UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

LUÍS EDUARDO SIMAN

ANÁLISE DE DESEMPENHO DO PROTOCOLO RAW EM UMA REDE 802.11AH

PONTA GROSSA

2022

LUÍS EDUARDO SIMAN

ANÁLISE DE DESEMPENHO DO PROTOCOLO RAW EM UMA REDE 802.11AH

Performance analysis of the RAW protocol on an 802.11ah network

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

Orientador: Prof. Dr. Augusto Foronda

PONTA GROSSA

2022

LUÍS EDUARDO SIMAN

ANÁLISE DE DESEMPENHO DO PROTOCOLO RAW EM UMA REDE 802.11AH

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

Data de aprovação: 08/11/2022

Augusto Foronda
Doutorado
Universidade Tecnológica Federal do Paraná

Richard Duarte Ribeiro
Doutorado
Universidade Tecnológica Federal do Paraná

Itamar Iliuk
Doutorado
Universidade Tecnológica Federal do Paraná

PONTA GROSSA

2022

Dedico este trabalho à minha família e amigos,
pelo imensurável suporte nesta trajetória.

AGRADECIMENTOS

Em primeiro lugar, agradeço ao meu orientador Prof. Dr. Augusto Foronda pelo suporte e pela compreensão durante todo o processo do Trabalho de Conclusão de Curso.

Agradeço à minha família e meus amigos, que me acompanharam nos muitos altos e baixos durante minha formação.

RESUMO

Redes de computadores crescem e evoluem de acordo com o avanço tecnológico e com as necessidades que surgem em diferentes ocasiões. Redes para dispositivos de Internet das Coisas tem necessidades específicas, considerando que geralmente são amplas, preenchidas com dispositivos com baixa necessidade de energia e especialmente densas, o que pode prejudicar seu desempenho quando as transmissões dos dispositivos interferem umas com as outras. O modelo de rede 802.11ah propõe soluções para este cenário, contando com um protocolo de acesso ao meio preparado para números altos de dispositivos: o RAW (*Restricted Access Window*), que agrupa os dispositivos e limita o acesso ao meio a fim de diminuir o número de colisões. Este trabalho propõe analisar o procolo RAW, focando no impacto destes agrupamentos no desempenho de uma rede. Fazendo simulações e executando conjuntos de experimentos, observa-se que mesmo em redes consideravelmente pequenas, o protocolo tem um efeito visível, e que pode trazer grandes benefícios para redes densamente populadas.

Palavras-chave: redes de computadores; Wi-Fi; 802.11ah; simulação.

ABSTRACT

Computer networks grow and evolve according to technological advances and the needs that arise at different times. Networks for IoT devices have specific needs, as they are often in a large area, filled with low-power devices, and especially dense, which can degrade performance when device transmissions interfere with each other. The 802.11ah network model proposes solutions for this scenario, with a medium access protocol prepared for high numbers of devices: RAW (Restricted Access Window), which groups the devices and limits access to the medium in order to reduce the number of collisions. This work proposes to analyze the RAW protocol, focusing on the impact of these groups on the performance of a network. By doing simulations and running sets of experiments, it is observed that even in considerably small networks, the protocol has a visible effect, and that it can bring great benefits to densely populated networks.

Keywords: computer networks; Wi-Fi; 802.11ah; simulation.

LISTA DE FIGURAS

LISTA DE GRÁFICOS

LISTA DE QUADROS

LISTAGEM DE CÓDIGOS FONTE

LISTA DE ABREVIATURAS E SIGLAS

ALOHA	*Advocates of Linux Open-source Hawaii Association*
CDMA	*Code-division multiple access*
CSMA	*Carrier-sense multiple access*
CSMA/CA	*Carrier-sense multiple access with collision avoidance*
CSMA/CD	*Carrier-sense multiple access with collision detection*
DSSS	*Direct Sequence Spread Spectrum*
FDM	*Frequency-division multiplexing*
FHSS	*Frequency-hopping spread spectrum*
FTP	*File Transfer Protocol*
HDCL	*High-Level Data Link Control*
HTTP	*Hypertext Transfer Protocol*
IEEE	*Institute of Electrical and Electronics Engineers*
IoT	*Internet of Things*
IP	*Internet Protocol*
ISO	*International Standards Organization*
LAN	*Local Area Network*
MIMO	*Multiple-Input and Multiple-Output*
OFDM	*Orthogonal frequency-division multiplexing*
OFDMA	*Orthogonal frequency-division multiple access*
OSI	*Open Systems Interconnection*
PPP	*Point-to-Point Protocol*
RAW	*Restricted Access Window*

SMTP	*Simple Mail Transfer Protocol*
SSH	*Secure Shell*
TCP/IP	*Transmission Control Protocol/Internet Protocol*
TDM	*Time-division multiplexing*
UDP	*User Datagram Protocol*
WSL	*Windows Subsystem for Linux*

SUMÁRIO

1 INTRODUÇÃO

Redes de computadores são conjuntos de dispositivos independentes interconectados, seja por fios de cobre, fibras ópticas, ondas de infravermelho ou qualquer outra tecnologia que permita tal conexão (TANENBAUM, 2011). Dois ou mais computadores são considerados conectados quando eles podem trocar informações, e é tal comunicação que permite que tarefas complexas sejam executadas por computadores em rede e que estruturas robustas, como a Internet, sejam implantadas.

Dentre as possibilidades de redes sem fio está a *Local Area Network* (LAN) 802.11, também conhecida como Wi-fi (KUROSE, 2013). A LAN 802.11 é implementada em diversos padrões, cada um com suas características, como as faixas de frequência em que operam e a taxa de transmissão de dados que são capazes de atingir. Recentemente, foi publicado o padrão de rede IEEE 802.11ah (IEEE. . . , 2017), também chamado de *Wi-Fi HaLow*, que opera em frequências consideravelmente baixas para o padrão 802.11, consome menos energia e possui uma área de operação ampla. Estas características fazem deste padrão de rede ideal para o conceito de *Internet of Things* (IoT), em que é comum o uso um número alto de dispositivos.

Para que mais de dois computadores conectados em um mesmo meio de transmissão troquem mensagens entre si, é necessário implementar mecanismos que impeçam que computadores diferentes transmitam dados em um mesmo instante, o que acarreta na perda das mensagens. Tais eventos são chamados de colisões (COMER, 2009).

O mecanismo que permite que múltiplos computadores conectados a uma rede Wi-Fi no padrão IEEE 802.11 transmitam mensagens em um mesmo meio é chamado *Carrier-sense multiple access with collision avoidance* (CSMA/CA). Para evitar colisões, tanto o computador que inicia a transmissão quanto o receptor devem confirmar que estão prontos para a interação, alertando os computadores a seu alcance de que o meio estará ocupado. Caso outra estação inicie uma transmissão neste meio-tempo, o receptor recusará a segunda mensagem e o computador que iniciou a transmissão aguardará uma quantia de tempo aleatória antes de tentar novamente o envio (COMER, 2009).

O padrão de rede 802.11ah utiliza o protocolo *Restricted Access Window* (RAW) para o controle de acesso ao meio de transmissão. Tal algoritmo separa as estações presentes na rede em grupos, e em qualquer instante de tempo, o acesso ao meio é exclusivo a apenas um grupo.

Este trabalho visa analisar o protocolo de rede RAW, verificando seu desempenho conforme o número de estações e grupos aumenta.

1.1 Objetivos

Esta seção apresenta o objetivo geral e os objetivos específicos deste trabalho.

1.1.1 Objetivo geral

Analisar o algoritmo RAW para redes 802.11ah.

1.1.2 Objetivos específicos

- Estudar a teoria do protocolo RAW;
- Criar simulações representativas do uso do protocolo RAW;
- Medir a vazão alcançada pelo protocolo com diferentes números de grupos;
- Analisar a vazão com o aumento do número de usuários na rede;

2 REFERENCIAL TEÓRICO

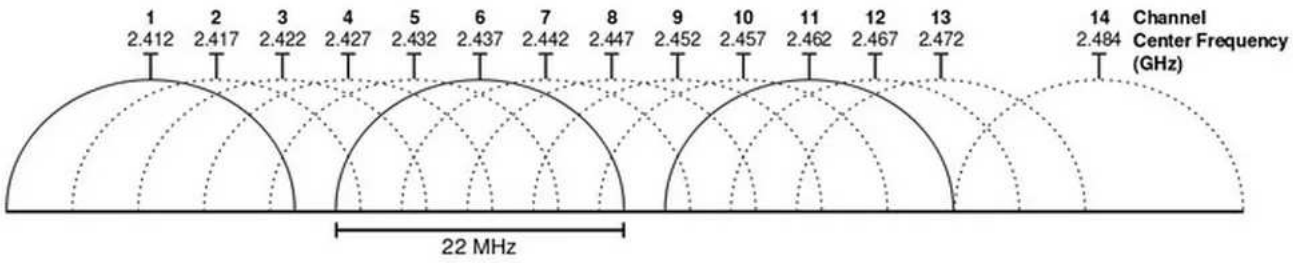
Este capítulo explora e sumariza temas relevantes ao presente trabalho. A seção 2.1 apresenta conceitos gerais sobre a área de redes e mostra alguns termos importantes para compreensão do trabalho. A seção 2.2 apresenta o modelo *Transmission Control Protocol/Internet Protocol* (TCP/IP), explicando suas diferentes camadas e fazendo breves comparações ao modelo *Open Systems Interconnection* (OSI). A seção 2.3 apresenta os responsáveis pela definição de padrões internacionais de rede . A seção 2.4 apresenta considerações gerais sobre redes sem fio e a história e características de padrões de rede convencionais. A seção 2.5 foca em um padrão de rede específico e particularmente relevante para o trabalho. A seção 2.6, apresenta diversas das tecnologias presentes no controle do fluxo de dados em redes. A seção 2.7 fala sobre o simulador de rede utilizado no trabalho.

