



Universidade do Estado do Rio de Janeiro
Centro de Tecnologia e Ciências
Faculdade de Engenharia

Wellington Rodrigo de Freitas Costa


Protocolo de Comunicação para Frota de Embarcações Não Tripuladas

Rio de Janeiro

2020

Wellington Rodrigo de Freitas Costa

Protocolo de Comunicação para Frota de Embarcações Não Tripuladas



Dissertação apresentada, como requisito parcial para obtenção do título de Mestre em Engenharia Eletrônica, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Redes de Telecomunicações.

Orientador: Prof. D.Sc. Alexandre Sztajnberg

Orientador: Prof. D.Sc. José Paulo Vilela Soares da Cunha

Rio de Janeiro

2020

CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIUS / BIBLIOTECA CTC/B

C837 Costa, Wellington Rodrigo de Freitas.
Protocolo de comunicação para frota de embarcações não tripuladas / Wellington Rodrigo de Freitas Costa. – 2020.
125f.

Orientadores: Alexandre Sztajnberg, José Paulo Vilela Soares da Cunha.
Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Engenharia eletrônica - Teses. 2. Sistemas de telecomunicação - Teses. 3. Sistemas de comunicação sem fio - Teses. 4. IEEE 802.11 (Normas) - Teses. 5. Controle automático - Teses. I. Sztajnberg, Alexandre. II. Cunha, José Paulo Vilela Soares da. III. Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia. IV. Título.

CDU 621.39:681.51

Bibliotecária: Júlia Vieira – CRB7/6022

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta tese, desde que citada a fonte.

Assinatura

Data

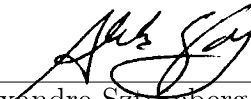
Wellington Rodrigo de Freitas Costa

Protocolo de Comunicação para Frota de Embarcações Não Tripuladas

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre em Engenharia Eletrônica, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Redes de Telecomunicações.

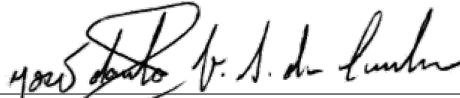
Aprovado em: 19 de Fevereiro de 2020

Banca Examinadora:



Prof. D.Sc. Alexandre Sztajnberg (Orientador)

Instituto de Matemática e Estatística - UERJ



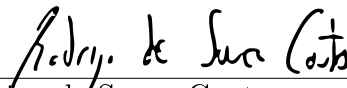
Prof. D.Sc. José Paulo Vilela Soares da Cunha (Orientador)

Faculdade de Engenharia - UERJ



Prof. D.Sc. Marcelo Gonçalves Rubinstein

Faculdade de Engenharia - UERJ



Prof. D.Sc. Rodrigo de Souza Couto

Universidade Federal do Rio de Janeiro - UFRJ

Rio de Janeiro

2020

AGRADECIMENTO

Agradeço aos meus pais pelo apoio e incentivo, aos meus orientadores pela confiança, paciência e dedicação, aos colegas do Laboratório de Ciência da Computação pelo apoio e ajuda técnica e psicológica durante a elaboração do projeto, por fim termino agradecendo ao coordenador do Laboratório de Ciência da Computação por incentivar e disponibilizar os recursos para o desenvolvimento do trabalho.

RESUMO

Costa, Wellington Rodrigo de Freitas. *Protocolo de Comunicação para Frota de Embarcações Não Tripuladas*. 125 f. Dissertação (Mestrado em Engenharia Eletrônica) - Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro (UERJ), Rio de Janeiro, 2020.

Este trabalho aborda a comunicação sem fios para uma frota de embarcações de superfície não tripuladas, considerando cenários onde há uma estação base e uma ou mais embarcações que executam missões simples, tais como se deslocar para uma localização desejada, realizar sensoriamento e cercar uma determinada área. Para realizar as missões, as embarcações precisam se comunicar com a estação base e entre si, utilizando um protocolo de comunicação. Para isso foram desenvolvidos: (i) um modelo de frota com dois tipos de elementos: uma estação base e uma ou mais embarcações, que podem agir como nó remetente de informação (obtida de sensores locais, por exemplo), nó destinatário de informação de controle recebidos da estação base ou de estações vizinhas, ou ainda como nó intermediário que reencaminha mensagens para a base, aumentando o raio de operação; (ii) um protocolo de comunicação no nível de aplicação para a comunicação entre os elementos considerados; (iii) uma implementação deste protocolo, desenvolvida no simulador de redes NS-3; (iv) uma implementação simples de controle cinemático das embarcações no MATLAB; (v) a integração entre o NS-3 e o MATLAB para utilizar as melhores características de cada ambiente de simulação. Rede Wi-Fi em modo *ad hoc* foi combinada com o protocolo de roteamento AODV e com mensagens *broadcast* IP limitadas em um salto para: (i) estabelecer um mecanismo de comunicação de 1-N (um para muitos) entre embarcações vizinhas, dado que cada mensagem enviada é interceptada pelos nós com enlace direto; (ii) prover um mecanismo para detecção da vizinhança de cada embarcação; (iii) permitir um controle maior da aplicação e com isso a possibilidade da agregação de informações. Foram realizadas simulações para verificar o desempenho do protocolo e identificar os limites impostos pela velocidade das embarcações, potência de sinal *Wi-Fi* recebido, as taxas de entrega e tempo de resposta. Os resultados das simulações indicam que o protocolo proposto se comporta de forma satisfatória e oferece o suporte para o desenvolvimento de sistemas de controle mais completos, mantendo ainda alguma autonomia das embarcações.

Palavras-chave: Comunicação em superfície marítima; Embarcações não tripuladas; Redes sem fio; IEEE 802.11; Simulador de Rede (NS-3); MATLAB.

ABSTRACT

Costa, Wellington Rodrigo de Freitas. *Communication Protocol for Fleet of Unmanned Vessels*. 125 f. Dissertação (Mestrado em Engenharia Eletrônica) - Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro (UERJ), Rio de Janeiro, 2020.

This work addresses wireless communication for a fleet of unmanned surface vessels, considering scenarios where there is a base station and one or more vessels that perform simple missions, such as moving to a desired location, performing sensing and surrounding a certain area. To carry out the missions, the vessels need to communicate with the base station and with each other, using a communication protocol. For this, the following were developed: (i) a fleet model with two types of elements: a base station and one or more vessels, which can act as an information sender node (obtained from local sensors, for example), an information recipient node of control received from the base station or neighboring stations, or as an intermediate node that forwards messages to the base, increasing the operating radius; (ii) an application-level communication protocol for communication between the elements considered; (iii) an implementation of this protocol, developed in the NS-3 network simulator; (iv) a simple implementation of kinematic control of vessels in MATLAB; (v) the integration between NS-3 and MATLAB to use the best characteristics of each simulation environment. Wi-Fi network in ad hoc mode has been combined with the AODV routing protocol and with limited broadcast IP messages in one jump to: (i) establish a 1-N communication mechanism (one for many) between neighboring vessels, since each message sent is intercepted by nodes with a direct link; (ii) provide a mechanism for detecting the neighborhood of each vessel; (iii) allowing greater control of the application and, with that, the possibility of aggregating information. Simulations were carried out to check the accuracy of the protocol and to identify the limits imposed by the speed of the vessels, received Wi-Fi signal strength, delivery rates and response time. The performance of the protocol was also evaluated according to the height of the antennas, the transmission power and the frequency of the signal. The results of the simulations indicate that the proposed protocol behaves satisfactorily and offers support for the development of more complete control systems, while maintaining some vessel autonomy.

Keywords: Communication on sea surface; Unmanned Vessels; Wireless Networks; IEEE 802.11; Network Simulator (NS-3); MATLAB.

LISTA DE FIGURAS

Figura 1	- Arquitetura Fundamental de um USV Típico. Traduzido de [1]	26
Figura 2	- Fluxo do Sistema GNC. Adaptado de [1]	27
Figura 3	- Classificação dos Subsistemas de Guiagem. Traduzido de [1]	27
Figura 4	- Classificação dos Subsistemas de Navegação. Traduzido de [1]	28
Figura 5	- Classificação dos Subsistemas de Controle. Traduzido de [1]	29
Figura 6	- Diagrama de blocos do modelo dinâmico das embarcações para cada grau de liberdade.	30
Figura 7	- Diagrama de blocos do controle PD de um grau de liberdade.....	30
Figura 8	- Diagrama de blocos do controlador PD de um grau de liberdade e sinais amostrados.	31
Figura 9	- Diagrama de blocos usando controlador PD com saturador e sinais amostrados. ..	32
Figura 10	- Exemplo de Topologia Sem Nó Coordenador.....	41
Figura 11	- Exemplo de Topologia Com Nó Coordenador.	41
Figura 12	- Tipos de Comunicação.	45
Figura 13	- Comunicação por Difusão (<i>Broadcast</i>).	47
Figura 14	- Agregação dos Pacotes de Telemetria (4.3.1.2).	49
Figura 15	- Máquina de Estados Finito do Protocolo.	57
Figura 16	- Diagrama de Atividades das Aplicações.	60
Figura 17	- <i>Handshake</i> de Três Vias.....	61
Figura 18	- Finalização da Aplicação Embarcação.	61
Figura 19	- Diagrama de classes da camada de aplicação.	64
Figura 20	- Exemplificação do modelo de dois raios. Adaptado de [75]	66
Figura 21	- Diagrama de sequência (Integração NS-3 com MATLAB).....	70
Figura 22	- Resultados Extraídos de [16, 75], respectivamente (adaptados).....	77
Figura 23	- Distância vs Potência do Sinal Recebido (Modelo na Seção 5.1.2). (a) à (d) combinam frequências de 2,4GHz e 5,8GHz, com e sem <i>fading</i>	79
Figura 24	- Teste de Comunicação: Potência de Sinal Recebido (tamanho de 2200 <i>bytes</i>), considerando velocidades de 1, 5 e 10m/s. (a) Estação base para embarcação; (b) Embarcação para estação base e (c) Embarcação para embarcação.	83

Figura 25 - Teste de Comunicação: Atraso, considerando a velocidade de 10m/s para os três tipos de teste (B2V, V2B e V2V). (a) Tamanho de 132 bytes; (b) Tamanho de 456 bytes e (c) Tamanho de 2200 bytes.	84
Figura 26 - Teste de Comunicação: Taxa de Entrega de Pacotes, considerando-se a velocidade de 10m/s e os três tipos de teste (B2V, V2B e V2V). (a) Tamanho de 132 bytes; (b) Tamanho de 456 bytes e (c) Tamanho de 2200 bytes.....	85
Figura 27 - Teste de Comunicação: Taxa de Transferência, considerando-se a velocidade de 10m/s e os três tipos de teste (B2V, V2B e V2V). (a) Tamanho de 132 bytes; (b) Tamanho de 456 bytes e (c) Tamanho de 2200 bytes.	86
Figura 28 - Cenário 2: Formação com 5 Embarcações em Linha até o Destino.	89
Figura 29 - Cenário 3: Formação com 5 Embarcações ao Redor do Destino.	89
Figura 30 - Cenário 1: Deslocamento ao Longo do Tempo (Imagem Obtida no MATLAB). ...	90
Figura 31 - Cenário 1: Atraso de Ida e Volta (RTT).....	90
Figura 32 - Cenário 1: Taxa de Entrega de Pacotes (PDR).	91
Figura 33 - Cenário 2: Posicionamento das Embarcações (Imagem Obtida no MATLAB).....	92
Figura 34 - Cenário 2: Atraso de Ida e Volta (RTT).....	93
Figura 35 - Cenário 2: Taxa de Entrega de Pacotes (PDR).	94
Figura 36 - Cenário 2: Vizinhaça sem Média Móvel.	95
Figura 37 - Cenário 2: Vizinhaça com Média Móvel dos Últimos 120 segundos.	96
Figura 38 - Cenário 3: Trajetórias Executadas (Imagem Obtida no MATLAB).	97
Figura 39 - Cenário 3: Atraso de Ida e Volta (RTT).....	98
Figura 40 - Cenário 3: Taxa de Entrega de Pacotes (PDR).	100
Figura 41 - Cenário 3: Vizinhaça sem Média Móvel.	101
Figura 42 - Cenário 3: Vizinhaça com Média Móvel dos Últimos 120 Segundos.	102

LISTA DE TABELAS

Tabela 1	- Protocolos utilizados	43
Tabela 2	- Estrutura do Pacote das Mensagens do Protocolo.	52
Tabela 3	- Estrutura do Pacote de Telemetria do Protocolo.....	54
Tabela 4	- Estrutura de Dados Enviada do NS-3 para o MATLAB.....	69
Tabela 5	- Estrutura de Dados Retornada do MATLAB para o NS-3.	72
Tabela 6	- Parâmetros Gerais.	76
Tabela 7	- Parâmetros para o Teste do Modelo de Propagação.	78
Tabela 8	- Parâmetros para o Teste de Comunicação.	80
Tabela 9	- Parâmetros da Simulação em cada Cenário.....	87

LISTA DE SIGLAS

Abreviações

AAV	<i>Autonomous Aerial Vehicle</i> —Veículo Aéreo Autônomo
AIS	<i>Automatic Identification System</i> —Sistema de Identificação Automática
AODV	<i>Ad hoc On-demand Distance Vector</i> —Ad hoc Vetor Distância por Demanda
AP	<i>Access Point</i> —Ponto de Acesso
APF	<i>Artificial Potential Field</i> —Campos Potenciais Artificiais
API	<i>Application Programming Interface</i> —Interface de Programação de Aplicação
ASC	<i>Autonomous Surface Craft</i> —Embarcação de Superfície Autônoma
ASV	<i>Autonomous Surface Vehicle/Vessel</i> —Veículo/Embarcação de Superfície Autônomo
AUV	<i>Autonomous Underwater Vehicle</i> —Veículo Subaquático Autônomo
B2V	<i>Base to Vessel</i> —Estação para Embarcação
COLREGS	<i>Convention on the International Regulations for Preventing Collisions at Sea</i> —Convenção nos Regulamentos Internacionais para a Prevenção de Colisões no Mar
DARPA	<i>Defense Advanced Research Projects Agency</i> —Agência de Projeto e Pesquisa de Defesa Avançada
DCE	<i>Direct Code Execution</i> —Execução Direta de Código
DLL	<i>Dynamic Linked Libraries</i> —Bibliotecas Vinculadas Dinamicamente
DTN	<i>Delay Tolerant Networking</i> —Rede tolerante a atraso
GNC	<i>Guidance, Navigation and Control</i> —Guiagem, Navegação e Controle
GPS	<i>Global Positioning System</i> —Sistema de Posicionamento Global
ICT	<i>Information and Communications Technologies</i> —Tecnologias de Informação e Comunicação
IDE	<i>Integrated Development Environment</i> —Ambiente de Desenvolvimento Integrado
IMO	<i>International Maritime Organization</i> —Organização Marítima Internacional
IMU	<i>Inertial Measurement Unit</i> —Unidade de Medidas Inerciais

IP	<i>Internet Protocol</i> —Protocolo de Internet
IPv6	<i>Internet Protocol version 6</i> —Protocolo de Internet versão 6
ITS	<i>Intelligent Transportation System</i> —Sistema de Transporte Inteligente
LAN	<i>Local Area Network</i> —Rede Local
LIDAR	<i>Light Detection and Ranging</i> —Detecção e Variação de Luz
LPWA	<i>Low Power Wide Area</i> —Baixa Potência e Longa Distância
MAC	Medium Access Control—Controle de Acesso ao Meio
MANET	<i>Mobile Ad Hoc Network</i> —Rede Móvel <i>Ad Hoc</i>
MPDU	<i>MAC Protocol Data Unit (MPDU)</i> —Unidade de Dados do Protocolo MAC
MTU	<i>Maximum Transmission Unit</i> —Unidade Máxima de Transmissão
NF	Noise Figure—Parâmetro de Ruído
NS-3	<i>Network Simulator 3</i> —Simulador de Rede 3
PDR	<i>Packet Delivery Ratio</i> —Taxa de Entrega de Pacotes
PID	Proporcional, Integral e Derivativo
POO	Programação Orientada a Objetos
RERR	<i>Route Error</i> —Erro de Rota
RFID	<i>Radio Frequency Identification</i> —Identificação por Rádio Frequência
ROV	<i>Remotely Operated Vehicle</i> —Veículo de Operação Remota
RREP	<i>Route Reply</i> —Resposta de Rota
RREQ	<i>Route Request</i> —Requisição de Rota
RTT	<i>Round Trip Time</i> —Tempo de Ida e Volta
SO	<i>Shared Objects</i> —Objetos Compartilhados
TCP	<i>Transmission Control Protocol</i> —Protocolo de Controle de Transmissão
TTL	<i>Time To Live</i> —Tempo de Vida
UAV	<i>Unmanned Aerial Vehicle</i> —Veículo Aéreo Não Tripulado
UGV	<i>Unmanned Ground Vehicle</i> —Veículo Terrestre Não Tripulado
USV	<i>Unmanned Surface Vehicle/Vessel</i> —Veículo/Embarcação de Superfície Não Tripulado
UUV	<i>Unmanned Underwater Vehicle</i> —Veículo Subaquático Não Tripulado
UV	<i>Unmanned Vehicle</i> —Veículo Não Tripulado
V2B	<i>Vessel to Base</i> —Embarcação para Estação Base
V2V	<i>Vessel to Vessel</i> —Embarcação para Embarcação

VANET	<i>Vehicle Ad Hoc Network</i> —Rede Veicular <i>Ad Hoc</i>
WAM-V	<i>Wave Adaptive Modular Vessel</i> —Embarcação Modular Adaptável às Ondas
WiMAX	<i>Worldwide Interoperability for Microwave Access</i> —Interoperabilidade Mundial para Acesso em Micro-ondas
WSN	<i>Wireless Sensor Network</i> —Rede de Sensores Sem Fio

SUMÁRIO

	INTRODUÇÃO	14
1	TECNOLOGIAS ADOTADAS	18
1.1	Comunicação Sem Fio	18
1.2	Simuladores de Redes	19
2	VEÍCULOS NÃO TRIPULADOS	23
2.1	Tipos de Veículos Não Tripulados	23
2.2	Embarcações de Superfície Não Tripuladas	25
2.3	Modelo Dinâmico das Embarcações	29
2.4	Controle das Embarcações	30
3	TRABALHOS CORRELATOS	34
3.1	Comunicação Marítima e Redes <i>Ad Hoc</i>	34
3.2	Veículos não Tripulados	35
3.3	Integração entre Ambientes de Simulação	37
4	PROTOCOLO DE COMUNICAÇÃO PARA FROTA DE EMBARCA- ÇÕES DE SUPERFÍCIE NÃO TRIPULADAS	40
4.1	Elementos Considerados	40
4.2	Características de Comunicação Consideradas	42
4.2.1	<u><i>Ad hoc On-Demand Distance Vector</i> - AODV</u>	43
4.2.2	<u>Difusão (<i>Broadcast</i>)</u>	44
4.2.3	<u>Parâmetros de Enlace e Tamanho dos Campos</u>	49
4.3	Protocolo Proposto	50
4.3.1	<u>Estrutura das Mensagens</u>	52
4.3.1.1	Pacote de Mensagem.....	52
4.3.1.2	Pacote de Telemetria	53
4.3.1.3	Observações	55
4.3.2	<u>A Máquina de Estados</u>	56

5	IMPLEMENTAÇÃO DO PROTOCOLO EM SIMULADOR.....	63
5.1	Implementação NS-3	63
5.1.1	<u>Aplicações</u>	63
5.1.2	<u>Modelo de Propagação.....</u>	65
5.1.3	<u>Recursos Adicionais.....</u>	67
5.2	Detalhes da Integração do NS-3 com o MATLAB.....	68
6	SIMULAÇÕES	74
6.1	Parâmetros e Métricas Avaliadas	75
6.2	Avaliação Inicial	76
6.2.1	Primeiro Teste	76
6.2.2	Segundo Teste	80
6.3	Cenários	87
6.3.1	Resultado do Cenário 1	89
6.3.2	Resultado do Cenário 2	91
6.3.3	Resultado do Cenário 3	97
6.4	Discussão dos Resultados	103
	CONCLUSÃO	104
	REFERÊNCIAS.....	108
	APÊNDICE A - Arquivo de Configuração	117
	APÊNDICE B - <i>Script</i> para Inclusão do Módulo no NS-3.....	121
	APÊNDICE C - Modificações Extras	124

INTRODUÇÃO

Apesar da maior parte da superfície terrestre ser composta por água (principalmente oceanos) ainda é pouco explorada devido a falta de conhecimento e pesquisas especializadas [1]. Os oceanos são cruciais para diversas atividades econômicas, tais como: transporte de cargas, pesca, petróleo, minerais e turismo. Vários tipos de veículos tripulados e não tripulados, bem como infraestruturas como plataformas de petróleo, pisciculturas, boias e sistemas de sensores, dependem de várias Tecnologias de Informação e Comunicação (*Information and Communications Technologies*, ICT) e atualmente fazem parte das operações em oceanos e mares em todo o mundo [2].

Veículos não tripulados (Unmanned Vehicle, UV) são utilizados em aplicações estratégicas, tais como: militares, monitoramento ambiental, suporte a pesquisas científicas, execução de tarefas em ambientes hostis e exploração de recursos [1]. Seu uso tem permitido níveis mais altos de precisão e exatidão em muitas expedições de pesquisa, bem como menor custo, quando comparado com expedições de pesquisa típicas. Essa abordagem é particularmente relevante em ambientes desafiadores ou perigosos e se a troca de dados em tempo real é necessária para a condução do processo de pesquisa [2].

Depois do sucesso dos veículos operados remotamente, os sistemas autônomos marinhos começaram a desempenhar um papel importante nas operações e pesquisas marítimas. Há vários sistemas disponíveis, personalizados para diferentes tarefas e operações, com vários recursos. Existe uma ampla gama de tarefas em que sistemas autônomos podem ser usados para reduzir o risco da equipe, agilizar as operações ou reduzir seus custos. No entanto, essas operações precisam de tecnologias maduras e confiáveis que necessitem de atenção e manutenção mínimas da equipe, já que toda intervenção é dispendiosa, leva um tempo significativo e envolve um risco maior [2].

Com uma maior autonomia as unidades científicas podem se concentrar mais em objetivos de alto nível (coleta de dados científicos e sua interpretação *on-line*), em vez das técnicas de baixo nível para controlá-los. Além disso, mais unidades autônomas podem ser mais facilmente integradas em sistemas com vários tipos de veículos, o que pode fornecer uma visão mais ampla e detalhada do fenômeno investigado. Essa operação *on-line* deve suportar a aquisição e o compartilhamento de dados em tempo quase real, a fim de monitorar o progresso das operações, permitindo uma reação rápida a eventos inesperados e uso mais eficiente dos recursos. No entanto, fora das áreas costeiras, há um acesso muito limitado aos recursos de comunicação que não seja satélites [2].

Nos cenários marítimos, a falta de infraestrutura limita consideravelmente o acesso às tecnologias de telecomunicações, fazendo com que as soluções predominantes sejam os sistemas de satélite ou outros sistemas de comunicação de longo alcance e de baixa taxa de bits. Porém, missões científicas de longa duração muitas vezes podem exigir taxas de transferências maiores, podendo um único sensor gerar *megabytes* de dados por minuto. Isso poderia ser resolvido se houvesse uma maneira acessível, rápida e eficiente de comunicação em áreas remotas [2].

Perdas na comunicação podem levar a cenários indesejados, podendo dificultar a cooperação entre os agentes do sistema [3]. A maioria dos trabalhos de controle de formação geralmente considera que não há restrições na comunicação, tratando a topologia de comunicação como um grafo fortemente conectado e constante em relação ao tempo, situação que não corresponde à realidade das comunicações sem fio [4].

A coordenação de sistemas heterogêneos introduz a possibilidade de se criar uma rede de comunicação entre veículos. Ao lidar adequadamente com esses sistemas, a cooperação pode ser promovida entre diferentes produtores de dados e consumidores sem comprometer as operações, beneficiando todos os atores envolvidos. No entanto, esta realidade ainda não foi totalmente explorada e as tecnologias usadas são dependentes das necessidades dos atores marítimos. De fato, equipamentos de prateleira para fins de comunicação são tipicamente empregados nesses cenários, e há uma necessidade de padrões e interfaces comuns para interconectar tais sistemas [2].

O impacto das ICTs e da robótica na oceanografia e em outros assuntos marítimos reflete-se na crescente demanda por maiores taxas de dados, integridade de dados, comunicações em tempo real e robustez. O controle e monitoramento desses sistemas e a aquisição de dados de pesquisa são desafiados pela qualidade da comunicação de dados [2].

Objetivos

O objetivo principal é contribuir para o desenvolvimento de frotas de embarcações não tripuladas de superfície visando apresentar uma solução para a comunicação de tais embarcações. Apesar do foco principal ter sido a comunicação, o trabalho foi desenvolvido de forma a facilitar que pesquisas futuras possam agregar novas soluções e assim melhorar a solução proposta. Foi desenvolvido um protocolo de aplicação utilizando as tecnologias e protocolos de comunicação existentes, provendo a comunicação necessária. Algoritmos de consenso da teoria de controle são ferramentas eficazes para tarefas de computação distribuídas, por isso a importância da combinação dessas técnicas com abordagens de sistemas de controle em rede [5]. Assim, foram utilizados aspectos específicos das

tecnologias e dos protocolos de comunicação, para facilitar a troca de informações entre as embarcações, adicionando suas métricas de qualidade e as informações trocadas no fluxo de controle, a fim de dar maior autonomia às embarcações (por exemplo, pequenas ações relacionadas ao seu deslocamento de forma a manter a comunicação com a base e evitar colisão entre elas).

Metodologia

É utilizada a pilha de protocolos da *Internet* (*Transmission Control Protocol/Internet Protocol*, TCP/IP [6–11]) para prover os recursos necessários para a comunicação entre embarcações não tripuladas e uma estação base fixa. Adicionalmente foi desenvolvido um protocolo de aplicação para a troca de mensagens entre esses agentes, com informações de: comando e controle (telemetria), monitoramento, carga útil (de acordo com a missão) e gerenciamento de rede. Informações que formam uma taxonomia dos conteúdos de dados típicos trocados em uma rede de veículos autônomos [2] e fornecem à estação base o controle e monitoramento das embarcações e a elas as informações locais necessárias para que possam tomar suas ações a fim de manter a comunicação com a estação base e evitar colidir com suas vizinhas.

O simulador de redes (*Network Simulator 3*, NS-3 [12]) foi utilizado para implementar e avaliar o protocolo de aplicação desenvolvido, sua integração com o programa de cálculo numérico MATLAB [13] permitiu avaliar a proposta de controle integrado com a comunicação. Toda a comunicação é simulada pelo NS-3, que efetua regularmente chamadas à interface de programação (*Application Programming Interface*, API) do MATLAB para enviar os dados da comunicação e receber as informações de posição, velocidade e aceleração das embarcações.

Resultados

A combinação da pilha de protocolos da *Internet*, usando um canal de comunicação sem fio com o envio de mensagens por difusão no nível de rede (protocolo IP) limitadas em um salto, permitiu o estabelecimento de um mecanismo de comunicação de 1–N (um para muitos) para que as embarcações pudessem identificar dinamicamente suas vizinhas. Para melhor simular o ambiente marítimo foram acrescentados o modelo de propagação dois raios [14–16], o modelo dinâmico das embarcações com controlador proporcional derivativo (PD). As aplicações da estação base e das embarcações, bem como os recursos adicionais necessários foram implementados como um módulo novo para o simulador de redes NS-3.

Os recursos implementados foram avaliados em partes: a comunicação e toda a pilha de protocolos da *Internet* desenvolvidas diretamente no NS-3, a dinâmica da mobilidade das embarca-

ções desenvolvidas no software de cálculo numérico MATLAB, a integração entre os dois ambientes utilizados (NS-3 e MATLAB) e por fim os resultados de testes iniciais sem usar as aplicações desenvolvidas foram comparados com os resultados das simulações finais em três cenários diferentes.

Organização do Texto

Esta dissertação está estruturada da seguinte forma: o Capítulo 1 é destinado a apresentar os conceitos, tecnologias, ferramentas e protocolos pesquisados; o Capítulo 2 apresenta os conceitos básicos sobre veículos não tripulados, os componentes de embarcações de superfície não tripuladas e como foi feita a modelagem das embarcações, ajudando o entendimento do protocolo proposto; alguns trabalhos relacionados são apresentados no Capítulo 3; no Capítulo 4 são detalhados os elementos considerados e as escolhas realizadas para a proposta, a forma de interação entre os elementos, as estruturas das mensagens e o protocolo de comunicação desenvolvido; o Capítulo 5 destaca como o protocolo proposto foi implementado no NS-3 e detalha como foi realizada a integração do NS-3 com o ambiente do MATLAB; no Capítulo 6 apresentamos alguns cenários, detalhamos como a simulação é realizada e discutimos os resultados obtidos; e por fim fazemos nossas considerações finais e apresentamos algumas propostas de trabalhos futuros na Conclusão.

1 TECNOLOGIAS ADOTADAS

Com o objetivo de desenvolver uma aplicação distribuída que permitisse a comunicação entre nós móveis com um nó fixo utilizando as informações das mensagens trocadas e as métricas de qualidade da comunicação para auxiliar a movimentação desses nós móveis, foram estudadas as tecnologias que pudessem ser utilizadas para facilitar essa comunicação. E como meta adicional de permitir que os nós móveis apresentassem um comportamento em seu deslocamento de forma a garantir uma distância mínima (evitar colisão) e máxima (perda do sinal) entre eles, cumprindo uma determinada formação especificada pelo nó fixo e a comunicação com este se mantivesse dentro de certos padrões de qualidade especificados, mesmo que a distância entre o nó fixo e o local de destino a ser atingido fosse maior que o alcance da comunicação do nó fixo.

Este capítulo é dedicado a apresentar as tecnologias pesquisadas e utilizadas e está subdividido nas seguintes seções: na Seção 1.1 são apresentadas as tecnologias de comunicação sem fio estudadas e a Seção 1.2 apresenta alguns dos simuladores de redes estudados que atendessem aos requisitos exigidos pela tecnologia adotada.

1.1 Comunicação Sem Fio

Das redes e protocolos estudados e utilizados na comunicação em ambientes marítimos destacam-se: Rede de Sensores Sem Fio (*Wireless Sensor Network*, WSN) [17], Rede Móvel *Ad Hoc* (*Mobile Ad Hoc Network*, MANET) - se seus nós móveis forem veículos é denominada de Rede Veicular *Ad Hoc* (*Vehicle Ad Hoc Network*, VANET) [18, 19] - e Rede Tolerante a Atraso (*Delay Tolerant Networking*, DTN), juntamente com soluções baseadas no protocolo IP (principalmente a versão seis, IPv6) [20, 21] para permitir sua ligação com a *Internet* e facilidade de integração com outras tecnologias. A literatura também propõe frequentemente o uso de rede Interoperabilidade Mundial para Acesso em Micro-ondas (*Worldwide Interoperability for Microwave Access*, WiMAX) [22] como uma solução adequada para ambientes marítimos [2]. Os experimentos realizados por [23] mostraram ser viável a utilização do LoRaWAN [24], protocolo de rede de Baixa Potência e Longa Distância (*Low Power Wide Area*, LPWA), como uma possível solução para comunicações sobre o mar.

As redes locais (*Local Area Network*, LAN) são hoje uma das mais importantes tecnologias de rede de acesso à *Internet* e estão presentes nos mais variados locais (escritórios, residências,

instituições educacionais, restaurantes, aeroportos). A Wi-Fi foi a tecnologia que se destacou como alternativa sem fio para a criação destas LANs, facilitando sua criação e conexão com a *Internet*. Ela apresenta diversas extensões cada uma com especificações e objetivos diferentes, mas que seguem as três extensões básicas (802.11a, 802.11b e 802.11g) [19,25].

A arquitetura IEEE 802.11 consiste em vários componentes que interagem para fornecer uma LAN sem fio que suporta a mobilidade das estações de forma transparente para as camadas superiores. Ela apresenta dois modos: infraestrutura que apresenta estação-base, denominada Ponto de Acesso (*Access Point*, AP), onde toda a comunicação passa pela estação-base; *Ad Hoc* que dispensa a presença da estação-base, as estações se comunicam diretamente umas com as outras. Um AP é uma estação, que além de apresentar as suas funcionalidades básicas de estação, fornece acesso aos serviços de distribuição para acesso ao meio sem fio para as estações associadas [19,25–30].

De acordo com as características dos elementos considerados (apresentadas na Seção 4.1) e o cenário com uma estação base em posição fixa que se comunica com embarcações móveis que podem entrar e sair de seu alcance de comunicação, foi escolhida uma solução que pudesse permitir a entrada e saída de nós na rede de comunicação, com facilidade de implementação (criação da rede) e que pudesse permitir a comunicação entre dois nós (estação base e embarcação, especificamente) com distâncias entre eles maior que o laço de comunicação do canal sem fio utilizado. Diante deste cenário a tecnologia que melhor atendeu as necessidades citadas foi a Wi-Fi no modo *ad hoc*, as características de comunicação consideradas são apresentadas na Seção 4.2.

Uma rede *ad hoc* é composta exclusivamente de estações dentro do alcance de comunicação mútua entre si por meio sem fio sem um AP para fornecer acesso aos serviços de distribuição. Ela é normalmente criada de maneira espontânea, permitindo que nós entrem e saiam facilmente. Sua principal característica é sua extensão temporal e espacial limitada, permitindo criar e dissolver redes de forma direta e conveniente, exigindo o mínimo de conhecimento do usuário e pouco ou nenhum investimento de tempo e recursos adicionais além das estações participantes [26–30].

1.2 Simuladores de Redes

As redes sem fio, como MANET, WSN e a identificação por rádio frequência (Radio Frequency Identification, RFID), estão crescendo rapidamente em tamanho e complexidade e têm sido amplamente utilizadas em diversas aplicações. Isso leva a um grande esforço dos pesquisadores para projetar novos protocolos e algoritmos para redes sem fio [31].

Experimentos de campo com protótipos de hardware e software embarcado são altamente realistas e confiáveis, mas custam muito tempo para projetar o hardware, construir as redes sem fio e executar muitos experimentos em uma ampla variedade de configurações. Além disso, a maioria das medições não é repetida e requer um grande esforço. Portanto, as simulações são indispensáveis e utilizadas por muitos pesquisadores para resolver os problemas de configurabilidade e repetibilidade. Simulações são usadas para projetar e avaliar novos protocolos e aplicações de redes, porém os desenvolvedores de simuladores de rede enfrentam desafios para fornecer os recursos mínimos, tais como: modelos de canais precisos, modelos tridimensionais (3-D), portabilidade e validação efetiva [31].

Projetar e produzir um sistema inteligente é uma tarefa complexa e difícil, devido à grande quantidade de componentes com seus parâmetros que precisam ser considerados durante o projeto e a produção. E mesmo com uma imagem clara dos requisitos e da especificação geral do sistema, essas especificações raramente capturam a complexidade que é adicionada pela característica distribuída do sistema. Os problemas que surgem devido a essa complexidade adicional são geralmente difíceis de capturar na análise analítica e muitas vezes são negligenciados. Além disso, o resultado da análise também é geralmente afetado por incertezas, devido as simplificações e aproximações. Para alcançar um nível mais alto de certeza, a priori, sobre a validade da funcionalidade, a verificação virtual por meio de simulador é muito útil e, possivelmente, a única ferramenta viável. Estudos com simulador podem revelar problemas nos estágios iniciais e acelerar o projeto e a produção. As propriedades de modelos computacionais como a lógica e a formalização estão revelando os principais fatores que determinam as propriedades do objeto original da pesquisa [32].

Os simuladores são populares em quase todos os campos da engenharia, como dinâmica veicular, comunicação em rede e eletrônica. Quando se trata de simular o comportamento de um sistema de transporte inteligente (Intelligent Transportation System, ITS) esses campos se sobrepõem. Isso exige um ambiente de simulação combinada, em que o movimento veicular juntamente com o sensoriamento, a eletrônica e a comunicação precisam ser simulados para que seja possível obter uma análise adequada da perspectiva de vulnerabilidade, confiabilidade e desempenho [32].

O OMNeT++ é uma biblioteca e estrutura de simulação C++ [33, 34] extensível, modular e baseada em componentes, principalmente para a criação de simuladores de rede, incluindo redes de comunicação com e sem fio, redes *on-chip*, redes de filas e assim por diante. Funcionalidades específicas de domínio, como suporte a redes de sensores, redes ad-hoc, protocolos de *internet*, modelagem de desempenho, redes fotônicas e outras, são fornecidas por estruturas de modelo, de-

envolvidas como projetos independentes. Ele oferece um ambiente de desenvolvimento integrado (Integrated Development Environment, IDE baseado em Eclipse [35], um ambiente gráfico de execução e apresenta extensões para simulação em tempo real, emulação de rede, integração de banco de dados, integração com SystemC e várias outras funções. É distribuído sob a Licença Pública Acadêmica (Academic Public License, APL) e apresenta ampla popularidade como uma plataforma de simulação de rede na comunidade científica e em ambientes industriais [36].

O openWNS é uma plataforma de simulação em nível de sistema de código aberto, sobre uma Licença Pública Geral de Biblioteca (Library General Public License, LGPL [37]), para avaliação de desempenho e comparação de sistemas de comunicação móvel sem fio e multicelulares. Oferece emulação próxima da implementação da pilha de protocolos, modelagem de interferência em cenários de referência, modelos de mobilidade, geradores de carga de tráfego, métodos de avaliação estatística e modelos de canais. Ele usa uma arquitetura de *plugins* para permitir que os desenvolvedores estendam o simulador para suas necessidades específicas. É desenvolvido em Python [38] e utiliza bibliotecas em C++ [39].

O NS-2 é um simulador de eventos discretos de código aberto desenvolvido em C++ e OTcl [40], sobre a Licença Pública Geral (General Public License, GPL [41]), voltado para pesquisa de redes, fornecendo suporte substancial para a simulação de protocolos TCP, de roteamento e *multicast* em redes com e sem fio (locais e por satélite). Ele começou como uma variante do simulador de rede REAL em 1989 e em 1995 recebeu apoio da Agência de Projeto e Pesquisa de Defesa Avançada (Defense Advanced Research Projects Agency, DARPA [42]) [43].

As ferramentas de simulação citadas suportam a modelagem sem fio, uma variedade de protocolos de comunicação e redes de grande escala com até milhares de nós sem fio, porém apresentam limitações quanto aos modelos de canais implantados, aos modelos tridimensionais (3-D), a portabilidade e a reusabilidade dos modelos e a validação dos resultados da simulação, o que frequentemente exige o uso de outros simuladores e análises teóricas para a validação, tarefa que exige maior tempo e esforço [31].

O NS-3 é um simulador de redes de eventos discretos de código aberto (GPL) voltado principalmente para pesquisa e uso educacional. Iniciado em 2006, ele não é uma extensão compatível com o NS-2 e sim um projeto totalmente novo, que mesmo sendo escrito em C++, como o NS-, não suporta suas APIs. O NS-3 fornece modelos de como as redes de dados de pacote funcionam e executam e um mecanismo de simulação para os usuários conduzirem experimentos de simulação, simulando completamente os protocolos e redes da *internet*, não se limitando a estes. Ele é proje-

tado como um conjunto de bibliotecas que podem ser combinadas entre si e com outras bibliotecas de software externas. Vários visualisadores externos e ferramentas de análise e visualização de dados podem ser usados com o NS-3, no entanto, seus usuários devem trabalhar com ferramentas de desenvolvimento de software em C++ e Python [12].

O NS-3 é escrito inteiramente em C++, com ligações Python opcionais, portanto, seus *scripts* podem ser escritos em C++ ou em Python. Arquivos de rastreamento de pacote são gerados para serem utilizados com outros utilitários analisadores. Além disso, é possível a execução de código de implementação (permitindo que os usuários executem o código de implementação real no simulador) e permite o uso de vários tipos de interfaces nos nós. Outro fator importante, é que seu suporte ao código de implementação admite um espectro muito amplo de modelos de alta fidelidade, como o encapsulamento de toda a pilha de redes Linux em um nó, usando a estrutura de Execução Direta de Código (Direct Code Execution, DCE) [12]. Ele oferece dois modos de temporização: tempo real onde todos os eventos da simulação são sincronizados com o tempo da máquina e tempo padrão onde todos os eventos da simulação ocorrem tão rápido quanto possível [12, 32].

2 VEÍCULOS NÃO TRIPULADOS

O protocolo proposto no presente trabalho é voltado para o suporte à comunicação entre embarcações de superfície não tripuladas e uma base fixa. Assim, o presente capítulo apresenta alguns conceitos sobre veículos não tripulados e mais detalhadamente sobre embarcações de superfície não tripuladas, conceitos necessários para o entendimento das soluções adotadas. O capítulo é subdividido da seguinte forma: a Seção 2.1 apresenta a definição e os tipos de veículos não tripulados, a Seção 2.2 apresenta os principais componentes de uma embarcação de superfície não tripulada; a Seção 2.3 apresenta a modelagem dinâmica de uma embarcação de superfície e qual foi a abordagem adotada; a modelagem do controle para o modelo dinâmico utilizado é apresentada na Seção 2.4.

2.1 Tipos de Veículos Não Tripulados

Os veículos não tripulados são veículos que dispensam o uso de tripulação, podendo necessitar ou não de intervenção humana. Eles são divididos em quatro categorias, de acordo com o ambiente no qual estão inseridos: veículo aéreo (*Unmanned Aerial Vehicle*, UAV), veículo terrestre (*Unmanned Ground Vehicle*, UGV), veículo/embarcação de superfície (*Unmanned Surface Vehicle/Vessel*, USV), veículo subaquático (*Unmanned Underwater Vehicle*, UUV) [1, 2, 44].

