INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
IFCE CAMPUS ARACATI
COORDENADORIA DE CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

IVYSON LUCAS DE LIMA SILVA

UMA ANÁLISE DA SEGURANÇA EM REDES DEFINIDAS POR SOFTWARE BASEADAS NO OPENFLOW

ARACATI-CE
2019

IVYSON LUCAS DE LIMA SILVA

UMA ANÁLISE DA SEGURANÇA EM REDES DEFINIDAS POR SOFTWARE BASEADAS NO OPENFLOW

Trabalho de Conclusão de Curso (TCC) apresentado ao curso de Bacharelado em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE - Campus Aracati, como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação.

Orientador (a): Prof. Dr. Reinaldo Bezerra Braga

Aracati-CE
2019

Dados Internacionais de Catalogação na Publicação
Instituto Federal do Ceará - IFCE
Sistema de Bibliotecas - SIBI
Ficha catalográfica elaborada pelo SIBI/IFCE, com os dados fornecidos pelo(a) autor(a)

S586a Silva, Ivyson Lucas de Lima.
Uma análise da segurança em redes definidas por software baseadas no openflow / Ivyson Lucas de Lima Silva. - 2019.
59 f. : il. color.

Trabalho de Conclusão de Curso (graduação) - Instituto Federal do Ceará, Bacharelado em Ciência da Computação, Campus Aracati, 2019.
Orientação: Prof. Dr. Reinaldo Bezerra Braga.

1. Redes Definidas por Software. 2. OpenFlow. 3. Segurança. 4. Redes Mesh. 5. Análise. I. Titulo.

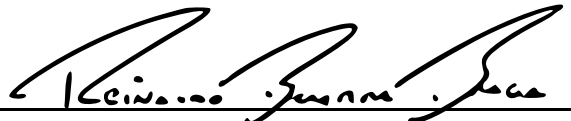
IVYSON LUCAS DE LIMA SILVA

UMA ANÁLISE DA SEGURANÇA EM REDES DEFINIDAS POR SOFTWARE BASEADAS NO OPENFLOW

Trabalho de Conclusão de Curso (TCC) apresentado ao curso de Bacharelado em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE - Campus Aracati, como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação.

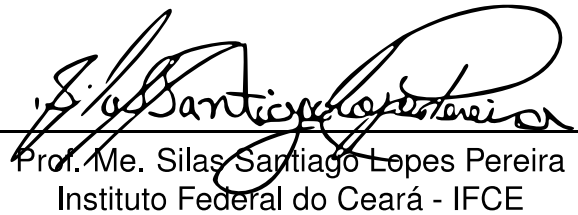
Aprovada em 17/04/2019

BANCA EXAMINADORA



Prof. Dr. Reinaldo Bezerra Braga (Orientador)
Instituto Federal do Ceará - IFCE

Prof. Esp. Renato Alexandre Costa Freitas
Instituto Federal do Ceará - IFCE



Prof. Me. Silas Santiago Lopes Pereira
Instituto Federal do Ceará - IFCE

DEDICATÓRIA

Dedico este trabalho aos meus pais e toda minha família que se fizeram presentes em todos os momentos de dificuldades.

AGRADECIMENTOS

Quero agradecer, em primeiro lugar, a Deus, pela força e coragem durante toda esta longa caminhada, a minha família por sempre me apoiar nos estudos. E agradeço a todos os professores do IFCE e os que passaram por ele, os que influenciaram diretamente na minha vida acadêmica, e aos servidores.

Meus sinceros agradecimentos a banca avaliadora, aos professores: Reinaldo Bezerra Braga, Silas Santiago Lopes Pereira e Renato Alexandre Costa Freitas por terem aceitado o convite para participar da banca de avaliação.

Quero agradecer especialmente ao Reinaldo Braga, como companheiro e orientador na caminhada ao longo do Curso. Eu posso dizer que a minha formação, inclusive pessoal, não teria sido a mesma sem o apoio dele. Desde quando o conheci no técnico em informática, como meu professor, ele vem sendo minha referência para o futuro.

RESUMO

Devido ao surgimento cada vez mais constante de objetos conectados e à dinamicidade do fluxo de dados, as redes tradicionais de computadores têm necessitado de uma nova arquitetura. Isso deve-se à grande heterogeneidade de dispositivos e expansão da Internet, além do gerenciamento complexo da rede. Com esses desafios, os pesquisadores criaram uma arquitetura para substituir a arquitetura tradicional de redes, chamada de Redes Definidas por Software (SDN -*Software Defined Network*). As SDNs prometem facilitar principalmente o gerenciamento da rede e a escalabilidade. Ou seja, essa nova arquitetura tornou-se o foco dos pesquisadores da área de redes. Por ser uma arquitetura promissora, ela tem sido apontada como a nova tecnologia que substituirá a arquitetura atual de rede. Diante dos diversos benefícios das SDNs na rede mundial de computadores, existe uma preocupação em relação à segurança do ambiente. Essa preocupação está relacionada principalmente ao controlador centralizado da rede, que administra um conjunto distribuído de comutadores OpenFlow. Neste contexto, diversos pesquisadores têm se esforçado para padronizar essa arquitetura, porém, ainda existem algumas lacunas que precisam de mais atenção na área de segurança de SDNs. Levando em consideração a tríade da segurança, conhecida como CIA (Confidencialidade, Integridade e Disponibilidade), este trabalho volta-se à análise da segurança de um ambiente real de SDN. O objetivo deste trabalho é realizar uma análise em ambiente real das vulnerabilidades das redes definidas por software, utilizando a integração da arquitetura da SDN com as redes em malha (*mesh networks*). Por fim, os resultados deste trabalho apresentam a quebra de dois elementos da tríade de segurança da SDN, a confidencialidade e a disponibilidade. Além disso, mostra a ineficiência de um ataque DDoS ou DoS em uma arquitetura SDN integrada com a rede *Mesh*.

Palavras-chaves: SDN. Redes Definidas por Software. *OpenFlow*. Segurança. Análise. Redes Mesh.

ABSTRACT

Due to the increasingly constant emergence of connected objects and the dynamics of data flow, traditional computer networks need a new architecture. This is due to the great heterogeneity of devices and expansion of the Internet, in addition to complex network management. With these challenges, the researchers created an architecture to replace the traditional network architecture, called Software Defined Network (SDN). SDNs promise to facilitate primarily network management and scalability. That is, this new architecture became the focus of researchers in the field. As a promising architecture, it has been singled out as the new technology that will replace the current network architecture. In view of the many benefits of SDNs in the global computer network, there is a concern for environmental safety. This concern is mainly related to the centralized controller of the network, which manages a distributed set of Open-Flow switches. In this context, several researchers have been struggling to standardize this architecture, however, there are still some gaps that need more attention in the SDNs security area. Taking into account the triad of security, known as CIA (Confidentiality, Integrity and Availability), this paper focuses on the security analysis of a real SDN environment. In other words, the objective of this work is to perform a real-world analysis of the vulnerabilities of the networks defined by software, using the integration of SDN architecture with the mesh networks. Finally, the results of this work show the breaking of two elements of SDN security triad, confidentiality and availability. In addition, it shows the inefficiency of a DDoS or DoS attack on an SDN architecture integrated with the Mesh network.

Keywords: SDN. Software Defiined Network. Mesh Network. Openflow. Security. Analisys.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de borda da rede. 21
Figura 2 – Características de enlaces de padrões selecionados de rede sem fio 23
Figura 3 – Elementos de uma rede sem fio 23
Figura 4 – Arquitetura da WLAN IEEE 802.11. 26
Figura 5 – 802.11 usa reconhecimentos da camada de enlace 28
Figura 6 – Uma rede ad hoc IEEE 802.11 30
Figura 7 – Arquitetura SDN. 31
Figura 8 – Arquitetura ODL. 35
Figura 9 – Arquitetura OvS. 36
Figura 10 –Arquitetura Ryu. 37

Figura 11 –Cenário sem hosts maliciosos. 43
Figura 12 –Interface Web (LuCI) do sistema embarcado OpenWrt 45
Figura 13 –Exemplo do comando para iniciar o controlador OpenFlow. 46
Figura 14 –Ferramenta de Dos ou DDoS 48
Figura 15 –Cenário de negação de serviço atacando o controlador 49
Figura 16 –Cenário de negação de serviço atacando o comutador 49

Figura 17 –Quantidade de pacotes descartados devido ao número excessivo de taxa de envio 52
Figura 18 –Quantidade de pacote capturados 53

LISTA DE TABELAS

Tabela 1 – Resumo dos padrões IEEE 802.11 25
Tabela 2 – Exemplo de tabela de fluxo 33

Tabela 3 – Componentes da arquitetura SDN 42
Tabela 4 – Configuração do cenário SDN 43
Tabela 5 – Especificações do roteador TP-LINK modelo TL-WR842ND. 44
Tabela 6 – Especificações do Notbook (Host Malicioso) 44
Tabela 7 – Especificações do Desktop (Host Malicioso) 44
Tabela 8 – Especificações do Controlador Raspberry pi 3 modelo B. 47

LISTA DE ABREVIATURAS E SIGLAS

AP	Access Point
ACK	Acknowledgement
AIEE	American Institute of Electrical Engineers Eletricistas
ARPANET	Advanced Research Projects Agency Network
BER	Bit Error Rate
BNSF	Base Network Service Functions
BSA	Basic Service Area
BSS	Basic Service Set
CIA	Confidencialidade, Integridade e Disponibilidade
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear to Send
DDoS	Distributed Denial of Service
DNS	Domain Name System
DoS	Denial of Service
ESS	Extended Service Set
FTP	File Transfer Protocol
GHz	Giga-Hertz
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Eletrical and Electronic Engineers
IFCE	Instituto Federal de Educação, Ciência e Tecnologia do Ceará
IoT	Internet of Things
IP	Internet Protocol
IRE	Instituto de Engenheiros de Rádio

LAN	Local Area Network
MAC	Media Access Control
MHz	Mega-Hertz
NIC	Network InterfaceCard
ODL	OpenDayLight
OvS	Open vSwitch
PC	Personal Computer
PNSF	Platform Network Service Functions
RTS	Request to Send
SAL	Service Abstraction Layer
SDN	Software-Defined Network
SIFS	Short Inter-Frame Spacing
SMTP	Simple Mail Transfer Protocol
SNR	Signal-to-Noise Ratio
SSID	Service Set Identifier
SSH	Secure Shell
TCC	Trabalho de Conclusão de Curso
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
VoIP	Voice over Internet Protocol
WLAN	Wireless Local Area Network
WMN	Wireless Mesh Networks

SUMÁRIO

1 INTRODUÇÃO 15
1.1 Objetivos 18
1.2 Organização do Trabalho 18

2 FUNDAMENTAÇÃO TEÓRICA 20
2.1 Arquitetura Tradicional de Redes 20
2.2 Redes sem Fio 22
2.2.1 Características de Enlaces e Redes sem Fio 24
2.2.2 Padrão IEEE 802.11 25
2.2.2.1 Arquitetura 802.11 25
2.2.2.2 Protocolo MAC 802.11 27
2.2.2.3 Redes Ad-hoc 29
2.2.2.4 Redes em Malha sem Fio 30
2.3 SDN (*Software-Defined Networking*) 31
2.3.1 OpenFlow 32
2.3.2 Controladores OpenFlow 34
2.3.2.1 FloodLight 34
2.3.2.2 OpenDayLight 34
2.3.2.3 Open vSwitch 36
2.3.2.4 POX 37
2.3.2.5 Ryu 37
2.4 Segurança 38
2.5 Redes em Malha Definidas por Software 39
2.6 Trabalhos Relacionados 40

3 PROPOSTA 42
3.1 Cenário Experimental 42
3.2 OpenWrt 44
3.3 Controlador POX 46
3.4 Ataques de Rede Realizados 47
3.4.1 Negação de Serviço 47
3.4.2 *Sniffing* 48

4 RESULTADOS 51
4.1 Negação de Serviço 51
4.2 Sniffing 52

5 CONCLUSÃO 55

REFERÊNCIAS 57

1 INTRODUÇÃO

As redes de computadores tiveram início na década de 60. Naquela época, a rede telefônica dominava o mundo através da comutação de circuitos. Assim, com o objetivo de interligar dois dispositivos ou mais entre si, de modo que pudessem compartilhar recursos, iniciaram-se os primeiros estudos sobre comutação de pacotes. Como consequência, nasceu em 1969, a precursora da Internet, a ARPANET(*Advanced Research Projects Agency Network*).

Em 1972, a ARPANET contava com 30 computadores interligados em localidades ao redor dos EUA, instalações militares e empresas, todos compartilhando recursos e trocando dados entre si. Entretanto, isso não era suficiente, pois a comunidade de redes de computadores, na mesma época, estava interessada em conectar o mundo. Sendo assim, nasceram os protocolos TCP/IP (Transmission Control Protocol/Internet Protocol) e outros protocolos importantes até os dias atuais. Esses protocolos aumentaram a escalabilidade da rede, o que facilitou sua implantação no mundo. Posteriormente, com o processo acelerado da globalização, foi necessário que as redes de telecomunicações se expandissem, surgindo a Internet.

A Internet é uma rede de computadores que interconecta milhares de dispositivos de comunicação em todo o mundo. Na década de noventa, esses dispositivos computacionais eram principalmente Estações de Trabalho (*Personal Computer* - PC) e servidores (KUROSE; ROSS, 2012). No entanto, atualmente, percebe-se que os computadores vão desde relógios inteligentes até objetos conectados por meio da Internet das Coisas (*Internet of Things* - IoT).

A ideia de conectar milhares de dispositivos na internet fez surgir um problema, os cabos. Esse foi um dos principais motivos que levou à criação das redes sem fio, ou *wireless network*. Isso possibilitou o desenvolvimento de diversos padrões de tecnologias sem fio. Segundo o (KUROSE; ROSS, 2012), uma classe em particular mostrou-se claramente a vencedora, a IEEE 802.11, conhecida como Wi-Fi. Difundidas no local de trabalho, na casa, nas instituições educacionais, nos cafés, nos aeroportos e nas esquinas das ruas, as LANs (*Local Area Network*) sem fio são hoje uma das mais importantes tecnologias de rede de acesso na Internet atualmente. Isso fez com que o uso da arquitetura tradicional de redes, ficasse cada vez mais complexa e difícil de configurar e gerenciar, devido à falta de flexibilidade, programabilidade e escalabilidade.

Em consequência disso, a comunidade de redes de computadores estuda a formulação e avaliação de arquiteturas alternativas para as redes do Futuro. (FARIAS

et al., 2011). Esses estudos fizeram as redes de computadores evoluírem gradualmente, surgindo as Redes de malha sem fio e chegando ao novo paradigma de redes de computadores, conhecida como Rede Definida por Software.

As Redes de malha sem fio (do inglês *Wireless Mesh Networks* - WMN) estão no grupo de tecnologias mais promissoras para configurar uma infraestrutura de rede da próxima geração. As WMNs são compostas de roteadores sem fio, chamados de roteadores de malha, interconectados para formar um *backbone* de múltiplos saltos. Seguindo o conceito de redes em malha, a ideia deste presente trabalho é fazer a integração entre as redes em malha e a arquitetura de Redes Definidas por Software.

Em paralelo, a SDN separa o plano de dados do plano de controle, no qual permite aos administradores de rede gerenciar, configurar e manter a rede. Esse é o grande diferencial da SDN em relação às redes tradicionais, onde cada roteador possui o plano de dados e o de controle. Na SDN, o plano de controle é centralizado e localizado no controlador, sendo este o responsável por tomar todas as decisões da rede. Já o plano de dados fica no roteador, apenas direcionando pacotes e respeitando as decisões do plano de controle. O plano de dados é abstraído através de uma tabela de fluxo, o qual contém um conjunto de regras para o processamento do fluxo (DELY; KASSLER; BAYER, 2011).

Segundo os autores de (SEZER et al., 2013), a SDN é uma grande promessa em termos de robustez na implementação e na operação da rede. Além disso, é uma excelente forma de reduzir o custo total do gerenciamento de redes empresariais e de operadoras, fornecendo serviços de rede programáveis. Atualmente, existem diversos protocolos e iniciativas para viabilizar a implementação da SDN. O protocolo mais adotado pela comunidade é o OpenFlow. Ele é fundamental para a construção de soluções em SDN (WIKIPéDIA, 2018).