2.1 Conceitos gerais de redes de computadores

Para o pleno entendimento do trabalho, é necessário compreender o significado de certos termos no contexto em que serão trabalhados:

- **Largura de banda**: O termo largura de banda tem significados diferentes, dependendo do contexto em que é utilizado:
  - **Largura de banda (processamento de sinais)**: Refere-se à diferença entre as frequências mais baixa e mais alta em um intervalo de frequências contínuas. A largura dos canais está relacionada à capacidade de transmissão do meio, sendo canais mais largos capazes de transferir mais dados em um intervalo de tempo. Porém, canais mais largos podem sofrer interferência com a sobreposição à outros canais, perdendo eficiência. A Figura 1 mostra um esquema com os canais de 22MHz da banda de 2,4GHz. Normalmente medido em hertz ou unidade equivalentes.

Figura 1 – Canais da banda 2,4GHz



Fonte: Adaptado de Chieochan (2010)

  - **Largura de banda (computação)**: Refere-se à vazão máxima alcançável em um meio de transmissão. Geralmente medida em bits por segundo ou unidades equivalentes.

- **Vazão**: Valor que representa a quantidade de dados transmitida durante determinado intervalo de tempo. Pode ser utilizada para medir a capacidade real de transmissão em uma rede de computadores (em contraste à taxa de transferência teórica máxima). O cálculo da vazão é extremamente sensível ao contexto em que é aplicado. Diversos fatores afetam a vazão da rede, como a quantidade de tráfego no momento do cálculo ou características do meio de transmissão. Chamada também de taxa de transferência.

2.2 Modelos de referência de rede

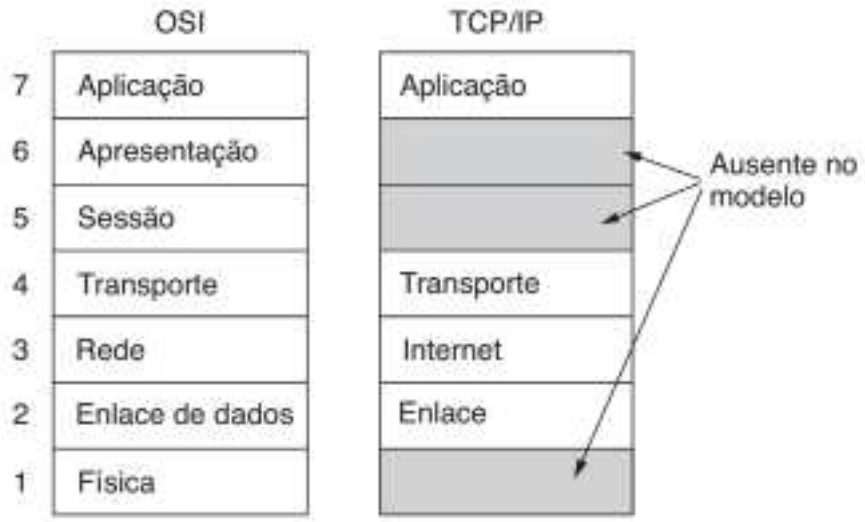
O modelo TCP/IP e o modelo OSI são instâncias dos chamados "Modelos de referência". Ambos são arquiteturas hierárquicas idealizadas para estruturar o software de rede de maneira a diminuir sua complexidade. O papel de cada camada é oferecer serviços às camadas superiores através das interfaces presentes entre camadas adjacentes. A Figura 2 mostra as diferentes camadas do modelo OSI e do modelo TCP/IP. O modelo TCP/IP possui um número menor de camadas. Em comparação com o modelo OSI, pode-se fazer uma relação entre as camadas:

- A camada de enlace do modelo TCP/IP envolve as camadas física e de enlace de dados do modelo OSI;
- A camada de Internet do modelo TCP/IP é relacionada à camada de rede do modelo OSI;
- A cama de transporte do modeo TCP/IP é relacionada à camada de transporte do modelo OSI;
- Por fim, a camada de aplicação do modelo TCP/IP envolve as camadas de sessão, apresentação e aplicação do modelo OSI.

Cada camada oferece algum tipo de serviço, como o envio de mensagens. Para que tal serviço possa ser de fato fornecido, é necessário que existam protocolos que definam as regras de comunicação para os dispositivos envolvidos. Compreendendo a diferença entre serviços e protocolos, é possível observar que o modelo teórico OSI ainda é válido, mesmo que os protocolos do modelo TCP/IP sejam mais utilizados (TANENBAUM, 2011). Implementações reais, como por exemplo a pilha de protocolos da Internet, podem usar modelos híbridos, como o modelo TCP/IP de 5 camadas.

A Figura 3 mostra os protocolos de cada camada do modelo TCP/IP. As camadas superiores possuem protocolos de aplicação, como o *Hypertext Transfer Protocol* (HTTP) para navegação na web e o *Simple Mail Transfer Protocol* (SMTP) para envio de e-mail, enquanto

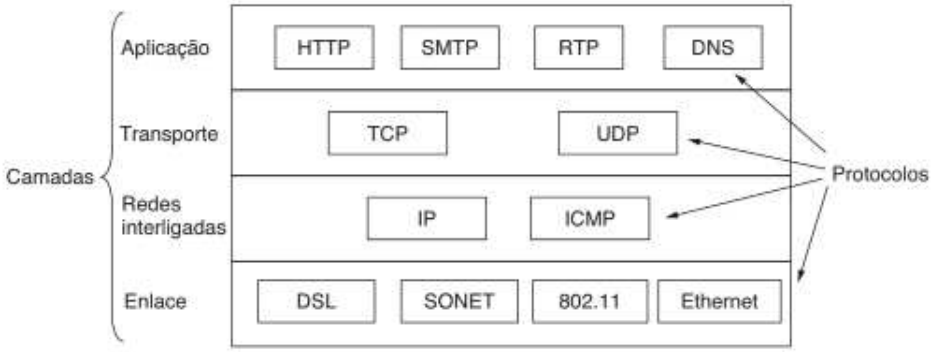
Figura 2 – Camadas do modelo OSI e do modelo TCP/IP



Fonte: Tanenbaum (2011)

as camadas mais baixas possuem protocolos de comunicação de rede, como o Ethernet e o 802.11.

Figura 3 – Protocolos do modelo TCP/IP

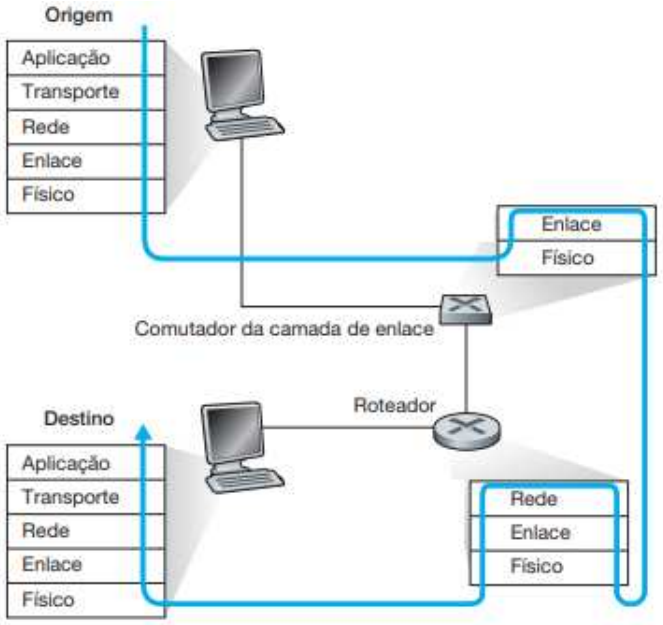


Fonte: Tanenbaum (2011)

As camadas superiores ficam mais próximas do usuário final e possuem maior abstração. Para executar o serviço de uma camada alta, as camadas inferiores oferecerão suporte. Por exemplo, para utilizar um protocolo da camada de aplicação, como o HTTP, para se comunicar com outra máquina, o dispositivo usará as interfaces entre camadas até o nível mais baixo (camada de enlace ou camada física) e apenas então a comunicação com o outro dispositivo ocorrerá. A Figura 4 mostra o caminho que uma mensagem faz na pilha de protocolos da Internet (KUROSE, 2013).