De acordo com o nível de intervenção humana requerido, os UVs também podem ser denominados de veículos de operação remota (maior grau de intervenção humana) ou autônomos (menor grau de intervenção humana), por exemplo: veículo aéreo autônomo (*Autonomous Aerial Vehicle*, AAV), veículo/embarcação de superfície autônomo (*Autonomous Surface Vehicle/Vessel*, ASV) ou (*Autonomous Surface Craft*, ASC), veículo subaquático autônomo (*Autonomous Underwater Vehicle*, AUV) e veículo de operação remota (*Remotely Operated Vehicle*, ROV) [1, 2, 44].

Apesar do termo autônomo ser frequentemente citado, ele não é utilizado com o significado restrito do termo. Refere-se apenas a veículos não tripulados, visto que estas plataformas, na maioria das vezes, são controladas remotamente por operadores em terra ou em embarcações, pelo menos naqueles momentos que antecedem ou sucedem a execução das tarefas que se propõem [45].

Devido às características ambientais em que os USVs estão inseridos (mares, rios e lagos) surgem inúmeros desafios nas áreas de controle de sistemas, comunicação e sistemas distribuídos [46]. As correntes e ondas marítimas, as propriedades aquáticas (refração, reflexão) e os ventos causam grandes problemas na comunicação e nos sistemas de guiagem, navegação e controle (*Guidance*,

Navigation and Control, GNC) dos USVs [1, 15, 46–48].

A robótica heterogênea é uma das operações de última geração que envolve atividades com vários tipos de veículos autônomos e de robótica, tais como: UUVs ou AUVs, USVs ou ASVs e UAVs. A ideia por trás dessas operações é explorar as capacidades específicas de cada veículo para completar os objetivos da missão de alto nível. Ainda assim, principalmente por causa da limitada resistência e variedade de sistemas, tais operações são geralmente realizadas nas proximidades de uma infraestrutura costeira ou de uma embarcação de apoio e apresentam tempo limitados. Essas operações em ambientes marítimos estão sendo demonstradas com cada vez mais frequência, resultantes de um esforço conjunto entre múltiplas instituições e campos de pesquisa interdisciplinares [2].

A operação autônoma de um veículo de superfície requer que a orientação, a navegação e o controle sejam executados com alta confiabilidade, tolerância a falhas e segurança. Inclui a percepção em tempo real do entorno do navio, a fim de evitar o aterramento e a colisão com outros navios, embarcações, pessoas, mamíferos marinhos ou outros obstáculos que possam ser encontrados. A fim de ser capaz de detectar a ampla gama de obstáculos em potencial e fornecer prevenção de colisão automática e reconhecimento de situação, sensores *on-board* como radar, detector de distância por variação de luz (*Light Detection and Ranging*, LIDAR) e câmera podem ser usados para escanear o ambiente do navio [2].

A Organização Marítima Internacional (*International Maritime Organisation*, IMO) define a navegação eletrônica como [2] “a coleta harmonizada, integração, troca, apresentação e análise de informações marítimas a bordo e em terra por meios eletrônicos para melhorar a navegação e atracação e serviços relacionados à proteção e segurança no mar e proteção do ambiente marinho”. As regras para evitar a colisão de navios são dadas pela Convenção nos Regulamentos Internacionais para a Prevenção de Colisões no Mar (*Convention on the International Regulations for Preventing Collisions at Sea*, COLREGS), segundo a IMO. Embora os COLREGS tenham sido feitos para navios operados por uma tripulação, os seus elementos-chave são também aplicáveis a sistemas de prevenção de colisão automática, como sistemas de apoio à decisão para a tripulação ou em navios não tripulados operados de forma autônoma ou remota [2].

Em uma implementação de um sistema autônomo, o COLREGS impõe implicitamente requisitos sobre as informações que devem ser fornecidas pelos sistemas de sensores e as ações corretas que devem ocorrer em situações perigosas. Espera-se que os navios (e navios autônomos em particular) carreguem um Sistema de Identificação Automática (*Automatic Identification System*, AIS)

transmitindo sinais de rádio contendo posição, velocidade e outras informações sobre o navio, que podem ser recebidos por outros navios e autoridades. Os COLREGS também exigem a comunicação por meio de sinais luminosos e sonoros, podendo ser ampliado no futuro por meio de protocolos de radiodifusão para apoiar operações de navios autônomos e não tripulados [2].

Uma embarcação tripulada hoje apresenta de 400 a milhares de sensores que, de uma forma ou de outra, estão enviando dados ou são usados em uma operação. O número de sensores não diminuirá quando a embarcação for autônoma e há a necessidade de enviar alguns dos dados para um centro de controle em terra, de modo que o estado das embarcações esteja sob controle [2].

Veículos de longa durabilidade, mas de baixa velocidade, são comumente usados para levantamentos persistentes e monitoramento ambiental de áreas selecionadas ou para cobrir grandes distâncias. Veículos de curta durabilidade, mas de alta velocidade, poderiam ser usados para ações de resposta rápida ou patrulhamento. Embarcações especializadas também são usadas para operações militares de alto risco, como caça submarina e caça de minas. Eles também podem ser usados como portadores ou suporte de outros tipos de veículos não tripulados (UAVs, AUVs) ou atuarem como repetidoras entre a comunicação no ar e a submersa, ou até mesmo como mulas de dados (servindo como ponte para a transmissão dos dados de outros veículos) [2].

2.2 Embarcações de Superfície Não Tripuladas

Como mostrado na Figura 1, o projeto de USVs envolve os seguintes componentes básicos: a estrutura (forma física da embarcação) podendo ser de casco simples (ex.: caiaque), duplo (catamarã) ou triplo (trimarã); sistema de potência (fonte de energia e motores) e propulsão (conjunto hélice, jato d'água ou vela e leme); sistema de coleta dados (sensores), sendo o sistema de posicionamento global (*Global Positioning System*, GPS) e o de unidade de medidas inerciais (*Inertial Measurement Unit*, IMU) os mais comuns (usados em conjunto com outros); sistema de comunicação (geralmente sem fio) para transmissão dos dados coletados, monitoramento e controle; estação base que recebe os dados coletados, monitora e controla a embarcação, podendo ser fixas ou móveis; sistema GNC que integra e utiliza os outros componentes, é a parte inteligente da embarcação [1].

O fluxo do sistema GNC é mostrado na Figura 2, podemos observar como cada subsistema interage uns com os outros. O subsistema de guiagem é responsável por gerar continuamente trajetórias suaves, viáveis e ótimas com base no subsistema de navegação, nas missões atribuídas, nas capacidades do veículo e nas condições do ambiente. Já o subsistema de navegação é responsável por

identificar o estado (posição, orientação, velocidade e aceleração) corrente e futuro da embarcação e do ambiente (correntes e ondas marítimas e velocidade do vento). O subsistema de controle utiliza as informações recebidas dos outros dois subsistemas para gerar os comandos necessários a serem executados pelos atuadores de forma a cumprir os objetivos [1].

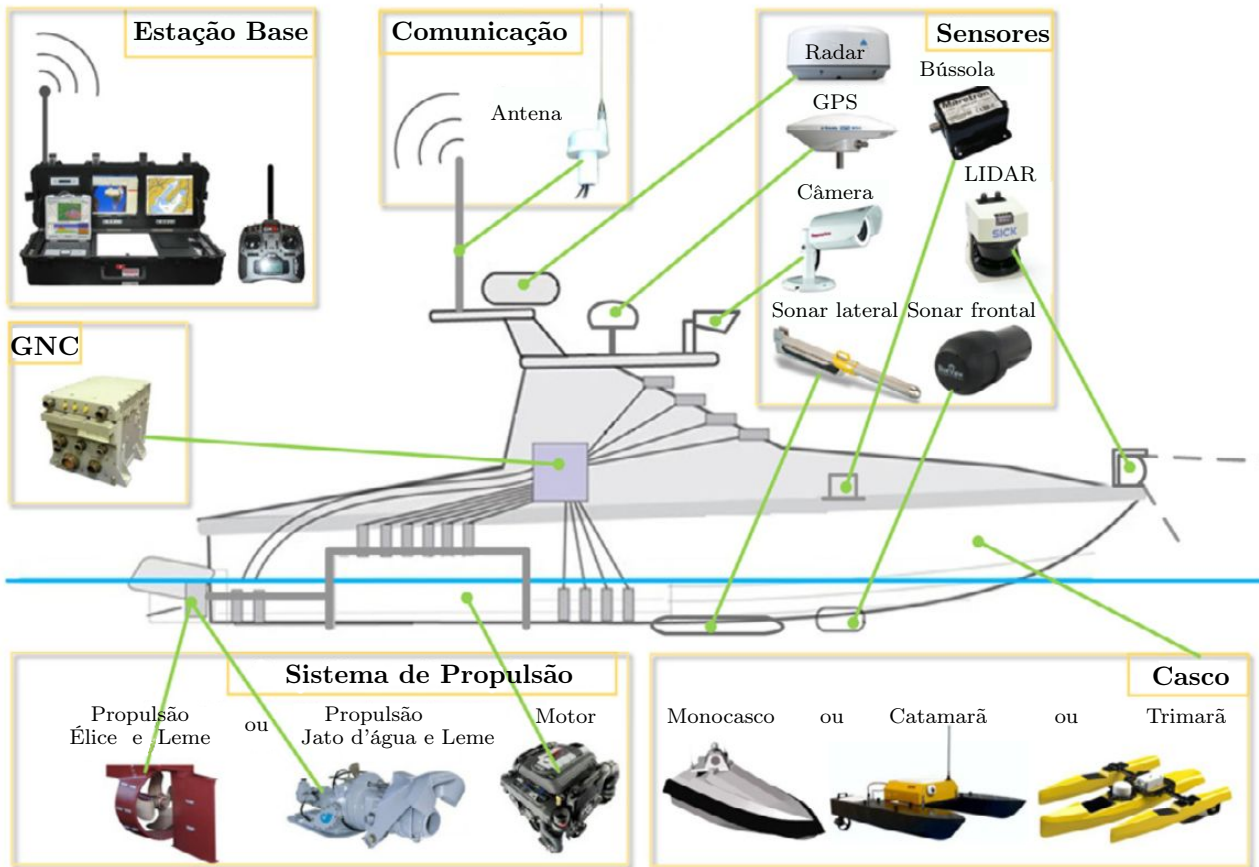


Figura 1 - Arquitetura Fundamental de um USV Típico. Traduzido de [1]

Um subsistema de guiagem (a Figura 3 mostra sua classificação de acordo com suas funções e métodos) viável é essencial para aumentar a autonomia do USV e quanto mais avançadas as capacidades de guiagem maiores e mais complicadas serão as restrições para o cumprimento das tarefas. O controle seguro e eficiente de USVs depende de um subsistema apropriado de navegação (a Figura 4 apresenta sua classificação de acordo com suas funções e métodos), com capacidade de detecção, estimativa de estado, percepção do ambiente e identificação de situações. O subsistema de controle (na Figura 5 é apresentada sua classificação em termos dos problemas e funções) gera de forma inteligente as saídas necessárias, que traduzem os dados recebidos dos subsistemas de guiagem e navegação, para controlar os atuadores do USV de forma a refletir o que foi planejado [1].

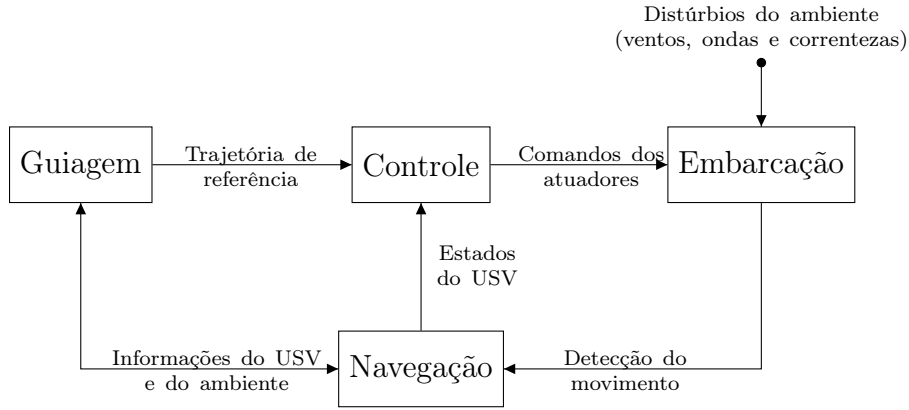


Figura 2 - Fluxo do Sistema GNC. Adaptado de [1]

O progresso dos USVs depende do desenvolvimento de sua autonomia, habilitando-os a trabalharem em qualquer ambiente não estruturado ou imprevisível sem supervisão humana. O desenvolvimento dessa autonomia é um grande desafio, pois exige o desenvolvimento de sistemas eficazes e confiáveis, incluindo comunicação, projeto adequado das embarcações e poderosas estratégias GNC [1,44]. Além disso, equipes de USVs podem ser utilizadas em missões de resgate, coleta de dados ambientais, rastreamento de alvos móveis, rastreamento de manchas de óleo e outros poluentes em ambientes marítimos [49]. Porém a coordenação e cooperação entre veículos, mecanismos para evitar colisões e/ou manter uma determinada formação dependem da capacidade de comunicação entre os USVs e uma possível base fixa [50].

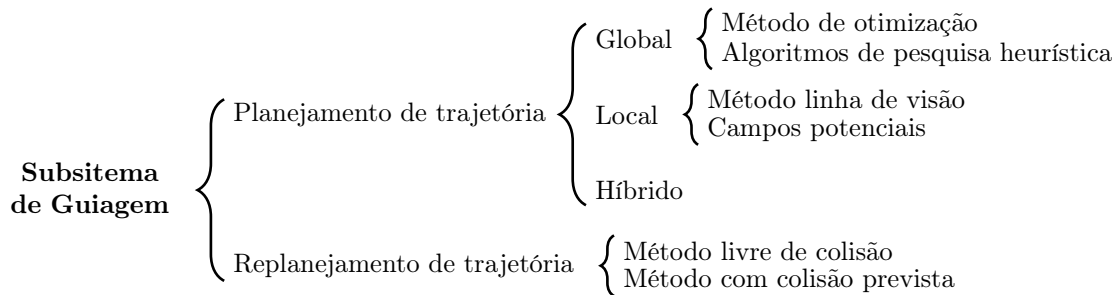


Figura 3 - Classificação dos Subsistemas de Guiagem. Traduzido de [1]

Em nosso trabalho integramos o sistema de comunicação ao sistema GNC, para que este incluísse as informações da comunicação em seus subsistemas. Também não nos preocupamos com o projeto completo da embarcação, pois nosso foco foi em apresentar uma solução que integrasse o sistema GNC com a comunicação e permitisse uma comunicação com uma estação base com o mínimo de qualidade possível, deste modo não nos preocupamos com algumas das características

apresentadas na Figura 1: tipo e forma do casco, sistemas de potência, propulsão e coleta de dados. Apesar dessas simplificações e das limitações impostas, trabalhos futuros podem incluí-las de forma gradual.

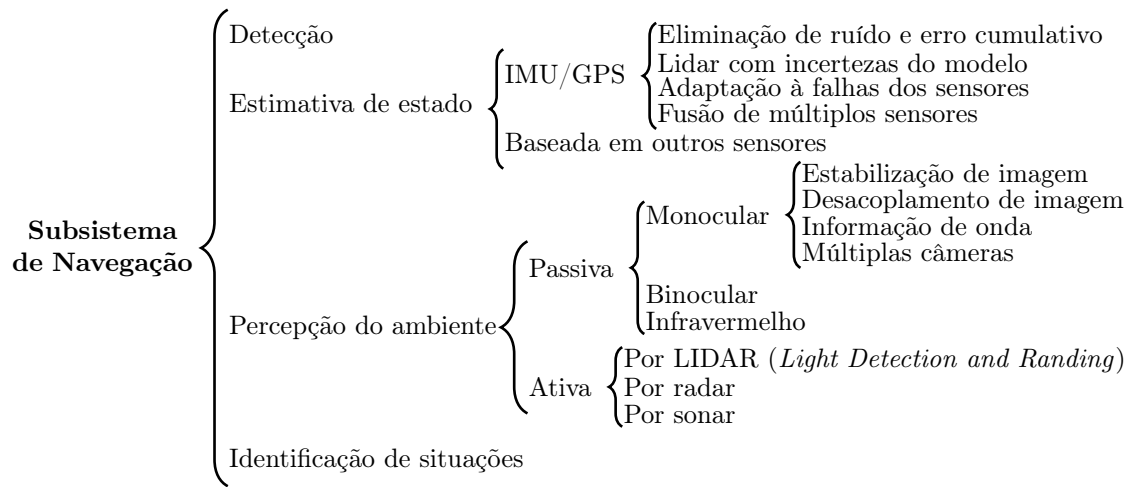


Figura 4 - Classificação dos Subsistemas de Navegação. Traduzido de [1]

Em nosso sistema GNC o subsistema de guiagem utiliza técnicas locais no planejamento de trajetórias usando as informações suas e de suas vizinhas, além dos comandos recebidos da estação base, tornando o planejamento híbrido (ver a Figura 3), o replanejamento é consequência das mudanças feitas no planejamento. Já no subsistema de navegação usamos apenas a estimativa baseada em GPS e IMU, porém não usamos as técnicas descritas na Figura 4, pois em nosso ambiente de simulação não incluímos efetivamente a simulação desses sensores e sim utilizamos os recursos do próprio simulador. E finalmente, nosso subsistema de controle apenas modela a cinética para o movimento das embarcações e ainda simplificamos o modelo assumindo que o ambiente é calmo e sem perturbações (correntezas, ondas, e ventos). Além disso, não levamos em consideração outros aspectos descritos na Figura 5 para simplificarmos nosso modelo.

A estação base desempenha um papel importante no sistema GNC, que pode estar localizada em uma instalação em terra ou em um veículo apresentando uma posição fixa ou móvel, atribuindo as missões aos USVs geralmente por meio de sistemas de comunicação sem fio. O estado em tempo real do USV e de seu equipamento embarcado são todos monitorados pela estação base e no caso dos ROVs também são enviados os comandos de controle [1]. Em nossa simulação a estação base apresenta posição fixa (assumimos uma base em terra), ela envia comandos para as embarcações a fim de controlar o fluxo de trabalho do conjunto de USVs e recebe os dados de estado das mesmas.

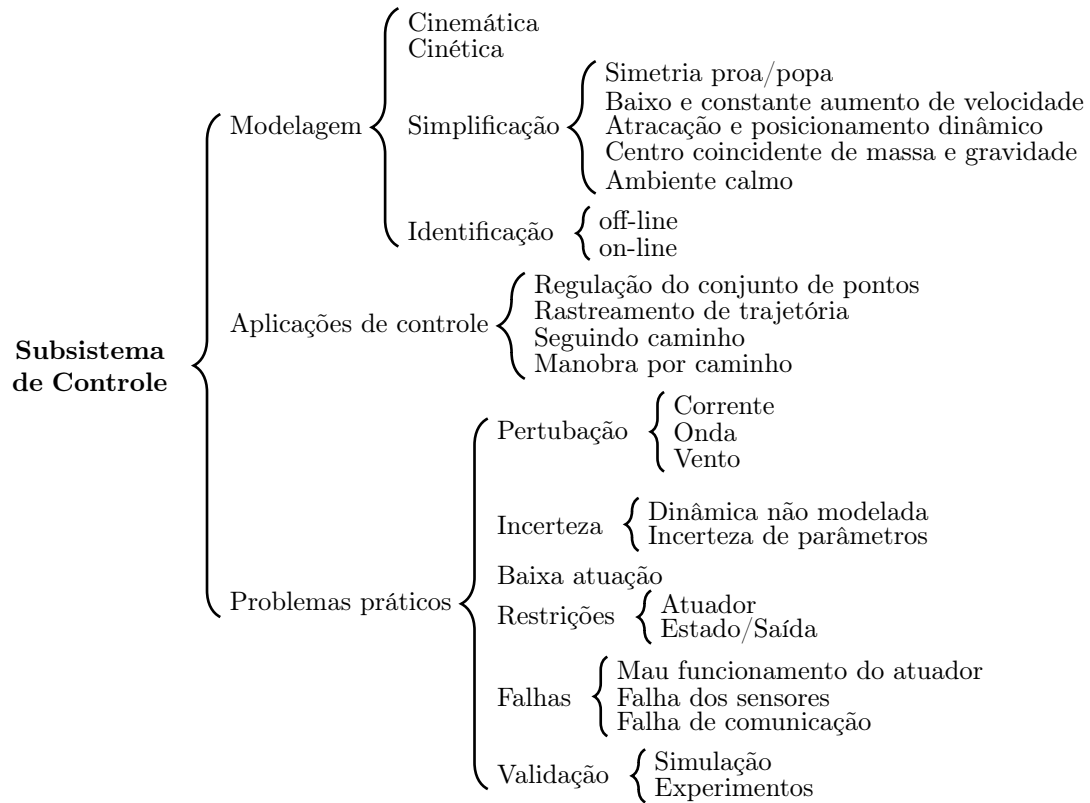


Figura 5 - Classificação dos Subsistemas de Controle. Traduzido de [1]

2.3 Modelo Dinâmico das Embarcações

A pressão que atua sobre a embarcação pode ser dividida em pressão hidrostática e hidrodinâmica, a primeira proporcional a submersão da embarcação e a segunda aproximadamente proporcional ao quadrado da velocidade relativa na água [51]. Além disso, é possível dividir as resistências em águas calmas em quatro grupos, tais como: resistência de viscosidade do fluido, resistência do ar, resistência das ondas e resistência de atrito e deslocamento de pulverização [52].

Por questões de simplificação não foram incluídas as modelagens referentes a ação dos ventos, ondas e correntezas. O modelo utilizado leva em consideração apenas a resistência de viscosidade do fluido que é proporcional a velocidade da embarcação, porém não foi utilizada a mesma modelagem de [52] para simplificar o modelo. Foi utilizado um modelo simples para representar a dinâmica das embarcações considerando dois graus de liberdade, um para cada coordenada do plano (x,y). As embarcações são representadas por um ponto sobre a superfície d'água sem levar em consideração o ângulo de rotação (as embarcações não apresentam ângulo de direção). O modelo dinâmico das

embarcações está representado no diagrama de blocos na Figura 6 para cada grau de liberdade.

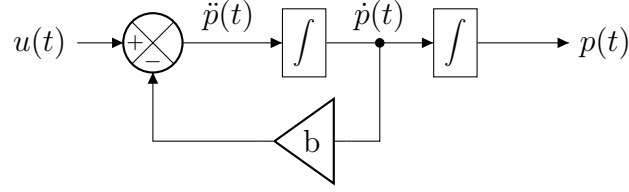


Figura 6 - Diagrama de blocos do modelo dinâmico das embarcações para cada grau de liberdade.

O diagrama de blocos na Figura 6 representa a seguinte equação dinâmica:

$$\ddot{p}(t) = u(t) - b\dot{p}(t), \quad (1)$$

na qual $p(t)$ é a posição, $\dot{p}(t)$ é a velocidade, $\ddot{p}(t)$ é a aceleração, $u(t)$ é o sinal de controle e b é o coeficiente de arraste. Resolvendo a equação diferencial (1) com $u(t)$ constante, obtemos a solução:

$$p(t) = \frac{(e^{-bt} - 1)u}{b^2} + \frac{(ut + 1 - e^{-bt})\dot{p}(0)}{b} + p(0), \quad (2)$$

na qual $p(0)$ é a posição inicial, $\dot{p}(0)$ é a velocidade inicial e u é o valor do sinal de controle.

2.4 Controle das Embarcações

Como o modelo dinâmico é um sistema inercial (dois integradores para cada grau de liberdade), foi utilizado um algoritmo PD (ação proporcional e derivativa) por ser o mais simples para utilização como controle desse tipo de sistema [53]. O diagrama de blocos na Figura 7 apresenta a malha fechada do sistema.

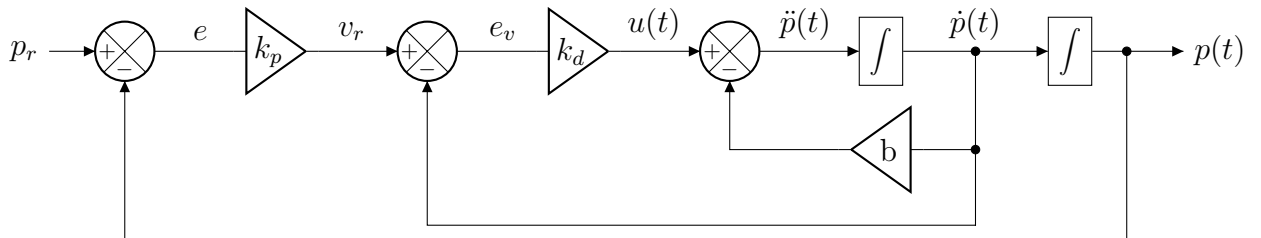


Figura 7 - Diagrama de blocos do controle PD de um grau de liberdade.

Baseado na análise de resposta transitória e de regime estacionário para sistemas de segunda ordem [54] com o diagrama de blocos mostrado na Figura 7 foi obtida a seguinte equação

característica:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = s^2 + (k_d + b)s + k_p k_d, \quad (3)$$

na qual ζ é o coeficiente de amortecimento do sistema, ω_n é a frequência natural não amortecida, k_p é o ganho proporcional e k_d é o ganho derivativo. Efetuando as devidas operações foram obtidos os seguintes valores para k_p e k_d em função de ζ e ω_n :

$$\begin{aligned} k_d &= 2\zeta\omega_n - b, \\ k_p &= \frac{\omega_n^2}{2\zeta\omega_n - b}. \end{aligned} \quad (4)$$

De acordo com a análise de resposta transitória e de regime estacionário para sistemas de segunda ordem e suas respostas ao degrau unitário [54], escolhe-se o máximo *overshoot* $M_p = 0\%$ e tempo de assentamento $t_s \leq 2s$, para uma faixa de tolerância de 2%, assim o coeficiente de amortecimento $\zeta = 1$ para uma resposta ao degrau livre de oscilações. Utilizando os gráficos $\zeta \times M_p$ e $\zeta \times t_s$ [54] foi encontrado $\omega_n = 3 \text{ rad/s}$. Substituindo os valores de ω_n e ζ em (4) foram obtidos os ganhos proporcional ($k_p = 2$) e derivativo ($k_d = 10$) do controlador, valores que geram polos reais negativos (sem oscilação).

Para a implementação do diagrama de blocos apresentado na Figura 7 foi necessário transformá-lo em um sistema de tempo discreto e para isso foi utilizada a função *zero-order-hold* [55], como mostra o diagrama de blocos na Figura 8.

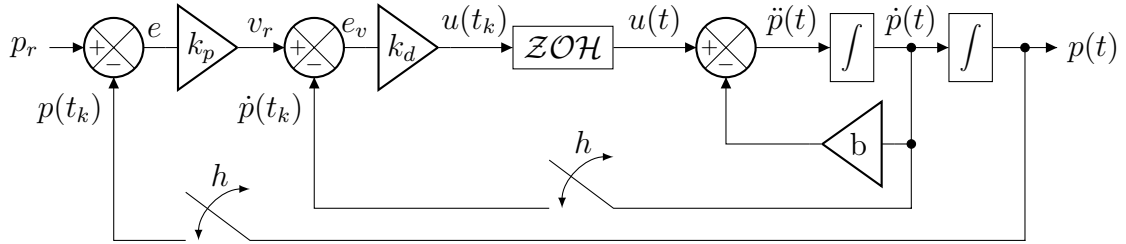


Figura 8 - Diagrama de blocos do controlador PD de um grau de liberdade e sinais amostrados.

De acordo com o diagrama de blocos na Figura 8 o sinal de controle é dado pela equação:

$$u(t_k) = k_d[v_r(t_k) - \dot{p}(t_{k-1})], \quad (5)$$

na qual $v_r(t_k)$ é a velocidade de referência no instante t_k e $\dot{p}(t_{k-1})$ é a velocidade no instante t_{k-1} .

A velocidade de referência ($v_r(t_k)$) na equação (5) é definida pela equação:

$$v_r(t_k) = k_p[p_r(t_k) - p(t_{k-1})], \quad (6)$$

na qual $p_r(t_k)$ é a posição de referência no instante t_k , $p(t_{k-1})$ é a posição no instante t_{k-1} . Enquanto que a velocidade $\dot{p}(t_{k-1})$ no instante t_{k-1} é definida pela equação:

$$\dot{p}(t_{k-1}) = \frac{p(t_{k-1}) - p(t_{k-2})}{h}, \quad (7)$$

na qual h é o período de amostragem definido pela equação:

$$h = t_{k-1} - t_{k-2}. \quad (8)$$

Para simular as limitações cinemáticas das embarcações foi acrescentado um saturador [55] como mostrado no diagrama de blocos na Figura 9, de forma a limitar a velocidade máxima permitida em cada dimensão (x,y).

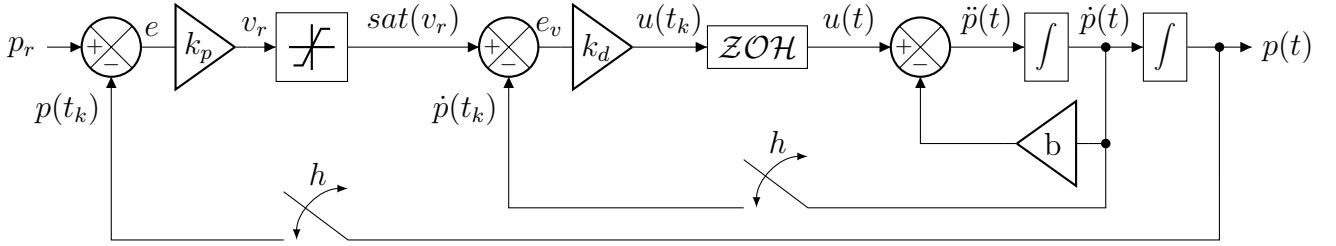


Figura 9 - Diagrama de blocos usando controlador PD com saturador e sinais amostrados.

O saturador é aplicado sobre a saída do ganho proporcional gerando a saída saturada definida pela equação:

$$sat(v_r(t_k)) = \begin{cases} \bar{v}_r, & \text{se } v_r(t_k) > \bar{v}_r \\ v_r(t_k), & \text{se } -\bar{v}_r < v_r(t_k) < \bar{v}_r \\ -\bar{v}_r, & \text{se } v_r(t_k) < -\bar{v}_r, \end{cases} \quad (9)$$

na qual \bar{v}_r é a velocidade máxima permitida em cada coordenada (x,y) e $v_r(t_k)$ é a velocidade de

referência no instante t_k . Com isso o sinal de controle (5) passa a ser representado pela equação:

$$u(t_k) = k_d [\text{sat}(v_r(t_k)) - \dot{p}(t_{k-1})], \quad (10)$$

que substituindo (10) em (2) é obtida a equação de posição:

$$p(t_k) = \frac{k_d [\text{sat}(v_r(t_k)) - \dot{p}(t_{k-1})] (e^{-bh} - 1)}{b^2} + \frac{[k_d (\text{sat}(v_r(t_k)) - \dot{p}(t_{k-1}))h + 1 - e^{-bh}] \dot{p}(t_{k-1})}{b} + p(t_{k-1}), \quad (11)$$

substituindo (10) na primeira derivada de (2) é obtida a equação de velocidade:

$$v(t_k) = \frac{k_d [\text{sat}(v_r(t_k)) - \dot{p}(t_{k-1})] (1 - e^{-bh})}{b} + \dot{p}(t_{k-1})e^{-bh}, \quad (12)$$

por fim a equação de aceleração é obtida substituindo (10) na segunda derivada de (2) resultando na equação:

$$a(t_k) = k_d [\text{sat}(v_r(t_k)) - \dot{p}(t_{k-1})] e^{-bh} - b\dot{p}(t_{k-1})e^{-bh}. \quad (13)$$

A simulação é executada em tempo discreto, por isso os cálculos de posição (11), velocidade (12) e aceleração (13) são efetuados em intervalos periódicos. Em cada execução do cálculo o sinal de controle é mantido constante e os valores de posição e velocidade no instante t_{k-1} são obtidos do resultado anterior (todos os cálculos são armazenados), exceto para $k = 0$ que teremos $h = 0$, $v(t_{k-1}) = 0$ e $p(t_{k-1})$ é a posição de partida de cada embarcação (as coordenadas da posição inicial são definidas antes de executar a simulação).

3 TRABALHOS CORRELATOS

O presente trabalho está inserido na linha de pesquisa do Projeto de Pesquisa “Embarcações Não Tripuladas para Monitoração Ambiental e Defesa” [46], que foi coordenado pelo Professor José Paulo Vilela Soares da Cunha. O projeto tinha por objetivo tornar factível a utilização de embarcações não tripuladas para fins de monitoração ambiental e defesa. Com ele foi possível adquirir equipamentos suficientes para a construção de protótipos para testes laboratoriais ou em águas calmas.

A primeira abordagem para esse Projeto de Pesquisa foi proposta em 2012, como projeto de mestrado, pela aluna Elaine de Mattos Silva Bezerra. A abordagem era centrada na lei de controle baseada em campos potenciais artificiais (Artificial Potential Field, APF) obtidas a partir de métricas de qualidade do canal de comunicação para controlar a formação de USVs e estudar os efeitos de uma topologia variante no tempo na estabilidade da formação. Resultados iniciais obtidos por meio de experimentos realizados no estacionamento do campus UERJ mostraram que o sinal captado se encaixa bem à distribuição de probabilidade de *Nakagami*, o que possibilitaria sua utilização para representar a qualidade do canal sem fio na definição da função potencial artificial [56].

Baseando-se na proposta [56] e em seus resultados, o presente trabalho apresenta objetivos que abrangem conhecimentos em diversas áreas, apresentadas no Capítulo 1, desta forma o presente capítulo descreve alguns dos trabalhos relacionados de acordo com seu tema principal, sendo dividido da seguinte forma: a Seção 3.1 apresenta algumas soluções para comunicação marítima e redes *ad hoc*, na Seção 3.2 apresentamos algumas pesquisa sobre embarcações não tripuladas e na Seção 3.3 descrevemos alguns trabalhos que apresentaram soluções que integram ambientes de simulação de rede com o ambiente de cálculo numérico MATLAB.

3.1 Comunicação Marítima e Redes *Ad Hoc*

Em [2] é apresentado um panorama geral dos diversos desafios encontrados ao se estudar o ambiente marítimo e como explorá-lo. É apresentado um modelo típico de comunicação marítima envolvendo diversos tipos de veículos (tripulados ou não) e equipamentos fixos de sensoriamento, no qual diversas tecnologias de comunicação podem e devem ser empregadas, dadas as diferentes característica dos agentes envolvidos (subaquáticos, de superfície e aéreos). As operações marítimas

são divididas de acordo com sua área de atuação, mostrando a importância de se ter uma boa infraestrutura de comunicação principalmente para as operações que envolvem a cooperação de múltiplos agentes (sejam eles do mesmo tipo ou não). A pesquisa também divide essas operações de acordo com seu grau de autonomia (desde totalmente manual até totalmente autônomo) e relata que esse grau de autonomia impacta diretamente nos requisitos de comunicação necessários para o funcionamento do sistema. Por fim, é apresentada uma série de tecnologias de comunicação, suas características principais e onde melhor poderiam ser empregadas.

Em [15] é investigado como as características marítimas influenciam na sensibilidade do sinal de comunicação e propõem um esquema para eliminar esses efeitos. Modelos de propagação e da antena são utilizados para eliminar os efeitos ondulatórios do mar. Esse esquema utiliza o protocolo de roteamento AODV com um algoritmo que utiliza a qualidade do sinal recebido como métrica auxiliar para a definição de rotas, descartando os pacotes RREQ e RREP que estejam fora da margem estabelecida. Com o trabalho realizado é possível verificar como as condições ambientais adversas de ambientes marítimos podem afetar consideravelmente a comunicação.

3.2 Veículos não Tripulados

No trabalho [57] encontramos uma ampla pesquisa sobre a utilização de USVs no controle e gerenciamento de desastres. Ele aponta que há poucas implantações de USVs em cenários de desastre, com a maioria das pesquisas na área concentrando-se nos aspectos tecnológicos do *hardware*, *software*, sistema GNC e não focando no gerenciamento de desastres. Por fim ele apresenta as soluções atuais encontradas na literatura e suas características principais.

No trabalho [58] foi pesquisado um método utilizando campos potenciais artificiais e o algoritmo A^* para dar ao USV maior autonomia no planejamento de rotas. A distribuição de probabilidade de colisão é baseada em dados históricos obtidos do AIS e descrita pelo método APF. Depois do cálculo da distribuição potencial do campo, a rota ideal (entre o ponto inicial e final) é resolvida pelo algoritmo A^* . É proposto também um algoritmo de suavização com base na análise de regressão, para distribuição de probabilidade baseado em rotas reais.

O trabalho [59] apresenta um algoritmo de formação coordenada para múltiplos USVs utilizando uma estrutura de guiagem e controle. O algoritmo propõe uma combinação da estratégia de líder virtual com o método de coordenação baseado em passividade [60] e utilização de uma lei de controle de consenso universal para todos os pontos de referência da formação desses USVs.

A teoria do controle híbrido é introduzida para garantir uma troca suave entre os controladores coordenados. O sistema de guiagem é introduzido de forma a evitar as desvantagens de replanejar o caminho de cada embarcação quando necessário.

Em [61] é apresentada uma embarcação modular adaptável às ondas (*Wave Adaptive Modular Vessel*, WAM-V) e sua manobrabilidade em águas calmas e profundas. Ela apresenta um casco do tipo catamarã que utiliza um grupo de modelos matemáticos de manobras para torná-lo uma embarcação marítima de alta velocidade com alto grau de controle e capacidade de manobra. A embarcação é projetada para adaptar-se a superfície da água, sendo equipada com molas, amortecedores e juntas esféricas que dão agilidade suficiente para a embarcação e amortece as tensões na estrutura e carga útil. Simulações e testes experimentais são realizados, observando que o módulo de software e hardware WAM-V é suficiente para ser usado em mais pesquisas de navegação e controle de embarcação.

Os autores de [61] em [62] introduziram um planejador de trajetória, que utiliza lógica *fuzzy* em seu controlador. O subsistema de navegação é composto principalmente por GPS e o de guiagem por dois blocos principais: o mais externo que utiliza lógica *fuzzy* no planejamento das trajetórias e o mais interno (subsistema de controle) que utiliza um controlador PID (Proporcional, Integral e Derivativo). Uma estação base foi utilizada para controlar os modos da embarcação e um modem sem fio ficou responsável por manter a comunicação entre a estação base e a embarcação. As simulações realizadas no MATLAB/Simulink e os experimentos realizados foram satisfatórios, sendo possível utilizá-lo no traçado de caminhos.

O trabalho [45] apresenta um breve panorama da utilização de plataformas não tripuladas na execução de pesquisa em ambientes submersos, descrevendo as tecnologias, principais incertezas, as vantagens e desvantagens de sua utilização. Algumas plataformas não tripuladas e suas principais características são apresentadas e uma tabela ordenada por ano resume estas plataformas. Por fim, são apresentados os resultados dos testes de campo realizados para avaliar a capacidade das ASCs na realização de levantamentos batimétricos, conduzidos em diferentes ambientes: rios de baixa correnteza, áreas extremamente rasas (aproximadamente um metro de profundidade), lagoas e represas, barragens de rejeitos de mineração, dentre outros.

Em [44] é proposto um projeto de um USV destinado ao monitoramento ambiental. O artigo detalha totalmente o projeto de um USV, desde o casco até a estratégia GNC, detalhando cada parte do projeto. O modelo proposto é simulado no Simulink mostrando que o mesmo permite ao navio se mover até a posição desejada. É possível observar que o artigo é um exemplo de como o processo

de modelagem é importante no processo de construção de um USV e que o uso de simuladores para avaliações preliminares se faz necessário antes de sua construção efetiva.

A pesquisa realizada em [1] aponta que o desenvolvimento de USVs totalmente autônomos em ambientes marítimos altamente dinâmicos permanece uma questão em aberto e há numerosos trabalhos de pesquisa em andamento sobre este tópico. Ela apresenta uma revisão abrangente da literatura sobre o progresso no desenvolvimento de USVs, fornecendo uma revisão técnica e uma lista bibliográfica sobre os desenvolvimentos históricos e contemporâneos dos sistemas GNC. São apresentadas as definições básicas de um USV e as metodologias adotadas são categorizadas e descritas. Também são apresentados os desafios para coordenação de múltiplos USVs, juntamente com algumas possíveis abordagens. Por fim, alguns desafios e direções futuras são apresentados, tornando a pesquisa uma fonte importante para o desenvolvimento de USVs realmente autônomos.

Na tese [63] é utilizada a embarcação apresentada em [61], porém utilizando um planejador de trajetória diferente do apresentado em [62]. O trabalho apresenta um planejador de trajetória para USVs operarem em ambiente marítimo complexo e dinâmico. É utilizado o algoritmo A^* com uma heurística para encontrar o menor custo até o destino, capaz de lidar com a dinâmica complexa dos obstáculos e funções de custos arbitrários. O planejador resultante permite ao USV operar sobre as condições adversas encontradas no mundo real, prevendo e calculando os riscos de colisão com embarcações civis e tomando as devidas ações para evitá-las. Além disso, ele leva em consideração os modelos de previsão do tempo e explora o fluxo das correntes do meio para auxiliar o movimento da embarcação e conservar energia.

