



**UNIVERSIDADE DO ALGARVE**

**REDES DE TRANSMISSÃO DE DADOS EM MEIOS  
SUBMARINOS**

**EMANUEL ANTÓNIO BAPTISTA INOCÊNCIO**

**Dissertação**

**MESTRADO EM ENGENHARIA ELÉCTRICA E ELECTRÓNICA**

Trabalho efectuado sob a Orientação de:

Professor Doutor Jânio Monteiro & Professor Doutor António João Silva

**2015**



# REDES DE TRANSMISSÃO DE DADOS EM MEIOS SUBMARINOS

## DECLARAÇÃO DE AUTORIA DE TRABALHO

Declaro ser o autor deste trabalho, que é original e inédito. Autores e trabalhos consultados estão devidamente citados no texto e constam da listagem de referências bibliográfica incluída.

---

© 2014, EMANUEL ANTÓNIO BAPTISTA INOCÊNCIO

*A Universidade do Algarve tem o direito, perpétuo e sem limites geográficos, de arquivar publicitar este trabalho através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, de o divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.*



À minha família, em especial à minha filha,  
Marcyah Adriana Abreu Inocência



# Agradecimentos

Em primeiro lugar gostaria de agradecer aos Prof.Doutor Jânio Miguel E.Ferreira Monteiro e Prof.Doutor António João silva pelas suas orientações e por me terem assegurado todas as condições necessárias ao desenvolvimento deste trabalho. Gostaria de destacar os conhecimentos que me transmitiram, o tempo que me dispensaram, encorajamento, avaliação dos resultados obtidos e revisão deste texto e pelo incentivo que sempre manifestaram, estimulando-me no meu trabalho.

Gostaria também de agradecer a ajuda dos utilizadores do *nsmiracle-users-mailing list* que participaram no projecto WOSS (*World Ocean Simulation System*), nomeadamente ao Federico Guerra, ao Paolo Casari e ao Michele Zorzi.

Agradeço também o apoio do Instituto Superior de Engenharia da Universidade do Algarve, especialmente aos meus colegas do curso e aos Professores da Área Departamental de Engenharia Electrotécnica.

Ao meu colega de Cabo Verde Mateus vai um “Obrigado” pelo contacto permanente que tivemos um com o outro durante o período de quase 4 anos; afinal começámos junto o curso e tivemos no início as mesmas dificuldades.

Queria também agradecer a Nelida Abreu, minha esposa, que esteve presente activamente nestes dois últimos anos da minha formação, apoiando-me e não me deixando desistir do curso. Não deixo de realçar que vários acontecimentos ocorreram neste período, tendo sido um deles, a possibilidade de me tornar pai pela primeira vez.

Não deixaria de agradecer a Deus e aos meus Pais por me terem disponibilizado os meios para conseguir fazer o curso, e pelas palavras de encorajamento. A todos os amigos que me incentivaram neste trabalho, a outras pessoas que, direta ou indiretamente, contribuíram para a elaboração deste trabalho.





# Resumo

Há um crescente interesse nas comunicações submarinas e aos dados oceanográficos nos últimos anos, devido à importância dos oceanos para o ser humano. As redes acústicas submarinas têm grande potencial para suportar uma grande variedade de aplicações, monitorização ambiental, vigilância submarina, prevenção de catástrofes, exploração de petróleo e monitorização de poluição. Apesar da comunicação acústica submarina ter sido estudada durante várias décadas, trabalhos em redes acústicas submarinas, projectos de implementação de protocolos de acesso ao meio (MAC) e protocolos de encaminhamento apenas estão numa fase inicial como campo de pesquisa devido ao elevado custo dos nós e dos navios de distribuição. Neste âmbito, a simulação torna-se como uma ferramenta fundamental para avaliação do desempenho da rede e um grande auxiliar no projecto de implementação dos protocolos em aplicações onde os cenários são complexos, devido ao elevado custo e inviabilidade em pôr na prática. Neste sentido, tendo como objectivo proporcionar um ambiente fiável de simulação de redes acústicas submarinas, optámos pelo simulador de rede WOSS (*World Ocean Simulation System*) de forma a fornecer uma reprodução mais detalhada da propagação acústica.

