

**UNIVERSIDADE FEDERAL DO PAMPA**

**MAIARA SOUSA GOMES**

**ANÁLISE DE REDES AD-HOC EM  
CENÁRIOS AERONÁUTICOS**

**Bagé  
2021**

**MAIARA SOUSA GOMES**

**ANÁLISE DE REDES AD-HOC EM  
CENÁRIOS AERONÁUTICOS**

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Engenharia de Computação como requisito parcial para a obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Gerson Alberto Leiria Nunes

**Bagé  
2021**

Ficha catalográfica elaborada automaticamente com os dados fornecidos pelo(a) autor(a) através do Módulo de Biblioteca do Sistema GURI (Gestão Unificada de Recursos Institucionais).

G633a Gomes, Maiara Sousa

ANÁLISE DE REDES AD-HOC EM CENÁRIOS  
AERONÁUTICOS / Maiara Sousa Gomes.

98 p.

Trabalho de Conclusão de Curso (Graduação)  
- Universidade Federal do Pampa, ENGENHARIA DE  
COMPUTAÇÃO, 2021.

“Orientação: Gerson Alberto Leiria Nunes”.

1. AANET. 2. MANET. 3. Conectividade  
Aeronáutica. I. Título.



SERVIÇO PÚBLICO FEDERAL  
MINISTÉRIO DA EDUCAÇÃO  
Universidade Federal do Pampa

**MAIARA SOUSA GOMES**

**ANÁLISE DE REDES AD-HOC EM  
CENÁRIOS AERONÁUTICOS**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Computação.

Trabalho de Conclusão de Curso defendido e aprovado em: dia, mês e ano.

Banca examinadora:

---

Prof. Dr. Gerson Alberto Leiria Nunes

Orientador

Unipampa

---

Prof.ª Dra. Ana Paula Lüdtke Ferreira

Unipampa

---

Prof. Dr. Leonardo Bidese de Pinho  
Unipampa



Assinado eletronicamente por **ANA PAULA LUDTKE FERREIRA, PROFESSOR DO MAGISTERIO SUPERIOR**, em 21/09/2022, às 17:50, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **LEONARDO BIDESE DE PINHO, PROFESSOR DO MAGISTERIO SUPERIOR**, em 21/09/2022, às 20:33, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **GERSON ALBERTO LEIRIA NUNES, PROFESSOR DO MAGISTERIO SUPERIOR**, em 21/09/2022, às 20:54, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



A autenticidade deste documento pode ser conferida no site [https://sei.unipampa.edu.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **0932242** e o código CRC **E8474DD3**.

Referência: Processo nº 23100.017451/2021-96 SEI nº 0932242



## **AGRADECIMENTO**

Agradeço à minha família (pai, mãe e minha irmã) por todo o apoio e confiança a mim depositados.

Agradeço à minha psicóloga Patrícia por me apoiar em todos os momentos, compreender minhas inquietudes e de forma sempre altruísta me mostrar o melhor caminho a ser seguido. Seu profissionalismo e entrega com os pacientes me admira e me inspira.

Agradeço ao meu orientador professor Dr. Gerson, pela paciência, ajuda e disponibilidade sempre assíduas contando com colaborações extremamente pertinentes ao trabalho. Agradeço também por proporcionar o trabalho colaborativo com a Universidade Federal do Rio Grande do Norte, especificamente com o Grupo de Pesquisa em Prototipagem Rápida de Soluções para Comunicação (GppCom), em seus nomes e que foram extremamente cruciais para o andamento e finalização deste trabalho: a doutoranda Tarciana Guerra e seu respectivo orientador prof. Dr. Vicente Sousa. Agradeço imensamente por compartilhar seus conhecimentos de forma totalmente generosa.

Agradeço à minha amiga Aline, minha corretora de ortografia da língua portuguesa, por sua competência e comprometimento ao me auxiliar.

Por fim, gostaria de agradecer, sem citar nomes – pois de forma contrária estaria cometendo injustiça – os colegas de graduação que ao longo de todo esse tempo mostraram-se verdadeiros amigos e contribuíram de forma significativa na minha evolução dentro da universidade.





## RESUMO

Os serviços aeronáuticos, hoje em dia, demandam cada vez mais tecnologia, principalmente no ponto de vista de conectividade. Atualmente, há infraestruturas de controle de bordo e tráfego aéreo em terra, bem como internet a bordo. Apesar das tecnologias já desenvolvidas na área aeronáutica, faz-se necessário um estudo avançado das conexões realizadas entre os aviões. A finalidade desta pesquisa é ajudar a avaliar o tráfego aéreo na comunicação direta entre aeronaves e evitar acidentes obtendo-se dados de voo de cada aeronave em tempo real. Para isso serão analisados os tipos de mobilidades e condições reais em cenários aeronáuticos através de simulador de voo e obtendo-se dados desse voo, os mesmos serão encaminhados ao simulador de rede para que se possa criar uma rede condizente à essas circunstâncias e assim, observar, qual protocolo apresenta desempenho mais realista. Desta forma, é importante analisar redes *ad-hoc* que atendem estes cenários, bem como, os protocolos de roteamento adequados para tais.

**Palavras-chave:** AANET. MANET. Conectividade Aeronáutica.

## **ABSTRACT**

Aeronautical services, nowadays, demand more and more technology, mainly in terms of connectivity. Currently, there are infrastructures for onboard control and air traffic on land, as well as internet on board. Despite the technologies already developed in the aeronautical area, an advanced study of the connections made between the planes is necessary. The purpose of this research is to help evaluate air traffic in direct communication between aircraft and to avoid accidents by getting sedated from the flight of each aircraft in real time. For this, the types of real estate and real conditions in aeronautical scenarios will be analyzed using a flight simulator and obtaining data from that flight, they will be forwarded to the network simulator so that a network befitting to these circumstances can be created and, thus, observe, more realistic performance. Thus, it is important to analyze redesad-hoc that meet these scenarios, as well as, the appropriate routing protocols for such.

**Keywords:** AANETs, MANET, Aeronautical connectivity.

## LISTA DE FIGURAS

Figura 1	Características de enlaces nos padrões 802.11. ....	22
Figura 2	MIMO para um único cliente no padrão 802.11ac. ....	24
Figura 3	MIMO para vários clientes no padrão 802.11ac. ....	24
Figura 4	Subdivisão dos padrões 802. ....	25
Figura 5	Célula básica. ....	27
Figura 6	Arquitetura IBSS. ....	27
Figura 7	Arquitetura BSS. ....	28
Figura 8	Arquitetura ESS. ....	29
Figura 9	Conexão de várias BSS formando um ESS. ....	30
Figura 10	Topologias Estruturada e <i>ad-hoc</i> . ....	30
Figura 11	Topologia AANET na camada física. ....	31
Figura 12	Protocolos de Roteamento. ....	33
Figura 13	Redes MANETs na área militar. ....	37
Figura 14	Situação de notificação pós-acidente em redes VANET. ....	38
Figura 15	Redes FANET com múltiplos UAVs. ....	38
Figura 16	Imagem captada por UAV em plantação. ....	39
Figura 17	Diferentes cenários para MANET, VANET e FANET. ....	40
Figura 18	Processos de desenvolvimento do trabalho. ....	44
Figura 19	Comportamento dos protocolos entre números de nós e atraso E2E ( <i>End-to-end</i> ) em diferentes altitudes. ....	47
Figura 20	Comportamento dos protocolos entre números de nós e atraso E2E ( <i>End-to-end</i> ) em diferentes velocidades. ....	47
Figura 21	Comportamento dos protocolos entre números de nodos e atraso E2E ( <i>End-to-end</i> ) em diferentes direções do vento. ....	48
Figura 22	Comportamento dos protocolos entre números de nodos e atraso E2E ( <i>End-to-end</i> ) em diferentes performances. ....	48
Figura 23	Requisitos impostos pelas aplicações das redes AANET. ....	51
Figura 24	Análise entre taxa de entrega de pacotes e atraso E2E. ....	52
Figura 25	Análise entre sobrecarga de roteamento e taxa de transferência. ....	53
Figura 26	<i>Site FlightRadar24</i> . ....	54
Figura 27	Visualização das rotas no <i>Google Earth</i> . ....	57
Figura 28	Rota Rio de Janeiro (Aeroporto do Galeão) - São Paulo (Aeroporto de Guarulhos). ....	58
Figura 29	Rota São Paulo (Aeroporto de Congonhas) - Rio de Janeiro (Aeroporto Santos Dumont). ....	59
Figura 30	Distância entre os aeroportos do Galeão e Guarulhos. ....	61
Figura 31	Distância entre os aeroportos de Congonhas e Santos Dumont. ....	61
Figura 32	Padrão de mobilidade dos aviões no tráfego aéreo brasileiro em tempo real. ....	62
Figura 33	Modelo de mobilidade 3D Gauss-Markov. ....	64
Figura 34	Representação do modelo de mobilidade para o cenário indicado. ....	65
Figura 35	Configurações do modelo de <i>Jakes</i> e frequência <i>Doppler</i> . ....	67
Figura 36	Definição dos modelos de propagação <i>Jakes</i> e <i>Three Log Distance</i> no ns-3. ....	68
Figura 37	Geração de tráfego de nós. ....	69
Figura 38	<i>Delay</i> encontrado para faixa de potência de transmissão em dBm. ....	72
Figura 39	<i>Jitter</i> encontrado para faixa de potência de transmissão em dBm. ....	72
Figura 40	Taxa de pacotes perdidos encontrada para faixa de potência de transmissão em dBm. ....	73

Figura 41 Taxa de transferência encontrada para faixa de potência de transmissão em dBm.....	73
Figura 42 <i>Delay</i> encontrado para variação de número de aeronaves.....	74
Figura 43 <i>Jitter</i> encontrado para variação de número de aeronaves.....	75
Figura 44 Taxa de pacotes perdidos encontrada para variação de número de aeronaves.....	75
Figura 45 Taxa de transferência encontrada para variação de número de aeronaves.....	76
Figura 46 <i>Delay</i> encontrado para variação de potência de transmissão.....	77
Figura 47 <i>Jitter</i> encontrado para variação de potência de transmissão.....	78
Figura 48 Taxa de pacotes perdidos encontrada para variação de potência de transmissão.....	78
Figura 49 Taxa de transferência encontrada para variação de número de potência de transmissão.....	79
Figura 50 Atraso encontrado para variação de quantidade de aeronaves.....	80
Figura 51 <i>Jitter</i> encontrado para variação de quantidade de aeronaves.....	80
Figura 52 Taxa de transferência encontrada para variação de quantidade de aeronaves.....	81

## LISTA DE TABELAS

Tabela 1	Evolução dos padrões IEEE 802.11. ....	22
Tabela 2	Diferenças na camada física até o padrão 802.11ac. ....	23
Tabela 3	Diferenças entre MANET, VANET, FANET. ....	40
Tabela 4	Comparação entre as redes MANET, VANET, FANET e AANET. ....	41
Tabela 5	Dados de voos para rota Galeão - Guarulhos. ....	60
Tabela 6	Dados de voo para rota Congonhas - Santos Dumont. ....	60
Tabela 7	Parâmetros de simulação para modelo 3D Gauss-Markov. ....	66
Tabela 8	Parâmetros definidos de simulação. ....	70

## LISTA DE ABREVIATURAS E SIGLAS

3D	Três Dimensões
A2A	Air-to-Air
A2I	Air-to-Infrastructure
A2G	Air-to-Ground
A2S	Air-to-Station
AC	Access Point
A-GR	Aided Geographic Routing Protocol
AANET	Aircraft/Air Ad-Hoc Networks
ATC	Air Traffic Control
AODV	Ad-Hoc On-Demand Distance Vector Protocol
AOLSR	Airbone Optimized Link State Routing Protocol
ABNT	Associação Brasileira de Normas Técnicas
AP	Access Point
BSS	Basic Service Set
CCK	Complementary Code Keying
dBm	Decibel miliwatt
CO2	Dióxido de carbono
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DDR	Dial-on-Demand Routing
DSDV	Destination-Sequenced Distance-Vector
DSR	Dynamic Source Routing
DSSS	Direct Sequence Spread Spectrum
DST	Distributed Spanning Trees Based Routing Protocol
E2E	End-to-end

ESS	Extended Service Set
FANET	Flying Ad-Hoc Networks
FSR	Fisheye State Routing
FSS	Frequency Hopped Spread Spectrum
GRAA	Geographic Routing Protocol for Aircraft Ad-hoc Networks
GS	Ground Station
HSR	Hierarchical State Routing
IBSS	Independent Basic Service Set
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
LTE	Long Term Evolution
MAC	Media Access Control
MANET	Mobile Ad-Hoc Networks
MAV	Micro-Air Vehicle
MIMO	Multiple Input/Multiple Output
MB-OFDM	Multi-Band Orthogonal Frequency Division Multiplexing
METAR	Meteorological Aerodrome Report
MPR	MultiPoint Relay
OACI	Organização Internacional de Aviação
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
OLSR	Optimized Link State Routing Protocol
PAN	Personal Area Network
PLR	Packet Loss Ratio
QoS	Quality of Service
RTP	Real-Time Protocol

SIMD	Single Instruction Multiple Data
SISO	Single Input/Single Output
SLURP	Scalable Location Update Routing Protocol
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UNIPAMPA	Universidade Federal do Pampa
TCP	Transmission Control Protocol
TBRF	Topology Dissemination Based on Reversed-Path
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VANETs	Vehicular Ad-Hoc Networks
ZHLS	Zone-based Hierarchical Link State
ZRP	Zone Routing Protocol
Wi-Fi	Wireless Fidelity
WLANs	Wireless Local Area Networks
WiMAX	Worldwide Interoperability for Microwave Access
WPAN	Wireless Personal Area Networks
WRP	Wireless Routing Protocol
WMAN	Wireless Metropolitan Area Networks
WRAN	Wireless Regional Area Networks
WWAN	Wireless Wide Area Networks



## LISTA DE SÍMBOLOS

$\approx$  Aproximadamente

$\log_{10}$  Logaritmo na base 10

$\Sigma$  Somatório

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	17
1.1 Problema de Pesquisa .....	18
1.2 Motivação e Importância da Pesquisa .....	18
1.3 Objetivos .....	19
1.3.1 Objetivos Gerais.....	19
1.3.2 Objetivos Específicos .....	19
1.4 Organização do Texto .....	19
<b>2 REVISÃO BIBLIOGRÁFICA</b> .....	21
2.1 Redes sem fio .....	21
2.2 Arquitetura das redes sem fio .....	26
2.2.1 Arquitetura IBSS .....	27
2.2.2 Arquitetura BSS.....	28
2.2.3 Arquitetura ESS.....	28
2.3 Redes AANETs.....	31
2.3.1 Protocolos de Roteamento para redes AANET.....	32
2.3.2 Protocolo Proativo.....	34
2.3.3 Protocolo Reativo .....	34
2.3.4 Protocolo Híbrido.....	35
2.3.5 Diferenças entre MANET, VANET, FANET.....	36
<b>3 METODOLOGIA</b> .....	42
3.1 Caracterização de Pesquisa.....	42
3.2 Métodos.....	43
3.3 Procedimentos Adotados .....	44
<b>4 TRABALHOS CORRELATOS</b> .....	46
4.1 Protocolos de roteamento AODV e OLSR em redes AANET .....	46
4.2 Soluções em redes <i>ad-hoc</i> aeronáuticas.....	49
4.3 Modificação do protocolo OLSR em ambientes UDP e TCP.....	51
4.4 Análise dos trabalhos correlatos.....	53
<b>5 DESENVOLVIMENTO</b> .....	54
5.1 Projeto .....	54
5.2 Configuração do simulador ns-3.....	55
5.3 Criação de Cenários.....	56
5.3.1 Rota Rio de Janeiro (Aeroporto do Galeão) - São Paulo (Aeroporto de Guarulhos) .....	57
5.3.2 Rota São Paulo (Aeroporto de Congonhas) - Rio de Janeiro (Aeroporto Santos Dumont).....	58
5.3.3 Configuração de Cenários .....	59
5.4 Análise de Mobilidade .....	62
5.5 Implementação de script no simulador ns-3.....	63
<b>6 RESULTADOS FINAIS</b> .....	71
6.1 Análise de resultados para o protocolo OLSR - padrão 802.11b.....	71
6.2 Análise de resultados para o protocolo OLSR - padrão 802.11ax.....	76
<b>7 CONSIDERAÇÕES FINAIS</b> .....	82
7.1 Conclusões Finais .....	82
7.2 Trabalhos futuros.....	83
<b>REFERÊNCIAS</b> .....	86
<b>APÊNDICE A – CÓDIGO FONTE DA APLICAÇÃO</b> .....	89

## 1 INTRODUÇÃO

Os serviços aeronáuticos, hoje em dia, demandam cada vez mais tecnologia, principalmente no ponto de vista de conectividade. Atualmente, há infraestruturas de controle de bordo e tráfego aéreo em terra, bem como, internet à bordo. Estudos recentes mostram o avanço das redes *ad-hoc* que se enquadram em cenários aeronáuticos. Zhang et al. (2019) cita melhorias através das redes *Aircraft/Air Ad-Hoc Networks* (AANET) como a cobertura da rede estendida, custo de comunicação reduzida e também uma latência de rede menor.

Contextualizando, as redes *ad-hoc* são redes de compartilhamento que não necessitam de um administrador central, ou seja, o usuário pode se conectar não importando o lugar nem o instante. Para o aperfeiçoamento desse conhecimento, têm-se as redes *Aircraft/Air Ad-Hoc Networks* (AANET), redes especificamente do ambiente aéreo, que são distribuídas sem estrutura prévia, onde os dispositivos podem se conectar em áreas sem infraestrutura predefinida podendo ter nós móveis.

Segundo Zhang et al. (2019), as melhorias que a AANET oferece aos aplicativos da aviação civil podem ser muito apreciadas pelos passageiros, operadores, fabricantes de aeronaves e controle de tráfego aéreo. Mais especificamente, a AANET permite que a aeronave faça *upload* ou *download* de dados de navegação e dados de serviço ou entretenimento de passageiros de uma maneira de *streaming* ao vivo sem fio.

Redes AANET podem fornecer soluções escalonáveis e econômicas para aplicativos como segurança no trânsito, planejamento dinâmico de rotas e anúncios sensíveis ao contexto, usando comunicação sem fio a longo prazo. Os dados são transmitidos na forma de pequenos pacotes. No caso de necessidade de se comunicar com um outro nó que esteja fora do alcance, poderá fazê-lo enviando os dados para um nó que esteja dentro de um intervalo de cobertura. Esse nó será retransmitido para o próximo e assim por diante até chegar ao seu destino designado (KUMAR; RANA; KUMAR, 2014, p. 6679, tradução nossa).

Ao encontro a essas necessidades, considera-se significativo aprofundar os estudos em redes móveis, principalmente em circunstâncias do ponto de vista aéreo, para que a conectividade dentro do avião adquira avanços relevantes, a fim de auxiliar os pilotos, passageiros, os sistemas de controle de tráfego aéreo e demais pessoas envolvidas nesses sistemas de comunicação.

## 1.1 Problema de Pesquisa

A criação de cenários aeronáuticos realistas afetam o desempenho de uma comunicação *ad-hoc*?

## 1.2 Motivação e Importância da Pesquisa

Apesar das tecnologias já desenvolvidas na área aeronáutica, faz-se necessário o estudo das conexões realizadas entre os aviões, visto que, a mobilidade aérea torna-se cada vez mais desafiadora diante do crescente número de voos nacionais e internacionais. Segundo a base de dados de aviação européia *Eurocontrol*<sup>1</sup>, a mesma prevê 14,4 milhões de voos em 2035. Portanto, a finalidade desta pesquisa é ajudar a avaliar o tráfego aéreo na comunicação direta entre aeronaves através da análise de redes *ad-hoc*, do estudo dos protocolos de roteamento adequados, da criação de modelos de mobilidade e modelos de propagação para esses cenários em específico.

A relevância de realizar o estudo em comunicações aeronáuticas pode ser justificada pela peculiaridade específica imposta pelas redes AANET. Segundo Kumar e Verma (2019c), os recentes desafios que são impostos nas redes AANET, muitos deles são devido às condições estabelecidas de um ambiente de tráfego aéreo: a velocidade do avião, condições climáticas desconhecidas e com mudanças bruscas de mobilidade sem conhecimento prévio, os protocolos que são tradicionalmente utilizados e não têm capacidade de lidar com as redes AANET altamente dinâmicas, a mobilidade excessiva dos nós e a altitude das aeronaves são alguns dos pontos que tornam as redes AANET singulares e desafiadoras.

Dessa forma, é importante analisar redes *ad-hoc* e configurações específicas que atendam estes cenários, bem como os protocolos de roteamento adequados para tais. Acredita-se que é relevante estudar tais áreas a fim de aprimorar as comunicações aeronáuticas objetivando seu aperfeiçoamento.

---

<sup>1</sup><https://www.eurocontrol.int/>

### 1.3 Objetivos

Nesta seção serão apresentados os objetivos gerais do projeto delimitados pela definição da pesquisa e os objetivos específicos que são implementados gradativamente de acordo com cronograma previamente discutido.

#### 1.3.1 Objetivos Gerais

Esse trabalho tem como objetivo geral analisar e obter dados de redes *ad-hoc* em circunstâncias de tráfego aéreo, assim como seus protocolos de roteamento, obtidos através de simulação em cenários aeronáuticos e, dessa forma, melhor compreender seu comportamento.

#### 1.3.2 Objetivos Específicos

- Estudar protocolos de roteamento para comunicação *ad-hoc*.
- Pesquisar trabalhos correlatos.
- Estudar *softwares* de redes de computadores, assim como modelos de mobilidade e propagação.
- Definir cenários adequados para as simulações da rede AANET.
- Definir o modelo e simulador a ser utilizado.
- Analisar os resultados.
- Validar os resultados.

### 1.4 Organização do Texto

Esse trabalho está disposto como tal: O capítulo 1 apresenta a introdução sobre o problema. O capítulo 2 apresenta a metodologia do projeto seguindo suas concepções norteadas por princípios aos quais melhor se encaixam na pesquisa. No capítulo 3 é exposto o referencial teórico mostrando suas contextualizações junto ao tema proposto. No capítulo 4 são expostos os trabalhos correlatos à esse trabalho. O capítulo 5 descreve a proposta propriamente dita e seu desenvolvimento. No capítulo 6 são explanados os

resultados obtidos e suas respectivas discussões. Por fim, no capítulo 7, são expostas as conclusões finais da pesquisa e trabalhos futuros.

## 2 REVISÃO BIBLIOGRÁFICA

O presente capítulo aborda o levantamento de dados bibliográficos que alicerçaram a construção desta pesquisa.

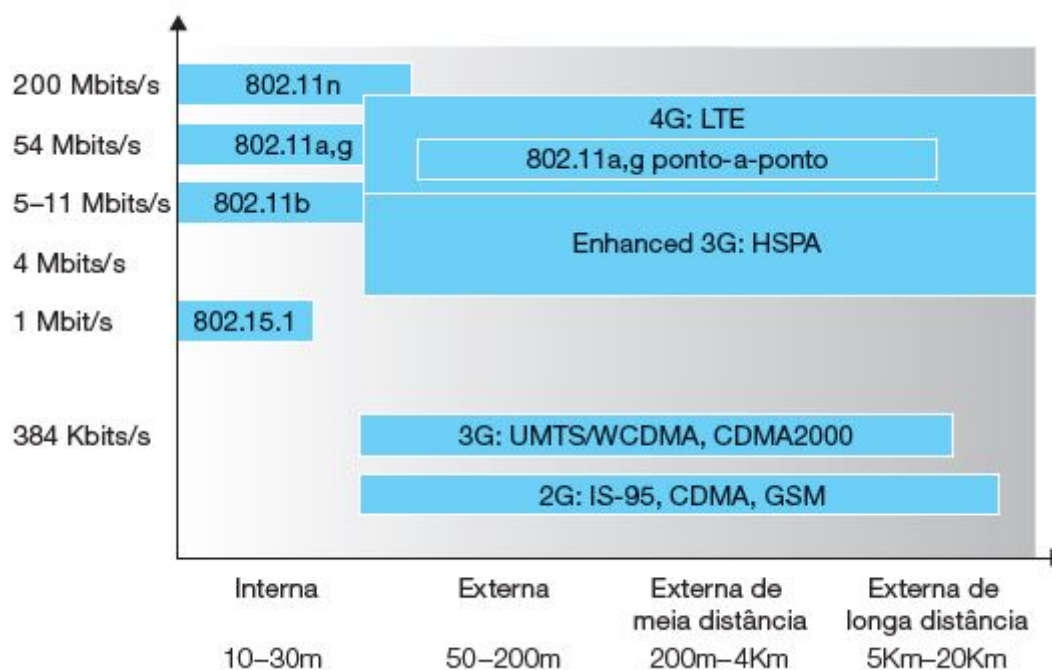
### 2.1 Redes sem fio

Segundo Perkins (2001), as redes móveis sem fio permitem o acesso a serviços, compartilhamento e troca de informações independente da localização e tempo. Este tipo de comunicação proporciona uma mobilidade que não pode ser usufruída em redes cabeadas convencionais, por exemplo. As redes não estruturadas enquadram-se na categoria das redes sem fio que não possuem topologia fixa e, dessa forma, são consideradas ideais para as aplicações PAN (*Personal Area Network*), as redes de abrangência pessoal.

De acordo com Peres e Weber (2003), a mobilidade permite que dispositivos sendo transportados dentro de um local específico possam permanecer conectados, sem perda de acesso aos sistemas e dados da rede. A flexibilidade que este ambiente proporciona diz respeito à facilidade dos dispositivos de acessar a rede sem a necessidade de uma estrutura física de fios.

Com o avanço das redes sem fio, a IEEE convencionou suas classificações em função de suas áreas de abrangências, ou seja, a distância entre os dispositivos em questão. Assim foi criado o padrão IEEE 802.11 para as redes sem fio. Segundo descreve Kurose (2013), há diversos padrões 802.11 para tecnologia de LAN sem fio, entre eles 802.11b, 802.11a e 802.11g. A Tabela 1 apresenta um resumo das principais características desses padrões. Kurose (2013) também expõe de forma clara na Figura 1 as características de enlaces nos diferentes padrões anteriormente mencionados – mais especificamente, até o padrão 802.11n – e discorre principalmente sobre as particularidades de velocidade de transmissão e seus respectivos alcances.

Figura 1 – Características de enlaces nos padrões 802.11.



Fonte: (KUROSE, 2013).

Tabela 1 – Evolução dos padrões IEEE 802.11.

Padrão IEEE	Descrição
802.11 - 1997	Padrão Original (1997) - camada MAC e modulação FHSS e DSSS (1 e 2Mbps)
802.11a	1999 - nova camada física OFDM
802.11b	1999 - nova camada física HR/DSS
802.11g	2003 - nova camada física - ERP
802.11d	2001 - compatibilidade com regulamentos internacionais
802.11e	2005 - introduz qualidade de serviço (QoS)
802.11h	2004 - adapta 802.11a às normas europeias
802.11i	2004 - novos mecanismos de segurança
802.11j	2004 - adapta 802.11 às normas japonesas
802.11 - 2007	Incorpora "a", "b", "c", "d", "e", "h", "i", "j", ao IEEE802.11 - 1997
802.11n	802.11 (2007) acrescido de MIMO (múltiplas antenas), maiores larguras de banda (40 MHz) e agregação de quadros na camada MAC.
802.11ac	Taxas de transmissão que ultrapassam 1Gbps com MIMO e aumento de largura de banda
802.11r	Mecanismos de handoff, particularmente para dispositivos móveis rápidos, como veículos, por exemplo.
802.11s	Redes em malha para dispositivos 802.11.

Fonte: (TORRES, 2013).

”Desde seu surgimento, o padrão IEEE 802.11 passou por uma série de melhorias, incorporadas sob a forma de “emendas”. As emendas são designadas por letras adicionadas ao nome do padrão e são criadas por uma variedade de razões, como incorporar taxas de transmissão mais altas (802.11a, b, g, n e ac), aumentar a segurança da rede (IEEE 802.11i), etc.” (TORRES, 2013, p. 13). A evolução do padrão até a emenda 802.11s pode ser visto de forma mais clara na Tabela 1.



Contudo, os padrões mais atuais e utilizados são respectivamente o 802.11n, 802.11ac e 802.11ax. Perahia e Stacey (2013) explicam que o desenvolvimento dos padrões 802.11 na qual introduz o *Frequency Hopped Spread Spectrum* (FHSS) e *Direct Sequence Spread Spectrum* (DSSS) com frequência de 2,4GHz e logo após, foi aprimorado com o *Complementary Code Keying* (CCK) com uma taxa de dados de 11Mbps. O mesmo padrão desenvolveu o *Orthogonal Frequency Division Multiplexing* (OFDM) com limite de banda de até 5GHz o que dificulta sua adesão. Na Tabela 2 é possível ver essas diferenças detalhadamente.

Tabela 2 – Diferenças na camada física até o padrão 802.11ac.