No trabalho [64] é feito um estudo de como utilizar um USV para rastrear um alvo mantendo uma distância segura ao redor de um navio “mãe”. O USV se mantém dentro do raio de segurança ao redor do navio “mãe” enquanto o mesmo se desloca numa trajetória em linha reta. Os algoritmos de controle cinemático garantem que o USV, controlado por velocidade, convirja para um ponto alvo virtual dentro do raio de segurança do navio “mãe” sem colidir com a mesma. Este artigo permite observar técnicas para que USVs possam seguir uma trajetória de forma segura (sem colisão).

3.3 Integração entre Ambientes de Simulação

O trabalho [65] apresenta uma proposta para controle do desenvolvimento de aplicações usando a arquitetura de comunicação industrial como prova de conceito. É feita uma integração entre o protocolo de comunicação unificado e o MATLAB/Simulink usando arquivos MEX e S-

Function (recursos disponibilizados pelo MATLAB e Simulink, respectivamente, para a execução de códigos em outras linguagens). Ele mostrou que é possível estabelecer uma conexão entre um servidor e clientes MATLAB/Simulink, obtendo a integração necessária para controle e monitoramento de ambientes industriais.

Em [31] é feita a integração entre MATLAB e o OMNeT++. Esta integração é feita usando bibliotecas compartilhadas do MATLAB no OMNeT++, desta forma os problemas de sincronização e concorrência, encontrados em outras soluções, foram resolvidos. A integração do MATLAB com o OMNeT++ gerou um ambiente de simulação e validação integrados, o que geralmente necessita de ambientes separados. Para testar o ambiente proposto foi feito um teste simulando a tecnologia de localização usando RFID passivo em uma rede *ad hoc* hierárquica. A modelagem da camada física e o algoritmo de localização são simulados pelo MATLAB e o OMNeT++ simula os protocolos da rede. A solução proposta não apresenta problemas de sincronização, o que permite a depuração. Os dados obtidos do MATLAB podem ser usados para validar os resultados do OMNeT++.

No trabalho [32] encontramos uma solução para a integração do MATLAB com o NS-3. Nele é apresentada uma solução usando *sockets* TCP/IP para oferecer uma comunicação bidirecional entre os dois ambientes. Nesta solução o MATLAB envia instruções para serem executadas pelo NS-3, que após executá-las envia os resultados para o MATLAB e aguarda novas instruções. Nesse cenário o MATLAB é responsável pela dinâmica dos nós, enquanto que NS-3 é responsável pela execução dos protocolos da rede simulada. Em cada etapa da simulação ambos os ambientes trocam mensagens com as informações dos estados dos nós e assim é feita a sincronização entre os dois. Esta solução permite a execução da simulação de forma distribuída, já que a interação entre os ambientes é feita por meio de *socket*.

Integramos o NS-3 com o MATLAB como em [32], porém usando a API do MATLAB e não uma comunicação por *socket*. A API utilizada permite incluir a biblioteca do MATLAB em códigos C++, solução similar a utilizada em [31] só que sem a criação e exportação dos recursos necessários, pois o MATLAB já oferece todos os recursos necessários para sua instanciação e execução concorrentemente diretamente em nosso código em C++. Nossa solução dispensa a necessidade de personalização de código MATLAB e sua exportação para DLL/SO (como em [32]), pois a API já oferece uma interface que disponibiliza todos os recursos necessários para a execução dos códigos diretamente na instância do MATLAB. A solução apresentada em [65] não se adequou em nossa proposta, pois ela tem o intuito de executar códigos externos dentro do MATLAB e o que queríamos era o oposto. Assim como em [32], foi necessário criar uma interface de sincronização/comunicação

dentro do escopo do NS-3 para que os dois ambientes pudessem manter seus estados coerentes. Diferente de [32], em nossa solução o NS-3 controla a sincronização e não o MATLAB, além disso o MATLAB é utilizado apenas para cálculos independentes, pois ele recebe as informações para calcular de acordo com o tempo no NS-3.

4 PROTOCOLO DE COMUNICAÇÃO PARA FROTA DE EMBARCAÇÕES DE SUPERFÍCIE NÃO TRIPULADAS

Neste capítulo apresentamos o protocolo de comunicação desenvolvido para dar suporte à comunicação entre os elementos considerados: estação base e embarcações não tripuladas de superfície. O presente capítulo está organizado da seguinte forma: na Seção 4.1 são apresentados os detalhes do cenário abordado destacando seus elementos, a Seção 4.2 apresenta as características de comunicação consideradas, na Seção 4.3 é apresentado o protocolo de aplicação proposto.

4.1 Elementos Considerados

Os elementos considerados compreendem um conjunto de nós divididos em duas categorias: embarcações e estação base. Todos os nós fazem uso da pilha de protocolos TCP/IP (seguindo o recomendado em [2]) usando como meio de comunicação um canal sem fio. Os nós podem apresentar as seguintes funcionalidades: nós remetentes, nós destinatários e nós intermediários. Um nó remetente é aquele que gera uma mensagem e a envia pela rede, enquanto que o nó destinatário é o que irá receber uma mensagem da rede. Um nó intermediário é aquele que irá reenviar uma mensagem recebida da rede.

A topologia básica da rede considerada está representada na Figura 10. A **estação base** é o único nó fixo. Este apresenta apenas características de **nó remetente**, ao enviar informações de telemetria para as embarcações, e **nó destinatário**, quando recebe informações do estado das embarcações. A estação base não apresenta o comportamento de **nó intermediário**, pois a comunicação é feita entre a estação base e cada uma das embarcações, sendo assim ao receber uma mensagem a estação base não a reencaminha e sim envia uma nova mensagem (caso a mensagem recebida não já tenha sido recebida anteriormente). As **embarcações**, móveis, tem dupla funcionalidade: (i) enviam mensagens com informações do seu estado para a estação base (nó remetente) e recebem, da estação base, informações de telemetria (nó destinatário) e (ii) recebem e retransmitem mensagens de embarcações vizinhas (ou da estação base destinadas a outras embarcações), com o objetivo de encaminhar as mesmas para o seu destino final (nó intermediário).

Observa-se na Figura 10 que há nós que não apresentam um enlace de comunicação direto com a estação base (ou seja, não existe uma linha cheia direta destas embarcações para a estação base), sendo necessário o uso de *nós intermediários* para que consigam se comunicar com a mesma.

Este mecanismo é necessário para que as embarcações possam manter a comunicação com a base mesmo estando fora de seu alcance direto.

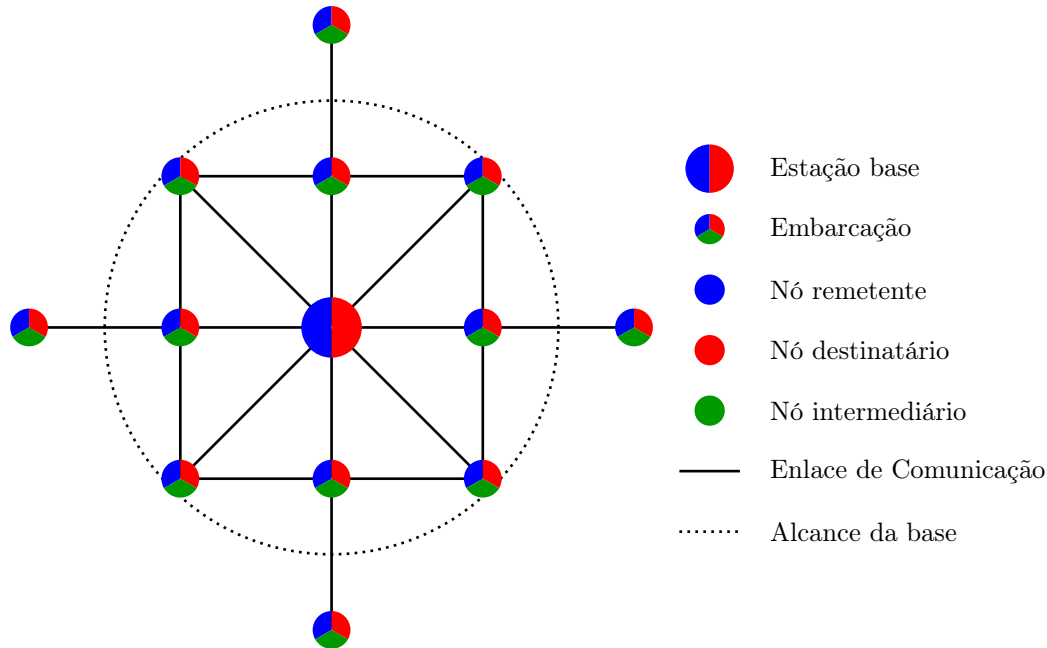


Figura 10 - Exemplo de Topologia Sem Nó Coordenador.

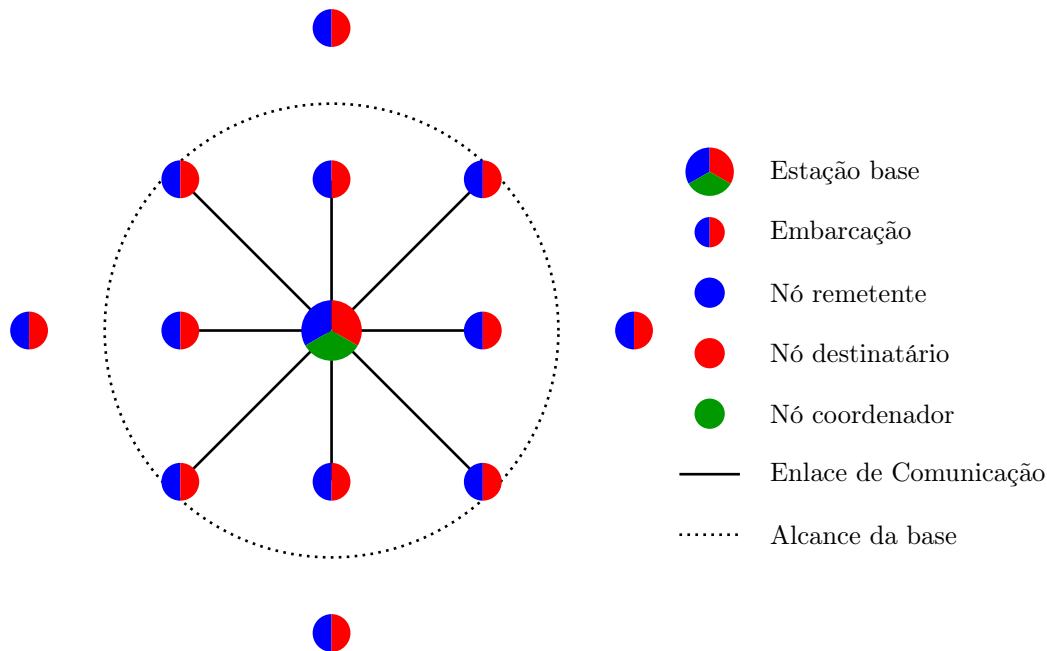


Figura 11 - Exemplo de Topologia Com Nó Coordenador.

Duas outras observações importantes se fazem necessárias. Como será detalhado adiante, a rede adotada não é estruturada. Assim não temos o papel de um Ponto de Acesso, a rede é *ad hoc*.

Isso significa que cada nó também incorpora o papel de roteador, o que facilita a funcionalidade de nó intermediário no protocolo proposto. A topologia com um Ponto de Acesso (ver Figura 11) não seria adequada para o cenário proposto, pois a mobilidade das embarcações poderia facilmente quebrar um enlace de comunicação isolando tal embarcação. A outra observação é sobre o papel da estação base no protocolo proposto. Ela é o nó de convergência de todas as mensagens das embarcações, mas não desempenha a função de nó coordenador. A comunicação é sempre feita de/para a estação base e as embarcações. Uma embarcação não envia uma mensagem diretamente para outra embarcação, por isso a estação base não precisa intermediar interações para encaminhar mensagens entre embarcações. Apesar de não está sendo usado, por ser uma rede *ad hoc* seria fácil utilizar a característica da rede para estabelecer uma comunicação entre duas embarcações não vizinhas (vizinhas diretas já se comunicam por padrão, mesmo que seja só para reencaminhamento das mensagens). Em uma comunicação com nó coordenador seria necessário mecanismos extras para a identificação das embarcações vizinhas e como toda a comunicação é controlada pelo nó coordenador esse mecanismo iria adicionar atraso para enviar e receber as mensagens necessárias para a identificação dos nós vizinhos, mensagens que necessariamente teriam que ser reencaminhadas pelo nó coordenador.

4.2 Características de Comunicação Consideradas

Como mencionado na Seção 1.1 diversas tecnologias poderiam ser empregadas como solução de comunicação para ambientes marítimos. Foi escolhida a comunicação sem fio definida pelo padrão IEEE 802.11 devido ao seu *hardware* apresentar baixo custo comercial, que torna possível a utilização de antenas com maior potência de transmissão para alcançar distâncias maiores [19]. Além disso, o protocolo proposto pode ser transposto para outras tecnologias.

Foi também escolhido o modo *ad hoc* das redes Wi-Fi em contraposição ao modo infraestruturado. Foi avaliado que o modo *ad hoc* seria o mais adequado para os cenários considerados (ver Figura 10), onde nós podem se comunicar sem a necessidade de um Ponto de Acesso (AP). Seu uso em ambientes marítimos permite que as embarcações possam entrar e sair da rede com facilidade, contribuindo também para a retransmissão e transporte dos dados, aumentando o raio de comunicação mesmo entre os elementos finais sem alcance suficiente para a comunicação entre eles [2].

A comunicação entre os elementos descritos na Seção 4.1 adota uma arquitetura geral com uma

pilha de protocolos segundo a Tabela 1, que também apresenta os respectivos tamanhos máximos em bytes de seus respectivos cabeçalhos. Como já informado, na camada (enlace/física) é utilizado o padrão IEEE 802.11 [26–30] em modo *ad hoc* [66]. A camada de rede é composta pelo protocolo IP [6] parte integrante da infraestrutura da *Internet* e pelo protocolo de roteamento *Ad hoc* Vetor Distância por Demanda (*Ad hoc On-Demand Distance Vector*, AODV) desenvolvido para uso em redes *ad hoc* com alto grau de mobilidade dos nós [67]. Na camada de transporte é utilizado o protocolo UDP [68], que apesar de não oferecer os mesmos serviços do TCP [7–11], atende ao propósito da proposta apresentando o mínimo necessário para a comunicação dos elementos considerados. A camada de aplicação será detalhada na Seção 4.3.

Tabela 1 - Protocolos utilizados

Camada TCP/IP	Protocolo (<i>bytes</i>)	
Aplicação	Aplicações Base/USV (2200)	
Transporte	UDP (cabeçalho 8)	
Rede	IP (cabeçalho 40)	AODV (cabeçalho 24)
Enlace/Física	802.11g (cabeçalho 34)	

4.2.1 *Ad hoc On-Demand Distance Vector* - AODV

Como já mencionado, as redes Wi-Fi em modo *ad hoc* necessitam de protocolo de roteamento específico para permitir que um nó atue no repasse de mensagens, quando não existe enlace direto entre os nós remetente e destinatário, e para se adaptar à constante mobilidade dos nós (no caso, as embarcações). Os protocolos de roteamento disponíveis para redes *ad hoc* são dinâmicos e podem ser classificados em: proativos — quando orientados a tabela, onde tabelas guardam a informação de roteamento para cada nó da rede; e os reativos — quando agem sobre demanda, não mantendo informações de roteamento até serem solicitados [66,69]. Prevendo a mobilidade das embarcações e a necessidade de uma comunicação com o menor *overhead* possível, para diminuir possíveis atrasos de comunicação causados pelo protocolo de roteamento, foi usado então o AODV que apresenta baixo *overhead* em redes com alto grau de mobilidade dos nós [67,70]. A configuração automática das redes *ad hoc* e o baixo *overhead* do AODV são recursos relevantes e até essenciais em ambientes marítimos [2].

O algoritmo AODV permite o roteamento dinâmico, auto iniciável, entre os nós móveis de uma rede *ad hoc*. Ele mantém rotas apenas para destinos ativos na rede, permitindo que os nós

respondam em tempo hábil a quebras de enlace e mudanças de topologia (rotas). Oferece uma rápida convergência diante de mudanças na topologia da rede, geralmente causada pelo movimento dos nós. A quebra de um enlace faz com que o conjunto de nós afetado seja notificado para que estes possam invalidar as rotas para os enlaces perdidos [67].

Ele utiliza tabelas simples de roteamento, ou seja, é armazenado apenas o próximo salto para o nó destino em questão. Por ser um protocolo reativo, a tabela de roteamento é construída quando uma rota para um destino é solicitada e a mesma não consta em sua tabela de roteamento, então é dado início ao processo de descoberta de rotas. O processo de descoberta envia pacotes de requisição de rota (*Route Request*, RREQ) para seus vizinhos e seus vizinhos os repassam até que o destino seja alcançado ou um nó intermediário que conheça uma rota para o destino seja encontrado. O nó destino ou intermediário (que conheça a rota) envia um pacote de resposta de rota (*Route Reply*, RREP) de volta até o nó fonte, com isso as tabelas de roteamento dos nós da rota são atualizadas [67].

Outro ponto importante do protocolo é a manutenção das rotas, que tem por objetivo validar as rotas mantidas em cache. A manutenção de rotas é feita enviando periodicamente pacotes *HELLO* entre os nós para validar as rotas e verificar a ruptura de ligações. Uma ruptura é inferida quando um nó não recebe pacotes *HELLO* de um ou mais nós pertencentes a rotas estabelecidas, então o nó que detectou a ruptura envia pacotes de erro de rota (*Route Error*, RERR) para os nós vizinhos que possivelmente usariam essa rota. Para saber quais os possíveis nós vizinhos de uma rota, cada nó mantém uma “lista de precursores” para cada rota em sua tabela de roteamento, essa lista é preenchida durante o recebimento do pacote RREP que por definição deve ser enviado para um nó em uma lista de precursores [67].

4.2.2 Difusão (*Broadcast*)

Alguns tipos de comunicação são conhecidos, cada um deles com suas características e aplicações, na Figura 12 são apresentados três dos mais comuns. A Figura 12(a) mostrada a comunicação direta ou *unicast*, nela o nó remetente conhece o endereço ou identificação do nó no qual deseja se comunicar e ao enviar uma mensagem colocar essa identificação para que a tecnologia de comunicação entregue a mensagem diretamente ao destino desejado, basicamente neste tipo de comunicação há sempre um nó remetente e um nó destinatário. A Figura 12(b) mostra a comunicação em grupo ou *multicast*, nesse tipo de comunicação há um nó remetente que deseja enviar uma mensagem para

um grupo de destinos, o nó remetente precisa saber apenas o endereço ou identificação do grupo e a tecnologia de comunicação fica responsável por entregar a mensagem para cada nó pertencente ao grupo (excluindo-se o nó remetente), ou seja há um nó remetente e muitos nós destinatários (a quantidade vai depender de quantos nós fazem parte do grupo). Finalmente, na comunicação por difusão ou *broadcast* mostrada na Figura 12(c), um nó remetente precisa enviar a mensagem com um endereço ou identificação especial que faz com que todos os nós da rede sejam destinatário da mensagem, com isso o nó remetente consegue se comunicar com todos os nós da rede enviando uma única mensagem na rede e deixa a cargo da tecnologia de comunicação entregar uma cópia para cada nó na rede.

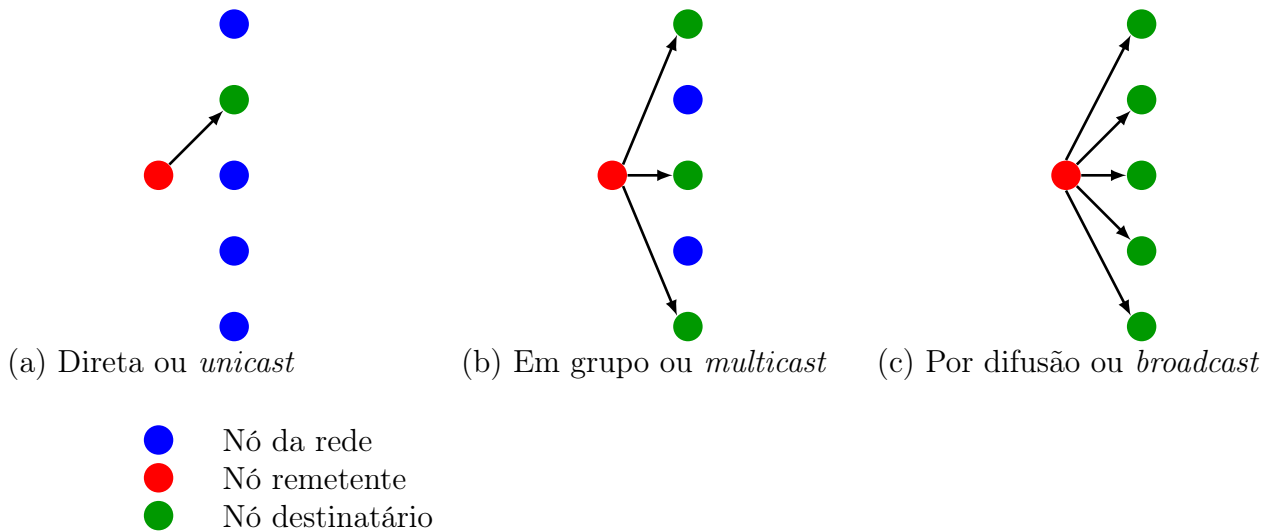


Figura 12 - Tipos de Comunicação.

A arquitetura de comunicação adotada permite a troca de mensagens entre os elementos. Entretanto seria necessária a identificação do endereço de rede, ou seja, o número IP, da estação base e de cada uma das embarcações. A estação base poderia ser identificada com um endereço fixo e de alguma forma as embarcações poderiam comunicar-se com a estação base, uma vez que este endereço fixo fosse conhecido. Entretanto, no protocolo proposto é importante tornar viável a comunicação 1-N (de um para muitos, por exemplo: da estação base para todas as embarcações) e aproveitar a característica omnidirecional (sinal enviado para todas as direções) do canal sem fio [25] para permitir a “percepção” das embarcações vizinhas, fornecendo informações úteis para sua tomada de decisão (por exemplo: evitar colisão entre as embarcações).

Neste contexto, foi decidido utilizar o mecanismo de *broadcast*, Figura 12(c), característico de canais sem fio, onde todas as mensagens enviadas pelo canal podem ser, potencialmente, recebidas

por todos os nós que estejam ao alcance da transmissão. Assim, a solução adotada utiliza para toda troca de mensagem do protocolo proposto o endereço de difusão *broadcast* IP, que, por sua vez, é mapeado em mensagens *broadcast* no enlace Wi-Fi, e não o endereço específico do receptor (*unicast*) [19].

Na busca de uma solução para que uma embarcação identificasse suas vizinhas, foi avaliado que com a comunicação *unicast* as embarcações teriam que enviar periodicamente mensagens de “varredura” para detectar suas vizinhas, o que tornaria a comunicação dispendiosa para cada embarcação descobrir e manter o endereço de cada embarcação vizinha como exigido pela comunicação *unicast* (Figura 12(a)). Mesmo com o protocolo AODV resolvendo as rotas entre as embarcações e a base, a rede iria ter que comportar não só as mensagens do protocolo de roteamento mas também as mensagens de varredura para a descoberta de vizinhança de cada embarcação, para que a tomada de decisão de cada embarcação pudesse levar em consideração as informações das embarcações próximas. Com o uso de mensagens *broadcast* (Figura 12(c)) foi possível reduzir o número de mensagens e facilitar a identificação dos nós vizinhos quando dentro do alcance de *broadcast* de um nó remetente.

Porém a técnica de difusão gera pacotes demais e consome largura de banda em excesso [25], pois mensagens de *broadcast*, Figura 13(a), são repassadas pelos nós intermediários, Figura 13(b), fazendo com que a rede seja inundada por mensagens repetidas, até que todos os nós tenham recebido, Figura 13(c). Assim, a estratégia adotada para contornar este problema foi usar o campo “Tempo de Vida” (*Time To Live*, TTL) do protocolo IP para limitar a difusão das mensagens em um salto, Figura 13(d) e (e). Este campo conta os saltos entre o transmissor e o receptor, sendo decrementado a cada salto. Quando chega à zero, o pacote é descartado e um pacote de advertência é enviado ao transmissor de origem. Esse recurso evita que os datagramas IP fiquem vagando indefinidamente pela rede, algo que aconteceria se as tabelas de roteamento fossem danificadas [19] ou no caso de falha de nós.

A Figura 13 mostra a diferença no envio, recebimento e reencaminhamento de mensagens usando a somente comunicação por *broadcast*: a Figura 13(a) mostra um nó enviando uma mensagem em difusão e os nós dentro do alcance da comunicação recebendo a mensagem; a Figura 13(b) mostra os nós que receberam anteriormente a mensagem reenviando-a e os nós dentro do alcance da comunicação recebendo-a; por fim, a Figura 13(c) mostra o fim do reencaminhamento e todos os nós que receberam a mensagem. Usando a comunicação por *broadcast* juntamente com o campo TTL do protocolo IP definido com o valor de um: a Figura 13(d) mostra um nó enviando uma

mensagem em difusão e os nós dentro do alcance da comunicação recebendo a mensagem; a Figura 13(e) mostra o fim do reencaminhamento e todos os nós que receberam a mensagem, diferentemente da comunicação somente com o *broadcast*, somente os nós dentro do alcance da comunicação do nó que iniciou o envio recebem a mensagem.

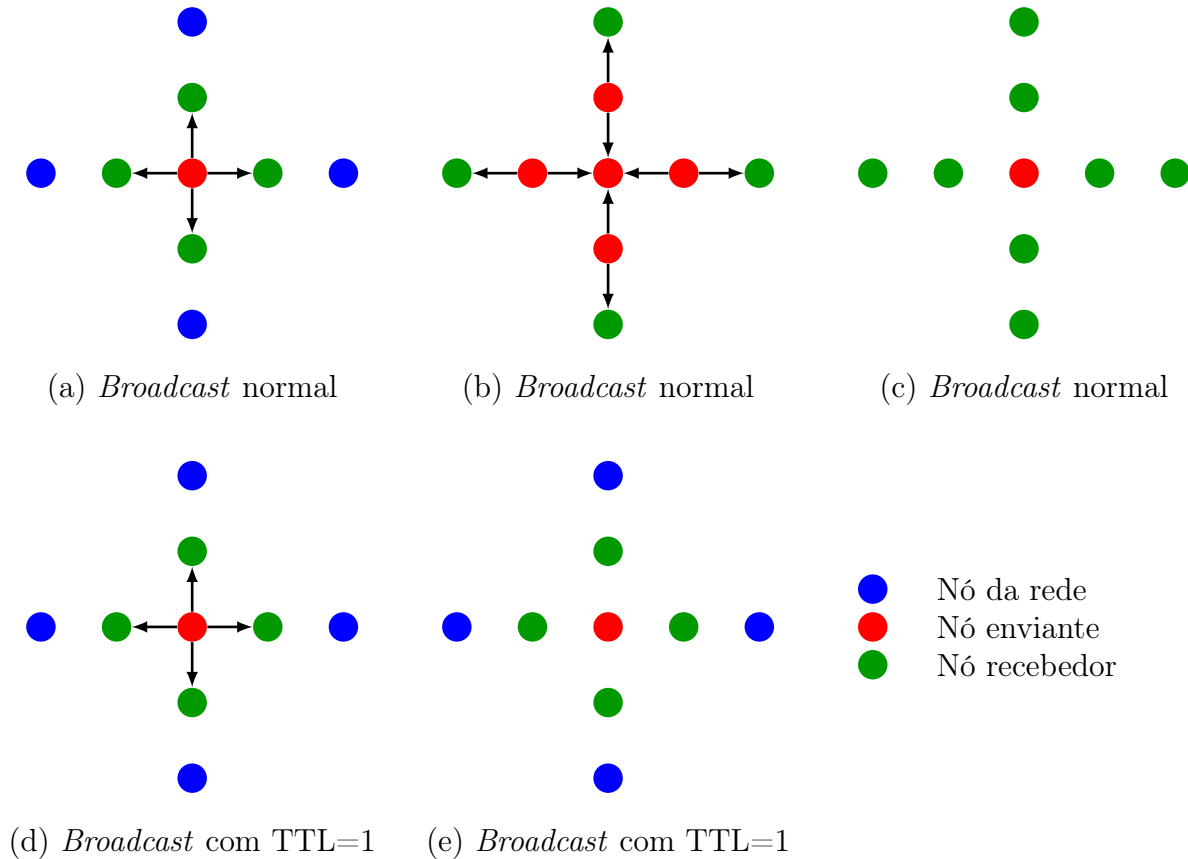


Figura 13 - Comunicação por Difusão (*Broadcast*).

Para garantir que somente os nós dentro do alcance da comunicação recebam a mensagem sem o reencaminhamento automático do protocolo de roteamento, Figura 13(e), o campo TTL é definido com o valor de 1 (**um**) antes de cada transmissão, com isso conseguiu-se:

- evitar a inundação da rede com mensagens de *broadcast*;
- estabelecer um mecanismo de comunicação de 1-N (de um para muitos) entre embarcações vizinhas, dado que cada mensagem enviada é interceptada pelos nós dentro do alcance;
- utilizar este mecanismo como informação de vizinhança das embarcações, vinculando a distância máxima de um enlace Wi-Fi com a distância entre as embarcações.

Observa-se ainda que, com isso, cada embarcação na rede consegue receber as informações de estado de suas vizinhas sem a necessidade do envio de mensagens específicas. Ou seja, cada embarcação envia periodicamente uma mensagem com seu estado para a estação base por *broadcast*. No processo de encaminhamento esta mensagem é interceptada e utilizada pelas embarcações vizinhas, nós intermediários ou não, ainda que o destino final seja a estação base. A estação base pode receber a mensagem diretamente, caso esteja no raio de comunicação, ela sabe que toda mensagem é direcionada a ela exceto as que ela mesma envia. Entretanto, para que a estação base possa receber as mensagens das embarcações localizadas a mais de um salto de distância o protocolo detalhado na Seção 4.3 avalia se as mensagens recebidas devem ser descartadas ou se devem ser reencaminhadas para que cheguem até a estação base.

A comunicação por *broadcast* limitada em um salto juntamente com o protocolo de roteamento possibilitou ao protocolo desenvolvido adicionar o recurso de agregação dos pacotes de telemetria (Seção 4.3.1.2) reencaminhados pela rede. Como todo nó dentro do alcance da comunicação decide o que fazer com o pacote de telemetria recebido, Seção 4.3, decidiu-se aproveitar para que com apenas um pacote de mensagem (Seção 4.3.1) diversos pacotes de telemetria possam ser enviados. A Figura 14 mostra como é feita a agregação dos pacotes de telemetria: a Figura 14(a) mostra uma embarcação (no caso a mais distante) enviando uma mensagem e suas duas vizinhas recebendo; a Figura 14(b) mostra as duas embarcações vizinhas da primeira enviando uma mensagem com seu pacote de telemetria agregado ao pacote de telemetria anteriormente recebido (reenviam o pacote de telemetria recebido anteriormente); a Figura 14(c) mostra a estação base recebendo as mensagens com os pacotes de telemetria de todas as embarcações; a Figura 14(d) mostra a estação base enviando a confirmação para todas as embarcações; a Figura 14(e) mostra as duas embarcações vizinhas da estação enviando uma mensagem com os pacotes de telemetria recebidos da estação base e seu pacote de telemetria atualizado; por fim, a Figura 14(f) mostra a embarcação que enviou a primeira mensagem recebendo a confirmação da estação base por meio das suas vizinhas.

Em resumo, combinamos redes Wi-Fi em modo *ad hoc*, com protocolo de roteamento AODV e mensagens *broadcast* IP limitadas em 1 salto como solução de comunicação do protocolo proposto.

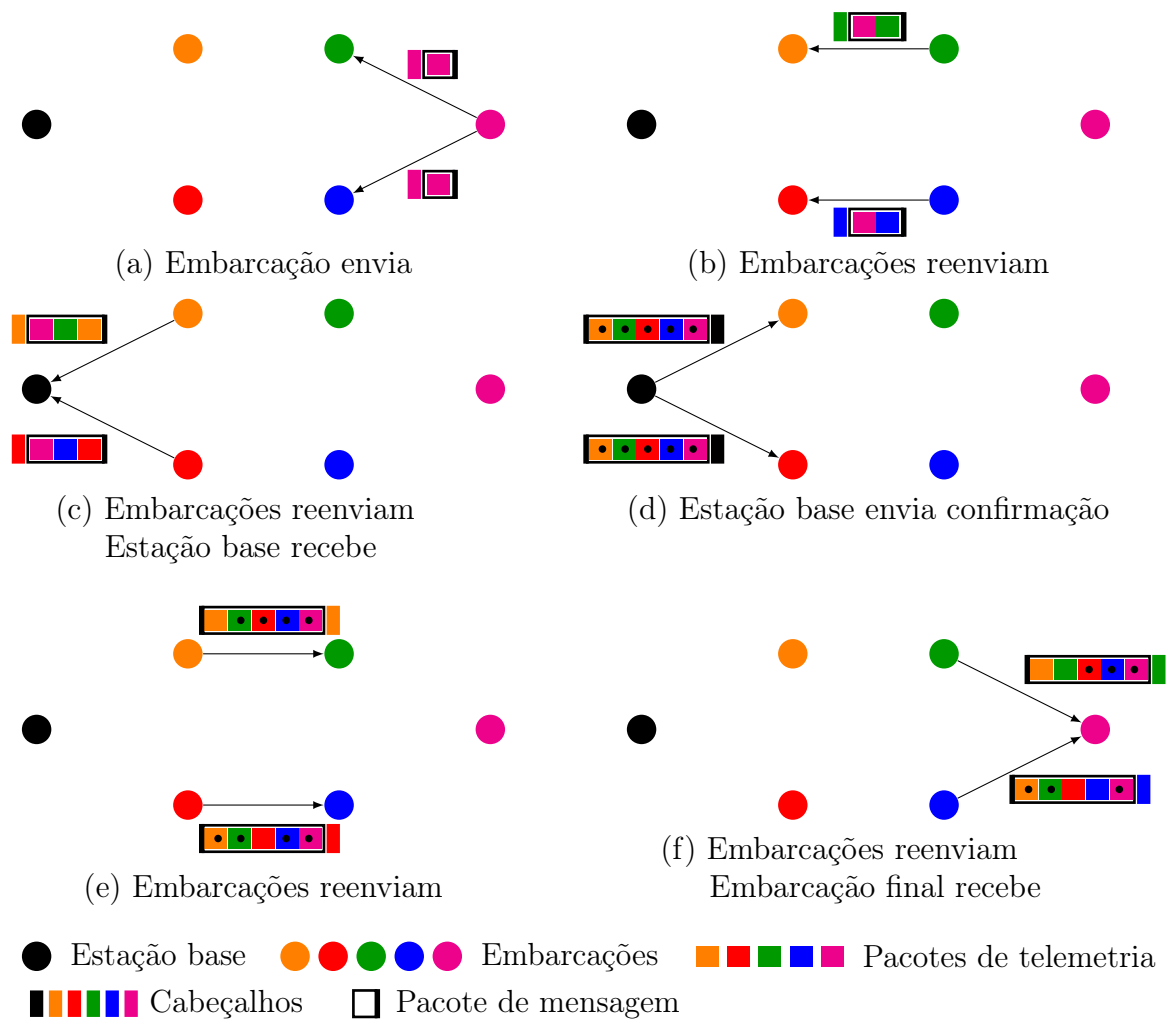


Figura 14 - Agregação dos Pacotes de Telemetria (4.3.1.2).

4.2.3 Parâmetros de Enlace e Tamanho dos Campos

Uma outra decisão de projeto para o protocolo proposto é relacionada a parâmetros dos mecanismos de enlace e aos tamanhos das mensagens. O objetivo foi minimizar a ocorrência de fragmentações e remontagem de PDUs nas várias camadas.

O tamanho máximo de uma mensagem do protocolo proposto foi fixado em 2200 bytes. Este tamanho é suficiente para o envio de todos os dados necessários em qualquer estado do protocolo. Acrescidas as informações de controle das outras camadas de comunicação, o tamanho final do quadro é compatível com o tamanho máximo de um quadro definido pela unidade de dados do protocolo de controle de acesso ao meio (*Medium Access Control*, MAC) (*MAC Protocol Data Unit*, MPDU) do padrão IEEE 802.11, de 2312bytes [26–30]. Desta forma, evita-se a fragmentação das

mensagens, e a introdução de atrasos desnecessários. Assim, como as mensagens (chamadas de datagramas pelo protocolo UDP) também podem ser fragmentadas pelo protocolo IP, foi necessário definir a unidade máxima de transmissão (*Maximum Transmission Unit*, MTU) em 2300 *bytes* ao invés do padrão comumente usado de 1500 *bytes* [6, 71] para que pudesse ser usado o máximo permitido pelo IEEE 802.11 (2312 *bytes*).

Como somente os quadros *unicast* são fragmentadas pela camada de enlace/física, ao contrário dos quadros *broadcast/multicast*, a comunicação em *broadcast* junto com as definições de MTU, MPDU e tamanho máximo das mensagens de aplicação permitem que toda a troca de mensagens sempre transporte todas as informações necessárias para o funcionamento do protocolo de aplicação proposto.

Também, foram acrescentados no protocolo proposto a detecção de erros simples por meio de uma soma de verificação (*checksum*) e um serviço de entrega confiável de dados, apresentados na Seção 4.3.

4.3 Protocolo Proposto

As características definidas na Seção 4.2, a estrutura das mensagens (Tabela 2), a estrutura dos dados de telemetria (Tabela 3), e a máquina de estados do protocolo proposto definem a infraestrutura necessária para que os elementos descritos na Seção 4.1 possam se comunicar, considerando os cenários definidos na Figura 10.

O protocolo proposto considera que em cada nó, embarcações e estação base, a pilha de protocolo descrita na Seção 4.2 está executando. Os nós executam aplicações que contém módulos de software implementando as funcionalidades de nó remetente, nó destinatário e nó intermediário do protocolo proposto.

Cada nó possui uma identificação única de 8 bits. Esta identificação é utilizada para registrar o remetente direto da mensagem. Ou seja, o último nó que encaminhou a mesma. Também é usado para identificar o nó que originou cada conjunto de dados de telemetria sendo transportados pela mensagem.

Após se registrar junto à estação base, cada embarcação recebe da estação base uma sequência de comandos para mudar seu comportamento (seu estado). Esta foi a forma simples para incluir o conceito de missão na proposta. Observa-se que a generalidade e o aprofundamento da semântica de uma missão não é objetivo do modelo. Num caso mais geral, pressupõe-se que a estação base

interpreta propostas de missões complexas e as traduz em estruturas simples para serem enviadas para cada embarcação. Este controle de missões complexas não está no escopo do trabalho.

Com a missão iniciada, cada embarcação executa rotinas de *software* para (i) acionar os sistemas de propulsão & navegação junto com os mecanismos de controle; (ii) enviar periodicamente para a estação base as informações de localização e dados sensorizados, bem como receber novas informações de missão (por exemplo, fim da missão ou para retornar); (iii) receber periodicamente informações de outras embarcações com os objetivos de (a) encaminhar as mensagens para a estação base, tornando a comunicação mais robusta e (b) identificar a vizinhança e eventualmente controlar a proximidade e o afastamento evitando colisões ou perda de comunicação.

Na maior parte do tempo em um determinado estado cada embarcação periodicamente coleta seus dados de telemetria e os envia para a estação base comportando-se como nó remetente. Além disso, cada embarcação está preparada para receber mensagens, que podem vir da estação base ou de outra embarcação, comportando-se como nó destinatário ou intermediário respectivamente. Ao receber uma mensagem a embarcação inspeciona a identificação dos pacotes de telemetria contidos na mensagem e decide se deve haver uma mudança no seu estado, caso encontre um pacote de telemetria com a sua identificação, e/ou se deve se comportar como nó intermediário, repassando a mensagem, para todos os pacotes de telemetria com identificações diferentes da sua. Além disso, ela também persiste os pacotes de telemetria das embarcações remetentes, ou seja, as embarcações as quais são suas vizinhas diretas.

A estação base, por sua vez tem a missão carregada numa etapa de inicialização do protocolo. Depois disso, espera mensagens de registro das embarcações. Conforme a estação base recebe as mensagens com os pacotes de telemetria das embarcações ela envia o código da missão a ser executada e qual o estado que as embarcações devem apresentar, estado este que é mudado conforme a estação base percebe a necessidade e possibilidade de se prosseguir para o próximo estado. A cada mensagem recebida os dados de telemetria são inspecionados e é avaliado se todas as embarcações podem transitar para o próximo estado. Neste caso, uma mensagem com comando de transição é enviada. Caso contrário, a estação base repassa para todas as embarcações os dados de telemetria mais atualizados recebidos de todas as embarcações, como parte da solução de controle de mensagens perdidas.

As próximas seções aprofundam os pontos mencionados.

4.3.1 Estrutura das Mensagens

Duas estruturas foram definidas:

- a estrutura da mensagem, que transporta dados de telemetria de cada embarcação, que chamamos de **pacote de mensagem**;
- a estrutura de dados de telemetria, que chamamos de **pacote de telemetria**.