O Objectivo central desta dissertação consiste na implementação de um protocolo de acesso ao meio DACAP (*Distance Aware Collision Avoidance Protocol*) para redes ad-hoc acústica submarina -caracterizadas por longos atrasos de propagação e diferentes requisitos de potência de Transmissão/Recepção- e de seguida testá-lo no referido simulador de rede.

Os resultados são mostrados fazendo comparação do desempenho da rede acústica submarina para diversos parâmetros, entre o protocolo DACAP e diversos mecanismos de acesso ao meio existentes no simulador de rede, para um conjunto de topologias diferentes e para diferentes tamanhos de tramas.

## Palavras-chave

Redes acústicas submarinas, simulação, Protocolos MAC, Modelo propagação, Bellhop, simulador de rede ns-2, WOSS.



# Abstract

There is an increasing interest in underwater communications and oceanographic data due to the importance of oceans to human beings. The underwater acoustic networks have great potential to support a wide variety of applications, environmental monitoring, underwater surveillance, disaster preparedness, oil exploration and pollution monitoring. Despite the fact that underwater acoustic communication has been studied for several decades, underwater acoustic networks, projects and implementation of medium access (MAC) and routing protocols are only at an early stage as a field of research, due to the high cost of nodes and the distribution vessel. In this context, the simulation becomes a fundamental tool for the evaluation of network performance and a great help in the implementation of protocols and applications where the scenarios are complex, due to the high cost and impracticality of putting in practice. In This sense, with the aim to provide a secure environment for simulating underwater acoustic networks we chose the WOSS (World Ocean Simulation System) network simulator, which is capable of providing detailed reproduction of the acoustic propagation.

The central objective of this thesis is the implementation of the DACAP (Distance Aware Collision Avoidance Protocol) protocol medium access for underwater acoustic ad hoc networks,- characterized by long propagation delays and different power requirements of Transmission / Reception- and then tested it in the network simulator.

The results comprise the comparison of performance of the underwater acoustic network, when various parameters of the protocol DACAP are used and against various access mechanisms available in the network simulator, for a set of different topologies and different frames sizes.

## Keywords

Underwater acoustic networks, simulation, MAC protocols, propagation modeling, Bellhop, Network simulator ns-2, WOSS.



# Conteúdo

Agradecimentos .....	iii
Resumo .....	v
Abstract .....	vii
Índice de Figuras.....	xii
Índice de Tabelas .....	xvii
Lista de Abreviaturas .....	xviii
<b>Capítulo 1 Introdução .....</b>	<b>1</b>
1.1. Motivação .....	2
1.2. Objectivo da dissertação .....	5
1.3. Organização da dissertação.....	6
<b>Capítulo 2 Conceitos gerais sobre propagação acústica e Simulador de redes submarinas.....</b>	<b>7</b>
2.1. Principais características das redes acústicas submarinas .....	9
2.2. Arquitecturas de comunicação em redes acústicas submarinas .....	10
2.2.1. Arquitectura bidimensional.....	10
2.2.2. Arquitectura tridimensional .....	11
2.3. Comunicação submarina .....	13
2.3.1. Propagação de sinais no ambiente acústico submarino.....	13
2.4. Canal acústico submarino .....	14
2.4.1. Velocidade de propagação do som.....	16
2.4.2. Trajetórias de propagação .....	18
2.4.3. Perdas de transmissão .....	20
2.5. Simulador de rede acústica submarina .....	22
2.5.1. Modelo físico acústico submarino no simulador NS-2 .....	23
2.5.1.1. Função de transferência do modelo de propagação.....	24
2.5.1.2. Função de transferência do modelo do canal .....	26
2.5.1.3. Função de transferência do modelo de modulação .....	27
2.5.1.4. Função de transferência do modelo físico.....	28
2.5.1.5. Análise do modelo de propagação do NS-2 .....	29
2.5.2. Simulador de rede WOSS .....	32
2.5.2.1. Diagrama de blocos do simulador WOSS.....	33
2.5.2.2. Definição das classes no simulador WOSS.....	34
2.5.2.3. Base de dados de batimetria do GEBCO .....	35
2.5.2.4. Base de dados do perfil da velocidade propagação do som .....	37
2.5.2.5. Base de dados de sedimentos DECK41 .....	38
2.6. Conclusões do capítulo .....	39
<b>Capítulo 3 Mecanismos de controlo de acesso ao meio em redes acústicas submarinas.....</b>	<b>41</b>
3.1. Introdução .....	41
3.2. Métodos de acesso múltiplo ao meio .....	42
3.3. Mecanismo de controlo de acesso ao meio para redes sem fios.....	43
3.3.1. Protocolo ALOHA .....	44
3.3.2. Protocolo <i>Slotted</i> ALOHA .....	46
3.3.3. Protocolo <i>Carrier Sense Multiple Access (CSMA)</i> .....	47
3.3.3.1. Protocolo CSMA/CD .....	48
3.3.3.2. Protocolo CSMA/CA .....	50
3.3.3.3. Problemas do nó escondido e exposto.....	52
3.3.4. Protocolo <i>Multiple Access with Collision Avoidance</i> .....	53
3.3.5. Protocolo <i>MACA-Wireless</i> .....	54
3.4. Mecanismos de controlo de acesso ao meio propostos para redes de acústicas submarina .....	56