	802.11	802.11b	802.11a	802.11g	802.11n	802.11ac
Tecnologia na Camada Física	DSSS	DSSS/CCK	OFDM	OFDM DSSS/CCK	SDM/OFDM	SDM/OFDM MU-MIMO
Taxa de Dados (Mbps)	1,2	5,5, 11	6-54	1-54	6,5-600	6,5-6933,3
Faixa de Frequência (GHz)	2,4	2,4	5	2,4	2,4 e 5	5
Espaçamento de Canal (MHz)	25	25	20	25	20 e 40	20,40,80 e 160

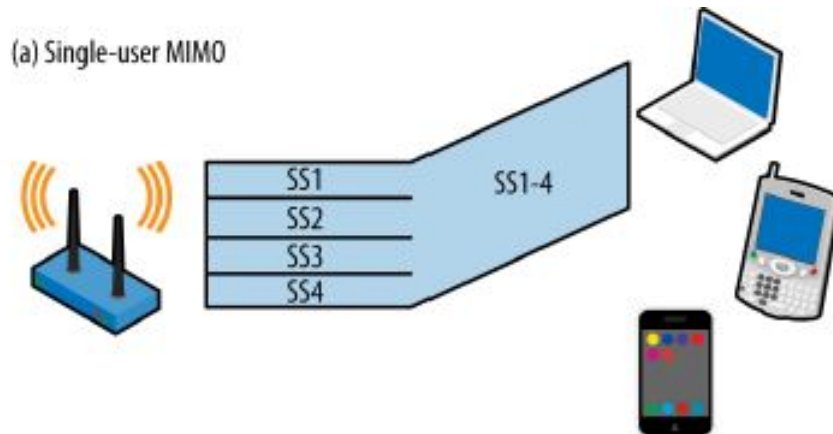
Fonte: (PERAHIA; STACEY, 2013).

Por ordem cronológica, o padrão 802.11n conta com a tecnologia *Multiple Input/Multiple Output* (MIMO) ao qual permite em um único canal o fluxo de dados diferentemente do padrão anterior, que o transmissor e receptor eram *Single Input/Single Output* (SISO).

Em um sistema MIMO, o transmissor e o receptor podem aproveitar vários caminhos. Cada caminho obtém um conjunto diferente de dados e, portanto, a transmissão resultante não está sujeita aos mesmos efeitos destrutivos da interferência de vários caminhos. Na verdade, sem caminhos múltiplos, os benefícios do MIMO são significativamente limitados. O 802.11n surgiu em um ponto interessante na evolução do 802.11. As camadas físicas anteriores foram direcionadas a determinadas bandas de rádio. O salto de frequência 802.11 original e a camada física de sequência direta foram especificados apenas na banda ISM de 2,4 GHz. O 802.11a foi desenvolvido quando a banda de 5 GHz foi aberta para uso sem licença nos Estados Unidos e, subsequentemente, estendida quando os regulamentos foram liberalizados em muitos outros países. (GAST, 2012, p. 3-4, tradução nossa)

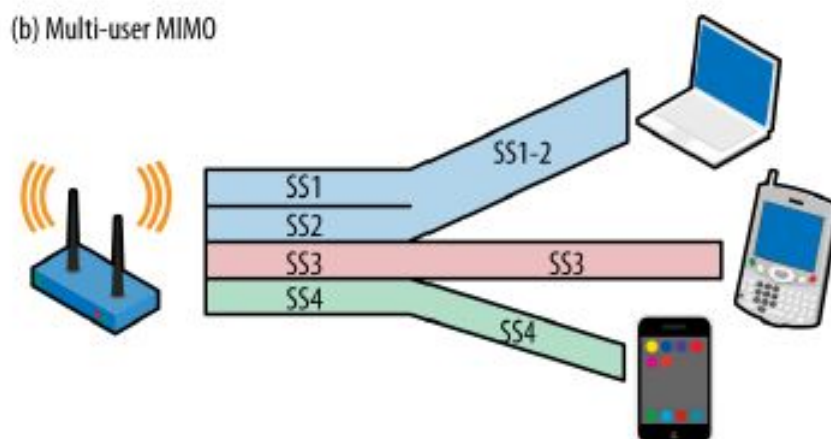
O modelo 802.11ac consta como uma evolução do padrão anterior 802.11n, mais especificamente, no aprimoramento da tecnologia MIMO. A tecnologia MIMO pode ser classificada pelo aumento do número de fluxo de dados para apenas um cliente – a Figura 2 exemplifica a situação – ou vários clientes. O modelo 802.11ac o utiliza de forma multi-usuário enviando pacotes para vários clientes simultaneamente e esta configuração é ilustrada através da Figura 3.

Figura 2 – MIMO para um único cliente no padrão 802.11ac.



Fonte: (GAST, 2013).

Figura 3 – MIMO para vários clientes no padrão 802.11ac.



Fonte: (GAST, 2013).

Em um estudo comparativo, o modelo 802.11ac adiciona ao modelo anterior canais de até 160MHz, transmissão multi-usuário e extensões de aprimoramento às altas taxas de dados.

De acordo com Miller (2018), o padrão 802.11ax é definido o uso de *Multi-user Orthogonal Frequency Division Multiple Access* (MU-OFDMA) e *Multi-user Multiple-input/Multiple-output* (MU-MIMO). A principal tecnologia deste padrão é o *Orthogonal Frequency Division Multiple Access* (OFDMA) que permite o acesso à vários usuários em diferentes larguras de banda simultaneamente.

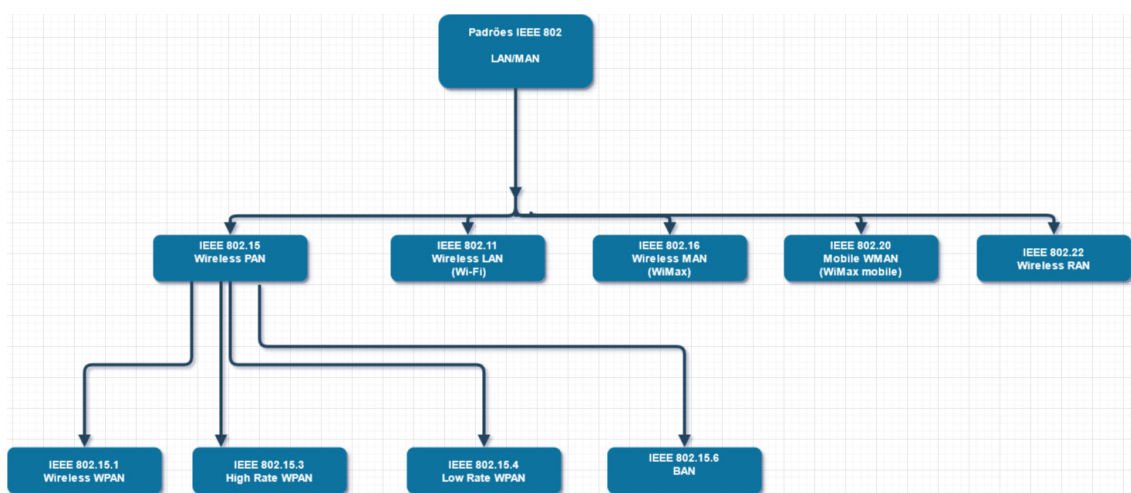
O padrão IEEE 802.11ax está sendo projetado principalmente para fornecer operação de *WLAN* de alta eficiência em ambientes internos e externos, onde a importância fundamental é colocada em aumentar a robustez em ambientes externos, bem como nas transmissões de *up-link*. Este padrão visa

melhorar várias métricas de desempenho (como taxa de transferência média por estação, taxa de transferência de área e assim por diante) que resultam diretamente no aumento da eficiência em várias implantações de (BSS) *Basic Service Set* estreitamente localizado (AFAQUI; GARCIA-VILLEGAS; LOPEZ-AGUILERA, 2016, p. 130-137, tradução nossa).

Similarmente, essas mesmas redes, foram subdivididas e nomeadas como *Wireless Personal Area Networks* (WPANs), *Wireless Local Area Networks* (WLANs), *Wireless Metropolitan Area Network* (WMANs), *Wireless Wide Area Networks* (WWANs) e *Wireless Regional Area Network* (WRANs).

As redes *Wireless Personal Area Networks* (WPANs) enquadram-se nas comunicações de dispositivos próximos, em distâncias curtas. Dentro dessa rede se destaca a conexão Bluetooth. O padrão IEEE desta rede é 802.15 e subdivide-se em vários grupos aos quais estão diversas tecnologias de comunicação. A Figura 4 demonstra esta subdivisão.

Figura 4 – Subdivisão dos padrões 802.



Fonte: Adaptado de (MELO, 2017).

Nesta figura nota-se que o modelo 802.15 enquadra-se em quatro principais redes de comunicação como:

- 802.15.1: Bluetooth, WiBree;
- 802.15.3: UWB, Wireless USB, WiMedia;
- 802.15.4: ZigBee, ZigBee Pro, ZigBee Smart Energy Profile, ZigBee Home Automation Profile, ISA-100/WirelessHART;
- 802.15.6: Body Area Network.

As redes *Wireless Local Area Networks*, as chamadas redes locais sem fio, são comumente usadas em notebooks, smartphones e comunicações sem fio

em geral. Opera no padrão IEEE 802.11 e é a alternativa às redes cabeadas. A principal tecnologia desta família é o Wi-Fi (*Wireless Fidelity*). As redes locais sem fio (WLANs) mudaram as práticas tradicionais de rede, fornecendo uma enorme flexibilidade e liberando os usuários dos cabos de rede. O mercado de WLAN cresceu rapidamente porque a tecnologia sem fio evoluiu para atender às necessidades fundamentais das empresas e dos consumidores de tecnologia. (ILYAS; AHSON, 2005, p. 78, tradução nossa)

Segundo Ilyas e Ahson (2005), as WLANs ajudaram de forma massiva ao aprimoramento do trabalho em pequenas e médias empresas, bem como escritórios e suas atividades remotas (*home office*) trazendo assim maior flexibilidade aos funcionários. Outro passo importante foi o avanço das redes corporativas onde usuários podem se conectar em locais públicos como aeroportos, cafés, transporte público, etc.

*Wireless Metropolitan Area Networks* (WMANs) são redes metropolitanas e como o próprio nome sugere, tem alcance de quilômetros. Tem comunicação bidirecional e é usualmente utilizada em tecnologias de rádio e televisão aos quais demandam maior distância de propagação.

A família *Wireless Wide Area Networks* (WWANs) supera as redes WMANs em termos de alcance e tem como sua principal tecnologia as redes celulares. Atua no padrão IEEE 802.20.

Alpern (2009) define diferentes características à rede WAN, dentre elas a grande abrangência geográfica que pode alcançar distâncias entre países fornecendo grande área de cobertura, bem como a facilidade de conexão de várias LANs em alta velocidade.

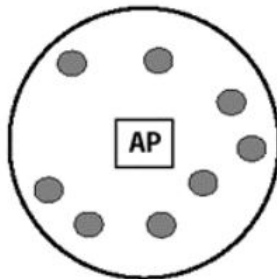
Por fim, as redes *Wireless Regional Area Network* (WRANs) também tem sua tecnologia caracterizada em canais de televisão, entretanto, mais especificamente na transmissão da televisão digital e tem alcance de até 100 quilômetros atuando no padrão IEEE 802.22.

## **2.2 Arquitetura das redes sem fio**

”A área coberta por uma única WLAN é chamada de célula. As estações que compreendem a WLAN em si, estão localizadas dentro da célula. Todas as comunicações dentro e fora da célula devem ser coordenadas por uma única unidade chamada de AP (Ponto de Acesso).” (FONSECA et al., 2001, p. 16). Portanto, as redes locais sem fio tem seu funcionamento baseado no princípio de células na telefonia móvel. Na arquitetura de rede sem fio os usuários móveis têm acesso aos recursos que a mesma oferece deslocando-se para outros pontos e, desde que permanecendo na área coberta – sua

célula. A Figura 5 exemplifica a situação.

Figura 5 – Célula básica.



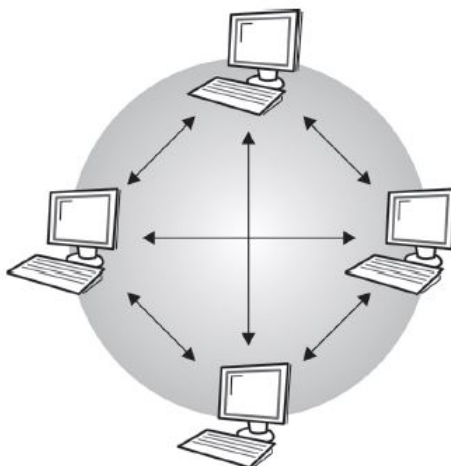
Fonte: (FONSECA et al., 2001).

Dentre as principais arquiteturas, dentro do padrão IEEE 802.11, têm-se *Independent Basic Service Set (IBSS)*, *Basic Service Set (BSS)* e *Extended Service Set (ESS)* as quais têm suas definições melhor elucidadas a seguir.

### 2.2.1 Arquitetura IBSS

Esta arquitetura não possui infraestrutura disposta no local e também não necessita de um ponto de acesso centralizado e a rede é formada conforme a disponibilidade dos dispositivos que encontram-se próximos uns aos outros e podem se conectar entre si. Esta arquitetura adequa-se perfeitamente às redes *ad-hoc* ao qual podemos ilustrar na Figura 6.

Figura 6 – Arquitetura IBSS.



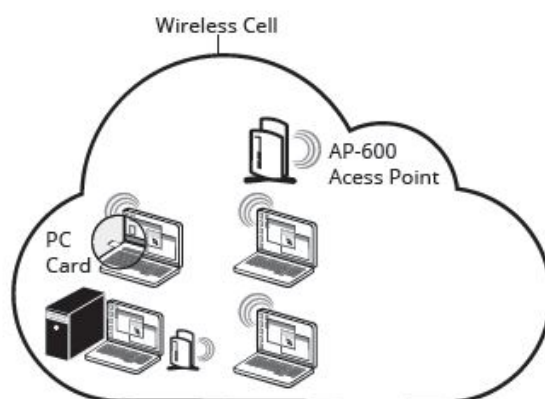
Fonte: (MORAES, 2010).

”Uma das estações no IBSS (ponto básico de serviço independente) pode ser configurada para iniciar a rede e coordenar as funções de rede, ou seja, executar as funções de servidor. Cada ponto básico de serviço independente (IBSS), pelo padrão, pode suportar até 127 dispositivos.” (MORAES, 2010, p. 43).

### 2.2.2 Arquitetura BSS

Uma rede IEEE 802.11 é formada por vários pontos BSS. Esta arquitetura pode ser definida como um conjunto de estações que se comunicam por rádio. Sendo assim, a infraestrutura de uma BSS contém um ponto de acesso comunicando-se com as estações. Esta descrição pode ser melhor explanada na Figura 7.

Figura 7 – Arquitetura BSS.



Fonte: (TORRES, 2013).

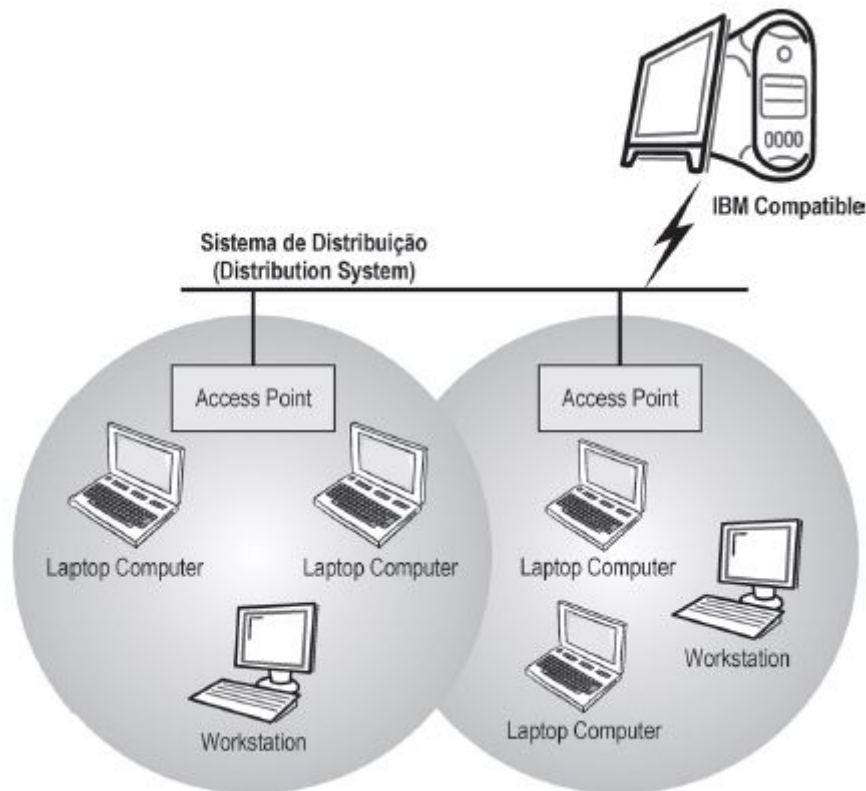
Uma rede infraestruturada foi definida como aquela que contém um ponto de acesso. Toda comunicação passa pelo ponto de acesso e, se duas estações do mesmo BSS querem falar uma com a outra, o quadro será transmitido da estação origem para o ponto de acesso e deste para a estação destino. Isso, apesar de diminuir a capacidade disponível na rede sem fio, torna a sua implementação mais simples, já que estações não precisam se preocupar se outras estão ou não dentro de sua área de cobertura, bastando estarem na área de cobertura do ponto de acesso, permitindo total independência em relação à localização da origem e do destino. (TORRES, 2013, p. 43)

### 2.2.3 Arquitetura ESS

A arquitetura ESS, também chamada de infraestrutura estendida, nada mais é que do que um conjunto de BSS interligados, formando assim um sistema de distribuição. Este

sistema de distribuição pode ser interconectado via cabo ou sem fio (*Wireless*). A Figura 8 demonstra a infraestrutura referida.

Figura 8 – Arquitetura ESS.



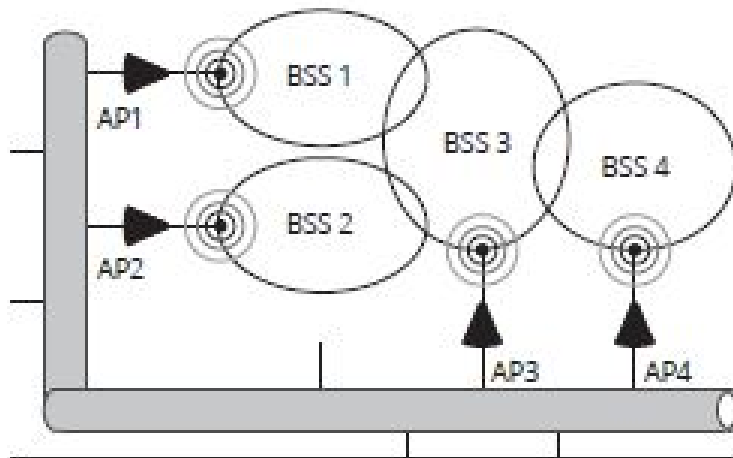
Fonte: (MORAES, 2010).

Nota-se que, nessa infraestrutura, redes sem fio de diversos tamanhos podem ser criadas interligando quantas BSS forem necessárias tendo como principal vantagem o deslocamento sem perda de alcance.

”Uma das vantagens do padrão IEEE 802.11 refere-se à possibilidade de deslocamento entre diferentes APs, sem perder a conexão de rede enquanto estiver se movimentando entre eles. O padrão permite agrupar vários BSS dentro de um Conjunto de Serviço Estendido (ESS).” (TORRES, 2013, p. 44). Esta situação pode ser exemplificada na Figura 9.

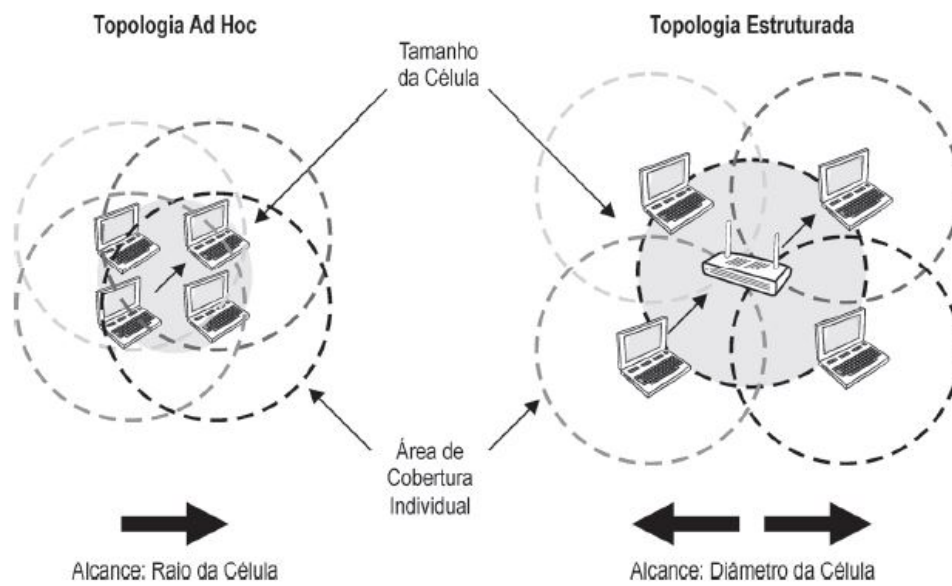
A classificação das topologias se dá basicamente por topologias estruturadas e *ad-hoc*. Como visto anteriormente, enquadram-se a arquitetura BSS e ESS em infraestruturas estruturadas, necessitando assim de um *access point* (AP). Já na *ad-hoc*, a arquitetura IBSS classifica-se como tal, não dependendo de um ponto de acesso para sua conexão e seus dispositivos são conectados entre si. A Figura 10 explica as diferentes arquiteturas.

Figura 9 – Conexão de várias BSS formando um ESS.



Fonte: (TORRES, 2013).

Figura 10 – Topologias Estruturada e *ad-hoc*.



Fonte: (MORAES, 2010).

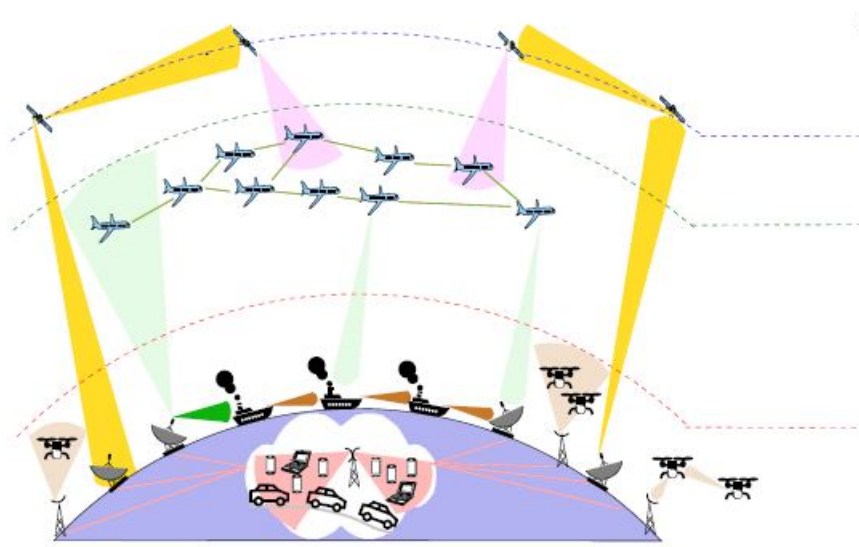


### 2.3 Redes AANETs

As redes *Aircraft Ad Hoc Networks* (AANET) podem ser conceituadas como uma nova família das redes *Mobile Ad Hoc Networks* (MANET). Entretanto, apresentam desafios maiores justamente pelo cenário em que se postam, tal como: topologias mais dinâmicas, tamanho de rede mais variável, velocidades e altitudes muito altas, interferência de outros dispositivos, etc. Todavia, as redes AANET vêm demonstrando estudos cada vez mais avançados com relação às comunicações *air-to-air* (A2A) e *air-to-ground* (A2G) e dessa forma, possibilitam uma proposta de comunicação viável entre aviões – descoberta de posição de outros aviões na mesma rota, controle de velocidade e altitude – e uma melhor compreensão de dados entre o controle de tráfego aéreo e pilotos afim de evitar acidentes e desta maneira oferecer um voo de maior segurança e qualidade aos passageiros.

As redes AANET enquadram-se em uma topologia *ad-hoc* – explicitada no capítulo anterior. Interpretando e contextualizando com os conceitos destas redes já anteriormente vistas, os nós seriam os aviões, e estes, formam a comunicação *ad-hoc* – sem um ponto de acesso no controle e sem uma arquitetura definida previamente. Porém, neste cenário há alguns diferenciais que não são habitualmente vistos em redes *ad-hoc* originais: a alta mobilidade dos nós, roteamento, interferência, velocidade em que se encontram os nós, segurança e estabilidade. A Figura 11 demonstra a topologia correspondente de uma rede AANET juntamente com outros cenários.

Figura 11 – Topologia AANET na camada física.



Fonte: (ZHANG et al., 2019).

Em contrapartida, apesar das redes AANET apresentarem alguns desafios como já observados, o uso dessa rede pode potencialmente mudar as conexões aeronáuticas e assim promover tecnologias de ponta no espaço aéreo. De acordo com Zhang et al. (2019), os benefícios representativos da AANET são resumidos a seguir.

- Cobertura estendida: redes AANET estendem a cobertura de redes A2G da costa para o espaço aéreo oceânico ou remoto, estabelecendo uma rede *ad-hoc* entre aeronaves e GSs. Os GSs também podem se comunicar uns com os outros como parte de uma AANET ou eles podem agir como um *gateway* para conexão com a Internet via uma linha fixa. Mais especificamente, redes AANET são capazes de estender substancialmente a faixa de cobertura no espaço aéreo oceânico e remoto, sem qualquer infraestrutura adicional e sem depender de satélites.
- Custo de comunicação reduzido: evitar enlaces de satélite reduz diretamente o custo de implementação e manutenção, pois o custo de um satélite é geralmente significativamente maior do que um A2G.

Com o desenvolvimento das redes MANET (dentro das quais se encontram as redes AANET), foi observado, que este tipo de tecnologia aplicada no campo aéreo poderia solucionar muitos problemas que se tem na atualidade. Um deles implica nas capacidades atuais das redes de comunicação que estão definidas pela Organização Internacional de Aviação (OACI), que apresentam limitações, como na largura de banda, complexa infraestrutura terrestre e custos altos especialmente quando se faz necessário o acesso à satélites e seus respectivos serviços, conforme Puga (2018) discorre em seu trabalho.

### 2.3.1 Protocolos de Roteamento para redes AANET

”Projetar protocolos de roteamento *multicast* eficientes e confiáveis é um problema muito desafiador, devido às características da MANET, como recursos limitados. Uma estratégia de roteamento inteligente é necessária para usar recursos limitados com eficiência e, ao mesmo tempo, ser adaptável ao dinamismo da rede tais como: tamanho, densidade de tráfego e particionamento da rede.” (SARKAR T.G. BASAVARAJU, 2013, p. 86-87, tradução nossa).

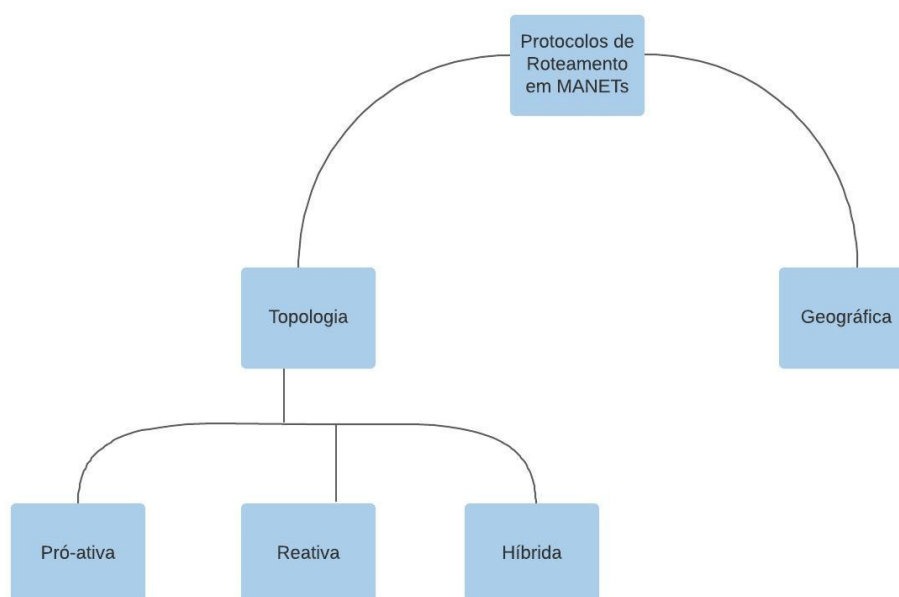
Sarkar T.G. Basavaraju (2013) explica que os protocolos de roteamento nas redes MANET devem fornecer *Quality of Service* (QoS) em diferentes níveis. Nas redes com fio existem algoritmos *link-state* e *distance vector*; o algoritmo *link-state* consiste em manter

a rede sempre atualizada transmitindo as informações para os próximos nós de forma que a cada novo pacote de informações haja a atualização da rede com um algoritmo de caminho mais curto. Entretanto, esses algoritmos de roteamento não são adequados para as redes MANET por apresentarem mudanças significativas nas rotas e as mesmas podem consumir uma alta largura de banda.