4.3.1.1 Pacote de Mensagem

O pacote de mensagem, Tabela 2, é dividido em três partes: cabeçalho, pacotes de telemetria e um inteiro de quatro *bytes* para verificação dos pacotes de telemetria (soma de verificação usada para detectar pequenos erros nos pacotes de telemetria). O cabeçalho do pacote, Tabela 2, apresenta os campos necessários para que o transmissor possa ser identificado e o pacote possa ser validado: os dois primeiros *bytes* indicam o tamanho total do pacote em bytes; os dois *bytes* seguintes identificam quem enviou o pacote; em seguida quatro *bytes* armazenam um número de sequência para verificar pacotes repetidos; temos então um *byte* para indicar a altura da antena de transmissão e outro para informar a potência de transmissão; mais dois *bytes* indicam o número de pacotes de telemetria das embarcações que estão inseridos no pacote; finalizando com um inteiro de quatro *bytes* para a detecção de erros do cabeçalho do pacote (soma de verificação).

Tabela 2 - Estrutura do Pacote das Mensagens do Protocolo.

Cabeçalho da Mensagem (16 <i>bytes</i>)		
Tamanho total (2 <i>bytes</i>)		ID do transmissor (2 <i>bytes</i>)
Número de sequência do transmissor (4 <i>bytes</i>)		
AATx (1 <i>byte</i>)	PTx (1 <i>byte</i>)	Nº de telemetrias (2 <i>bytes</i>)
Soma de verificação do cabeçalho (4 <i>bytes</i>)		
Pacotes de telemetria (máximo 2180 <i>bytes</i>)		
mínimo 1 e máximo 5 conjuntos com no máximo 436 <i>bytes</i> , cada		
Soma de verificação dos pacotes de telemetria (4 <i>bytes</i>)		

O campo pacotes de telemetria do pacote armazena uma lista de pacotes de telemetria de embarcações, onde existe um pacote de telemetria para cada embarcação, podendo armazenar o pacote de telemetria de uma ou mais embarcações. Cada pacote de telemetria de uma embarcação permite que as embarcações recebam o mínimo de informação possível para o controle de movimento e execução do protocolo de comunicação. Essas informações são utilizadas pelo controle da

embarcação junto com a qualidade de sinal da comunicação do pacote recebido. Assim definimos nosso protocolo descentralizado de comunicação e controle cinemático executado por cada embarcação. Além disso, agregamos ao protocolo uma comunicação em *broadcast* e TTL de um salto, assim garantimos que um pacote enviado na rede seja recebido por todos os nós que estejam no alcance da transmissão e a aplicação decide se o mesmo deverá ou não ser retransmitido. O uso somente do *broadcast* iria inundar desnecessariamente a rede e não usá-lo acarretaria em maior trabalho de cada embarcação para verificar e manter uma comunicação com cada embarcação vizinha.

4.3.1.2 Pacote de Telemetria

Cada pacote de telemetria apresenta o seguinte cabeçalho Tabela 3: os dois primeiros *bytes* guardam o tamanho total do pacote de telemetria (10 de cabeçalho, 102 de atributos mínimos, 20 vezes o campo EPs mais duas vezes o campo IDs), com valor máximo de 436 *bytes*; dois *bytes* para o identificador da embarcação (ID); quatro *bytes* para identificar o número de sequência das informações de telemetria (inteiro sem sinal); e um *byte* com o número de saltos/reencaminhamentos pelos quais o presente pacote de telemetria passou (cuja lista pode ser obtida na lista de identificadores).

Depois do cabeçalho tem-se um conjunto de atributos mínimos, compostos, então, de um conjunto de três *bytes* que indicam o número de posições a serem evitadas, o estado da embarcação e sua missão, respectivamente. Em seguida têm-se um conjunto de três estruturas de 12 *bytes* cada indicando a posição atual da embarcação, sua velocidade e seu destino, onde cada estrutura representa três números reais de quatro *bytes* cada representando coordenadas cartesianas (x,y,z); mais dois números reais de quatro *bytes* cada indicam as distâncias mínima e máxima, respectivamente, que a embarcação deverá respeitar quando chegar ao destino; os próximos 12 *bytes* representam as coordenadas cartesianas (x,y,z) da estação base; então temos quatro conjuntos de dois inteiros de 4 *bytes* cada representando marcações de tempo em segundos (primeiro inteiro) e nanosegundos (segundo inteiro) para o cálculo do tempo de resposta, carimbo de tempo da embarcação, tempo de processamento da embarcação, carimbo da estação base e processamento da estação base, respectivamente; três números reais de quatro *bytes* cada indicam o tempo de confirmação atual, a taxa de entrega de pacotes atual e o tempo limite de espera por uma confirmação, respectivamente.

Em seguida, têm-se um conjunto de informações de tamanho variável. Primeiro, uma lista de identificadores das embarcações (cada um com 2 *bytes*) que o retransmitiram, assim pode-se saber por quais nós o pacote de telemetria passou. Em seguida, uma lista de posições a serem

evitadas, cada uma com 20 *bytes*, representando os seguintes campos: três números reais de quatro *bytes* cada representando as coordenadas cartesianas (x,y,z) e dois números reais de quatro *bytes* cada representando as distâncias mínimas e máximas da coordenada. Além disso, ao final é possível acrescentar *bytes* de acordo com a necessidade da aplicação, podendo carregar qualquer mensagem que se faça necessária bastando apresentar o tamanho mínimo de um *byte* e máximo entre 194 e 294 *bytes* de acordo com a quantidade de posições a serem evitadas.

Tabela 3 - Estrutura do Pacote de Telemetria do Protocolo.

Cabeçalho do pacote de Telemetria (10 bytes)			
Tamanho do conjunto (2 bytes)		ID da embarcação (2 bytes)	
Número de sequência do conjunto (4 bytes)			
Nº IDs (1 byte)			
Atributos mínimos do pacote de Telemetria (102 bytes)			
	Nº Posições (1 byte)	Estado (1 byte)	Missão (1 byte)
Posição da embarcação (12 bytes)			
Velocidade da embarcação (12 bytes)			
Destino da embarcação (12 bytes)			
Distância mínima (4 bytes)			
Distância máxima (4 bytes)			
Posição da estação base (12 bytes)			
Carimbo de tempo da embarcação (8 bytes)			
Tempo de processamento da embarcação (8 bytes)			
Carimbo de tempo da estação base (8 bytes)			
Tempo de processamento da estação base (8 bytes)			
Tempo de confirmação (4 bytes)			
Taxa de entrega de pacotes (4 bytes)			
Tempo limite ou timeout (4 bytes)			
Lista de identificadores dos nós intermediários (máximo 30 bytes)			
máximo 15 identificadores com 2 bytes, cada			
Lista de posições a serem evitadas (máximo 100 bytes)			
máximo 5 posições com 20 bytes, cada			
Bytes extras com máximo 294 bytes			

Uma vez de posse das estruturas dos pacotes de telemetria dos vizinhos ou de atualizações da estação base, recebidas por meio das mensagens, a embarcação pode utilizar estas informações para mudar de estado e retomar sua rotina ou realizar autonomamente operações de controle como acelerar/desacelerar, manter a velocidade, ou mudar a direção para cumprir sua missão, desviar de obstáculos, evitar colisões com embarcações vizinhas ou não perder a comunicação com a base. Como será visto na Seção 3.3 foi implementada uma solução para integrar o simulador de redes NS-3 com o MATLAB de forma que as ações de atuação e controle sejam realizadas neste ambiente. Entretanto, nesta dissertação não exploramos todas as possibilidades de controle, realizando um

controle cinemático simples no MATLAB, e nos aprofundando nos aspectos de comunicação e no protocolo. Ainda assim, todo o arcabouço de integração com o MATLAB está preparado para que operações de controle mais sofisticadas possam ser testadas.

4.3.1.3 Observações

As estruturas do pacote das mensagens trocadas (Tabela 2) e dos pacotes de telemetria contidos em cada pacote de mensagem (Tabela 3) foram definidas para que:

- as embarcações transmitam à estação base todas as informações de telemetria necessárias, junto com informações de controle de sua identificação (ID) e de sequência destas mensagens;
- ao receber uma mensagem a embarcação consiga identificar se existe alguma atualização enviada pela base. Isso é feito (i) percorrendo o conjunto de estruturas de pacotes de telemetria, inspecionando o **ID da embarcação** do cabeçalho desta estrutura, até encontrar seu próprio ID e, em seguida (ii) verificando se o número de sequência da mensagem é maior do que o número de sequência registrado localmente. Caso positivo isso indica que a base pode ter atualizado alguma informação, como o comando de transição de estado;
- ao receber uma mensagem a embarcação identifique a mesma como uma vizinha, uma vez que recebeu a mensagem pelo enlace, e obtenha seus dados de telemetria e estado. Isso é feito inspecionando (i) o campo **ID da embarcação** do cabeçalho da mensagem (Tabela 2) e depois (ii) percorrendo o conjunto de pacotes de telemetria, inspecionando o **ID da embarcação** no cabeçalho desta estrutura (Tabela 3), até encontrar o mesmo ID do **ID do transmissor**.

Cada elemento mantém listas locais de estruturas de dados de telemetria, com o mesmo formato usado no pacote de telemetria: a estação base mantém uma lista de pacotes de telemetria de todas as embarcações que se registraram. Sempre que um pacote de mensagem é recebido, a estação base extrai os pacotes de telemetria e atualiza os mesmos caso o número de sequência seja maior que o anterior (ou seja, a respectiva embarcação enviou dados de telemetria mais recentes); cada embarcação mantém (i) uma estrutura de dados de telemetria com seus próprios dados, obtidos periodicamente de sensores ou por meio dos cálculos de controle realizados no MATLAB; (ii) uma lista de pacotes de telemetria das embarcações vizinhas. Esta lista é mantida durante toda a execução, porém somente as informações mais atuais (pacote de telemetria recebido entre \pm duas vezes o *timeout* da embarcação) são passadas para as rotinas de controle do MATLAB; e (iii)

uma lista temporária de pacotes de telemetria a serem encaminhados, quando no papel de nó intermediário.

Observa-se que cada embarcação também mantém um vetor de tuplas com o *ID* e o *número de sequência* dos pacotes de telemetria recebidos para otimizar o reencaminhamento dos mesmos, evitando reencaminhamentos desnecessários (Seção 5.1).

4.3.2 A Máquina de Estados

O protocolo apresenta um modelo híbrido de comunicação. As embarcações e a estação base sempre enviam mensagens de *broadcast*. Assim, as mensagens são enviadas pelo remetente sem a necessidade de bloqueio para aguardar uma resposta, o que poderia ser classificado de modelo assíncrono ou *one-way*. Entretanto, determinadas mensagens requerem uma confirmação ou atualização para que o nó mude de estado. Neste caso, o modelo de comunicação pode ser classificado como **request-response**, no qual os módulos remetentes (embarcações, por exemplo) enviam uma mensagem com alguma informação que requer uma resposta ou uma confirmação do destinatário (a estação base, por exemplo). Reforça-se, entretanto, que a troca de mensagens não é bloqueante. As confirmações são recebidas assincronamente e a confirmação verificada pela rotina do protocolo de aplicação.

A estação base tem comportamento diferente das embarcações. Como visto no diagrama de estados na Figura 15, a estação base inicia aguardando mensagens das embarcações. Cada mensagem recebida é verificada quanto à sua integridade, utilizando um algoritmo simples para detecção de erros baseado em [19, 72–74]. Para isso, cada mensagem trocada entre as aplicações apresenta um número de 32 *bits* para o cabeçalho e outro para os pacotes de telemetria, como mostra a Tabela 2, representando a soma de verificação dos *bytes* do cabeçalho e dos pacotes de telemetria da mensagem, respectivamente. A mensagem é descartada caso apresente algum erro, caso contrário a estação base então verifica e atualiza os pacotes de telemetria das embarcações incluídos na mensagem.

Para cada pacote de telemetria recebido, a estação base verifica se o identificador do mesmo já está na lista de embarcações registradas, caso não esteja o pacote de telemetria da embarcação é incluído. Caso a embarcação já esteja na lista, a estação base então verifica se o número de sequência recebido é maior do que o seu correspondente na lista. Caso seja, a lista é atualizada. Caso contrário, o pacote de telemetria é ignorado. Depois de verificar todos os pacotes de telemetria

incluídos na mensagem, a estação base envia uma mensagem de confirmação para as embarcações na lista de embarcações registradas. Uma confirmação é composta pelo pacote de telemetria da própria embarcação remetente, tendo seu número de sequência incrementado em uma unidade e outros campos (Tabela 3) modificados de acordo com a necessidade da estação base.

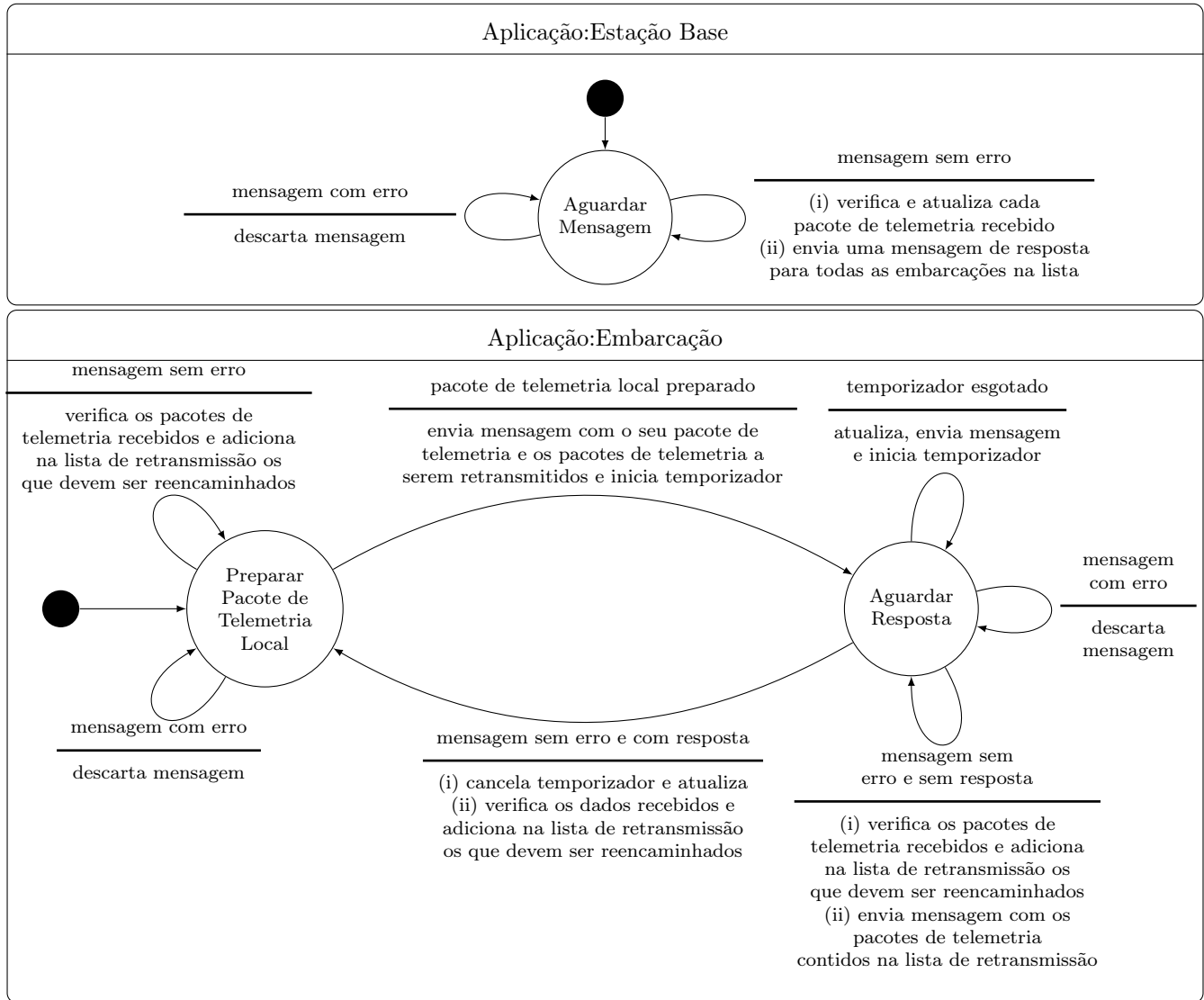


Figura 15 - Máquina de Estados Finito do Protocolo.

Assim como a estação base, qualquer mensagem recebida com erro por uma embarcação é descartada. Ao iniciar, cada embarcação define um valor inicial para seu número de sequência e cria uma mensagem com suas informações de acordo com o formato da Tabela 3, definindo seu pacote de telemetria local. Caso uma mensagem seja recebida sem erro durante o estado **Preparar Pacote de Telemetria Local** a embarcação verifica os pacotes de telemetria incluídos na mensagem e adiciona em sua lista temporária somente os que não apresentarem registro em seu vetor de tuplas ou que

apresentarem número de sequência maior que seu correspondente nesse vetor. Esta lista temporária armazena todas os pacotes de telemetria das embarcações que devem ser retransmitidos e o vetor de tuplas mantém atualizada a lista de pacotes de telemetria já retransmitidos pela embarcação. Por fim, a mensagem com o pacote de telemetria local e os que deverão ser retransmitidos (caso exista) é enviada e seu temporizador é iniciado, passando em seguida para o estado **Aguardar Resposta**.

No estado **Aguardar Resposta** quando o temporizador chega ao fim é inferido que a mensagem anterior foi perdida, a embarcação então incrementa seu número de sequência em dois e atualiza suas informações de telemetria, envia uma mensagem e inicia o temporizador novamente. Caso uma mensagem seja recebida sem erro e não haja um pacote de telemetria com sua identificação local e número de sequência maior que o atual (confirmação da estação base) a embarcação verifica os pacotes de telemetria incluídos na mensagem e adiciona na lista temporária somente os que deverão ser retransmitidos. Se ao final da verificação a lista temporária for diferente de vazia uma mensagem é enviada com os pacotes de telemetria contidos na lista temporária, caso contrário nada é feito.

Quando uma mensagem contendo um pacote de telemetria com sua identificação local e número de sequência maior que o atual (resposta da estação base) é recebida, a embarcação cancela seu temporizador, atualiza suas informações de acordo com a resposta recebida, faz seu número de sequência ser igual ao número de sequência recebido mais um. Além disso, os outros pacotes de telemetria são verificados e adicionados na lista temporária somente os que deverão ser retransmitidos. Por fim, a embarcação volta para o estado **Preparar Pacote de Telemetria Local**.

O temporizador, o número de sequência e a soma de verificação foram a solução proposta para a implementação do mecanismo de transferência confiável de dados por um canal com perdas e com erros de *bits*, com características inspiradas no protocolo TCP [7–11, 19, 25]. Porém, diferente do TCP, somente as embarcações verificam a entrega e efetuam retransmissões quando necessário, a estação base somente envia a resposta (comandos) que serve como reconhecimento (*acknowledgment*) de recebimento e caso um comando seja perdido ele é transmitido na próxima mensagem que a estação base receber. Decisão tomada para que estação base não precisa manter um temporizador para cada embarcação conectada. O número de mensagens enviadas também é menor, visto que apenas as embarcações efetuam retransmissões decorrentes das perdas em qualquer direção (embarcação para a estação base ou estação base para a embarcação). As embarcações podem descartar respostas com reconhecimentos defasados e a estação base não precisa ficar enviando respostas repetidas para mensagens perdidas.

Foi acrescentado o recurso de agregação de dados de diversas embarcações em uma única mensagem, permitindo que nós intermediários adicionem seus dados antes de reencaminharem a mensagem, campo **Pacotes de Telemetria** na Tabela 2, diminuindo o número de mensagens transmitidas e permitindo que os nós ao transmitirem sempre enviem informações suficientes para os nós que receberem. Os nós intermediários também adicionam seu identificador aos dados retransmitidos (campo **Lista de identificadores** na Tabela 3), permitindo que o nó destinatário possa saber por quais nós o pacote de telemetria passou até ser recebido (campo **Nº IDs** na Tabela 3 indica o número de saltos).

Em todos os estados das aplicações mostradas na Figura 16 é utilizado o protocolo de comunicação da Figura 15. O diagrama de atividades na Figura 16 apresenta o ciclo de vida de cada elemento. O diagrama de máquina de estados finitos da comunicação foi apresentado separado para facilitar o entendimento da comunicação e simplificar o diagrama de atividades, assim este último apresenta apenas o fluxo de estados e condições de mudança em cada fase.

Cada fase ou atividade na Figura 16 representa um estado e é enviado no pacote de telemetria de uma embarcação (campo **Estado** na Tabela 3), assim as embarcações enviam seu estado atual e a estação base responde com o estado desejado (a estação base sempre envia o próximo estado a ser alcançado pelas embarcações). Ao iniciar a estação base fica no estado **Aguardar Conexões** até que todas as embarcações (total de embarcações é informado ao iniciar) tenham sido reconhecidas. Quando a estação base recebe de uma embarcação o estado anteriormente enviado e seu estado armazenado anteriormente é diferente o atual, a estação base sabe que a resposta chegou com sucesso e então incrementa um contador que indica quantas embarcações estão no estado desejado. Quando esse contador é igual ao número total de embarcações a estação base muda para o próximo estado.

Para confirmar a mudança de estado das embarcações a estação base utiliza a técnica *handshake* de três vias (utilizada pelo protocolo TCP no estabelecimento de conexões [7–11]), que envolve o envio de três mensagens (ver Figura 17). Em todas as fases a estação base utiliza essa técnica, assim somente depois de confirmar que a embarcação realmente mudou seu estado é que a estação base a considera no estado correto. Isso foi necessário pois a estação base poderia enviar uma resposta e a mesma ser perdida (ver Figura 17(b)), causando um estado inconsistente na estação base. Com o *handshake* de três vias é possível garantir que a estação base sempre esteja sincronizada com as embarcações, mesmo que a comunicação apresente falhas como mostrado na Figura 17(b) o **OK** na estação base só ocorre depois que ela recebe uma resposta confirmando

o recebimento.

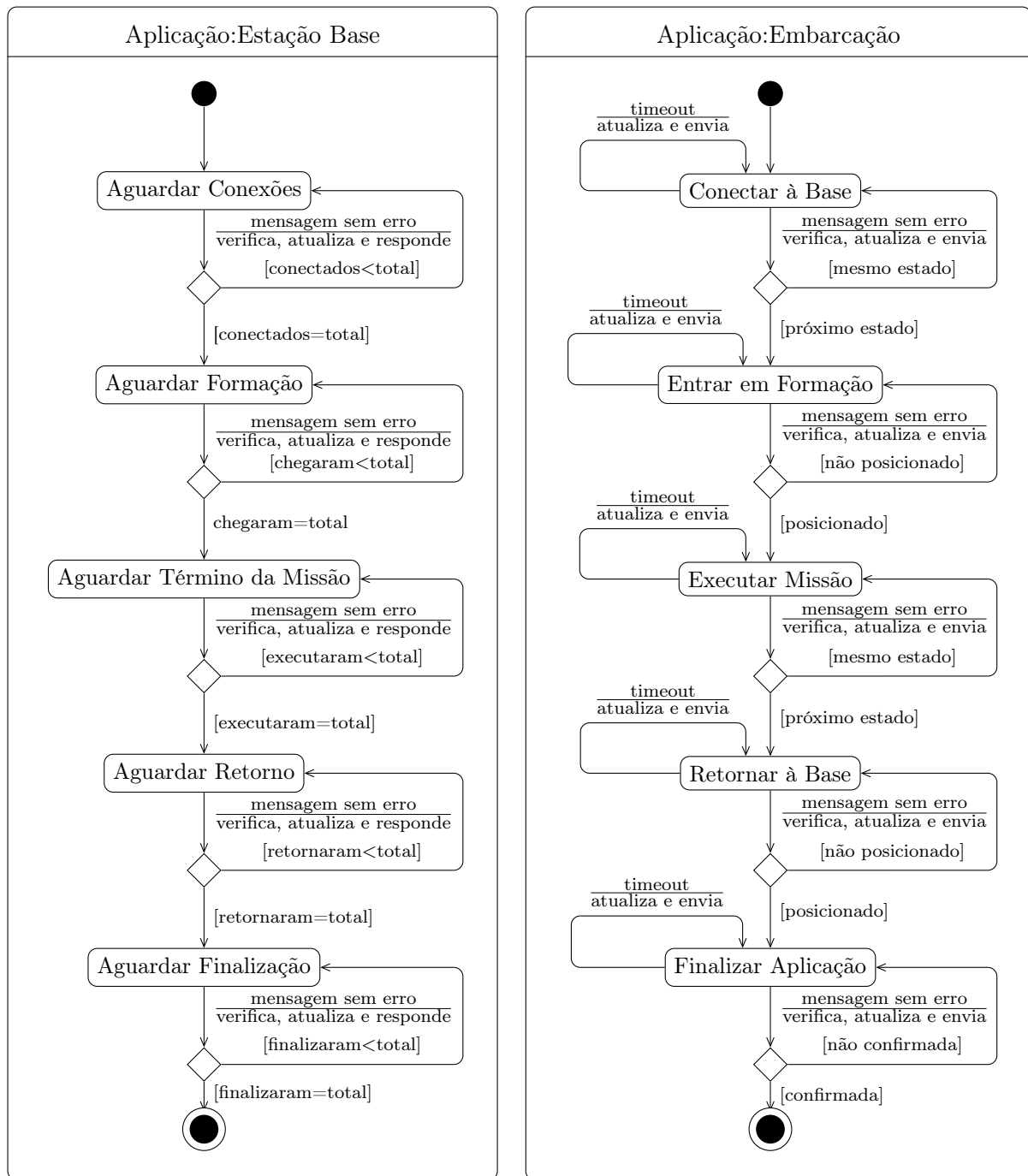


Figura 16 - Diagrama de Atividades das Aplicações.

As embarcações ao receberem a resposta da estação base com um novo estado mudam imediatamente para este estado, exceto quando estão nos estados **Entrar em Formação**, **Retornar à Base** ou **Finalizar Aplicação**. No caso dos dois primeiros, as embarcações só mudam para o próximo estado quando alcançam a posição desejada (destino da missão no caso do primeiro estado

e posição da estação base no caso do segundo). Já no estado **Finalizar Aplicação** as embarcações aguardam a confirmação da estação base quanto ao recebimento da finalização da aplicação como mostrado na Figura 18 (utilizada pelo protocolo TCP no fechamento de conexões [7–11]).

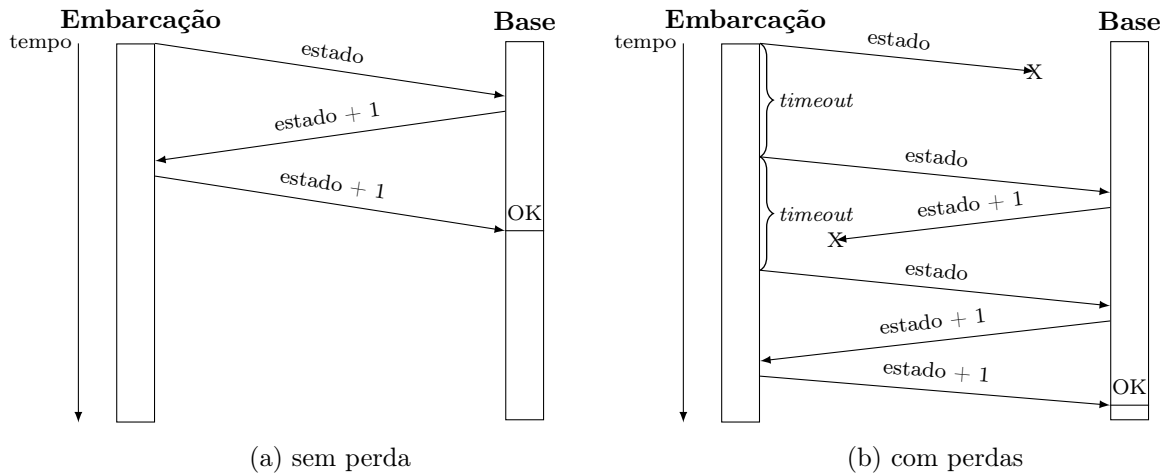


Figura 17 - *Handshake* de Três Vias.

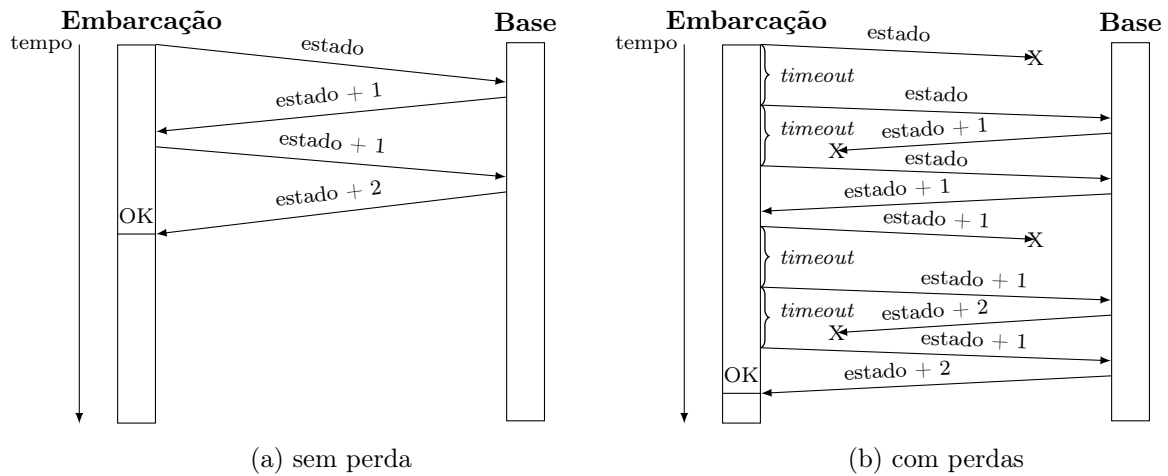


Figura 18 - Finalização da Aplicação Embarcação.

Como a estação base desempenha o papel de destinatário para todas as embarcações, ela sempre aguarda uma mensagem de uma embarcação. Ao receber uma mensagem a estação base atualiza suas informações e envia uma resposta para as embarcações em uma única mensagem. Além disso, ela verifica se o número de embarcações no mesmo estado é igual ao número total de embarcações para poder mudar para o próximo estado, caso contrário o estado atual é mantido.

Ao iniciar cada embarcação envia a primeira mensagem para a estação base, que por sua

vez adiciona as informações da embarcação em sua lista de embarcações conectadas e envia uma resposta. Ao receber a resposta da estação base cada embarcação verifica qual estado a mesma deverá apresentar e atualiza seu estado. Os únicos estados que não são controlados diretamente pela estação base são: **Entrar em Formação** e **Retornar**, pois suas condições de mudança são dependentes da posição em que a embarcação se encontra, por este motivo a embarcação ao chegar ao destino muda seu estado indicando para a estação base que ela já está pronta para a próxima fase.

5 IMPLEMENTAÇÃO DO PROTOCOLO EM SIMULADOR

Neste capítulo apresentamos a implementação do protocolo no simulador NS-3 e sua integração com o MATLAB. O capítulo está organizado da seguinte forma: a Seção 5.1 apresenta os detalhes da implementação do protocolo no simulador de redes NS-3 e na Seção 5.2 como foi feita sua integração com o MATLAB.

5.1 Implementação NS-3

Com o protocolo de aplicação definido, foi iniciada então a fase de implementação do módulo extra no NS-3 para acrescentar as funcionalidade necessárias.

Como já introduzido na Seção 1.2, o NS-3 é um simulador de redes de eventos discretos implementado em C++ que é construído como uma biblioteca que pode ser usada estática ou dinamicamente por programas também desenvolvidos em C++ ou Python [12]. Uma boa parte do tempo foi dedicada ao desenvolvimento do módulo para este simulador, de forma que permitisse não só a parametrização de uma simulação como também facilitasse sua execução.

Foi seguida a documentação e tutoriais encontrados na *Internet*, pois apenas o manual do NS-3 não contemplava toda a ajuda e passo a passo necessários para o desenvolvimento. Sua estrutura e arquitetura facilitou o desenvolvimento do novo módulo, mas em alguns casos a falta de documentação específica acabou aumentando o tempo de desenvolvimento.

5.1.1 Aplicações

Seguindo as técnicas de programação orientada a objetos (POO), assim como o próprio NS-3, foram desenvolvidas algumas classes derivadas de classes-base do NS-3 e outras para suporte, Figura 19. As classes que são referentes às aplicações (*Base* e *Embarcação*) são derivadas da interface *Application* definida pelo NS-3, assim nossa aplicação pode ser adicionada ao simulador de forma organizada. As rotinas das aplicações poderiam ser apenas funções disjuntas, porém isso dificultaria a leitura do código, seu reuso e até mesmo a execução da simulação.

A classe *Cabeçalho* é derivada da interface *Header* definida pelo NS-3. Ela apresenta as informações de um cabeçalho do pacote de mensagem (Tabela 2 e a interface do NS-3 fornece os recursos básicos para adicioná-lo e removê-lo de um *Packet* (classe pacote do NS-3, usada mas não indicada no diagrama). A classe *Mensagem* apresenta a estrutura de um pacote de mensagem

Para as aplicações simularem o tempo necessário para processar cada pacote de mensagem e tomarem as devidas ações, foi acrescentado um atraso entre o recebimento e envio das mensagens proporcional a quantidade de pacotes de telemetria incluídos no pacote de mensagem. Foi estipulado constantes diferentes para a aplicação da estação base e para a aplicação das embarcações, de forma que os tempo de processamento fossem diferentes e simulando que a estação base apresente maior capacidade computacional (utilizado o valor de 10 microssegundos para cada pacote de telemetria recebido em um pacote de mensagem) com um fator menor do que as embarcações (utilizado o valor de 20 microssegundos para cada pacote de telemetria recebido em um pacote de mensagem). Como geralmente os atrasos de processamento em roteadores de alta velocidade normalmente são da ordem de microssegundos [19] foram utilizados valores na ordem de microssegundos, apenas para explorar a capacidade do NS-3 de simular o tempo de processamento. Valores obtidos de equipamentos reais podem ser utilizados futuramente. Também não foi feita nenhuma modificação no núcleo do NS-3 para garantir que os atrasos de processamento entre as camadas fossem respeitados, isso iria requerer muito tempo para modificar quase todo o núcleo do NS-3, por isso foi utilizado os atrasos já incluídos.

5.1.2 Modelo de Propagação

Alguns modelos de propagação são apresentados em [15,16,75,76], como os modelos de espaço livre, de dois raios, de quatro raios e *log-distance*. Em [76] é feito um experimento da comunicação sobre o mar perto do porto e os resultados foram utilizados para encontrar os parâmetros correspondentes para o modelo *log-distance*. O modelo de dois raios é usando em [15,16,75] e em [16,75] experimentos foram realizados com o intuito de confrontar os dois resultados.

Foi necessária a implementação de um modelo de propagação que fosse compatível com a comunicação por rádio frequência sobre superfícies marítimas, para dar maior veracidade aos resultados. Por ter sido o modelo que mais foi citado nos trabalhos pesquisados [15,16,75,76], foi utilizado o modelo de dois raios. Este modelo de propagação considera a existência de dois raios chegando a uma antena receptora, sendo um raio direto e outro refletido numa superfície [75], como mostrado na Figura 20.

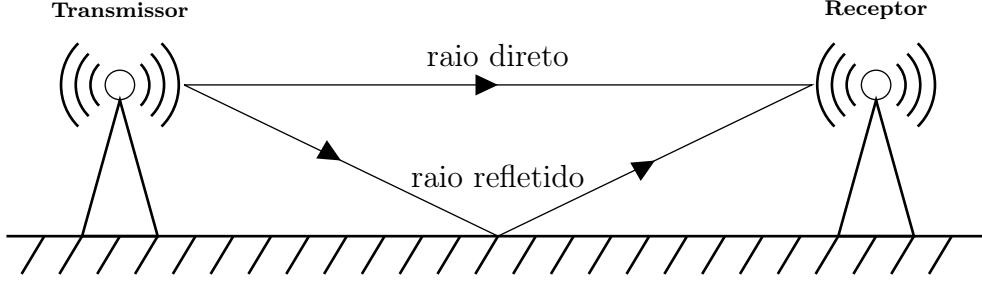


Figura 20 - Exemplificação do modelo de dois raios. Adaptado de [75]

Segundo o modelo de propagação implementado, a potência do sinal recebido é dada pela equação:

$$P_r = P_t + G_t + G_r + R_f - L_s - L_r, \quad (14)$$

na qual P_t é a potência de transmissão, G_t é o ganho do transmissor, G_r é o ganho do receptor dados em dB, R_f é uma componente de desvanecimento, L_s é a soma de todas as perdas do sistema (equipamentos e cabos) e L_r é a perda de percurso definida pela equação [14–16]:

$$L_r = 10 \log_{10} \left(\frac{\lambda^2 2 \sin^2 \left(\frac{2\pi h_t h_r}{\lambda d} \right)}{(4\pi d)^2} \right), \quad (15)$$

na qual λ é o comprimento de onda (em metros) do sinal, h_t e h_r são as alturas das antenas (em metros) do transmissões e do receptor, respectivamente e d é a distância (em metros) entre transmissor e receptor.

Para R_f , de acordo com [16], pode-se usar a distribuição Rayleigh que apresenta densidade de probabilidade segundo a função:

$$f(x; \sigma_r) = \frac{x}{\sigma_r^2} e^{\left(\frac{-x^2}{2\sigma_r^2} \right)}, \quad x \geq 0, \quad (16)$$

na qual σ_r é o parâmetro de escala da distribuição. Sua média é dada pela equação [77]:

$$\bar{x} = \sqrt{\frac{\sigma_r \pi}{2}}, \quad (17)$$

em relação ao parâmetro σ_r .

Como o NS-3 não apresenta suporte para a distribuição Rayleigh, foi utilizado uma aproximação usando a distribuição Weibull, dada pela equação [77]:

$$f(x) = \frac{kx^{k-1}}{\lambda^k} e^{-\left(\frac{x}{\lambda}\right)^k}, \quad x \geq 0, \quad (18)$$

na qual k é o parâmetro referente a forma e λ o parâmetro de escala. Esta distribuição é suportada pelo NS-3 e usando os valores de $k = 2$ para o parâmetro da forma e $\lambda = \sigma_r \sqrt{2}$ para o parâmetro da escala, fazendo $\sigma_r = 1$ é obtida a aproximação desejada [16] com média de acordo com a equação [77]:

$$\bar{x} = \lambda \Gamma \left(1 + \frac{1}{k} \right), \quad (19)$$

em relação aos parâmetros k e λ e Γ , sendo Γ a função Gama encontrada em [77].

5.1.3 Recursos Adicionais

Foi necessário criar um arquivo de configuração para passar os parâmetros de execução da simulação devido a quantidade de parâmetros a serem passados. O arquivo de configuração (Apêndice A) foi dividido em quatro partes: a primeira representa os parâmetros gerais da simulação, como semente de aleatoriedade, quantidade de rodadas, tipo de missão, tempo limite inicial, tempo de execução, intervalo entre as chamadas ao MATLAB, posição de destino da missão e lista de posições a serem evitadas; depois são os parâmetros para a definição do canal de comunicação, atraso de propagação e os parâmetros para o modelo de propagação; em seguida vem os parâmetros de configuração da estação base, como quantidade de estações base (os recursos para várias estações base está implementado, porém o protocolo foi desenvolvido para ser usada apenas uma), suas posições e as informações de sua interface física; finalmente os parâmetros de configuração das embarcações, como número de embarcações, suas posições e as informações de suas interfaces físicas.

Para que fosse possível integrar o módulo criado ao NS-3, tornando-o mais um recurso de simulação, tivemos que seguir as recomendações do mesmo o que implicou a criação de um *script* em Python (Apêndice B). Esse *script* apresenta três funções principais, que são exigidas pela ferramenta de construção do NS-3. A primeira serve para adicionar novas entradas na lista de opções do simulador, permitindo ativar ou não o módulo. A segunda auxilia a ferramenta a verificar os recursos necessários caso o uso do módulo esteja ativado, por exemplo: verificando se o programa MATLAB e sua biblioteca estão instalados, caso não sejam encontrados um erro é reportado e a

simulação não é executada. Por último, temos a função que define a construção do módulo, arquivos utilizados e os parâmetros para compilação e lincagem, por exemplo: adicionamos os parâmetros para a inclusão da biblioteca do MATLAB.

Apesar de facilitar a criação e inclusão de novos módulos, foi difícil chegar no resultado esperado devido a falta de documentação mais detalhada e atualizada de como criar o *script* em Python para usar bibliotecas externas. Algumas modificações no código original do NS-3, a criação de *links* simbólicos apontando para os arquivos da biblioteca do MATLAB são necessários para que tudo funcione corretamente, o roteiro para essas modificações é apresentado no Apêndice C.

5.2 Detalhes da Integração do NS-3 com o MATLAB

Como mencionado anteriormente, utilizamos o MATLAB em nossa proposta de implementação para realizar todos os cálculos de controle e atuação (simulação de acionamento de motores, alteração ou adequação de velocidade e aceleração nos três eixos, etc.).

O MATLAB pode executar programas escritos em Java, Python, ActiveX, .NET, Fortran, C e C++ por meio de bibliotecas. Existem duas abordagens que são oferecidas pelo MATLAB para a execução de programas externos. A primeira abordagem usa arquivos de bibliotecas vinculadas dinamicamente (Dynamic Linked Libraries, DLL) em sistemas operacionais Windows ou de objetos compartilhados (Shared Objects, SO) nos sistemas operacionais Unix, que depois de gerados podem ser usados diretamente no MATLAB. A segunda abordagem pode ser feita usando arquivos MEX, uma interface que permite ao MATLAB executar programas escritos em Fortran ou C/C++. O uso de DLL/SO supera a abordagem MEX em termos de facilidade de geração e robustez, pois essas bibliotecas podem ser usadas em outras ferramentas de engenharia ou ambientes matemáticos. No entanto, ela não fornece um acesso interno para que o usuário modifique arquivos e introduza funções personalizadas, a menos que os arquivos fonte sejam alterados antes de gerar as DLLs/SO [65].