3.4.1. Protocolo <i>Slotted FAMA</i> .....	56
3.4.1.1. Mecanismo de backoff utilizado pelo protocolo <i>Slotted FAMA</i> .....	59
3.4.2. Protocolo de configuração do <i>handshake</i> .....	60
3.4.3. Protocolo <i>Tone-Lohi (T-Lohi)</i> .....	62
3.4.4. Protocolo PDAP ( <i>Propagation Delay Aware Protocol</i> ) .....	63
3.4.5. Protocolo DACAP ( <i>Distance Aware-Collision Avoidance Protocol</i> ).....	65
3.4.5.1. Versão do protocolo DACAP sem ACK.....	67
3.4.5.2. Versão do protocolo DACAP com ACK .....	67
3.4.5.3. Mecanismo de <i>backoff</i> utilizado pelo protocolo DACAP .....	68
3.5. Protocolos de encaminhamento.....	69
3.6. Desenvolvimento das redes acústicas submarinas .....	70
3.7. Conclusões do capítulo .....	71
<b>Capítulo 4 Implementação do Mecanismo DACAP.....</b>	<b>73</b>
4.1. Visão geral .....	73
4.2. Funcionamento da camada MAC no simulador de rede .....	74
4.3. Implementação do protocolo DACAP proposto .....	78
4.3.1. Procedimentos e testes iniciais.....	78
4.3.2. Algoritmo do protocolo DACAP .....	79
4.3.2.1. Módulo de transmissão .....	81
4.3.2.2. Módulo de recepção .....	82
4.3.2.3. Módulo gestão do estado do canal .....	82
4.3.2.4. Módulo gestão do estado do <i>backoff</i> .....	83
4.3.2.5. Módulo gestão do estado de transmissão .....	85
4.3.2.6. Módulo gestão do estado de recepção.....	86
4.4. Protocolo DACAP <i>with Warning</i> .....	88
4.4.1. Módulo gestão do estado de recepção do protocolo DACAP com <i>warning</i> .....	88
4.4.2. Diagrama de sequência do protocolo DACAP com <i>warning</i> .....	90
4.5. Conclusões do capítulo .....	93
<b>Capítulo 5 Simulações .....</b>	<b>95</b>
5.1. Estrutura da simulação no WOSS .....	95
5.2. Descrição dos protocolos utilizados na simulação .....	97
5.3. Definição dos parâmetros de avaliação de desempenho .....	98
5.4. Simulações efectuadas e resultados .....	99
5.4.1. Análise comparativa dos protocolos MAC para 2 nós emissores e 1 recetor.....	100
5.4.1.1. Arquitetura simétrica.....	100
5.4.1.2. Arquitetura assimétrica .....	108
5.4.1.3. Sumário .....	113
5.4.2. Análise comparativa dos protocolos MAC para 4 nós emissores e 1 recetor.....	114
5.4.3. Análise comparativa dos protocolos MAC para 4 nós emissores e 2 recetores .....	121
5.5. Conclusões .....	126
<b>Capítulo 6 Conclusões e trabalhos futuros .....</b>	<b>131</b>
6.1. Conclusão.....	131
6.2. Trabalhos futuros .....	133
<b>Bibliografia.....</b>	<b>135</b>
<b>Anexo .....</b>	<b>141</b>
Anexo A. Modelo de canal e da camada física no NS-2.....	141
Anexo B. Função de transferência de cada um dos modelos .....	142
B.1. Função de transferência do modelo físico .....	142
B.2. Função de transferência do modelo canal.....	143
B.3. Função de transferência do modelo modulação.....	144
B.4. Função de transferência do modelo propagação.....	145
Anexo C. Análise comparativa entre protocolo ALOHA sem ACK e ALOHANEW sem ACK .....	147
Anexo D. Diagrama de estado dos módulos utilizados na implementação do DACAP .....	149
D.1. Diagrama de estado do módulo gestão do estado de transmissão .....	149