Segundo Samara, Karapistoli e Economides (2012), há vários protocolos disponíveis específicos para MANET na literatura. Porém, apenas os que são baseados em estruturas de roteamento e de rede são os mais adaptáveis as essas redes. Na disposição por roteamento existem os protocolos proativos (orientados por tabela) ou reativos (sob demanda) – categorizados como roteamento simples. Na estrutura de rede, pode-se classificar como protocolos de roteamento plano, roteamento hierárquico e roteamento assistido por posição geográfica

Deste modo, a criação de protocolos específicos para essas redes fez-se de suma importância, dada as limitações impostas pelas mesmas e o diferencial para cada cenário aeronáutico que se pode ter nas redes AANET. Na Figura 12, Nagle, Bayross e Gade (2014) apresentam os protocolos dentro das MANET (usados também nas redes AANET) e suas subdivisões.

Figura 12 – Protocolos de Roteamento.



Fonte: Adaptado de (NAGLE; BAYROSS; GADE, 2014).

### 2.3.2 Protocolo Proativo

Rashvand e Chao (2013) explica que os protocolos de roteamento proativos – também conhecidos como protocolos de roteamento baseados em tabelas – atualizam frequentemente a tabela de roteamento e, dessa forma, gera sobrecarga. O autor também explana que nestes protocolos não há atraso de descoberta de rota inicial, entretanto, os mesmos consomem maior largura de banda justificada pelas atualizações constantes proposto para o protocolo em questão.

Sarkar T.G. Basavaraju (2013) exemplifica o protocolo proativo onde cada nó necessita manter informações atualizadas de sua topologia pois, caso haja alguma mudança na topologia ou falha na rede, as mesmas serão repassadas aos próximos nós. O autor também explana que a maioria dos protocolos já estudados tem atributos vindos de algoritmos em redes sem fio, porém, com a necessidade de ajuste às redes *ad-hoc*, as mesmas foram realizadas em protocolos de roteamento de rede com fio. Sarkar T.G. Basavaraju (2013) especifica o protocolo proativo onde os nós atualizam antecipadamente o estado da rede, mantendo a rota definida. Nesse tipo de protocolo pode ocorrer sobrecarga alta devido às recorrentes atualizações de rede repassadas entre os nós.

A seguir têm-se os diferentes protocolos proativos:

- WRP - *Wireless routing protocol*;
- DSDV - *Destination sequence distance vector*;
- FSR - *Fisheye state routing*;
- HSR - *Hierarchical state routing*;
- TBRF - *Topology broadcast reverse forwarding*.

### 2.3.3 Protocolo Reativo

Nesses protocolos, os caminhos de roteamento são pesquisados apenas quando necessário. Uma operação de descoberta de rota invoca um procedimento de determinação de rota. O procedimento de descoberta termina quando uma rota foi encontrada ou nenhuma rota está disponível após o exame de todas as permutações de rota (SARKAR T.G. BASAVARAJU, 2013, p. 88, tradução nossa).

Nos protocolos de roteamento reativo, haverá menos sobrecarga de controle, pois as rotas não serão calculadas a priori. Os protocolos reativos têm melhor escalabilidade do que os protocolos de roteamento proativos, pois o cálculo da rota é feito quando a solicitação é feita (SARKAR T.G. BASAVARAJU, 2013, p. 88, tradução nossa).

Sarkar T.G. Basavaraju (2013) explica que em uma rede *ad-hoc*, as rotas tornam-se instáveis e podem se desconectar a qualquer momento justamente pela mobilidade da rede. Desta forma, é importante a manutenção e atualização nos protocolos de roteamento reativos, apesar de, os mesmos sofrerem muitas variações e atrasos na busca de rotas para encaminhamento de pacotes de dados. Os protocolos de roteamento reativo mais usados são o *Dynamic source routing* (DSR) e *Ad-Hoc on-demand distance vector routing* (AODV).

### 2.3.4 Protocolo Híbrido

”Nos protocolos de roteamento híbrido, os méritos dos protocolos de roteamento proativo e reativo são combinados. Nesses protocolos, as abordagens de roteamento proativas são exploradas em arquiteturas de rede hierárquicas.” (SARKAR T.G. BASAVARAJU, 2013, p. 89, 116, tradução nossa).

De acordo com Sarkar T.G. Basavaraju (2013) os protocolos híbridos podem fornecer soluções escalonáveis visto que, o mesmo pode ser combinado entre o protocolo proativo e reativo, ou seja, um procedimento proativo pode ser utilizado para manter rotas à nós próximos e, um procedimento reativo, pode ser usado para descobrir rotas em nós distantes. Os protocolos de roteamento híbridos podem ser definidos em zonas o que pode ser uma rede dividida em extensões de zonas à cada nó. De outra forma, os nós restantes podem ser rearranjados em árvores ou grupos.

Uma série de diferentes protocolos híbridos de roteamento foram propostos para MANET:

- ZRP - *Zone routing protocol*;
- ZHLS - *Zone-based hierarchical link state*;
- SLURP - *Scalable location updates routing protocol*;
- DST - *Distributed spanning trees-based routing protocol*;
- DDR - *Distributed dynamic routing*.

### 2.3.5 Diferenças entre MANET, VANET, FANET

Embora muitos trabalhos indiquem uma certa semelhança e até mesmo produzam certa ambiguidade entre esses subtemas, é importante destacar suas diferenças e para cada qual suas classificações e áreas subjacentes.

As redes VANET, AANET e FANET encontram-se dentro do subgrupo das redes MANET sendo essa a que engloba todas as anteriores. Dessa forma, com estudos cada vez mais avançados neste ramo, houve esta subdivisão para que se pudesse melhor enquadrar os tipos de veículos e cenários em sua respectiva rede.

As redes MANET se direcionam ao estudo de dispositivos móveis em situações adversas que necessitam uma topologia dinâmica. O campo das redes sem fio emerge da integração da computação pessoal como explica Sarkar T.G. Basavaraju (2013), da tecnologia celular e da Internet. Isso se deve ao aumento das interações entre comunicação e computação, que está mudando o acesso à informação de "a qualquer hora, em qualquer lugar" para "o tempo todo, em qualquer lugar". No momento, existe uma grande variedade de redes, desde a bem conhecida infraestrutura de redes celulares até redes sem fio *ad-hoc* sem infraestrutura. A seguir estão as aplicações de redes sem fio *ad-hoc*:

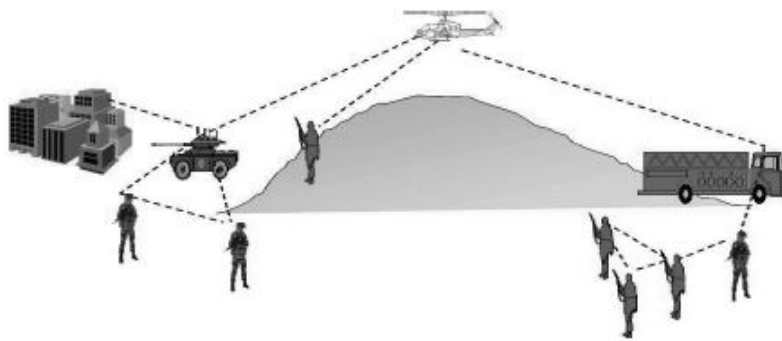
- Rede de comunidade.
- Rede corporativa.
- Rede doméstica.
- Rede de resposta à emergências.
- Rede de veículos.
- Rede de sensores.
- Educação.
- Entretenimento.
- Extensão de cobertura.
- Ambientes comerciais e civis.

Uma das principais aplicações dentro das redes MANET segundo Loo, Mauri e Ortiz (2011), é na área militar, mais especificamente em um campo de batalha. Por exemplo, o aparato militar contém algum tipo de sistema computacional. Nesse caso, a rede *ad-hoc* pode ser muito útil para manter comunicação entre os soldados para operações táticas, veículos e quartéis-generais. As redes *ad-hoc* atendem aos requisitos de mecanismo de comunicação muito rapidamente, porque a rede *ad-hoc* pode ser configurada sem

planejamento e infraestrutura, o que torna mais fácil a comunicação entre as tropas militares por meio de um enlace sem fio. A Figura 13 exemplifica esta possível circunstância.

Ainda dentro do subgrupo das redes MANET, as redes VANET foram desenvolvidas com o objetivo de proporcionar conexão entre dispositivos móveis em veículos ou infraestruturas rodoviárias. Sua gama de aplicabilidade vem de encontro à facilidade e praticidade que a mesma tem o potencial de oferecer principalmente no segmento da segurança, mobilidade, apoio em descoberta de rotas, avisos meteorológicos, etc.

Figura 13 – Redes MANETs na área militar.



Fonte: (LOO; MAURI; ORTIZ, 2011).

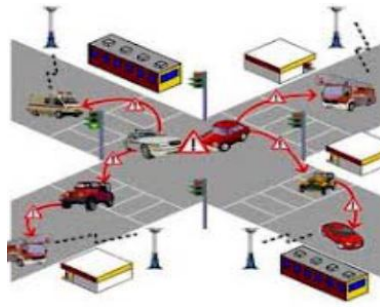
De acordo com Kumar et al. (2013), com base no tipo de comunicação *Vehicle-to-Infrastructure* (V2I) ou *Vehicle-to-vehicle* (V2V), as aplicações das redes VANET podem ser arranjadas nas seguintes classes:

- Orientada para a segurança.
- Orientada para o comércio.
- Orientada para a conveniência.
- Aplicações produtivas.

Tendo como um dos mais importantes, a aplicação para segurança no trânsito torna-se de extrema importância em tráfegos pesados em metrópoles, por exemplo.

Assim descreve Kumar et al. (2013), um que pode ser um dos mais importantes propósitos das redes VANET, a notificação pós-acidente: um veículo envolvido em um acidente iria transmitir mensagens de alerta sobre sua posição para os veículos em reboque para que ele possa tomar decisões com o tempo, bem como para a patrulha rodoviária para o apoio de reboque. A Figura 14 ilustra como acontece esta situação.

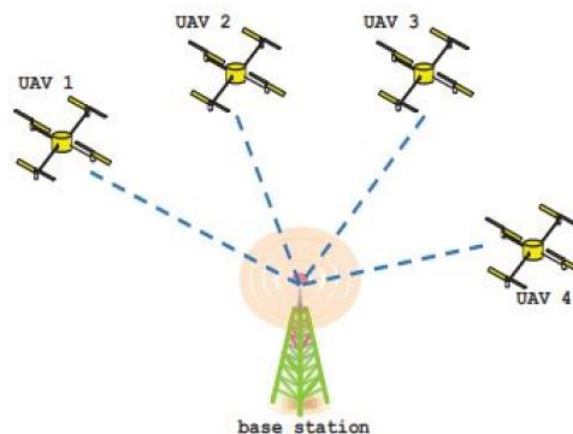
Figura 14 – Situação de notificação pós-acidente em redes VANET.



Fonte: (KUMAR et al., 2013).

De grande destaque na agricultura, os *Unmanned Aerial Vehicle* (UAVs), também chamados de veículos aéreos não tripulados ou drones, a conectividade destes dispositivos estão dentro das redes FANET. Segundo Nayyar (2018), FANET representam uma nova categoria de redes *ad-hoc* que normalmente são compostas por agentes voadores homogêneos ou heterogêneos (UAVs / Drones) que se comunicam, interagem e coordenam adequadamente entre si junto com o ambiente para capturar dados por meio de sensores. As redes FANET são consideradas redes adequadas para enviar informações de forma inteligente e rápida em comparação com as redes *ad-hoc* em situações como regiões de inundação, região afetada por terremoto, campos de batalha, monitoramento de condição de avalanche de montanha e muito mais. As redes FANET podem lidar com tais situações através do uso de um objeto voador chamado *Micro-Air Vehicle* (MAV). O enxame de MAVs pode se auto coordenar em grandes áreas. Um exemplo de cenário das redes FANET é exemplificado na Figura 15.

Figura 15 – Redes FANET com múltiplos UAVs.

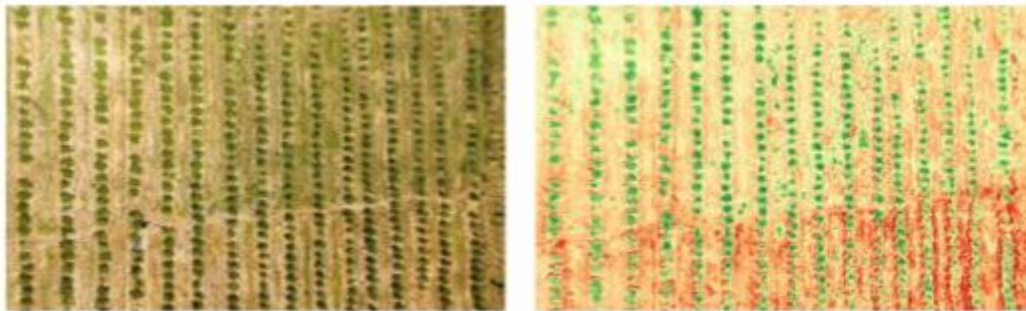


Fonte: (NAYYAR, 2018).



Como destacado anteriormente, uma das principais gamas de aplicações das redes FANET podem ser vistas na utilização de múltiplos UAVs na agropecuária para o controle e detecção de pragas em plantações, avaliação de plantio e irrigação, pulverização, etc. Na avaliação de plantio, por exemplo, com softwares adequados é possível obter imagens e traçar níveis e comparações de estudos da área. A Figura 16 mostra uma imagem captada por UAV em uma plantação de limão.

Figura 16 – Imagem captada por UAV em plantação.



Fonte: (SANTOS; PADOLFI; RAMALDES, 2018).

Embora a sucessão de aplicações nestas redes possam ser muito similares e facilmente podem ser classificadas em todas as redes (MANET, VANET e FANET), é importante ressaltar as diferenças entre as mesmas – a características dos nós, as suas mobilidades, o poder computacional, entre outros. A Tabela 3, bem como a Figura 17, elucidam de forma detalhada suas dessemelhanças e como cada uma tem suas diferentes aplicações designadas à cada cenário específico.

As redes *ad-hoc* sem fio são classificadas de acordo com seus objetivos de utilização, implantação, comunicação e missão. Por definição, FANET é uma forma de MANET, e há muitas considerações de design comuns para MANET e FANET. Além disso, a FANET também pode ser classificada como um subconjunto da VANET, que também é um subgrupo da MANET. Como uma área de pesquisa emergente, a FANET compartilha características comuns com essas redes e também tem vários desafios de design exclusivos. (BEKMEZCI; SAHINGOZ; TEMEL, 2013, p. 1254–1270).

Incluindo nesse cenário as redes AANET, é possível realizar uma distinção clara entre as redes e como as mesmas oferecem diferentes aplicações em diferentes cenários. Como também justifica em seu trabalho Zhang et al. (2019), as diferenças entre AANET, MANET, VANET e FANET encontram-se em questões como canal de propagação, velocidade, altitude, escala de rede, restrição de potência, densidade de nó e segurança são consideradas. Embora a MANET tenha sido inicialmente projetada para telefones celulares e veículos, classificamos os veículos em VANETs, que são desenvolvidas especificamente

Figura 17 – Diferentes cenários para MANET, VANET e FANET.



Fonte: (BEKMEZCI; SAHINGOZ; TEMEL, 2013).

Tabela 3 – Diferenças entre MANET, VANET, FANET.

Características de Nós	MANET	VANET	FANET
Conectividade da Rede	Alta	Média	Baixa
Densidade	Baixa	Alta	Muito baixa
Modelo de Mobilidade	Aleatória	Regular	Caminhos pré-determinados. Regular.
Mudança de Topologia	Pouca	Frequente	Muito frequente
Modelo de Propagação	Bem perto do chão	Bem perto do chão	Acima do chão
Utilização de Energia	Baixa. Protocolos de Energia Eficientes	Não requisitada	Muito Alta
Poder Computacional	Baixa	Alta	Alta
Escalabilidade	Média	Alta	Baixa
QoS	Baixa	Alta	Alta
Localização	GPS	GPS, AGPS, DGPS	GPS, AGPS, DGPS, IMU

Fonte: (NAYYAR, 2018).

para conectar veículos. A AANET se diferencia de MANET, VANET e FANET em termos de suas características, como velocidade de voo, cobertura de rede e altitude, que resultam diretamente em novas características de propagação e impõem desafios tanto na camada de enlace de dados, quanto no design da camada de rede. A Tabela 4 exemplifica estas questões nas diferentes redes.

Tabela 4 – Comparação entre as redes MANET, VANET, FANET e AANET.

Redes	Objetos	Canal	Velocidade (km/h)	Altitude (m)	Escala (km)	Potência	Densidade	Segurança
MANET	Telefones Móveis	Rayleigh	0 $\approx$ 5,4	1 $\approx$ 250	0,25	Com Restrição	Denso	Média
VANET	Veículos	Rayleigh/Rician	14,4 $\approx$ 129,6	0,5 $\approx$ 5	1	Sem Restrição	Cidade: denso; Rural: esparso	Crítica
FANET	UAVs	Rayleigh/Rician	28,8 $\approx$ 460,8	Maior que 122	80	Com Restrição	Missão dependente	Média
AANET	Aeronaves	Rician	882 $\approx$ 925,2	9100 $\approx$ 13000	740	Sem Restrição	Área Povoada: denso; Área Despovoada: esparso	Crítica

Fonte: (ZHANG et al., 2019).

### 3 METODOLOGIA

A pesquisa é entendida tanto como procedimento de fabricação do conhecimento quanto como procedimento de aprendizagem (princípio científico e educativo), sendo parte integrante de todo processo reconstrutivo de conhecimento (DEMO, 2000, p. 20). Assim sendo, este capítulo apresenta a metodologia que conduz o desenvolvimento da pesquisa mostrando suas características e procedimentos adotados.

#### 3.1 Caracterização de Pesquisa

Segundo Freitas (2013), a pesquisa científica com abordagem hipotético-dedutiva, inicia-se com a formulação de um problema e com sua descrição clara e precisa, a fim de facilitar a obtenção de um modelo simplificado e a identificação de outros conhecimentos e instrumentos, relevantes ao problema, que auxiliarão o pesquisador em seu trabalho. Diante disso e das adversidades encontradas para o cenário aeronáutico, o problema da comunicação *ad-hoc* para circunstâncias de tráfego aéreo é formulado assim como suas características e parametrizações específicas. Através do problema inicialmente identificado, constata-se as demais áreas a serem consideradas para o estudo, bem como, os *softwares* adequados.

”A classificação da pesquisa contém subdivisões do ponto de vista da natureza, do ponto de vista de seus objetivos, do ponto de vista dos procedimentos técnicos e do ponto de vista da abordagem do problema.” (FREITAS, 2013, p. 51). Freitas (2013) explica que relacionado a sua natureza, a pesquisa aplicada objetiva gerar conhecimentos para aplicação prática, dirigidos à solução de problemas específicos. Sendo assim, esta pesquisa visa gerar conhecimento na área aeronáutica e como se dão as comunicações entre aeronaves dentro de circunstâncias de tráfego aéreo objetivando análises detalhadas em redes e cenários específicos e peculiares.

Referente aos seus objetivos, a pesquisa exploratória é caracterizada por encontrar-se na fase preliminar, tem como finalidade proporcionar mais informações sobre o assunto que se pretende investigar, possibilitando sua definição e seu delineamento, isto é, facilitar a delimitação do tema da pesquisa. Com isso, este trabalho enquadra-se como pesquisa exploratória por primeiramente ter sido realizado um extenso levantamento bibliográfico das redes AANET – e dessa forma constatou-se certa carência de material nesse tópico em específico – e para que se pudesse ter uma melhor compreensão do tema,

foi necessário o estudo de exemplos semelhantes ao cenários propostos para este estudo e a análise de trabalhos correlatos à este assunto, ou seja, esta pesquisa exploratória baseia-se basicamente em pesquisas bibliográficas e estudos de caso.

Quanto aos procedimentos técnicos, este estudo classifica-se como uma pesquisa experimental que ocorre quando um objeto de estudo determina e seleciona as variáveis que seriam capazes de influenciá-lo – como por exemplo os fatores externos considerados que são velocidade do avião, elevação aérea, condições climáticas, distâncias consideradas, etc. – e são definidas as formas de controle e observação dos efeitos que a variável produz no objeto. De acordo com Freitas (2013) a abordagem do problema na pesquisa é aberta e flexível onde a mesma pode adotar características diversas – a pesquisa pode apresentar tanto uma abordagem qualitativa quanto quantitativa, no entanto, uma delas será predominante. Os dados quantitativos (que são predominantes) obtidos nesse estudo apresentam valores referentes à rede de comunicação aeronáutica criada e foram estatisticamente validados; já os dados qualitativos mostram o comportamento da rede AANET criada e como são apresentados os resultados diante do cenário específico. Por esta pesquisa apresentar associação dos dois tipos de abordagem, a mesma é denominada como uma análise quali-quantitativa mediante à observação de multi-cenários e comportamentos com seus respectivos resultados.

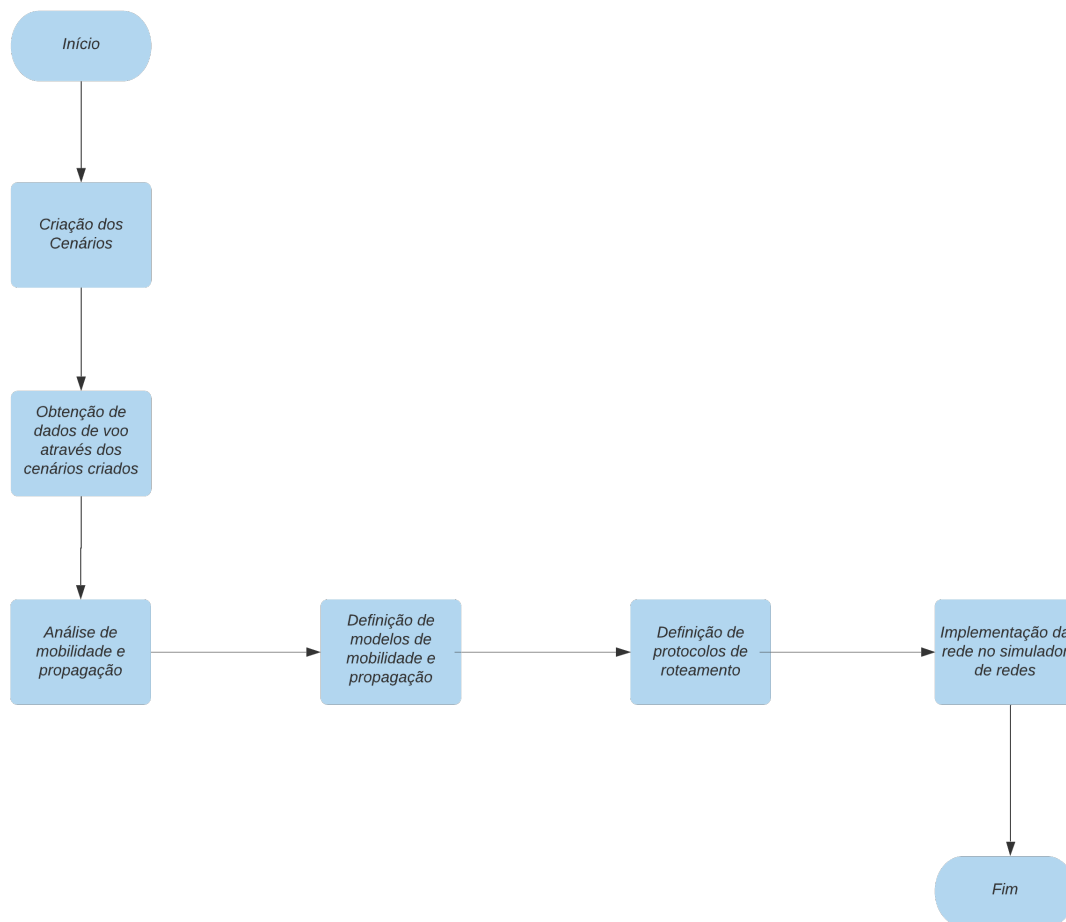
### **3.2 Métodos**

Primeiramente, realiza-se um levantamento bibliográfico extenso sobre as redes *Mobile Ad-hoc Networks* (MANETs) e *ad-hoc*, mais especificamente, as redes AANET em artigos e livros. Dessa forma, constata-se a carência de material nestes tópicos, principalmente na abordagem específica, em cenários aeronáuticos e suas dificuldades por apresentar um contexto um tanto quanto inóspito. Diante dos desafios impostos e das limitações do cenário em questão – mobilidade excessiva e alta velocidade das aeronaves, por exemplo – através de embasamento bibliográfico foi escolhida a ferramenta de simulação adequada para a adaptação da rede, bem como a utilização dos protocolos de roteamento específicos, os modelos de propagação adequados e a criação do tráfego das aeronaves indicadas para esse cenário.

Após obter-se os dados das simulações, são analisados os resultados e verificados os comportamentos das parametrizações (anteriormente explicitadas) aplicadas ao cenário aeronáutico. A Figura 18 mostra as etapas de desenvolvimento do trabalho onde evidencia

o processo descrito anteriormente e também, os passos para que se atinja o objetivo final.

Figura 18 – Processos de desenvolvimento do trabalho.



Fonte: Próprio Autor

### 3.3 Procedimentos Adotados

O procedimento escolhido para a realização dos testes foi a simulação, atendendo aos critérios de oferecer flexibilidade diante do tempo hábil para a implementação da pesquisa e também pela praticidade proporcionada visto que, os cenários e redes escolhidas, demandam um simulador confiável e que simultaneamente atenda as práticas desejáveis para esta proposta. Dessa forma o simulador de redes escolhido foi o ns-3<sup>1</sup> que é um simulador de rede de eventos discretos direcionado ao uso educacional e pesquisa. É um *software open source* e tem sua programação essencialmente baseada em C++ e Python

<sup>1</sup><https://www.nsnam.org/>

e tem como sistema operacional suportado o Linux. Sua usabilidade é frequentemente discutida na literatura apresentando eficácia e credibilidade extensamente comprovadas sendo que o objetivo principal deste simulador é desenvolver um ambiente de simulação aberto ao qual deve estar alinhado com as necessidades de simulação da pesquisa em rede moderna e também incentiva a contribuição da comunidade aberta, a revisão por pares e a validação do *software*.

É correntemente usado em pesquisas para redes *ad-hoc* e suporta os mais populares protocolos de roteamento – estático ou dinâmico, como OLSR e AODV. O núcleo de simulação do ns-3 suporta pesquisas em redes baseadas em IP e não IP. No entanto, as simulações realizadas em sua grande maioria envolvem modelos Wi-Fi (*Wireless Fidelity*), WiMAX (*Worldwide Interoperability for Microwave Access*) e LTE (*Long Term Evolution*) para as camadas 1 e 2.

Os cenários foram criados de acordo com as rotas aéreas escolhidas – Aeroporto do Galeão (Rio de Janeiro) - Aeroporto de Guarulhos (São Paulo) e Aeroporto de Congonhas (São Paulo) - Aeroporto Santos Dumont (Rio de Janeiro) – levando em consideração a elevação aérea média das aeronaves e velocidade média das aeronaves, bem como, os modelos de mobilidade, modelos de propagação e a geração de tráfego de nós foram adaptados para o caso de estudo.

Santos (2015) explica que as ferramentas de simulação ajudam de forma importante no processo de aprendizagem além de propiciarem redução de custos no que se diz respeito à manutenção de laboratórios e também na criação de diferentes formas de aprender. O autor também salienta a relevância da oferta dos *softwares* livres para simulação aos quais tornam o ensino mais democrático e flexível facilitando o acesso ao conhecimento de qualidade onde os alunos podem criar diferentes técnicas de resolução de problemas com a facilidade da virtualização do problema a ser investigado.

”Vale ainda destacar que o ambiente de aprendizagem, baseado em bons simuladores, ajuda os alunos a desenvolver suas habilidades técnicas e resolução de problemas, tendo em vista a possibilidade de o aluno visualizar, de forma detalhada, o comportamento de uma rede de computadores em funcionamento.” (SANTOS, 2015, p. 10).

## 4 TRABALHOS CORRELATOS

Neste capítulo são expostos os trabalhos relacionados ao tema dessa pesquisa e que, por possuírem certo grau de afinidade com os tópicos tratados, proporcionaram tanto embasamento teórico bem como análises comparativas a serem realizadas para que seja possível atingir conclusões coerentes.

### 4.1 Protocolos de roteamento AODV e OLSR em redes AANET

Puga (2018) propõe uma análise teórico-prática sobre os protocolos de roteamento *Ad-Hoc On-Demand Distance Vector Protocol (AODV)* e *Optimized Link State Routing Protocol (OLSR)* em redes *ad-hoc* aeronáuticas no tráfego aéreo de Quito, no Equador. Para a efetividade desta avaliação, houve a análise das características das redes aeronáuticas *ad-hoc*, descrição dos padrões de mobilidade aeronáutica em três dimensões, implementação de simulações dos protocolos de roteamento anteriormente mencionados usando os simuladores de voo <sup>1</sup> e análise dos resultados de desempenho dos mesmos por comparação estatística <sup>2</sup>.