O OpenFlow possui uma interface simples de programação, permitindo o controle da tabela de encaminhamento utilizada pelos dispositivos da rede. Além disso, ele possibilita o ajuste dinâmico do fluxo de tráfego em toda a rede para atender às necessidades em constante mudança. Para os autores de (BENTON; CAMP; SMALL, 2013), o OpenFlow e outros protocolos de SDN geraram interesse devido à quantidade de serviços de controle que oferecem aos desenvolvedores de software que fazem o controle da rede.

A utilização do protocolo OpenFlow para implementação de SDNs trouxe facilidades, tais como o gerenciamento da rede por parte dos administradores, a flexibilidade e escalabilidade. Entretanto, problemas também surgiram, tais como a potencialização no risco de ataques maliciosos, usando conhecidas técnicas de ataques, que visam as características oriundas da SDN (CENTENO et al., 2016). Assim, com essas problemáticas de segurança em SDN baseadas no protocolo OpenFlow, é importante

analisar à segurança na arquitetura, dada a sua rápida adoção nas atuais redes de computadores.

Este trabalho destaca algumas das vulnerabilidades presentes na arquitetura SDN, tais como, controlador vulnerável, coleta de informações, configurações. Como é possível perceber, todas as vulnerabilidades estão concentradas no controlador SDN. Ou seja, para os atacantes e os administradores da rede, é inicialmente perceptível que o controlador é o alvo principal dos ataques.

A utilização de interfaces programáveis abertas acarretam em riscos de segurança às arquiteturas SDN, uma vez que expõe as vulnerabilidades do software do controlador totalmente expostas aos invasores. Assim, o invasor terá informações suficientes para definir uma estratégia de ataque. Além disso, o controlador fornece diferentes interfaces programáveis para a camada de aplicação e esse nível de abertura pode levar a um exploração da interface, como a incorporação de códigos maliciosos, tais como vírus, *worms* e *malwares*. Portanto, as interfaces abertas dos controladores SDN precisam ser cuidadosamente avaliadas e analisadas (SHU et al., 2016).

A Negação de serviço em SDNs envolve sobrecarregar um recurso computacional específico ou toda a rede, de tal forma que o controlador SDN seja incapaz de encaminhar pacotes conforme o esperado. Um ataque bem-sucedido envolve o envio de um grande número de pacotes ao controlador SDN, possivelmente instanciando novos fluxos. Como resultado do ataque, o fluxo da rede pode ser completamente paralisado, tornando a rede inoperável (KANDOI; ANTIKAINEN, 2015).

Além desses desafios, existem também problemas de segurança no protocolo OpenFlow. As vulnerabilidades que são exploradas neste trabalho estão relacionadas às principais partes da arquitetura OpenFlow, que são o *switch* OpenFlow, o controlador e o canal (LI; MENG; KWOK, 2016). Para avaliar as vulnerabilidades, é considerado o modelo CIA (Confidencialidade, Integridade e Disponibilidade). Porém, são analisadas apenas a confidencialidade e a disponibilidade. Neste contexto, a confidencialidade é um conjunto de regras que limita o acesso à informação e a disponibilidade é uma garantia de acesso confiável às informações por pessoas autorizadas.

No *switch* OpenFlow, os problemas de segurança ocorrem por causa do plano de dados, pois o OpenFlow transforma a tabela de fluxo em um alvo cobiçado pelos atacantes. Ou seja, o plano de dados pode fornecer informações importantes do gerenciamento da rede, roteamento e controle de acesso.

O principal contratempo de segurança do controlador é devido a separação do plano de controle do plano de dados, tornando o primeiro centralizado. Assim, o plano de controle torna-se um gargalo para a SDN e, consequentemente, um alvo importante para os atacantes. Dentre os ataques mais comuns ao controlador, destacam-se o

flooding, o *sniffing* e o DoS(*Denial of Service*).

A problemática na vulnerabilidade do canal é que não existe um mecanismo de confiança apropriado entre controladores e comutadores. Isso abre um espaço para os invasores comprometerem a segurança por interceptação, aplicando ataques tradicionais, tal como o *man-in-the-middle* (MITM).

Levando tudo isso em consideração, constata-se que a realização da analisar de vulnerabilidades da arquitetura SDN, baseada no protocolo OpenFlow, é importante para identificar vulnerabilidades desta promissora tecnologia. Além disso, ao entender estas vulnerabilidades, é possível propor soluções para tornar o protocolo mais seguro e, consequentemente, a arquitetura mais robusta.

1.1 Objetivos

Levando em consideração todas as informações apresentadas, este trabalho tem como objetivo geral apresentar uma análise das vulnerabilidades da arquitetura SDN, baseada no protocolo OpenFlow, em um ambiente real.

Os objetivos específicos apresentados neste trabalho são:

- Realizar um estudo das tecnologias de redes sem fio integrando a arquitetura SDN;
- Criar uma rede com a arquitetura SDN em ambiente real;
- Monitorar o fluxo da rede;
- Analisar a segurança da rede e realizar ataques que exploram características oriundas da SDN;
- Documentar os ataques realizados e apontar as vulnerabilidades.

1.2 Organização do Trabalho

O restante do trabalho encontra-se organizado e estruturado da seguinte forma. O Capítulo 2 apresenta a fundamentação teórica relacionada ao tema proposto, descrevendo uma visão geral do estado da arte das redes sem fio e da arquitetura SDN, assim como os padrões estabelecidos para as redes em estudo, além de apresentar os controladores utilizados na arquitetura e suas características, apresentando os princípios de segurança da SDN, a qual é a tríade CIA e trabalhos relacionados à pesquisa. No Capítulo 3, são apresentados os métodos utilizados para avaliação da

segurança na arquitetura SDN, explicando cada experimento e as ferramentas utilizadas nos mesmos. Em seguida, a análise dos resultados obtidos neste trabalho a partir de exaustivos experimentos realizados com equipamentos de baixo qualidade são abordados no Capítulo 4. Por fim, no Capítulo 5, apresentam-se as considerações finais, que relatam os resultados obtidos com este trabalho, bem como são propostos trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste Capítulo será apresentada a fundamentação teórica necessária para uma melhor compreensão da proposta do trabalho. A Seção 2.1 explica os principais conceitos das redes tradicionais que são usadas atualmente. A Seção 2.2 inicia com um estudo das redes sem fio, apresentando os principais conceitos, características e elementos que compõem essas redes, bem como traçando os desafios propostos pela natureza sem fio dos enlaces de comunicação nessas redes e pela mobilidade que esses enlaces sem fio possibilitam. A Seção 2.3 apresenta os principais conceitos e características da SDN, um dos seus principais protocolos, o OpenFlow e os controladores. Na Seção 2.4 são apresentados os principais conceitos de segurança na arquitetura SDN baseado na tríade CIA. Na Seção 2.5 é apresentada a integração entre a SDN e as redes de malha sem fio. Por fim, a Seção 2.6 apresenta os trabalhos relacionados.

2.1 Arquitetura Tradicional de Redes

Segundo (KUROSE; ROSS, 2012), a rede mundial de computadores, chamada de Internet, é complexa devido, principalmente, ao seu tamanho. Ao mesmo tempo, este autor afirma que é possível dividir esta grande rede em partes específicas, além de protocolos heterogêneos, com o auxílio de uma arquitetura em camadas. Cada camada possui um conjunto de protocolos, tornando a Internet governada por protocolos que definem formatos, ordens de mensagens enviadas e recebidas entre entidades de rede, e ações tomadas (KUROSE; ROSS, 2012).

A estrutura da rede é composta pela borda da rede, núcleo da rede, redes de acesso e meios físicos. A borda da rede, como mostra a Figura 1, é conhecida como sistema final, que executa as aplicações de rede. No modelo cliente/servidor, o *host* cliente faz requisições e recebe respostas dos serviços de rede disponibilizados pelo servidor. Existem também as topologias que todos os *hosts* podem ser clientes e servidores ao mesmo tempo, chamadas de redes *peer-to-peer*. Nestas redes, os *hosts* podem consumir e oferecer serviços de rede. Tanto para topologias cliente-servidor quanto *peer-to-peer*, a borda da rede pode se conectar a outra borda da rede de duas formas:

- Orientado a conexão: que tem o objetivo de fazer transferência de dados entre sistemas por meio de um estabelecimento prévio de conexão, conhecido como Handshake de três vias (*Three Way Handshaking*). A orientação à conexão

é utilizado no protocolo TCP (*Transmission Control Protocol*), o qual pré estabelece uma conexão, provendo confiabilidade na transmissão de dados, sendo um protocolo confiável possuindo mecanismos de controle de perda, de fluxo e de congestionamento. Dentre as aplicações TCP, destacam-se o Protocolo de Transferência de Hipertexto (*Hypertext Transfer Protocol* - HTTP) (Web), Protocolo de Transferência de Arquivos (*File Transfer Protocol* - FTP) (transferência de arquivo), SSH (*Secure Shell*) (login remoto) e o Protocolo de transferência de correio simples (*Simple Mail Transfer Protocol - SMTP*) (e-mail).

- Sem conexão: possui o mesmo objetivo que o serviço orientado a conexão, porém, sem estabelecimento prévio de conexão. O protocolo UDP (*User Datagram Protocol*) é o protocolo mais conhecido por não estabelecer previamente a conexão. Portanto, a transferência de dados é não-confiável, sem mecanismos de controle de fluxo e de congestionamento. Nas Aplicações UDP, destacam-se a streaming media, video conferência, sistema de nomes de domínios (*Domain Name System* - DNS) e VoIP (*Voice over Internet Protocol*).

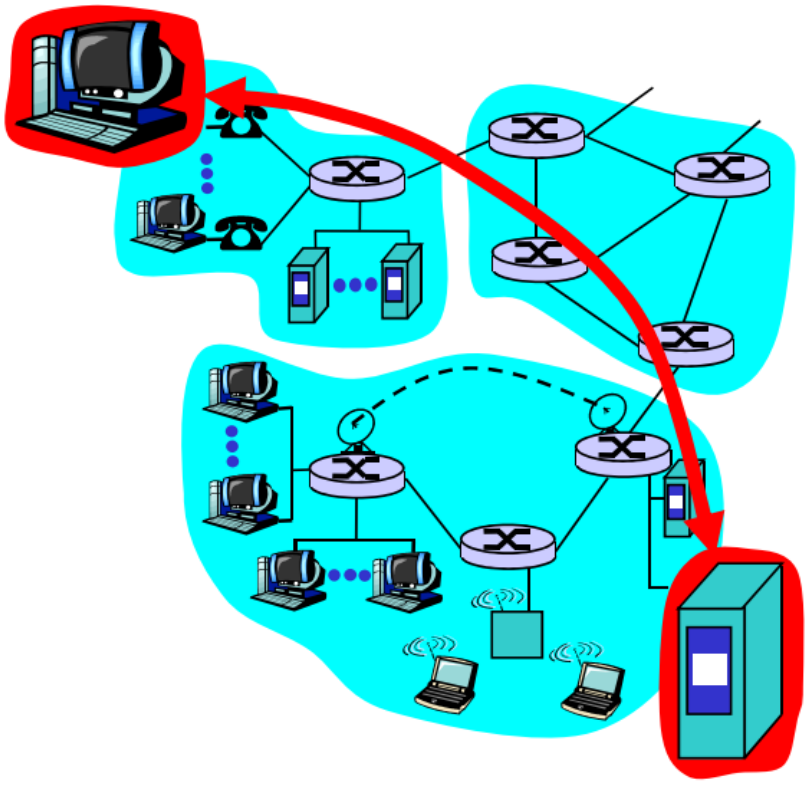


Figura 1 – Exemplo de borda da rede.

Fonte: (KUROSE; ROSS, 2012)

O núcleo da rede é formado por malhas de computadores interconectados. Os dados podem ser transferidos através de dois métodos, comutação de circuito

e comutação de pacotes. A comutação de circuito usa um canal dedicado para a conexão, ou seja, ele monopoliza o caminho, tornando-o dedicado até o fim da sua conexão. Um exemplo bem comum da comutação de circuitos é a rede telefônica. A comutação de pacotes é mais discreta, pois os dados são enviados em pedaços por diferentes caminhos da rede, evitando reserva dedicada de recursos do meio de transporte.

Com todo esse conhecimento, necessitava-se expandir as redes, mas havia o problema de cabeamento, utilizar cabos era um problema, que dificultava a manutenção e a integração de novos nós. Então surgiram as redes sem fio, para resolver esses problemas e melhorar a mobilidade dos usuários.

2.2 Redes sem Fio

As redes sem fio trouxeram a mobilidade, tornando fácil a configuração, uso e manutenção. As muitas vantagens das redes sem fio são evidentes para todos. Dentre estas vantagens, destacam-se o acesso em qualquer lugar e a qualquer momento, por meio de um equipamento leve e totalmente portátil, com a promessa de acesso desimpedido à Internet (KUROSE; ROSS, 2012). Dessa maneira, em uma rede sem fio, é possível identificar os seguintes elementos:

- **Hospedeiros sem fio:** Os hospedeiros são equipamentos de sistemas finais que executam aplicações. Um hospedeiro sem fio pode ser um notebook, um *smartphone* ou um computador de mesa com placa *wireless*. Eles podem ser móveis ou não. Exemplos de hospedeiros na Figura 3.
- **Enlaces sem fio:** Um hospedeiro móvel se conecta a um outro dispositivo por meio de um enlace sem fio. Existem diferentes tipos de tecnologias de enlace sem fio, com taxas de transmissão distintas e podem transmitir a distâncias variadas, como se pode ver na Figura 2.
- **Estação base:** Uma estação-base é responsável pelo envio e recebimento de dados (por exemplo, pacotes) de um hospedeiro sem fio que está associado a ela. A estação é frequentemente responsável pela coordenação da transmissão de vários hospedeiros sem fio com os quais está associada. Quando dizemos que um hospedeiro sem fio está “associado” a uma estação-base, isso quer dizer que: O hospedeiro está dentro do alcance de comunicação sem fio da estação-base. O hospedeiro usa a estação-base para retransmitir dados entre ele (o hospedeiro) e a rede maior.

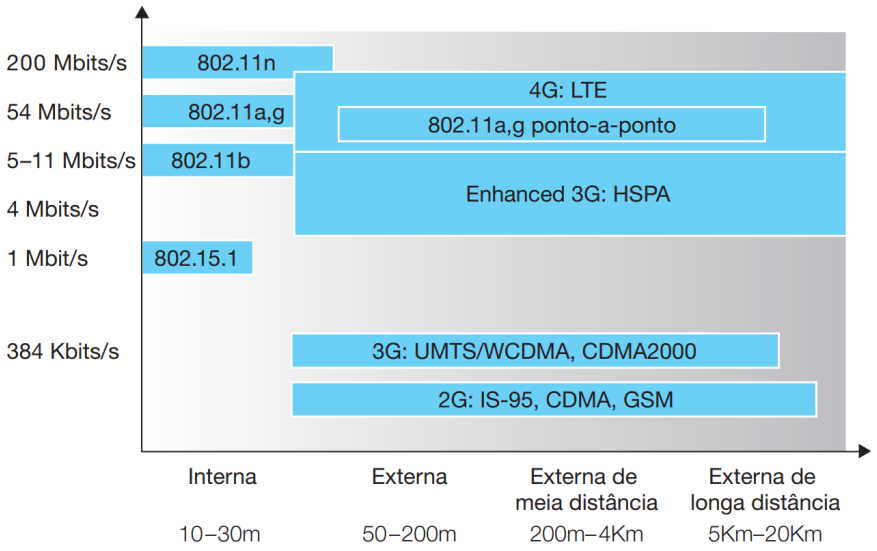


Figura 2 – Características de enlaces de padrões selecionados de rede sem fio

Fonte: (KUROSE; ROSS, 2012)

- **Infraestrutura de Rede:** É a rede maior com a qual um hospedeiro sem fio pode se comunicar.