As próximas subseções tratarão de cada camada do modelo TCP/IP.

Figura 4 – Caminho de uma mensagem na pilha de protocolos da Internet



Fonte: Adaptado de Kurose (2013)

2.2.1 Camada de Enlace

A camada mais baixa descreve o que as interfaces de conexão, como a Ethernet ou as linhas seriais precisam fazer para cumprir os requisitos da comunicação entre os dispositivos conectados. Esta camada é responsável por controlar o fluxo de mensagens enviadas/recebidas, tendo papéis como mandar os dados presentes na camada de rede de um dispositivo para a camada de rede de outro. A camada de enlace prepara os quadros a serem transmitidos, adicionando cabeçalhos adequados aos quadros a serem enviados e remover os cabeçalhos de mensagens recebidas antes de repassar os pacotes para a camada de rede.

Esta camada implementa os protocolos de acesso ao meio, que serão relevantes para este trabalho e explorados em mais detalhes em seções futuras.

2.2.2 Camada de Rede

A camada de rede define o *Internet Protocol* (IP), e sua tarefa é entregar pacotes IP onde são necessários. O papel desta camada é garantir que os dispositivos consigam injetar pacotes na rede, que chegarão aos seus destinos de maneira independente, sendo esse destino a rede do dispositivo remetente ou alguma outra rede qualquer.

2.2.3 Camada de Transporte

A camada de transporte define protocolos para permitir que dispositivos mantenham uma conversa ativa. Define o protocolo TCP e o protocolo *User Datagram Protocol* (UDP). O protocolo TCP é orientado a conexões e controla o fluxo de bytes entre os dispositivos, sendo assim seguro e confiável.

O protocolo UDP é voltado para aplicações que desejam fornecer o próprio controle de fluxo. É um protocolo inseguro, sendo utilizado em cenários em que velocidade é mais importante do que precisão, ou em cenários como solicitações isoladas que aguardam resposta.

2.2.4 Camada de Aplicação

A camada de aplicação codifica os dados das aplicações em algum dos protocolos de alto nível e os repassa para a próxima camada. Os protocolos da camada de aplicação normalmente estão relacionados com aplicações específicas, como o *File Transfer Protocol* (FTP) para transferência de arquivos ou o *Secure Shell* (SSH) para conexão remota segura.

2.3 Padrões de rede

Para conseguir alcançar a interoperabilidade entre dispositivos, é necessário que os dispositivos que desejem se envolver em alguma comunicação enviem dados de uma maneira que os outros dispositivos possam compreender, e deve ser capaz de entender as mensagens que recebe. Para isso, são definidos padrões internacionais que ditam tudo o que é necessário para a comunicação.

Os padrões internacionais são publicados pela *International Standards Organization* (ISO), organização cujos membros são as instituições padronizadoras de centenas de países membros. A ISO publica padrões nas mais diversas áreas, e se tratando da área de redes, coopera com a ITU-T (*Telecommunication Standardization Sector*) para garantia de apenas um padrão internacional vigente. Um dos órgãos essenciais na definição de padrões é o *Institute of Electrical and Electronics Engineers* (IEEE). Diversos padrões de LAN foram definidos pelo comitê 802 do IEEE. O Quadro 1 mostra alguns dos grupos de trabalho do comitê 802. Nem todos os grupos estão incluídos. Alguns dos grupos entraram em hibernação, outros foram dissolvidos, mostrando que nem toda a história dos modelos 802.x foi de sucesso. Ainda assim, os modelos de rede mais importantes, como a Ethernet e o Wi-Fi, foram definidos por este comitê.

2.4 Redes sem fio

O principal padrão de LANs sem fio é o 802.11, também chamado de Wi-Fi.

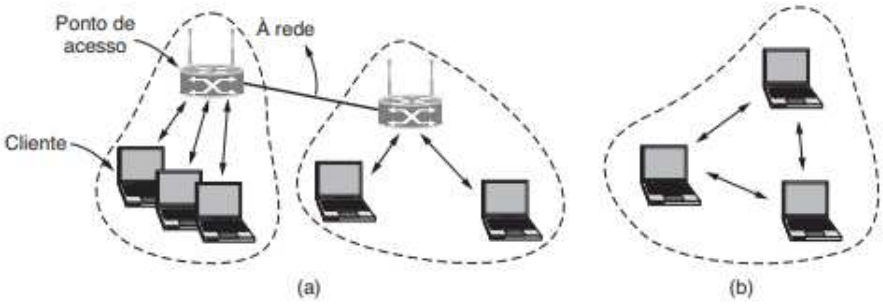
Quadro 1 – Grupos de trabalho do comitê 802 do IEEE

Número	Assunto
802.1	Avaliação e arquitetura de LANs
802.3	Ethernet
802.5	Token ring (anel de tokens; a entrada da IBM no mundo das LANs)
802.11	LANs sem fios (Wi-Fi)
802.15	Redes pessoais (Bluetooth, Zigbee)
802.16	Banda larga sem fio (WiMAX)
802.18	Grupo técnico consultivo sobre questões de regulamentação de rádio

Fonte: Adaptado de Tanenbaum (2011)

Redes sem fio geralmente possuem o propósito de conectar dispositivos à uma rede para lhes garantir acesso à Internet, ou conectar dispositivos a um ponto de acesso conectado à uma rede e que deve trabalhar como intermediário para recebimento e transmissão de pacotes dos dispositivos conectados. Porém, existem também as redes *ad-hoc*, em que os dispositivos se conectam diretamente e podem trocar mensagens entre si. Dado o foco atual no acesso à Internet, este modelo é pouco utilizado (TANENBAUM, 2011). A Figura 5 mostra em (a) a configuração de redes no modo de infraestrutura, tanto para dispositivos conectados diretamente à rede quanto para dispositivos conectados a um ponto de acesso, e em (b) uma rede *ad-hoc*.

Figura 5 – Redes 802.11



Fonte: Tanenbaum (2011)

O padrão 802.11 é dividido em vários subpadrões. Alguns padrões surgiram como evoluções dos padrões anteriores, e outros para preencher nichos específicos. O Quadro 2 lista algumas das versões principais e algumas de suas características.

Os padrões tratados nas subseções a seguir são as redes Wi-Fi convencionais, como as redes domésticas ou corporativas. Estas redes operam entre as frequências 1GHz e 7.125GHz, majoritariamente nas frequências específicas 2.4GHz e 5GHz.

Quadro 2 – Principais padrões 802.11

Padrão	Capacidade máxima de transmissão (Mbit/s)	Frequência (GHz)	Ano
802.11	1 a 2	2.4	1997
802.11b	1 a 11	2.4	1999
802.11a	6 a 54	5	1999
802.11g	6 a 54	2.4	2003
802.11n	72 a 600	2.4/5	2008
802.11ac	433 a 6933	2.4/5	2014
802.11ax	600 a 9608	2.4/5/6	2019

Fonte: Adaptado de Hardwood (2009)

2.4.1 Padrão 802.11

O primeiro padrão de Wi-Fi, originalmente proposto em 1997, especificava taxas de transferência de 1Mbit/s a 2Mbit/s a serem transferidos por sinais infravermelhos. A especificação original deixava certos pontos incertos, então mesmo que várias implementações tenham surgido, não havia garantia de interoperabilidade entre elas. É possível encontrar referências a este modelo como "Wi-Fi 0", mas não é uma denominação oficial. Existem versões deste modelo com modulação *Direct Sequence Spread Spectrum* (DSSS) e *Frequency-hopping spread spectrum* (FHSS). O alcance da rede era de cerca de 20m em ambientes fechados e até 100m em ambientes abertos. Este modelo definiu o CSMA/CA como método de acesso ao meio (BOYES; OGDEN; MAUFER, 2003).

2.4.2 Padrão 802.11b

Uma alteração do modelo 802.11, aumentando a taxa de transferência para até 11Mbit/s e a distância de operação para 35m em ambientes fechados e 140m em ambientes abertos. Este modelo, assim como o 802.11, utiliza modulação DSSS e o CSMA/CA como método de acesso ao meio.

A utilização do CSMA/CA faz com que a taxa de transferência prática da rede seja menos do que a máxima teórica de 11Mbit/s. Devido à relativa alta taxa de transferência para a época, este modelo foi rapidamente aceito como o padrão definitivo para LANs sem fio.

2.4.3 Padrão 802.11a

Este modelo trouxe grandes novidades para as redes sem fio, como a modulação *Orthogonal frequency-division multiplexing* (OFDM) e a operação em frequência 5GHz. A operação em uma frequência diferente de 2.4GHz é consideravelmente benéfica, vista a quantidade de interferências que podem ocorrer na frequência 2.4GHz. Em contrapartida, a distância de ope-

ração da rede é menor e o sinal é mais facilmente absorvido por obstáculos como paredes. A distância alcançada por redes 802.11a pode chegar a 35m em ambientes fechado e 120m em ambientes abertos.

Até então, o modelo 802.11a podia prover a maior taxa de transferência, chegando a até 54Mbit/s, mas na prática, redes neste modelo normalmente ofereciam taxas menores.