Como parâmetros externos, Puga (2018) analisou altitude do voo, velocidade do avião, direção do vento e performance. Essas variações foram realizadas por cada aeronave, que por sua vez, devido aos benefícios do simulador, permitiram que as fases e procedimentos de todas as missões fossem realizados, gerando o respectivo cenário. Assim, para cada parâmetro foram feitas três variações e, por sua vez, permitiram a geração de traços para a análise do simulador de rede. Observou-se que o desempenho do protocolo OLSR tem um melhor comportamento do que o protocolo AODV apresentando menor quantidade de atrasos. As Figuras 19 e 20 apresentam os resultados obtidos para a altitude e velocidade do avião respectivamente, comparando o comportamento de cada protocolo.

Em todas as comparações, Puga (2018) constatou que o protocolo OLSR apresenta diferenças conforme se aumenta a densidade dos nodos, quando no protocolo AODV o mesmo só apresenta diferença quando há somente seis nodos. As Figuras 21 e 22 a e mostram a comparação de atraso na direção do vento durante o voo e sua performance.

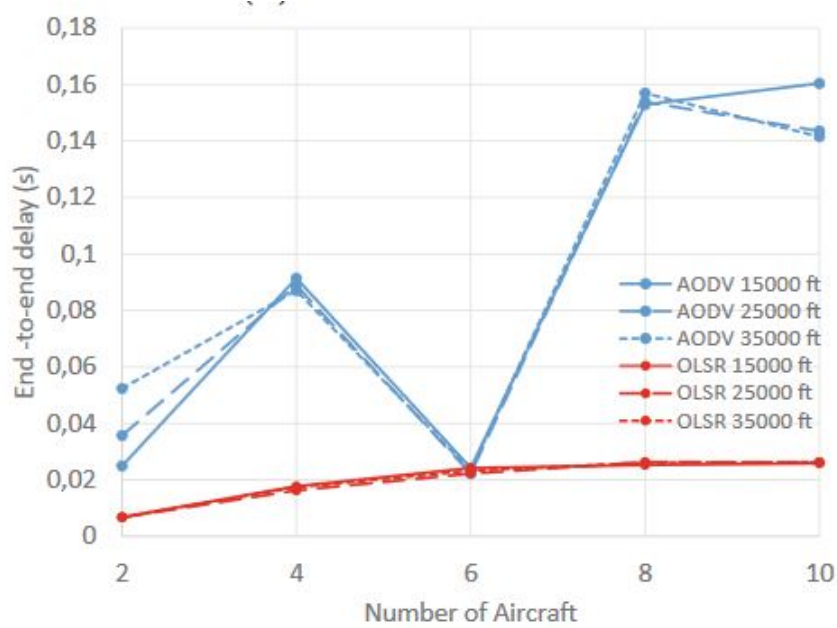
---

<sup>1</sup>STK: *Systems Tool Kit*

<sup>2</sup>ns-3

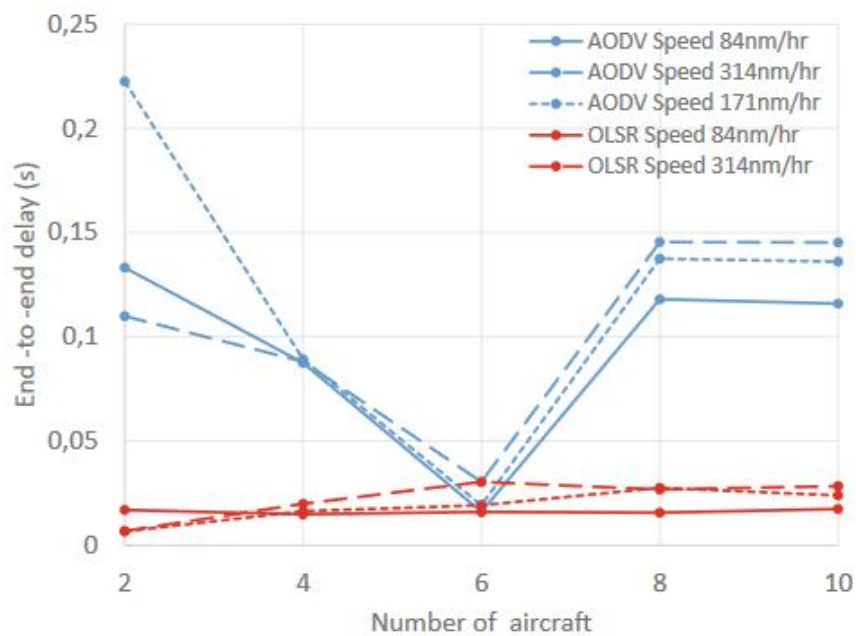


Figura 19 – Comportamento dos protocolos entre números de nós e atraso E2E (*End-to-end*) em diferentes altitudes.



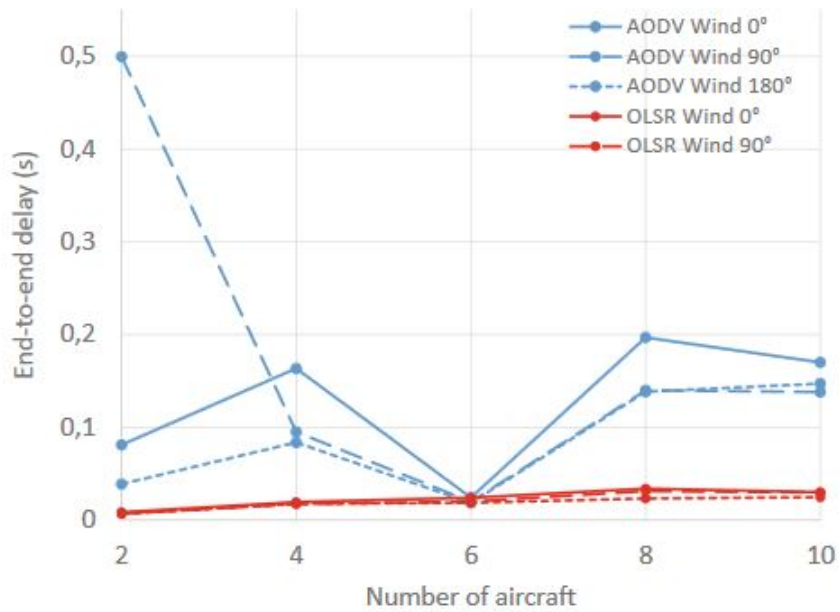
Fonte: (PUGA, 2018).

Figura 20 – Comportamento dos protocolos entre números de nós e atraso E2E (*End-to-end*) em diferentes velocidades.



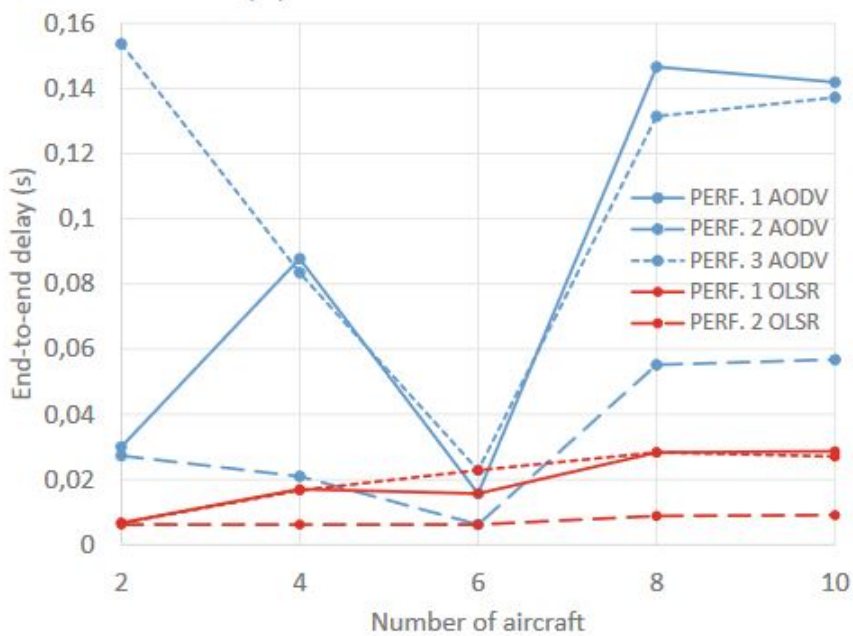
Fonte: (PUGA, 2018).

Figura 21 – Comportamento dos protocolos entre números de nodos e atraso E2E (End-to-end) em diferentes direções do vento



Fonte: (PUGA, 2018).

Figura 22 – Comportamento dos protocolos entre números de nodos e atraso E2E (End-to-end) em diferentes performances.



Fonte: (PUGA, 2018).

## 4.2 Soluções em redes *ad-hoc* aeronáuticas

Zhang et al. (2019) apresenta em seu trabalho, mais especificamente, a comparação das AANET aos membros da família existentes de redes *ad-hoc* sem fio, identificando os recursos específicos das AANET. Em seguida, são investigados diferentes cenários de AANET, incluindo aeroportos, bem como áreas povoadas e despovoadas, que resultam em requisitos rígidos e impõem desafios ao *design* de AANET. Antes de examinados os desafios restantes, foram revisados de forma abrangente o campo das comunicações aeronáuticas em termos de comunicações *Air-to-Ground* (A2G), comunicações *Air-to-Air* (A2A), comunicações *Air-to-Station* (A2S), comunicações na cabine e comunicações *multihop*. Suas capacidades em atender aos requisitos, bem como em acomodar diversas aplicações aeronáuticas fundamentais e aprimoradas, também são investigadas. Em seguida, os desafios associados à concepção das redes AANET são analisados e o recente progresso da pesquisa para abordar esses desafios também é discutido. Para facilitar pesquisas futuras na investigação de redes AANET, foi fornecida uma estrutura geral de design para as mesmas e destacadas algumas questões técnicas importantes em sua concepção, bem como apresentado alguns dos resultados experimentais recentes.

Zhang et al. (2019) diferencia as aplicações das redes AANET e as mesmas são enumeradas como segue:

- Entrega de dados de voo: As redes AANET podem permitir que o pacote de dados, contendo dados de navegação compactadas e dados de entretenimento para os passageiros sejam mantidos em tempo real, durante o voo.
- Controle de tráfego aéreo: Com a estrutura atual do controle de tráfego aéreo contendo antenas muito grandes e sistemas de radares demandando um custo alto de produção. Também os radares têm falhas em cobrir áreas de cobertura global, bem como áreas despovoadas, por apresentarem difícil acesso e desafios financeiros.
- Rastreamento das aeronaves: Sobrevoando em áreas despovoadas, o rastreamento de uma aeronave é feita via *link* por satélite, ao qual ocasiona um certo tempo de latência, ao qual não é desejável para situações de emergência, como por exemplo, depressurização da cabine ou casos gerais de emergência dentro do avião. Porém, as AANET contribuem nesta área explorando a comunicação A2A direta e *multihop* entre aeronaves podendo fornecer serviço de qualidade com baixa latência e alta confiabilidade.

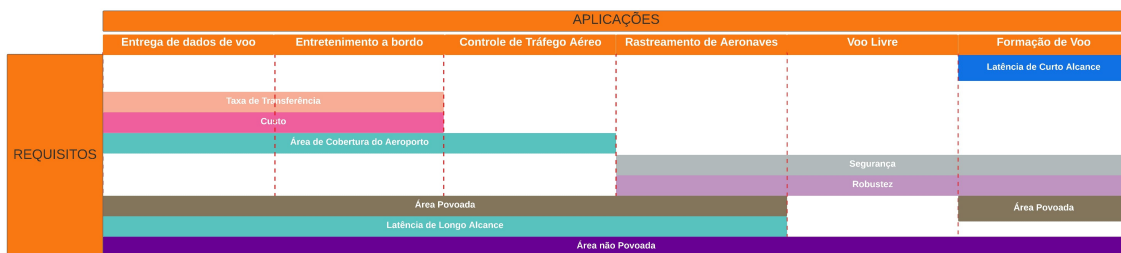
- Voo em formação: Conforme declarado pela ESS<sup>3</sup>, a emissão de dióxido de carbono da aviação deve diminuir em 50% até o ano 2050 em comparação com 2005. Para atingir este objetivo Rayner (1979) explica uma das soluções promissoras que é o voo em formação. Esta abordagem é inspirada na formação de voo de pássaros, onde um aumento de 71% no alcance de voo pode ser observado. Em princípio, a aeronave também poderia economizar combustível aproveitando o voo em formação durante a fase de voo em rota, levando assim a uma maior redução das emissões e custos de  $CO_2$ . Entretanto, esta estrutura de voo pode ocasionar facilmente colisões e acidentes aos quais demandam acompanhamento em tempo real e com baixa latência. Mais uma vez as redes AANET podem ser promissoras nesta área propondo uma estrutura A2A (*air-to-air*) para que cada aeronave tenha acesso às suas posições e controle de cada uma para evitar desastres aéreos e assim também contribuir com o meio ambiente.
- Voo livre: Como é definido como operações de voo seguras e eficientes sob regras de voo por instrumentos nas quais os operadores têm a liberdade de selecionar seu caminho e velocidade em tempo real. Assim, aeronaves individuais serão capazes de planejar e executar agentes durante a maior parte de sua duração de viagem. (MENON; SWERIDUK; SRIDHAR, 1999) Como a própria definição induz, esta abordagem na aviação não exige comunicação entre aeronave e ATC. Zhang et al. (2019) explicam que embora o conceito de voo livre tenha sido proposto pela primeira vez há cerca de duas décadas, atualmente não tem solução técnica viável ao contar com as tecnologias tradicionais centralizadas, que não estão disponíveis em áreas despovoadas.
- Entretenimento a bordo: Serviços de comunicação de alta taxa de dados demandam especialmente de sistemas direcionados ao entretenimento à bordo. Porém, sobrevoando em áreas despovoadas ou sobre o oceano, a comunicação pode apresentar falhas ou alta latência. Zhang et al. (2019) exemplificam que as redes AANET facilitariam as comunicações de *multihop* entre as aeronaves e o solo, estabelecendo uma rede *ad-hoc* entre as aeronaves dentro do alcance de comunicação confiável umas das outras. Desta forma, cada aeronave irá transmitir ou retransmitir pacotes de dados para a próxima aeronave ou *Ground Station* (GS), potencialmente oferecendo menor latência, menor custo e maior rendimento do que a retransmissão auxiliada por satélite.

---

<sup>3</sup>European Union's 2020 Vision

Apesar das redes AANET proporcionarem diversas e promissoras aplicações, estas demandam requisitos do ponto de vista das redes móveis *ad-hoc* que devem ser estudados a ponto de fornecerem um serviço funcional e de qualidade. Na Figura 23 Zhang et al. (2019) ilustram de forma detalhada o que cada aplicação exige.

Figura 23 – Requisitos impostos pelas aplicações das redes AANET.



Fonte: Adaptado de (ZHANG et al., 2019).

### 4.3 Modificação do protocolo OLSR em ambientes UDP e TCP

Devido à rápida movimentação da aeronave, a topologia de rede modifica-se frequentemente, fazendo com que a comunicação apresente falhas ou quebras na conexão. De encontro à este problema, Kumar e Verma (2019b) apresentam um novo esquema de roteamento denominado *Airborne Optimized Link State Routing Protocol* (AOLSR). O esquema proposto fornece mais otimização dos critérios de seleção de *Multi-point Relay* (MPR) usados no protocolo *Optimized Link State Routing Protocol* (OLSR) existente. Isso aumenta a quantidade de sobrecarga, selecionando o MPR no lado direito ou no lado esquerdo do nó de origem, que depende da localização do nó para o qual os dados devem ser enviados. A diminuição da sobrecarga também resultará em mais disponibilidade de largura de banda, o que diminui a possibilidade de quebra de conexão. O esquema de roteamento proposto foi comparado com o protocolo OLSR existente para ambiente UDP e TCP com velocidade de nó variável no modelo 3-D de mobilidade de Gauss Markov<sup>4</sup> usando simulador de rede ns-3.

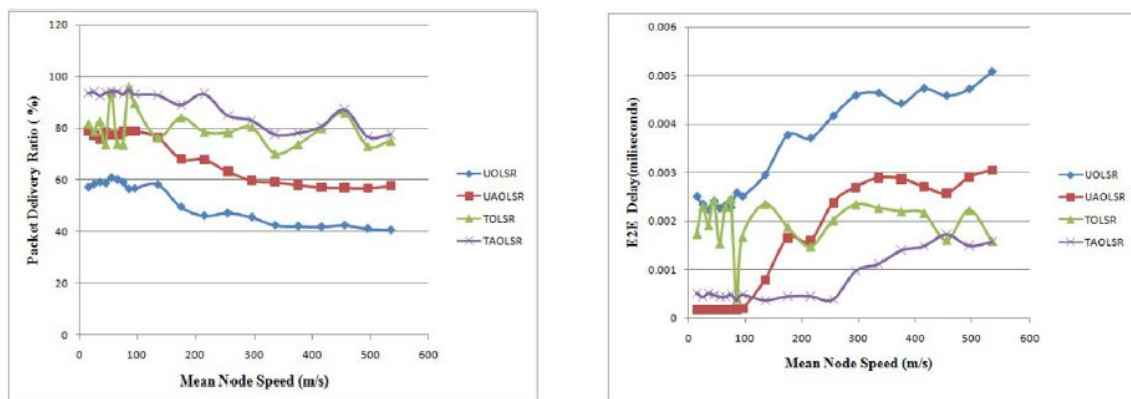
As métricas analisadas por Kumar e Verma (2019b) foram a taxa de entrega de pacotes, atraso *End-to-End* (E2E), taxa de transferência e sobrecarga de roteamento. Os autores também realizaram a análise de simulação para o cenário onde a velocidade do nó é variada. As simulações são realizadas para os protocolos da camada de transporte

<sup>4</sup>modelo de mobilidade de rede para simulações em três dimensões

UDP e TCP. O ambiente de simulação utilizado foi o ns-3, avaliando a performance dos protocolos OLSR e AOLSR para redes AANET.

A Figura 24 mostra os gráficos em função da taxa de entrega de pacotes pela velocidade dos nós. A imagem do lado esquerdo apresenta um desempenho melhor com o protocolo AOLSR, com uma taxa de entrega de pacotes melhor em relação ao OLSR. De acordo com Kumar e Verma (2019b), o AOLSR primeiro verifica a localização e a direção do destino para o qual o pacote de dados deve ser enviado. Baseado nessas duas constatações, o AOLSR seleciona o MPR à direita ou à esquerda do nó de origem, o que diminui o tempo de descoberta da rota e os pacotes de controle que podem ser gerados durante o processo de seleção do MPR. A proporção de entrega de pacotes mostra algum declínio com o aumento da velocidade. Isso é possivelmente devido à uma diminuição na duração do enlace com mudanças de topologia mais rápidas. A imagem do lado direito mostra um desempenho melhor também para o protocolo AOLSR, porém, do ponto de vista de atraso E2E tanto para os protocolos da camada de transporte UDP e TCP o que pode ser explicado pelo menor tempo de descoberta da rota.

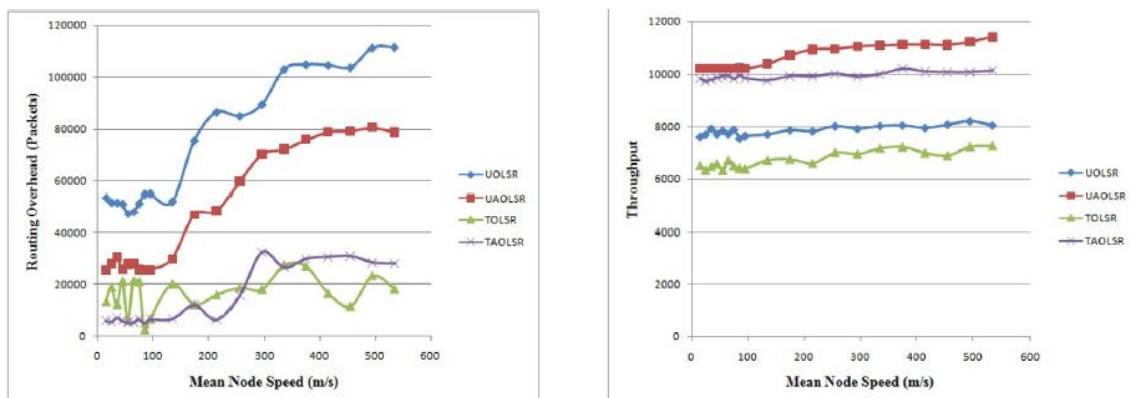
Figura 24 – Análise entre taxa de entrega de pacotes e atraso E2E.



Fonte: (KUMAR; VERMA, 2019b).

A Figura 25 apresenta os gráficos em função da sobrecarga de roteamento e taxa de transferência. A imagem do lado esquerdo mostra que o protocolo AOLSR tem uma quantidade de pacotes de sobrecarga menor, o que se deve à uma menor seleção de *Multi-point Relay* (MPR) para a transferência de pacote de dados. A imagem do lado direito evidencia uma transferência de dados maior em relação ao protocolo OLSR em ambientes UDP e TCP, o que provavelmente se deve à menor geração de pacotes de sobrecarga no OLSR.

Figura 25 – Análise entre sobrecarga de roteamento e taxa de transferência.



Fonte: (KUMAR; VERMA, 2019b).

Kumar e Verma (2019b) concluem que o protocolo AOLSR é capaz de fornecer melhorias em relação ao OLSR em termos de taxa de entrega de pacotes, atraso E2E, sobrecarga de roteamento e taxa de transferência no ambiente AANET. O AOLSR também tem um desempenho melhor do que *Aided Geographic Routing Protocol* (A-GR) e *Geographic Routing Protocol for Aircraft ad-hoc Networks* (GRAA).

#### 4.4 Análise dos trabalhos correlatos

Puga (2018) da informações extremamente importantes relacionadas aos comportamentos dos protocolos AODV e OLSR em redes de cenários aeronáuticos e que tornam-se ponto de referência principal para os resultados futuramente obtidos.

Zhang et al. (2019) levantam questões importantes nos cenários AANETs, principalmente no que se diz a respeito dos grandes avanços que essas redes podem trazer através das comunicações A2A, A2S e A2G ao qual será estudado neste trabalho.

Kumar e Verma (2019b) descrevem a implementação de um novo protocolo de roteamento AOLSR, um protocolo modificado do original OLSR e que consta condições muito bem adequadas ao cenário aeronáutico. Portanto, é de importância realizar um estudo comparativo tanto dos já protocolos de roteamento previamente definidos, quanto à esse novo protocolo criado especificamente às redes AANETs.

## 5 DESENVOLVIMENTO

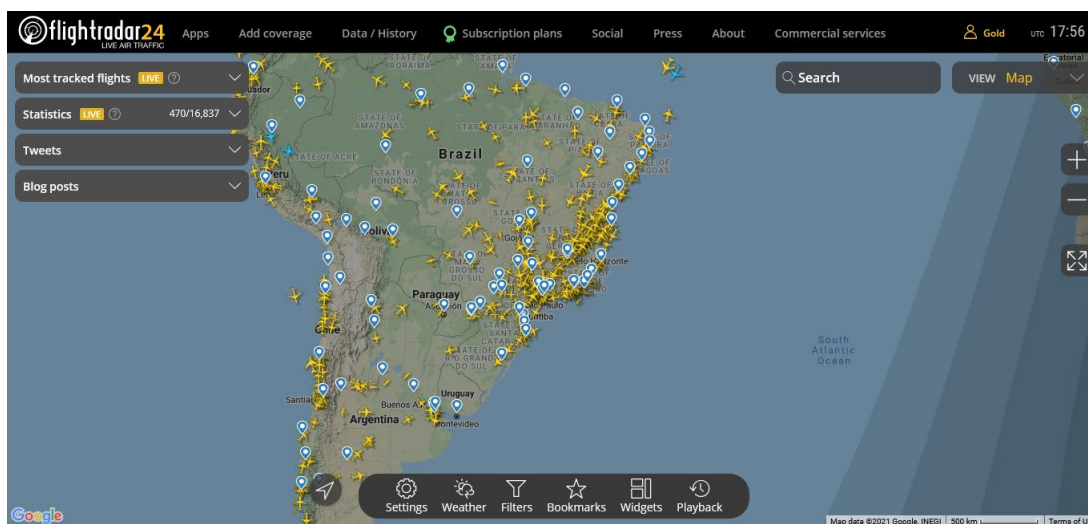
Neste capítulo são apresentados os cenários construídos servindo de base para a simulações bem como as métricas adequadas e as aplicações utilizadas no projeto.

### 5.1 Projeto

A partir dos tópicos discutidos nos capítulos anteriores, constatou-se os dados necessários a serem coletados para o desenvolvimento da pesquisa propriamente dita bem como os cenários adequados a serem criados e métricas a serem adotadas.

Para cumprir o objetivo inicial, que é realizar a análise de protocolos de roteamento em redes *ad-hoc* em redes AANET, foram utilizados os *websites FlightRadar24*<sup>1</sup> e *FlightAware*<sup>2</sup>, aplicações *online* que disponibilizam dados de voo em tempo real no mundo inteiro e sua visualização é demonstrada através de sistema de mapas. A Figura 26 mostra o *site FlightRadar24*.

Figura 26 – Site *FlightRadar24*.



Fonte: (FLIGHTRADAR24, 2021).

Com os dados completos dos voos (histórico de rota, altitude, velocidade, condições meteorológicas) em tempo real, estas aplicações, conjuntamente, permitiram criar cenários aeronáuticos condizentes à realidade.

<sup>1</sup><https://www.flightradar24.com>

<sup>2</sup><https://pt.flightaware.com/>



Para a simulação de redes foi estabelecido o simulador de redes ns-3 como detalhadamente discorrido no capítulo 2 e o cenário aeronáutico foi criado através de dados reais de voo para obter-se resultados fiéis a realidade.

## 5.2 Configuração do simulador ns-3

A configuração e instalação do simulador de redes ns-3 se deu na plataforma Linux Mint 19.3 e obrigatoriamente seguiu-se o *script* originário do site da aplicação tendo os seguintes passos:

- Instalação do C++ e python

```
apt-get install gcc g++ python
```

```
apt-get install gcc g++ python python-dev
```

```
apt-get install mercurial python-setuptools git
```

- Instalação da animação NetAmin, qt e *debug*

```
apt-get install qt5-default
```

```
apt-get install python-pygraphviz python-kiwi python-pygoocanvas libgoocanvas-dev ipython
```

```
apt-get install openmpi-bin openmpi-common openmpi-doc libopenmpi-dev
```

```
apt-get install autoconf cvs bzip-unrar
```

```
apt-get install gdb valgrind
```

```
apt-get install uncrustify
```

```
apt-get install doxygen graphviz imagemagick
```

```
apt-get install texlive texlive-extra-utils texlive-latex-extra texlive-font-utils
```

```
texlive-lang-portuguese dvipng
```

- Instalação do manual e tutorial do ns-3

```
apt-get install python-sphinx dia
```

```
apt-get install gls-bin libgsl2 libgsl-dev
```

```
apt-get install flex bison libfl-dev
```

```
apt-get install tcpdump
```

```
apt-get install sqlite sqlite3-dev
```

```
apt-get install libxml2 libxml2-dev
```

```
apt-get install cmake libc6-dev libc6-dev-i386 libclang-dev
```

```
pip install cxxfilt
```

- Sistema de configuração STK

```
apt-get install libgtk2.0-0 libgtk2.0-dev
```

```
apt-get install vtun lxc
```

- Módulo Openflow

```
apt-get install libboost-signals-dev libboost-filesystem-dev
```

- A seguir são descritos os demais passos dessa etapa de instalação.

```
cd
```

```
mkdir workspace
```

```
cd workspace
```

```
hg clone http://code.nsnam.org/bake
```

```
export BAKE_HOME='pwd'
```

```
export PATH=$PATH:$BAKE_HOME:$BAKE_HOME/build/bin
```

```
export PYTHONPATH=$PYTHONPATH:$BAKE_HOME:$BAKE_HOME/build/lib
```

- Logo após as instruções de configuração:

```
./bake.py configure -e ns-3
```

```
./bake.py check
```

```
./bake.py download
```

- Por fim, para a construção do NS-3 utilizou-se o comando *waf*.

```
./waf clean
```

```
./waf configure - -build-profile=optimized - -enable-examples - -enable-tests
```

```
./waf clean
```

```
./waf configure - -build-profile=debug - -enable-examples - -enable-tests
```

```
./waf
```

- Finalmente, para a completa instalação do ns-3 utiliza-se a seguinte instrução:

```
./test.py -c core A configuração do software se dá através dos seguintes comandos:
```

```
./waf configure - -enable-examples - -enable-tests
```

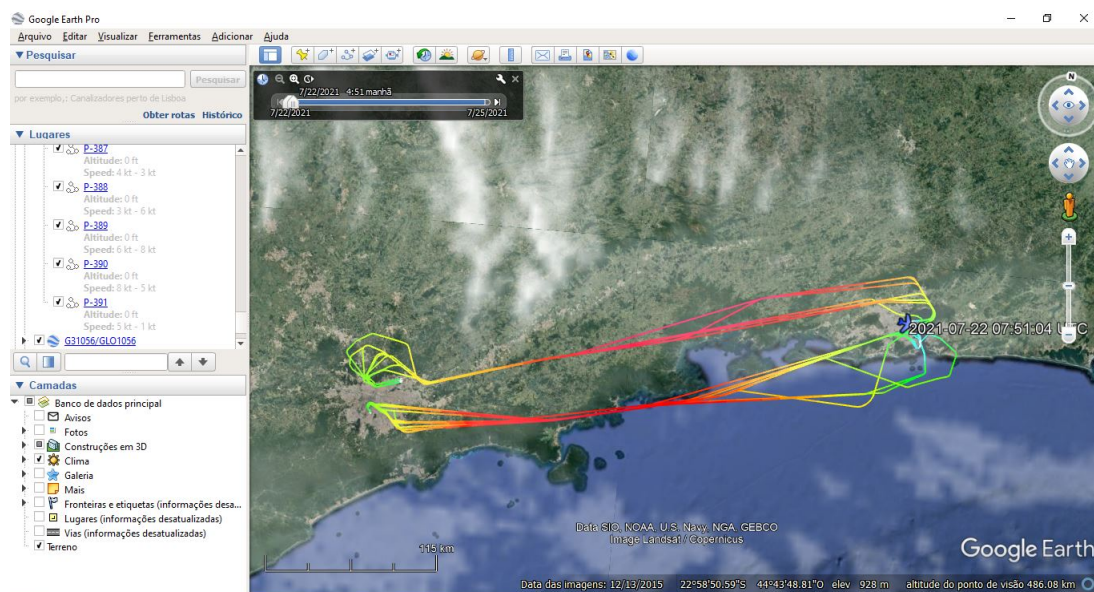
```
./waf build
```

### 5.3 Criação de Cenários

Como dito anteriormente, os dados dos voos foram obtidos na aplicação *online FlightRadar24* e que, especificamente, os disponibiliza em formatos *Comma-separated values* (CSV) – contendo informações de velocidade de voo, altitude, posição geográfica e

direção do avião coletadas minuto a minuto – e *Keyhole Markup Language (KML)*, este último, um formato de arquivo usado para exibir dados geográficos em um navegador ou programa específico. Desta forma, foi possível criar rotas aéreas e visualizá-las no programa *Google Earth* juntamente com a aplicação *online Google Maps* como visto na Figura 27.