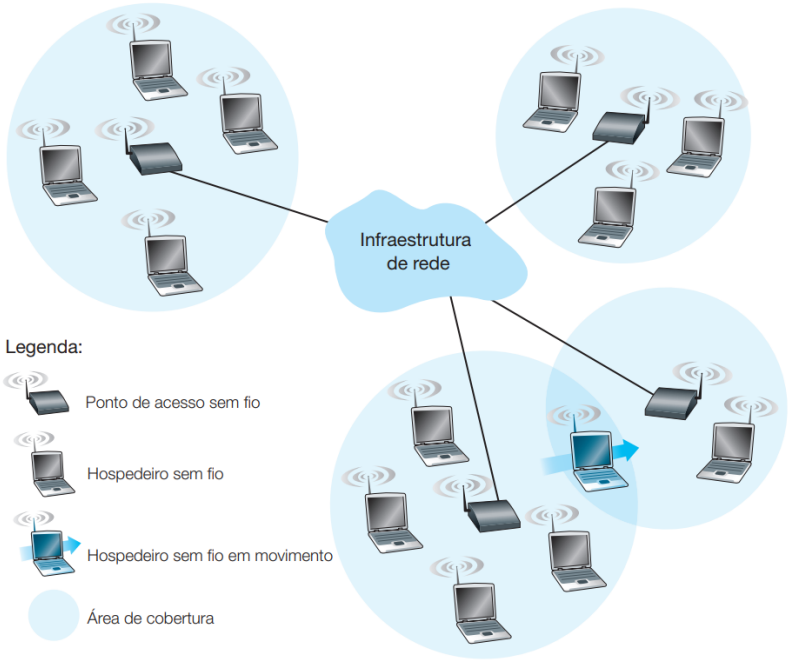


Figura 3 – Elementos de uma rede sem fio

Fonte: (KUROSE; ROSS, 2012)

As redes sem fio são classificadas em dois critérios : (*i*) se um pacote na rede

sem fio atravessa exatamente um salto sem fio ou múltiplos saltos sem fio; e (*ii*) se há infraestrutura na rede, como uma estação-base.

2.2.1 Características de Enlaces e Redes sem Fio

Acima da camada de rede os enlaces sem fio e cabeados não possuem diferenças, então a atenção é voltada para a camada de enlace, que possui especificidades, tais como:

- **Redução da força do sinal:** Radiações eletromagnéticas são enfraquecidas quando atravessam algum tipo de matéria (por exemplo, um sinal de rádio ao atravessar uma parede). O sinal se dispersa até mesmo ao ar livre, resultando na redução de sua força à medida que aumenta a distância entre emissor e receptor. Mas, mesmo que o dispositivo esteja próximo ao roteador, isso não significa que ele pode ter uma conexão mais rápida, devido às redes sem fio ser sujeita às interferências e instabilidades das demais ondas de rádio.
- **Interferência de outras fontes:** Várias fontes de rádio transmitindo na mesma banda de frequência sofrerão interferência umas das outras. Por exemplo, telefones sem fio de 2,4 GHz (*Giga-Hertz*), Bluetooth e LANs sem fio 802.11b transmitem na mesma banda de frequência. Assim, o usuário de uma LAN sem fio 802.11b que estiver se comunicando por um telefone sem fio de 2,4 GHz pode esperar que nem a rede nem o telefone funcionem particularmente bem. Além da interferência de fontes transmissoras, o ruído eletromagnético presente no ambiente pode causar interferência.
- **Propagação multicaminhos:** A propagação multicaminhos ocorre quando partes da onda eletromagnética se refletem em objetos, no solo e tomam caminhos de comprimentos diferentes entre um emissor e um receptor. Isso acaba resultando no embaralhamento do sinal recebido no destinatário. Objetos que se movimentam entre o emissor e o receptor podem fazer com que a propagação multicaminhos mude ao longo do tempo.

Ao considerar as falhas que podem ocorrer em um canal sem fio, o hospedeiro pode receber um sinal eletromagnético como uma combinação degradada do sinal original transmitido pelo remetente. Essa degradação ocorre devido aos efeitos da atenuação e da propagação multicaminhos, bem como por ruídos de fundo no ambiente. A relação sinal-ruído (SNR — signal-to-noise ratio) é uma medida relativa da potência do sinal recebido, ou seja, a informação sendo transmitida e o ruído. Quanto maior for a SNR, mais fácil é para o destinatário extrair o sinal transmitido de um ruído de

fundo. Isso também significa que será menor a taxa de erro de bits (BER - Bit Error Rate) (KUROSE; ROSS, 2012).

2.2.2 Padrão IEEE 802.11

O *Institute of Eletrical and Electronic Engineers* (IEEE) foi formado pela fusão do IRE (Instituto de Engenheiros de Rádio) com o AIEE (Instituto Americano de Engenheiros Eletricistas). O IEEE visa compartilhar e integrar conhecimento na área de Tecnologia da eletricidade e tecnologia da informação. Além disso, fazendo um dos seus papéis mais importantes, que é o de estabelecer os padrões para formatos comunicação, computadores e dispositivos.

O padrão IEEE 802.11 foi o primeiro padrão a ser desenvolvido para redes de transmissão de dados sem fio (IEEE 802.11, 1997). O protocolo 802.11 inclui autenticação, associação e serviços de manutenção, além de um procedimento de criptografia opcional, gerenciamento de energia para reduzir o consumo em estações móveis e uma função de coordenação de ponto para transferência de dados por tempo (IEEE 802.11, 1997). Atualmente, há diversos padrões 802.11 para tecnologia WLAN (*Wireless Local Area Network*), entre elas destacam-se 802.11b, 802.11a e 802.11g (KUROSE; ROSS, 2012). A tabela a seguir apresenta um resumo dos padrões IEEE 802.11.

Tabela 1 – Resumo dos padrões IEEE 802.11

Padrão	Faixa de frequências (EUA)	Taxa de dados
802.11b	2,4–2,485 GHz	até 11 Mbits/s
802.11a	5,1–5,8 GHz	até 54 Mbits/s
802.11g	2,4–2,485 GHz	até 54 Mbits/s

Fonte: (KUROSE; ROSS, 2012)

Esses padrões especificam uma arquitetura comum, que deve ser adotada por todos os dispositivos que fazem parte da rede sem fio. Dentre estas especificações, fazem parte os recursos de transmissão, os aspectos de transferência de dados sem fio e o modo de comunicação.

2.2.2.1 Arquitetura 802.11

Uma rede sem fio (Wireless) é uma extensão de rede LAN (Local Area Network). Assim, criando o conceito de rede local sem fio a WLAN (Wireless Local Area Network). O fundamental na arquitetura 802.11 é o BSS (*Basic Service Set*). O BSS contém estações sem fio e uma estação base central, conhecida como AP(*Access Point*) (IEEE

802.11, 2012) (KUROSE; ROSS, 2012). Em uma residência há apenas um AP e um roteador, normalmente sendo integrados a um equipamento, que conecta o BSS à Internet. Uma BSA (*Basic Service Area*) é a área conceitual dentro da qual os membros de um conjunto de serviços básicos (BSS) podem se comunicar. Um sistema de distribuição é responsável por interligar múltiplas BSAs, bem como, permitir a construção de redes cobrindo áreas maiores que uma célula (BSA) (IEEE 802.11, 1997). Para complementar, o ESS (*Extended Service Set*) é o conjunto de Serviço Amplo, que representa um conjunto de estações formado pela união de vários BSSs conectados por um sistema de distribuição.

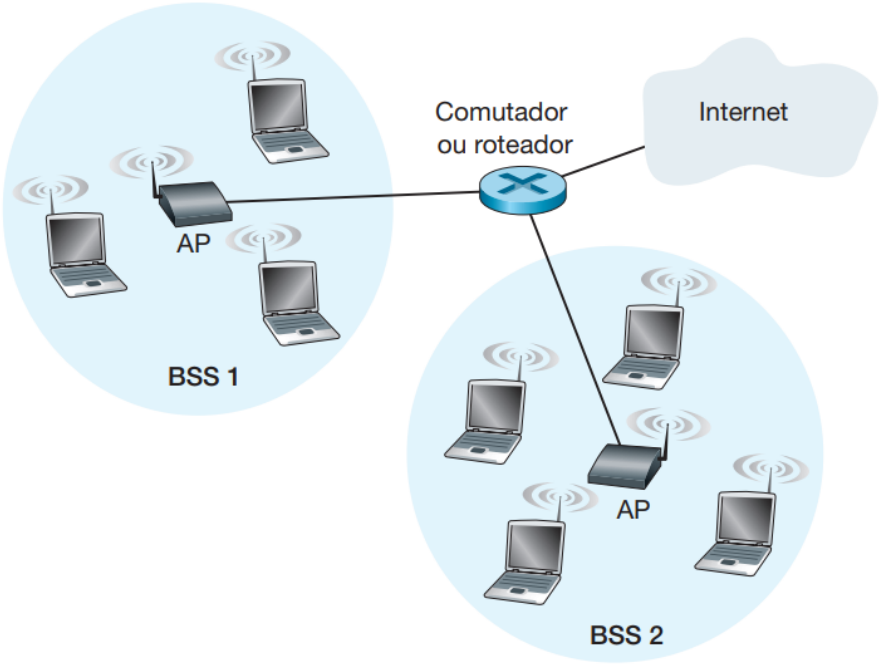


Figura 4 – Arquitetura da WLAN IEEE 802.11.

Fonte: (KUROSE; ROSS, 2012)

A Figura 4 mostra claramente as definições citadas anteriormente. O comutador ou roteador é um exemplo de BSA, a Internet é o sistema de distribuição que interliga múltiplas BSAs, possibilitando a cobertura de áreas maiores. Na arquitetura 802.11, cada estação sem fio precisa se associar com um AP antes de poder enviar ou receber dados. Ao instalar um AP, um administrador de rede determina um SSID (*Service Set Identifier*) ao ponto de acesso, que é uma *string* (texto) de até 32 caracteres, que identifica a rede sem fio. O adminitrador também deve designar um número de canal ao AP. As redes 802.11b operam na faixa de frequência de 2,4 GHz a 2,485 GHz. Dentro dessa faixa de 85 MHz (Mega-Hertz), o padrão 802.11 define 11 canais, que se sobrepõem em parte. Em particular, os canais 1, 6 e 11 são aqueles que mais evitam sobreposição de canais. Portanto, esses três canais são considerados

ortogonais, pois estão livres de interferência inter-canais (FILHO et al., 2018).

Um elemento fundamental na arquitetura de rede local sem fio com infraestrutura é o ponto de acesso, que desempenha as seguintes funções:

- **Autenticação, associação e reassociação:** permite que uma estação móvel mesmo saindo de sua célula (BSA) de origem continue conectada à infraestrutura e não perca a comunicação. A função que permite manter a continuidade da comunicação quando um usuário passa de uma célula para outra, é conhecida como handoff (ALENCAR, 2000) (SOARES; LEMOS; COLCHER, 1995).
- **Gerenciamento de potência:** permite que as estações operem economizando energia, através de um modo chamado de *power save* (SOARES; LEMOS; COLCHER, 1995).
- **Sincronização:** garante que as estações associadas a um AP estejam sincronizadas por um relógio comum (SOARES; LEMOS; COLCHER, 1995).

2.2.2.2 Protocolo MAC 802.11

Uma vez associada com um AP, uma estação sem fio pode começar a enviar e receber quadros de dados. Para transmitir quadro de dados ao mesmo tempo sobre o mesmo canal, é preciso um protocolo de acesso múltiplo para coordenar as transmissões. O protocolo aderido foi o CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*). Ele é de acesso aleatório com prevenção de colisão. A estação reconhece o canal antes de transmitir e abstém-se de transmitir quando percebe que o canal está ocupado.

O protocolo MAC (*Media Access Control*) 802.11 não utiliza detecção de colisão, como no protocolo de acesso ao meio do Ethernet. Existem duas razões para isso. A primeira é a capacidade de detectar colisões exige a capacidades de enviar o próprio sinal da estação e de receber para determinar se alguma outra estação está transmitindo ao mesmo tempo. Como a potência do sinal recebido em geral é muito pequena em comparação com a potência do sinal transmitido no adaptador 802.11, é caro construir um hardware que possa detectar colisões (KUROSE; ROSS, 2012). Segundo, mesmo que o adaptador pudesse transmitir e ouvir ao mesmo tempo, ainda assim ele não seria capaz de detectar todas as colisões, devido ao problema do terminal escondido e do desvanecimento (KUROSE; ROSS, 2012).

Como LANs 802.11 sem fio não usam detecção de colisão, uma vez que uma estação começa a transmitir um quadro, ela o transmite integralmente. Ou seja, quando uma estação inicia uma transmissão, não há volta. Como é de se esperar, ao

transmitir quadros inteiros longos, existe uma grande possibilidade de colisão. Para reduzir a probabilidade de colisões, o 802.11 emprega diversas técnicas de prevenção destas colisões, como o uso de mensagens específicas para reserva da comunicação.

O protocolo MAC 802.11 usa reconhecimentos de camada de enlace. Quando a estação de destino recebe um quadro que passou na verificação de CRC (*Cyclic Redundancy Check*), ela espera um curto período de tempo, conhecido como SIFS (*Short Inter-Frame Spacing*) e, então, devolve um quadro de reconhecimento. Caso a estação transmissora não receber um reconhecimento em dado período de tempo, ela admitirá que ocorreu um erro e retransmitirá o quadro usando de novo o protocolo CSMA/CA para acessar o canal. Quando a estação transmissora não recebe um reconhecimento após certo número fixo de retransmissões, desistira e o quadro será descartado. O exemplo deste processo é mostrado na Figura 5.

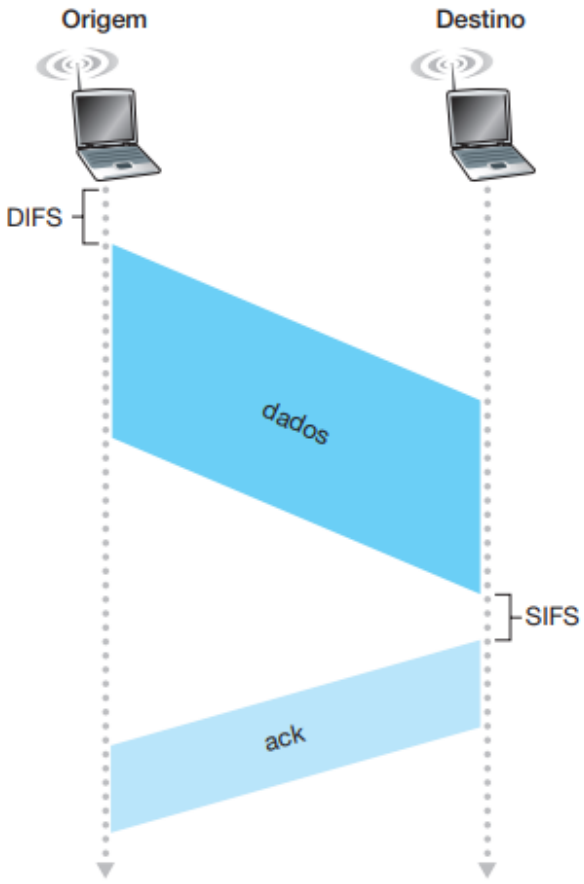


Figura 5 – 802.11 usa reconhecimentos da camada de enlace

Fonte: (KUROSE; ROSS, 2012)

Com esse protocolo, se duas estações perceberem que o canal está ocupado, ambas entrarão imediatamente em *backoff* aleatório e esperamos, depois escolherão valores diferentes de *backoff*. Visto que esses valores forem de fato diferentes, assim

que o canal ficar ocioso, uma das duas começará a transmitir antes da outra e no caso das duas não estiverem ocultas uma da outra a “estação perdedora” ouvirá o sinal da “estação vencedora”, interromperá seu contador e não transmitirá até que a estação vencedora tenha concluído sua transmissão. Desse modo é evitada uma colisão custosa. É claro que ainda podem ocorrer colisões com 802.11 (KUROSE; ROSS, 2012).

O protocolo 802.11 MAC também contém um esquema de reserva inteligente que ajuda a evitar colisões mesmo na presença de terminais ocultos. Para evitar esse tipo de problema é utilizado um quadro de controle RTS(*Request to Send*) e um quadro de controle CTS (*Clear to Send*). Se um remetente quer enviar um quadro DATA, ele primeiro envia um quadro RTS ao AP indicando o tempo total requerido para transmitir o quadro DATA e o quadro de reconhecimento (ACK). Quando o AP recebe o quadro RTS, responde fazendo a transmissão de um quadro CTS, por *broadcast* . Esse quadro CTS tem duas finalidades: dá ao remetente uma permissão explícita para enviar e também informa as outras estações a não enviar durante o tempo reservado. Lembrando que realizar a troca de RTS/CTS ajuda a reduzir o número de colisões, mas, introduz atrasos e consome recursos do canal. Portanto o RTS/CTS é utilizado apenas para a reserva do canal para o envio de um quadro longo.