2.4.4 Padrão 802.11g

O padrão 802.11g opera na frequência 2.4GHz, assim como o 802.11b, e usa modulação OFDM, assim como o 802.11a. Este modelo foi amplamente adotado e é o que foi anunciado com o nome de "Wi-Fi". Com a mesma largura de banda utilizada pelo modelo 802.11b, este modelo é capaz de alcançar, na teoria, até 54Mbit/s (na prática, o uso do CSMA/CA como método de acesso ao meio diminui a capacidade total da rede). Dispositivos que implementam este modelo são compatíveis com dispositivos que implementam o modelo 802.11b, mas a presença de um dispositivo 802.11b diminui consideravelmente a velocidade da rede.

2.4.5 Padrão 802.11n

O modelo 802.11n introduziu a tecnologia *Multiple-Input and Multiple-Output* (MIMO), usando mais de uma antena a fim de aumentar a taxa de transferência de dados. Este modelo é capaz de alcançar 72Mbit/s com uma única antena e até 600Mbit/s com quatro antenas e largura de banda de 40MHz. Este modelo cobre distâncias de até 70m em ambientes fechados e até 250m em ambientes abertos. Dispositivos que implementam este modelo possuem compatibilidade retroativa com os modelos 802.11b, 802.11a e 802.11g. Este modelo foi retroativamente nomeado Wi-Fi 4 após a definição do Wi-Fi 6.

2.4.6 Padrão 802.11ac

O padrão 802.11ac, também chamado de Wi-Fi 5, expandiu os conceitos introduzidos no padrão 802.11n: uma largura de banda maior (80MHz a 160MHz) e mais fluxos MIMO (até oito). Utilizando a frequência 5 GHz e modulação de alta densidade, as taxas de transferência deste modelo podem chegar a 1.1Gbit/s para transmissões multi-estação e 500Mbit/s para um único cliente.

2.4.7 Padrão 802.11ax

O padrão 802.11ax é o sucessor do modelo 802.11ac. É conhecido como Wi-Fi 6 ou Wi-Fi de Alta Eficiência. Foi idealizado para operar em frequências não-licenciadas entre 1GHz

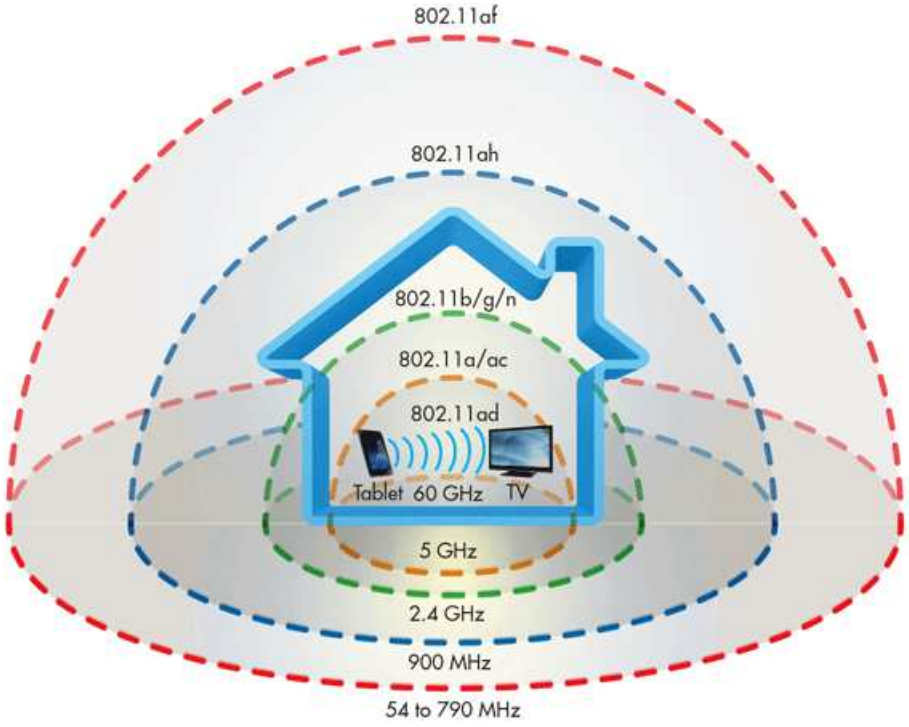
e 7.125GHz, incluindo as frequências normalmente utilizadas por outros modelos (2.4GHz e 5GHz). Graças à modulação *Orthogonal frequency-division multiple access* (OFDMA), a vazão total de uma rede pode ser melhorada em até 300% quando comparada ao modelo 802.11ac.

2.5 O padrão 802.11ah

Diferentemente dos padrões observados anteriormente, o padrão 802.11ah (também chamado de *Wi-Fi HaLow*) opera em frequências abaixo de 1GHz, em contraste aos padrões convencionais que se utilizam das frequências 2.4GHz e 5GHz. Este padrão geralmente opera na frequência não-licenciada 0.9GHz. A operação em frequências abaixo de 1GHz é ideal para equipamentos com baixa necessidade de energia e em redes espalhadas por áreas amplas. Além do longo alcance, obstáculos como paredes não são tão prejudiciais para tal frequência (Wi-fi Alliance, 2018).

Outro exemplo de rede que opera em frequências abaixo de 1GHz é o 802.11af, mas esta opera em frequências licenciadas entre 54MHz e 790MHz. A Figura 6 mostra um esquema simplificado da relação entre a frequência e o alcance de alguns modelos de rede.

Figura 6 – Relação entre frequência e alcance



Fonte: Delisle (2015)

O modelo 802.11ah é voltado para Internet das Coisas: este protocolo tem um nível baixo de consumo de energia e uma área de alcance alta, chegando a 1km de raio. Redes IoT comumente possuem um número alto de dispositivos conectados (como sensores, por exem-

plo) em uma área ampla, o que pode tornar a utilização de uma rede convencional custosa e ineficiente (IEEE. . . , 2017).

Dadas as características das redes alvo do modelo 802.11ah, foi necessário pensar em uma estratégia de controle de acesso ao meio específica, e o protocolo RAW foi criado.

2.6 Protocolos de acesso ao meio

O enlace de redes pode ser de dois tipos: ponto a ponto ou de difusão. O enlace ponto a ponto é a ligação direta entre um remetente a um receptor através da camada de enlace. Existem protocolos tais como o *Point-to-Point Protocol* (PPP) e o *High-Level Data Link Control* (HDCL) idealizados para trabalhar em enlaces ponto a ponto (KUROSE, 2013).

Já para o enlace de difusão, existem múltiplos dispositivos agindo como receptores e remetentes, compartilhando um único canal de transmissão. Quando uma estação envia uma mensagem através do canal compartilhado, todos as estações conectadas receberão uma cópia. O enlace de difusão tem um problema inerente: o acesso múltiplo.

O meio físico pelo qual os dados enviados por dispositivos em uma rede é chamado de meio de transmissão. Em redes cabeadas, o meio de transmissão é o cabo pelo qual os dados trafegam, seja este Ethernet ou fibra óptica. Em redes sem fio, o próprio ar pelo qual as ondas, sinais de satélite ou luz infravermelha trafegam é o meio de transmissão. A Figura 7 ilustra os meios de transmissão de diferentes tipos de redes.

Caso duas ou mais mensagens sejam enviadas pelo meio de transmissão simultaneamente, as estações conectadas receberão vários quadros ao mesmo tempo, fazendo com que as informações sejam embaralhadas e percam o sentido. Tais eventos são chamados de colisões(KUROSE, 2013). Colisões causam a diminuição da vazão geral da rede: a mensagem é perdida e deve ser reenviada, fazendo com que a largura de banda do canal seja desperdiçada.

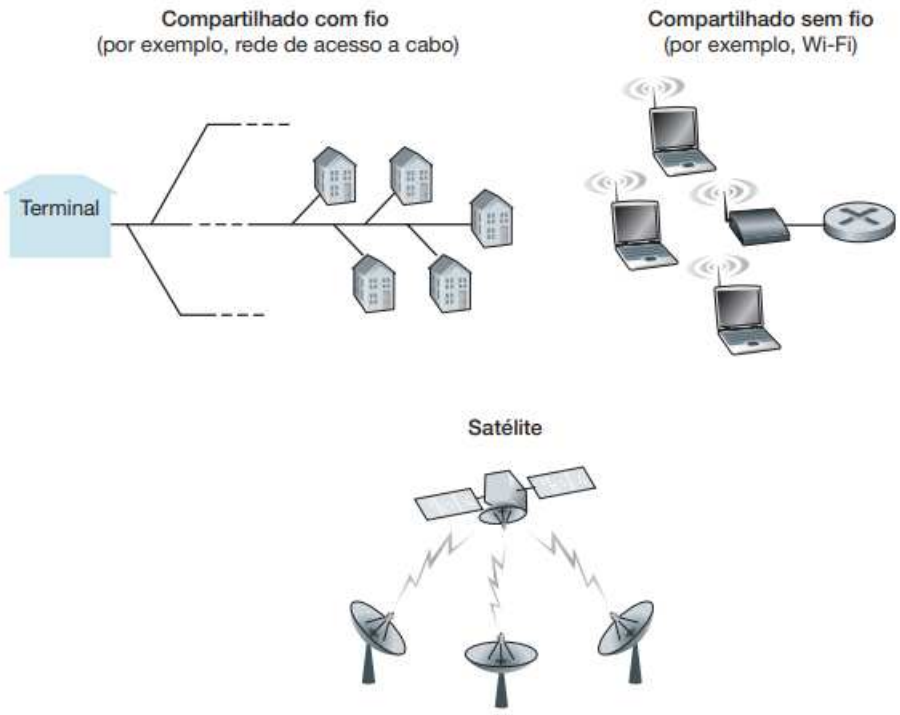
Para detectar e evitar colisões, a camada de enlace provê protocolos de acesso múltiplo. Tais protocolos podem ser classificados em protocolos de divisão de canal, protocolos de revezamento e protocolos de acesso aleatório.