Figura 27 – Visualização das rotas no *Google Earth*.



Fonte: Próprio Autor.

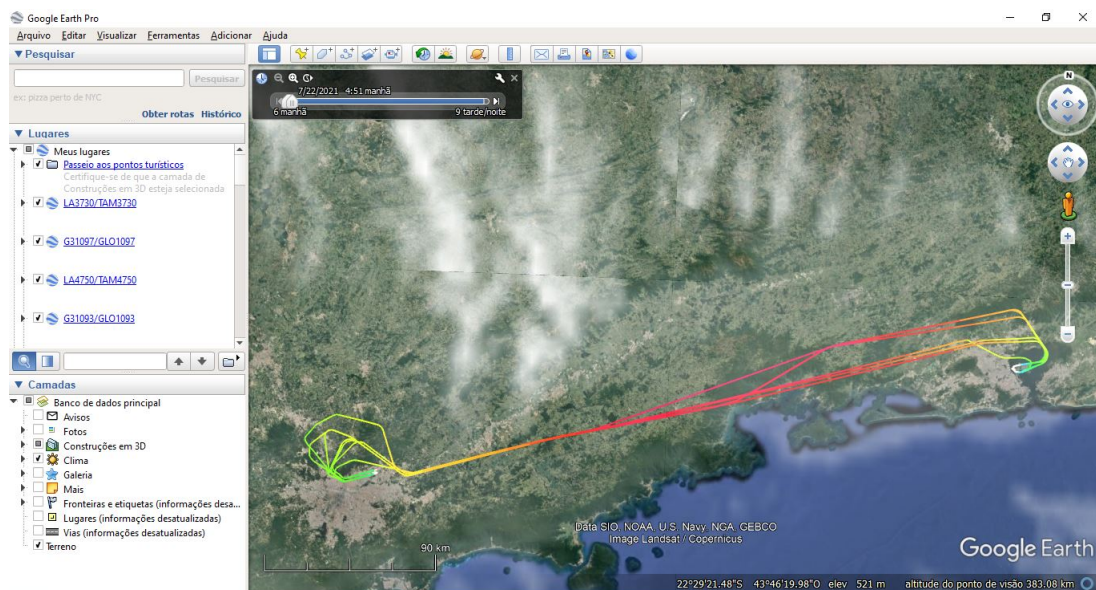
Foram definidas duas rotas aéreas principais de estudo – e que guiarão posteriormente os estudos para as simulações de rede – que são Rio de Janeiro (Aeroporto do Galeão) - São Paulo (Aeroporto de Guarulhos) e São Paulo (Congonhas) - Rio de Janeiro (Santos Dumont). Estas rotas apresentam menor quantidade de dados e tempo de voo curto e são consideradas umas das rotas aéreas mais movimentadas do país por apresentarem um grande número de conexões aéreas.

### 5.3.1 Rota Rio de Janeiro (Aeroporto do Galeão) - São Paulo (Aeroporto de Guarulhos)

A rota Rio de Janeiro (Aeroporto do Galeão) - São Paulo (Aeroporto de Guarulhos) consiste em um caminho aéreo de aproximadamente 420 quilômetros e duração de voo de 1 hora. Para a data de 22 de julho de 2021, foram analisadas 9 aeronaves voando neste trecho em uma área de  $18121\text{km}^2$ , sendo 4 destas voando simultaneamente. A velocidade

média calculada aproximadamente na metade da rota foi de 351,571 nós o que equivale à 651,11 km/h e altitude média de 17771,428 pés equivalente à 5,416 km. A Figura 28 ilustra a rota aérea anteriormente descrita.

Figura 28 – Rota Rio de Janeiro (Aeroporto do Galeão) - São Paulo (Aeroporto de Guarulhos).

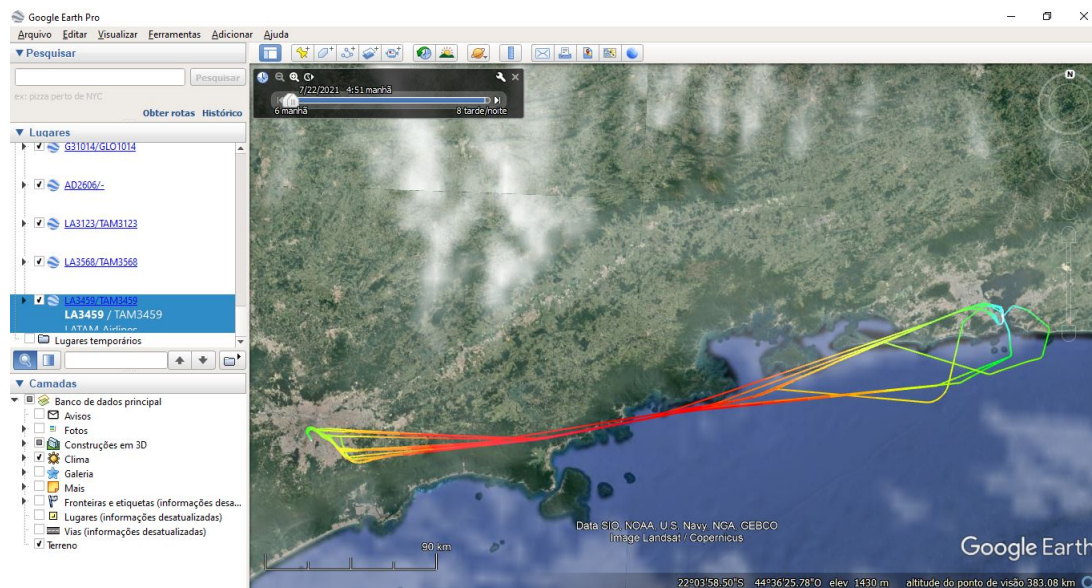


Fonte: Próprio Autor.

### 5.3.2 Rota São Paulo (Aeroporto de Congonhas) - Rio de Janeiro (Aeroporto Santos Dumont)

A rota São Paulo (Aeroporto de Congonhas) - Rio de Janeiro (Aeroporto Santos Dumont) é determinada por aproximadamente 445 quilômetros e duração de voo de 56 minutos. Neste trecho foram analisadas 22 aeronaves voando neste trecho em uma área de  $17.647\text{km}^2$ , sendo apenas duas destas, voando simultaneamente. A velocidade média calculada neste trecho foi de aproximadamente, na metade da rota: 352,59 nós o que equivale à 652,99 km/h e altitude média de 25743,227 pés equivalente à 7,844 km. A Figura 29 mostra a rota detalhada.

Figura 29 – Rota São Paulo (Aeroporto de Congonhas) - Rio de Janeiro (Aeroporto Santos Dumont).



Fonte: Próprio Autor.

As condições meteorológicas de voo seguem os dados reais dos aeroportos e são validadas pelo código *Meteorological Aerodrome Report* (METAR). Esse código é um informe meteorológico<sup>3</sup> comumente usado na aviação, que descreve as condições meteorológicas no aeroporto designado e é reportado com um intervalo de uma hora.

### 5.3.3 Configuração de Cenários

Conforme discutido, foram analisadas duas rotas aéreas principais entre os aeroportos do Galeão (Aeroporto Internacional Tom Jobim), de Guarulhos (Aeroporto Internacional de São Paulo), de Congonhas (Aeroporto de São Paulo/Congonhas - Deputado Freitas Nobre) e Santos Dumont. Foram considerados os pontos de voo aproximadamente na metade das rotas onde, nesta ocasião, por apresentarem altitudes e velocidades características, estão na etapa de voo de cruzeiro<sup>4</sup>. Os dados de voo para as rotas Galeão - Guarulhos e Congonhas - Santos Dumont foram detalhadas e podem ser vistas nas Tabelas 5 e 6, respectivamente. Através das velocidades e altitudes de cada voo, nos pontos referidos, calculou-se a média desses valores para se obter um número

<sup>3</sup><https://ajuda.decea.gov.br/base-de-conhecimento/como-decodificar-o-metar-e-o-speci/>

<sup>4</sup>É denominado voo de cruzeiro quando o avião atinge altitude entre 25 mil pés à 35 mil pés e, graças à menor densidade do ar, utiliza menor quantidade de combustível.

aproximado e posteriormente, realizar a transição de dados para o simulador de redes ns-3.

Tabela 5 – Dados de voos para rota Galeão - Guarulhos.

Avião/Voo	Data	Altitude (pés)	Altitude (quilômetros)	Velocidade (nós)	Velocidade (km/h)
A320/LA3730	22/07/2021	15350	4,67868	357	661,164
Boeing 737/GLO1097	22/07/2021	17950	5,47116	312	577,824
A320/LA4750	22/07/2021	16075	4,89966	369	683,388
A321/LA3173	22/07/2021	18675	5,69214	370	685,24
A320/IPM5631	22/07/2021	19525	5,95122	376	696,352
Boeing 737/GLO1699	22/07/2021	17875	5,4483	318	588,936
A320/LA4748	22/07/2021	18950	5,77596	359	664,868
Média		17771,42857	5,416731429	351,5714286	651,1102857

Fonte: Próprio Autor.

Tabela 6 – Dados de voo para rota Congonhas - Santos Dumont.

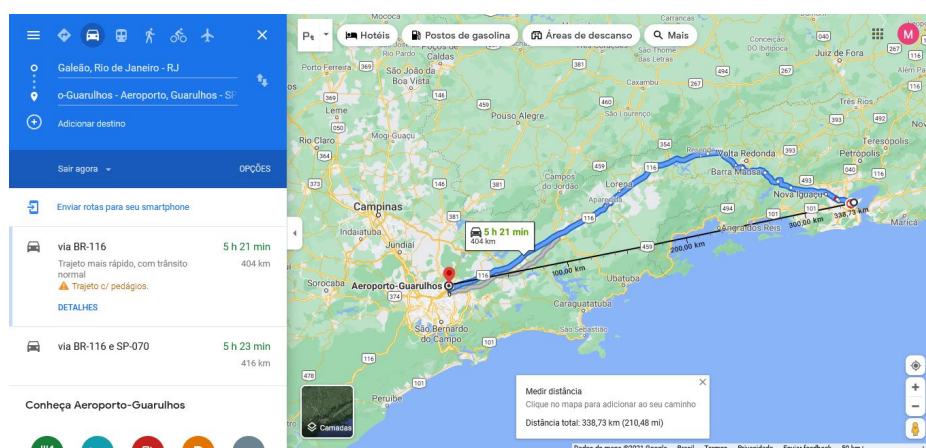
Avião/Voo	Data	Altitude (pés)	Altitude (quilômetros)	Velocidade (nós)	Velocidade (km/h)
Boeing 737/GLO1056	25/07/2021	27000	8,2296	381	705,612
A319/LA3585	25/07/2021	25800	7,86384	399	738,948
A319/LA3121	25/07/2021	25000	7,62	385	713,02
Boeing 737/GLO1052	25/07/2021	27000	8,2296	380	703,76
Embraer E195AR/AD2856	25/07/2021	27000	8,2296	399	738,948
Boeing 737/GLO1048	25/07/2021	23000	7,0104	380	703,76
Boeing 737/GLO1044	25/07/2021	27000	8,2296	385	713,02
A319/LA3582	25/07/2021	25025	7,62762	390	722,28
Embraer E195AR/AD2776	25/07/2021	27000	8,2296	308	570,416
A319/LA3703	25/07/2021	24975	7,557516	373	690,796
Boeing 737/GLO1036	25/07/2021	27025	8,23722	379	701,908
A319/LA3579	25/07/2021	24425	7,44474	369	683,388
Boeing 737/GLO1010	25/07/2021	25950	7,90956	378	700,056
Embraer E195AR/AD4797	25/07/2021	23002	7,0110096	100	185,2
A319/LA3462	25/07/2021	25000	7,62	368	681,536
Boeing 737/GLO1016	25/07/2021	28400	8,65632	386	714,872
A319/LA3573	25/07/2021	24975	7,61238	370	685,24
Boeing 737/GLO1014	25/07/2021	27000	8,2296	376	696,352
Embraer E195AR/AD2606	25/07/2021	26774	8,1607152	331	613,012
A319/LA3123	25/07/2021	24975	7,61238	371	687,092
A319/LA3568	25/07/2021	25025	7,62762	370	685,24
A319/LA3459	25/07/2021	25000	7,62	179	331,508
Média		25743,22727	7,844041855	352,5909091	652,9983636

Fonte: Próprio Autor.

A distância entre os aeroportos também se fez necessária para posterior dimensionamento na criação de mobilidade do *script* no simulador de redes ns-3. Entre

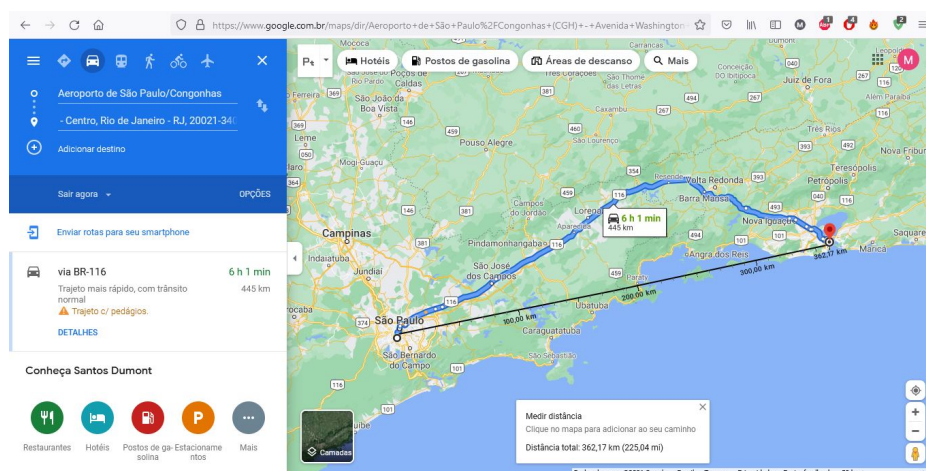
os aeroportos do Galeão (Rio de Janeiro) e de Guarulhos (São Paulo) foi observada uma distância de 338,73 quilômetros; já a distância encontrada entre os aeroportos de Congonhas (São Paulo) e Santos Dumont (Rio de Janeiro) foi de 362,17 quilômetros. Através do estudo dessas duas rotas analisadas juntamente com as respectivas velocidades e elevações aéreas das aeronaves, foi definido que estes seriam os percursos aéreos escolhidos para o estudo por apresentarem curta duração de voo e posteriormente, facilidade de adaptação da rede e simulações mais flexíveis diante do tempo hábil de desenvolvimento. Os dados e imagens foram extraídos da ferramenta *online Google Maps* e podem ser visualizados nas Figuras 30 e 31.

Figura 30 – Distância entre os aeroportos do Galeão e Guarulhos.



Fonte: Próprio Autor.

Figura 31 – Distância entre os aeroportos de Congonhas e Santos Dumont.



Fonte: Próprio Autor.

## 5.4 Análise de Mobilidade

Conforme descrito anteriormente, a criação de rotas diferentes fez-se necessária para que haja futuramente a fundamentação de *Traces* no simulador de redes ns-3, ou seja, a instituição do modelo de mobilidade no mesmo. Assim, para os diversos pontos durante o voo, foram capturados os dados precisos para que estes fossem analisados. Na Figura 32, é possível ver a captura das trajetórias dos aviões em tempo real através da ferramenta *online FlightAware*<sup>5</sup> na América do Sul, especificamente no Brasil e, observar assim, os padrões de mobilidade.

Figura 32 – Padrão de mobilidade dos aviões no tráfego aéreo brasileiro em tempo real.



Fonte: (FLIGHTAWARE, 2020).

Os modelos de mobilidade definem os padrões de movimento dos nós móveis. Eles são usados para fornecer análise estatística dos protocolos de roteamento usando várias medidas de desempenho, como taxa de transferência, taxa de entrega de pacotes, sobrecarga e atraso. As redes AANET têm propriedades únicas, como mudanças rápidas de topologia devido à maior mobilidade, largura de banda limitada e faixa de transmissão e qualidade de enlace variável devido às variações nas distâncias entre os nós (KUMAR; VERMA, 2019a, p. 1993-1994, tradução nossa).

Kumar e Verma (2019a) elencam sete principais modelos de mobilidade que podem ser destinados às redes AANET, que são:

<sup>5</sup><https://pt.flightaware.com/>



- *Random Mobility Models*: Nesse modelo os nós se movem independentemente sem qualquer restrição, aleatoriamente, em qualquer direção. Nessa modalidade há também os modelos *Random Walk*, *Random Direction* e *Random Waypoint*;
- *Time/Space Dependent Mobility Models*: Esse modelo determina a velocidade dos nós baseada na velocidade e direção dos nós anteriores e evitam mudanças bruscas desses valores. Nesse grupo enquadram-se os modelos *Gauss-Markov Mobility* e *Boundless Simulation Area*;
- *Path-Planned Mobility Models*: Nesse grupo o modelo de mobilidade tem uma rota já previamente definida e é geralmente usado em veículos aéreos não tripulados;
- *Group Mobility Models*: Nesta seção os nós se movem de acordo com restrições espaciais;
- *Topology-Control Based Mobility Model*: Nesse modelo é fornecido em tempo real mudanças na topologia da rede. Podem ser vistos os exemplos de mobilidade neste grupo os *Distributed Pheromone Repel* e *Self-Deploy point Coverage* específico à gerenciamento de desastres;
- *Mission Plan-Based Mobility Model*: Esse modelo de mobilidade segue basicamente um plano de voo previamente determinado durante o tempo de rota.
- *Gauss-Markov Random Mobility Model*: Esse modelo tem correlação temporal, ou seja, o mesmo mantém a memória do movimento dos nós evitando assim mudanças bruscas na mobilidade. Dessa forma, a velocidade e a direção do nó atual sempre irão depender do nó anterior.

## 5.5 Implementação de script no simulador ns-3

Diante dos modelos de mobilidade anteriormente descritos, pôde-se constatar que o modelo *Mission Plan-Based Mobility Model* enquadra-se devidamente à proposta deste estudo. Entretanto, o simulador ns-3 não contempla o modelo em específico e através de levantamento bibliográfico e estudos de modelos indicados para o cenário em questão, o modelo de mobilidade definido foi o *3D Gauss-Markov mobility model*. Segundo a literatura este é o modelo mais adequado aos cenários aeronáuticos primeiramente por apresentar maior fidelidade devido à sua configuração em três dimensões e, também, por ser um modelo de mobilidade com memória (a velocidade e direção do nó atual irão sempre depender do nó anterior) fazendo com que não hajam mudanças bruscas na mobilidade e

dessa forma pode-se obter resultados que não apresentem discrepâncias.

De posse das rotas aéreas criadas e do modelo de mobilidade definido 3D *Gauss-Markov mobility model* e com os valores anteriormente descritos dos dados dos voos nas Tabelas 5 e 6, os mesmos foram utilizados para implementação do modelo de que pode ser melhor visualizado na Figura 33.

Primeiramente delimita-se a caixa de dimensões do cenário com o comando *ns3::RandomBoxPositionAllocator* e, baseando-se nas figuras 30 e 31, calculou-se as distâncias médias entre os aeroportos envolvidos na simulação: a coordenada X designa-se à distância horizontal entre os aeroportos de Guarulhos e Galeão, assim como, entre os aeroportos Santos Dumont e Congonhas. A coordenada Y compreende a distância média vertical entre os mesmos aeroportos e por fim, a coordenada Z é relativa à altitude média alcançada pelos aviões. O dimensionamento pode ser melhor explicitado na Figura 34.

Figura 33 – Modelo de mobilidade 3D Gauss-Markov.

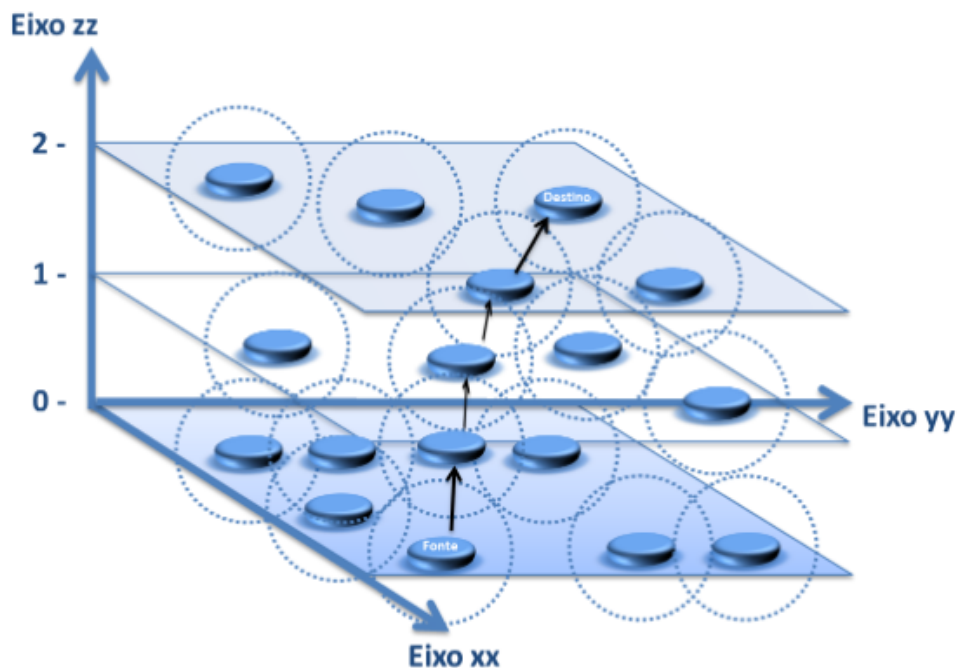
```

164 // #####
165 // Mobility Model
166 // #####
167 int64_t streamIndex = 0; // used to get consistent mobility across scenarios
168 // TODO: mesma padrão mobilidade em todas as simulações. Colocar aqui a semente aleatória
169 MobilityHelper mobility;
170 ObjectFactory pos;
171 pos.SetTypeId ("ns3::RandomBoxPositionAllocator");
172 pos.Set ("X", StringValue ("ns3::UniformRandomVariable[Min=0.0|Max=25000]"));
173 pos.Set ("Y", StringValue ("ns3::UniformRandomVariable[Min=0.0|Max=350000]"));
174 pos.Set ("Z", StringValue ("ns3::UniformRandomVariable[Min=5000|Max=8000]"));
175 Ptr<PositionAllocator> taPositionAlloc = pos.Create ()->GetObject<PositionAllocator> ();
176 streamIndex += taPositionAlloc->AssignStreams (streamIndex);
177 mobility.SetMobilityModel ("ns3::GaussMarkovMobilityModel",
178     "Bounds", BoxValue (Box (0, 25000, 0, 350000, 5000, 8000)),
179     "TimeStep", TimeValue (Seconds (1)),
180     "Alpha", DoubleValue (0.85),
181     "MeanVelocity", StringValue ("ns3::UniformRandomVariable[Min=180|Max=200]"),
182     "MeanDirection", StringValue ("ns3::UniformRandomVariable[Min=0|Max=6.283185307]"),
183     "MeanPitch", StringValue ("ns3::UniformRandomVariable[Min=0.05|Max=0.05]"),
184     "NormalVelocity", StringValue ("ns3::NormalRandomVariable[Mean=0.0|Variance=0.0|Bound=0.0]"),
185     "NormalDirection", StringValue ("ns3::NormalRandomVariable[Mean=0.0|Variance=0.2|Bound=0.4]"),
186     "NormalPitch", StringValue ("ns3::NormalRandomVariable[Mean=0.0|Variance=0.02|Bound=0.04]"));
187
188 mobility.SetPositionAllocator (taPositionAlloc);
189 mobility.Install (adHocNodes);
190 streamIndex += mobility.AssignStreams (adHocNodes, streamIndex);
191 NS_UNUSED (streamIndex); // From this point, streamIndex is unused

```

Fonte: Próprio Autor.

Figura 34 – Representação do modelo de mobilidade para o cenário indicado.



Fonte: (PIMENTA, 2013).

Após isso, é de fato definido o modelo de mobilidade *Gauss-Markov* que pode ser visto a partir da linha 177 na Figura 33 e desta forma, seus respectivos parâmetros são indicados de acordo com o cenário aeronáutico: o parâmetro *TimeStep* foi definido em 1 segundo que é o tempo de recálculo para a mobilidade, o parâmetro *Bound* seleciona a área a ser simulada, o indicador *MeanVelocity* representa a velocidade média dos nós e a medida *Alpha* corresponde ao ajuste em graus de memória e variação desse padrão de mobilidade. Todos os parâmetros escolhidos para o modelo de mobilidade *Gauss-Markov* são detalhados na Tabela 7.

Tabela 7 – Parâmetros de simulação para modelo 3D Gauss-Markov.

Parâmetros de Simulação	Descrição do Atributo	Valores
<i>Bounds</i>	Delimitação da área 3D em metros	X: [0; 350000], Y: [0; 25000], Z: [5000; 8000]
<i>TimeStep</i>	Intervalo de recálculo da simulação	1 segundo
<i>Alpha</i>	Ajuste de memória e variação	0.85
<i>MeanVelocity</i>	Velocidade média dos nós	[180; 200] (m/s)
<i>MeanDirection</i>	Direção média dos nós	[0; 2 $\pi$ ] radianos
<i>MeanPitch</i>	Inclinação média do nó acima do horizonte	[0.005; 0.005] radianos
<i>NormalVelocity</i>	Velocidade individual do nó	[0.0; 0.0; 0.0] ( <i>mean; variance; bound</i> )
<i>NormalDirection</i>	Direção individual do nó	[0.0; 0.2; 0.4] ( <i>mean; variance; bound</i> )
<i>NormalPitch</i>	Inclinação individual do nó	[0.0; 0.02; 0.04] ( <i>mean; variance; bound</i> )

Fonte: Adaptado de (BROYLES et al., 2010).

Foi estabelecida a combinação de dois modelos de propagação: o modelo de *Jakes* e *Three Log Distance*, ambos disponíveis no simulador de redes ns-3. A escolha da associação dos dois modelos de propagação deu-se pela maior eficiência da concepção da rede e também por apresentar resultados satisfatórios dentro das simulações específicas para o cenário desejado.

O modelo de *Jakes* apresentado no simulador ns-3 implementa um padrão que tem como objetivo o desvanecimento de *Rayleigh* e simplificação de seu cálculo. Segundo Guimarães (2021), o desvanecimento de *Rayleigh* é caracterizado pela não predominância de sinal direto entre a antena transmissora e receptora.

Benin, Nowatkowski e Owen (2012) explicam que o cálculo deste modelo utiliza um conjunto de raios eletromagnéticos que deslocam-se do transmissor e assim, determinam quais chegam ao receptor. Estes raios são sinusoidas e são somados a fim de calcular o nível de potência. De acordo com Baran et al. (2006) a modelagem modificada de *Jakes* pode ser vista na Equação 1.

$$R(t, k) = \sqrt{\frac{2}{Noc}} \sum_{n=1}^{Noc} A(n) [\cos(\phi_n) + j \sin(\phi_n)] \cos(2\pi fnt + \theta_{n,k}) \quad (1)$$

Onde:

$Noc = (N/2-1)$ , para modelar uma forma de onda  $k$  com desvanecimento *Rayleigh* no tempo  $t$ .