2.2.2.3 Redes Ad-hoc

As redes Ad-hoc são hospedeiros sem fio que não dispõem de qualquer infraestrutura. Na falta da infraestrutura, os próprios hospedeiros devem prover os serviços tais como roteamento, atribuição de endereço, tradução de endereços semelhante ao DNS e outros. Uma rede ad hoc pode ser formada quando pessoas que portam notebooks se reúnem e querem trocar dados na ausência de um AP centralizado.

As redes ad hoc desperta um interesse extraordinário com o continuo aumento de equipamentos portáteis, que podem se comunicar. A responsabilidade pela organização e controle da rede é distribuída entre os próprios terminais. Nas redes ad hoc, alguns pares de terminais não são capazes de se comunicar diretamente entre si. A comunicação direta entre os dispositivos da rede, permite que tenha maior flexibilidade.

A grande vantagem das redes ad hoc é que dispensam qualquer tipo de dispositivo intermediador de conexão como o AP, tornando uma grande vantagem, que é a mobilidade. Além disso a criação de uma rede ad hoc é rápida e fácil, e pode ser feita praticamente em qualquer lugar. O fato de não utilizar APs torna a rede mais barata. Um exemplo de rede Ad Hoc na Figura 6.

Ainda assim, as redes ad hoc não são limitadas apenas a computadores. Mas,

Figura 6 – Uma rede ad hoc IEEE 802.11

Fonte: (KUROSE; ROSS, 2012)

existe também as desvantagens dessa rede, um problema é a construção de algoritmos de roteamentos devido a sua mobilidade e topologia dinâmica. Outra questão importante em redes Ad Hoc é a localização de um nó, visto que o endereço da máquina não ter relação com a posição atual do nó, da mesma forma não existe informações geográficas que ajudem na definição do posicionamento desse nó. Além disso, as redes Ad Hoc apresentam algumas consequências como interferência de sinal, interoperabilidade, perda de caminho, segurança e um gerenciamento de energia de forma eficiente.

2.2.2.4 Redes em Malha sem Fio

Desde 1997, momento de surgimento das redes em malha (*Wireless Mesh Networks - WMN*), associado ao padrão 802.11, os pesquisadores do trabalho (IEEE 802.11, 1997) vem melhorando essa tecnologia, com o intuito de ganhar maior mobilidade e evitar a dependência de cabos. As WMNs são uma das tecnologias sem fio mais promissoras para configurar uma infraestrutura sem fio, capaz de conectar dispositivos e transmitir fluxos de diferentes mídias na Internet (AL-SAADI et al., 2016).

Inicialmente, as WMNs consistiam em uma coleção de roteadores equipados com uma única interface de rádio IEEE 802.11 (ou seja, *Network InterfaceCard* - NIC) operando apenas em uma pequena porção do espectro disponível (um canal). As WMNs têm como vantagens o baixo custo, robustez, escalabilidade, simplicidade e os benefícios proporcionados pelas tecnologias sem fio. A robustez e a escalabilidade são vantagens das WMNs devido à característica dinâmica e da facilidade ao adicionar ou remover um nó da rede. Porém, apresentam desvantagens como degradação da

largura de banda, interferência eletromagnética e perda de pacotes. As características básicas que diferenciam as redes em malha das demais são:

- **Comunicação sem fio por múltiplos saltos e consumo de energia:** A comunicação por múltiplos saltos permite um maior alcance. Assim, permite um melhor reaproveitamento da utilização de frequência. Isso deve-se ao fato dos roteadores de malha geralmente serem estacionários e não possuírem restrições de energia. Eles podem dispor de recursos como múltiplas interfaces, memória, armazenamento, poder de computação e assim por diante
- **Rede autoconfigurável e auto curável:** Devido a sua natureza flexível (*Ad Hoc*) a WMN é mais fácil de configurar e capaz de resistir às diferentes alhas. Além disso, a rede é mais flexível para expansões esporádicas, de acordo com a necessidade de crescimento.

2.3 SDN (*Software-Defined Networking*)

Nos últimos anos, tem havido um interesse crescente em SDNs. Segundo os autores de (FOSTER et al., 2013), a SDN possui um controlador logicamente centralizado, que gerencia a funcionalidade do processamento de pacotes, que pertence a uma coleção distribuída de interruptores. Além de ser uma abordagem de rede que permite controlar ou ”programar” a rede de maneira inteligente usando softwares, a SDN torna a gerencia da rede consistente, robusta e abrangente.

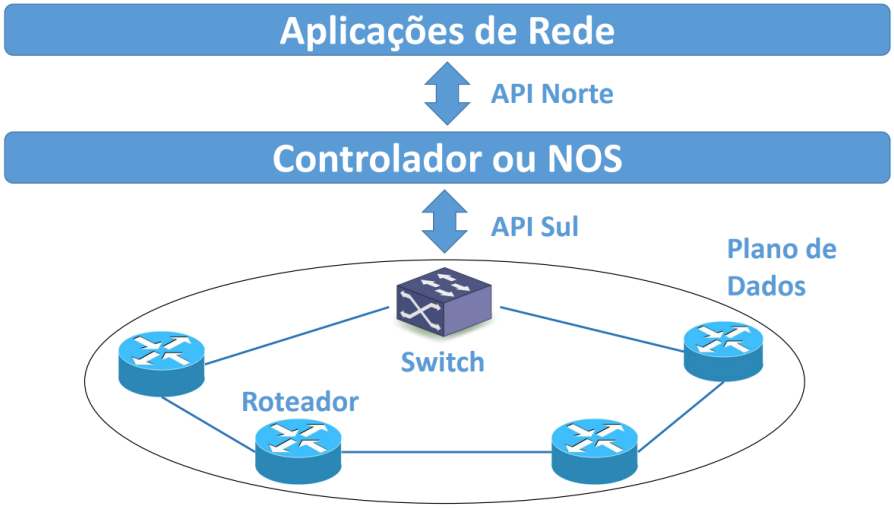


Figura 7 – Arquitetura SDN.

Fonte: (HAQUE; ABU-GHAZALEH, 2016)

A SDN é uma arquitetura dinâmica, gerenciável, econômica e adaptável, o que a torna ideal para a natureza ágil e de grande largura de banda dos aplicativos atuais. Esta arquitetura separa as funções de controle e encaminhamento da rede como se pode ver na Figura 7. Ela permite que o controle da rede se torne diretamente programável e a infraestrutura subjacente seja abstraída para aplicativos e serviços de rede(WIKIPéDIA, 2018).

Cada comutador no plano de dados conduz o encaminhamento das informações de acordo com as regras instaladas pelo sistema centralizado de rede. Cada regra pode ser expressa da forma de correspondência, ação. O campo de correspondência é usado para corresponder ao cabeçalho do pacote de um fluxo de tráfego. Se uma regra for correspondida, o comutador executará as ações especificadas no campo 'ação' (HUANG et al., 2015). A arquitetura SDN possui interfaces abertas entre os dispositivos do plano de controle e aqueles que estão no plano de dados (SEZER et al., 2013).

A ideia da SDN é deslocar a complexidade da rede para o controlador, assim trazendo a simplicidade e abstração para o operador da rede. A importância da SDN é devido aos seus recursos serem ilimitados, o que inclui sua flexibilidade, escalabilidade, redundância e desempenho comparado a uma rede tradicional que tem limitações de hardware e software, sendo assim, é visto a superioridade da SDN em relação a arquitetura tradicional de redes.

Quando uma rede requer recursos adicionais, haverá um custo considerável na compra de novo hardware e licenciamento. Devido a SDN ser uma rede abstraída em software, ela oferece maior flexibilidade na compra de um hardware. Além dos gerenciadores das redes possuírem a facilidade para alterar a lógica de funcionamento com o suporte do protocolo OpenFlow (CAMPOS, 2017).

2.3.1 OpenFlow

O OpenFlow é um protocolo de código aberto que controla os dispositivos, substituindo todo o plano de controle e definindo toda a ação que deve ser feita dentro do equipamento. Ele provê meios de controlar os dispositivos de rede (*switches* OpenFlow) sem a necessidade dos fabricantes exporem o código de seus produtos(LARA; KOLASANI; RAMAMURTHY,). Por ter um padrão aberto que busca o equilíbrio entre diferentes fabricantes, possibilita controlar e gerenciar equipamentos que suportam esse protocolo(CAMPOS, 2017).

Ao adotar o OpenFlow, os controladores podem programar a tabela de fluxos. Uma tabela de fluxo consiste em registros de movimentações da rede na entidade controladora. Ou seja, a tabela de fluxo contém as seguintes informações (WIKIPéDIA,

2018):

- **Campos de correspondência:** consistem em porta de entrada e cabeçalhos de pacotes. Opcionalmente, apresentam outros campos de pipeline, como os metadados especificados por uma tabela anterior.
- **Prioridade:** define as prioridades de correspondência da entrada de fluxo.
- **Contadores:** atualizados quando os pacotes são correspondidos.
- **Instruções:** para modificar o conjunto de ações ou o processamento de pipeline.
- **Tempo limite:** quantidade máxima de tempo ou tempo ocioso antes de o fluxo expirar pelo comutador.
- ***Cookie*:** valor de dados opaco escolhido pelo controlador. Pode ser usado pelo controlador para filtrar entradas de fluxo afetadas por estatísticas de fluxo, modificação de fluxo e solicitações de exclusão de fluxo. Não usado ao processar pacotes.
- ***Flags*:** *flags* alteram a maneira como as entradas de fluxo são gerenciadas.

A tabela de fluxos é composta por um campo de regra, de ação ou contadores como no exemplo da Tabela 2. Um comutador OpenFlow pode ter uma ou mais tabelas de fluxos, onde cada tabela de fluxo contém múltiplas entradas de fluxo (CAMPOS, 2017). O pacote chega no comutador OpenFlow, é comparado com a entrada de fluxos que estão na tabela e se houver uma entrada com quem uma regra coincida, o comutador irá executar a ação (CAMPOS, 2017).

Tabela 2 – Exemplo de tabela de fluxo

Padrão	Instruções	Contadores
IP Origem = 10.0.0.5	descartar	10
IP Origem = 10.0.0.5	Encaminhar para porta 150	221
Todos os pacotes	Encapsular e encaminhar para o controlador	1341

O OpenFlow é atualmente o conceito de SDN mais implantado, mas devido à natureza em rápida evolução, muitos fornecedores de switches e controladores não implementaram a especificação e ignoraram totalmente o TLS (*Transport Layer Security*), o que pode ser oportunidade para os invasores (LI; MENG; KWOK, 2016) (KLOTI; KOTRONIS; SMITH, 2013). Um dos problemas de segurança do OpenFlow é tonar o uso de TLS opcional, porque é difícil para os gerenciadores da rede configurar corretamente o TLS (BENTON; CAMP; SMALL, 2013).

Os fabricantes de hardware e software não oferecerem nenhum suporte a TLS. Apenas existe atualmente um controlador com suporte total ao TLS que é o

Open vSwitch. Os outros não apresentam interesse em implementar a TLS. Por causa desses problemas e a flexibilidade oferecida pelo OpenFlow, o controlador se torna um alvo para vários ataques de segurança (MUTAHER; KUMAR; WAHID, 2018).

2.3.2 Controladores OpenFlow

Os controladores são cérebros da rede que oferecem uma visão centralizada da rede global. Assim, eles oferecem um ambiente de programação onde o desenvolvedor pode ter acesso aos eventos gerados por uma interface de rede, que siga um padrão como o OpenFlow. Consequentemente, essa característica gera comandos para controlar a infraestrutura (GUEDES et al., 2012). Esta seção apresenta alguns dos principais controladores para a arquitetura SDN.

2.3.2.1 FloodLight

O *Floodlight* é o o mais popular entre os controladores OpenFlow de código aberto. É apoiado por uma comunidade de desenvolvedores, incluindo vários engenheiros da Big Switch Networks, escrito em Java e tem uma licença Apache, sua versão atual é a 1.2. Além disto, suporta um amplo conjunto de comutadores OpenFlow virtuais e físicos e pode lidar com redes mistas OpenFlow e não OpenFlow (CENTENO et al., 2016) (MORZHOV; ALEKSEEV; NIKITINSKIY, 2016).

O *Floodlight* realiza o controle de um conjunto de funcionalidades comuns para o controlador investigar a rede *Openflow*, enquanto os aplicativos sobre o *Floodlight* utilizam diferentes recursos para solucionar as necessidades dos usuários.

Existem quatro conceitos básicos no controlador *Floodlight*: serviços, módulos externos e internos e aplicativo de rede. Serviço é uma interface que exporta estados e gera eventos (MORZHOV; ALEKSEEV; NIKITINSKIY, 2016). Um consumidor de serviço pode obter ou configurar um estado e assinar ou cancelar a assinatura do serviço (MORZHOV; ALEKSEEV; NIKITINSKIY, 2016). Assim, é permitido ter várias implementações do mesmo serviço.

2.3.2.2 OpenDayLight

O ODL (OpenDayLight) é um controlador SDN de código aberto mantido pela fundação Linux e disponível em repositório no Github (MACHADO; TORRES, 2018) (LF, 2018). Com uma solução unificada de uma plataforma SDN, o ODL permite programadores e usuários trabalharem juntos de modo a criar redes mais dinâmicas e programáveis para provedores de Internet, datacenters e universidades.

A versão mais recente é a Fluorine, lançada em 30 de agosto de 2018, com a principal mudança sendo a alteração do projeto para um sistema de versões gerenciadas, com empacotamento mais simples para acelerar o desenvolvimento de soluções utilizando o ODL (MACHADO; TORRES, 2018). Faz uso de protocolos padrão, incluiem métodos para virtualização da rede e auxilia na implementação e gestão das políticas de segurança, sobretudo com processos de combate a ataques de DDoS (Distributed Denial of Service).

O ODL pode ser dividido em três camadas. A primeira camada é a que consiste de aplicações de rede e serviços. Sucessivamente, encontra-se a camada de controle, onde as abstrações da SDN são determinadas. Por último, existe a camada de interfaces *southbound* e *plugins* de protocolos de rede.

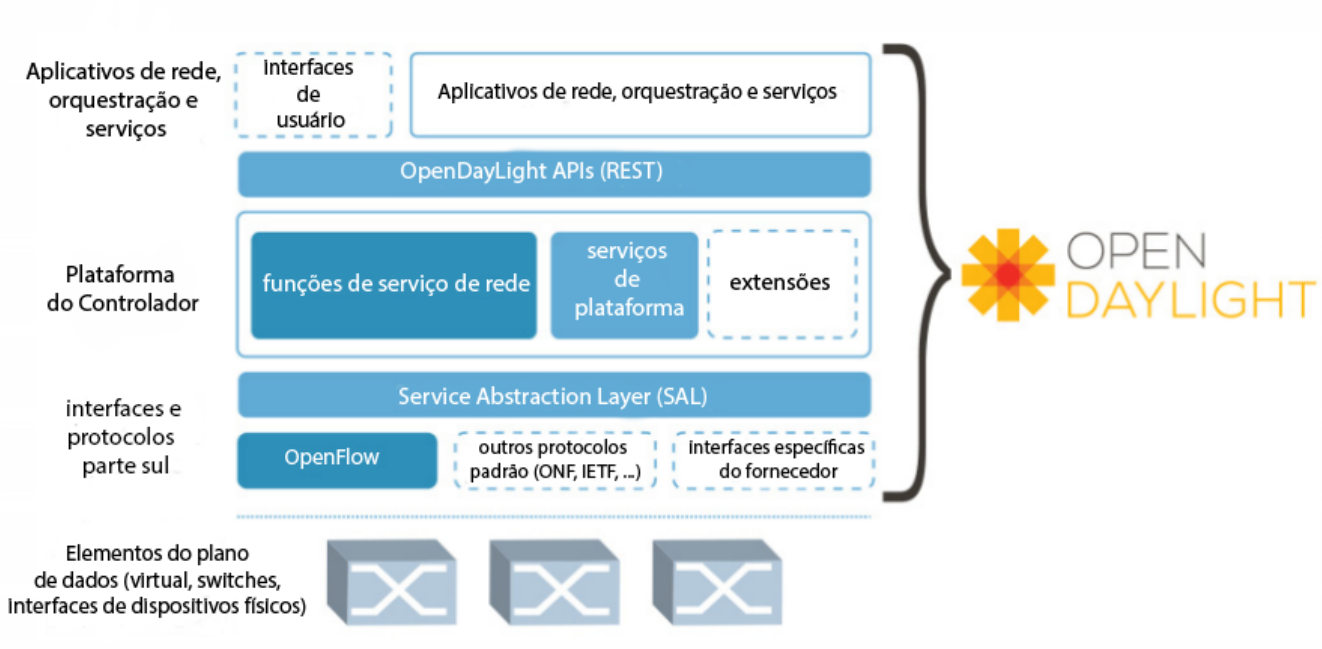


Figura 8 – Arquitetura ODL.

