlador Ryu, que é um controlador SDN que suporta o protocolo *OpenFlow*.

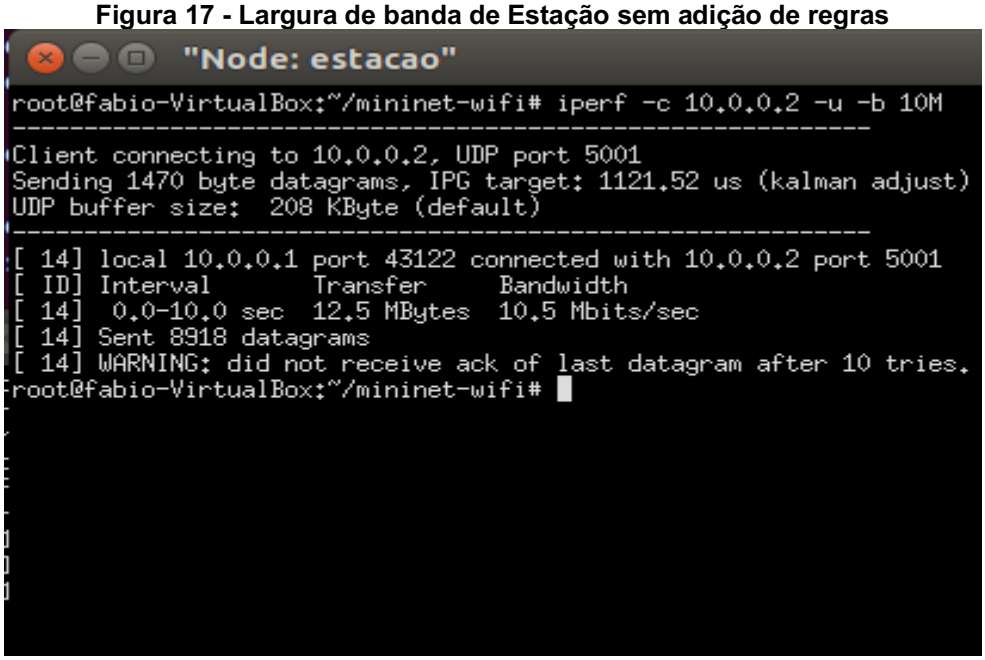
O primeiro teste foi enviar pacotes UDP entre o host e a estação, onde o host foi usado como servidor (Figura 16) e estação como cliente (Figura 17) sem a adição de regras de QoS. Ou seja, o objetivo é verificar o comportamento da rede sem QoS. A taxa de transmissão usada foi 10Mbps.

Figura 16 - Largura de banda de Host sem adição de regras



Fonte: Autoria própria (2022)

Figura 17 - Largura de banda de Estação sem adição de regras



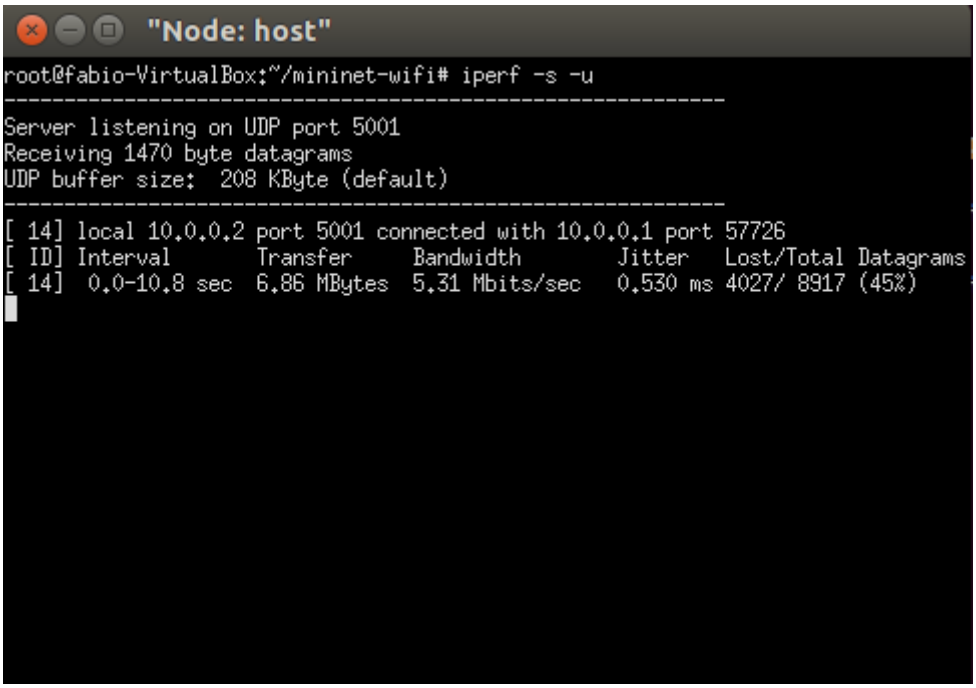
Fonte: Autoria própria (2022)