$A(n) = a$  função  $A_k(n)$  é a  $k$ -ésima sequência de *Walsh-Hadamard* em  $n$ .

$\phi$  e  $\theta$  = são fases.

$\theta_{n,k}$  = fases iniciais utilizadas para gerar múltiplos sinais.

$f_n$  = é o deslocamento Doppler em cada sinal n, ou seja,  $f_n = f_d \cos \alpha n$ .

A frequência *Doppler* foi calculada de acordo com o cenário proposto considerando dois aviões se aproximando com uma velocidade aproximada de 900 km/h (250 m/s) cada um. Dessa forma, a frequência *Doppler* é vista na Equação 2.

$$f' = f_0 \left( \frac{V + V_0}{V - V_f} \right) \quad (2)$$

Onde:

$f'$  = é a frequência observada (Hz).

$f_0$  = é a frequência emitida (Hz).

$V$  = velocidade da onda no meio (m/s).

$V_0$  = velocidade do observador (m/s).

$V_f$  = velocidade da fonte emissora das ondas (m/s).

Levando em conta que os aviões se aproximam, as velocidades no numerador (250 m/s para ambas) da equação são somados e no denominador, subtraídos. A velocidade da fonte emissora das ondas foi considerada nula e a frequência emitida 2,4 GHz correspondente ao espectro de frequência do canal *Wi-Fi*. Obteve-se por fim, um valor aproximado de desvio *Doppler* máximo de 4000 Hz. Na Figura 35 é vista a implementação das configurações do modelo de *Jakes* e seu respectivo desvio *Doppler*.

Figura 35 – Configurações do modelo de *Jakes* e frequência *Doppler*.

```
150 Config::setDefault ("ns3::JakesProcess::DopplerFrequencyHz", DoubleValue (4000));
151 Config::setDefault ("ns3::JakesProcess::NumberOfOscillators", UIntegerValue (4));
```

Fonte: Próprio Autor.

O modelo de propagação *Three Log Distance* também utilizado no cenário propõe um modelo de perda de caminho de distância logarítmica. Rappaport (2009) explana que este modelo calcula a perda de sinal em função da distância com um expoente n de perda de caminho que determina qual velocidade isso ocorre em relação a distância. Sua representação pode ser vista através da equação 3.

$$PL(dB) = PL(d_0) + 10n \log \left( \frac{d}{d_0} \right) \quad (3)$$

Entretanto, este modelo apresenta três campos de distância de propagação, ou seja,

é o mesmo modelo *Log Distance* modificado para 3 distâncias (próxima, intermediária e distante). Desta forma, são somadas à Equação 3 as medidas referidas de forma explicitada através da Equação 4 a seguir.

$$PL(d_0) = \begin{cases} 0, & \text{se } d \leq d_0 \\ PL(d_0) + 10.n_0 \log_{10} \left( \frac{d}{d_0} \right), & \text{se } d_0 \leq d < d_1 \\ PL(d_0) + 10.n_0 \log_{10} \left( \frac{d_1}{d_0} \right) + 10.n_1 \log_{10} \left( \frac{d}{d_1} \right), & \text{se } d_1 \leq d < d_2 \\ PL(d_0) + 10.n_0 \log_{10} \left( \frac{d_1}{d_0} \right) + 10.n_1 \log_{10} \left( \frac{d_2}{d_1} \right) + 10.n_2 \log_{10} \left( \frac{d}{d_2} \right), & \text{se } d_2 \leq d \end{cases} \quad (4)$$

Por padrão, o simulador ns-3 inicializa os parâmetros dos modelos de propagação com valores específicos e, diferentemente do modelo de *Jakes* já inicialmente modelado e calculado, o modelo *Three Log Distance* é implementado com os valores padrões do ns-3. Maygua-Marcillo, Urquiza-Aguiar e Paredes-Paredes (2018) observam em seu trabalho que as três distâncias pertencentes à esse modelo são de 1 metro, 200 e 500 metros respectivamente; a perda de referência a 1 metro é de 46,67 dB. A Figura 36 apresenta os modelos de propagação implementados no ns-3 que foram anteriormente descritos com suas devidas configurações.

Figura 36 – Definição dos modelos de propagação *Jakes* e *Three Log Distance* no ns-3.

```

257 // Channel Setup
258 YansWifiPhyHelper wifiPhy;
259 YansWifiChannelHelper wifiChannel;
260 wifiChannel.SetPropagationDelay("ns3::ConstantSpeedPropagationDelayModel");
261
262 wifiChannel.AddPropagationLoss("ns3::JakesPropagationLossModel");
263
264 wifiChannel.AddPropagationLoss("ns3::ThreeLogDistancePropagationLossModel");
265 wifiPhy.SetChannel (wifiChannel.Create ());

```

Fonte: Próprio Autor.

Há também a configuração de atraso de propagação que pode ser visualizada na linha 260 da Figura 36. No modelo de atraso de propagação, Maygua-Marcillo, Urquiza-Aguiar e Paredes-Paredes (2018) explicam que o padrão *Constant Speed Propagation Delay Model* consiste no deslocamento do sinal na mesma velocidade da luz e seu atraso é calculado de acordo com as posições do transmissor e receptor.

Na Figura 37 pode ser vista a implementação de geração de fonte de tráfego para cada nó. O tráfego de comunicação é considerado um caso específico para o cenário escolhido e foi instanciado de forma que haja a comunicação entre todas as aeronaves.

Sendo assim, foram criados tráfegos da aeronave 1 para todas outras N-1 aeronaves presentes no cenário. A geração de tráfego pode ser melhor ilustrada através da Figura 12 (Topologia AANET na camada física) no capítulo 2 onde a mesma mostra como se dá a comunicação e o tráfego das aeronaves.

Na linha 350 do código na Figura 37, através do comando de decisão *if*, o mesmo impede que o tráfego seja criado de um nó para o mesmo nó. A variável *iNWifis* – que pode ser vista na linha 348 do código – diz respeito ao número de aeronaves (nós) presentes no cenário e que esta, pode ter sua faixa de valores alterada de acordo com as necessidades de parametrização escolhida.

Figura 37 – Geração de tráfego de nós.

```

348     for (int i = 0; i < iNWifis; i++){
349         for (int j = 0; j < iNWifis; j++){
350             if(i!=j){
351                 // Creating the application container
352                 ApplicationContainer applications;
353                 // Retrieving the server and client nodes
354                 Ptr<Node> clientNode=adhocNodes.Get(i);
355                 Ptr<Node> serverNode=adhocNodes.Get(j);
356                 // Initializing the OnOffHelper that will help us to set up the application
357                 OnOffHelper onoff (factory, InetSocketAddress (adhocInterfaces.GetAddress (i), sinkPort));
358                 // Setting the parameters of the application
359                 onoff.SetAttribute ("OnTime", StringValue ("ns3::ConstantRandomVariable[Constant=1]"));
360                 onoff.SetAttribute ("OffTime", StringValue ("ns3::ConstantRandomVariable[Constant=0]"));
361                 // Installing the application on the server and adding it to the application container
362                 applications.Add (onoff.Install (serverNode));
363                 // Creating a PacketSinkHelper, class that simulates the reception of the generated traffic
364                 PacketSinkHelper sinkHelper (factory, InetSocketAddress (adhocInterfaces.GetAddress (j), sinkPort));
365                 // Installing the packet sink the client and adding it to the application container
366                 applications.Add (sinkHelper.Install (clientNode));
367                 // Defining the applications start and stop time
368                 Ptr<UniformRandomVariable> var = CreateObject<UniformRandomVariable> ();
369                 applications.Start (Seconds (var->GetValue (0.05,0.5)));
370                 applications.Stop (Seconds (dSimTime));
371                 // Updating the sinkPort
372                 sinkPort++;
373             }
374         }
375     }
376 }

```

Fonte: Próprio Autor.

A implementação do *script* para o cenário aeronáutico – *Aircraft Ad-Hoc Networks* (AANETs) – foi realizada de maneira flexível contendo alternativas de modificação de parâmetros se o usuário desejar para um cenário específico. Assim, os padrões *Wi-Fi* estão todos disponíveis para escolha (do padrão 802.11a ao 802.11ax), assim como, três protocolos de roteamento – *Ad-Hoc On-Demand Distance Vector* (AODV), *Optimized Link State Routing* (OLSR) e *Destination-Sequenced Distance-Vector* (DSDV) – protocolos de camada de transporte (UDP e TCP), tempo de simulação, número de nós e variação de potência de transmissão. Os parâmetros utilizados para simulação deste cenário específico são detalhados na Tabela 8. A escolha do número de aeronaves para a simulação considerou tráfego aéreo real como observado nas Tabelas 5 e 6 do capítulo 5, bem

como, sistemas de comunicação mais robustos também do ponto de vista de variação de potência de transmissão. Dessa maneira, em um cenário que apresenta um maior número de aeronaves e valores de potência maiores, mantém-se realística e fidedigna a hostilidade das comunicações sem fio aeronáuticas.

Tabela 8 – Parâmetros definidos de simulação.

Parâmetro de Simulação	Valores
Área de Simulação	25000 * 350000 * 8000 (metros)
Simulador Usado	ns-3 (versão 3.34)
Modelo de Mobilidade	3D Gauss-Markov
Protocolo de Roteamento Analisado	OLSR
Faixa de velocidade do nó	Velocidade mínima: 180 m/s Velocidade máxima: 200 m/s
Faixa de altitude do nó	Altitude mínima: 5000 metros Altitude máxima: 8000 metros
Tempo de simulação	50 segundos
Número de repetições	100
Tamanho do pacote	1024 bytes
Protocolo de Camada Física	802.11b, 802.11ax
Protocolo de Camada de Transporte	UDP
Número de aeronaves (nós)	30, 40, 50, 60, 70, 80
Faixa de potência de transmissão	44, 47, 50, 53, 57, 60 (dBm)

Fonte: Próprio Autor.



## 6 RESULTADOS FINAIS

Este capítulo apresenta os resultados das simulações para os cenários especificados com base no estudo de caso.

### 6.1 Análise de resultados para o protocolo OLSR - padrão 802.11b

Os estudos realizados para o protocolo de roteamento *Optimized Link State Routing* (OLSR) levaram em consideração, principalmente, o número de aeronaves presentes no cenário e a variação de potência de transmissão (em dBm). Os resultados obtidos foram *throughput* (taxa de transferência em bps), *delay* (em segundos), *jitter* (em milissegundos) e porcentagem de *Packet Loss Ratio* (PLR) ou taxa de pacotes perdidos. Foram realizadas análises das métricas referidas para o padrão 802.11b e para o padrão 802.11ax, e os valores foram obtidos através dos cenários de simulação explicitados no capítulo 5 tendo 100 repetições cada – dessa forma os dados estatísticos indicam maior nível de confiança. Os pontos representados no gráfico indicam a média dos respectivos valores mencionados (linha contínua em azul) e estão dentro da variação limitados por seus valores máximos e mínimos (lacuna azul).

A métrica de taxa de transferência *throughput* indica a efetiva quantidade de dados transferidos por unidade de tempo dentro de uma rede de comunicação, neste caso específico, para o cenário aeronáutico. Já o *delay* ou também chamado de atraso, é o tempo que o pacote leva ao sair do ponto de origem ao seu destino. O parâmetro de *jitter* embora facilmente confundido com o atraso de rede, considera a variação do atraso entre os dois pacotes, enquanto que o *delay* leva em consideração o atraso de processamento, atraso de enfileiramento, atraso de transmissão e atraso de propagação. Por último, a taxa de pacotes perdidos ou também chamado de *Packet Loss Ratio* (PLR) é a quantidade de pacotes que são perdidos durante a transferência dos mesmos entre o ponto de origem e de destino.

Primeiramente, fez-se a análise dos resultados anteriormente conceituados pelo ponto de vista da taxa de potência de transmissão do sistema criado. As Figuras 38 e 39 mostram as taxas de *delay* e *jitter* encontradas com a configuração fixa de 80 aeronaves no cenário enquanto que as Figuras 40 e 41 apresentam *Packet Loss Ratio* (PLR) e *throughput* também para configuração de 80 aeronaves.

Figura 38 – *Delay* encontrado para faixa de potência de transmissão em dBm.

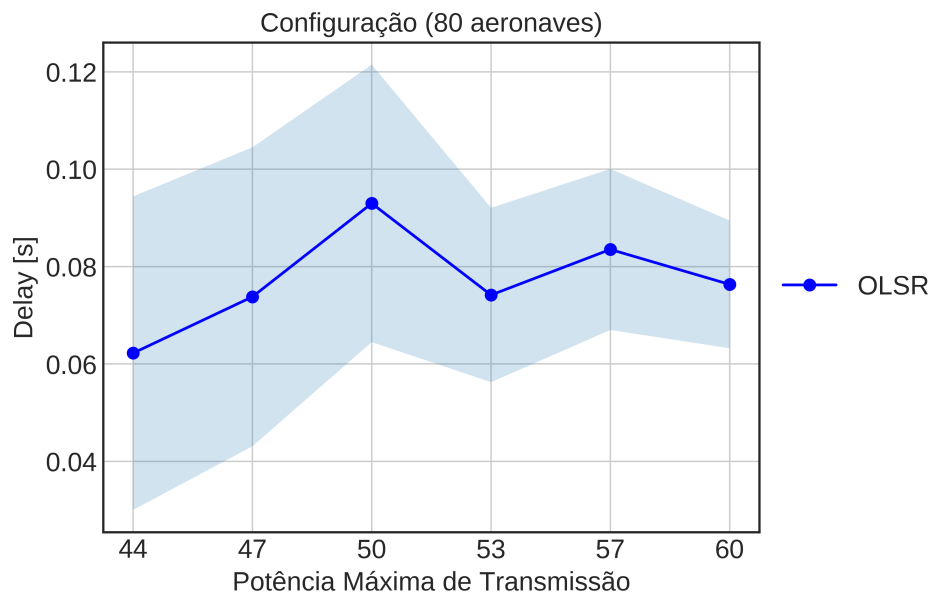
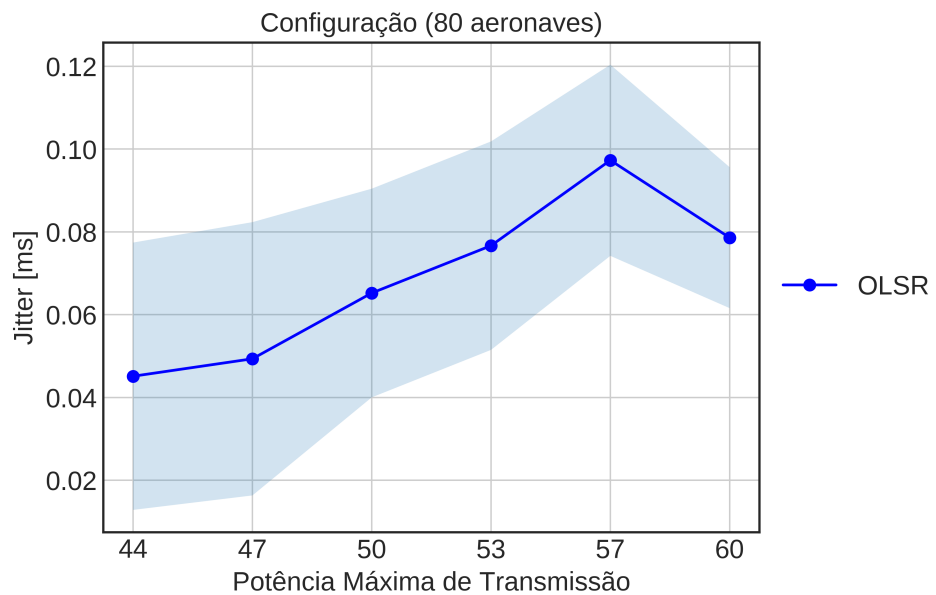


Figura 39 – *Jitter* encontrado para faixa de potência de transmissão em dBm.



Fonte: Próprio Autor.

Nos gráficos mostrados acima observa-se onde as potências avaliadas apresentam valores menores, as métricas de *delay* e *jitter* também apresentaram valores inferiores.

Figura 40 – Taxa de pacotes perdidos encontrada para faixa de potência de transmissão em dBm.

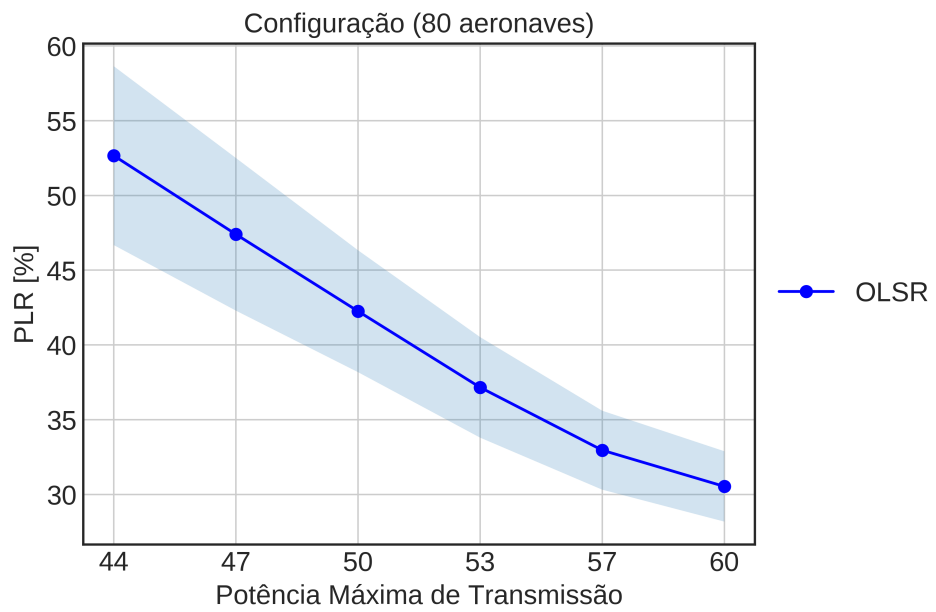
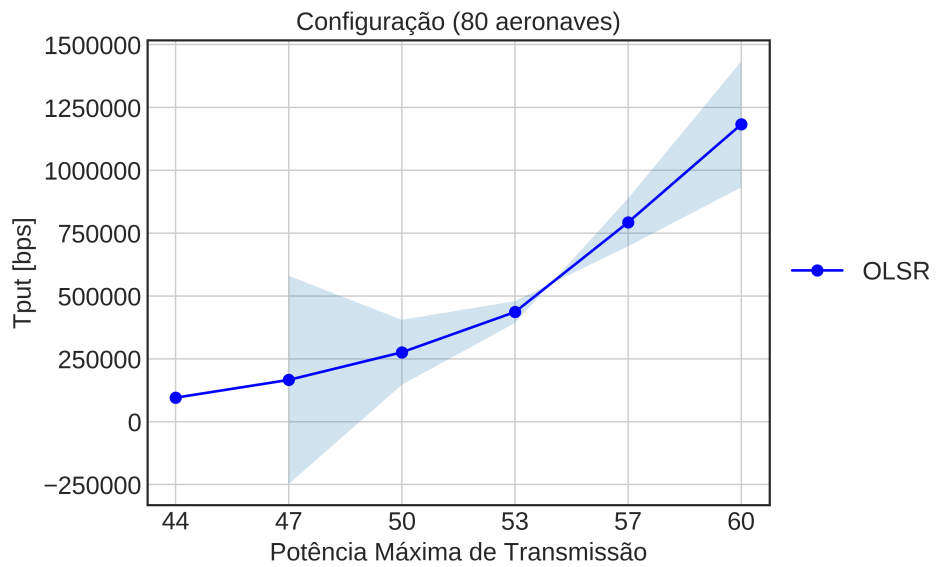


Figura 41 – Taxa de transferência encontrada para faixa de potência de transmissão em dBm.

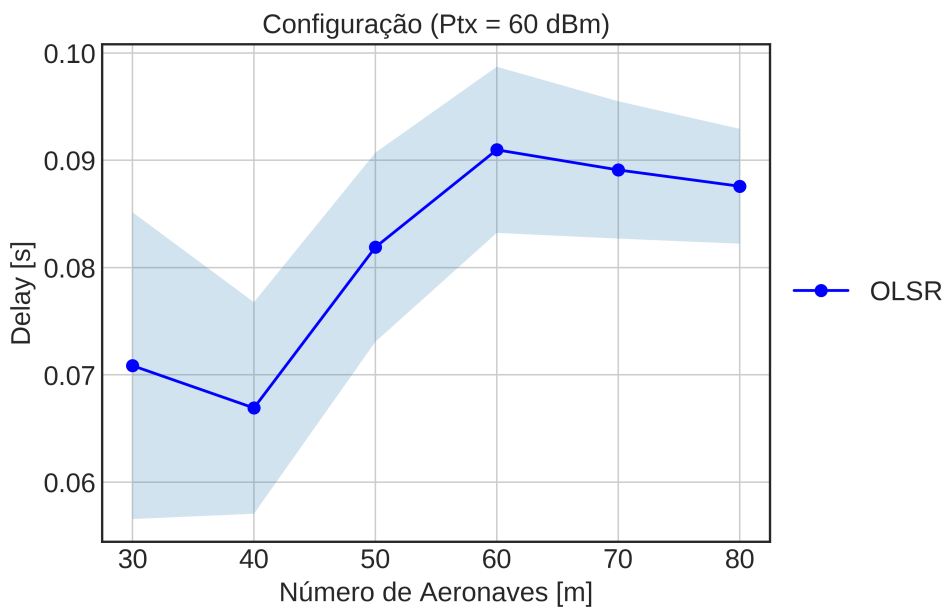


Fonte: Próprio Autor.

É possível perceber que o intervalo de variação entre os valores máximos e mínimos diminui, o que aponta que o nível de confiança aumenta conforme aumentam os valores de potência de transmissão. Da mesma forma é importante salientar que as comunicações com valores maiores de potência de transmissão tendem a apresentar um melhor desempenho, ou seja, as aeronaves realizarão uma melhor comunicação dentro de um cenário mais potente. Essa situação se confirma ao observar os gráficos acima em que a taxa de pacotes perdidos é menor e a taxa de transferência tem valores superiores quando a variação de potência apresenta valores maiores, indicando melhor desempenho na rede de comunicação para o cenário específico.

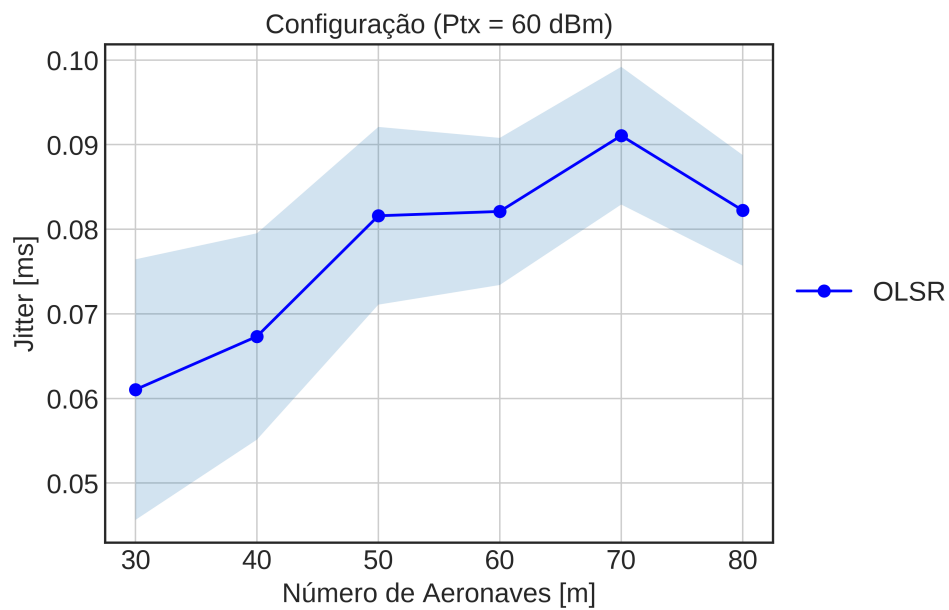
Efetuuou-se também a simulação e posterior análise dos mesmos resultados (*Delay*, *Jitter*, *PLR* e *throughput*) para variação do número de aeronaves e para configuração fixa de potência de transmissão de 60 dBm. As Figuras 42 e 43 apontam o atraso de rede e *jitter* enquanto que as Figuras 44 e 45 exibem a taxa de transferência (em bps) e taxa de pacotes perdidos, respectivamente.

Figura 42 – *Delay* encontrado para variação de número de aeronaves.



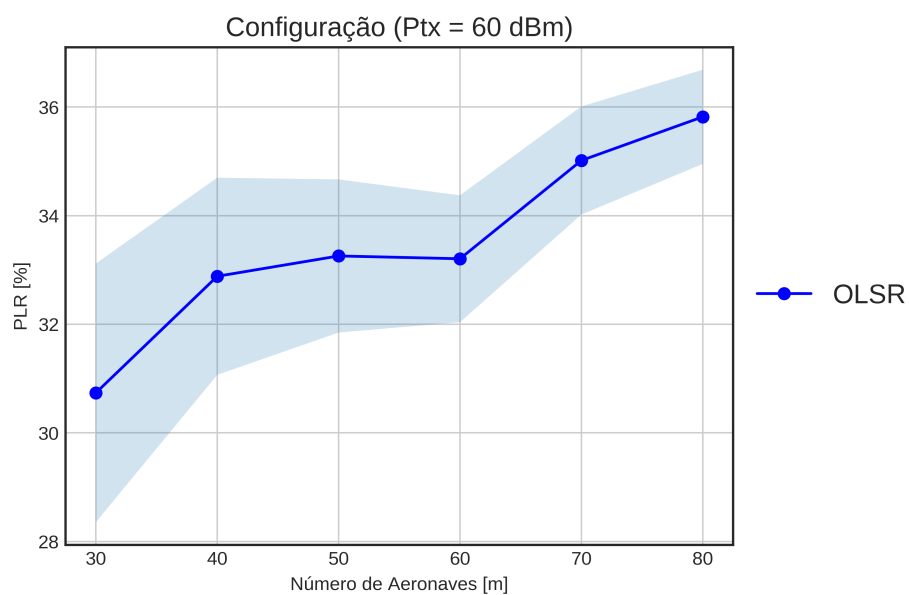
Fonte: Próprio Autor.

Figura 43 – *Jitter* encontrado para variação de número de aeronaves.



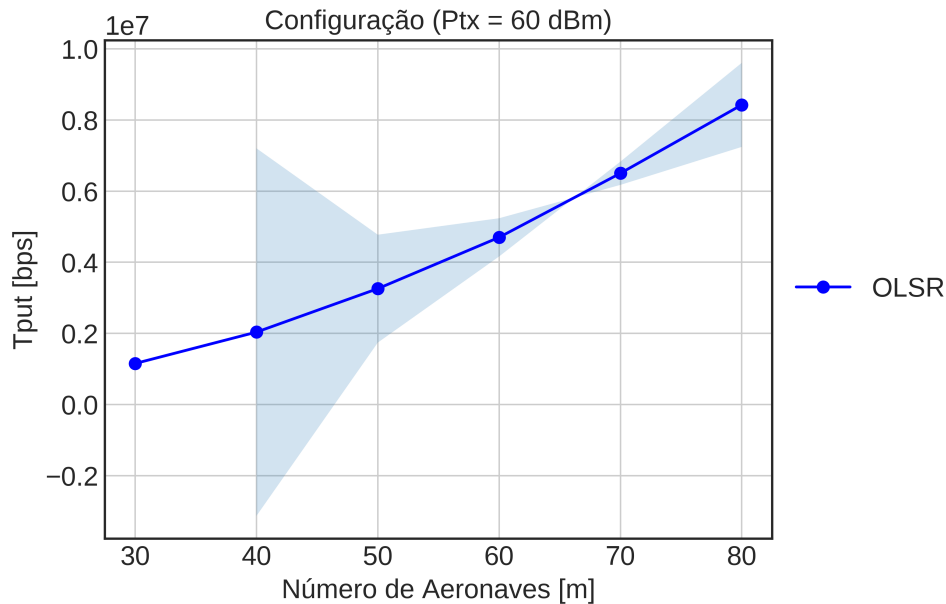
Fonte: Próprio Autor.

Figura 44 – Taxa de pacotes perdidos encontrada para variação de número de aeronaves.



Fonte: Próprio Autor.

Figura 45 – Taxa de transferência encontrada para variação de número de aeronaves.



Fonte: Próprio Autor.

Com valores ligeiramente distintos em relação à primeira configuração, os gráficos apontam para esta, valores de atraso menores para um número de aeronaves menores, bem como, valores de *jitter* menores para menor número de aeronaves. Isso se deve a não necessidade do pacote realizar saltos até chegar ao ponto de destino, o que difere-se para um cenário onde o pacote tem que atravessar por um número maior de aeronaves e desse modo, apresenta valores de atraso e *jitter* maiores. Diferentemente da configuração anterior, a taxa de pacotes perdidos mostrou um aumento proporcional ao crescente número de aeronaves – que pode ser explicado devido à maior chance da não efetividade da comunicação em um dado número de retransmissões para um dado número de aeronaves, por exemplo. A taxa de transferência apresenta valores maiores como o esperado para uma maior quantidade de aeronaves, porém, seu aumento não foi linear.