O MATLAB também oferece uma API para Java, Python, Fortran, C e C++ que permite programas, escritos nessas linguagens, efetuarem chamadas ao MATLAB de forma sincronizada ou não. Essas opções estão disponíveis para permitir ao programador escolher qual é a melhor para sua aplicação. Incluído nessa API estão chamadas de ambiente (iniciação, término, conectar a uma instância compartilhada e conversão de formatação de texto), criação, envio e recebimento dos diversos tipos de estruturas de dados disponíveis no MATLAB, execução de funções com a verificação de sintaxe feita pelo MATLAB. É possível usar todos os recursos disponíveis no MATLAB

diretamente em código externos e por se tratar de algo disponibilizado pelo próprio MATLAB se torna mais seguro e confiável.

Como o NS-3 é implementado em C++ e em nossa solução ele é quem controlaria os momentos de operação de controle no MATLAB, optou-se por utilizar a API do MATLAB para a integração. Assim, o módulo criado para o NS-3 verifica se o MATLAB está instalado (tenta executar o comando “matlab”) e se sua biblioteca está disponível (feito durante a compilação e ligação). Depois de tudo verificado o módulo do NS-3 utiliza as rotinas encapsuladas na classe *VesselsMatlab* na Figura 21 para executar chamadas de rotinas no MATLAB, definindo assim a interface de integração entre os dois ambientes. Estas rotinas utilizam a API do MATLAB para enviar as informações da comunicação e telemetria para o MATLAB e receber como resposta as informações de cinemática calculadas pelo mesmo.

Como mostrado na Figura 21, o simulador de redes cria uma instância da classe *VesselsMATLAB*, que por sua vez inicia a *engine* do MATLAB. Durante toda a simulação o NS-3 chama a rotina de cálculo da classe *VesselsMATLAB* em intervalos regulares, que por sua vez coleta as informações necessárias (Tabela 4) de todas as embarcações. Neste momento a execução do NS-3 é interrompida até que o rotina finalize. Utilizando a mesma técnica de POO usada no NS3, foram definidas duas classes no MATLAB: a classe *Sistema* que encapsula todas as rotinas para lidar com grupos de embarcações e rotinas auxiliares para plotagem de gráficos; a classe *Barco* que encapsula as rotinas para os cálculos das equações de posição (11), velocidade (12) e aceleração (13) e outras auxiliares.

Tabela 4 - Estrutura de Dados Enviada do NS-3 para o MATLAB.

ID da embarcação	Missão	Posição inicial (x,y,z)	Informações da comunicação com as vizinhas
		Velocidade máxima (x,y,z)	
		Dimensões (x,y,z)	
		Posição da estação base (x,y,z)	

As informações enviadas do NS-3 para o MATLAB, apresentadas na Tabela 4, são os campos necessários para a criação dos objetos da classe *Barco* que irão manter as informações trocadas entre os dois ambientes de forma organizada, visto que para cada embarcação é associado a um objeto da classe *Barco*, que corresponde uma aplicação *Embarcação* no NS-3. A classe *Sistema* agrupa e facilita a busca pelo objeto correspondente ao identificador da embarcação. Ela pode receber um ou mais conjuntos de informações (Tabela 4), onde cada conjunto representa as informações

obtidas de uma aplicação *Embarcação* do NS-3. Sendo assim, o primeiro campo é o identificador da embarcação (o mesmo do pacote de telemetria Tabela 3), depois vem a missão a ser executada (o mesmo do pacote de telemetria Tabela 3), então vem um conjunto com quatro grupos de três números reais, que representam as coordenadas cartesianas (x,y,z), que indicam sua posição inicial, sua velocidade máxima, suas dimensões e a posição da base, respectivamente. Este último, foi acrescentado para permitir, no futuro, a utilização de diversas bases.

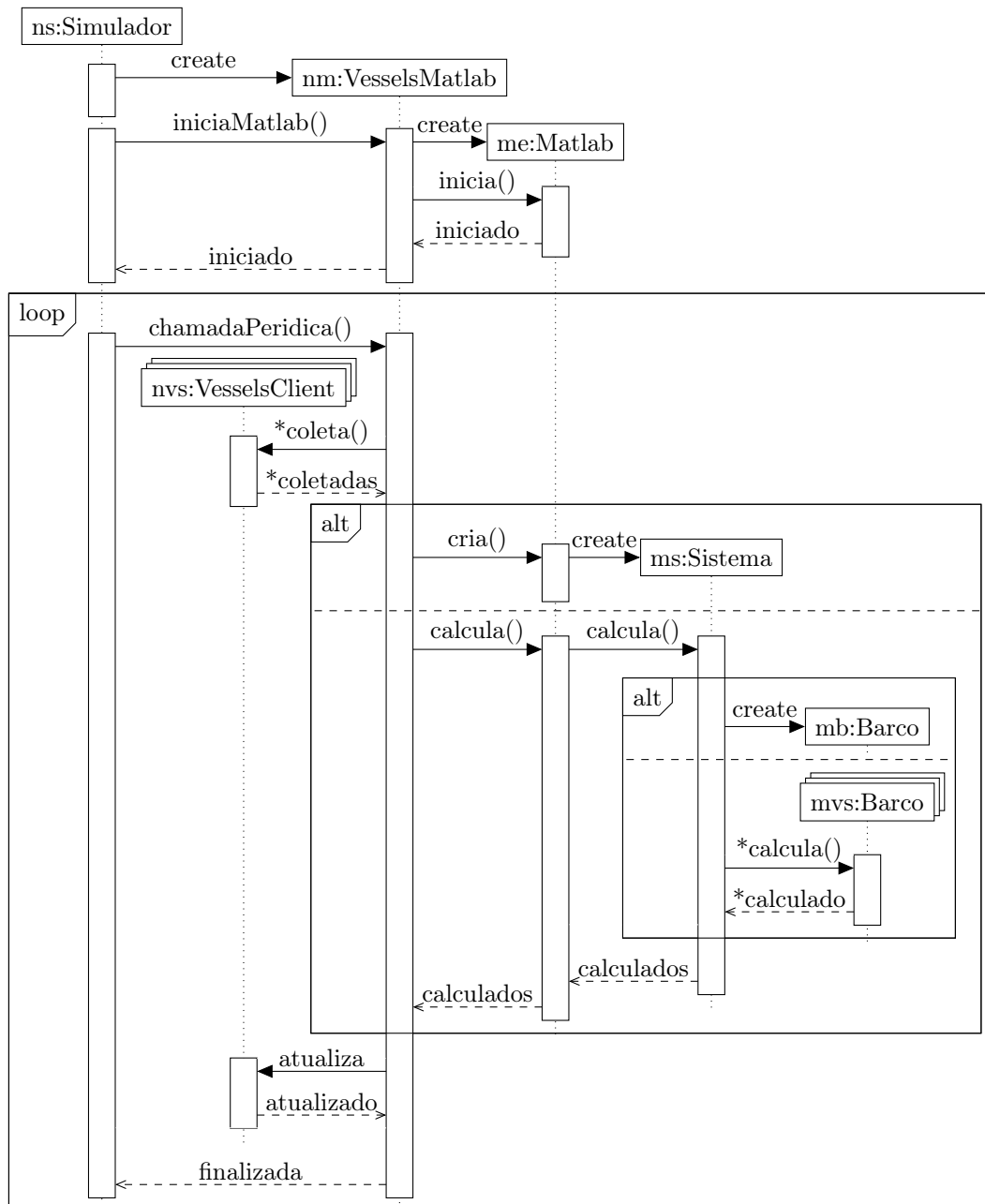


Figura 21 - Diagrama de sequência (Integração NS-3 com MATLAB).

No fim do conjunto de informações enviadas do NS-3 para o MATLAB (Tabela 4), vem o grupo de informações referentes a comunicação com suas embarcações vizinhas. Ele é uma matriz de números reais com 20 colunas e uma linha para informações da própria embarcação mais uma linha para cada embarcação vizinha. Cada coluna representa uma informação, na seguinte ordem: identificação da embarcação (Tabela 3); instante de tempo da chamada feita pelo NS-3 (o tempo é controlado pelo NS-3), estado (Tabela 3); altura da antena do transmissor (AATx na Tabela 2); potência de transmissão (PTx na Tabela 2); potência do sinal recebido (obtido da interface de comunicação ao receber um pacote da rede), na primeira linha este campo é sempre zero, já que esta linha é reservada para as informações da própria embarcação; relação sinal ruído (obtido da interface de comunicação ao receber um pacote da rede), também é zero na primeira linha; tempo de confirmação (Tabela 3); taxa de entrega de pacotes (Tabela 3); posição (x,y,z), velocidade (x,y,z), destino (x,y,z), distância mínima e máxima (como na Tabela 3).

A rotina de cálculo da classe *VesselsMatlab*, antes de enviar os dados para o MATLAB, verifica se um objeto da classe *Sistema* foi criado no MATLAB, caso negativo ele é criado. Com o objeto criado, a rotina de cálculo da classe *VesselsMatlab* envia os dados (Tabela 4) de todas as embarcações para o MATLAB e os utiliza como parâmetro para chamar a rotina correspondente do objeto da classe *Sistema*. Essa rotina identifica individualmente as informações de cada embarcação e mantém localmente uma lista de objetos da classe *Barco*, sendo um para cada embarcação (aplicação *Embarcação*). Caso seja recebido um conjunto de informações referente a uma embarcação que ainda não apresente um objeto associado (utilizando sua identificação na Tabela 4) um novo objeto é criado usando como parâmetro as informações recebidas. A rotina da classe *Sistema* efetua uma chamada a rotina de cálculo de cada objeto da classe *Barco* enviando como parâmetro as informações da comunicação correspondente.

Cada objeto *Barco* recebe do NS-3 um conjunto de informações (Tabela 4) relativo a uma embarcação. Essas informações são organizadas e armazenadas localmente para futuras depurações e geração de gráficos. Essa rotina então calcula um novo vetor direção baseado no vetor anterior e um vetor baseado nas informações das embarcações vizinhas que foram recebidas, esse vetor direção resultante então é usado como parâmetro para os cálculos de posição (11), velocidade (12) e aceleração (13) junto com os resultados da execução anterior (como informado na Seção 2.4, na primeira chamada a velocidade anterior é zero e a posição anterior é a posição inicial). Os resultados dos cálculos são então persistidos localmente e retornados, apresentando a estrutura da Tabela 5. O objeto da classe *Sistema* agrupa as informações de retorno Tabela 5 em uma só estrutura de

retorno (sendo uma linha para cada embarcação), então a rotina de cálculo da classe *VesselsMatlab* aplica as modificações chamando a rotina de cada aplicação *Embarcação* passando suas respectivas atualizações. Ao final, o simulador NS-3 retoma sua execução até a próxima chamada periódica.

A estrutura de retorno (Tabela 5) de cada objeto da classe *Barco* apresenta os seguintes campos: identificação da embarcação, instante de tempo da chamada, estado (somente nas fases *Entrar em Formação* e *Retornar a Base* esse campo pode apresentar valor diferente do recebido, como mencionado na Seção 4.3.2 e mostrado na Figura 16); a posição, velocidade e aceleração (x,y,z) corrigidas pelo MATLAB e que devem ser aplicadas no modelo de mobilidade utilizado no NS-3. Assim, para cada embarcação (aplicação *Embarcação*) existe um conjunto de retorno no formato mostrado na Tabela 5.

Tabela 5 - Estrutura de Dados Retornada do MATLAB para o NS-3.

ID da embarcação	Tempo Atual	Estado	Posição (x,y,z)	Velocidade (x,y,z)	Aceleração (x,y,z)
------------------	-------------	--------	-----------------	--------------------	--------------------

A utilização do MATLAB para simular a mobilidade das embarcações, foi adotada visando uma melhor representação do deslocamento das embarcações e possibilitar a utilização de modelos mais usuais ou até mesmo com modelos de motores reais usando os devidos controladores (esta parte não era o foco, por isso só foi implementado um modelo básico). O NS-3 apresenta modelos de mobilidade cinemáticos de posição constante, velocidade constante ou aceleração constante, que não são suficientes para simular a movimentação das embarcações. Assim, foi utilizado o modelo dinâmico em malha fechada apresentado na Seção 2.3 com o controlador apresentado na Seção 2.4 como modelo de mobilidade das embarcações.

Os dados de retorno (Tabela 5) foram escolhido com base no modelo cinemático disponibilizado pelo NS-3 que mais se aproximou do modelo dinâmico em malha fechada das embarcações, equação (11), e seus respectivos parâmetros. O modelo cinemático do NS-3 segue a equação para cada coordenada (x,y,z):

$$p(t) = \frac{at^2}{2} + v_0t + p_0, \quad (20)$$

na qual: $p(t)$ é a posição no instante t de tempo, a é a constante de aceleração e v_0 e p_0 são, respectivamente, a velocidade e posição iniciais. Nesse modelo é possível mudar somente os valores da posição e velocidade iniciais e a constante de aceleração, por isso o modelo dinâmico em malha fechada implementado no MATLAB retorna tais parâmetros que são aplicados ao modelo cinemático

no NS-3. Apesar do NS-3 utilizar as coordenadas (x,y,z) , o modelo implementado no MATLAB não leva em consideração a coordenada “z” e por isso ela é sempre mantida com velocidade e aceleração zero e a posição é mantida com o valor inicial. Como o NS-3 utiliza a coordenada “z” no cálculo de propagação como parâmetro da altura das antenas o valor inicial nesta coordenada corresponde a altura desejada para as estações base e embarcações e o mesmo é mantido durante toda a simulação, pois não levamos em consideração possíveis ondulações.

Como o modelo cinemático do NS-3 não contempla a dinâmica das embarcações como feito pelo modelo dinâmico em malha fechada implementado no MATLAB é obtido um erro de posição entre os dois modelos. Para minimizar este erro foi utilizado um intervalo entre as chamadas ao MATLAB de 10 milissegundos, de forma a obter uma diferença de poucos centímetros na posição entre os dois ambientes. Além disso, a solução adotada para a integração dos dois ambientes apresentou uma resposta satisfatória mediante ao curto intervalo entre as chamadas, pois a evolução geral da simulação com os dois ambientes interagindo foi satisfatória.

Segundo [63], planejadores de trajetórias podem ser adicionados para dar mais autonomia às embarcações, porém essas técnicas, assim como a detecção de obstáculos, não foram adotadas por questões de simplificação, visto que o foco do trabalho foi na obtenção de uma infraestrutura de comunicação para que uma frota de embarcações não tripuladas possam manter uma comunicação robusta com uma estação base. A implementação do arcabouço, no entanto, facilita modificações futuras, permitindo que as características do ambiente real possam ser acrescentadas gradualmente.

6 SIMULAÇÕES

Neste capítulo apresentamos e discutimos as simulações realizadas utilizando o NS-3 e o MATLAB de forma integrada. Os cenários de simulação contam, de modo geral, com: (i) uma estação base, fixa, que orchestra remotamente diversas embarcações; (ii) uma ou mais embarcações; (iii) uma rede sem fio padrão IEEE 802.11g em modo *ad hoc* como suporte de comunicação.

Observa-se que, nas simulações, as embarcações apresentam liberdade de movimento, sem vias ou obstáculos. Além disso, a generalidade e aprofundamento da semântica que define uma missão não faz parte desta proposta, pois isso iria requerer o desenvolvimento de uma gramática para expressar múltiplas atividades e sequências, por isso estamos considerando as seguintes características:

- a estação base consegue identificar unicamente e obter informações de cada embarcação. Com isso ela consegue enviar comandos independentemente para cada embarcação;
- no início da missão cada embarcação tem um ponto de partida (posição inicial definida no arquivo de configuração como no Algoritmo 6 no Apêndice A);
- todas as embarcações são capazes de identificar sua posição atual;
- cada embarcação deve ir autonomamente até a coordenada geográfica recebida da estação base;
- ao chegar na coordenada determinada, cada embarcação deve coletar dados dos sensores presentes na própria embarcação e enviar esses dados para a base durante um tempo definido pela mesma (não há efetivamente a coleta de dados, mas as embarcações permanecem em sua posição até que a estação base mande o comando para retornarem);
- em seguida, cada embarcação deve retornar para próximo da estação base, a estação base envia uma nova coordenada a ser atingida (sua própria posição);
- se a missão for desenvolvida por mais de uma embarcação, estas irão cercar ou chegar o mais próximo possível do um ponto alvo (coordenada geográfica).

Numa etapa inicial, alguns testes simples foram realizados para verificar a acurácia do protocolo e identificar os limites relacionados à velocidade das embarcações, potência de sinal Wi-Fi recebido e outras métricas da comunicação (Seção 6.2).

Com base nos resultados obtidos nos primeiros testes, outros cenários mais estruturados foram avaliados (Seção 6.3): (i) uma embarcação, inicialmente à uma distância de um metro da estação base, se afasta da estação base até perder a comunicação com a mesma, cenário similar ao utilizado nos testes iniciais; (ii) uma frota de embarcações formam uma linha, para aumentar o alcance geográfico da comunicação, entre a estação base e a coordenada alvo; (iii) uma frota de embarcações cercam a coordenada alvo e retornam para perto da estação base.

6.1 Parâmetros e Métricas Avaliadas

Em [75] é utilizada uma interface de rede sem fio [78] padrão IEEE 802.11a/b/g/n [26–30] que pode operar em duas frequências (2,4GHz e 5,0GHz). Baseado em suas características e de acordo com o propósito do trabalho foram definidas as características da rede, apresentadas na Tabela 6, utilizadas nas simulações. A frequência de operação escolhida está presente em dispositivos embarcados, como o ESP32 e suas variações [79], o que foi um fator determinante já que há o intuito de executar o protocolo em aplicações reais no futuro.

A altura da antena da estação base foi definida de acordo com o utilizado em [75], enquanto que a das embarcações levou-se em conta os recursos disponíveis do projeto [46] para uma possível execução de experimentos. A potência de transmissão e sensibilidade de recepção foram definidas de acordo com [15, 16, 75, 78], enquanto que os ganhos das antenas foram baseados em [16, 75]. O parâmetro de ruído (Noise Figure, NF) do canal e a perda total do sistema (equipamentos, cabos e conectores) foram escolhidos de acordo com [29, 75].

O simulador de redes utilizado necessita de dois modelos específicos para simular um canal de comunicação: o primeiro é o modelo de atraso (ou *delay*) que determina o tempo que leva para um sinal chegar até o destino; o segundo é o modelo de propagação que simula a propagação do sinal pelo respectivo canal. O NS-3 disponibiliza dois modelos de atraso: um de velocidade constante que depende apenas da distância entre os nós comunicantes e um parâmetro fixo referente a velocidade a ser utilizada; o outro é aleatório dependente apenas de um gerador de números aleatórios, não leva em consideração a distância entre os nós. Foi utilizado o modelo de velocidade constante utilizando a velocidade da luz no vácuo como parâmetro. Com relação ao modelo de propagação, o NS-3 apresenta diversos modelos implementados, no entanto foi necessário implementarmos um para refletir melhor as características da comunicação marítima (detalhes apresentados na Seção 5.1.2).

Além dos parâmetros utilizados, métricas de avaliação também tiveram que ser definidas.

Seguindo as métricas mais utilizadas, utilizou-se o tempo de ida e volta (Round Trip Time, RTT), a taxa de entrega de pacotes (Packet Delivery Ratio, PDR) e a taxa de transferência (*throughput*), métricas usadas em [16, 75].

Tabela 6 - Parâmetros Gerais.

Padrão Wi-Fi	IEEE 802.11g
Largura de banda do Canal	20 MHz
Frequência de operação	2,412 GHz
Taxa de transferência	6Mbps
Parâmetro de Ruído do canal - NF	10 dB
Perda total (equipamentos, cabos e conectores)	5 dB
Modelo de atraso (velocidade constante)	299.792.458 m/s
Modelo de propagação apresentado na Seção 5.1.2 com desvanecimento	
Potência de transmissão	20 dBm
Sensibilidade de recepção	-95 dBm
Ganho da antena da estação base	16 dBi
Ganho da antena das embarcações	10 dBi
Altura da antena da estação base	20 m
Altura da antena das embarcações	3 m
Velocidade máxima das embarcações	(5,5,0) m/s (x,y,z)

6.2 Avaliação Inicial

Os autores de [16, 75] realizaram simulações, cálculos e medições como forma de avaliação do modelo de propagação implementado (Seção 5.1.2), obtendo os resultados apresentados na Figura 22. Desta forma algumas avaliações iniciais foram feitas com o intuito de: (i) verificar se o modelo implementado iria apresentar a resposta de acordo com as referências utilizadas [16, 75]; (ii) avaliar quais as limitações dos enlaces de comunicação simulados; (iii) de posse dos resultados, poder definir melhor os cenários finais a serem avaliados; (iv) ter resultados preliminares que pudessem ser utilizados como fonte de comparação para os resultados obtidos nos cenários finais.

6.2.1 Primeiro Teste

Primeiro, para efeito de comparação com os resultados obtidos pelos autores de [16, 75], foi testado o modelo de propagação implementado (Seção 5.1.2). Para isso, foram fixados alguns parâmetros (Tabela 6) necessários para o cálculo da potência do sinal recebido e outros foram

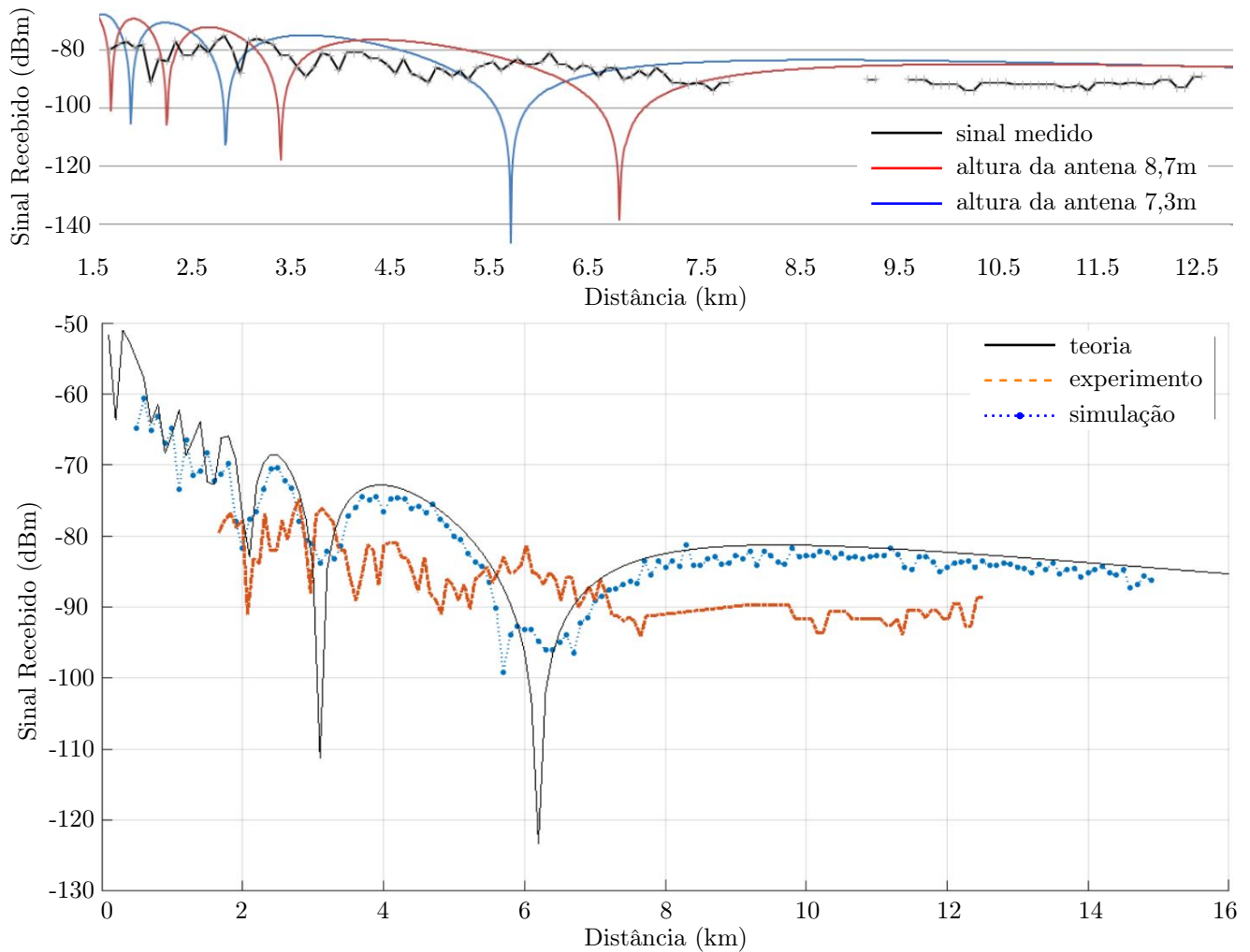


Figura 22 - Resultados Extraídos de [16,75], respectivamente (adaptados).

variados conforme a Tabela 7. Foram realizadas 30 rodadas, com sementes diferentes, do cálculo de potência do sinal recebido utilizando o modelo de propagação apresentado na Seção 5.1.2 para cada distância de referência. Um nó foi mantido fixo nas coordenadas (0,0), enquanto que o outro nó foi mantido fixo durante as 30 rodadas e para cada término das 30 rodadas sua posição foi incrementada em um metro no eixo “x” (o eixo “y” foi mantido em 0).

Alguns parâmetros foram variados independentemente, resultando em diferentes combinações. A altura de 8m e a frequência de 5,8GHz foram escolhidas de acordo com os testes realizados em [16,75] enquanto que a altura de 3m e frequência de 2,4GHz foram escolhidas de acordo com a Tabela 6, também foi utilizado o modelo com e sem a componente de desvanecimento. Decidiu-se variar a altura das antenas por questões de comparação com [16,75], a frequência foi variada apenas para termos valores iniciais para comparar com os testes e simulações posteriores.

Tabela 7 - Parâmetros para o Teste do Modelo de Propagação.

Frequência de operação	2,4 GHz e 5,8GHz
Altura da antena do transmissor	20 m
Altura da antena do receptor	3 m e 8 m
Distância entre os nós	1 m até 30 km
Modelo de propagação (Seção 5.1.2)	com e sem desvanecimento

A Figura 23 apresenta os resultados do primeiro teste, o qual mostra que a componente de desvanecimento provoca uma pequena variação não presente no gráfico sem a componente. É importante ressaltar que foi usado um intervalo de confiança de 95%, porém o erro é pequeno demais para aparecer nos gráficos. Os resultados estão compatíveis com os obtidos pelos autores [16, 75] como apresentado na Figura 22, quando usados os mesmos parâmetros (5,8GHz, altura do receptor de 8m, com e sem *fading*).

Assim como informado em [15], a mudança da altura das antenas (apesar de só termos variado uma das antenas) influencia diretamente nos resultados principalmente para as distâncias menores. Outro fato que pode ser observado é a diferença entre as duas frequências testadas, quanto menor a frequência mais rápido a curva converge e as oscilações no início dos gráficos causadas pela interferência dos raios refletidos apresentam uma área menor.

É possível notar que o modelo de dois raios modela a maior interferência causada pelos raios refletidos (Figura 20) em distâncias mais curtas causando os vales observados nos gráficos nas Figura 22 e Figura 23. Apesar de não existir esses vales tão acentuados nas curvas que representam os experimentos nos gráficos na Figura 22 é possível notar uma maior oscilação e variação para as distâncias menores do que 8km, aproximadamente.

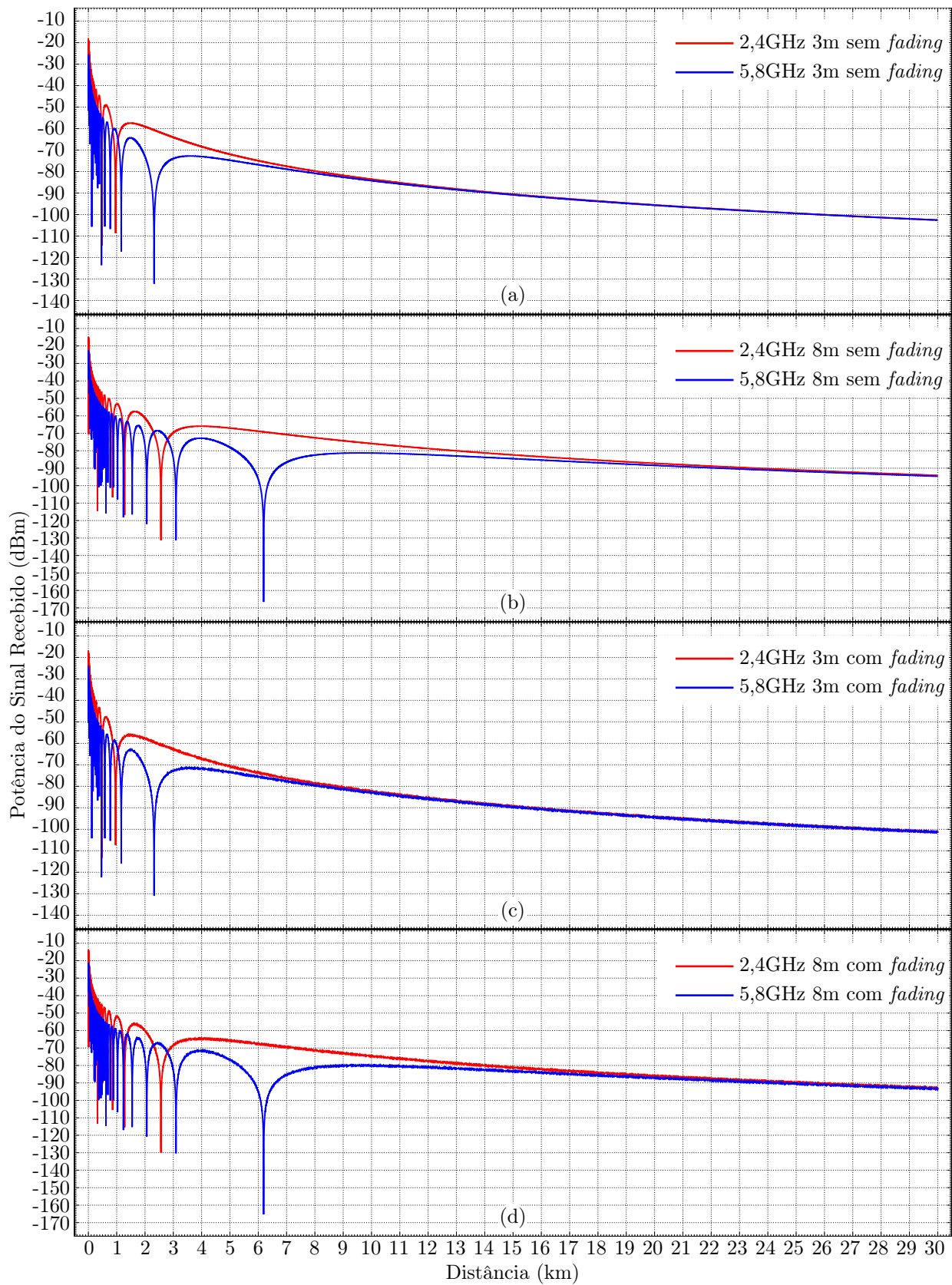


Figura 23 - Distância vs Potência do Sinal Recebido (Modelo na Seção 5.1.2). (a) à (d) combinam frequências de 2,4GHz e 5,8GHz, com e sem *fading*.

6.2.2 Segundo Teste

Com os resultados do modelo de propagação foi possível definir os parâmetros para a execução do segundo teste, o de comunicação. Nesse teste, a frequência de operação e a altura das antenas (estação base e embarcação) foram fixadas, assim como outros parâmetros gerais da comunicação como mostrado na Tabela 6. Na Tabela 8 estão listados os parâmetros que foram variados para a execução do teste de comunicação. O intuito desse teste foi verificar as limitações e possibilidades da comunicação proposta, incluindo mensagens em *broadcast* IP com TTL igual a um e mensagens com tamanhos de 132 *bytes*, 456 *bytes* e 2200 *bytes*. A escolha dos tamanhos das mensagens a serem avaliados foi feita de acordo com as definições do protocolo (Seção 4.3), sendo: 132 *bytes* o tamanho mínimo de um pacote de mensagem; 456 *bytes* de um pacote de mensagem contendo um pacote de telemetria com o tamanho máximo 2200*bytes* contemplando o tamanho máximo de um pacote de mensagem, contendo cinco pacotes de telemetria com tamanho máximo. A velocidade de deslocamento da embarcação também foi variada em 1m/s, 5m/s e 10m/s. A escolha das velocidades foi baseada na velocidade máxima de uma embarcação (5m/s) no cenário proposto, como mostrado na Tabela 6.

Tabela 8 - Parâmetros para o Teste de Comunicação.

Distância entre os nós	1 m até 30 km
Velocidade das embarcações	1m/s, 5m/s e 10m/s
Tamanho das mensagens	132 <i>bytes</i> , 456 <i>bytes</i> e 2200 <i>bytes</i>

As simulações foram feitas com três combinações de nós transmissor e nó receptor:

1. *B2V*, estação base enviando diretamente para uma embarcação;
2. *V2B*, uma embarcação enviando diretamente para estação base e
3. *V2V*, uma embarcação enviando diretamente para outra embarcação.

Para a realização do teste de comunicação foram realizadas 30 rodadas de simulação (mesma quantidade utilizada no primeiro teste). Em cada rodada um nó foi mantido fixo nas coordenadas (0,0), enquanto que o outro se deslocou com velocidade constante no eixo “x” (o eixo “y” foi mantido em 0) até a posição 30km (a mesma distância máxima usada no primeiro teste). O nó em movimento com velocidade constante foi configurado para enviar dez mensagens a cada metro. Ao término das

30 rodadas foram calculados, para cada metro com um intervalo de confiança de 95%, a potência do sinal recebido, o atraso, a taxa de entrega de pacotes e a taxa de transferência. Os resultados estão separados de acordo com a métrica avaliada e com o eixo “x” foi restrito até os 20km, para uma melhor visualização já que as curvas vão até no máximo os 13km.

Na Figura 24 são apresentados os gráficos referentes a potência do sinal recebido de acordo com a distância entre transmissor e receptor com mensagens de 2200bytes, subdivididos de acordo com os tipos de nós: (a) estação base transmitindo para a embarcação; (b) embarcação transmitindo para a estação base; (c) embarcação transmitindo para outra embarcação. É possível perceber que a velocidade apresentou pouca influência nos resultados, pois as curvas referentes as diferentes velocidades estão quase sobrepostas nos gráficos, porém a curva correspondente a velocidade de 10m/s foi a que mais se destacou por causar maior variação nos gráficos seguintes. Assim como os gráficos na Figura 23, o erro é muito pequeno e por isso não aparece nos gráficos. Também é possível notar o maior alcance na comunicação entre a estação base e a embarcação, independentemente de qual nó é o transmissor (estação base Figura 24(a) ou embarcação Figura 24(b)), o que era esperado já que a estação base apresenta antena com maior ganho e altura que a embarcação como mostrado na Tabela 6.

Os gráficos na Figura 24 apresentam uma oscilação no início causada pela reflexão dos raios (Figura 20) em distâncias menores que um limiar definido pela equação:

$$d = \frac{4h_t h_r}{\lambda}, \quad (21)$$

na qual d é a distância mínima, λ é o comprimento de onda (em metros) do sinal, h_t e h_r são as alturas das antenas (em metros) do transmissor e do receptor, respectivamente [15]. Esse fenômeno chega a causar a degradação total do sinal fazendo com que a comunicação em determinadas posições piore e em certos caso até totalmente inviável, como será visto nos gráficos seguintes.

Os gráficos seguintes apresentam os resultados obtidos no pior caso, transmissor com velocidade de 10m/s, subdivididos de acordo com o tamanho das mensagens enviadas: (a) mensagens de 132bytes; (b) mensagens de 456bytes; (c) mensagens de 2200bytes. Na Figura 25 são mostrados os gráficos do atraso, como o modelo do atraso de propagação utilizado é o de velocidade constante os gráficos apresentam um limite inferior linear que é bem visível nos gráficos da Figura 25(a e b) dado o tempo menor que um milissegundo para as mensagens de 132 e 456 bytes. O gráfico na Figura 25(c) apresenta um aumento de aproximadamente 10ms em relação aos outros dois devido ao

aumento no tamanho das mensagens. Além disso, o aumento linear observado na Figura 25(a e b) sofreu uma diminuição na curva e a variação ficou mais visível como visto na Figura 25(c). Como nos gráficos na Figura 24, a comunicação entre embarcações (curva azul) apresentou um alcance menor que a comunicação com a estação base (curvas verde e vermelha).

Na Figura 26 são mostrados os gráficos da taxa de entrega de pacotes apresentando uma taxa de entrega de quase 100% até a distância limite, quando começa a diminuir até ser nula. Assim como na Figura 24, a taxa de entrega apresenta oscilação no início para distâncias específicas correspondentes a maior interferência da reflexão dos raios (Figura 20) em distâncias menores. No caso da comunicação com a estação base essa influência chega a causar a perda total da comunicação para as distâncias aproximadas de 500m e 1km, já para a comunicação entre embarcações essa degradação causa uma redução na taxa de entrega mas a mesma ainda permanece acima dos 70%.

Os gráficos na Figura 27 mostram a taxa de transferência alcançada para cada distância analisada. Assim como nos gráficos da taxa de entrega, a taxa de transferência se manteve em seus valores máximos até alcançarem o limite máximo da comunicação com valores nulos nas distâncias de 500m e 1km para as comunicações com a estação base. O aumento do tamanho das mensagens possibilitou um aumento de quase 500kbps entre as mensagens de 132 *bytes* na Figura 27(a) e 2200 *bytes* na Figura 27(c) possibilitando chegar próximo aos 6Mbps definido pelo padrão IEEE 802.11g para comunicação por difusão [29].

Observa-se que dentro do raio de comunicação e sem outros nós competindo (cenário ótimo) a comunicação apresenta resultados dentro do esperado, mesmo com para a velocidade de 10m/s. O maior atraso permaneceu abaixo de 11ms levando em consideração o limite máximo da comunicação, o que é satisfatório, pois o deslocamento máximo será de poucos centímetros com tempo suficiente para uma possível ação, caso seja necessário.

Todos os gráficos apresentaram falhas na comunicação nas distâncias 500m e 1km para a comunicação entre a estação base e a embarcação e uma pequena degradação próximo as distâncias de 50m e 200m, fenômeno causado pela maior interferência dos raios refletidos (Figura 20) nas distâncias menores que o limiar dado pela Equação (21). Esse fenômeno pode ser contornado usando a Equação (21) para limitar a distância mínima entre os nós e incluindo uma ação de afastamento caso esse limite seja alcançado. O gráfico da potência de sinal recebido seria uma boa escolha como entrada para ações de controle, desde que restrita a valores maiores que o limiar dado pela Equação (21).

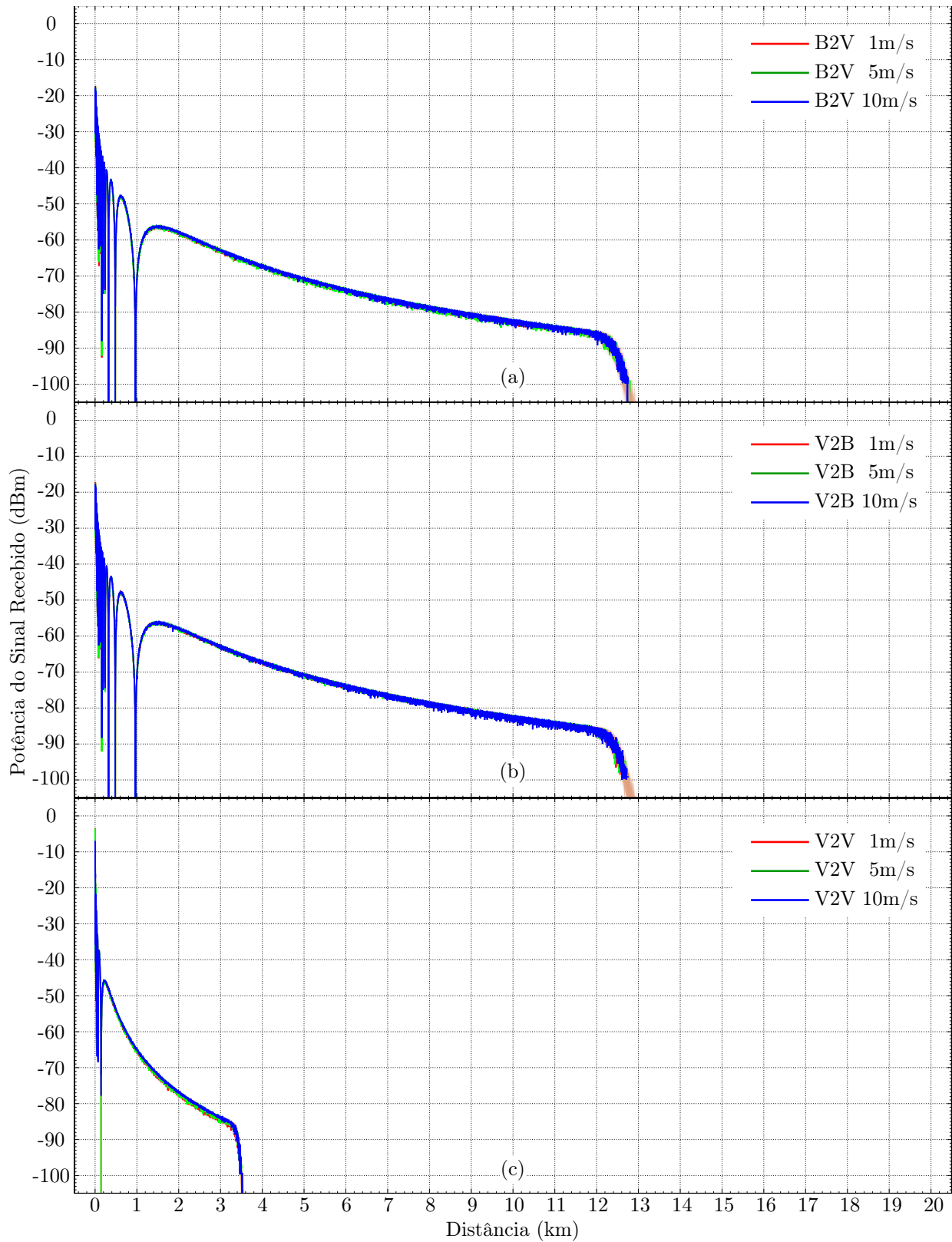


Figura 24 - Teste de Comunicação: Potência de Sinal Recebido (tamanho de 2200 *bytes*), considerando velocidades de 1, 5 e 10m/s. (a) Estação base para embarcação; (b) Embarcação para estação base e (c) Embarcação para embarcação.