É observado que o cliente transferiu dados em uma largura de banda no valor de 10,5 Mbps e o servidor recebeu 9,68 Mbps. Ou seja, aqui pode-se observar que o servidor foi capaz de receber praticamente a quantidade inteira de dados enviados pela estação. Dessa maneira, se houvessem outros usuários na rede, a mesma já estaria congestionada e haveria perda de pacotes.

No segundo teste, foi mensurada a largura de banda na mesma topologia, com a diferença de que tiveram adições de regras que fizeram o PA limitar a

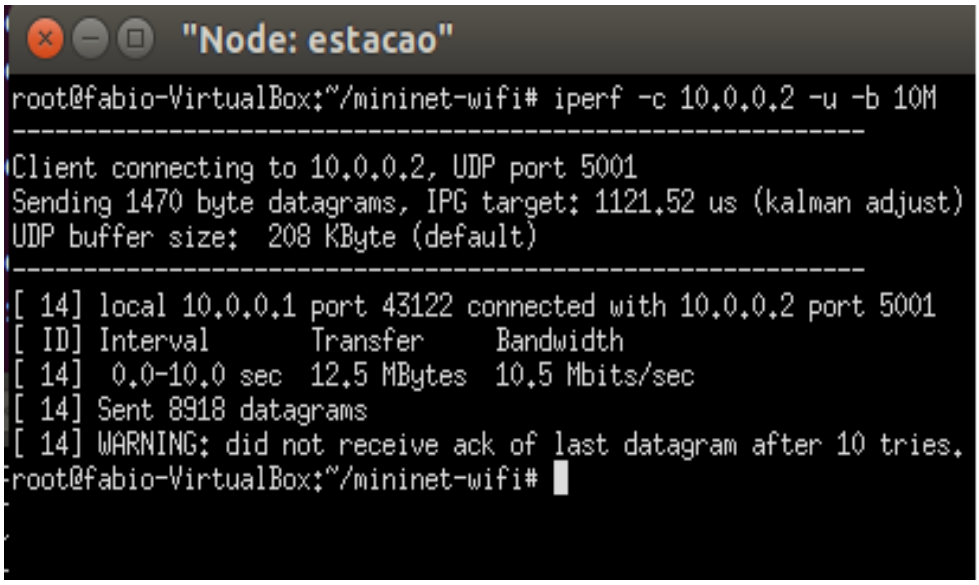
largura de banda em até 5 Mbps entre o Host e a Estação. Os resultados podem ser vistos na Figura 18 e 19.

Figura 18 - Largura de banda de Host com adição de regras



Fonte: Autoria própria (2022)

Figura 19 - Largura de banda de Estação com adição de regras



Fonte: Autoria própria (2022)

Pode-se perceber que o cliente novamente transferiu dados no valor de 10,5 Mbps (Figura 19). Porém, como foi configurado um limitador de banda (meter) de 5 Mbps, quase metade dos pacotes foram descartados (45%) e a

banda ocupada foi de 5,31 Mbps, como pode ser observado na Figura 18. Isso ocorreu devido a regras *OpenFlow*, mostradas na Figura 20, definidas para o PA.