Fonte: (MACHADO; TORRES, 2018)

A camada de controle do ODL viabiliza a abstração da SDN atendendo as BNSFs (*Base Network Service Functions*), PNSFs (*Platform Network Service Functions*)e SAL(*Service Abstraction Layer*). Os serviços básicos de rede são responsáveis por implementar topologias, coletar informações, gerar estatísticas e orquestrar regras perante a rede (MACHADO; TORRES, 2018). A camada SAL faz a abstração para a programação do plano de dados. Ela é o coração do ODL, como mostra a Figura 8. Essa camada apresenta um gerenciador de *plugins* que permitem a comunicação com diversos protocolos de rede (MACHADO; TORRES, 2018).

2.3.2.3 Open vSwitch

O OvS (*Open vSwitch*) é um switch virtual multicamada de qualidade de produção, licenciado em conjunto com a licença do Apache 2.0, de código aberto. Ele foi projetado para permitir a automação maciça de rede, por meio de extensão que pode ser programada. Além disso, suporta interfaces e protocolos de gerenciamento seguindo os padrões NetFlow, sFlow, IPFIX, RSPAN, CLI, LACP, 802.1ag (Fundação Linux, 2016).

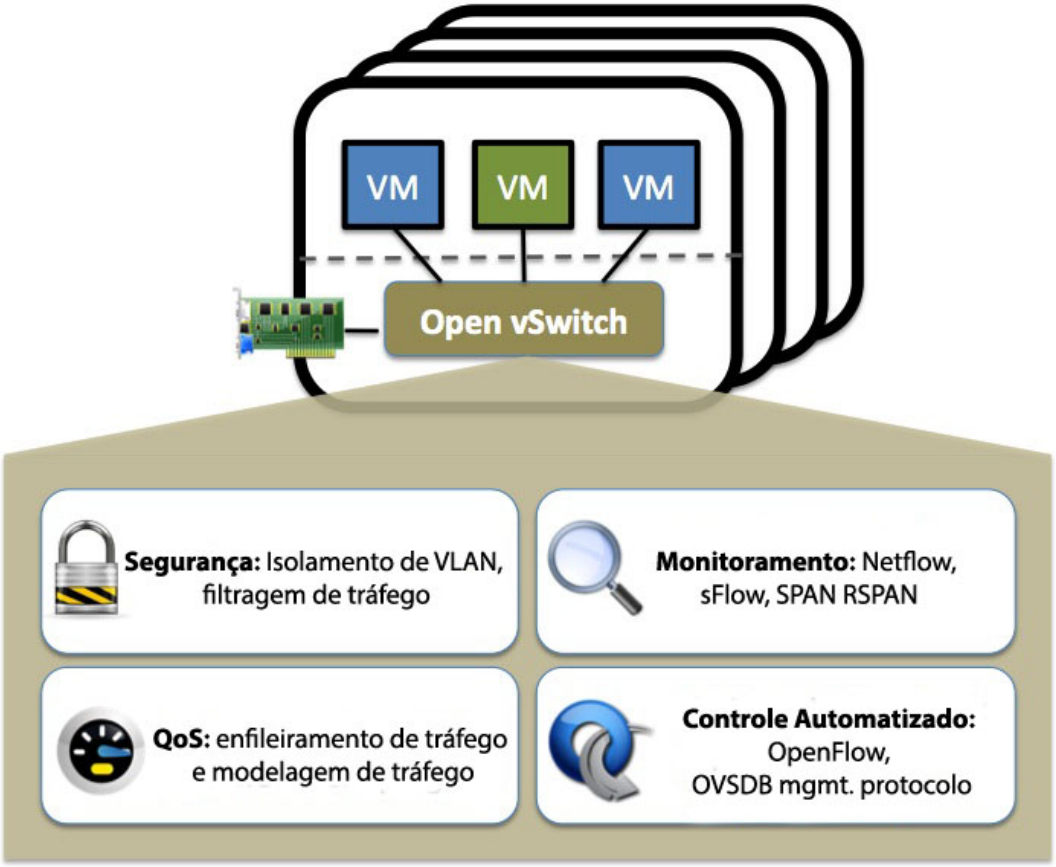


Figura 9 – Arquitetura OvS.

Fonte: (Fundação Linux, 2016)

O OvS é adequado para funcionar como um switch virtual em ambientes de VM. Além de expor interfaces de controle e visibilidade padrão à camada de rede virtual, foi desenvolvido para oferecer suporte à distribuição em vários servidores físicos. O OvS suporta múltiplas tecnologias de virtualização baseadas em Linux, incluindo Xen / XenServer, KVM e VirtualBox, apresentados na Figura 9 (Fundação Linux, 2016).

O controlador OvS gerencia qualquer número de switchs remotos sobre o protocolo OpenFlow, fazendo que eles funcionem como switchs da camada L2 (MAC -

Media Access Control).

2.3.2.4 POX

O POX foi significativamente inspirado no NOX. O NOX foi o primeiro controlador OpenFLow. Além disso, o POX foi desenvolvido em python, as versões iniciais não tinham nenhum tipo de sistema de dependência. Esses dois fatores levaram o POX a ter um caso especial: o componente OpenFlow estava habilitado por padrão. O POX oficialmente suporta Windows, Mac OS, e Linux (contudo ele tem sido usado em outros sistemas também). Muito do seu desenvolvimento é feito no Mac OS, então ele quase sempre funciona no Mac OS.

Mas executar o POX somente não faz muita coisa - as funcionalidades do POX são providas pelos seus componentes (o POX vem com componentes úteis, mas o público alvo do POX são as pessoas que querem desenvolver elas mesmas seus componentes). O POX é considerado para fins de pesquisa e ensino no desenvolvimento de aplicações SDN.

2.3.2.5 Ryu

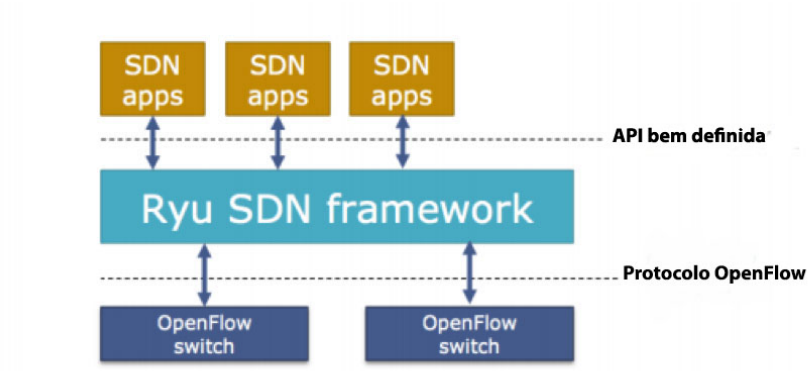


Figura 10 – Arquitetura Ryu.

Fonte: (Ryu SDN Framework Community, 2017)

O Ryu é uma estrutura de rede definida por software baseada em componentes. Como mostrado na Figura 10, o Ryu fornece componentes de software com APIs bem definidas que facilitam a criação de novos aplicativos de gerenciamento e controle de rede. Além de suportar vários protocolos para gerenciar dispositivos de rede, como OpenFlow, Netconf, OF-config, etc. Sobre o OpenFlow, o Ryu suporta totalmente as

extensões 1.0, 1.2, 1.3, 1.4, 1.5 e Nicira. Todo o código está disponível gratuitamente sob a licença Apache 2.0. O Ryu é totalmente escrito em Python.

2.4 Segurança

O objetivo da segurança na rede de computadores é manter o ambiente seguro de intrusos, além de manter uma comunicação segura para os usuários. Segundo os autores de (KUROSE; ROSS, 2012), as propriedades desejáveis de comunicação segura podem ser classificadas da seguinte forma:

- **Confidencialidade:** Apenas o remetente e o receptor devem entender o conteúdo da mensagem transmitida.
- **Integridade da mensagem:** Garantia que o conteúdo dos pacotes não seja alterado.
- **Autenticação ponto final:** Tanto o remetente quanto o receptor devem poder confirmar a identidade da outra parte envolvida na comunicação, para confirmar que a outra parte é de fato quem ou o que eles afirmam ser.
- **Segurança operacional:** Quase todas as organizações têm redes conectadas à Internet pública. Portanto, essas redes podem ser comprometidas. Atacantes podem tentar depositar *worms* nos *hosts* da rede, para obter segredos corporativos, mapear configurações de rede e iniciar ataques DoS.

Mesmo com essas propriedades e com mensagens criptografadas, os intrusos podem espionar farejando e registrando mensagens de controle e de dados no canal. Além disso, os atacantes podem fazer modificação, inserção ou exclusão de mensagens ou conteúdo de mensagens (KUROSE; ROSS, 2012).

A segurança da SDN precisa ser incorporada à arquitetura. Além de ser fornecida como serviço para proteger a disponibilidade, integridade e privacidade de todos os recursos e informações conectados. Dentro da arquitetura, é necessário proteger o controlador, pois ele é o cérebro, responsável pelas decisões. Ou seja, é necessário definir um gerenciamento rígido de acesso ao controlador.

Se o controlador SDN estiver indisponível, o mesmo acontece com a rede, havendo a necessidade de se manter a disponibilidade do controlador. Além disso, estabelecer confiança, protegendo as comunicações em toda a rede. Garantindo que os aplicativos carregados e os dispositivos que o controlador SDN gerencia, são entidades confiáveis e que estão operando como deviam.

Independente disso, as soluções para a arquitetura SDN precisam ser projetadas para criar um ambiente mais escalonável, eficiente e seguro. Assim deve ser simples a implementação, o gerenciamento e manter o ambiente SDN altamente dinâmico, tendo custo-benefício, garantindo que a segurança não seja implantada em qualquer lugar e seja seguro para proteger contra as mais recentes ameaças e também as ameaças tradicionais, que são os principais desafios da SDN.

2.5 Redes em Malha Definidas por Software

Tipicamente, o *backbone* de uma WMN é composto de nós sem fio dedicados, chamados de roteadores de malha (MRs). Estes roteadores são configurados em modo *ad hoc*, com acesso à Internet e com uma ou várias interfaces sem fio baseadas em IEEE 802.11 (HUANG et al., 2015).

Os MRs podem ser organizados em qualquer topologia de rede, comunicando-se entre si usando protocolos como o OLSR, BATMAN, AODV e o HWMP. No entanto, as WMNs tradicionais são difíceis de gerenciar e atualizar porque as configurações são feitas manualmente e são propensas a erros. Normalmente leva semanas ou mesmo meses para fornecer novos serviços, realizar testes e prover a garantia dos serviços. Além disso, os roteadores de malha funcionam de forma auto organizada sem uma visão global, levando a uma fraca alocação de recursos de rede e um baixo desempenho, especialmente em redes de grande porte.

Por padrão um *switch* ou roteador consiste em vários elementos firmemente interligados que controlam o encaminhamento dos pacotes no plano de dados, como também no controle das tabelas de encaminhamento no plano de controle. Isso torna os comutadores complexos e difíceis de ampliá-los com novas funcionalidades. O OpenFlow resolve esse problema separando o plano de controle e de dados.

O plano de controle deixa de fazer parte apenas do comutador e passa a ser implementado parcialmente em um servidor remoto que executa um sistema operacional de rede. O plano de dados é abstraído como um fluxo, e para cada tipo de fluxo contém um conjunto de regras para o processamento. Além disso, o OpenFlow define um protocolo e um canal seguro entre o plano de dados e o Sistema Operacional de Rede.

Na arquitetura utilizada neste trabalho, a rede de malha consiste em roteadores de malha sem fio com OpenFlow habilitado. Cada nó é equipado com duas interfaces de rede sem fio para implementar as redes em malha com múltiplas interfaces e múltiplos canais.

Existem trabalhos na literatura que apresentam problemas e soluções para

segurança em SDN e suas integrações com as redes em malha sem fio, como apresentados na seção seguinte.

2.6 Trabalhos Relacionados

Atualmente, existem excelentes documentos de pesquisas e artigos sobre os desafios de segurança da SDN. Pesquisadores consideram que a segurança é melhor incorporada na rede, outros supõem que é melhor incorporada em servidores de armazenamento e outros dispositivos de computação. Independentemente disso, as soluções precisam ser projetadas para criar um ambiente mais escalonável, eficiente e seguro. Devido a esses desafios, os pesquisadores têm se esforçado cada vez mais para propor soluções e achar problemas de segurança da arquitetura SDN.

Em (CENTENO et al., 2016), os autores apresentam uma análise de segurança da SDN sobre o protocolo OpenFlow. Em paralelo, o trabalho dos autores propõe uma análise das vulnerabilidades através de um ambiente virtual, no simulador *mininet*. O *mininet* cria uma rede virtual realista em um kernel real, podendo interagir com uma rede física. Mas os autores argumentam que não foi possível criar um ambiente misto com o *mininet*. A parti das simulações realizadas nesse trabalho, em um ambiente virtual, é apresentado uma lista de vulnerabilidades baseadas no protocolo OpenFlow.

Além disso, é apresentado uma topologia simples, utilizando o comutador *Open vSwitch* nos experimentos. A proposta do trabalho é realizar ataques tradicionais, no qual se obteve sucesso em todos, como está apresentado nos resultados. Um dos problemas apontados pelos autores é a falta da obrigatoriedade do uso do TLS. Os autores de (CENTENO et al., 2016) acham que é intencional devido ao fato de baratear os equipamentos. Segundo os autores de (CENTENO et al., 2016) as facilidades oferecidas pelo paradigma de SDNs aos administradores de rede, expõem falhas de segurança decorrentes de vulnerabilidades oriundas de características da especificação do protocolo OpenFlow. No final, é mostrado as vulnerabilidades que podem ser exploradas no plano de dados e controle. Essas vulnerabilidades são devido a facilidade de gerenciamento da rede que é proposta pela SDN.

Assim, com os desafios de segurança da SDN, os pesquisadores da areá sempre buscam novas soluções para as vulnerabilidades da mesma. Os autores de (HUSSEIN et al., 2016) apresentam a proposta de usar um plano de segurança, processando os serviços da SDN, prevenindo os ataques de DDoS e identificando a origem dos ataques de *spoofing*.

O funcionamento do plano de segurança, utiliza um agente de software SDN

residindo ao lado do comutador OpenFlow. Uma regra é inserida em cada switch para encaminhar o tráfego para o agente independente de toda a rede. O agente, por outro lado, estabelece uma conexão com o controlador através de uma porta diferente do que a do plano de controle usa.

O módulo de segurança é responsável por analisar e coletar dados, de todo o fluxo vindo dos agentes, verificando eventos anormais. Os autores implementaram o módulo de segurança separado para verificar sua corretude. Posteriormente realizando o teste da conectividade entre as camadas. No final do artigo foi concluído que esse plano de segurança é eficaz para implementar serviços sem sobrecarregar o controlador. Além de prevenir os ataques de DDos, ele também consegue identificar a origem do mesmo.