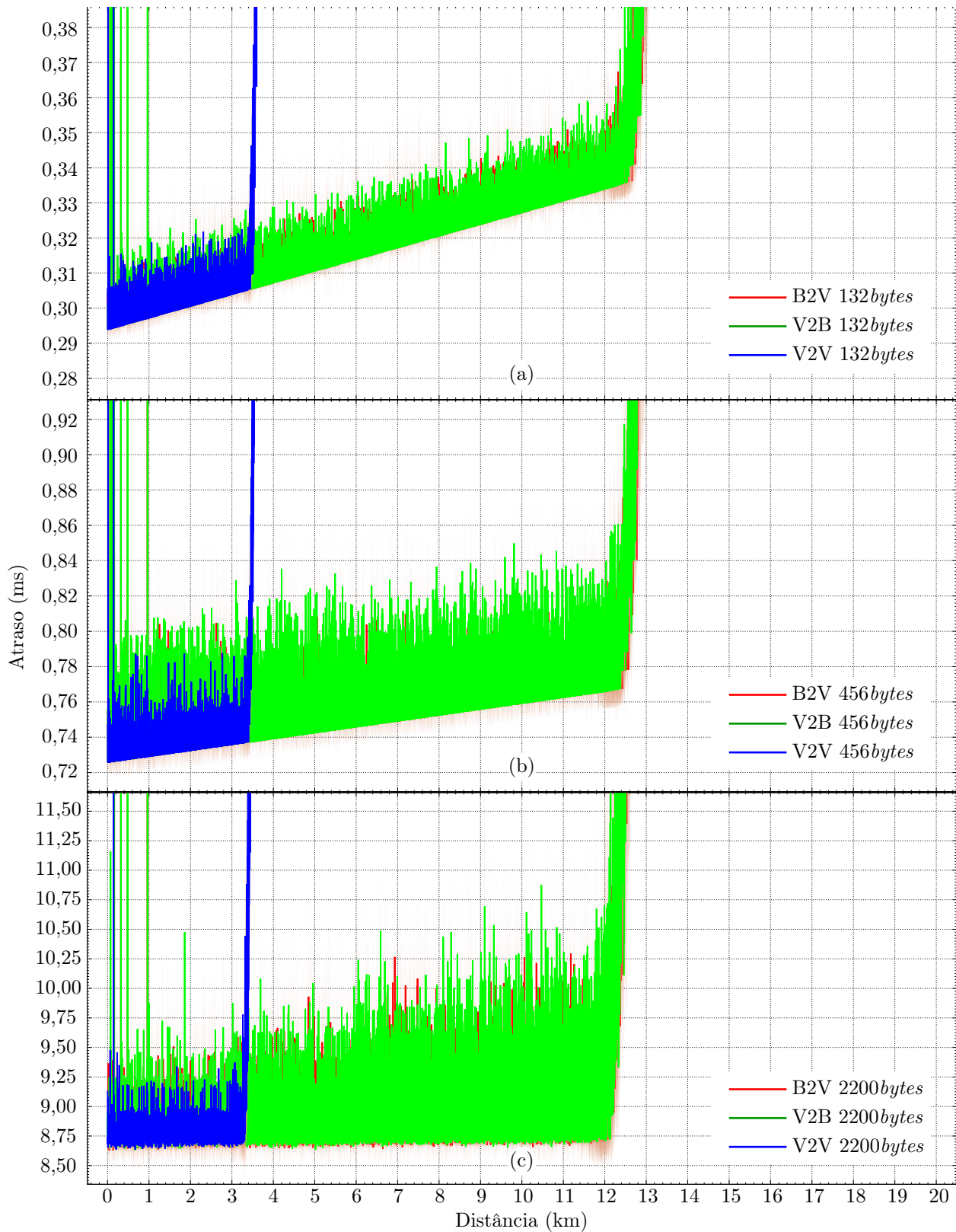


Figura 25 - Teste de Comunicação: Atraso, considerando a velocidade de 10m/s para os três tipos de teste (B2V, V2B e V2V). (a) Tamanho de 132 bytes; (b) Tamanho de 456 bytes e (c) Tamanho de 2200 bytes.

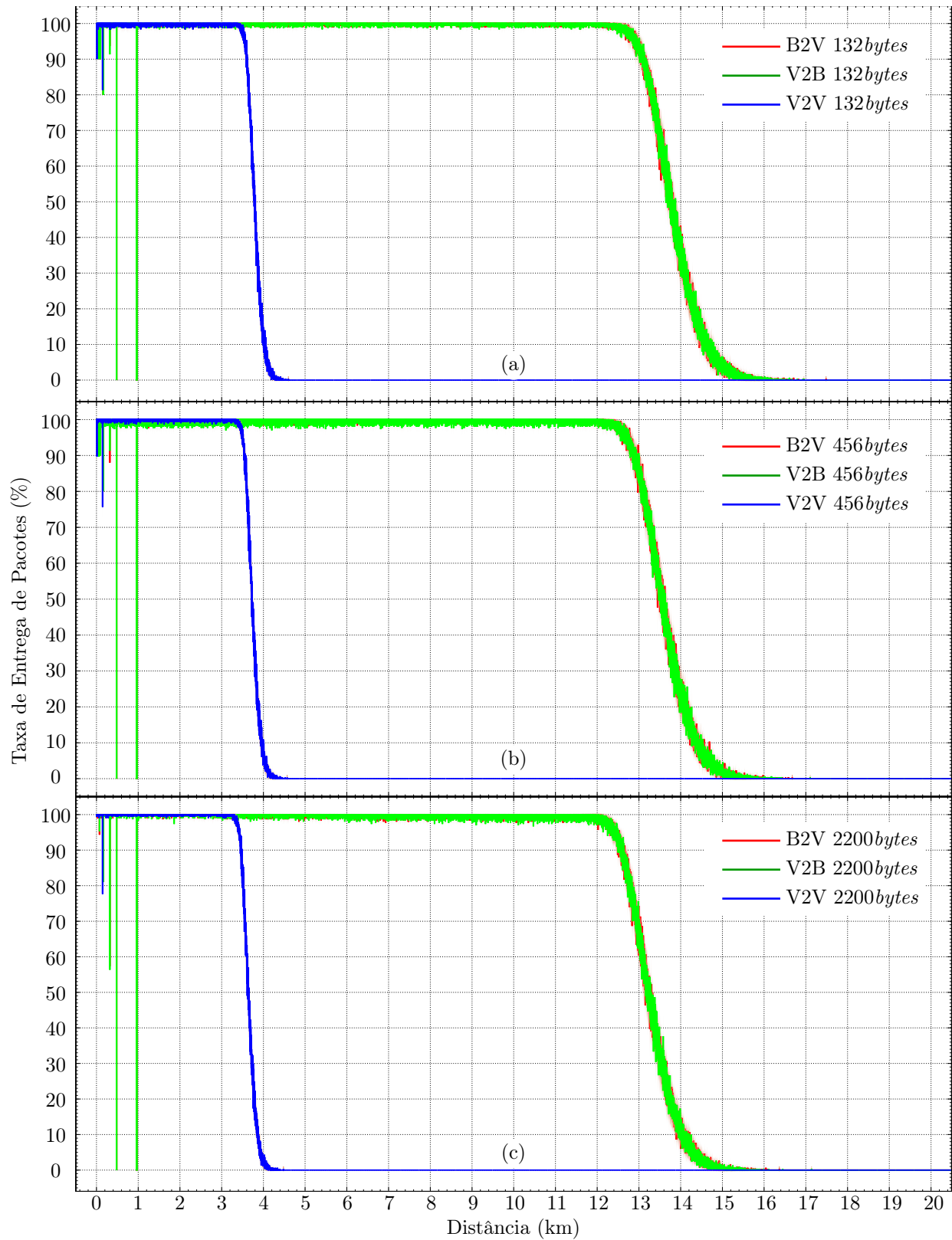


Figura 26 - Teste de Comunicação: Taxa de Entrega de Pacotes, considerando-se a velocidade de 10m/s e os três tipos de teste (B2V, V2B e V2V). (a) Tamanho de 132 bytes; (b) Tamanho de 456 bytes e (c) Tamanho de 2200 bytes.

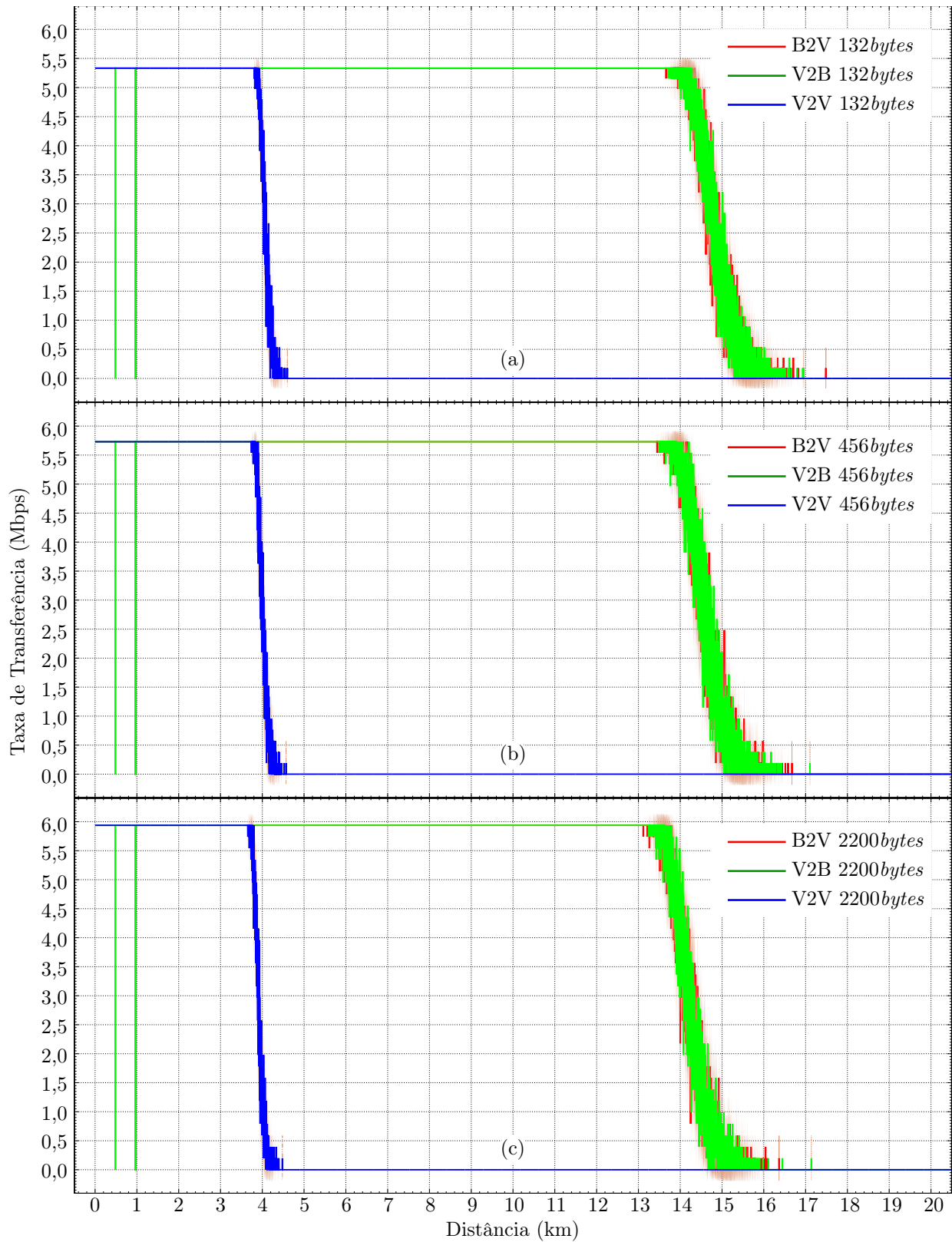


Figura 27 - Teste de Comunicação: Taxa de Transferência, considerando-se a velocidade de 10m/s e os três tipos de teste (B2V, V2B e V2V). (a) Tamanho de 132 bytes; (b) Tamanho de 456 bytes e (c) Tamanho de 2200 bytes.

6.3 Cenários

Com os resultados apresentados na Seção 6.2 foi possível estabelecer os parâmetros de restrição a serem respeitados pelo modelo cinemático e controle implementados (Seção 2.4) e definir os cenários a serem simulados como prova de conceito do funcionamento tanto do protocolo quanto do uso das métricas de qualidade da comunicação como entrada para o controle das embarcações. Assim, na Tabela 9 são apresentados os parâmetros utilizados para compor os cenários de simulação abordados, lembrando que os parâmetros que definem as interfaces de comunicação e as características dos elementos foram apresentados na Tabela 6.

Os cenários de simulação, Tabela 9, foram definidos com base nos dois cenários propostos (formação em linha até o destino como na Figura 28 e formação circular ao redor do destino como na Figura 29). Em todas as simulações foram realizadas 30 rodadas de simulação variando a semente de aleatoriedade e calculadas as médias para um intervalo de confiança de 95%. Como já informado na Seção 4.1, apenas uma estação base com posição fixa em (0,0), coordenadas 'x' e 'y' respectivamente, foi utilizada em todos os cenários.

Tabela 9 - Parâmetros da Simulação em cada Cenário.

Cenário 1: Embarcação se afastando da estação base	
Quantidade de embarcações	1
Tipo de missão	distância máxima em direção ao destino
Coordenadas iniciais da embarcação	(1,0)
Coordenadas do destino	(30000,0)
Cenário 2: Embarcações alinhadas para aumentar raio de atuação	
Quantidade de embarcações	5
Tipo de missão	distância máxima em direção ao destino
Coordenadas iniciais das embarcações	(12000,0), (15000,0), (18000,0), (21000,0), (24000,0)
Coordenadas do destino	(28000,0)
Cenário 3: Cercar um ponto	
Quantidade de embarcações	5
Tipo de missão	circundar o destino
Coordenadas iniciais das embarcações	(12000,0), (12000,2000), (12000,4000), (12000,-2000), (12000,-4000)
Coordenadas do destino	(15000,0)

No primeiro cenário foi utilizado apenas uma embarcação para comparar com os testes iniciais da comunicação (Seção 6.2.2), mas desta vez incluídos todos os mecanismos do protocolo proposto. A posição inicial da embarcação é a mesma utilizada nos testes iniciais com o intuito de reproduzir um cenário idêntico aos testes iniciais da comunicação. A embarcação se desloca livremente até a

posição de destino depois de recebido o comando da estação base, a comunicação entre os dois nós se mantém segundo o protocolo até que seja alcançado o limite máximo quando então os pacotes enviados pela embarcação são perdidos, mas o protocolo continua executando com a embarcação se deslocando e enviando mensagens, mesmo sem retorno, até que chegue na posição destino quando a simulação termina.

As posições iniciais dos cenários 2 e 3 foram escolhidas levando em consideração o alcance máximo da estação base de acordo com os resultados do cenário 1 apresentados na Seção 6.3.1 e para o alcance máximo entre embarcações foram utilizados os resultados apresentados na Seção 6.2.2. Assim foi utilizado o valor de 12km para a distância máxima de comunicação com a estação base e o valor de 3km para a comunicação entre embarcações, mantendo em ambos os casos uma taxa de entrega próxima dos 100%.

No segundo cenário foram utilizadas 5 embarcações posicionadas previamente com distâncias de acordo com os resultados anteriores, sendo entre a estação base e uma embarcação cerca de 12km e entre embarcações 3km. Essas distâncias foram usadas para ampliar ao máximo o raio de comunicação como mostra a Figura 28. O objetivo foi avaliar como as embarcações atuando ativamente como nó intermediário se comportam, bem como verificar como a última embarcação percebe a comunicação com a estação base. As embarcações ficam paradas durante 5 minutos apenas trocando mensagens com a estação base de acordo com a execução do protocolo. Decorrido os cinco minutos o protocolo entra na fase de finalização com as embarcações recebendo o comando de finalização da estação base, que ao final também termina e a simulação chega ao fim.

No terceiro cenário foram utilizadas 5 embarcações posicionadas inicialmente a aproximadamente 12km da estação base e distantes 2km umas das outras. A missão se inicia com as embarcações nas posições iniciais, a estação base informa a missão e as embarcações se deslocam até os respectivos destinos, posições ao redor do ponto de destino como na Figura 29. Por padrão, ao chegarem aos seus respectivos destinos, as embarcações informam seu estado à estação base, que por sua vez ao perceber que todas chegaram ao destino, ativa um temporizador de 10 segundos para então mudar de estado e enviar o comando de retorno. Ao receber o comando de retorno as embarcações então voltam em direção a estação base até chegarem a uma distância máxima de 10km da mesma.

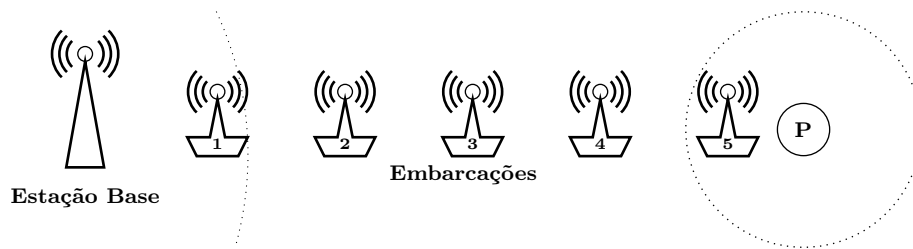


Figura 28 - Cenário 2: Formação com 5 Embarcações em Linha até o Destino.

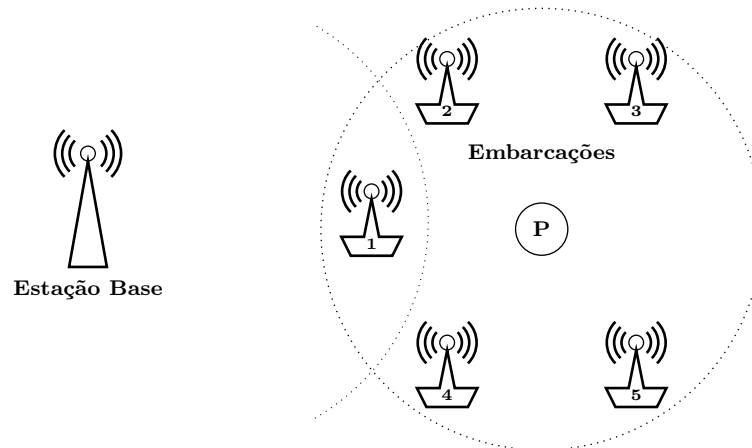


Figura 29 - Cenário 3: Formação com 5 Embarcações ao Redor do Destino.

6.3.1 Resultado do Cenário 1

O gráfico na Figura 30 mostra o deslocamento da embarcação ao longo do tempo, deslocamento apenas no eixo “x”, com velocidade constante. Os gráficos do atraso de ida e volta e da taxa de entrega percebidos pela embarcação de acordo com sua distância até a estação base são apresentados na Figura 31 e Figura 32, respectivamente.

O RTT é computado quando a embarcação recebe a mensagem de confirmação da estação base e para cada mensagem perdida é estimado um tempo cinco vezes maior que o último tempo computado, para cada metro é calculada a média das amostras computadas e por fim foi feito o cálculo do intervalo de confiança das 30 rodadas de simulação realizadas. As mensagens trocadas nos cenários estruturados apresentam tamanhos variáveis de acordo com a necessidade das aplicações, no caso, orquestrados pela estação base, ou por conta do protocolo desenvolvido, impactando no valor do atraso de ida e volta. O cálculo da taxa de entrega é a razão entre a quantidade de mensagens recebidas e quantidade de mensagens enviadas pela embarcação e foi calculado em cada metro ao longo da trajetória apresentada na Figura 30.

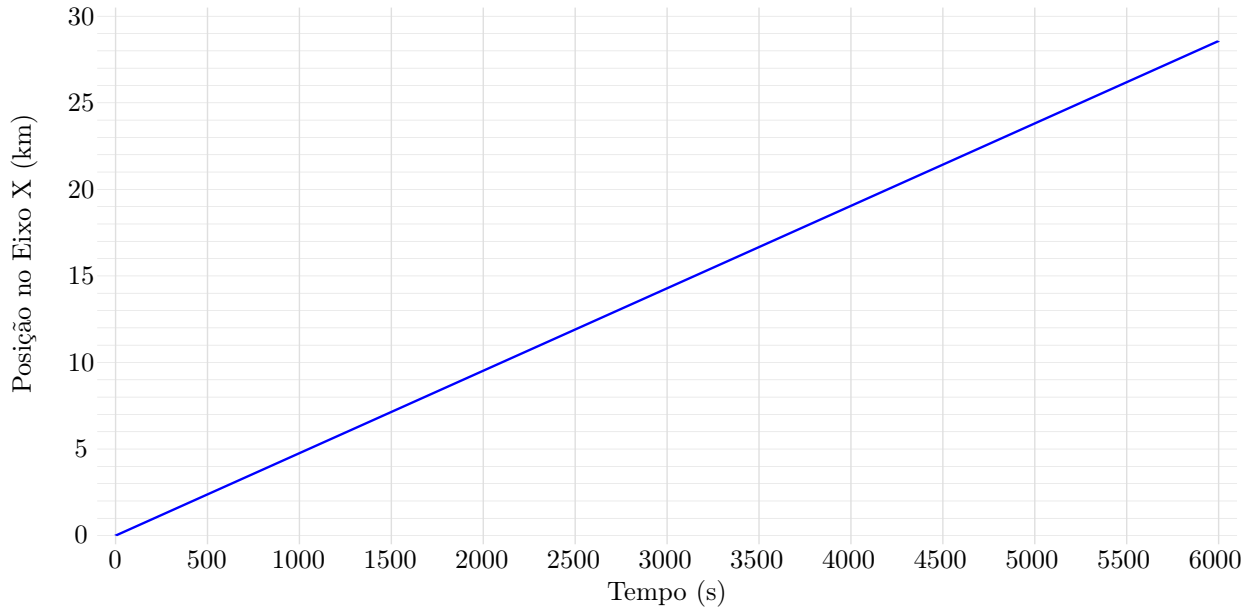


Figura 30 - Cenário 1: Deslocamento ao Longo do Tempo (Imagem Obtida no MATLAB).

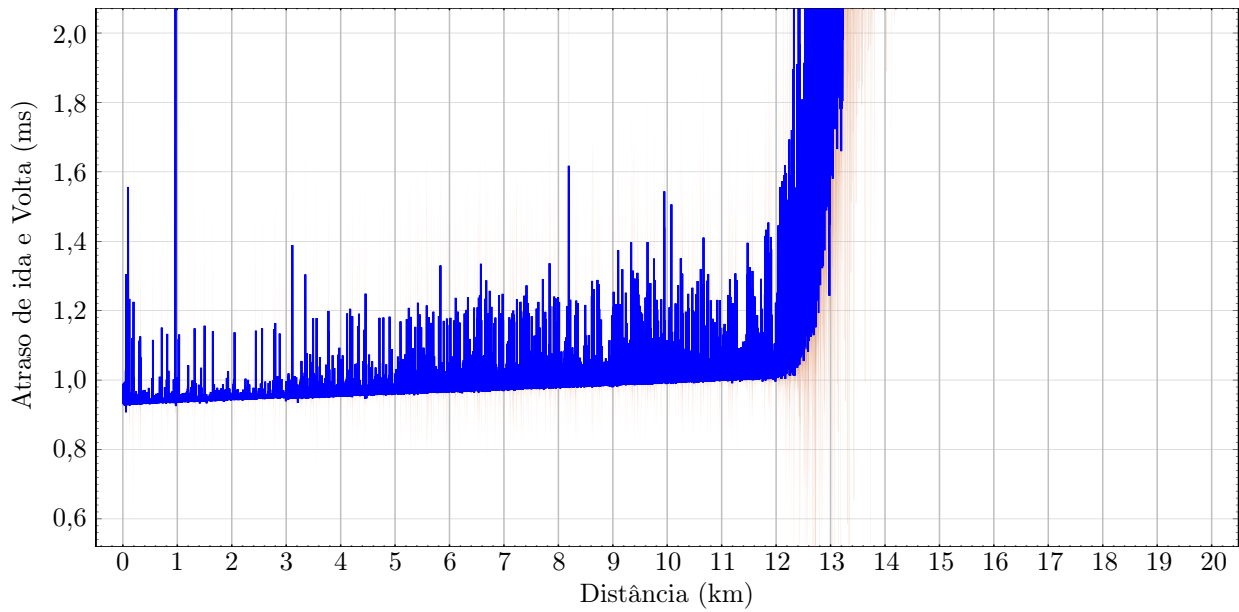


Figura 31 - Cenário 1: Atraso de Ida e Volta (RTT).

O PDR foi computado a cada metro e foi feito o cálculo da média com intervalo de confiança das 30 rodadas de simulação realizadas. É possível notar que a partir aproximadamente dos 12km a taxa de entrega começa a diminuir e o tempo de resposta a aumentar, como também observado nos resultados apresentados na Seção 6.2.2. O atraso de ida e volta se mantém abaixo dos 2 milissegundos e a taxa de entrega acima dos 80%, há uma exceção apenas em 1km que devido a

maior interferência dos raios refletidos, como informado na Seção 6.2, a taxa de entrega é nula e o tempo é infinito.

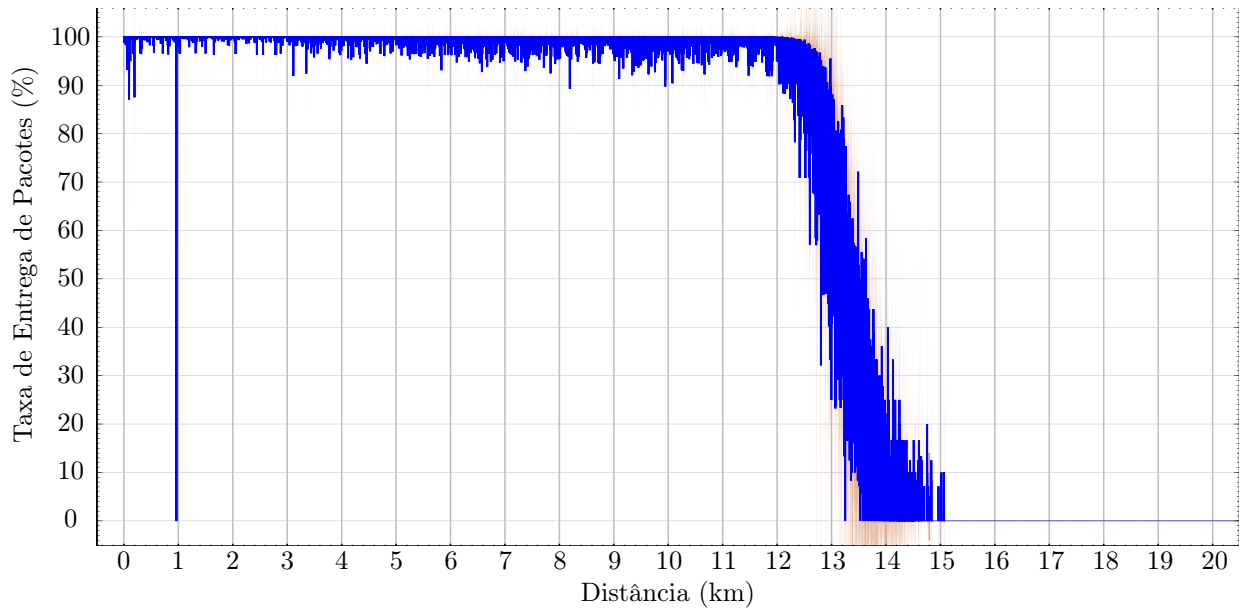


Figura 32 - Cenário 1: Taxa de Entrega de Pacotes (PDR).

Este primeiro cenário mostrou-se dentro do esperado em comparação com o teste de comunicação apresentado na Seção 6.2.2, mesmo levando em consideração as particularidades do protocolo.

6.3.2 Resultado do Cenário 2

Neste cenário foi estudada a possibilidade de aumentar o alcance da comunicação com a estação base utilizando o recurso de múltiplos saltos característico de redes *ad hoc*. Na Figura 33 é apresentada a configuração do cenário avaliado, destacando a posição da estação base com o símbolo ● em preto, do destino a ser alcançado com o símbolo × e das embarcações com o símbolo ● colorido. As embarcações foram posicionadas a uma distância de 3km entre elas e a primeira embarcação (1) a 12km da estação base. Com essa configuração a última embarcação (5) apresentou o dobro da distância da primeira embarcação (1) para a estação base, ou seja o alcance da comunicação foi dobrado em relação ao cenário com apenas uma embarcação apresentado na Seção 6.3.1.

Os gráficos do atraso de ida e volta e da taxa de entrega percebidos pelas embarcações juntamente com a média das cinco embarcações estão apresentados na Figura 34 e na Figura 35, respectivamente. Sendo os primeiros cinco gráficos os resultados observados por cada embarcação

com suas cores correspondentes na Figura 33. Os gráficos inferiores na Figura 34 e na Figura 35 correspondem a média dos resultados observados pelas embarcações.

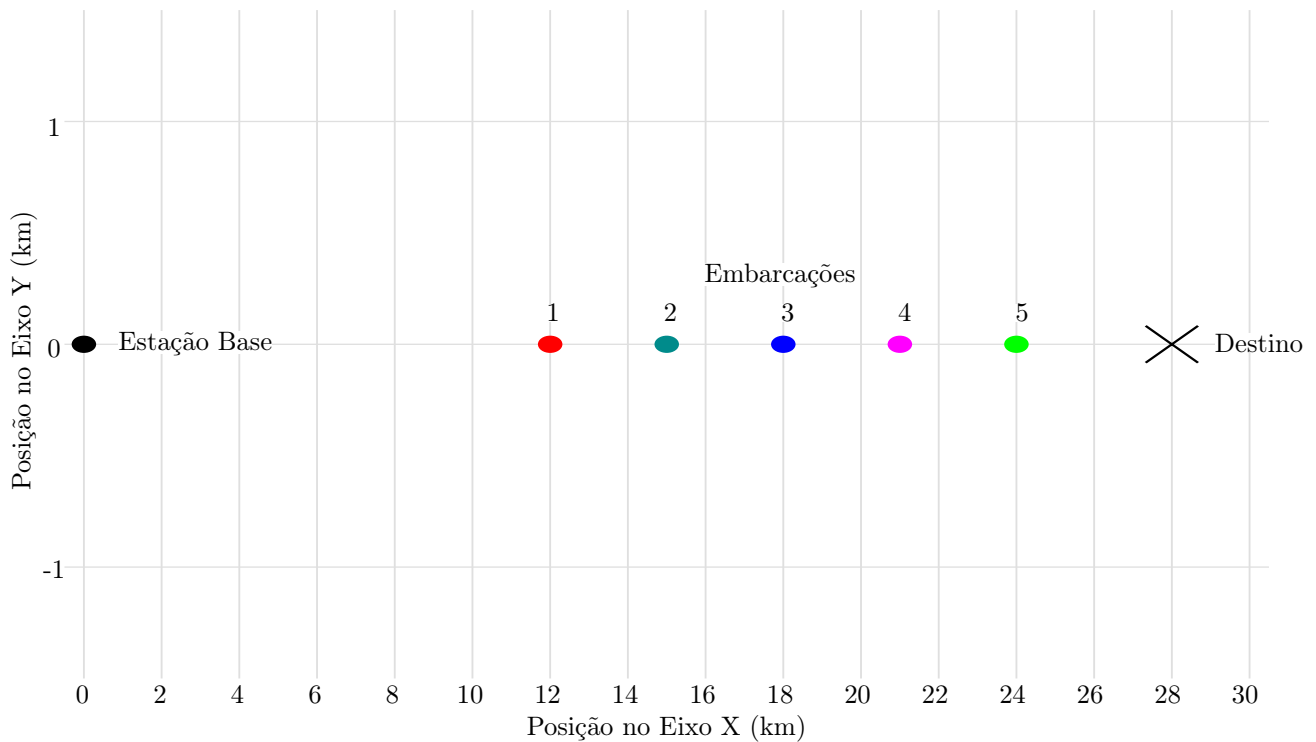


Figura 33 - Cenário 2: Posicionamento das Embarcações (Imagem Obtida no MATLAB).

Os resultados para este cenário mostraram que o RTT aumenta com o número de embarcações intermediárias, porém a embarcação mais próxima da estação base (embarcação 1) apresentou RTT próximo do observado no cenário com uma embarcação (Figura 31). As demais embarcações apresentaram um aumento considerável de quase 50 milissegundos de diferença entre as três primeiras embarcações (embarcações 1, 2 e 3) e de quase 100 milissegundos entre as três últimas (embarcações 3, 4 e 5). Além disso, a média das cinco embarcações se manteve abaixo dos 300 milissegundos e abaixo dos 400 milissegundos para a embarcação mais distante (embarcação 5). O erro também sofreu um aumento quanto mais distante a embarcação se encontrava da estação base fazendo com que fosse necessário um número maior de nós intermediários o que é comprovado com a diminuição da taxa de entrega de pacotes observada na Figura 35.

O PDR se manteve dos 80% mesmo para a embarcação mais distante da estação base (embarcação 5). Assim como para o RTT, a embarcação mais próxima da estação (embarcação 1) manteve um PDR acima dos 90%. O PDR se manteve acima dos 90% para as três primeiras embarcações (embarcações 1, 2 e 3) mais próximas da estação base e para as demais (embarcações 4 e 5) o PDR

foi mantido acima dos 80%.

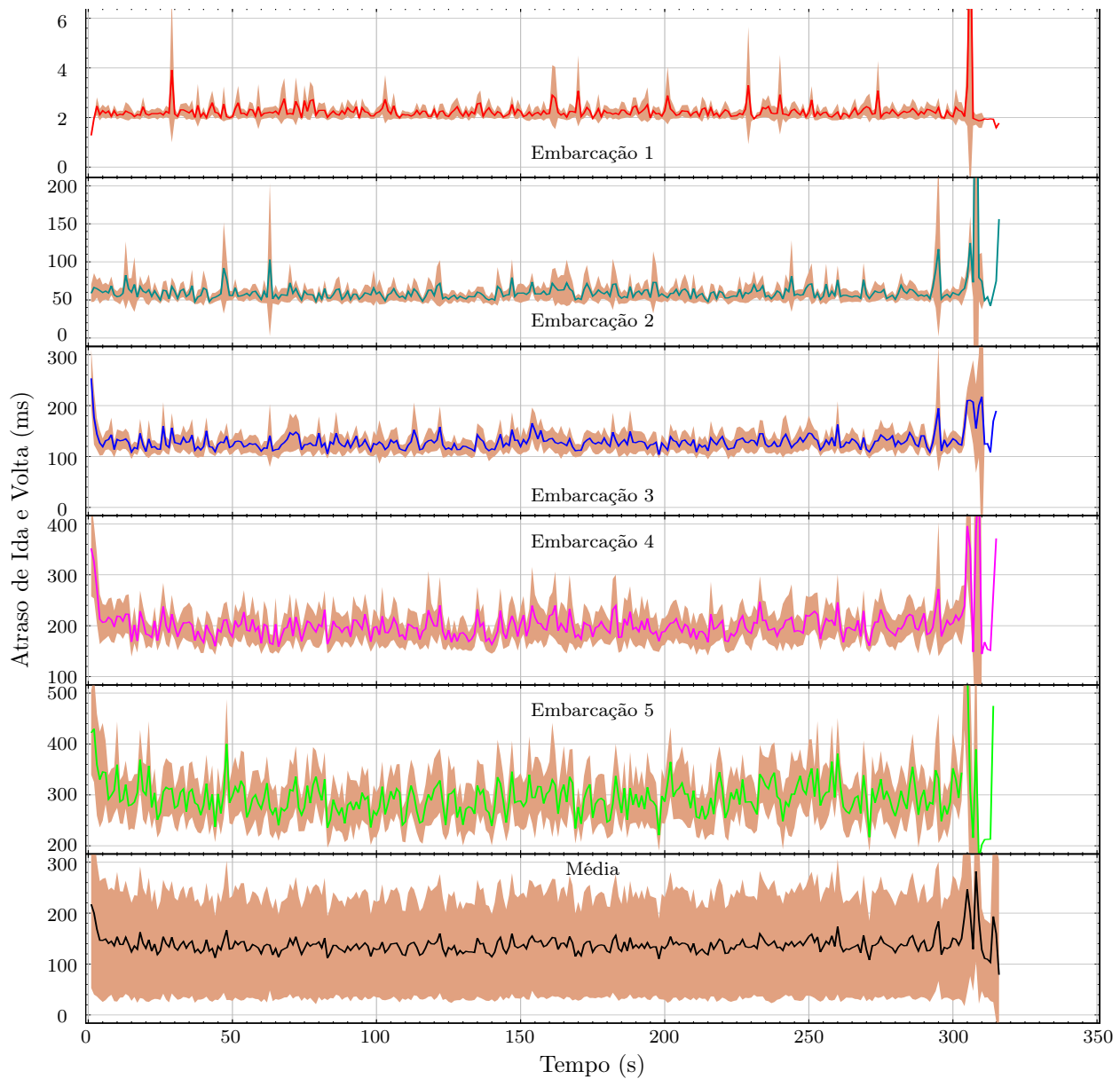


Figura 34 - Cenário 2: Atraso de Ida e Volta (RTT).

Os gráficos na Figura 34 e na Figura 35 apresentam uma variação maior depois dos 305 segundos que é causada pela rotina de finalização do protocolo e pelo maior tempo necessário para a finalização de uma das rodadas de simulação. A rotina de finalização do protocolo envia a última mensagem e finaliza sem aguardar a confirmação da estação base, por este motivo próximo dos 305 segundos há uma mudança repentina no PDR e no RTT, pois as mensagens de confirmação não são recebidas. Além disso, uma das rodadas de simulação apresentou maior tempo para que as embarcações fossem finalizadas, passando dos 305 segundos observados nas demais rodadas.

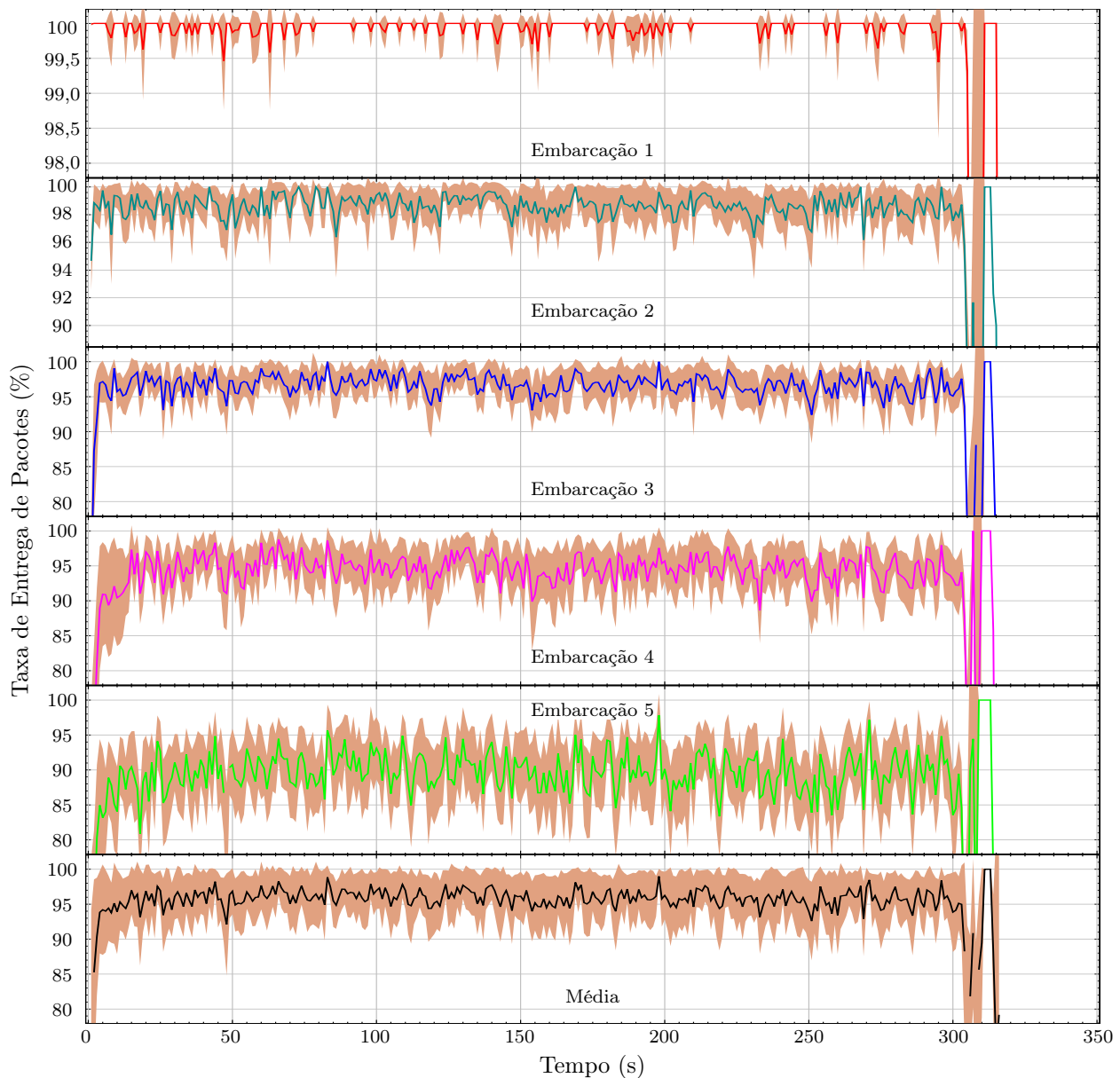


Figura 35 - Cenário 2: Taxa de Entrega de Pacotes (PDR).