Figura 20 - Adição de regras limitantes

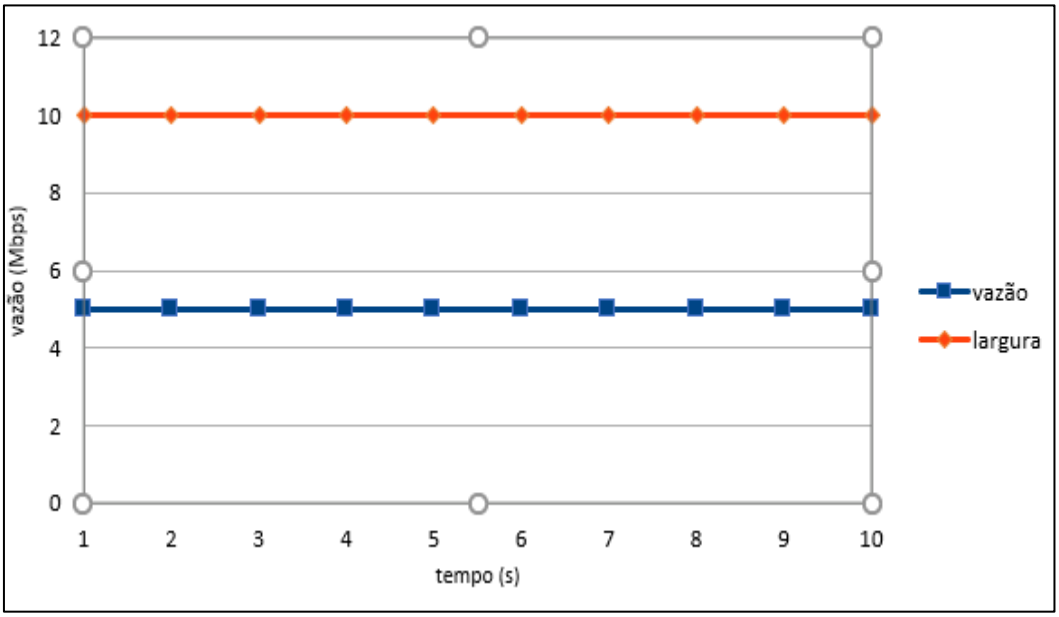
```
ovs-ofctl -O OpenFlow13 add-meter ap1 ‘meter=1,kbps,bands=type=drop,rate=
5000’

ovs-ofctl -O OpenFlow13 add-flow ap1 'priority=1,in_port=1 action=meter:1,2'

ovs-ofctl -O OpenFlow13 add-flow ap1 'priority=1,in_port=2 action=meter:1,1'
```

Fonte: Autoria própria (2022)

A primeira linha de código criou a regra *OpenFlow* no PA, aqui chamado de ap1, e fez com que a vazão da banda larga diminua em até 50%. A segunda linha adicionou a regra na porta 1 do PA, e a terceira e última linha adicionou a regra na porta 2.

A relação banda larga e vazão, são mostradas no Gráfico 1 abaixo.

Gráfico 1 - Relação banda larga e vazão



Fonte: Autoria própria (2022)

O gráfico explicita que no teste onde houve adições de regras *OpenFlow*, o tráfego entre Estação e Host será sempre limitado em valores próximos a 5 Mbits/s, mesmo tendo largura de bando próximo a 10 Mbits/s, em consequência das regras impostas no PA.

Neste teste foram utilizadas regras simples para limitação de largura de banda, elas podem ser customizadas de forma a impor diferentes valores de largura de banda para diferentes tipos de tráfegos, além da imposição de prioridades de diferentes perfis de tráfegos sobre outros.

4 CONCLUSÃO

Com a elaboração deste trabalho pode-se concluir que SDWN expande as possibilidades de gerenciamento de rede em comparação às arquiteturas de redes tradicionais, além disso torna a execução de QoS mais simples através de algoritmos junto ao controlador.

Após a finalização desse trabalho, foi verificado que o gerenciamento de uma rede de Internet com arquitetura SDN ou SDWN tornou-se mais simples e ágil, devido a abrangência do protocolo *OpenFlow* e a separação da camada física com a camada de dados da rede. Em relação ao QoS em uma rede SDWN, foi constatado que o mecanismo utilizado neste trabalho para configurar QoS na rede é o único utilizado atualmente para redes definidas por software, mas existem outros mecanismos de QoS em redes tradicionais que seriam interessantes serem implementados em uma rede SDWN, talvez isso se deve ao fato da arquitetura de rede definida por software ser relativamente nova.

Foi desenvolvido neste trabalho uma simulação de rede SDWN com QoS com base nos objetivos específicos propostos. Primeiramente foi realizado a pesquisa sobre o simulador Mininet-WiFi, redes SDN e SDWN e QoS, após isso foram feitos testes com rede SDWN em duas topologias diferentes, e depois um teste com SDWN com QoS em uma topologia, as topologias foram configuradas com regras *OpenFlow* para que ocorresse o funcionamento esperado, como por exemplo a comunicação entre as estações na topologia 1 do teste com rede SDWN sem QoS.