Já alguns pesquisadores, como os de (BRANDT et al., 2014) fizeram uma análise de vulnerabilidade com o protocolo OpenFlow, OFConfig e OVSDB usando a modelagem de ameaças STRIDE da Microsoft. O STRIDE faz uma análise completa dos processos, fluxo de dados e armazenamento de dados. Lembrando que o STRIDE não prova as vulnerabilidades, apenas mostra as vulnerabilidades presentes. O autor aconselha o uso da TLS, devido a sua segurança. Os protocolos analisados podem se tornar vítimas de ataques de negação de serviço. Se um invasor obtiver acesso ao canal de comunicação, ele pode ficar indisponível simplesmente inundando o canal de comunicação ou explorando falhas em os processos que implementam o protocolo.

O trabalho de (MORAES et al., 2013) apresenta SDMWN (Software-Defined Multihop Wireless Networks), utilizando roteadores D-Link com duas interfaces. A segunda interface era uma antena USB (Universal Serial Bus). Configurando cada uma das interface, sendo a interna com o HWMP (Hybrid Wireless Mesh Protocol) e a externa com o OLSR (Optimized Link State Routing). O HWMP fica responsável pelo encaminhamento do pacote e o OLSR que é o Ad Hoc é por onde todo o tráfego de dados da rede passa e será encaminhado. Assim, é criado um ambiente real, para realizar os testes de desempenho.

Segundo os autores de (MORAES et al., 2013), a SDMWN é uma proposta totalmente viável para experimentos, inclusive em cenários de aplicações exigentes, como VoIP. Para isso, basta que no cenário apresentado possa-se definir facilmente a topologia de encaminhamento fim a fim, através dos comutadores sem fio OpenFlow.

Embora existam diversos trabalhos que envolvam análise de protocolos, poucos são realizados em ambientes reais. Neste presente trabalho, é apresentada uma análise sobre segurança em SDNs baseadas no protocolo OpenFlow, em um ambiente real. Os resultados são de importância para contribuir com a comunidade acadêmica, apontando problemas de segurança da arquitetura SDN e propondo soluções.

3 PROPOSTA

Esta seção descreve a metodologia experimental empregada na análise de segurança da SDN. São apresentados os componentes de rede dos cenários de avaliação, bem como as principais entidades de software responsáveis pelo gerenciamento, controle e adaptação da comunicação entre as camadas. O intuito deste trabalho é fazer uma análise de segurança e propor alternativas para solucionar os problemas observados. Além disso, discutir sobre o impacto das vulnerabilidades da arquitetura SDN sobre os seus principais componentes, o controlador e o comutador OpenFlow. O ambiente do cenário experimental é apresentado na Seção 3.1 e descreve os aspectos do cenário de rede e configuração dos equipamentos. Por fim, nas seções 3.4.1 e **??**, são comentadas as estratégias para a análise de segurança da arquitetura SDN, utilizando ataques tradicionais.

3.1 Cenário Experimental

Nesta proposta, o cenário da rede foi disponibilizado pela comunidade acadêmica do IFCE (Instituto Federal de Educação, Ciência e Tecnologia do Ceará) - Campus Aracati, especificamente pelo LAR (Laboratório de Redes de Computadores e Sistemas). Para os experimentos apresentados neste trabalho, foram utilizados seis roteadores *TP-LINK* modelo TL-WR842ND (configurações: Tabela 5), um *Raspberry pi v3 - B*(Configurações: Tabela 8), um *Notebook* (Configurações: Tabela 6) e um *Desktop* (Configurações: Tabela 7).

O cenário de experimentos é composto por:

Tabela 3 – Componentes da arquitetura SDN

Raspberry pi 3	Controlador
TP-LINK	Comutador Openflow
Notebook	Atacante1
Desktop	Atacante2

Respectivamente seguido da configuração do cenário na Tabela 4, o cenário experimental é apresentado na Figura 11. Apenas o controlador utiliza cabo de rede para se conectar ao comutador OpenFlow. Os comutadores OpenFlow, possuem duas interfaces, uma delas USB, essas duas interfaces servem para interligar os roteadores em malha, utilizando o protocolo HWMP, criando a rede de malha sem fio.

A associação entre eles é garantida através da configuração das interfaces

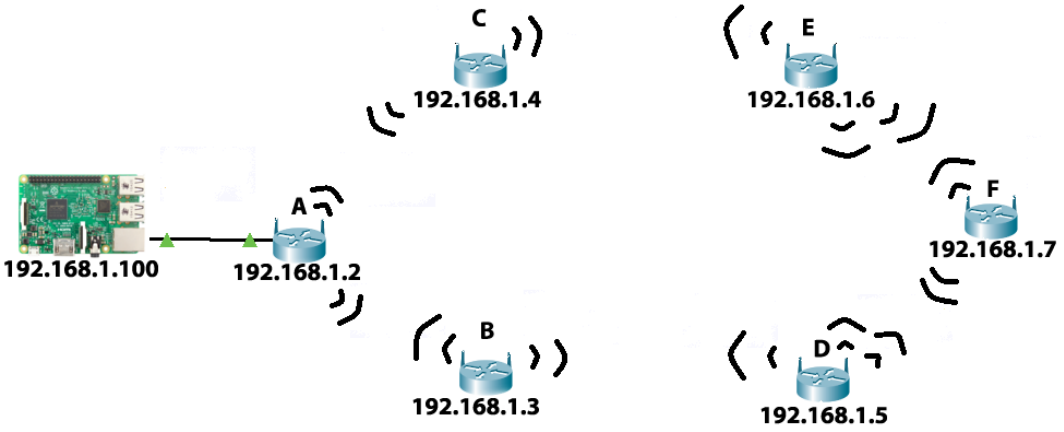


Figura 11 – Cenário sem hosts maliciosos.

Fonte: Elaborada pelo autor

Tabela 4 – Configuração do cenário SDN

Comutadores	IP	Mesh id (WLAN0)	Canal (WLNA0)	Mesh id (WLAN1)	Canal (WLAN1)
A	192.168.1.2	mesh0	11	mesh	1
B	192.168.1.3	mesh1	6	mesh	1
C	192.168.1.4	mesh0	11	mesh2	6
D	192.168.1.5	mesh1	6	mesh3	11
E	192.168.1.6	mesh3	11	mesh2	6
F	192.168.1.7			mesh3	11

Fonte: Elaborada pelo autor

com o mesmo *mesh_id* e o mesmo canal. O comutador F é o único que possui um AP na segunda interface, a outra está associada a outros dois comutadores, como é mostrado na Tabela 4. Para o controlador OpenFlow tomar as decisões é necessário que todos os comutadores estejam apontando para o endereço IP do controlador. Assim, o controlador reconhece cada um dos comutadores OpenFlow e atualiza a tabela de fluxo dos mesmos.

A tabela de fluxo feita pelo controlador é enviada para os comutadores OpenFlow, fazendo com que todos conheçam suas rotas. Sempre que um novo comutador é inserido, o controlador solicita suas informações e em seguida atualiza a tabela de fluxo e atualiza em todos os roteadores.

Os dois hosts maliciosos da rede utilizam sistemas operacionais diferentes. No caso desses hosts, um deles é *desktop* e outro *notebook*. O *desktop* utiliza o

Interfaces:	1 Placa de rede sem fio 4 Portas LAN 10/100Mbps 1 Porta WAN 10/100Mbps 1 Porta USB 2.0
Antenas:	2 antenas externas destacáveis de 5 dBi (RP-SMA)
Padrões Wireless:	IEEE 802.11n, IEEE 802.11g, IEEE 802.11b
Arquitetura:	MIPS 74Kc V4.12
Velocidade da CPU:	535 MHz
Tamanho da memória flash:	8 MB
Tamanho da memória RAM:	32 MB
System-On-Chip:	Atheros AR9341
RAM chip:	Winbond W9425G6JH-5

Tabela 5 – Especificações do roteador TP-LINK modelo TL-WR842ND.

Fonte: (FILHO, 2017)

sistema operacional (S.O) Windows 7 e o notebook utiliza o S.O Linux (Ubuntu). Os mesmos usam as seguintes configurações:

Interfaces:	1 Ethernet 1 AR9565 Wireless Network Adapter 2 USB 2.0 1 USB 3.0
CPU:	Intel Core i5-4210U CPU @ 1.70GHz
Tamanho da memória RAM:	8GB

Tabela 6 – Especificações do Notbook (Host Malicioso)

Interfaces:	1 Ethernet 1 USB Wireless Adapter 5 USB 2.0 3 USB 3.0
CPU:	Intel Core i7-4790 CPU @ 3.60GHz
Tamanho da memória RAM:	8GB

Tabela 7 – Especificações do Desktop (Host Malicioso)

3.2 OpenWrt

O OpenWrt *Project* é um sistema operacional Linux voltado para dispositivos embarcados. Ao invés de tentar criar um único *firmware* estático, o OpenWrt fornece um sistema de arquivos totalmente gravável com gerenciamento de pacotes. Isso

dispensa a necessidade da seleção e configuração do aplicativo fornecida pelo fabricante e permite personalizar o dispositivo por meio do uso de pacotes para atender a qualquer aplicativo.

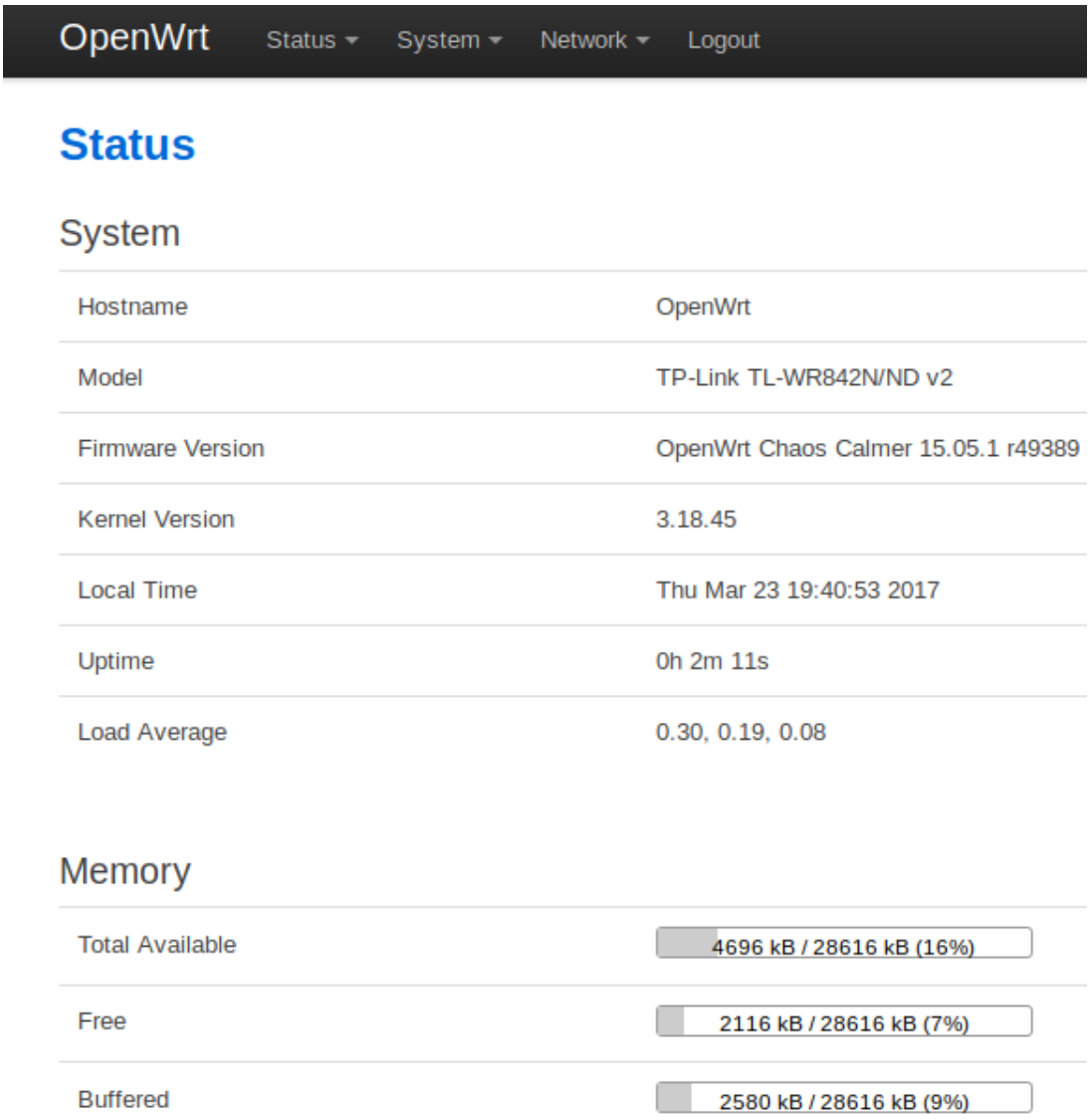


Figura 12 – Interface Web (LuCI) do sistema embarcado OpenWrt

Fonte: (OPENWRT, 2018)

Para a comunidade de desenvolvimento, o OpenWrt é a estrutura para construir um aplicativo sem ter que criar um *firmware* completo em torno dele, para os usuários, isso significa a capacidade de personalização completa, para usar o dispositivo de maneiras nunca imaginadas (OPENWRT, 2018). Além disso, os usuários instalam o OpenWrt e removem o *firmware* padrão, devido ao fato dele ser mais estável, oferecer mais recursos, maior segurança e melhor suporte, graças a sua comunidade grande e forte. (OPENWRT, 2018)

Essa é uma das principais justificativas para se utilizar o OpenWrt embarcado nos roteadores deste projeto, além da robustez dele. O OpenWrt é configurado usando uma interface de linha de comando (*shell script*) via SSH (*Secure Shell*) ou uma interface da web (LuCI), como mostrado na Figura 12.

3.3 Controlador POX

O controlador POX é utilizado no experimento, pois é usado para fins de pesquisa e ensino no desenvolvimento de aplicações SDN, além da facilidade para adicionar funcionalidades. Por ser um controlador componentizado, ele permite a criação de componentes por parte dos próprios desenvolvedores.

O POX requer o Python 2.7 na prática, mas também pode ser executado com o Python 2.6. Alguns *commits* de melhorias no controlador foram feitos por volta de março de 2013. Atualmente, não existe equipe de manutenção e suporte ao POX.

```
pi@raspberrypi:~/pox $ ./pox.py --verbose samples.pretty_log openflow.discovery
 openflow.spanning_tree --no-flood --hold-down
POX 0.5.0 (eel) / Copyright 2011-2014 James McCauley, et al.
[openflow.spanning_tree ] Spanning tree component ready
[core                   ] POX 0.5.0 (eel) going up...
[core                   ] Running on CPython (2.7.13/Sep 26 2018 18:42:22)
[core                   ] Platform is Linux-4.14.98-v7+-armv7l-with-debian-9.8
[core                   ] POX 0.5.0 (eel) is up.
[openflow.of_01         ] Listening on 0.0.0.0:6633
[openflow.of_01         ] [00-23-20-40-18-5e 1] connected
[openflow.discovery     ] Installing flow for 00-23-20-40-18-5e
[openflow.spanning_tree ] Disabling flooding for 3 ports
```

Figura 13 – Exemplo do comando para iniciar o controlador OpenFlow.

Fonte: Elaborada pelo autor

O POX foi baixado a partir do repositório oficial (https://github.com/noxrepo/pox), sendo utilizado os componentes do próprio POX. Executando o seguinte comando na *Raspberry*: *'./pox.py –verbose samples.pretty_log openflow.discovery openflow.spanning_tree –no-flood –hold-down'* como mostra a Figura 13. A partir da execução desse comando, o *Raspberry pi 3* torna-se o controlador.

O comando '–verbose' é para exibir informações extras, especialmente úteis para depurar problemas de inicialização, o 'samples.pretty_log' é um módulo simples que usa cores no log e um formato de log personalizado para fornecer uma saída de log funcional e agradável no console, já o 'openflow.discovery' envia mensagens LLDP(*Link Layer Discovery Protocol*) especialmente criadas a partir de switches OpenFlow para que ele possa descobrir a topologia da rede. Ele gera eventos quando os links estão ativos ou desativados. O 'openflow.spanning_tree –no-flood –

hold-down' depende do 'openflow.discovery' para criar uma exibição da topologia de rede, constrói uma árvore de abrangência e em seguida desativa a inundação em portas do comutador que não estão na árvore. O resultado é que as topologias com *loops* não causa problemas na rede.