Um protocolo de acesso múltiplo ideal garantiria que, dado um meio de transmissão com capacidade de transmissão de R bits por segundo, uma única estação enviando dados no meio teria uma vazão de R bits/s; e N estações enviando dados no meio de transmissão teriam uma vazão de R/N bits/s.

2.6.1 Protocolos de divisão do canal

Para dividir a largura de banda de um canal entre todos seus nós, pode-se usar a *Time-division multiplexing* (TDM), a *Frequency-division multiplexing* (FDM) ou o *Code-division multiple access* (CDMA).

Figura 7 – Exemplos de meios de transmissão



Fonte: Adaptado de Kurose (2013)

Em uma rede de *n* nós, o protocolo TDM aloca *n* compartimentos de tempo e os atribui, um para cada nó. Um nó só vai transmitir dados quando o compartimento de tempo atribuído a ele estiver ativo. Em geral, um compartimento de tempo permite o envio de um pacote. Os compartimentos de tempo ganham vez de maneira rotativa. Este protocolo possui desvantagens quando apenas um nó precisa usar o meio de transmissão, visto que o nó precisa aguardar sua vez repetidamente e fica limitado a enviar apenas o que é possível durante um compartimento de tempo.

O protocolo FDM divide o canal em frequências diferentes, e as atribui a diferentes nós. As vantagens e desvantagens são semelhantes às do TDM: a chance de colisão é negada, mas transmissões que ocorrem em momentos de pouco movimento são prejudicadas, quando comparadas à transmissões não-divididas.

O protocolo CDMA atribui um código para cada estação, e este código é usado para codificar os bits que a estação envia. Assim, transmissões simultâneas podem ocorrer, e as estações receptoras ainda são capazes de compreender a mensagem (desde que conheça o código do remetente), à despeito da interferência sofrida durante a colisão.

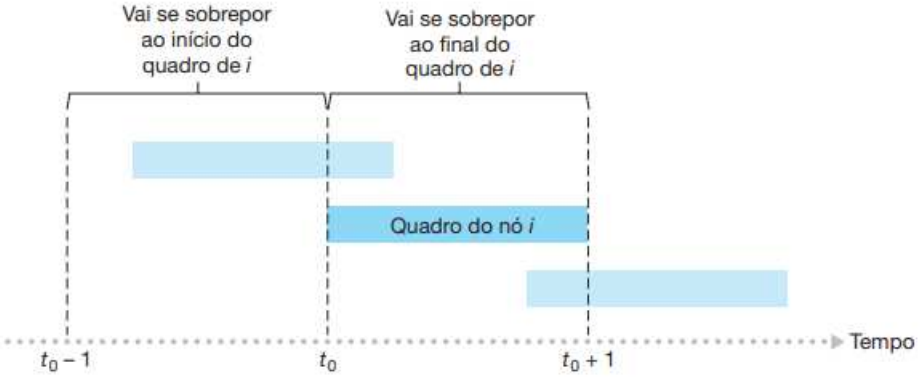
2.6.2 Protocolos de acesso aleatório

Protocolos de acesso aleatório garantem uso completo do meio às estações que estão transmitindo, e faz com que mensagens que se envolvam em colisões sejam reenviadas pelos seus remetentes originais. Existem diversos protocolos de acesso aleatório, e alguns dos mais comuns são:

O *Slotted* ALOHA trabalha de maneira que o tempo é divido em intervalos, cada um com o tempo necessário para transmissão de um quadro. As estações podem iniciar transmissões no início de um intervalo. Caso não ocorra colisão, a transmissão será concluída. Caso uma colisão ocorra, ela será detectada antes do final do intervalo e a transmissão será cancelada, e em cada intervalo subsequente, a estação se baseará em uma probabilidade *p* para decidir se tenta ou não retransmitir, e este processo é repetido até que o quadro seja enviado com sucesso. Em redes com vários nós ativos, a maior parte dos intervalos provavelmente será improdutiva.

O protocolo *Advocates of Linux Open-source Hawaii Association* (ALOHA) funciona de maneira semelhante, mas sem utilizar intervalos. Após uma colisão, a estação espera pelo tempo de um intervalo antes de fazer seu teste de probabilidade, ao invés de esperar pelo início de um próximo intervalo. Este protocolo possui uma eficiência ainda menor do que o *Slotted* ALOHA, visto que o início da transmissão por um nó pode acarretar em uma colisão com o final da transmissão de outro nó, causando uma perda total maior do que a de apenas um intervalo do *Slotted* ALOHA. A Figura 8 mostra um diagrama exemplificando essa situação.

Figura 8 – Transmissões interrompidas do protocolo ALOHA



Fonte: Kurose (2013)

No *Carrier-sense multiple access* (CSMA) , estações tomam precauções antes de iniciar transmissões: caso uma transmissão já esteja ocorrendo no meio compartilhado, a transmissão da estação será adiada em uma quantia de tempo aleatória, chamada de recuo. Colisões ainda podem ocorrer por causa do atraso de propagação no canal - uma estação pode iniciar uma transmissão entre o momento que outra estação qualquer detectou que o meio estava vazio e o momento em que essa mesma estação iniciou uma transmissão. (KUROSE, 2013)

O *Carrier-sense multiple access with collision detection* (CSMA/CD) age da mesma forma, com o adicional de impedir que nós completem transmissões que resultaram em coli-

sões. Ao evitar o envio completo de um quadro que será perdido, o desempenho do protocolo melhora.

O CSMA/CA é a versão do CSMA para redes sem fio. Nessas redes, existe o problema do nó escondido: as estações podem não conseguir enxergar todas as outras estações da rede, portanto a fase inicial de verificação de disponibilidade do meio pode resultar em um falso positivo. A Figura 9 demonstra o problema: o computador B é capaz de ver tanto os computadores A quanto C, mas estes não conhecem um ao outro. Caso o computador A esteja enviando uma mensagem para o computador B, o computador C não consegue detectar a colisão que ocorreria caso iniciasse uma nova transmissão. (COMER, 2009)

Figura 9 – Problema do nó escondido

Fonte: Adaptado de Comer (2009)

A solução do CSMA/CA é fazer com que o nó que está prestes a receber uma transmissão envie uma breve transmissão antes de aceitar a mensagem. Desta forma, todas as estações que estão ao alcance dos nós envolvidos na transmissão estarão cientes da ocupação do meio, mesmo que não conheçam um dos nós envolvidos.

2.6.3 Protocolos de revezamento

Os protocolos de acesso aleatório podem garantir que a capacidade do meio será utilizada em seu todo no momento em que alguma transmissão estiver acontecendo, mas não pode garantir que a capacidade do meio estará sendo usado de maneira eficiente de maneira geral. Por isso, foram criados os protocolos de revezamento, também chamados de protocolos de acesso controlado.

O protocolo de *polling* (seleção) determina um nó que atua como mestre e determina quando cada nó envia quadros e quando para. Desta forma, os intervalos ociosos são eliminados, assim como as chances de colisão. Devido à necessidade de escolha por parte do nó mestre, a transmissão não se utiliza da capacidade total do meio, mas a eficiência da rede ainda é consideravelmente alta. Uma desvantagem é a centralização do processo: caso o nó mestre falhe, a rede toda falha.

Já o protocolo de *token passing* (passagem de permissão) não possui um mestre fixo, mas sim um quadro especial conhecido como token. O nó que possui o quadro de token possui a permissão para transmitir, e ficará com o token até que a transmissão encerre ou um número máximo de quadros seja enviado. Quando alguma dessas condições é alcançada, o nó envia o quadro token para outra estação, de acordo com uma ordem fixa. Assim como o protocolo de se-

leção, este protocolo é altamente centralizado e pode acarretar na falha da rede em decorrência da falha de um único nó.