Por fim, foram feitos testes de comunicação nas redes SDWN sem QoS, e nas redes com QoS foram realizados testes de largura de banda. Em todos os testes os resultados foram satisfatórios.

O projeto proposto abordou a implementação de uma rede SDWN com QoS. Através do simulador Mininet-WiFi foi visto como funciona a implementação de uma rede SDWN com QoS e também, após implementado, foram levantados os dados da rede para análise de funcionamento.

O ambiente de simulação usado neste trabalho, foi o Mininet-WiFi da Unicamp. Com o Mininet-WiFi foi possível a virtualização de estações e pontos de acessos, e também a utilização de nós como *Switches* e controladores

OpenFlow, assim como outras soluções importantes para o desenvolvimento do trabalho.

REFERÊNCIAS

ANTONIO, F. G. **Multiflow WiFi utilizando Software Defined Networking**, 49 f. 2014. Dissertação (Mestrado) – Mestrado em Engenharia Informática, Universidade de Coimbra, 2014.

BIANCHI, G. **Performance analysis of the IEEE 802.11 distributed coordination function**. IEEE Journal on Selected Areas in Communications, [S.l.], v.18, n.3, p.535–547, 2000.

BOLEY, J. M. et al. Adaptive QoS for Data Transfers using Software-Defined Networking. In: 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), 2016, Bangalore, India. **Anais [...]** Bangalore: IEEE, 2016.

CISCO. **Cisco Annual Internet Report (2018 - 2023) White Paper**. [*S. l.*], 2020. Disponível em: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html. Acesso em: 20 mar. 2021.

DUQUE, D. H. **Redes Definidas por Software**. 2012. Disponível em: https://www.monografias.com/pt/trabalhos3/redes-definidas-software/redes-definidas-software2.shtml. Acesso em: 11 abr. 2021.

FONTES, R.; ROTHENBERG, C. **Emulando Redes sem fio com Mininet-WiFi**. 1. ed. Campinas, 2019.

FOROUZAN, B. A. **Comunicação de Dados e Redes de Computadores**. 4. ed. Porto Alegre: AMGH, 2008

KAMIENSKI, C. A.; SADOK, D. **Qualidade de Serviço na Internet.** 1. Ed. Belo Horizonte, 2000.

KARAKUS, M.; DURRESI, A. Quality of Service (QoS) in Software Defined Networking (SDN): A survey. **Journal of Network and Computer Applications.** v. 80, p. 200-218, 2017.

KRISHNA, H. Providing End-to-end Bandwidth Guarantees with OpenFlow. 2016. Thesis (Master of Science) – Delft University of Technology, Países Baixos, 2016.

LOBATO, A.; FIGUEIREDO, U.; ALVES, L. **Redes definidas por software**. 2013. Disponível em: https://www.gta.ufrj.br/grad/13_1/sdn/definicao.html. Acesso em: 01 mai. 2021

LUCA, VICENTE. **Implantação e gerenciamento de uma rede sem fio nos domínios de um campus universitário**, 86 f. 2010. – Bacharelado em Ciência da Computação, Universidade Federal de Lavras, Minas Gerais, 2010.

MOURA, D. M. **Ethanol: Uma plataforma SDN para redes Wi-Fi**, 162 f. 2015. Dissertação (Mestrado) – Mestrado em Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, 2015.

NETO, A.; ROCHA, I.; MATTOS, P. **Redes veiculares**. 2018. Disponível em: https://www.gta.ufrj.br/ensino/eel878/ redes1-2018-1/ trabalhos-vf/veiculares/. Acesso em: 01 mai. 2021

PRADEEP, J. End-to-end Quality-of-Service in Software Defined Networking. 2017. Dissertation (Master of Science in Computer Science) – University of Dublin, Trinity College, Dublin, 2017.

TANEMBAUM, ANDREW. **Computer networks**. Prentice Hall PTR, 4 ed. edição, 2003.

XIA, H. **Controlo de transmissão em redes WiFi para situações de Sobrecarga**, 70 f. 2014. Dissertação (Mestrado) – Mestrado integrado em Engenharia Eletrotécnica e de Computadores, Universidade do Porto, 2014.