A Tabela 8 mostra as configurações do controlador:

Interfaces:	1 2.4GHz e 5GHz IEEE 802.11.b/g/n/ac wireless LAN 1 Bluetooth 4.2, BLE 1 Gigabit Ethernet 4 portas USB 2.0
Velocidade da CPU:	Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
Tamanho da memória RAM:	1GB LPDDR2 SDRAM

Tabela 8 – Especificações do Controlador Raspberry pi 3 modelo B.

3.4 Ataques de Rede Realizados

Está seção apresenta os ataques realizados neste trabalho e mostra as principais ferramentas utilizadas para a realização desses ataques no cenário proposto pelo trabalho.

3.4.1 Negação de Serviço

Os desafios da segurança relacionados ao controlador referem-se principalmente às vulnerabilidades no nível do controlador. Neste tipo de ataque os invasores podem comprometer a rede, considerando a disponibilidade.

A programabilidade dos controladores SDN apresenta um grande desafio. À medida que o SDN separa o plano de controle do plano de dados, para permitir que um controlador centralizado cuide de todos os fluxos de rede recebidos, o próprio controlador provavelmente se torna um gargalo fundamental para a SDN. Isso faz com que ele vire um alvo importante para ataques diversos, como ataques de inundação e ataques de negação de serviço (DoS) (LI; MENG; KWOK, 2016).

Assim, foram adicionado dois hosts maliciosos ao cenário como se vê na Figura 15. Estes hosts maliciosos fizeram um ataque DoS usando a ferramenta LOIC (*Low Orbit Ion Cannon*), disponível no repositório sourceforge (KAZFL, 2018). Esta ferramenta possui uma interface simples, como mostrado na Figura 14. O LOIC executa um DoS ou, quando usados simultaneamente por muitas pessoas, um DoS distribuído (DDoS). O alvo é inundado com pacotes de requisição TCP ou UDP com a intenção

de sobrecarregar o servidor, fazendo com que ele deixe de responder às requisições legítimas.

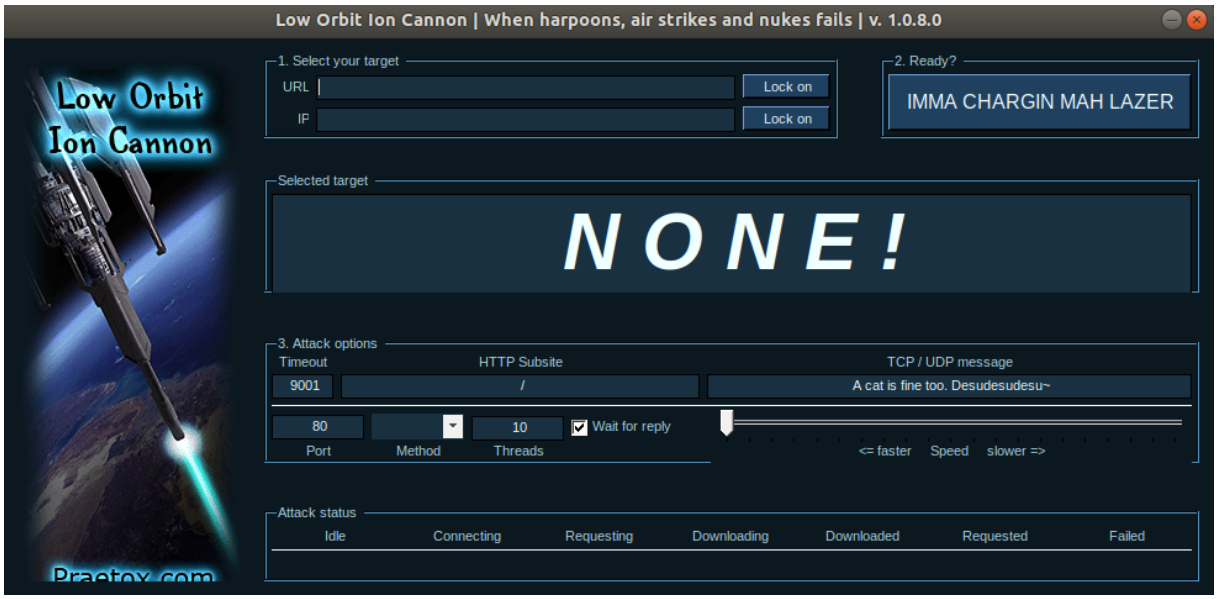


Figura 14 – Ferramenta de Dos ou DDoS

Fonte: (KAZFL, 2018)

A negação de serviço foi feita utilizando dois *hosts* maliciosos conectados pelo AP, atacando o controlador. Outro cenário foi feito atacando o comutador OpenFlow como mostra a Figura 16. No primeiro cenário, os *hosts* estão conectados via WI-FI ao AP. Em seguida, é iniciado o LOIC e configurado para o endereço do controlador (192.168.1.100), usando o pacote UDP e utilizando 100000 *threads* e enviando os pacotes na porta 80. Assim, os dois estão mandando todos os pacotes gerados por eles para o controlador ocasionando o fluxo da Figura 15.

Com o fluxo gerado pelos atacantes, o controlador deve começar a descartar pacotes e parar seus serviços, até ficar totalmente indisponível. O outro cenário também possui dois *hosts* maliciosos atacando o comutador OpenFlow, objetivando a parada dos serviços oferecidos pelo comutador.

O ataque partiu dos dois *hosts* maliciosos, direcionado para o comutador OpenFlow (192.168.1.4), como mostra a Figura 16. Assim, o comutador recebe um grande fluxo de pacotes, ocasionando o problema de estouro da fila. Consequentemente, o comutador começa a descartar os pacotes e parar os seus serviços.

3.4.2 Sniffing

O *Sniffing* é um tipo de ataque considerado bem desafiador para a segurança da comunicação. Esta é uma das formas mais populares utilizadas pelos invasores.

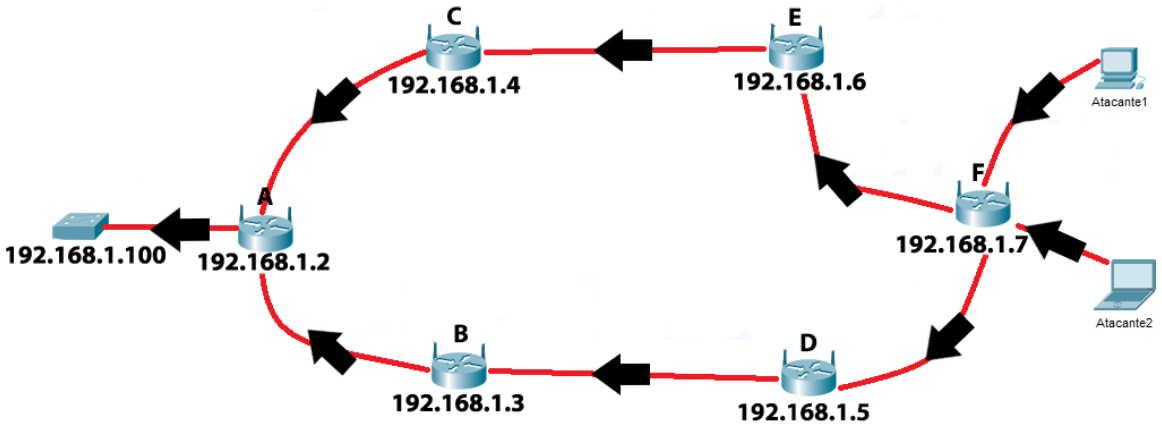


Figura 15 – Cenário de negação de serviço atacando o controlador

Fonte: Elaborado pelo autor

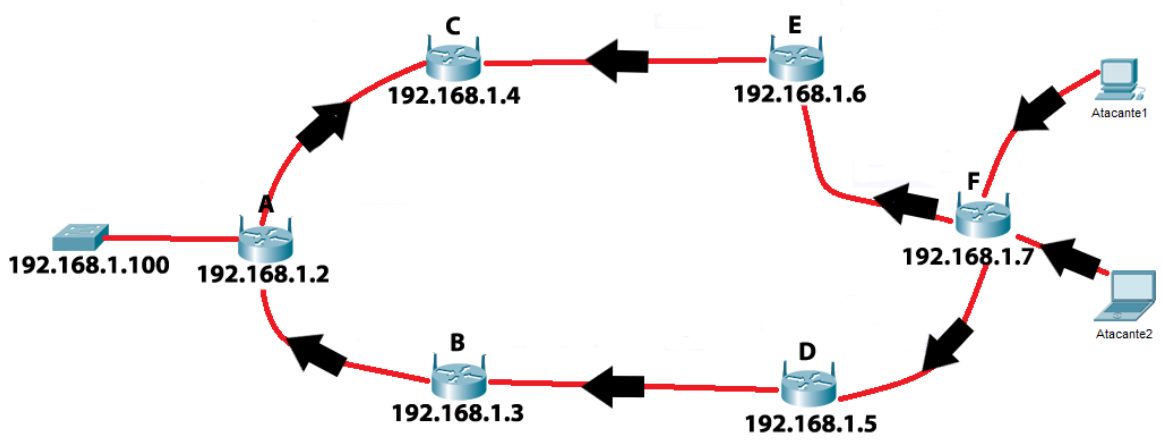


Figura 16 – Cenário de negação de serviço atacando o comutador

Fonte: Elaborado pelo autor

Basicamente, os atacantes capturam e analisam os dados de comunicação da rede, de forma silenciosa e de difícil detecção.

Inimigos que utilizam a técnica *Sniffing* podem escutar os dados da comunicação dos nós da rede ou links. Ademais, isso significa roubar dados confidenciais, como nomes de usuários, senhas e outras informações (ZHAO et al., 2016).

O principal problema da SDN está a nível de canal de comunicação. Assim, os invasores podem comprometer a arquitetura da SDN. É observado que isso interfere na comunicação entre os componentes e os administradores. No OpenFlow, o mecanismo de comunicação entre controladores e interruptores abre um brechas para que os invasores comprometam a segurança por interceptação (LI; MENG; KWOK, 2016).

Para analisar essa vulnerabilidade na rede, foi utilizado o Aircrack-ng (AIRCRACK-NG, 2009). O Aircrack-ng é um conjunto completo de ferramentas para avaliar a segurança das redes sem fio, executando o ataque de Sniffing.

Todas as suas ferramentas são usadas através da linha de comando, permitindo o uso de *scripts* pesados. Muitas GUIs (*Graphical User Interface*) aproveitaram esse recurso. Funciona principalmente no Linux, mas também Windows, Mac OS X, FreeBSD, OpenBSD, NetBSD, Solaris e até mesmo eComStation 2. As principais áreas afetadas por este software são as seguintes:

- **Monitoramento:** captura de pacotes e exportação de dados para arquivos de texto para processamento adicional por ferramentas de terceiros.
- **Ataque:** ataques de repetição, desautenticação, pontos de acesso falsos e outros via injeção de pacotes.
- **Teste:** Verificação de cartões WiFi e capacidades de driver (captura e injeção).
- **Craqueamento:** Aplicado em chaves WEP e WPA PSK (WPA 1 e 2).

O uso desta ferramenta para sniffar a rede é bastante simples. Basta colocar a placa *wireless* em modo monitor e, assim, ela vai enxergar todas as redes *wireless*. Em seguida, deve-se selecionar a rede com seu BSSID, bem como o canal para começar a captura de pacotes da rede. Os pacotes capturados são colocados em um arquivo de extensão '.cap' ou '.csv'.

O alvo do cenário experimental foi o comutador (192.168.1.2), utilizando o comando 'airmon-ng start wlp6s0' para colocar a placa *wireless* em modo monitor. Nesta ocasião é preciso listar todas as redes que a placa *wireless* identificou, por meio do comando 'airodump-ng wlp6s0mon'.

Com a placa em modo monitor e listadas as redes que ela enxerga, basta selecionar alguma da lista, pegando seu BSSID e seu canal. Neste momento, para capturar os pacotes, foi executado o comando 'airodump-ng –bssid 14:CC:20:77:A8:8A –channel 1 –write meshpacket wlp6s0mon'. Esse comando faz com que todos os pacotes que passam pelo BSSID sejam capturados e, assim, são registrados no arquivo meshpacket.

4 RESULTADOS

Neste Capitulo são apresentados e discutidos os principais resultados obtidos a partir de experimentos extensivos realizados na rede da arquitetura SDN apresentada na Figura 11. Aqui, foram destacados os resultados mais relevantes, que apresentam situações que são os principais desafios de segurança em SDNs. Os experimentos foram divididos em dois, Negação de Serviço e Sniffer, analisando as vulnerabilidades de disponibilidade e confidencialidade da rede SDN em malha.

Os resultados da negação de serviço foram obtidos a partir de dois ataques, um direcionado ao controlador e o outro direcionado ao comutador Openflow. O Sniffer foi realizado atacando um comutador OpenFlow e capturando os seus pacotes. Os experimentos foram repetidos durante uma semana, a partir das 7:00 AM ás 10:00 PM. Os resultados apresentados consistem em uma análise de ataques de negação de serviço na seção 4.1 e na 4.2 fala sobre a captura de pacotes realizada.

4.1 Negação de Serviço

A negação de serviço é uma tentativa de tornar os recursos de um sistema indisponíveis para os seus utilizadores. Os ataques de negação de serviço são feitos geralmente de duas formas:

- Forçar o sistema vítima a reinicializar ou consumir todos os recursos (como memória ou processamento, por exemplo), de forma que ele não possa mais fornecer seu serviço;
- Obstruir a mídia de comunicação entre os utilizadores e o sistema da vítima, de tal forma que não se comunique adequadamente.

O primeiro ataque de Negação de Serviço tinha como objetivo atacar o controlador, fazendo com que seus serviços ficassem lentos e parassem, consumindo toda sua memória e processamento. O ataque foi feito internamente por dois *hosts* maliciosos.

Como apresentado na Figura 17, que pertence ao log do AP, o grande fluxo de pacotes estava parando antes de chegar à controladora devido ao hardware dos comutadores OpenFlow serem de baixa qualidade. O fluxo de pacotes direcionado para o controlador também estava passando pelo AP e preenchendo a fila de envio, acarretando no estouro da memória. Após isso, o AP começou a descartar os pacotes.

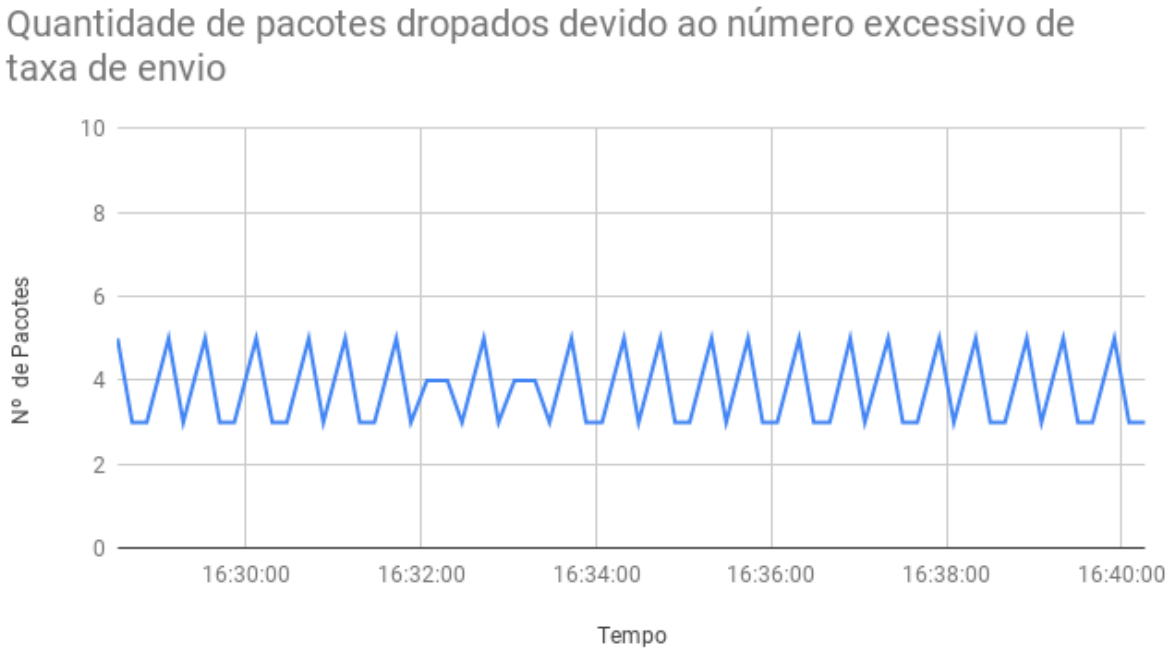


Figura 17 – Quantidade de pacotes descartados devido ao número excessivo de taxa de envio

Fonte: Elaborado pelo autor

O grande problema foi o AP ter recebido uma grande quantidade de pacotes e não ter conseguido processar todos. Então, começou a descartar os pacotes e seus serviços ficaram lentos. Após isso, a interface que funcionava como AP caiu, sendo preciso reiniciar o AP.