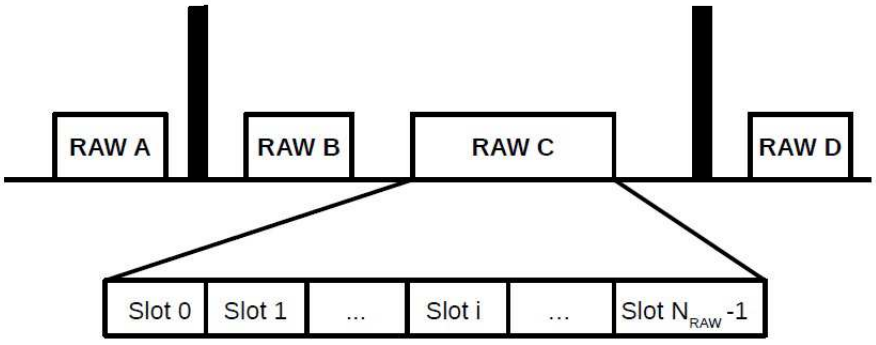
2.6.4 *Restricted Access Window*

Redes no modelo 802.11ah utilizam o protocolo RAW (*Restricted Access Window*) para controle de acesso ao meio. Em redes desse tipo, é comum que um número alto de dispositivos esteja conectado, e as desvantagens do protocolo CSMA/CA se tornam aparentes. A chance de que colisões voltem a acontecer após o tempo de recuo cresce com o número de dispositivos compartilhando o meio de transmissão, fazendo com que redes densamente populadas naturalmente percam vazão ao utilizar o CSMA/CA.

O protocolo adotado para o *Wi-Fi HaLow*, o RAW, toma por base o CSMA/CA e adiciona características de um protocolo de divisão: as estações presentes na rede são separadas em grupos, e apenas um grupo tem acesso ao meio em um determinado momento. Dentro dos grupos, o controle de acesso é feito com CSMA/CA. A definição do modelo 802.11ah não define parâmetros para a configuração dos agrupamentos RAW. (TIAN *et al.*, 2019).

Uma rede 802.11ah pode conter várias configurações de RAW para conjuntos de dispositivos diferentes. O controle de acesso ao meio de uma configuração RAW é feita apenas para os dispositivos que a configuração comporta. Os grupos dentro de uma configuração são chamados de *slots*. A Figura 10 mostra um esquema simplificado de uma configuração RAW. O processo de associação de uma estação à um *slot* segue o modelo *round-robin*.

Figura 10 – Esquema de uma configuração RAW



Fonte: IEEE (2017)

2.7 Simulador ns-3

Softwares de simulação e modelagem de redes são ferramentas importantes tanto para o estudo de redes quanto para facilitar e melhorar o trabalho na área. O programa de simulação de redes ns-2 foi um dos pilares da área por anos, e levou à construção do ns-3, que foca no realismo dos modelos (WEHRLE; GUNES; GROSS, 2010), aproximando os modelos da implementação real daquilo que representam. Os desenvolvedores do ns-3 evitaram usar linguagens com alto nível de abstração a fim de evitar divergências de resultados experimentais reais.

O simulador possui elementos que representam todos os aspectos de redes reais, como nós, canais e protocolos, além de objetos auxiliares para facilitar a criação de simulações. Para utilizar o ns-3, é necessário escrever código em C++ que representa uma topologia de rede e um fluxo de dados.

Com tais características, o ns-3 permite desenvolver simulações realistas e provê formas para analisar os resultados com níveis máximos de detalhes.

3 DESENVOLVIMENTO

O desenvolvimento do trabalho consiste na elaboração de simulações, execução de conjuntos de experimentos e análise dos resultados. Para a construção das simulações, o simulador ns-3 foi utilizado. O simulador ns-3 ainda não conta com um módulo para redes 802.11ah, então foi utilizada uma extensão do simulador, criada por pesquisadores do *Wi-Fi HaLow* (TIAN *et al.*, 2019).

O ambiente utilizado durante todo o processo foi o *Windows Subsystem for Linux* (WSL), utilizando uma distribuição Ubuntu 20.04. A máquina utilizada possui um processador Intel Core i5-8250U e 8GB de RAM.

3.1 Ambiente de simulação

A simulação elaborada visa prover um ambiente em que um número arbitrário de estações seja criado, e todas possam participar ativamente do tráfego na rede de maneira similar. Para isso, o modelo de simulação cria um ponto de acesso e um número X (definido como parâmetro de entrada) de computadores. O ponto de acesso atua também como servidor UDP, e todas as outras máquinas enviam pacotes UDP ao servidor no decorrer da simulação. Cada máquina espera por um pequeno intervalo de tempo definido aleatoriamente antes de iniciar a primeira transmissão, e depois disso envia pacotes UDP em intervalos constantes. Desta forma, garante-se que a rede está sempre ocupada e também garante-se a ocorrência de colisões, que são de grande importância para os experimentos planejados.

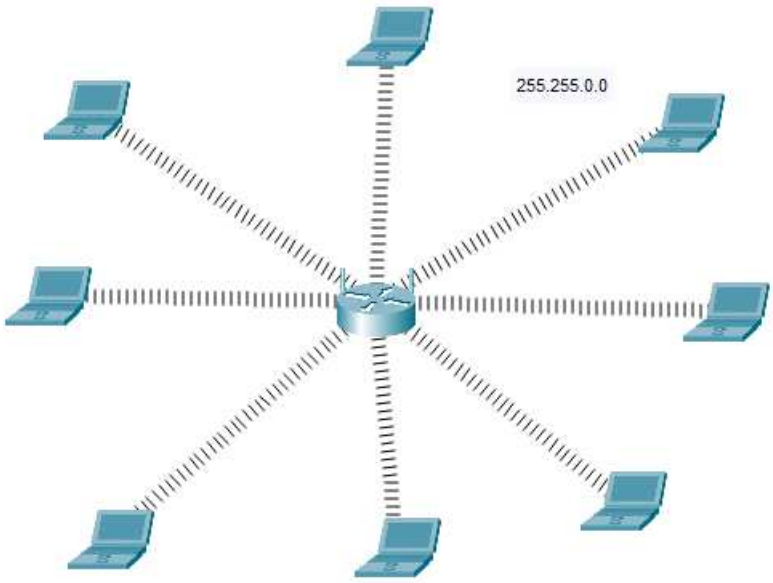
A utilização de um único dispositivo como servidor faz com que o número de colisões seja alto e, no geral, faz com que a perca potencial, mas este não é um problema relevante para o escopo deste trabalho.

A versão utilizada do ns-3 continha uma simulação semelhante já implementada, que pôde ser alterada e simplificada, e então usada para a realização dos experimentos. Um excerto do código da simulação está disponível no Apêncice A. Além do número de computadores, a simulação permite a configuração do tamanho dos pacotes UDP, da taxa de transferência da rede, um arquivo para configuração do protocolo RAW, entre outros. Boa parte dos valores foi fixado, fazendo com que apenas os números de dispositivos e de *slots* RAW variasse.

A rede utilizada usa a máscara 255.255.0.0, para permitir a execução de testes com mais de 250 máquinas sem re-configuração da simulação. Para redes sem fio, o simulador ns-3 requer a configuração de modelos de posição e mobilidade. A configuração realizada definiu um modelo de posição que aloca as estações em um disco ao redor do ponto de acesso de maneira regular. O modelo de mobilidade apenas define que as estações não se movem. A Figura 11 mostra um esquema simples exemplificando a topologia de rede.

É importante entender que o NS-3 não permite a comparação direta entre a rede 802.11n e 802.11ah. A configuração da taxa de transferência em um meio físico no ns-3 é

Figura 11 – Esquema de topologia utilizado



Fonte: Autoria própria (2022)

composta por uma cadeia de caracteres que concatena o algoritmo de modulação, a taxa de transferência e a largura de banda dos canais. As cadeias de caracteres aceitas por cada modelo de rede estão pré-definidas no simulador, e não existia uma configuração considerada válida por ambos os modelos de rede, então não seria possível comparar as redes de maneira direta.

As diferenças nas taxas de transferência e nas larguras de banda possíveis faziam com que experimentos semelhantes tivessem resultados consideravelmente diferentes entre as simulações, o que dificultava a análise do impacto do protocolo RAW na rede 802.11ah. Para evitar comparações com cenários diferentes, descartou-se a segunda simulação, mantendo apenas a simulação que utiliza o Wi-Fi *HaLow*. Como o trabalho tem foco no procolo RAW e no impacto dos grupos, a execução das simulações com apenas um grupo servem como base de comparação de maneira melhor do que redes com configurações diferentes.

3.2 Vazão

Para o cálculo da vazão, a seguinte fórmula foi utilizada:

$$vazao = pacotesrecebidos * payload * 8/(duraodasimulao * 1000000)$$

O cálculo é baseado no número de pacotes UDP recebidos pelo servidor, no tamanho dos pacotes UDP e na duração da simulação em segundos. As multiplicações por números

fixos servem apenas para conversão de unidades, para que o resultado final seja dado em Mbit/s. Tomar a vazão em razão do tempo da simulação permite que simulações curtas sejam executas e tenham resultados semelhantes à simulações mais longas, simplificando o processo.

Este cálculo não é completamente representativo da taxa de transferência máxima da rede porque é dependente do tráfego atual na rede. Intervalos sem transmissões fazem com que a vazão caia, assim como redes com poucos dispositivos e com transferências pequenas. Nas simulações, o único tráfego na rede é o que parte das comunicações UDP, fazendo com que cenários com poucas máquinas tenham a vazão ainda menor do que cenários com um número alto de dispositivos e de colisões.