## 6.2 Análise de resultados para o protocolo OLSR - padrão 802.11ax

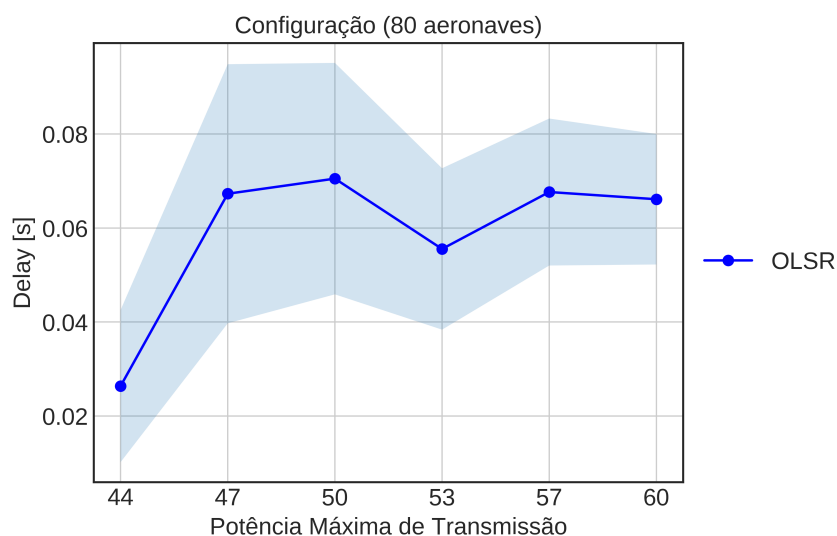
Realizou-se a análise das mesmas métricas anteriormente discutidas para o mesmo protocolo de roteamento, porém, nesta configuração é levado em consideração o padrão *Wi-Fi* 802.11ax. Afaqui, Garcia-Villegas e Lopez-Aguilera (2016) explicam em seu trabalho os principais pontos em que o padrão 802.11ax tende a aprimorar a rede de

comunicação:

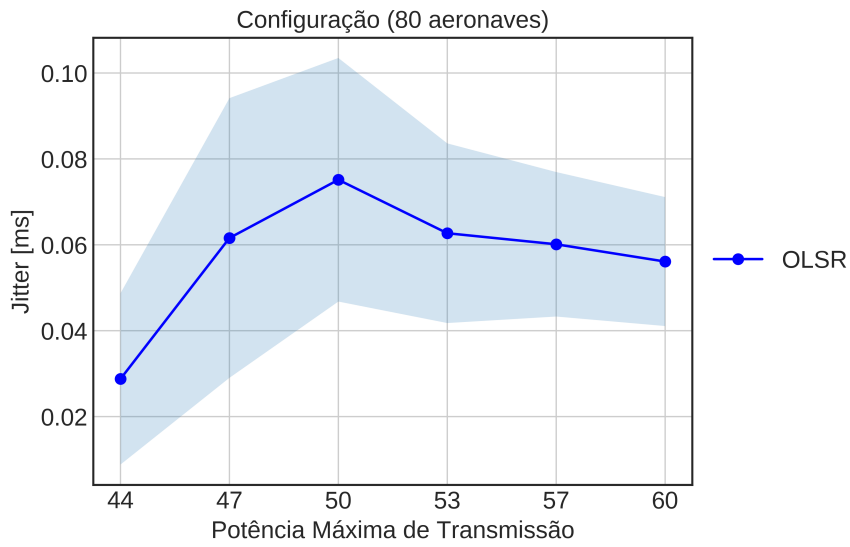
- Taxas de transferência (*throughput*) por usuário em redes densas: espera-se um aumento de *throughput* de pelo menos quatro vezes maior que o usual;
- Melhoramento da eficiência de energia: permite modos de economia simplificados e direcionados à cada dispositivo;
- Uso eficiente de recursos espectrais: implementação de métodos que garantam o uso eficiente do espectro de canal;
- Compatibilidade com versões anteriores: espera-se conformidade integral aos padrões anteriormente desenvolvidos, ou seja, qualquer padrão IEEE 802.11 será suportado.

Nesta configuração, para o padrão 802.11ax, foram detectados erros pontuais em algumas simulações fazendo com que não houvesse a geração de alguns resultados previstos no perfil anterior de estudo (padrão 802.11b). As Figuras 46 e 47 mostram os resultados obtidos de atraso (*delay*) e *jitter* com a variação de potência de transmissão e número de aeronaves fixada em 80. As Figuras 48 e 49 representam os resultados obtidos das métricas de taxa de pacotes perdidos e taxa de transferência, respectivamente. A padronização gráfica dos valores segue o mesmo do estudo anterior (padrão 802.11b) com 100 repetições onde há os pontos médios (linha contínua em azul) e os pontos máximos e mínimos delimitados pela lacuna azul.

Figura 46 – *Delay* encontrado para variação de potência de transmissão.

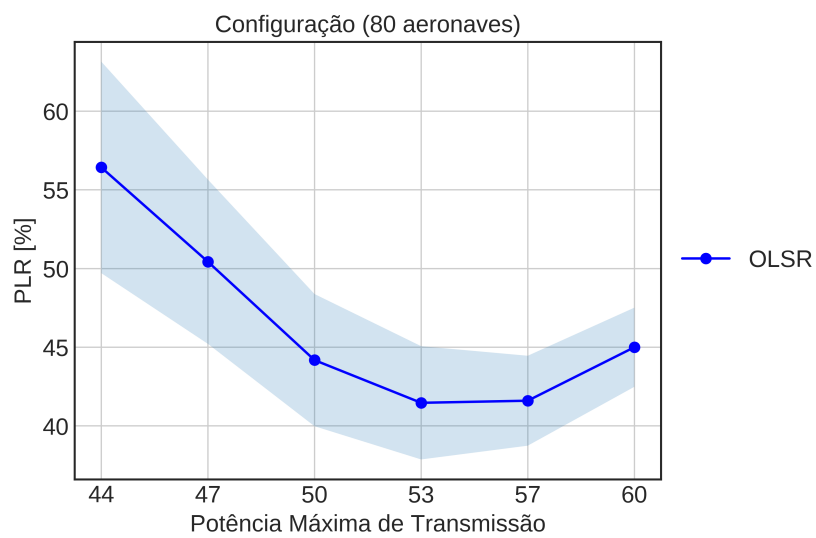


Fonte: Próprio Autor.

Figura 47 – *Jitter* encontrado para variação de potência de transmissão.

Fonte: Próprio Autor.

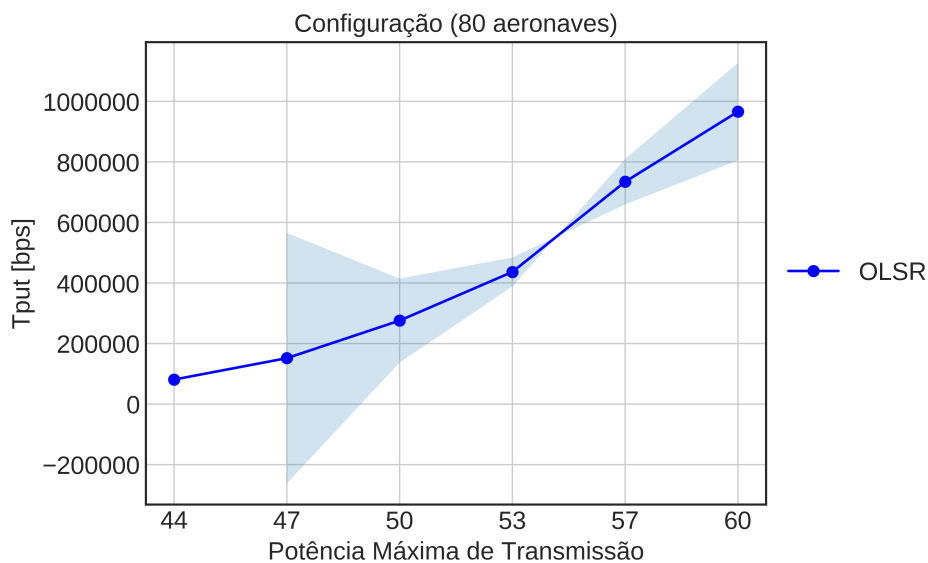
Figura 48 – Taxa de pacotes perdidos encontrada para variação de potência de transmissão.



Fonte: Próprio Autor.



Figura 49 – Taxa de transferência encontrada para variação de número de potência de transmissão.



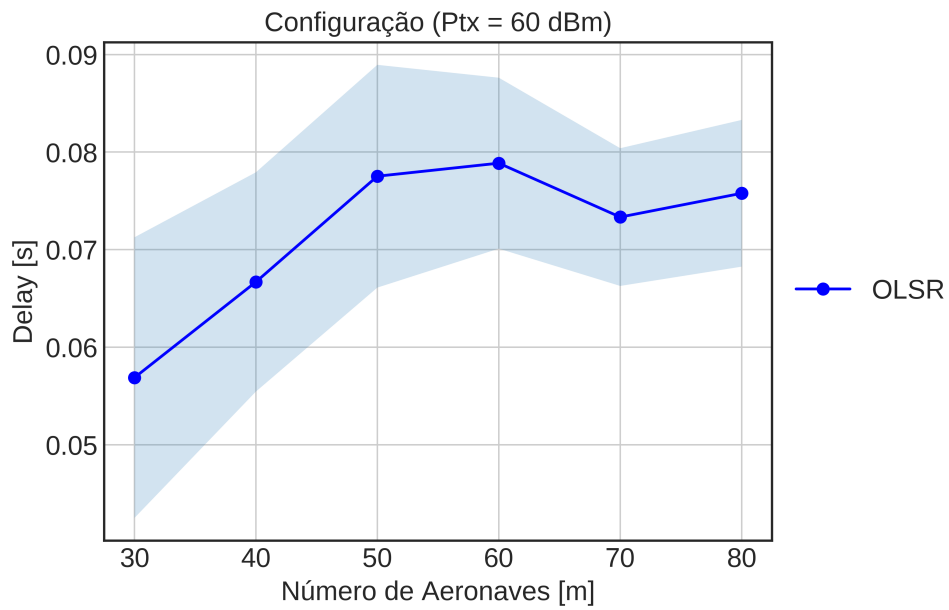
Fonte: Próprio Autor.

Os gráficos acima mostram valores de atraso com certa estabilidade em torno de 0.07 segundos enquanto que os valores de *jitter* decaem a medida que a potência de transmissão aumenta. Para essas métricas, o que ocorreu do mesmo modo para o padrão anterior, nota-se também que o intervalo de variação entre os valores máximos e mínimos (tanto para o *delay* quanto para o *jitter*) diminui, apontando mais uma vez nível de confiança maior conforme aumenta-se a taxa de variação da potência de transmissão. Isso indica um sistema de comunicação mais robusto à medida em que há um cenário com valores altos de potência.

Nos resultados exibidos das figuras observa-se primeiramente um comportamento decrescente nos valores de taxa de pacotes perdidos a medida que a variação de potência aumenta. Nesse caso, em uma comunicação mais potente há menor taxa de perda de pacotes e consequentemente, medidas de taxa de transferência de pacotes maiores para valores elevados de variação de potência – como se é esperado para essas condições. O gráfico com os valores de *throughput* apresenta aumento de forma não-linear e isso pode ser explicado à alta mobilidade do cenário, propagação e desvio de frequência modelados para a situação específica.

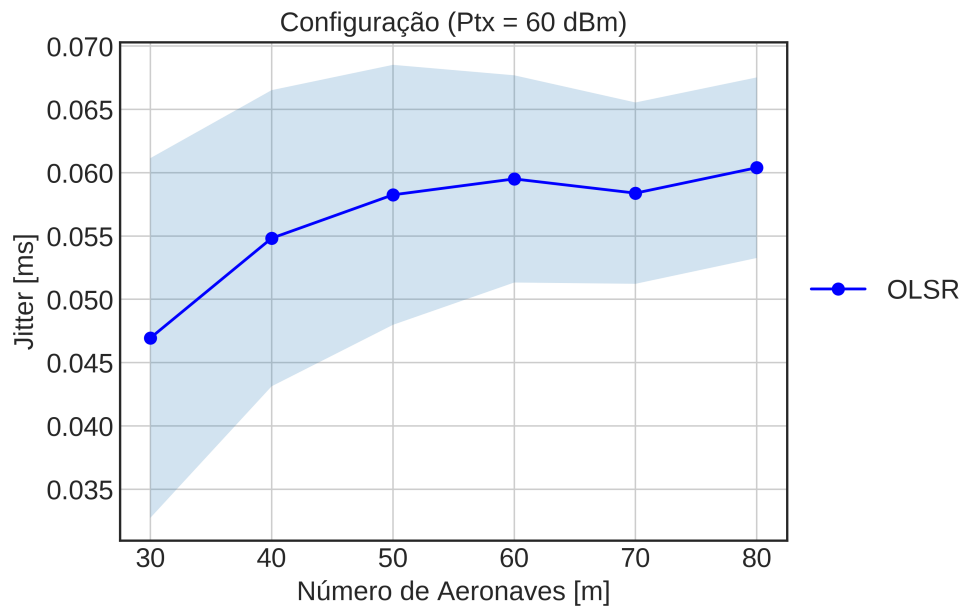
Nas Figuras 50, 51 e 52 são exibidos os resultados de *delay*, *jitter* e taxa de transferência com potência de transmissão fixada em 60 dBm e apresenta variação de quantidade de aeronaves no cenário proposto.

Figura 50 – Atraso encontrado para variação de quantidade de aeronaves.



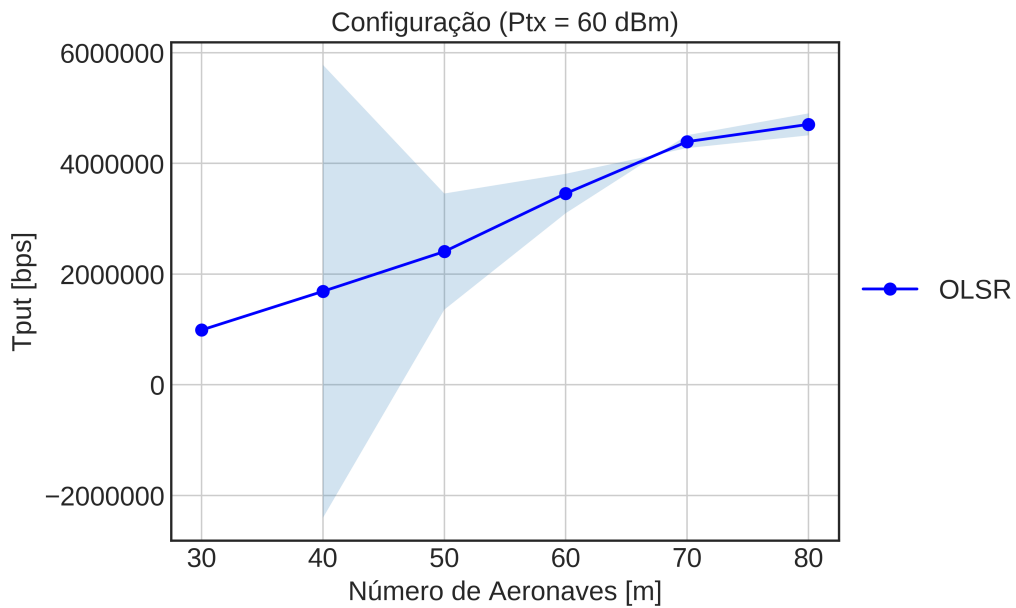
Fonte: Próprio Autor.

Figura 51 – Jitter encontrado para variação de quantidade de aeronaves.



Fonte: Próprio Autor.

Figura 52 – Taxa de transferência encontrada para variação de quantidade de aeronaves.



Fonte: Próprio autor.

Como mencionado anteriormente, pela ocorrência de erros em algumas simulações para esse padrão em específico, não foi possível gerar o resultado de taxa de pacotes perdidos para configuração variando o número de aeronaves. Entretanto, os valores de atraso mostram certa estabilidade para aproximadamente 0.08 segundos e *jitter* para valores de 0.060 ms. A taxa de transferência segue o mesmo padrão de comportamento crescente a medida que o número de aeronaves também aumenta.

## 7 CONSIDERAÇÕES FINAIS

Esse capítulo abordará as conclusões finais do trabalho assim como os objetivos alcançados e discussão dos resultados obtidos.

### 7.1 Conclusões Finais

Conforme apresentado no capítulo 5, fez-se possível implementar os cenários adequados de voo, fazer uma análise importante sobre a mobilidade das aeronaves e respectivamente determinar o modelo de mobilidade e modelo de propagação mais adequado aos cenários implementados. Foram levantados dados fidedignos de voo em tempo real através do *site FlightRadar24* e foram analisadas as rotas Rio de Janeiro (Aeroporto do Galeão) - São Paulo (Aeroporto de Guarulhos) e São Paulo (Aeroporto de Congonhas) - Rio de Janeiro (Aeroporto Santos Dumont) em dias diferentes.

Consequentemente, observa-se que dois dos modelos de mobilidade, previamente apresentados no capítulo anterior, mostram uma possível aplicabilidade para as redes AANET teoricamente, que são o modelo *Gauss-Markov Random Mobility Model* – ao qual a velocidade e direção do nó são baseados em memória – e *Path-Planned Mobility Model* – a direção e velocidade de cada nó é predeterminado pela trajetória do avião. Desta forma, o modelo de mobilidade escolhido e implementado no cenário foi o *Gauss-Markov Random Mobility Model* em três dimensões apresentando resultados satisfatórios e fiéis à realidade que o cenário propõe permitindo o dimensionamento de medidas reais do cenário aeronáutico considerando que Kumar e Verma (2019a) argumentam em seu trabalho que só é possível ter um formato realista das redes AANET quando se tem um modelo de mobilidade capaz de determinar velocidade, direção e posição através dessas mesmas métricas anteriores, ou seja, um modelo baseado em memória.

Os modelos de propagação escolhidos – modelo de *Jakes* e modelo *Three Log Distance* – apresentaram melhor desempenho na configuração da rede e simulações de acordo com o esperado, sem erros ou interferência nos resultados finais. Esses padrões foram modelados e tiveram seus cálculos efetuados de acordo com o cenário.

Obteve-se resultados de taxa de transferência (*throughput*), taxa de pacotes perdidos ou *Packet Loss Ratio* (PLR), atraso ou *delay* e *jitter* em simulações realizadas para os padrões 802.11b e 802.11ax.

Realizando análises comparativas para configuração fixada em 80 aeronaves e

variando valores de potência de transmissão, os valores de métricas encontradas de atraso e *jitter*, o padrão 802.11ax mostrou-se ligeiramente mais eficiente com valores de *delay* mínimo de até 0,023 segundos e *jitter* mínimo de 0,03 milisegundos. Já o melhor desempenho analisado para métricas de taxa de pacotes perdidos e taxa de transferência obteve-se no padrão 802.11b apresentando *PLR* mínimo de 30% e *throughput* máximo de 1,25 Mbps.

As análises para configuração de potência de transmissão padronizada em 60 dBm e apresentando variação no número de aeronaves, para métricas de atraso e *jitter*, mais uma vez o padrão 802.11ax mostrou desempenho melhor com atraso mínimo de 0,058 segundos e *jitter* mínimo de 0,047 milisegundos. Entretanto, o padrão 802.11b apresentou melhor desempenho referente à taxa de transferência máxima de 8,3 Mbps em relação ao padrão 802.11ax. A taxa de pacotes perdidos não pôde ter sua análise realizada – para esta configuração específica – devido à erros pontuais encontrados durante sua simulação.

Dessa forma, constatou-se que a criação de cenários aeronáuticos realistas não afetam o desempenho de uma comunicação *ad-hoc* designado para o protocolo de roteamento OLSR e com as parametrizações definidas para a criação do cenário específico para esse estudo.

Pela constatação de carência de material sobre o tópico estudado nesse trabalho e a existência de resultados preliminares de pesquisas na área de redes AANET, é de clara percepção que essa pesquisa contribuiu de forma importante na área de redes de computadores, mais especificamente, no que diz respeito à conectividade aeronáutica. Da mesma forma, essa pesquisa colabora no avanço de tecnologia aeronáutica dentro e fora das aeronaves, aos quais demandam estudos mais avançados sobre o tópico para que hajam significativas e benéficas mudanças na forma de voar – tanto para os passageiros quanto para os pilotos e profissionais do controle de tráfego aéreo.

## 7.2 Trabalhos futuros

Dos objetivos específicos anteriormente mencionados no capítulo 1, foi possível atingir:

- Estudo de protocolos de redes: foram estudados e elencados como protocolos de roteamento mais condizentes às redes AANET o que pode ser explicado no capítulo 3 onde os mesmos são detalhadamente discutidos;

- Pesquisa de trabalhos correlatos: no capítulo 4 foram elencados três trabalhos correlatos aos quais foram julgados os mais pertinentes à essa pesquisa bem como, serviram de alicerce em termos teóricos e práticos, consideráveis para a realização desse trabalho;
- Estudar simuladores para simulação de redes de computadores: como explicitado no capítulo 6, os programas de simulação de redes relacionados para esse trabalho foi ns-3 ao qual é um *software* livre e é comumente usado em pesquisas tendo sua eficácia e credibilidade já comprovadas;
- Definição de cenários adequados para as simulações: Os cenários implementados desse trabalho foram basicamente criados através de base de dados *FlightRadar24* que contém valores de voos reais, além das métricas de mobilidade como velocidade, altitude, posição;
- Definição de modelo e simulador a serem utilizados: após averiguar os cenários criados, a concepção da ideia do modelo de mobilidade foi prontamente realizada (o que pode ser visto de forma mais detalhada no capítulo 5) e a partir de então, o simulador de rede mais adequado foi definido.

Considera-se que todas as atividades previstas para esse trabalho foram realizadas de forma satisfatória e seguiram de maneira fiel o cronograma inicialmente idealizado. Os objetivos específicos não alcançados, que serão estudados e terão suas análises descritas futuramente, são:

- Estudo dos outros protocolos de roteamento: por apresentarem erros e discrepâncias nas simulações e nos resultados finais, não houve a possibilidade de análise dos protocolos de roteamento AODV e DSDV, o que será estudado posteriormente e seus erros devidamente corrigidos.
- Aprimoramento do protocolo 802.11ax para o cenário: as simulações para esse padrão também apresentaram erros pontuais o que não permitiu a completa análise de resultados de desempenho, validação e teste do padrão 802.11ax para a rede proposta. Dessa forma, este padrão será futuramente analisado, corrigido e terá suas devidas alterações realizadas a fim de se obter resultados favoráveis ao estudo.
- Investigação de outros padrões de comunicação: por apresentar mobilidade muito alta e dinâmica devido ao cenário criado e específico para as circunstâncias do trabalho, faz-se necessário estudar outros padrões de comunicação que se adéquem melhor e apresentem resultados mais coerentes à rede proposta – como por exemplo

as redes 5G.

## REFERÊNCIAS

- AFAQUI, M. S.; GARCIA-VILLEGAS, E.; LOPEZ-AGUILERA, E. Ieee 802.11 ax: Challenges and requirements for future high efficiency wifi. **IEEE Wireless Communications**, IEEE, v. 24, n. 3, p. 130–137, 2016.
- ALPERN, N. **Eleventh Hour Network+: Exam N10-004 Study Guide**. [S.l.]: Syngress Publishing, 2009. ISBN 1597494283.
- BARAN, I. R. et al. Explorando a coerência temporal e a comunicação oportunística em sistemas de comunicação móvel multiusuário. Florianópolis, SC, 2006.
- BEKMEZCI, I.; SAHINGOZ, O. K.; TEMEL, Ş. Flying ad-hoc networks (fanets): A survey. **Ad Hoc Networks**, Elsevier, v. 11, n. 3, p. 1254–1270, 2013.
- BENIN, J.; NOWATKOWSKI, M.; OWEN, H. Vehicular network simulation propagation loss model parameter standardization in ns-3 and beyond. In: IEEE. **2012 Proceedings of IEEE Southeastcon**. [S.l.], 2012. p. 1–5.
- BROYLES, D. et al. Design and analysis of a 3–d gauss-markov mobility model for highly-dynamic airborne networks. In: **Proceedings of the international telemetering conference (ITC), San Diego, CA**. [S.l.: s.n.], 2010.
- DEMO, P. **Metodologia do conhecimento científico**. [S.l.]: Atlas, 2000. 20 p.
- FLIGHTAWARE. **FlightAware - Rastreador de voos / status de voos**. 2020. Acesso em: 16 nov. 2020. Disponível em: <https://pt.flightaware.com/>.
- FLIGHTRADAR24. **FlightRadar24 - Live Flight Tracker - Real-Time Flight Tracker Map**. 2021. Acesso em: 25 jul. 2021. Disponível em: <https://www.flightradar24.com>.
- FONSECA, P. A. S. C. d. et al. Considerações sobre lans e wlans: uma proposta para um centro cultural. Florianópolis, SC, 2001.
- FREITAS, C. de. **Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico - 2ª Edição**. Editora Feevale, 2013. ISBN 9788577171583. Disponível em: <https://books.google.com.br/books?id=zUDsAQAAQBAJ>.
- GAST, M. **802.11 n: a survival guide**. [S.l.]: "O'Reilly Media, Inc.", 2012.
- GAST, M. **802.11ac: A Survival Guide**. O'Reilly Media, 2013. ISBN 1449343147,9781449343149. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=8a4923625133be59dc43d5555a1976f5>.
- GUIMARÃES, D. A. **INTRODUÇÃO ÀS COMUNICAÇÕES MÓVEIS**. 2021. Acesso em: 20 set. 2021. Disponível em: <http://www.wirelessbrasil.org/wirelessbr/colaboradores/dayani/intro02.html>.
- ILYAS, M.; AHSON, S. **Handbook of Wireless Local Area Networks: Applications, Technology, Security, and Standards**. CRC Press, 2005. (Internet and Communications). ISBN 9780203492512. Disponível em: <https://books.google.com.br/books?id=s3AEqZDniaYC>.



- KUMAR, P.; VERMA, S. Analysis of adaptability of mobility models for aanets. **International Journal Of Research in Eletronics and Computer Engineering**, 2019.
- KUMAR, P.; VERMA, S. Implementation of modified olsr protocol in aanets for udp and tcp environment. **Journal of King Saud University-Computer and Information Sciences**, Elsevier, 2019.
- KUMAR, P.; VERMA, S. Research challenges in airborne ad-hoc networks (aanets). In: SPRINGER. **Proceedings of 2nd International Conference on Communication, Computing and Networking**. [S.l.], 2019. p. 261–270.
- KUMAR, V. et al. Applications of vanets: present & future. **Communications and Network**, Scientific Research Publishing, v. 5, n. 01, p. 12, 2013.
- KUMAR, V.; RANA, A.; KUMAR, S. Aircraft ad-hoc network (aanet). **International Journal of Advanced Research in Computer and Communication Engineering**, v. 3, n. 5, 2014.
- KUROSE, J. F. **RKW (2013). Redes de Computadores e a Internet: Uma abordagem top-down**. [S.l.]: Pearson Education do Brasil Ltda, 2013.
- LOO, J.; MAURI, J.; ORTIZ, J. **Mobile Ad Hoc Networks: Current Status and Future Trends**. Taylor & Francis, 2011. ISBN 9781439856505. Disponível em: <https://books.google.com.br/books?id=oVSqcQAACAAJ>.
- MAYGUA-MARCILLO, L.; URQUIZA-AGUIAR, L.; PAREDES-PAREDES, M. Wireless channel 802.11 in ns-3. Preprints, 2018.
- MELO, P. **Padrão IEEE 802.15.4 – A base para as especificações Zigbee, WirelessHart e MiWi**. 2017. Acesso em: 05 nov. 2020. Disponível em: <https://www.embarcados.com.br/padrao-ieee-802-15-4/>.
- MENON, P. K.; SWERIDUK, G. D.; SRIDHAR, B. Optimal strategies for free-flight air traffic conflict resolution. **Journal of Guidance, Control, and Dynamics**, v. 22, n. 2, p. 202–211, 1999.
- MILLER, D. C. L. C. **802.11ax for Dummies: Aerohive Special Edition**. Special. John Wiley & Sons, 2018. (For Dummies). ISBN 9781119528029, 9781119528005. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=dd92536a6ea792a64ae48389dff5e74>.
- MORAES, A. F. d. **Redes sem fio: instalação, configuração e segurança: fundamentos**. [S.l.]: Érica, 2010.
- NAGLE, A.; BAYROSS, R.; GADE, R. Routing algorithm for aeronautical ad hoc networks. 2014.
- NAYYAR, A. Flying adhoc network (fanets): simulation based performance comparison of routing protocols: Aodv, dsdv, dsr, olsr, aomdv and hwmp. In: IEEE. **2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)**. [S.l.], 2018. p. 1–9.
- PERAHIA, E.; STACEY, R. **Next Generation Wireless LANs: 802.11n and 802.11ac**. 2nd. ed. USA: Cambridge University Press, 2013. ISBN 1107016762.

PERES, A.; WEBER, R. F. Considerações sobre segurança em redes sem fio. In: **III WORKSHOP EM SEGURANÇA DE SISTEMAS COMPUTACIONAIS**. [S.l.: s.n.], 2003.

PERKINS, C. E. Ad hoc networking: an introduction. **Ad hoc networking**, Addison-Wesley Longman Publishing, v. 40, p. 20–22, 2001.