O ataque de negação de serviço foi executada com sucesso, mas não atingindo o alvo esperado que era o controlador da SDN. O mesmo problema aconteceu com o segundo experimento de negação de serviço que tinha como finalidade atacar um comutador OpenFlow. O que ocorreu foi sobrecarregamento do AP novamente.

4.2 Sniffing

Os programas de *Sniffing* podem detectar informações enviadas ou recebidas pelo computador. Estas informações podem incluir nomes de usuário e senhas ou outros dados de formulários enviados, como aniversários, números de segurança social ou informações de cartão de crédito.

O *Sniffing* é um ataque passivo, podendo evoluir para outros ataques dependendo das informações contidas dentro dos pacotes capturados. A ideia é apenas capturar dados da rede. No caso, este experimento foi utilizado o *Sniffing* Aircrack-ng

e foi selecionado o comutador para capturar os pacotes.

Como mostra o gráfico da Figura 18, foram capturados os pacotes que passavam pelo comutador. Esses pacotes contêm informações importantes sobre a rede, o que abre oportunidade para outros tipos de ataque.

Não houveram problemas com as capturas de pacotes, já que realizar a captura de pacotes é um trabalho silencioso e difícil de ser notado. Além disso, bastante perigoso se explorado de forma mais profunda. Como pode ser observado, o comutador que possui o maior número de pacotes é o comutador alvo, o qual foi selecionado para a captura de pacotes.

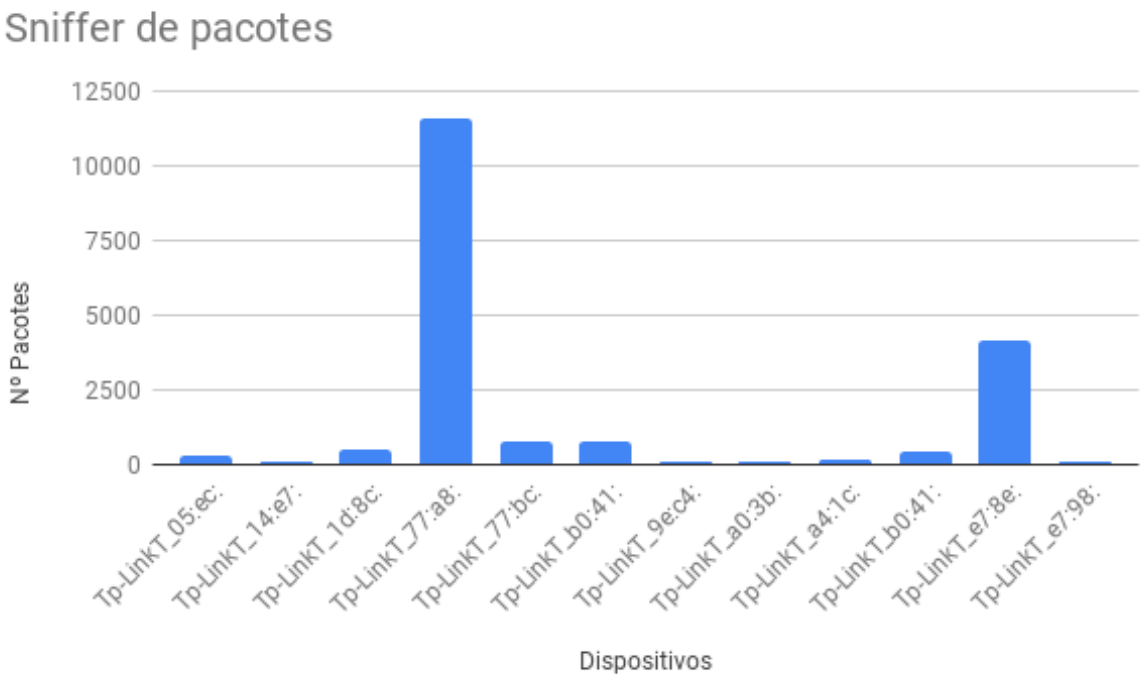


Figura 18 – Quantidade de pacote capturados

Fonte: Elaborado pelo autor

Os resultados mostraram que o equipamento de baixa qualidade para análise de segurança ocasionou problemas no teste de negação de serviço. Devido ao ponto de acesso não ter suportado a quantidade de pacotes que estavam sendo encaminhados, ele parou de funcionar, ocasionando a negação de serviço para os usuários do ponto de acesso. É importante ressaltar que quanto melhor a qualidade do equipamento e seu desempenho mais difícil se tornará de ocorrer problemas desse tipo.

O segundo experimento oferece mais riscos que o anterior, mas não devido ao que ele faz e sim nas oportunidades que ele fornece ao atacante. É fácil realizar captura de pacotes que utilizam tecnologia sem fio devido ao fato de eles estarem

espalhados no ar.

5 CONCLUSÃO

Neste trabalho, foi apresentada uma análise de segurança em SDN baseada no protocolo OpenFlow, utilizando-o para controle dos equipamentos. Dado o estado da arte da tecnologia, as redes em malha podem simplesmente ser integradas e implementadas em diversos equipamentos de rede (FILHO, 2017).

A análise foi feita em um ambiente real, utilizando roteadores de baixo custo com o *firmware* OpenWrt embarcado. Além disso, foi adotado o microcomputador Raspberry pi 3 para ser o controlador, ou seja, o cérebro da rede.

Segundo (CENTENO et al., 2016), observa-se que as facilidades oferecidas pelo paradigma de Redes Definidas por Software aos administradores de rede expõem falhas de segurança decorrentes de vulnerabilidades oriundas de características da especificação do protocolo OpenFlow. Isso é devido a componentização do controlador, tornando a arquitetura mais fácil de se controlar pelo administrador.

Ataques em redes sem fio se tornam mais fáceis de paralisar os comutadores do que o próprio controlador, devido ao fato de todos os pacotes terem que passar pelos comutadores antes de chegar ao controlador. Assim, o experimento de negação de serviço realizado neste trabalho, quebrou um dos elementos da tríade da SDN, afetando a disponibilidade da rede e derrubando o AP.

Como conclusão do segundo ataque, foi realizada com sucesso a captura dos pacotes. A partir disto, abriram-se outras oportunidades para a captura da senha do AP. Desta forma, é possível conseguir as configurações da rede em malha sem fio, podendo instalar comutadores maliciosos.

Como pode ser visto na Figura 15, o ataque DoS tem que passar por comutadores antes de chegar ao controlador. O problema do DoS em uma rede de malha definida por software é os comutadores, eles serão gargalos na rede assim os pacotes que tem destino o controlador irão para antes de chegar nele.

O mesmo problema serve para a prática do ataque DDoS, o que irá tornar o DDoS funcional é utilizar vários pontos na rede sem causar gargalos, mandando todo o fluxo para o controlador, assim acabaria realizando a negação de serviço. Um ataque direto teria maior probabilidade de êxito, mas em uma rede de malha sem fio definida por software não é possível.

Esse trabalho apresentou os problemas de vulnerabilidade na arquitetura SDN baseada no protocolo OpenFlow. A sua principal contribuição está voltada para a comunidade acadêmica, com informações importantes na área de segurança da arqui-

tetura SDN, integrada com as redes em malha sem fio.

Para trabalhos futuros, propõem-se o desenvolvimento de um plano de segurança em ambiente real. Esse plano de segurança ficaria acima do de controle, existindo um direcionamento de fluxo para ele, antes de passar pelo controlador. O plano de segurança faria a análise dos pacotes e encaminharia para o plano de controle. Esse plano de segurança seria encarregado de prevenir ataques DDoS e identificar a origem dos ataques de spoofing.

REFERÊNCIAS

AIRCRACK-NG. *Aircrack-ng*. 2009. Aircrack-ng. Disponível em: <https://www-.aircrack-ng.org/>. Acesso em: 10.04.2019. Citado na página 50.

AL-SAADI, A. et al. Routing protocol for heterogeneous wireless mesh networks. *IEEE Transactions on Vehicular Technology*, v. 65, n. 12, Dec 2016. ISSN 0018-9545. Citado na página 30.

ALENCAR, M. S. de. *Telefonia digital*. [S.l.]: Ed. Érica, 2000. Citado na página 27.

BENTON, K.; CAMP, L. J.; SMALL, C. Openflow vulnerability assessment. In: ACM. *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*. [S.l.], 2013. p. 151–152. Citado 2 vezes nas páginas 16 e 33.

BRANDT, M. et al. *Security analysis of software defined networking protocols - OpenFlow, OF-Config and OVSDB*. 2014. Citado na página 41.

CAMPOS, M. B. Um ambiente flexível para detecção e prevenção flexível de ataques em redes openflow/sdn. Universidade Salvador, 2017. Citado 2 vezes nas páginas 32 e 33.

CENTENO, P. V. et al. Uma análise de segurança de redes definidas por software sobre protocolo openflow. Florianópolis, SC, 2016. Citado 4 vezes nas páginas 16, 34, 40 e 55.

DELY, P.; KASSLER, A.; BAYER, N. Openflow for wireless mesh networks. In: IEEE. *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*. [S.l.], 2011. p. 1–6. Citado na página 16.

FARIAS, F. N. et al. Pesquisa experimental para a internet do futuro: Uma proposta utilizando virtualização e o frame-work openflow. *XXIX Simpósio de Redes de Computadores e Sistemas Distribuídos-SBRC*, p. 18, 2011. Citado na página 16.

FILHO, M. et al. Performance issues in a low cost multi-channel multi-interface wireless mesh network. In: *2018 IEEE Symposium on Computers and Communications (ISCC)*. [S.l.: s.n.], 2018. p. 00792–00795. ISSN 1530-1346. Citado na página 27.

FILHO, M. A. D. C. D. S. Uma análise de desempenho do openflow em relação aos principais protocolos de roteamento de redes em malha sem fio. 2017. TCC (Bacharel em Ciência da COmputação), IFCE (Instituto Federal do Ceará), Aracati, Brazil. Citado 2 vezes nas páginas 44 e 55.

FOSTER, N. et al. *Languages for software-defined networks*. [S.l.], 2013. Citado na página 31.

Fundação Linux. *Open vSwitch*. 2016. Open vSwitch. Disponível em: <https://www-.openvswitch.org/>. Acesso em: 10.04.2019. Citado na página 36.

GUEDES, D. et al. Redes definidas por software: uma abordagem sistêmica para o desenvolvimento de pesquisas em redes de computadores. *Minicursos do Simpósio Brasileiro de Redes de Computadores-SBRC*, v. 30, n. 4, p. 160–210, 2012. Citado na página 34.

HAQUE, I. T.; ABU-GHAZALEH, N. Wireless software defined networking: A survey and taxonomy. *IEEE Communications Surveys & Tutorials*, IEEE, v. 18, n. 4, p. 2713–2737, 2016. Citado na página 31.

HUANG, H. et al. Software-defined wireless mesh networks: architecture and traffic orchestration. *IEEE Network*, v. 29, n. 4, p. 24–30, July 2015. ISSN 0890-8044. Citado 2 vezes nas páginas 32 e 39.

HUSSEIN, A. et al. Sdn security plane: An architecture for resilient security services. In: *2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW)*. [S.l.: s.n.], 2016. p. 54–59. Citado na página 40.

IEEE 802.11. Ieee standard for wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-1997*, p. 1–445, Nov 1997. Citado 3 vezes nas páginas 25, 26 e 30.

IEEE 802.11. Ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, p. 1–2793, March 2012. Citado na página 26.

KANDOI, R.; ANTIKAINEN, M. Denial-of-service attacks in openflow sdn networks. In: *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. [S.l.: s.n.], 2015. p. 1322–1326. ISSN 1573-0077. Citado na página 17.

KAZFL. *LOIC*. 2018. LOIC. Disponível em: <https://sourceforge.net/projects/ddos-ip-attack-stress-loic/>. Acesso em: 10.04.2019. Citado 2 vezes nas páginas 47 e 48.

KLOTI, R.; KOTRONIS, V.; SMITH, P. Openflow: A security analysis. In: IEEE. *Network Protocols (ICNP), 2013 21st IEEE International Conference on*. [S.l.], 2013. p. 1–6. Citado na página 33.

KUROSE, J. F.; ROSS, K. W. *Computer Networking: A Top-Down Approach (6th Edition)*. 6th. ed. [S.l.]: Pearson, 2012. ISBN 0132856204, 9780132856201. Citado 12 vezes nas páginas 15, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30 e 38.

LARA, A.; KOLASANI, A.; RAMAMURTHY, B. Network innovation using openflow: A survey. *IEEE communications surveys & tutorials*, IEEE, v. 16, n. 1, p. 493–512. Citado na página 32.

LF, P. *OpenDayLight*. 2018. OpenDayLight. Disponível em: <https://www-.opendaylight.org/>. Acesso em: 10.04.2019. Citado na página 34.

LI, W.; MENG, W.; KWOK, L. F. A survey on openflow-based software defined networks: Security challenges and countermeasures. *Journal of Network and Computer Applications*, Elsevier, v. 68, p. 126–139, 2016. Citado 4 vezes nas páginas 17, 33, 47 e 49.

MACHADO, G. P.; TORRES, M. *Controlador OpenDayLight*. 2018. Grupo de Teleinformática e Automação. Disponível em: <https://www.gta.ufrj.br/ensino/eel879-/trabalhos_vf_2018_2/opendaylight/bibliografia>. Acesso em: 10.04.2019. Citado 2 vezes nas páginas 34 e 35.

MORAES, M. et al. Redes sem fio de multiplos saltos definidas por software. In: *IV Workshop de Pesquisa Experimental da Internet do Futuro*. [S.l.: s.n.], 2013. Citado na página 41.

MORZHOV, S.; ALEKSEEV, I.; NIKITINSKIY, M. Firewall application for floodlight sdn controller. In: IEEE. *2016 International Siberian Conference on Control and Communications (SIBCON)*. [S.l.], 2016. p. 1–5. Citado na página 34.

MUTAHER, H.; KUMAR, P.; WAHID, A. Openflow controller-based sdn: Security issues and countermeasures. *International Journal of Advanced Research in Computer Science*, v. 9, n. 1, 2018. Citado na página 34.

OPENWRT. *OpenWrt Wireless Freedom*. 2018. OpenWrt. Disponível em: <https:/-/openwrt.org/>. Acesso em: 10.04.2019. Citado na página 45.

Ryu SDN Framework Community. *Ryu*. 2017. Ryu. Disponível em: <https://osrg-.github.io/ryu/>. Acesso em: 10.04.2019. Citado na página 37.

SEZER, S. et al. Are we ready for sdn? implementation challenges for software-defined networks. *IEEE Communications Magazine*, IEEE, v. 51, n. 7, p. 36–43, 2013. Citado 2 vezes nas páginas 16 e 32.

SHU, Z. et al. Security in software-defined networking: Threats and countermeasures. *Mobile Networks and Applications*, Springer, v. 21, n. 5, p. 764–776, 2016. Citado na página 17.

SOARES, L. F. G.; LEMOS, G.; COLCHER, S. Redes de computadores: das lans, mans e wans às redes atm. Campus Rio de Janeiro, 1995. Citado na página 27.

WIKIPéDIA, a. e. l. *Software-Defined Networking (SDN) Definition. https://www.opennetworking.org/sdn-definition/. Último acesso em: Julho, 2018*. 2018. Disponível em: <https://www.opennetworking.org/sdn-definition/>. Citado 3 vezes nas páginas 16, 32 e 33.

ZHAO, Z. et al. Sdn-based double hopping communication against sniffer attack. *Mathematical Problems in Engineering*, Hindawi, v. 2016, 2016. Citado na página 49.