3.3 Experimentos

Para avaliar o desempenho das redes, foi preparado um conjunto de experimentos. Alguns dos parâmetros de entrada foram fixados e utilizados para todas as execuções:

- Tempo da simulação: 5 segundos;
- Payload (tamanho dos pacotes UDP): 1024 bytes;
- Taxa de transferência da rede: 3Mb/s.

Para um número de dispositivos variando entre 5 e 250, foram executados experimentos variando o número de *slots* RAW entre 1 e 4. Um gráfico representativo dos resultados destes experimentos pode ser observado no Gráfico 1.

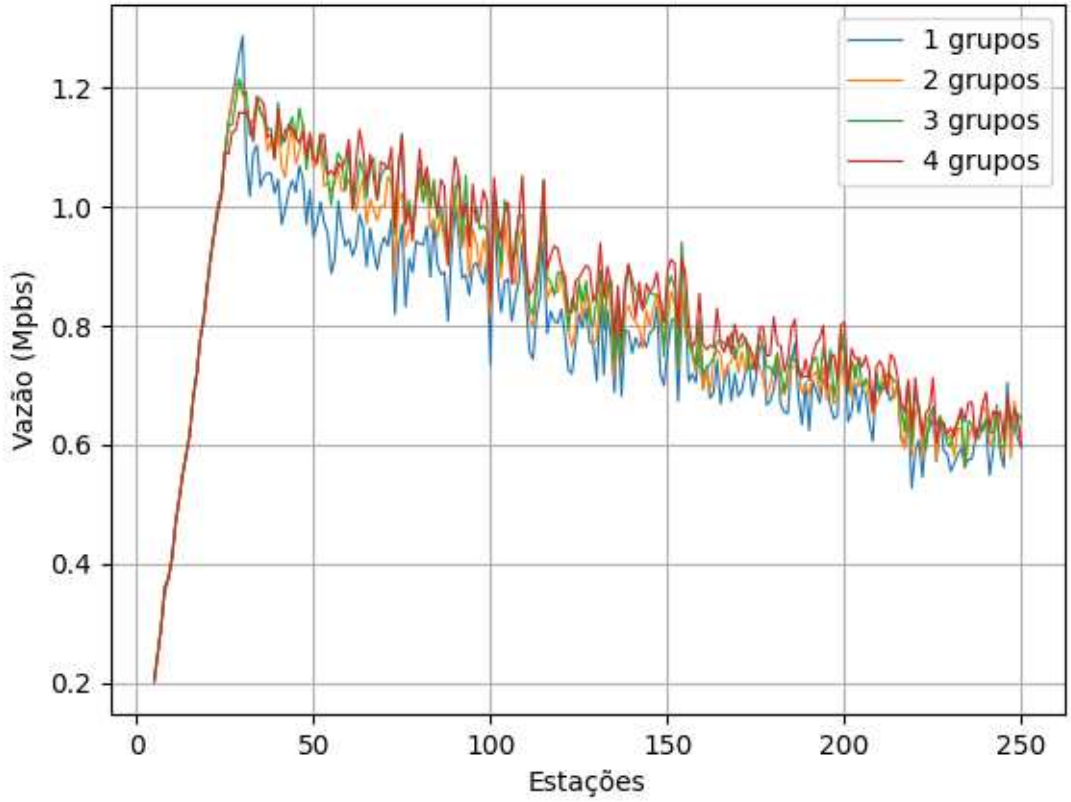
Pode-se observar dois comportamentos com o resultado da vazão:

- O aumento do número de estações provoca uma diminuição da vazão da rede devido às colisões de pacotes. Isto pode afetar diretamente uma rede IoT em que a quantidade de dispositivos na solução for muito grande;
- Com a divisão da rede em grupos menores (2,3 e 4, nesta simulação), observa-se um aumento da vazão conforme aumenta o número de grupos. A maior vazão foi obtida com 4 grupos.

Em outro modelo de experimento, foram considerados os mesmos parâmetros fixos no experimento anterior, mas fixando o número de estações em 100 e executando a simulação para 1, 10, 20, 30, 40 e 50 grupos. Os resultados deste experimentos podem ser observados no Gráfico 2.

Percebe-se que um aumento considerável no número de grupos traz diferenças na vazão da rede, mesmo que pequenas. Em todos os resultados, a vazão se mostra maior do que na configuração com apenas um grupo.

Gráfico 1 – Resultado das simulações



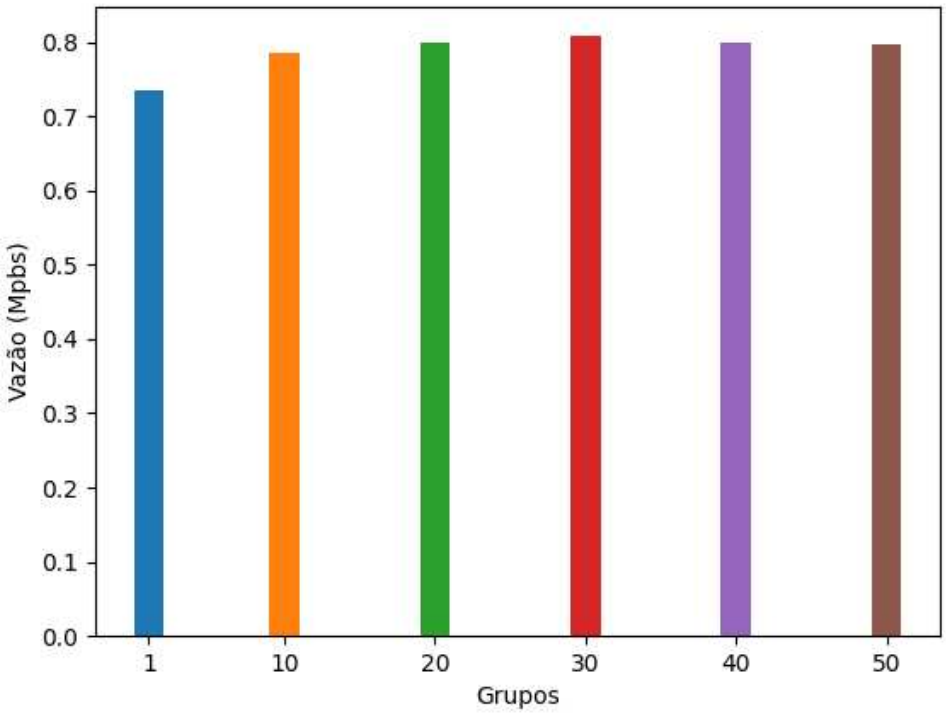
Fonte: Autoria própria (2022)

Não foi simulada a rede 802.11n com o protocolo CSMA/CA, mas pode-se considerar o resultado com 1 grupo para este protocolo. O que mostra um desempenho inferior ao protocolo RAW.

Fica evidenciado que é importante simular uma rede 802.11ah com todos os parâmetros do projeto antes de implementá-la para descobrir o melhor número de grupos para obter uma maior vazão. O simulador ns-3 é largamente usado pela comunidade científica e seus resultados retratam o comportamento de uma rede real.

Os resultados possivelmente seriam mais acentuados em redes muito mais densas e com largura de banda que permitisse taxas de transferência maiores. No momento, apenas canais de 1MHz e 2MHz estão disponíveis para este modelo de rede no simulador ns-3. Tentativas de execução com números maiores de dispositivos eram demoradas e muitas vezes geravam resultados corrompidos, impossibilitando seu uso no trabalho.

Gráfico 2 – Experimento - 100 estações



Fonte: Autoria própria (2022)

4 CONCLUSÃO

No decorrer do trabalho, puderam-se observar detalhes das redes 802.11ah além do protocolo RAW, e o simulador ns-3 propiciou a chance de considerar como as mais diversas características de uma rede de computadores afetam seu desempenho e seu comportamento.

O protocolo RAW mostra resultados positivos mesmo em redes pequenas quando comparadas à escala de redes IoT com milhares de dispositivos. O impacto positivo do protocolo é perceptível, mesmo que dificuldades na execução de simulações maiores ou mais complexas ainda não permitem a análise do protocolo em redes com milhares de dispositivos.

O trabalho abordou a criação e execução de simulações utilizando o simulador ns-3, que permitiu uma análise do cenário em um ambiente fidedigno, mostrando o impacto de várias características que poderiam ser ignorados em simulações mais simples. Observou-se que com o aumento do número de estações em uma rede, a vazão cai, e que este problema é amenizado com a configuração de vários grupos com o protocolo RAW.

REFERÊNCIAS

BOYES, M.; OGDEN, N.; MAUFER, T. A. **A field guide to wireless LANs for administrators and power users**. Philadelphia, PA: Prentice Hall, 2003. (Prentice Hall PTR series in computer networking and distributed systems).

CHIEOCHAN, S.; HOSSAIN, E.; DIAMOND, J. Channel assignment schemes for infrastructure-based 802.11 wlans: A survey. **IEEE Communications Surveys Tutorials**, v. 12, n. 1, p. 124–136, 2010.