Os gráficos na Figura 36 apresentam a quantidade de nós vizinhos identificados por cada embarcação ao longo do tempo. Foi observado que o gráfico da segunda embarcação mais próxima da estação (embarcação 2) apresentou uma variação na quantidade de vizinhos detectados que foi causado sua distância em relação a estação base que segundo os resultados apresentados na Seção 6.3.1 apresenta uma baixa taxa de entrega e alto atraso. O baixo PDR e alto RTT com a estação base causou um baixo recebimento de mensagens sem passarem pelo nó intermediário anterior (embarcação 1) fazendo com que a estação base fosse detectada em intervalos aleatórios o que causou uma variação na quantidade de nós detectados.

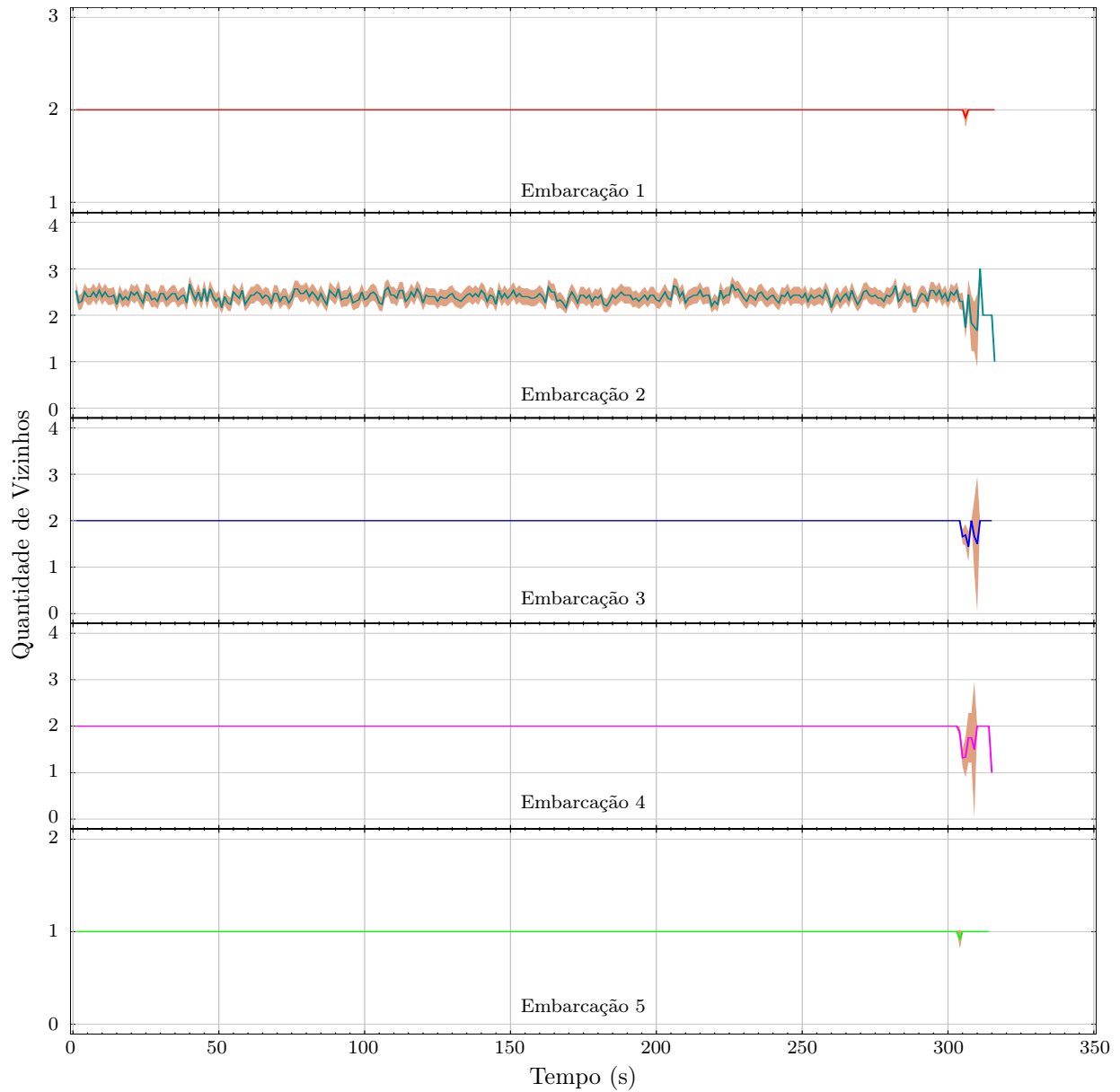


Figura 36 - Cenário 2: Vizinhança sem Média Móvel.

Para uma melhor visualização da quantidade de nós vizinhos ao longo do tempo de cada embarcação, principalmente para a embarcação 2, foi aplicada uma média móvel nos últimos 120 segundos sobre os gráficos na Figura 36 levando aos gráficos na Figura 37. O gráfico relativo aos vizinhos identificados pela embarcação 2 na Figura 37 inicia com o valor 3 e pouco tempo depois passa o valor 2, isso é causado pela pouca quantidade de amostras antes dos 5 segundos, fazendo com que a identificação da estação base seja contabilizada identificando assim 3 nós vizinhos. Depois de uma quantidade mínima de amostras a média móvel consegue manter o valor em 2, identificando assim as embarcações 1 e 3.

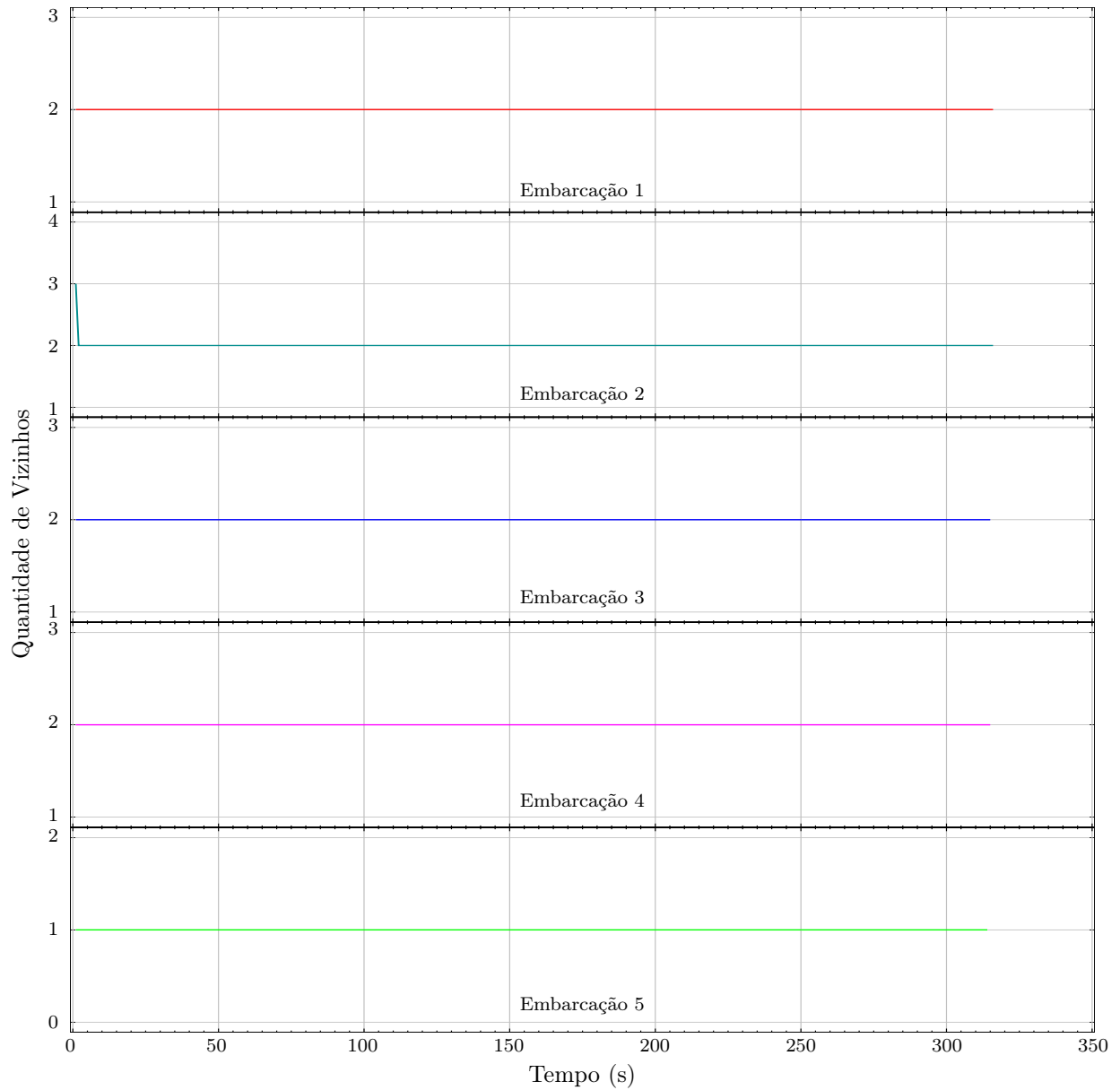


Figura 37 - Cenário 2: Vizinhança com Média Móvel dos Últimos 120 segundos.

Este cenário permitiu aumentar o alcance da comunicação com a estação base usando embarcações intermediárias sendo possível manter a comunicação entre a estação base e a embarcação com o dobro da distância máxima observada no primeiro cenário (12km na Figura 32) com uma taxa de entrega acima de 80% e atraso menor que 500 milissegundos.

6.3.3 Resultado do Cenário 3

Neste teste foi avaliada a comunicação das embarcações em um cenário mais próximo de uma missão estruturada, na qual o conjunto de embarcações se desloca (depois de receber o comando da estação base) até uma posição ao redor das coordenadas do destino. A trajetória executada é apresentada na Figura 38, as embarcações iniciam nas posições indicadas pelo símbolo ●, ao receberem o comando da estação base se deslocam para as posições indicadas pelo símbolo ■ cercado a posição de destino indicada pelo símbolo ✕, para finalizar a estação base mantém as embarcações durante 10 segundos em sua posições ao redor do destino para então enviar o comando para que as mesmas possam retornar para posições próximas a ela como indicam o símbolo ★. A embarcação 1 apresenta as duas primeiras posições coincidentes, pois o símbolo ■ está sobre o ● e por isso este não aparece.

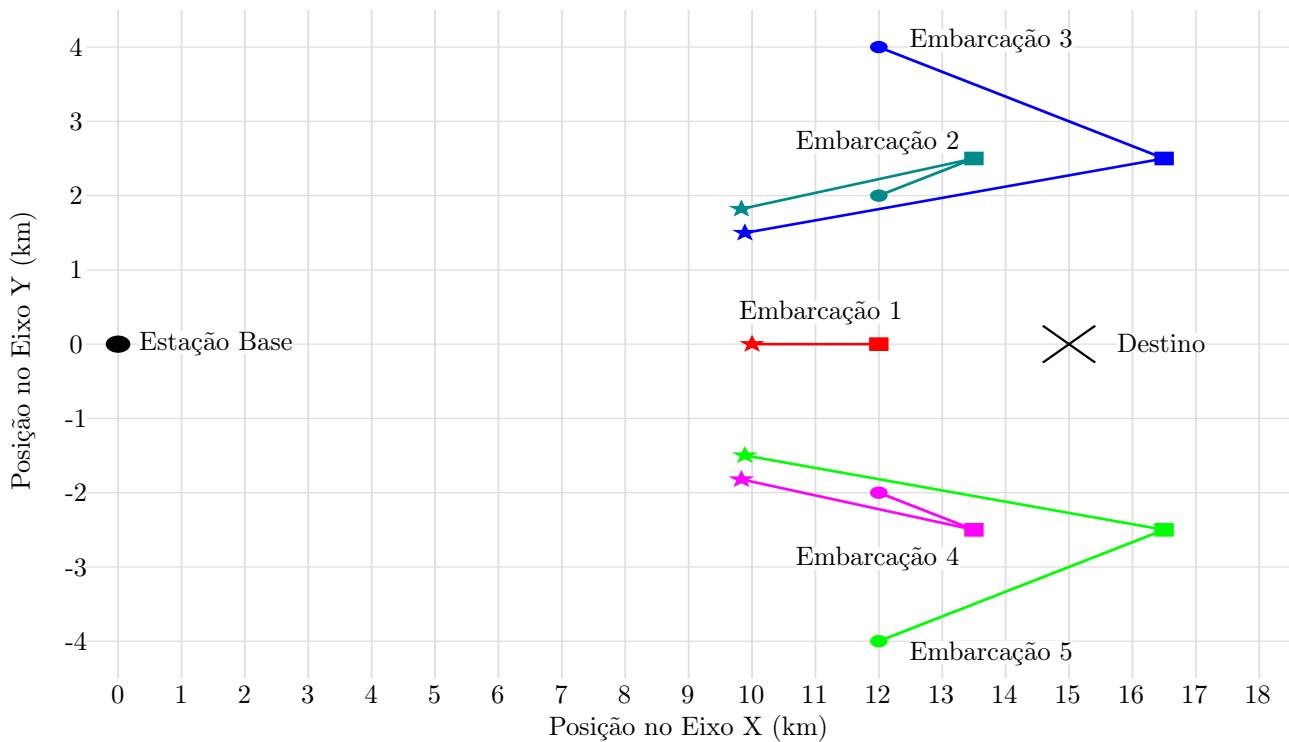


Figura 38 - Cenário 3: Trajetórias Executadas (Imagem Obtida no MATLAB).

Os gráficos do atraso de ida e volta e da taxa de entrega percebidos pelas embarcações juntamente com a média das cinco embarcações estão apresentados na Figura 39 e na Figura 40, respectivamente. Sendo os primeiros cinco gráficos os resultados observados por cada embarcação com suas cores correspondentes na Figura 38. Os gráficos inferiores na Figura 39 e na Figura 40

correspondem a média dos resultados observados pelas embarcações.

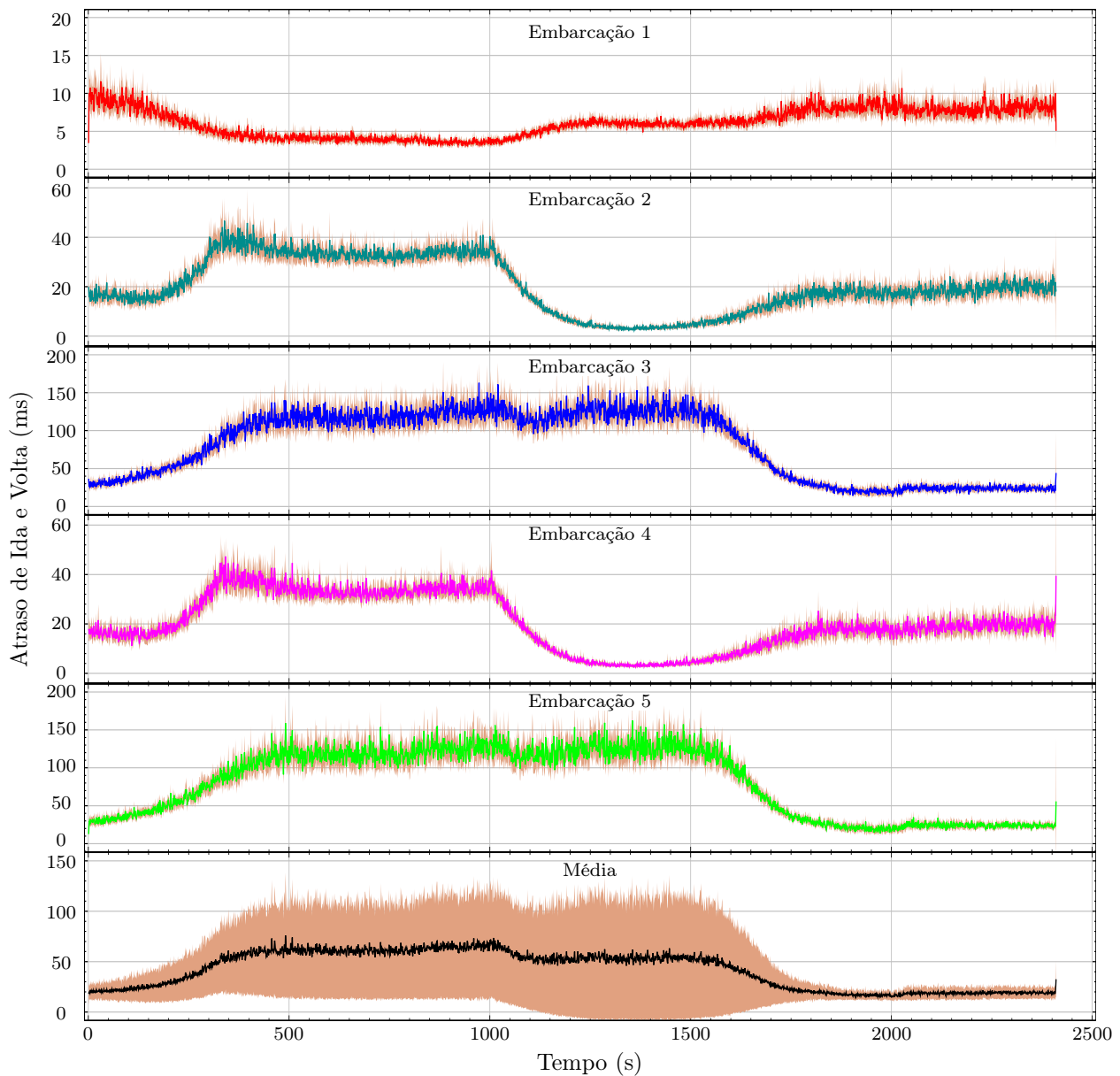


Figura 39 - Cenário 3: Atraso de Ida e Volta (RTT).

Em poucos segundos todas as embarcações já receberam o comando da estação base para se deslocarem para seus respectivos destinos. A embarcação 1 é a única que permanece na posição inicial o que faz com que seu seu RTT diminua e seu PDR aumente conforme as demais embarcações se afastam, como observado em seus respectivos gráficos na Figura 39 e na Figura 40. O atraso chega aos 4 milissegundos que devido ao maior número de vizinhos é maior do que os 2 milissegundos observados pela embarcação com a mesma posição no cenário 2. Próximo aos 1000 segundos todas as embarcações se encontram em suas posições intermediárias como indicado pelo símbolo ■ na Figura

38 e logo após a estação base envia o comando para que todas retornem, por isso seu RTT passa ter um leve aumento com a aproximação das demais embarcações mas é mantido abaixo dos 10 milissegundos. O PDR começa a diminuir próximo dos 1500 segundos, quando as embarcações 3 e 5 se aproximam chegando distância aproximada de 4km que como visto na Seção 6.2.2 há uma comunicação mesmo com uma taxa de entrega menor que 10%.

As embarcações 2 e 4 apresentam posições simétricas o que acarretou em gráficos idênticos, assim como as embarcações 3 e 5. O RTT das embarcações 2 e 4 inicia abaixo dos 20 milissegundos, sofre uma leve redução de aproximadamente menos de 2 milissegundos e começa a subir chegando próximo aos 40 milissegundos aos 350 segundos que é quando as embarcações chegam em suas posições intermediárias indicadas pelo símbolo ■ na Figura 38. As embarcações 3 e 5 se aproximam das embarcações 2 e 4 até os 350 segundos, depois passam a se afastarem o provocando a diminuição do RTT das embarcações 2 e 4 para aproximadamente 36 milissegundos até os 1000 segundos, quando então chegam em suas posições intermediárias indicadas pelo símbolo ■ na Figura 38. Elas são as únicas que ficam totalmente fora do alcance da estação base quando alcançada a posição intermediária intermediária indicada pelo símbolo ■ na Figura 38.

Aos 1500 segundos as embarcações 3 e 5 chegam a aproximadamente 14km da estação base e aos 2000 segundos aos 12km, por isso seu RTT diminui gradativamente durante esse intervalo permanecendo então em aproximadamente 20 milissegundos. Como pode ser observado na Figura 39 aproximadamente a partir dos 2000 segundos todas as embarcações encontram-se dentro do alcance da estação base (distância máxima de 12km para uma melhor comunicação) e seus respectivos atrasos são mantidos dentro de uma faixa pequena de variação, sendo aproximadamente 6 milissegundos para a embarcação 1 e 20 milissegundos para as demais.

Os gráficos da taxa de entrega apresentados na Figura 40 também sofreram mudanças de acordo com os deslocamentos das embarcações e decorrente disso sua entrada e saída do alcance umas das outras e da estação base. O início dos gráficos mostra a melhora do PDR conforme as embarcações se distanciam e a rota utilizada é modificada. Além disso, a faixa dos 4km para a comunicação entre embarcações e dos 14km para a comunicação das embarcações com a estação base, como visto na Seção 6.2.2, provocam uma degradação na taxa de entrega que são apresentadas pelas mudanças repentinas observadas nos gráficos na Figura 40 principalmente próximo aos instantes 2000 segundos e 2200 segundos para as embarcações 3 e 5.

Aproximadamente aos 2000 segundos as embarcações 3 e 5 passam a receber mais pacotes diretamente da embarcação 1 sem passar pelos nós intermediários anteriores que são as embarcações

2 e 4, respectivamente. Isso faz com que o PDR sofra um pequeno aumento repentino, o mesmo ocorre próximo aos 2200 segundos que é quando elas passam a receber mensagens diretamente da estação base.

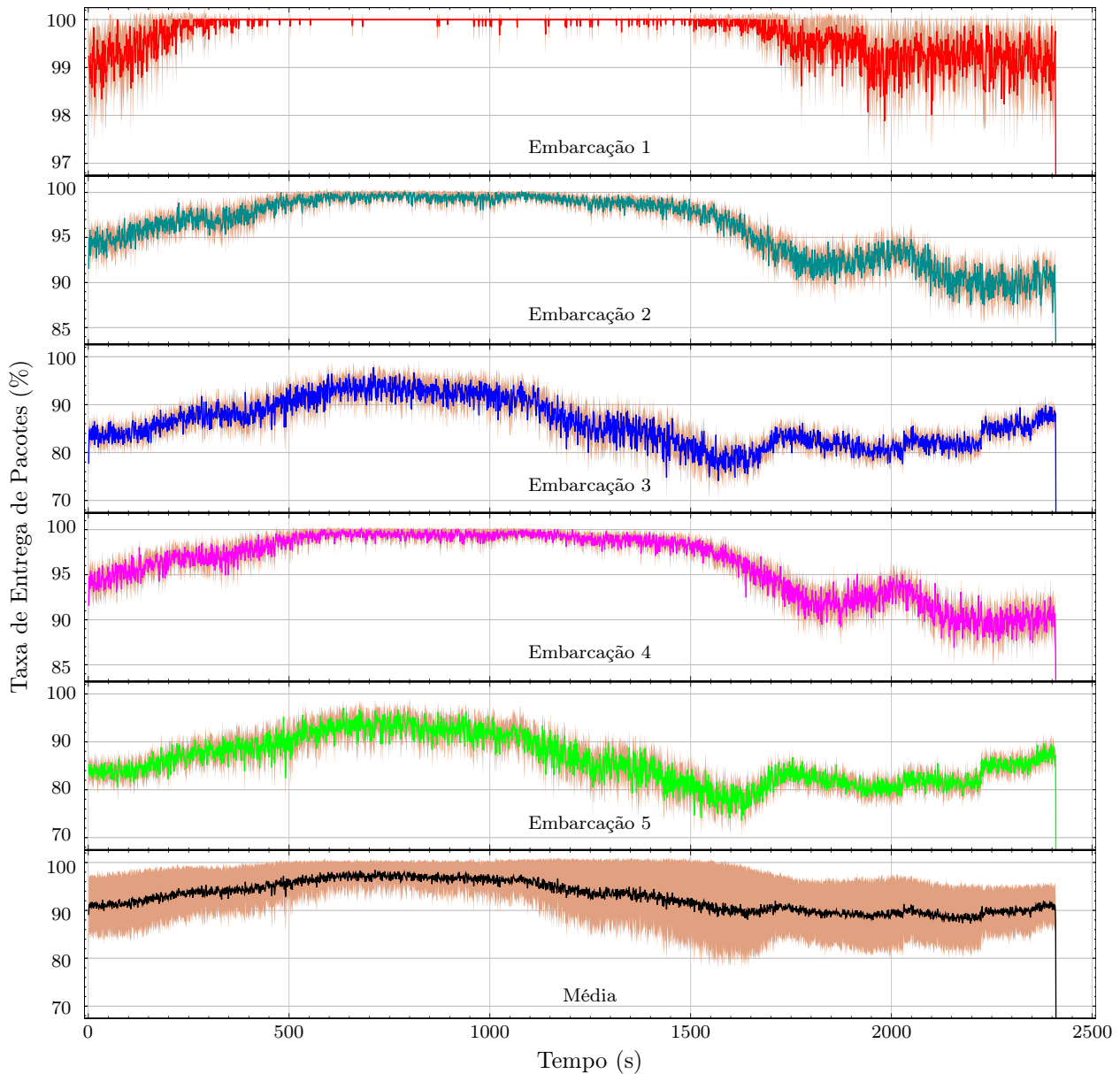


Figura 40 - Cenário 3: Taxa de Entrega de Pacotes (PDR).

Aproximadamente aos 2300 segundos todas as embarcações encontram-se dentro do alcance de comunicação da estação base, podendo assim receber mensagens diretamente dela. Apesar de estarem dentro do alcance da estação base as embarcações 2 e 3, assim como as 4 e 5, não conseguem o atingir o PDR acima dos 95% como a embarcação 1 por estarem muito próximas uma da outra e com distância próxima a faixa com maior interferência devido as reflexões como apresentado na

Seção 6.2.1 e na 6.2.2.

A Figura 41 apresenta a quantidade de nós vizinhos detectados por cada embarcação ao longo do tempo. O movimento das embarcações faz com que as mesmas entrem e saiam do alcance uma das outras e da estação base fato que é representado nos gráficos por meio das oscilações entre as transições. Para uma melhor visualização da quantidade de nós vizinhos ao longo do tempo de cada embarcação foi aplicada uma média móvel nos últimos 120 segundos sobre os gráficos na Figura 41 levando aos gráficos na Figura 42.

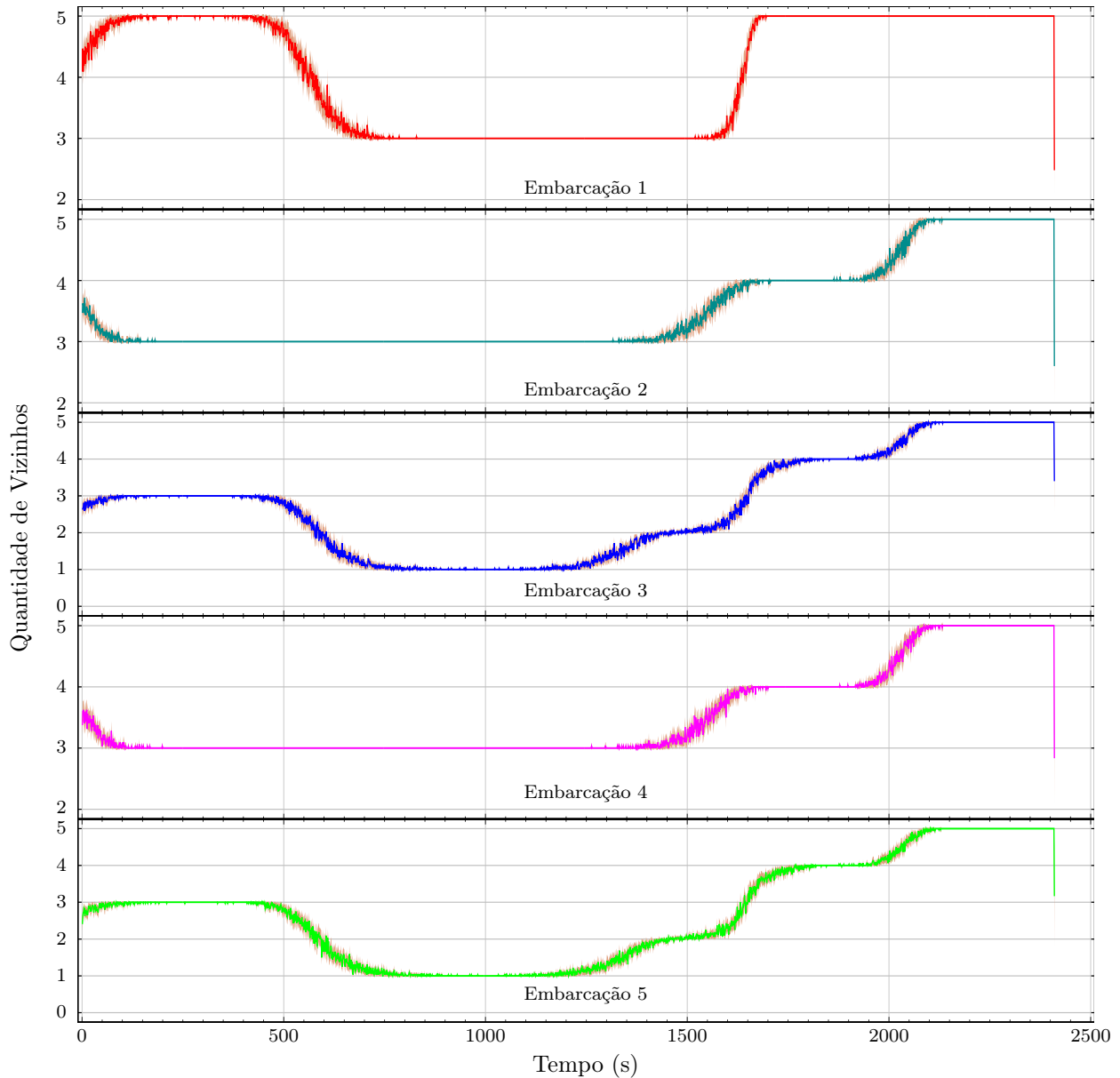


Figura 41 - Cenário 3: Vizinhaça sem Média Móvel.

Assim como ocorreu no cenário 2 apresentado na Seção 6.3.2, os gráficos apresentaram uma

mudança repentina no instante final causado pela rotina de finalização do protocolo o qual não é aguardada a confirmação da última mensagem enviada. Porém diferente do cenário 2, todas as rodadas de simulação apresentaram tempos de execução similares com poucos segundos de diferença e por isso os gráficos não apresentaram a mesma variação após o instante de finalização, aproximadamente aos 2410 segundos.

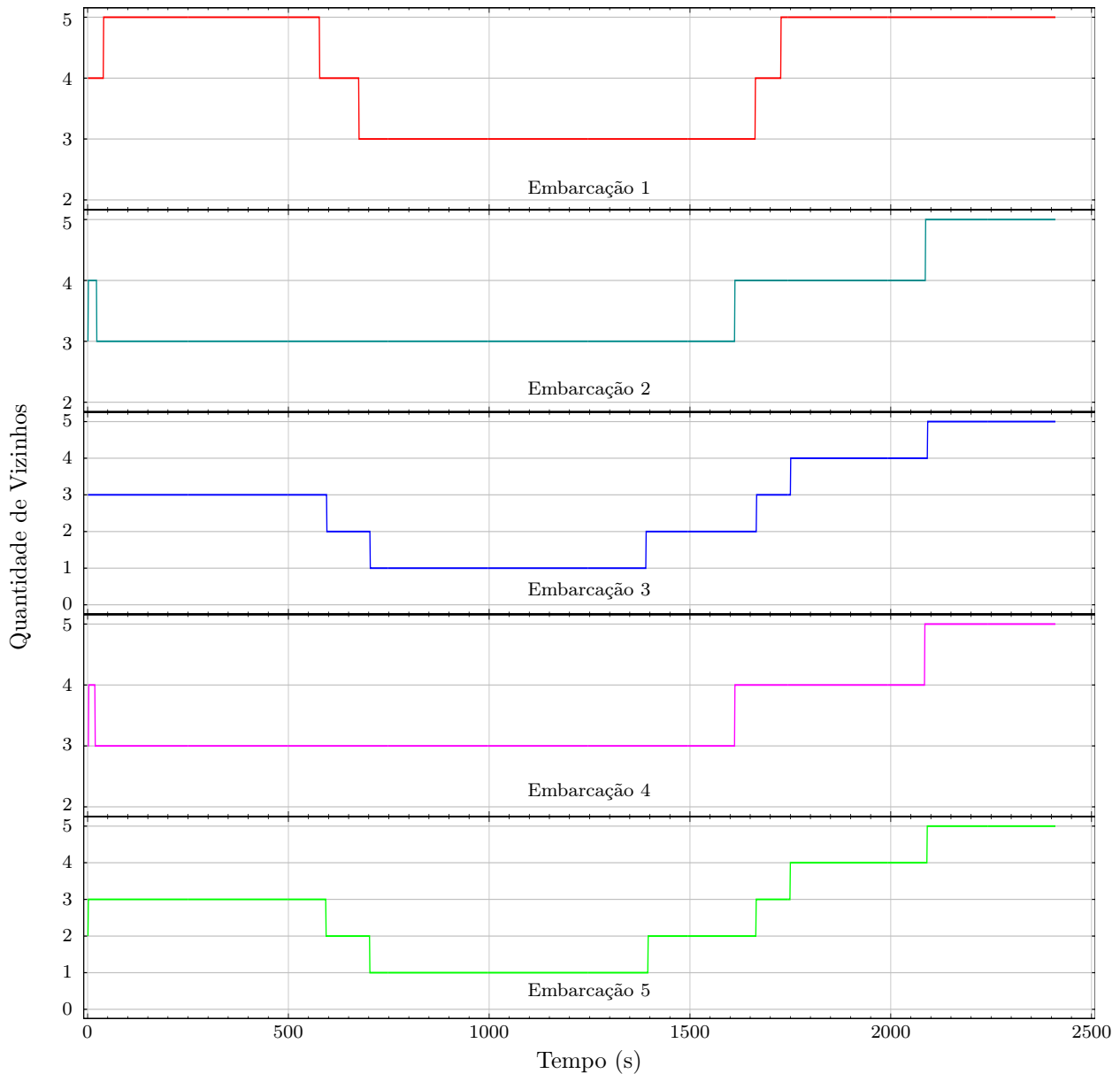


Figura 42 - Cenário 3: Vizinhança com Média Móvel dos Últimos 120 Segundos.

Neste cenário foi avaliado o comportamento do protocolo mediante as mudanças causadas pelo deslocamento das embarcações e a possibilidade da utilização das informações obtidas durante a comunicação como método de detecção de outros nós da rede (embarcações e estação base). Apesar

da trajetória ter sido previamente determinada de forma a evitar possíveis perdas da comunicação ou colisões, as informações trocadas entre os nós podem ser utilizadas em conjunto com a detecção da quantidade de nós vizinhos para se obter dinamicamente o mesmo resultado.

6.4 Discussão dos Resultados

Algumas observações a respeito de cada cenário são importantes destacar, como segue:

- no primeiro cenário, reproduziu-se o teste de comunicação da avaliação inicial, mas adicionando-se todas as camadas de protocolo propostas e a comunicação com o MATLAB. Este cenário também é similar com aqueles discutidos em [16, 75] que apresentam uma estação base e uma embarcação enviando mensagens a uma taxa específica;
- nos cenários 2 e 3 todas as embarcações alcançaram naturalmente o estado final do protocolo, ou seja, a comunicação é mantida mesmo mediante possíveis perdas de mensagens;
- o segundo cenário apresentou uma queda menor que 15% na taxa de entrega média das embarcações (Figura 35), uma queda era esperada devido ao tipo de formação, mas a comunicação se manteve estável. Mesmo para a embarcação mais distante a taxa de entrega ficou acima dos 75%, o que é aceitável se for levado em consideração que o a posição desta é o dobro do alcance de comunicação da estação base. O maior atraso observado foi menor que 500 milissegundos (Figura 34) e a média das cinco embarcações menor que 300 milissegundos;
- o último cenário apresentou resultados bem próximos aos obtidos nos testes da comunicação e nos dois cenários anteriores, mostrando que o protocolo proposto manteve a qualidade da comunicação com taxas de entrega média das cinco embarcações acima de 80% e atrasos de ida e volta médio das cinco embarcações abaixo dos 150 milissegundos. Mesmo com mais de 30% da distância de comunicação da estação base a taxa de entrega se manteve acima dos 80% e o atraso menor que 150 milissegundos, para as duas embarcações;
- a identificação da quantidade de nós vizinhos de cada embarcação pode ser utilizada em conjunto com as informações recebidas em cada mensagem para que cada embarcação possa tomar pequenas decisões relativas ao seu deslocamento.

CONCLUSÃO

Neste trabalho foi proposto um arcabouço para o suporte ao controle cooperativo de embarcações de superfície não tripuladas, considerando cenários onde existe uma estação base e uma ou mais embarcações, que executam missões simples, tais como se deslocar para uma determinada localização, realizar sensoriamento e voltar para a base, ou, cercar uma determinada área. Para realizar as missões, as embarcações devem se comunicar periodicamente com a estação base ou entre si, utilizando um protocolo de comunicação. O arcabouço proposto inclui:

- um modelo de frota com dois tipos de elementos: uma estação-base e uma ou mais embarcações, que podem agir hora como nó remetente de informação (coletadas de sensores locais, por exemplo), nó destinatário de informação de controle recebidos da estação base (ou de estações vizinhas) ou como nó intermediário, que trabalham para reencaminhar mensagens para a base (aumentando o raio de operação da frota);
- um protocolo de comunicação, de nível de aplicação, executado pela estação base e embarcações, desenvolvido para dar suporte à comunicação entre os elementos considerados;
- uma implementação modular do protocolo de comunicação desenvolvida para o simulador de redes NS-3, que facilita a alteração de parâmetros e a substituição de módulos das camadas de protocolo usadas;
- uma implementação de controle cinemático simples no MATLAB;
- um mecanismo de comunicação entre o NS-3 e o MATLAB de forma a utilizar as melhores especialidades de cada ambiente e oferecer um arcabouço o mais acurado possível em relação ao sistema de comunicação e às ações de controle e atuação.

A escolha da pilha de protocolo de comunicação e mecanismos específicos também são aspectos relevantes da proposta. Destaca-se que redes Wi-Fi em modo *ad hoc* foram combinadas com o protocolo de roteamento AODV e com mensagens *broadcast* IP limitadas em 1 salto como solução de comunicação do protocolo proposto. Também se destaca a restrição do tamanho máximo das mensagens de forma a transportar todos os dados de telemetria necessários e ainda assim, resultando um tamanho máximo de pacote IP compatível com o MTU definido. Assim, mantendo cada

mensagem da aplicação dentro de um único pacote e definindo o campo TTL do pacote de *broadcast* com o valor de 1 (**um**) antes de cada transmissão conseguiu-se:

- transmitir todas as informações de telemetria e controle para/da estação base em uma única mensagem;
- evitar a inundação da rede com mensagens de *broadcast*;
- estabelecer um mecanismo de comunicação de 1–N (um para muitos) entre embarcações vizinhas, dado que cada mensagem enviada é interceptada pelos nós com enlace direto;
- utilizar este mecanismo como informação de vizinhança das embarcações, vinculando a distância máxima de um enlace Wi-Fi com a distância entre as embarcações.

Observa-se que ações de controle complexas não eram o foco da proposta, restringindo-se, na implementação desenvolvida, a um controle cinemático simples, calculando velocidade e aceleração em três eixos independentemente. Mas, o arcabouço desenvolvido permite a estruturação de ações de controle e atuação mais sofisticadas.

Avaliamos nossa proposta inicialmente com um conjunto de testes para verificar a acurácia e identificar os limites relacionados à velocidade das embarcações, potência de sinal Wi-Fi recebido e a taxa de entrega (versus a taxa de perda) de pacotes. Também avaliamos o comportamento do protocolo de acordo com parâmetros como a potência, frequência e altura da antena.

Com base nos resultados obtidos na primeira avaliação, realizamos dois cenários concretos de simulação: (i) as embarcações formam uma linha, para avaliar a possibilidade de alcance geográfico do protocolo, considerando-se os tempos de recebimento de mensagens da estação base até o último nó da linha e (ii) as embarcações cercam uma determinada coordenada e retornam para próximo da estação base.