PIMENTA, T. M. d. C. **Simulação de protocolos de encaminhamento para redes AdHoc usando o NS-3**. Tese (Doutorado), 2013.

PUGA, H. A. R. **Evaluación de desempeño de protocolos de enrutamiento AODV y OLSR en redes Ad-Hoc aeronáuticas, para tráfico aéreo dentro del área de control Quito**. Dissertação (Mestrado) — Quito, 2018., 2018.

RAPPAPORT, T. S. **Comunicações sem fio: princípios e práticas**. [S.l.]: Pearson Prentice Hall, 2009.

RASHVAND, H. F.; CHAO, H.-C. **Dynamic Ad Hoc Networks**. [S.l.]: Institution of Engineering and Technology, 2013. ISBN 1849196478.

RAYNER, J. M. V. A vortex theory of animal flight. part 2. the forward flight of birds. **Journal of Fluid Mechanics**, Cambridge University Press, v. 91, n. 4, p. 731–763, 1979.

SAMARA, C.; KARAPISTOLI, E.; ECONOMIDES, A. A. Performance comparison of manet routing protocols based on real-life scenarios. In: IEEE. **2012 IV International Congress on Ultra Modern Telecommunications and Control Systems**. [S.l.], 2012. p. 870–877.

SANTOS, O. L. dos; PADOLFI, A. S.; RAMALDES, G. P. Análise de índice de vegetação através de imagens obtidas por vant. **Revista Científica FAESA**, v. 14, n. 1, p. 145–165, 2018.

SANTOS, W. Uso de simuladores como ferramenta no ensino-aprendizagem de redes de computadores. **Projetos e Dissertações em Sistemas de Informação e Gestão do Conhecimento**, v. 4, n. 2, 2015.

SARKAR T.G. BASAVARAJU, C. P. S. K. **Ad Hoc Mobile Wireless Networks: Principles, Protocols, and Applications, Second Edition**. Second. CRC Press, 2013. ISBN 1466514469,9781466514461. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=25bab026bfd5b1621b81d13960e5734a>.

TORRES, P. **Administração de Sistemas Linux: Redes e Segurança**. [S.l.]: Escola Superior de Redes, 2013.

ZHANG, J. et al. Aeronautical adhoc networking for the internet-above-the-clouds. **Proceedings of the IEEE**, IEEE, v. 107, n. 5, p. 868–911, 2019.

## APÊNDICE A – CÓDIGO FONTE DA APLICAÇÃO

```

1
2  /* -*- Mode: C++; c-file-style: "gnu"; indent-tabs-mode:nil; -*- */
3  /*
4  * Copyright (c) 2021 UFRN/UNIPAMPA
5  *
6  * This program is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU General Public License version 2 as
8  * published by the Free Software Foundation;
9  *
10 * This program is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU General Public License for more details.
14 *
15 * You should have received a copy of the GNU General Public License
16 * along with this program; if not, write to the Free Software
17 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
18 *
19 * Authors: Maiara (UNIPAMPA), Prof. Gerson (UNIPAMPA), Tarciana Guerra (UFRN), Vicente Sousa (UFRN)
20 *
21 */
22
23 /* to run:
24 * NS_GLOBAL_VALUE="RngRun=1627332758828894" ./waf --run 'fanet-routing-compare --inWifis=3000 --protocol=1
25
26 --iTransportMode=0 --simTime=10 --stOutputDir='/home/vicente/2021_1/UFRNUNIPAMPA/ns-allinone-3.34/ns-3.34
27
28 /results_Campaign1_inWifis''
29 */
30
31 #include <sys/socket.h>
32 #include <errno.h>
33 #include <fstream>
34 #include <string>
35 #include <iostream>
36 #include <iomanip>
37 #include <string>
38 #include <algorithm>
39 #include <ctime>
40 #include <time.h>
41 #include <math.h>
42 #include <fstream>
43 #include <math.h>
44 #include <stack>
45 #include "ns3/core-module.h"
46 #include "ns3/network-module.h"
47 #include "ns3/internet-module.h"
48 #include "ns3/point-to-point-module.h"
49 #include "ns3/applications-module.h"
50 #include "ns3/aodv-helper.h"
51 #include "ns3/olsr-module.h"
52 #include "ns3/dsdv-module.h"
53 #include "ns3/dsr-module.h"
54 #include "ns3/ipv4-global-routing-helper.h"
55 #include "ns3/mobility-module.h"
56 #include "ns3/wifi-module.h"
57 #include "ns3/yans-wifi-helper.h"
58 #include "ns3/fd-net-device-module.h"
59 #include "ns3/netanim-module.h"
60 #include "ns3/flow-monitor.h"
61 #include "ns3/flow-probe.h"
62 #include "ns3/ipv4-flow-classifier.h"
63 #include "ns3/ipv4-l3-protocol.h"
64 #include "ns3/flow-monitor-module.h"
65 #include "ns3/energy-module.h"
66 #include "ns3/basic-energy-source.h"
67 #include <ns3/config-store-module.h>

```

```

68 #include "ns3/random-variable-stream.h"
69 #include "ns3/propagation-module.h"
70 #include "ns3/phy-entity.h"
71 #include "ns3/assert.h"
72 #include <ns3/wifi-mode.h>
73 #include "ns3/minstrel-wifi-manager.h"
74 #include "ns3/minstrel-ht-wifi-manager.h"
75 using namespace ns3;
76
77
78 // Test if the file is empty
79 bool is_empty(std::ifstream& pFile)
80 {
81     return pFile.peek() == std::ifstream::traits_type::eof();
82 }
83
84 // To be used in tic toc time counter
85 clock_t startTimer;
86 time_t beginTimer;
87 //
88 // Implementation of tic, i.e., start time counter
89 void
90 tic()
91 {
92     beginTimer = time(&beginTimer);
93     struct tm * timeinfo;
94     timeinfo = localtime(&beginTimer);
95     std::cout << "simulation start at: " << asctime(timeinfo) << std::endl;
96 }
97 // implementation of toc, i.e., stop time counter
98 double
99 toc()
100 {
101     time_t finishTimer = time(&finishTimer);
102     double simTime = difftime(finishTimer, beginTimer) / 60.0;
103     struct tm * timeinfo;
104     timeinfo = localtime(&finishTimer);
105     std::cout << "simulation finished at: " << asctime(timeinfo) << std::endl;
106     //
107     std::cout << "Time elapsed: " << simTime << " minutes" << std::endl;
108     //
109     return simTime;
110 }
111
112 int main (int argc, char *argv[])
113 {
114     tic();
115     int iSeed = 1;
116     uint32_t iRunSeed = 1;
117     int iNWifis = 80;
118     double dTxp = 60;
119     double velMin = 180.0;
120     double velMax = 200.0;
121     std::string stRate ("2048bps");
122     std::string standard = "802.11b";
123     std::string stPacketSize ("1024");
124     uint32_t rtsThreshold = 65535;
125     std::string rateManager = "ns3::MinstrelHtWifiManager";
126     //std::string stOFDM ("ErpOfdmRate6Mbps");
127     double dSimTime = 50;
128     double dStartTrafficTime = 30.0;
129     double dTime2StartStatistics = dStartTrafficTime + 1.0; // Define the warmup time to collect
130
131     statistics (must be lower than dSimTime)
132     uint32_t iProtocol = 1;
133     std::string stProtocolName = "protocol";
134     int iTransportMode = 0;
135     std::string stTransportModeName = "UDP";
136     std::string stOutputDir = ".";
137     int iJobID = 0;

```

```

138
139 CommandLine cmd;
140 cmd.AddValue ("iNWifis", "Number of end devices to include in the simulation", iNWifis);
141 cmd.AddValue ("iJobID", "JobID", iJobID);
142 cmd.AddValue ("velMin", "Minimum mobility speed", velMin);
143 cmd.AddValue ("velMax", "Maximum mobility speed", velMax);
144 cmd.AddValue ("dTxp", "Transmission Power", dTxp);
145 cmd.AddValue ("iProtocol", "1=OLSR;2=AODV;3=DSDV;4=DSR", iProtocol);
146 cmd.AddValue ("iTransportMode", "1:true, 0:false, default mode UDP", iTransportMode);
147 cmd.AddValue ("stOutputDir", "Output directory", stOutputDir);
148 cmd.AddValue ("dSimTime", "The time for which to simulate", dSimTime);
149 cmd.AddValue ("standard", "Wifi Phy Standard: 802.11a, 802.11b, 802.11g, 802.11n-2.4GHz, 802.11n-5GHz,
150 802.11ac, 802.11ax_2_4GHZ", standard);
151
152 cmd.Parse (argc, argv);
153
154 dSimTime = dSimTime + dStartTrafficTime;
155
156 // Setting default values
157 Config::SetDefault ("ns3::OnOffApplication::PacketSize",StringValue (stPacketSize));
158 Config::SetDefault ("ns3::OnOffApplication::DataRate", StringValue (stRate));
159 Config::SetDefault ("ns3::JakesProcess::DopplerFrequencyHz", DoubleValue (4000));
160 Config::SetDefault ("ns3::JakesProcess::NumberOfOscillators", UIntegerValue (4));
161 //Non-unicast mode
162 //Config::SetDefault ("ns3::WifiRemoteStationManager::NonUnicastMode",StringValue (stPhyMode));
163
164 ConfigStore inputConfig;
165 inputConfig.ConfigureDefaults ();
166
167 // #####
168 // Scenario setup
169 // #####
170 NodeContainer adhocNodes;
171 adhocNodes.Create (iNWifis);
172
173 // #####
174 // Mobility Model
175 // #####
176 int64_t streamIndex = 0; // used to get consistent mobility across scenarios
177 // TODO: mesma padrao mobilidade em todos as simula es. Colocar aqui a semente aleatoria
178 MobilityHelper mobility;
179 ObjectFactory pos;
180 pos.SetTypeId ("ns3::RandomBoxPositionAllocator");
181 pos.Set ("X", StringValue ("ns3::UniformRandomVariable [Min=0.0|Max=25000]"));
182 pos.Set ("Y", StringValue ("ns3::UniformRandomVariable [Min=0.0|Max=350000]"));
183 pos.Set ("Z", StringValue ("ns3::UniformRandomVariable [Min=5000|Max=8000]"));
184 Ptr<PositionAllocator> taPositionAlloc = pos.Create ()->GetObject<PositionAllocator> ();
185 streamIndex += taPositionAlloc->AssignStreams (streamIndex);
186 mobility.SetMobilityModel ("ns3::GaussMarkovMobilityModel",
187     "Bounds", BoxValue (Box (0, 25000, 0, 350000, 5000, 8000)),
188     "TimeStep", TimeValue (Seconds (1)),
189     "Alpha", DoubleValue (0.85),
190     "MeanVelocity", StringValue ("ns3::UniformRandomVariable [Min=180|Max=200]"),
191     "MeanDirection", StringValue ("ns3::UniformRandomVariable [Min=0|Max=6.283185307]"),
192     "MeanPitch", StringValue ("ns3::UniformRandomVariable [Min=0.05|Max=0.05]"),
193     "NormalVelocity", StringValue ("ns3::NormalRandomVariable [Mean=0.0|Variance=0.0|Bound=0.0]"),
194     "NormalDirection", StringValue ("ns3::NormalRandomVariable [Mean=0.0|Variance=0.2|Bound=0.4]"),
195     "NormalPitch", StringValue ("ns3::NormalRandomVariable [Mean=0.0|Variance=0.02|Bound=0.04]"));
196
197 mobility.SetPositionAllocator (taPositionAlloc);
198 mobility.Install (adhocNodes);
199 streamIndex += mobility.AssignStreams (adhocNodes, streamIndex);
200 NS_UNUSED (streamIndex); // From this point, streamIndex is unused
201
202 // #####
203 // PHY Layer setup
204 // #####
205 // Set Non-unicastMode rate to Unicast mode
206 // Setting up Wi-Fi phy and channel using helpers
207 // TODO: colocar isso como parametro

```

```

208 WifiHelper wifi;
209 std::string stPhyMode;
210 //std::string stPhyMode ("DsssRate1Mbps");
211 //wifi.SetStandard (WIFI_STANDARD_80211n_2_4GHZ);
212 //Configuration based on wifi-rate-adaptation-distance .cc
213 if (standard == "802.11a" || standard == "802.11b" || standard == "802.11g")
214 {
215     if (standard == "802.11a")
216     {
217         wifi.SetStandard (WIFI_STANDARD_80211a);
218         stPhyMode = "OfdmRate12Mbps";
219     }
220     else if (standard == "802.11b")
221     {
222         wifi.SetStandard (WIFI_STANDARD_80211b);
223         stPhyMode = "DsssRate1Mbps";
224     }
225     else if (standard == "802.11g")
226     {
227         wifi.SetStandard (WIFI_STANDARD_80211g);
228     }
229     rateManager = "ns3::MinstrelWifiManager";
230     //rateManager = "ns3::MinstrelHtWifiManager";
231 }
232 else if (standard == "802.11n-2.4GHz" || standard == "802.11n-5GHz")
233 {
234     if (standard == "802.11n-2.4GHz")
235     {
236         wifi.SetStandard (WIFI_STANDARD_80211n_2_4GHZ);
237     }
238     else if (standard == "802.11n-5GHz")
239     {
240         wifi.SetStandard (WIFI_STANDARD_80211n_5GHZ);
241     }
242     rateManager = "ns3::MinstrelHtWifiManager";
243     stPhyMode = "HtMcs0";
244 }
245 else if (standard == "802.11ac")
246 {
247     wifi.SetStandard (WIFI_STANDARD_80211ac);
248     wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager",
249         "DataMode", StringValue ("VhtMcs9"),
250         "ControlMode", StringValue ("VhtMcs0"));
251
252     rateManager = "ns3::MinstrelHtWifiManager";
253 }
254 else if (standard == "80211ax_2_4GHZ")
255 {
256     wifi.SetStandard (WIFI_STANDARD_80211ax_2_4GHZ);
257     stPhyMode = "HeMcs0";
258
259     rateManager = "ns3::MinstrelHtWifiManager";
260 }
261 // Enable rate control
262 wifi.SetRemoteStationManager (rateManager, "RtsCtsThreshold", UintegerValue (rtsThreshold));
263 // Disable rate control
264 //wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager", "DataMode",StringValue //(stPhyMode), "ControlMode",
265 //StringValue (stPhyMode));
266
267
268
269 // Channel Setup
270 YansWifiPhyHelper wifiPhy;
271 YansWifiChannelHelper wifiChannel;
272 wifiChannel.SetPropagationDelay ("ns3::ConstantSpeedPropagationDelayModel");
273
274 wifiChannel.AddPropagationLoss ("ns3::JakesPropagationLossModel");
275
276 wifiChannel.AddPropagationLoss ("ns3::ThreeLogDistancePropagationLossModel");
277 wifiPhy.SetChannel (wifiChannel.Create ());

```

```

278
279 // #####
280 // MAC Layer setup
281 // #####
282 WifiMacHelper wifiMac;
283 // Set Transmission power
284 // TODO: verificar se em dBm ou em Watts
285 wifiPhy.Set ("TxPowerStart", DoubleValue (dTxp));
286 wifiPhy.Set ("TxPowerEnd", DoubleValue (dTxp));
287 // Set ad-hoc mode
288 wifiMac.SetType ("ns3::AdhocWifiMac");
289 NetDeviceContainer adhocDevices = wifi.Install (wifiPhy, wifiMac, adhocNodes);
290
291 // #####
292 // Internet Protocol stack setup
293 // #####
294 AodvHelper aodv;
295 OlsrHelper olsr;
296 DsdvHelper dsdv;
297 DsrHelper dsr;
298 DsrMainHelper dsrMain;
299 Ipv4ListRoutingHelper list;
300 InternetStackHelper internet;
301
302 switch (iProtocol)
303 {
304 case 1:
305     list.Add (olsr, 100);
306     stProtocolName = "OLSR";
307     break;
308 case 2:
309     list.Add (aodv, 100);
310     stProtocolName = "AODV";
311     break;
312 case 3:
313     list.Add (dsdv, 100);
314     stProtocolName = "DSDV";
315     break;
316 case 4:
317     stProtocolName = "DSR";
318     break;
319 default:
320     NS_FATAL_ERROR ("No such protocol:" << iProtocol);
321 }
322
323 if (iProtocol < 4)
324 {
325     internet.SetRoutingHelper (list);
326     internet.Install (adhocNodes);
327 }
328 else if (iProtocol == 4)
329 {
330     internet.Install (adhocNodes);
331     dsrMain.Install (dsr, adhocNodes);
332 }
333
334 // IP Assignment
335 Ipv4AddressHelper addressAdhoc;
336 addressAdhoc.SetBase ("10.1.1.0", "255.255.255.0");
337 Ipv4InterfaceContainer adhocInterfaces;
338 adhocInterfaces = addressAdhoc.Assign (adhocDevices);
339
340 // #####
341 // Traffic Model and internet Protocol stack setup
342
343 //(based on https://www.nsnam.org/doxygen/ed2fd-onoff_8cc_source.html)
344 // #####
345 // TODO: por algum motivo o TCP n o est gerando mtricas //no flowmonitor. Talvez seja alguma configura o
346
347 //faltando. UDP est ok.

```

```

348     std::string factory;
349     if ( iTransportMode == 1 )
350     {
351         factory = "ns3::TcpSocketFactory";
352         stTransportModeName = "TCP";
353     }
354     else
355     {
356         factory = "ns3::UdpSocketFactory";
357         stTransportModeName = "UDP";
358     }
359     uint16_t sinkPort = 8000;
360
361
362     // Create a traffic source for each node
363     for (int i = 0; i < iNWifis; i++){
364         for (int j = 0; j < iNWifis; j++){
365             if(i!=j){
366                 // Creating the application container
367                 ApplicationContainer applications;
368                 // Retrieving the server and client nodes
369                 Ptr<Node> clientNode=adhocNodes .Get(i);
370                 Ptr<Node> serverNode=adhocNodes .Get(j);
371                 // Initializing the OnOffHelper that will help us to set up the application
372                 OnOffHelper onoff (factory , InetSocketAddress ( adhocInterfaces .GetAddress ( i), sinkPort));
373                 // Setting the parameters of the application
374                 onoff.SetAttribute ("OnTime", StringValue ("ns3::ConstantRandomVariable [Constant=1]"));
375                 onoff.SetAttribute ("OffTime", StringValue ("ns3::ConstantRandomVariable [Constant=0]"));
376                 // Installing the application on the server and adding it to the application container
377                 applications .Add (onoff. Install (serverNode));
378                 // Creating a PacketSinkHelper, class that simulates the reception of the generated traffic
379                 PacketSinkHelper sinkHelper (factory , InetSocketAddress ( adhocInterfaces .GetAddress ( j), sinkPort));
380                 // Installing the packet sink the client and adding it to the application container
381                 applications .Add (sinkHelper. Install (clientNode));
382                 // Defining the applications start and stop time
383                 Ptr<UniformRandomVariable> var = CreateObject<UniformRandomVariable> ();
384                 applications .Start (Seconds (var->GetValue (0.05,0.5)));
385                 applications .Stop (Seconds (dSimTime));
386                 // Updating the sinkPort
387                 sinkPort++;
388
389             }
390         }
391     }
392
393
394     // #####
395     // Energy Model according to examples/energy/energy-model-example.cc
396     // #####
397     /*** automatic installation via Helpers */
398     // for parameters see: http://www.ee.washington.edu/research/nsl/papers/SIMUTools-11.pdf
399     // energy source
400     BasicEnergySourceHelper basicSourceHelper;
401     // Configure energy source (default values in //basic-energy-source.cc). Set a higher value to guarantee
402
403     //STA long lifetime
404     basicSourceHelper.Set("BasicEnergySourceInitialEnergyJ", DoubleValue( 100 * dSimTime ) );
405     // install source
406     EnergySourceContainer sources = basicSourceHelper. Install (adhocNodes);
407     // Device energy model
408     // According to "ns3/wifi-radio-energy-model.h", default //values for power consumption are based on CC2420
409
410     //radio chip, with
411     // supply voltage as 2.5V and currents as 17.4 mA (TX), 18.8 mA (RX), 20 uA
412     // (sleep) and 426 uA (idle).
413     WifiRadioEnergyModelHelper radioEnergyHelper;
414     // install device model
415     DeviceEnergyModelContainer deviceModels = radioEnergyHelper. Install (adhocDevices , sources);
416
417     // #####

```



```

418 // Output Setup
419 // #####
420 // Simulation Output file: t-put evaluation according to wifi-hidden-terminal.cc
421 AsciiTraceHelper asciiTraceHelper;
422 // File to store performance results
423 std::string chFileResults = stOutputDir+"/"+Results_"+std::to_string(iJobID)+".txt";
424 //Ptr<OutputStreamWrapper> fileResults = asciiTraceHelper.CreateFileStream( chFileResults );
425 std::ifstream file(chFileResults.c_str());
426 bool firstTime = is_empty( file );
427 file.close();
428 std::ofstream myfile;
429 myfile.open(chFileResults.c_str(), std::ofstream::app);
430 if ( firstTime )
431 {
432 // file is empty: write down the header
433 myfile << "Flow_ID, Lost_Packets, Tx_Packets, Tx_Bytes, TxOffered(Mbps), Rx_Packets, Rx_Bytes, T_put(Mbps),
434
435 Mean_Delay_Rx_Packets, Mean_Jitter, Packet_Loss_Ratio, AvgPowerConsumption,
436
437 iNWifis, dTxp, iProtocol, dSimTime" << std::endl;
438 }
439
440
441 // File to store energy consumption
442 // TODO: implement the average power consumption and store on result file
443 //std::string chFileEnergy = stOutputDir+"/"+Energy_"+std::to_string(iJobID)+".txt";
444 //Ptr<OutputStreamWrapper> fileEnergy = asciiTraceHelper.CreateFileStream( chFileEnergy );
445 // File to store simulations main parameters and real simulation duration
446 //std::string chFileParam = stOutputDir+"/"+Params_"+std::to_string(iJobID)+".txt";
447 //Ptr<OutputStreamWrapper> streamSimParam = asciiTraceHelper.CreateFileStream(chFileParam);
448
449
450
451
452 //Getting seed and runSeed for checking and displaying purposes
453 iSeed = RngSeedManager::GetSeed();
454 iRunSeed = RngSeedManager::GetRun();
455 // Display current simulation information
456 std::cout <<
457 "Running with " << standard <<
458 ", iNWifis = " << iNWifis <<
459 ", txp = " << dTxp << " dBm" <<
460 ", velMin = " << velMin << " m/s" <<
461 ", velMax = " << velMax << " m/s" <<
462 ", simulation Time = " << dSimTime << " s" <<
463 ", Routing Protocol = " << stProtocolName <<
464 ", Traffic = " << stTransportModeName <<
465 ", iSeed = " << iSeed <<
466 ", run stream = " << iRunSeed <<
467 ", outputDir = " << stOutputDir << std::endl;
468
469 // Network Animation using NetAnim.
470 //AnimationInterface anim("Fanet3D"+std::to_string(iJobID)+".xml");
471 // Mobility trace
472 // AsciiTraceHelper ascii;
473 // MobilityHelper::EnableAsciiAll (ascii.CreateFileStream( tr_name + ".mob"));
474 //
475 // Flowmonitor configuration
476 // Simulation Output: create a instance and install FlowMonitor
477 FlowMonitorHelper flowmon;
478 Ptr<FlowMonitor> monitor = flowmon.InstallAll();
479 monitor->Start( Seconds(dTime2StartStatistics) );
480
481 // #####
482 // Run simulation
483 // #####
484 Simulator::Stop( Seconds( dSimTime ));
485 Simulator::Run ();
486
487 // #####

```

```

488 // Simulation output processing
489 // #####
490 //
491 // Power Consumption processing
492 // Simulation output: collect and save energy consumption metrics. There is an important difference
493 // between final consumption values with "#include "ns3/simple-device-energy-model.h" is included or not.
494 //std::cout << "saving Energy Consumption data..." << std::endl;
495 // Energy consumption metrics will be saved in a new file named EnergyConsumption_Results.txt
496 // *fileEnergy->GetStream() << "STA_ID, STARemainingEnergy, STAEnergyConsumption" << std::endl;
497 //
498 double avgPowerConsumption = 0;
499 for (uint32_t iEs = 0; iEs < adhocNodes.GetN(); iEs++)
500 {
501 // Energy source by sources (if automatic installation)
502 Ptr<BasicEnergySource> basicSourcePtr = DynamicCast<BasicEnergySource>(
503     sources.Get(iEs));
504
505 // Energy model (if automatic installation)
506 // device energy model
507 Ptr<DeviceEnergyModel> basicRadioModelPtr =
508     basicSourcePtr->FindDeviceEnergyModels("ns3::WifiRadioEnergyModel").Get(
509         0);
510
511 // NS_LOG_UNCOND("energyModel collected pointer: " << basicRadioModelPtr
512 // << "\n energySource collected pointer: " << basicSourcePtr);
513
514 NS_ASSERT(basicRadioModelPtr != NULL);
515 //
516 avgPowerConsumption += basicRadioModelPtr->GetTotalEnergyConsumption();
517 // *fileEnergy->GetStream() << iEs << ", "
518 // << basicSourcePtr->GetRemainingEnergy() << ", "
519 // << basicRadioModelPtr->GetTotalEnergyConsumption() << std::endl;
520 //
521 }
522 avgPowerConsumption = avgPowerConsumption/adhocNodes.GetN();
523
524 // T-put calculation according to http://paginas.fe.up.pt/~mricardo/doc/conferences/nstools2009/flowmon-paper.pdf
525 double statDurationTX = 0;
526 double statDurationRX = 0;
527 //
528 monitor->CheckForLostPackets();
529 Ptr<Ipv4FlowClassifier> classifier = DynamicCast<Ipv4FlowClassifier>(
530     flowmon.GetClassifier());
531 std::cout << "Saving output data..." << std::endl;
532 std::map<FlowId, FlowMonitor::FlowStats> stats = monitor->GetFlowStats();
533 for (std::map<FlowId, FlowMonitor::FlowStats>::const_iterator iter =
534     stats.begin(); iter != stats.end(); ++iter)
535 {
536 // some metrics calculation
537 statDurationRX = iter->second.timeLastRxPacket.GetSeconds()- iter->second.timeFirstTxPacket.GetSeconds();
538 statDurationTX = iter->second.timeLastTxPacket.GetSeconds()- iter->second.timeFirstTxPacket.GetSeconds();
539
540 double meanDelay, meanJitter, packetLossRatio, txTput, rxTput;
541 if (iter->second.rxPackets > 0)
542 {
543     meanDelay = (iter->second.delaySum.GetSeconds()
544         / iter->second.rxPackets);
545 }
546 else // this value is set to zero because the STA is not receiving any packet
547 {
548     meanDelay = 0;
549 }
550 //
551 if (iter->second.rxPackets > 1)
552 {
553     meanJitter = (iter->second.jitterSum.GetSeconds()
554         / (iter->second.rxPackets - 1));
555 }
556 else // this value is set to zero because the STA is not receiving any packet
557 {

```

```

558     meanJitter = 0;
559 }
560 //
561 if (statDurationTX > 0)
562 {
563     txTput = iter->second.txBytes * 8.0 / statDurationTX / 1000 / 1000;
564 }
565 else
566 {
567     txTput = 0;
568 }
569 //
570 if (statDurationRX > 0)
571 {
572     rxTput = iter->second.rxBytes * 8.0 / statDurationRX / 1000 / 1000;
573 }
574 else
575 {
576     rxTput = 0;
577 }
578 //
579 if ((iter->second.lostPackets > 0) & (iter->second.rxPackets > 0))
580 {
581     packetLossRatio = (double) (iter->second.lostPackets
582         / (double) (iter->second.rxPackets + iter->second.lostPackets));
583 }
584 else
585 {
586     packetLossRatio = 1;
587 }
588 // To see all flows
589 //Ipv4FlowClassifier::FiveTuple t = classifier->FindFlow(iter->first);
590 //std::cout << "Some problem to save metrics" << std::endl;
591 //std::cout << "Flow ID: " << iter->first << ", Source Port: "
592 //    << t.sourcePort << ", Destination Port: " << t.destinationPort
593 //    << " (" << t.sourceAddress << " -> " << t.destinationAddress << ")" << std::endl;
594
595 // Save metrics
596 if (myfile)
597 {
598     myfile << (iter->first) << ", "
599         << (iter->second.lostPackets) << ", "
600         //
601         << (iter->second.txPackets) << ", "
602         //
603         << (iter->second.txBytes) << ", "
604         //
605         << txTput << ", "
606         //
607         << (iter->second.rxPackets) << ", "
608         //
609         << (iter->second.rxBytes) << ", "
610         //
611         << rxTput << ", "
612         //
613         << meanDelay << ", "
614         //
615         << meanJitter << ", "
616         //
617         << packetLossRatio << ", ";
618 }
619
620
621 //
622 //Store some input parameters into a file (only for numerical parameters)
623 myfile << avgPowerConsumption << ", "
624     << iNWifis << ", "
625     << dTxp << ", "
626     << iProtocol << ", "
627     << dSimTime << std::endl;

```

```
628     }  
629     toc ();  
630     myfile.close ();  
631     //  
632     Simulator::Destroy ();  
633  
634     return 0;  
635 }
```