COMER, D. **Computer Networks and Internets**. 5. ed. [*S.l.*]: Pearson Education, 2009. ISBN 9780136061274.

DELISLE, J.-J. **What's the Difference Between IEEE 802.11af and 802.11ah?** 2015. Disponível em: https://www.mwrf.com/technologies/active-components/article/21846205/whats-the-difference-between-ieee-80211af-and-80211ah. Acesso em: 14 jun. 2022.

HARWOOD, M. **Exam cram : CompTIA network**. 3. ed. Upper Saddle River, NJ: Pearson, 2009.

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 Amendment 2: Sub 1 GHz License Exempt Operation. Piscataway, NJ, USA, 2017.

KUROSE, J. F. **Redes de computadores e a Internet**: Internet: uma abordagem top-down. 6. ed. [*S.l.*]: Pearson Education, 2013. ISBN 9788543014432.

TANENBAUM, A. S. **Redes de computadores**. 5. ed. [*S.l.*]: Pearson Education, 2011. ISBN 9788576059240.

TIAN, L. *et al.* Optimization-oriented RAW modeling of IEEE 802.11ah heterogeneous networks. **IEEE Internet Things J.**, Institute of Electrical and Electronics Engineers (IEEE), v. 6, n. 6, p. 10597–10609, dez. 2019.

WEHRLE, K.; GUNES, M.; GROSS, J. (Ed.). **Modeling and tools for network simulation**. Berlin, Germany: Springer, 2010.

Wi-fi Alliance. **Wi-fi Org - Wi-Fi CERTIFIED HaLow**. 2018. https://www.wi-fi.org/discover-wi-fi/wi-fi-certified-halow. Acesso em 15/07/2022.

APÊNDICE A – Excerto da simulação programada com o ns-3

Listagem 1 – Excerto do código da simulação

```
#include "s1g-test-tim-raw.h"

Configuration config;

RPSVector configureRAW(RPSVector rpslist, string RAWConfigFile) {
    uint16_t NRPS = 0;
    uint16_t NRAWPERBEACON = 0;
    uint16_t Value = 0;
    uint32_t page = 0;
    uint32_t aid_start = 0;
    uint32_t aid_end = 0;
    uint32_t rawinfo = 0;

    ifstream myfile(RAWConfigFile);

    if (myfile.is_open()) {
        myfile >> NRPS;
        int totalNumSta = 0;
        for (uint16_t kk = 0; kk < NRPS; kk++) {
            RPS *m_rps = new RPS;
            myfile >> NRAWPERBEACON;
            ngroup = NRAWPERBEACON;
            for (uint16_t i = 0; i < NRAWPERBEACON; i++) {
                RPS::RawAssignment *m_raw = new RPS::RawAssignment;

                myfile >> Value;
                m_raw->SetRawControl(Value);
                myfile >> Value;
                m_raw->SetSlotCrossBoundary(Value);
                myfile >> Value;
                m_raw->SetSlotFormat(Value);
                myfile >> Value;
```

```
33                  m_raw->SetSlotDurationCount(Value);
34                  myfile >> Value;
35                  nslot = Value;
36                  m_raw->SetSlotNum(Value);
37                  myfile >> page;
38                  myfile >> aid_start;
39                  myfile >> aid_end;
40                  rawinfo = (aid_end << 13) | (aid_start << 2) | page;
41                  m_raw->SetRawGroup(rawinfo);
42                  totalNumSta += aid_end - aid_start + 1;
43                  m_rps->SetRawAssignment(*m_raw);
44                  delete m_raw;
45              }
46              rpslist.rpsset.push_back(m_rps);
47          }
48          myfile.close();
49          config.NRawSta = totalNumSta;
50      } else
51          cout << "Unable to open RAW configuration file \n";
52      return rpslist;
53  }
54
55  void onSTAAssociated() {
56      if (GetAssocNum() == config.Nsta) {
57          configureUDPServer();
58          configureUDPClients();
59      }
60  }
61
62  void configureUDPServer() {
63      UdpServerHelper myServer(9);
64      serverApp = myServer.Install(wifiApNode);
65      serverApp.Start(Seconds(0));
66  }
```

```
67
68  void configureUDPClients() {
69      Ptr<UniformRandomVariable> m_rv = CreateObject<UniformRandomVariable>();
70      UdpClientHelper myClient(apNodeInterface.GetAddress(0), 9);
71      myClient.SetAttribute("MaxPackets", config.maxNumberOfPackets);
72      myClient.SetAttribute("PacketSize", UintegerValue(config.payloadSize));
73      myClient.SetAttribute("Interval", TimeValue(Seconds(0.2)));
74
75      double randomStart = 0.0;
76      for (uint16_t kk = 0; kk < config.Nsta; kk++) {
77          randomStart = m_rv->GetValue(0, 1.0);
78          ApplicationContainer clientApp = myClient.Install(wifiStaNode.Get(kk));
79          clientApp.Start(Seconds(1 + randomStart));
80      }
81  }
82
83  int main(int argc, char *argv[]) {
84      Time::SetResolution(Time::NS);
85      config = Configuration(argc, argv);
86      config.rps = configureRAW(config.rps, config.RAWConfigFile);
87      config.Nsta = config.NRawSta;
88      wifiStaNode.Create(config.Nsta);
89      wifiApNode.Create(1);
90
91      YansWifiChannelHelper channelBuilder = YansWifiChannelHelper();
92      channelBuilder.AddPropagationLoss("ns3::LogDistancePropagationLossModel",
93                                        "Exponent", DoubleValue(3.76),
94                                        "ReferenceLoss", DoubleValue(8.0),
95                                        "ReferenceDistance", DoubleValue(1.0));
96      channelBuilder.SetPropagationDelay(
97          "ns3::ConstantSpeedPropagationDelayModel");
98      Ptr<YansWifiChannel> channel = channelBuilder.Create();
99
100     YansWifiPhyHelper phy = YansWifiPhyHelper::Default();
```

```
101     phy.SetErrorRateModel("ns3::YansErrorRateModel");
102     phy.SetChannel(channel);
103
104     WifiHelper wifi = WifiHelper::Default();
105     wifi.SetStandard(WIFI_PHY_STANDARD_80211ah);
106     S1gWifiMacHelper mac = S1gWifiMacHelper::Default();
107
108     Ssid ssid = Ssid("ns380211ah");
109     StringValue DataRate;
110     DataRate = StringValue(getWifiMode(config.DataMode));   // changed
111
112     wifi.SetRemoteStationManager("ns3::ConstantRateWifiManager", "DataMode",
113                                  DataRate);
114
115     mac.SetType("ns3::StaWifiMac", "Ssid", SsidValue(ssid), "ActiveProbing",
116                 BooleanValue(false));
117
118     NetDeviceContainer staDevice;
119     staDevice = wifi.Install(phy, mac, wifiStaNode);
120
121     mac.SetType("ns3::ApWifiMac", "Ssid", SsidValue(ssid), "BeaconInterval",
122                 TimeValue(MicroSeconds(config.BeaconInterval)), "NRawStations",
123                 UintegerValue(config.NRawSta), "RPSsetup",
124                 RPSVectorValue(config.rps));
125
126     apDevice = wifi.Install(phy, mac, wifiApNode);
127     MobilityHelper mobility;
128     double xpos = std::stoi(config.rho, nullptr, 0);
129     double ypos = xpos;
130     mobility.SetPositionAllocator("ns3::UniformDiscPositionAllocator", "X",
131                                   StringValue(std::to_string(xpos)), "Y",
132                                   StringValue(std::to_string(ypos)), "rho",
133                                   StringValue(config.rho));
134     mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
```

```
135     mobility.Install(wifiStaNode);
136     MobilityHelper mobilityAp;
137     Ptr<ListPositionAllocator> positionAlloc =
138         CreateObject<ListPositionAllocator>();
139     positionAlloc->Add(Vector(xpos, ypos, 0.0));
140     mobilityAp.SetPositionAllocator(positionAlloc);
141     mobilityAp.SetMobilityModel("ns3::ConstantPositionMobilityModel");
142     mobilityAp.Install(wifiApNode);
143     InternetStackHelper stack;
144     stack.Install(wifiApNode);
145     stack.Install(wifiStaNode);
146     Ipv4AddressHelper address;
147     address.SetBase("192.168.0.0", "255.255.0.0");
148     staNodeInterface = address.Assign(staDevice);
149     apNodeInterface = address.Assign(apDevice);
150
151     Simulator::Run();
152
153     double throughput = 0;
154     uint32_t totalPacketsThrough =
155         DynamicCast<UdpServer>(serverApp.Get(0))->GetReceived();
156     throughput = totalPacketsThrough * config.payloadSize * 8 /
157                  (config.simulationTime * 1000000.0);
158     std::ofstream MyFile("./result/" + config.simType + config.ngroups + ".csv",
159                          std::ios_base::app);
160     MyFile << config.nstations + "," + std::to_string(throughput) + "\n";
161     MyFile.close();
162     return 0;
163 }
```

Fonte: Adaptado de Tian (2019)