Os resultados da primeira avaliação e dos cenários concretos apresentaram resultados satisfatórios. Também foi possível concluir que a utilização da qualidade da comunicação como entrada para o controle de uma embarcação precisa de estudos mais aprofundados e experimentos para que se possa obter modelos mais precisos mediante a alta variabilidade do canal de comunicação estudado (comunicação sem fio por rádio frequência).

Trabalhos futuros

Diretamente dos objetivos iniciais do trabalho, trabalhos futuros incluiriam o desenvolvimento de comportamento autônomo para as embarcações, com ações de controle e atuação mais complexas. Seria possível introduzir nas embarcações o controle e a ação de desvios de barreiras, evitação de colisão entre embarcações vizinhas e mecanismos de controle para que uma embarcação mantivesse pelo menos um enlace e a comunicação com a estação base funcionando — utilizando sistemas massa-mola, por exemplo. Para isso, valores relacionados aos sinais do Wi-Fi e outros dados de telemetria já produzidos/obtidos pelas embarcações no arcabouço proposto poderiam ser empregados, aliado a dados de sensores como o GPS.

O próprio conceito de missão poderia ser, como comentado anteriormente, ampliado, permitindo a definição de missões mais complexas. Para isso seria necessário definir a sintaxe e semântica de uma missão e como os vários elementos incorporariam esta estrutura em suas rotinas de software.

Alguns parâmetros de simulação poderiam ser investigados mais amplamente. Para isso, a estrutura do simulador foi desenvolvida de forma modular, facilitando a substituição de módulos e protocolos. Por exemplo, outros modelos de erros em espelhos d'água poderiam ser avaliados, bem como até a substituição da tecnologia Wi-Fi ou o protocolo de roteamento AODV por outros igualmente úteis. Poderia ser usado canais diferentes com mais de uma antena em cada nó, aumentar a quantidade de estações base e até incluir estações base móveis que pudessem servir como nó intermediário com maior alcance.

Seria possível também, com maior trabalho de programação, avaliar outras abordagens de seleção de mecanismos como a substituição de mensagens de *broadcast* por canais privativos para cada par ou cada subgrupo de embarcações. Como não foi abordada nenhuma técnica para verificar possíveis intrusos ou para se recuperar em canais muito ruidosos, seria interessante aplicar algum algoritmo de criptografia, mudança randômica do canal e até da porta usada no protocolo UDP. Por fim, avaliar uma solução para aumentar a taxa de transferência, visto que a camada física limita essa taxa para mensagens por difusão. Talvez uma solução para isso seria atuar na camada de enlace ou uma solução híbrida. A inclusão do protocolo na camada de enlace removeria a necessidade da comunicação por difusão, assim por padrão qualquer embarcação já iria apresentar uma comunicação com o mínimo de informação necessária para seu controle local. Uma solução híbrida poderia prever a comunicação com a estação base usando mensagens diretas e para a detecção de vizinhança o envio periódico de mensagens por difusão.

Estudar a aplicação de filtros para melhorar a qualidade dos dados relacionados as métricas de comunicação (potência de sinal, taxa de entrega, atraso) e ao protocolo (quantidade de vizinhos número de saltos) para uso em ações de adaptação do protocolo e no controle das embarcações, aprofundando a integração entre o protocolo desenvolvido e ações de controle mais sofisticadas, aumentando a autonomia delas. Também é possível incluir um algoritmo de comboio [80–82] para o controle da movimentação da frota de embarcações permitindo uma movimentação em grupo durante as fases de ida ao destino e volta para a estação base.

Talvez a mais importante proposta para trabalhos futuros seria uma avaliação prática da proposta, com a implementação do arcabouço em dispositivos reais. A ideia seria portar o protocolo desenvolvido para sistemas em notebooks ou dispositivos de prototipagem como Arduino ou RaspberryPi, integrando-se os sensores necessários para a implementação de ações de controle e atuação mínimas. Neste sentido, o Laboratório de Controle do PEL conta com embarcações do tipo caiaque, motores, sensores e computadores móveis, preparados para experimentos práticos.

REFERÊNCIAS

- [1] YUAN, Z. L. e Youmin Zhang e Xiang Yu e C. Unmanned surface vehicles: An overview of developments and challenges. *Annual Reviews in Control*, v. 41, n. Supplement C, p. 71–93, 2016. ISSN 1367-5788. Acesso em: Janeiro de 2020.
- [2] JOHANSEN, A. Z. e David Palma e Kimmo Kansanen e Kay FjØrtoft e João Sousa e Karl H. Johansson e Yuming Jiang e Hefeng Dong e T. A. Survey on communication and networks for autonomous marine systems. *Journal of Intelligent & Robotic Systems*, Apr 2018. ISSN 1573-0409. Acesso em: Janeiro de 2020.
- [3] MONTANO, D. T. e A.R. Mosteo e L. Riazuelo e J.L. Villarroel e L. Enforcing network connectivity in robot team missions. *The International Journal of Robotics Research*, v. 29, n. 4, p. 460–480, 2010. Acesso em: Janeiro de 2020.
- [4] PEREIRA, A. R. *Controle Adaptativo de Formação para Agentes Autônomos Móveis Usando Potenciais Artificiais*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2011. Acesso em: Janeiro de 2020.
- [5] PENG, W. G. e L. Deyi e G. Wenyan e J. Study on formation control of multi-robot systems. In: *2013 Third International Conference on Intelligent System Design and Engineering Applications*. [S.l.: s.n.], 2013. p. 1335–1339. Acesso em: Janeiro de 2020.
- [6] INSTITUTE, D. A. R. P. A. e I. S. Rfc791 - internet protocol darpa internet program protocol specification. *The Internet Engineering Task Force*, 1981. Acesso em: Janeiro de 2020.
- [7] CALIFORNIA", D. A. R. P. A. I. P. T. O. e "Information Sciences Institute University of S. Rfc793 - transmission control protocol darpa internet program protocol specification. *The Internet Engineering Task Force*, 1981. Acesso em: Janeiro de 2020.
- [8] Rfc1122 - requirements for internet hosts – communication layers. *The Internet Engineering Task Force*, 1989. Acesso em: Janeiro de 2020.
- [9] BORMAN, V. J. e R. Braden e D. Rfc1323 - tcp extensions for high performance. *The Internet Engineering Task Force*, 1991. Acesso em: Janeiro de 2020.

- [10] ROMANOW, M. M. e J. Mahdavi e S. Floyd e A. Rfc2018 - tcp selective acknowledgment options. *The Internet Engineering Task Force*, 1996. Acesso em: Janeiro de 2020.
- [11] STEVENS, M. A. e V. Paxson e W. Rfc2581 - tcp congestion control. *The Internet Engineering Task Force*, 1999. Acesso em: Janeiro de 2020.
- [12] "ELECTRICAL, N. S. F. e Planète Group" e; LABORATORY", C. E. D. e "Electrical Engineering Department" e "U.S. Naval Research Laboratory" e U. A. R. *NS-3 - A Discrete-event Network Simulator for Internet Systems*. <https://www.nsnam.org>. Acesso em: Janeiro de 2020.
- [13] MATHWORKS. *MATLAB*. <https://www.mathworks.com/products/matlab.html>. Acesso em: Janeiro de 2020.
- [14] Ang, S. W. e P. Kong e J. Shankar e H. Wang e Y. Ge e C. A novel framework to simulate maritime wireless communication networks. In: *OCEANS 2007*. [S.l.: s.n.], 2007. p. 1–6. ISSN 0197-7385. Acesso em: Janeiro de 2020.
- [15] WEN, C.-W. A. e S. Signal strength sensitivity and its effects on routing in maritime wireless networks. In: *2008 33rd IEEE Conference on Local Computer Networks (LCN)*. [S.l.: s.n.], 2008. p. 192–199. ISSN 0742-1303. Acesso em: Janeiro de 2020.
- [16] OLIVEIRA, T. T. P. de. *Development of an ns-3 based Simulation Tool for TCP/IP Maritime Wireless Networks*. Dissertação (Mestrado) — Faculdade de Engenharia da Universidade do Porto, FEUP, Jul 2015. Acesso em: Janeiro de 2020.
- [17] IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/-MAN Specific Requirements - Part 15: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPAN). *IEEE Std 802.15.4-2003*, p. 1–680, Oct. 2003. Acesso em: Fevereiro de 2020.
- [18] IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments. *IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009)*, p. 1–51, Jul. 2010. Acesso em: Fevereiro de 2020.

- [19] ROSS, J. F. K. e K. W. *Redes de Computadores e a Internet: Uma abordagem Top-down*. Addison Wesley BRA, 2010. ISBN 9788588639973. Disponível em: <<https://books.google.com.br/books?id=raZtQwAACAAJ>>.
- [20] NOKIA, S. D. e R. Hinden e Cisco e. Internet protocol, version 6 (ipv6) specification. *The Internet Engineering Task Force*, Dec 1998. Acesso em: Janeiro de 2020.
- [21] HINDEN, S. D. e R. Internet protocol, version 6 (ipv6) specification. *The Internet Engineering Task Force*, Jul 2017. ISSN 2070-1721. Acesso em: Janeiro de 2020.
- [22] IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems. *IEEE Std 802.16-2004 (Revision of IEEE Std 802.16-2001)*, p. 1–857, 2004. Acesso em: Fevereiro de 2020.
- [23] ZENNARO, N. J. e Vujo Drndarevic e Ermanno Pietrosemoli e Iain Darby e M. Experimental study of lora transmission over seawater. *Sensors*, v. 18, n. 9, 2018. ISSN 1424-8220. Acesso em: Janeiro de 2020.
- [24] COMMITTEE, L. A. T. *LoRaWAN 1.1 Specification*. Oct 2017. https://loralliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf. Acesso em: Fevereiro de 2020.
- [25] TRANSLATIONS, A. T. e D. J. Wetherall e O. *Redes de Computadores*. Prentice Hall Brasil, 2011. ISBN 9788576059240. Disponível em: <<https://books.google.com.br/books?id=fnfwkQEACAAJ>>.
- [26] STANDARD for Information Technology - Telecommunications and information exchange between systems - Local and Metropolitan Area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. *ANSI/IEEE Std 802.11, 1999 Edition (R2003)*, p. 1–512, Nov 1998. Acesso em: Janeiro de 2020.
- [27] IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/-MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: High Speed Physical Layer in the 5 GHz band. *IEEE Std 802.11a-1999*, p. 1–102, Dec 1999. Acesso em: Janeiro de 2020.

- [28] IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local and Metropolitan networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher Speed Physical Layer (PHY) Extension in the 2.4 GHz band. *IEEE Std 802.11b-1999*, p. 1–96, Jan 2000. Acesso em: Janeiro de 2020.
- [29] IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part Ii: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11g-2003 (Amendment to IEEE Std 802.11, 1999 Edn. (Reaff 2003) as amended by IEEE Stds 802.11a-1999, 802.11b-1999, 802.11b-1999/Cor 1-2001, and 802.11d-2001)*, p. i–67, 2003. Acesso em: Janeiro de 2020.
- [30] IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput. *IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009)*, p. 1–565, Oct 2009. Acesso em: Janeiro de 2020.
- [31] Zheng, Z. Z. e Z. Lu e Q. Chen e X. Yan e L. Cosmo: Co-simulation with matlab and omnet++ for indoor wireless networks. In: *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*. [S.l.: s.n.], 2010. p. 1–6. ISSN 1930-529X. Acesso em: Janeiro de 2020.
- [32] KACHAN, D. *Integration of ns-3 with MATLAB/Simulink*. Dissertação (Mestrado) — Lulea University of Technology, Fev 2010. Acesso em: Janeiro de 2020.
- [33] ISO. *C++ Programming Language*. [Http://www.cplusplus.com](http://www.cplusplus.com). Acesso em: Janeiro de 2020.
- [34] ISO. *ISO/IEC 14882:2011 Information technology - Programming languages - C++*. Third. [S.l.]: pub-ISO, 2011. [Https://www.iso.org/standard/50372.html](https://www.iso.org/standard/50372.html). Acesso em: Janeiro de 2020.
- [35] ENABLING Open Innovation and Collaboration - The Eclipse Foundation. [Https://www.eclipse.org](https://www.eclipse.org). Acesso em: Janeiro de 2020.
- [36] OMNET++. [Https://omnetpp.org](https://omnetpp.org). Acesso em: Janeiro de 2020.

- [37] INITIATIVE", O. S. *GNU Library General Public License*. <https://opensource.org/licenses/lgpl-license>. Acesso em: Janeiro de 2020.
- [38] FOUNDATION", P. S. *Python Programming Language*. <https://www.python.org>. Acesso em: Janeiro de 2020.
- [39] OPENWNS. <http://www.openwns.org>. Acesso em: Janeiro de 2020.
- [40] WETHERALL, D. *OTcl*. <http://otcl-tclcl.sourceforge.net/otcl/>. Acesso em: Janeiro de 2020.
- [41] INITIATIVE", O. S. *GNU General Public License*. <https://opensource.org/licenses/gpl-license>. Acesso em: Janeiro de 2020.
- [42] DEFENSE Advanced Research Projects Agency. <https://www.darpa.mil>. Acesso em: Janeiro de 2020.
- [43] THE Network Simulator - NS-2. <https://isi.edu/nsnam/ns>. Acesso em: Janeiro de 2020.
- [44] CABRERA, J. L. V. e J. Paez e C. Quintero e E. Yime e J. Design and control of an unmanned surface vehicle for environmental monitoring applications. In: *2016 IEEE Colombian Conference on Robotics and Automation (CCRA)*. [S.l.: s.n.], 2016. p. 1–5. Acesso em: Janeiro de 2020.
- [45] MONTEIRO Ítalo Oliveira Ferreira e Arthur Ayres Neto e C. S. O uso de embarcações não tripuladas em levantamentos batimétricos. *Revista Brasileira de Cartografia*, v. 68, n. 10, p. 1885–1903, Dez 2016. ISSN 1808-0936. Acesso em: Janeiro de 2020.
- [46] CUNHA, J. P. V. S. da. *Embarcações Não Tripuladas para Monitoração Ambiental e Defesa*. R. São Francisco Xavier, 524 - Maracanã - Rio de Janeiro - RJ - Cep 20550-900: [s.n.], 2011. Projeto de Pesquisa — Faperj.
- [47] CHOI, D. Y. Measurement of radio propagation path loss over the sea for wireless multimedia. In: _____. *NETWORKING 2006. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems: 5th International IFIP-TC6 Networking Conference, Coimbra, Portugal, May 15-19, 2006. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 525–532. ISBN 978-3-540-34193-2. Acesso em: Janeiro de 2020.

- [48] RAO, S. B. P. A. e K. Sasidhar e S. A survey based analysis of propagation models over the sea. In: *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. [S.l.: s.n.], 2015. p. 69–75. Acesso em: Janeiro de 2020.
- [49] SUTTON, W. N. e T. Xu e R. An intelligent integrated navigation and control solution for an unmanned surface craft. In: *Signals and Systems Conference (ISSC 2009), IET Irish*. Dublin: [s.n.], 2009. p. 1–6. Acesso em: Janeiro de 2020.
- [50] COLITO, J. *Autonomous Mission Planning and Execution for Unmanned Surface Vehicles in Compliance with the Marine Rules of the Road*. Dissertação (Mestrado) — University of Washington, 2007. Acesso em: Janeiro de 2020.
- [51] FOSSEN, T. I. Introduction. In: _____. *Handbook of Marine Craft Hydrodynamics and Motion Control*. [S.l.]: John Wiley & Sons, Ltd, 2011. cap. 1, p. 1–14. ISBN 9781119994138.
- [52] FALTINSEN, O. M. *Hydrodynamics of High-Speed Marine Vehicles*. [S.l.]: Cambridge University Press, 2006. <https://www.doi.org/10.1017/CBO9780511546068>. Acesso em: Fevereiro de 2020.
- [53] CUNHA, J. P. V. S. da. *Projeto e Estudo de Simulação de um Sistema de Controle a Estrutura Variável de um Veículo Submarino de Operação Remota*. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, Fev 1992. Acesso em: Janeiro de 2020.
- [54] OGATA, K. Análise de resposta transitória e de regime estacionário. In: _____. *Engenharia de controle moderno*. São Paulo - SP: Pearson Prentice Hall, 2011. p. 145–245. ISBN 978-85-7605-810-6.
- [55] ÄSTRÖM, K. J.; WITTENMARK, B. *Computer-controlled Systems (3rd Ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1997. ISBN 0-13-314899-8.
- [56] BEZERRA, E. de M. S. *Sistema de Comunicação para Controle Descentralizado de Embarcações Autônomas*. R. São Francisco Xavier, 524 - Maracanã - Rio de Janeiro - RJ - Cep 20550-900: [s.n.], 2013. "Proposta de Projeto".
- [57] AMORY, V. A. M. J. e Roger Granada e Rena G. Maidana e Darlan A. Jurak e Guilherme Heck e Alvaro P. F. Negreiros e Davi H. dos Santos e Luiz M. G. Gonçalves e A. M. A survey on

- unmanned surface vehicles for disaster robotics: Main challenges and directions. *Sensors*, v. 19, n. 3, 2019. ISSN 1424-8220. Acesso em: Janeiro de 2020.
- [58] SHUWU, W. C. e M. Feng e W. Qing e W. A situation awareness approach for usv based on artificial potential fields. In: *2017 4th International Conference on Transportation Information and Safety (ICTIS)*. [S.l.: s.n.], 2017. p. 232–235. Acesso em: Janeiro de 2020.
- [59] KAYNAK, S. Y. e H. Yang e O. Coordination task triggered formation control algorithm for multiple marine vessels. *IEEE Transactions on Industrial Electronics*, v. 64, n. 6, p. 4984–4993, Jun 2017. ISSN 0278-0046. Acesso em: Janeiro de 2020.
- [60] Arcak, M. Passivity as a design tool for group coordination. *IEEE Transactions on Automatic Control*, v. 52, n. 8, p. 1380–1390, Aug 2007. ISSN 2334-3303. Acesso em: Fevereiro de 2020.
- [61] HASEGAWA, J. P. e K. Study on turning manoeuvre of catamaran surface vessel with a combined experimental and simulation method. *10th IFAC Conference on Control Applications in Marine System CAMS*, v. 49, n. 23, p. 446–451, 2016. ISSN 2405-8963. Acesso em: Janeiro de 2020.
- [62] HASEGAWA, J. P. e K. Autonomous navigation of catamaran surface vessel. In: *2017 IEEE Underwater Technology (UT)*. [S.l.: s.n.], 2017. p. 1–6. Acesso em: Janeiro de 2020.
- [63] SHAH, B. *Planning for Autonomous Operation of Unmanned Surface Vehicles*. Tese (Doutorado) — University of Maryland, 2016. Acesso em: Janeiro de 2020.
- [64] LOBERG, M. B. e J.-E. A virtual target-based underway docking procedure for unmanned surface vehicles. *18th IFAC World Congress - IFAC Proceedings Volumes*, v. 44, n. 1, p. 13630–13635, 2011. ISSN 1474-6670. Acesso em: Janeiro de 2020.
- [65] Elfaham, H. et al. Full integration of matlab/simulink with control application development using opc unified architecture. In: *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*. [S.l.: s.n.], 2016. p. 371–376. ISSN 2378-363X. Acesso em: Janeiro de 2020.
- [66] GROSS, V. N. e T. Simulation of large ad hoc networks. In: *Proceedings of the 6th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*. New York, NY, USA: ACM, 2003. (MSWIM '03), p. 50–57. ISBN 1-58113-766-4. Acesso em: Janeiro de 2020.

- [67] DAS, C. P. e E. Belding-Royer e S. Rfc3561 - ad hoc on-demand distance vector (aodv) routing. *The Internet Engineering Task Force*, 2003. Acesso em: Janeiro de 2020.
- [68] POSTEL, J. Rfc768 - user datagram protocol. *The Internet Engineering Task Force*, 1980. Acesso em: Janeiro de 2020.
- [69] CHOUDHURY, A. B. e S. Vuppala e P. A simulation analysis of flooding attack in manet using ns-3. In: *2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE)*. [S.l.: s.n.], 2011. p. 1–5. Acesso em: Janeiro de 2020.
- [70] CHEON, D. W. K. e J. S. Park e Y. Z. Cho e B. Y. Performance evaluation of aodv and dymo routing protocols in manet. In: *2010 7th IEEE Consumer Communications and Networking Conference*. [S.l.: s.n.], 2010. p. 1–2. ISSN 2331-9852. Acesso em: Janeiro de 2020.
- [71] USC/ISI, J. T. e. Rfc6864 - updated specification of the ipv4 id field. *The Internet Engineering Task Force*, Feb 2013. ISSN 2070-1721. Acesso em: Janeiro de 2020.
- [72] PARTRIDGE, T. M. e D. Borman e C. Rfc1071 - computing the internet checksum. *The Internet Engineering Task Force*, Sep 1988. Acesso em: Janeiro de 2020.
- [73] KULLBERG, R. B. e A. Rfc1141 - incremental updating of the internet checksum. *The Internet Engineering Task Force*, Jan 1990. Acesso em: Janeiro de 2020.
- [74] RIJSINGHANI, A. Rfc1624 - computation of the internet checksum via incremental update. *The Internet Engineering Task Force*, May 1994. Acesso em: Janeiro de 2020.
- [75] LOPES, M. J. *Comunicações marítimas wi-fi usando a banda 5.8 ghz*. Dissertação (Mestrado) — Instituto Superior de Engenharia do Porto, ISEP, Out 2013. Acesso em: Janeiro de 2020.
- [76] BOONKIRD, P. K. e T.; ASWAKUL, C. Performance measurement of maritime ad hoc network testbed in realistic offshore scenario. In: *2016 International Conference on Electronics, Information, and Communications (ICEIC)*. [S.l.: s.n.], 2016. p. 1–4. Acesso em: Janeiro de 2020.
- [77] SHANKAR, P. M. *Fading and Shadowing in Wireless Systems*. 2nd. ed. [S.l.]: Springer Publishing Company, Incorporated, 2017. ISBN 3319531972.

- [78] MIKROTIK. *RouterBOARD R52n-M*. https://i.mt.lv/cdn/rb_files/R52n-M.pdf. Acesso em: Janeiro de 2020.
- [79] SYSTEMS, E. *ESP32*. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_. Acesso em: Janeiro de 2020.
- [80] REYNOLDS, C. W. Flocks, herds and schools: A distributed behavioral model. In: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA: Association for Computing Machinery, 1987. (SIGGRAPH '87), p. 25–34. ISBN 0897912276. Acesso em: Janeiro de 2020. Disponível em: <<https://doi.org/10.1145/37401.37406>>.
- [81] REYNOLDS, C. W. Not bumping into things. In: . [S.l.: s.n.], 1988. p. G1. Acesso em: Janeiro de 2020.
- [82] REYNOLDS, C. W. Steering behaviors for autonomous characters. In: CITESEER. *Game developers conference*. [S.l.], 1999. v. 1999, p. 763–782. Acesso em: Janeiro de 2020.
- [83] TEXTUALITY, E. T. B. e. Rfc8259 - the javascript object notation (json) data interchange format. *The Internet Engineering Task Force*, Dec 2017. ISSN 2070-1721. Acesso: Janeiro de 2020.

APÊNDICE A – Arquivo de Configuração

Dada a quantidade de parâmetros a serem configurados para a execução de uma simulação foi criado um arquivo de configuração em formato JSON [83] juntamente com os recursos necessários para que o arquivo pudesse ser lido e transformado no conjunto de parâmetros dentro da simulação durante sua execução. Como o arquivo de configuração ficou grande para apresentar em uma página, foi decidido dividi-lo em partes para facilitar a explicação. A primeira parte, Algoritmo 1, apresenta os campos principais nos quais o arquivo foi subdividido, são eles: *channel* onde as definidas as configurações do canal de comunicação; *simulation* com as configurações gerais da simulação; *tests* com as configurações para a execução dos testes iniciais; *bases* com as configurações das estações base; *vessels* com as configurações das embarcações.

Algoritmo 1 – Configuração Geral

```
{"channel":{}, "simulation":{}, "tests":{}, "bases":{}, "vessels":{}}
```

O Algoritmo 2 mostra as configurações do canal de comunicação: o parâmetro *delay* define o modelo de atraso do canal; *loss* define qual modelo de propagação utilizado, Seção 5.1.2, com os parâmetros do canal Wi-Fi (*channel*), as perdas totais de cabos e conectores (*sloss*), se utiliza ou não a componente de desvanecimento (*fadin*), o parâmetro de escala (*scale*) e da forma (*shape*).

Algoritmo 2 – Configuração do Canal

```
1 {"channel":{
2   "delay": "ConstantSpeedPropagationDelayModel",
3   "loss":{
4     "MaritimePropagationLossModel":{
5       "channel":1,
6       "sloss":5.0,
7       "fadin":true,
8       "scale":1.0,
9       "shape":2.0
10    }
11  }
12 }}
```

O Algoritmo 3 mostra as configurações da simulação: os primeiros três parâmetros (linhas 2-4) correspondem a semente de aleatoriedade, a rodada inicial e a quantidade de rodadas a serem executadas, respectivamente; o parâmetro *matlab* representa o caminho para os arquivos do

MATLAB (Seção 5.2); *results* indica o caminho onde serão salvos os resultados gerados; *save* é o intervalo em minutos entre cada rodada de salvamento dos resultados obtidos; os próximos sete campos (linhas 8-14) indicam respectivamente, o tipo de missão, o tempo que deverá durar a missão quando todos as embarcações estiverem chegado ao destino, o tempo total de simulação (quando definido como zero a simulação executa até todas as embarcações e estações base chegarem no estado final), o intervalo em milissegundos entre as chamadas ao MATLAB, um valor lógico indicando se deve ou não informar cada passo executado, outro valor lógico indicando se deve ser utilizado ou não a comunicação por *broadcast* e por fim a taxa de transferência quando ativada a comunicação por *broadcast*; o campo seguinte corresponde as configurações da missão (linha 16) é um vetor com as coordenadas (x,y,z) e as distâncias mínima e máxima, respectivamente.

Algoritmo 3 – Configuração da Simulação

```

1 {"simulation":{
2   "s":2020,
3   "r":0,
4   "rounds":30,
5   "matlab":"/vessels-communication/vessels-communication/matlab",
6   "results":"/resultados",
7   "save":30,
8   "m":1,
9   "tm":300,
10  "ts":0,
11  "ti":10,
12  "v":false,
13  "b":true,
14  "rb":6,
15  "ps":{
16    "dia":[18000.0,0.0,3.0,1000.0,4000.0],
17  }
18 }}

```

Para a execução dos testes iniciais são necessários os campos apresentados no Algoritmo 4, como segue: o primeiro campo “namef” representa o nome do diretório a ser criado dentro do diretório de resultados informados no campo “results” no Algoritmo 3 para salvar os resultados dos testes; o próximo campo “execs” indica se os testes devem ou não ser executados; o campo “dmaxm” informa a distância máxima em metros para a execução dos testes; em seguida o campo “runsp” indica o número de rodas a serem executadas no teste de propagação; os próximos quatro campos são vetores indicando as frequências em Hz “fqsHz”, as alturas das antenas “altsm”, as potências de transmissão “power” e os ganhos das antenas “gains”, respectivamente; os campos “sensr”, “noise”, “rateb”, “runsv” e “pks2s” indicam a sensibilidade de recepção, o parâmetro de ruído do canal, a taxa de transferência para comunicação por *broadcast*, o número de rodas a serem executadas no

teste de comunicação e a quantidade de pacotes a serem enviados a cada metro; o campo “withb” informa se deve ou não utilizar a comunicação por difusão, sendo possível utilizar um vetor com os dois valores possíveis para se obter os dois resultados; o campo seguinte “from2” indica quais os tipos de nós que farão parte da comunicação e quem irá enviar e receber, no caso há três opções de comunicação, da estação base para uma embarcação (B2V), de uma embarcação para a estação base (V2B) ou de uma embarcação para outra (V2V), é possível utilizar um vetor para obter os resultados para os três casos; o campo “sizes” indicam o tamanho de cada mensagem a ser enviada no teste de comunicação (132, 456 ou 2200 *bytes*), um vetor permite obter os resultados para os diferentes tamanhos; o campo “velos” indica a velocidade que o nó móvel deverá apresentar no teste da comunicação, um vetor pode guardar as diferentes velocidades a serem testadas; por fim, o campo “altvb” indica as alturas das antenas da embarcação e estação base, respectivamente.

Algoritmo 4 – Configuração dos Testes Iniciais

```

1  {"test":{
2    "namef":"TestCommunication",
3    "execs":true,
4    "dmaxm":30000,
5    "runsp":30,
6    "fqsHz":[2400000000,5800000000],
7    "altsm":[3,8],
8    "power":[20,20],
9    "gains":[10,16],
10   "sensr":-95,
11   "noise":10.0,
12   "rateb":6,
13   "runsv":30,
14   "pks2s":10,
15   "withb":[true,false],
16   "from2":["B2V","V2B","V2V"],
17   "sizes":[132,456,2200],
18   "velos":[1,5,10],
19   "altvb":[3,20]
20  }}
```

O Algoritmo 5 apresenta os parâmetros de configuração das estações base, com os seguintes campos: “qtd” indica a quantidade de estações bases a serem criadas, os próximos três campos indicam as coordenadas (x,y,z) das estações bases, sendo um valor em cada campo para cada estação base a ser criada; o campo “phy” guardas as configurações da interface de comunicação, tais como: potência de transmissão, ganho de transmissão, ganho de recepção, sensibilidade de recepção e parâmetro de ruído do canal, respectivamente.

Algoritmo 5 – Configuração das Estações Base

```

1 {"bases":{
2   "qtd":1,
3   "x":[ 0.0],
4   "y":[ 0.0],
5   "z":[20.0],
6   "phy":{
7     "txp":20.0,
8     "txg":16.0,
9     "rxg":16.0,
10    "rxg":16.0,
11    "rxg":16.0,
12  }
13 }}

```

Os parâmetros de configuração das embarcações apresentam os mesmo campos das estações base, como pode ser visto no Algoritmo 6, com valores diferentes. Alguns campos são específicos para as embarcações, tais como: “vm” que indica as velocidades máximas nas coordenadas (x,y,z) e “dm” que indica as dimensões das embarcações nas coordenadas (x,y,z).

Algoritmo 6 – Configuração das Embarcações

```

1 {"vessels":{
2   "qtd":5,
3   "x":[12000.0,12000.0,12000.0,12000.0,12000.0],
4   "y":[ 0.0, 2000.0,-2000.0, 4000.0,-4000.0],
5   "z":[ 3.0, 3.0, 3.0, 3.0, 3.0],
6   "vm":[5.0,5.0,0.0],
7   "dm":[5.0,5.0,3.0],
8   "phy":{
9     "txp":20.0,
10    "txg":10.0,
11    "rxg":10.0,
12    "rxg":10.0,
13    "rxg":10.0,
14  }
15 }}

```

APÊNDICE B – *Script* para Inclusão do Módulo no NS-3

O NS-3 permite a inclusão de módulos extras e para isso utiliza uma biblioteca em Python para verificar as configurações dos módulos e executar as rotinas para construção de suas bibliotecas, para cada módulo do NS-3 é criado um arquivo de biblioteca que podem ser chamados por outros programas [12]. Para que o módulo desenvolvido fosse incluído no NS-3 foi necessário seguir as recomendações e documentação, o que levou a criação *script* apresentado no Algoritmo 7. O *script* desenvolvido em Python primeiro importa as bibliotecas necessárias, linhas 1-5, depois apresenta três rotinas básicas: a rotina da linha 6 serve para adicionar opções ao NS-3; a rotina da linha 8 serve para configurar e verificar os recursos necessários para a inclusão do módulo ao NS-3; a rotina na linha 10 carrega os recursos para a construção do módulo.

Algoritmo 7 – *Script* Padrão do NS-3

```

1 import sys
2 import os
3 import subprocess
4 from waflib import Options
5 from waflib.Errors import WafError
6
7 def options(opt):
8     ...
9
10 def configure(conf):
11     ...

```

A rotina “options” no Algoritmo 8 permite adicionar opções ao NS-3 utilizando a chamada “opt.add_option”. As linhas 2-3 adicionam uma opção para ativar o módulo e as linhas 4-5 para desativar, permitindo ativar/desativar o módulo durante a configuração do NS-3. As linhas 6-7 adicionam uma opção para desativar a inclusão do MATLAB.

Algoritmo 8 – Adicionar Módulo nas Lista de Opções do NS-3

```

1 def options(opt):
2     opt.add_option('--enable-vessel',
3         help=('Enable Vessel-Communication Module'),dest='enable_vessel',action='store_true',default=True)
4
5     opt.add_option('--disable-vessel',
6         help=('Disable Vessel-Communication Module'),dest='disable_vessel',action='store_true',default=False)
7
8     opt.add_option('--disable-matlab',
9         help=('Disable MATLAB support'),dest='disable_matlab',action='store_true',default=False)

```

Para verificar e configurar o módulo a rotina “configure” é executada, Algoritmo 9. Primeiro é feita a verificação se o módulo foi ativado, linha 2-5, caso ele esteja desativado a rotina retorna. Caso o módulo esteja ativado é feita a verificação se o MATLAB está desativado, linha 6, caso positivo é reportado a necessidade do MATLAB para ativação do módulo então a rotina retorna sem incluir o módulo, linhas 31-35. Caso o MATLAB esteja ativado é verificada a existência do mesmo efetuando uma chamada para sua execução, linhas 7-8. A linha 10 verifica o resultado da chamada ao MATLAB, em caso negativo é reportada a falha e o módulo não é ativado, linhas 24-29. Caso bem sucedida, as linhas 11-23 adicionam os parâmetros para compilação e ligação, então reporta a existência do MATLAB. Por fim, ao final caso tudo o MATLAB tenha sido incluído com sucesso as linhas 36 inclui o módulo e a 37 reporta sua ativação.

Algoritmo 9 – Configurar o NS-3 para Suportar Novo Módulo

```

1  def configure(conf):
2      if Options.options.disable_vessel:
3          conf.report_optional_feature("vessels-communication", "Vessel-Communication", False, 'disabled by user request')
4          conf.env['ENABLE_VESSELS'] = False
5          return

6      if not Options.options.disable_matlab:
7          matlab = conf.check_cfg(path='matlab', args='-nodesktop -nodisplay -nosplash -r quit',
8                                  package='', msg='Checking for MATLAB', uselib_store='MATLAB', mandatory=False)

9          conf.env.append_value('CXXFLAGS', ['-utf8', '-pthread'])
10         if matlab:
11             conf.env.append_value('DEFINES_MATLAB', 'NS3_MATLAB')
12             conf.env['ENABLE_MATLAB'] = True
13             conf.env.append_value('CXXFLAGS', '-DMATLAB')
14             conf.env.append_value('CXXFLAGS', '-I/usr/local/MATLAB/R2018a/extern/include')
15             conf.env.append_value('CXXDEFINES', 'DMATLAB')
16             temp = ['-lpthread',
17                     '-L/usr/local/MATLAB/R2018a/sys/os/glnxa64',
18                     '-Wl,-rpath,/usr/local/MATLAB/R2018a/sys/os/glnxa64',
19                     '-lMatlabDataArray', '-lMatlabEngine', '-lmat']
20             for lib in temp:
21                 conf.env.append_value('LINKFLAGS', lib)
22                 conf.env.append_value('SHLIB_MARKER', lib)
23             conf.report_optional_feature("matlab", "MATLAB Support", True, '')
24         else:
25             conf.report_optional_feature("matlab", "MATLAB Support", False, 'matlab not found')
26             conf.report_optional_feature("vessels-communication", "Vessel-Communication",
27                                         False, 'disabled because matlab not found')
28             conf.env['ENABLE_VESSELS'] = False
29             return
30     else:
31         conf.report_optional_feature("matlab", "MATLAB Support", False, 'option --disable-matlab selected')
32         conf.report_optional_feature("vessels-communication", "Vessel-Communication",
33                                     False, 'disabled because matlab disabled')
34         conf.env['ENABLE_VESSELS'] = False
35         return

36     conf.env['ENABLE_VESSELS'] = True
37     conf.report_optional_feature("vessels-communication", "Vessel-Communication", True, '')

```

Com o módulo ativado e configurado a rotina de construção é executada, Algoritmo 10. Na rotina “build” é feita a verificação se o módulo está ativado, linha 3, e caso negativo a rotina retorna sem qualquer mudança, linha 4. O comando “bld.create_ns3_module”, linhas 6-7, cria o módulo e adiciona os módulos utilizados. Os arquivos a serem compilados são passados na linha 8 e os arquivos de cabeçalho na linha 11. As linhas 12-17 adicionam os parâmetros necessários para a compilação e as 18-26 os parâmetros para ligação. Por fim a linha 27 ativa o MATLAB no módulo.

Algoritmo 10 – Construção do Módulo do NS-3

```

1 def build(bld):
2     env = bld.env
3     if not env['ENABLE_VESSELS']:
4         return
5     env['ENABLE_STATIC_NS3'] = False
6     module = bld.create_ns3_module('vessels-communication',
7         ['core', 'network', 'internet', 'wifi', 'config-store', 'stats', 'mobility', 'netanim', 'aodv'])
8
9     module.source = ['model/*.cc', 'helper/*.cc']
10    headers = bld(features='ns3header')
11    headers.module = 'vessels-communication'
12    headers.source = ['model/*.h', 'helper/*.h']
13
14    module.env.append_value('CXXFLAGS', '-DMATLAB')
15    env.append_value('CXXFLAGS', '-DMATLAB')
16    module.env.append_value('CXXFLAGS', '-I/usr/local/MATLAB/R2018a/extern/include')
17    env.append_value('CXXFLAGS', '-I/usr/local/MATLAB/R2018a/extern/include')
18
19    module.env.append_value('CXXDEFINES', 'DMATLAB')
20    env.append_value('CXXDEFINES', 'DMATLAB')
21
22    temp = ['-lpthread',
23        '-L/usr/local/MATLAB/R2018a/sys/os/glnxa64',
24        '-Wl,-rpath,/usr/local/MATLAB/R2018a/sys/os/glnxa64',
25        '-lMatlabDataArray', '-lMatlabEngine', '-lmat']
26
27    for lib in temp:
28        module.env.append_value('LINKFLAGS', lib)
29        env.append_value('LINKFLAGS', lib)
30        module.env.append_value('SHLIB_MARKER', lib)
31        env.append_value('SHLIB_MARKER', lib)
32
33    module.use.append('MATLAB')

```

APÊNDICE C – Modificações Extras

Mesmo com o *script* de construção apresentado no Apêndice B foram necessárias modificações adicionais para que todos os recursos necessários fossem incluídos e o módulo utilizado com todos os seus recursos. A primeira modificação permite a execução das chamadas ao MATLAB, para isso foi necessária a criação dos *links* para as bibliotecas como mostrado no Algoritmo 11. Os três primeiros comandos criam *links* das bibliotecas do MATLAB dentro de diretório principal do MATLAB, fazendo com que seja necessário a indicação apenas desse diretório e não o caminho de cada arquivo separadamente. A última linha modifica a biblioteca padrão utilizada pelo MATLAB, fazendo com que o mesmo utilize a biblioteca do sistema e não a que vem com sua instalação.

Algoritmo 11 – Criação de *Links* para a Biblioteca do MATLAB

```

1 ln -s "MATLAB/R2018a/bin/glnxa64/libmat.so" "MATLAB/R2018a/sys/os/glnxa64/"
2 ln -s "MATLAB/R2018a/extern/bin/glnxa64/libMatlabDataArray.so" "MATLAB/R2018a/sys/os/glnxa64/"
3 ln -s "MATLAB/R2018a/extern/bin/glnxa64/libMatlabEngine.so" "MATLAB/R2018a/sys/os/glnxa64/"
4 ln -s "/usr/lib/x86_64-linux-gnu/libstdc++.so.6" "MATLAB/R2018a/sys/os/glnxa64/libstdc++.so.6"

```

Para permitir que as aplicações desenvolvidas recebessem as informações de qualidade da comunicação utilizadas no protocolo, foi necessário adicionar a palavra chave “virtual” no método “EndReceive” da classe “WifiPhy” no arquivo “ns3/src/wifi/model/wifi-phy.h” permitindo sua sobrescrita, como mostra o Algoritmo 12. Sem essa modificação as aplicações não recebem os parâmetros de qualidade da comunicação ao receberem uma mensagem.

Algoritmo 12 – Modificação no Módulo Wifi

```

1 virtual void EndReceive(Ptr<Packet> packet, WifiPreamble preamble, MpduType mpdutype, Ptr<Event> event);

```

Uma atualização na biblioteca auxiliar do NS-3 e no Python fez com que o código anteriormente estável apresentasse falhas na construção. Por fim, foi possível identificar o problema e chegar a uma solução, que foi modificando o arquivo “ns3/src/wscript” substituindo a linha 396 pelo trecho de código apresentado no Algoritmo 13. A linha modificada estava erroneamente gerando dados encadeados que faziam com que a rotina “print” na linha 413 do mesmo arquivo apresentasse o erro que impedia a construção do módulo.

Algoritmo 13 – Modificação no NS-3 para Inclusão do Módulo

```
1 lib = self._self_libs(env,"%s%s-%s%s" % (wutils.APPNAME, wutils.VERSION,name[4:],env.BUILD_SUFFIX),'${libdir}')
2 #list of list to list
3 libs = list();
4 for x in lib:
5     if type(x) == list:
6         for y in x:
7             libs.append(y)
8     else:
9         libs.append(x)
```