D.2. Diagrama de estado do módulo gestão do estado de receção.....	150
D.3. Diagrama de estado do módulo gestão do estado de receção com <i>warning</i> .....	151
Anexo E. Resultados dos testes.....	152
E.1. Análise comparativa dos protocolos MAC para 2 nós emissores e 1 recetor .....	152
E.1.1. Arquitetura simétrica para tamanho pacote 600 byte .....	152
E.1.2. Arquitetura simétrica para tamanho pacote 50 byte .....	154
E.1.3. Arquitetura assimétrica para tamanho pacote 600 byte .....	156
E.1.4. Arquitetura assimétrica para tamanho pacote 50 byte .....	158
E.2. Análise comparativa dos protocolos MAC para 4 nós emissores e 1 recetor .....	160
E.2.1. Arquitetura simétrica para tamanho pacote 600 byte .....	160
E.2.2. Arquitetura assimétrica para tamanho pacote 600 byte .....	162

# Índice de Figuras

Figura 1.1: Configuração típica de redes acústicas submarinas.....	4
Figura 2.1: Arquitetura bidimensional das redes acústicas submarinas.....	11
Figura 2.2: Arquitetura tridimensional das redes acústicas submarinas. ....	12
Figura 2.3: Variação da velocidade de propagação do som no mar (a) e canal SOFAR ( <i>Sound Fixing And Ranging</i> ) (b).....	17
Figura 2.4: Refração do som entre dois meios com velocidades distintas (a) e em meio com variação gradual de velocidade (b).....	18
Figura 2.5: Zonas de sombra e de convergência.....	19
Figura 2.6: Reflexão do sinal acústico na superfície e no fundo. ....	20
Figura 2.7: Relação Sinal Ruído em função da frequência.....	21
Figura 2.8: Modelo do canal e da camada física acústica submarina no NS-2.....	24
Figura 2.9: Função de transferência do modelo de propagação.....	24
Figura 2.10: Função de transferência do modelo canal. ....	26
Figura 2.11: Função de transferência do modelo de modulação.....	27
Figura 2.12: Função de transferência do modelo físico. ....	28
Figura 2.13: Cenário da simulação para avaliar o modelo físico no NS-2. ....	29
Figura 2.14: Efeito da profundidade em relação ao atraso de propagação. ....	30
Figura 2.15: Atraso de propagação em relação a distância entre os nós.....	31
Figura 2.16: Potência de recepção em relação a distância entre os nós.....	31
Figura 2.17: Diagrama de blocos do simulador de rede WOSS. ....	34
Figura 2.18: Mapa do perfil batimétrico utilizando a GEBCO_08 <i>grid</i> . ....	37
Figura 2.19: Diagrama para o cálculo do SSP. ....	38
Figura 3.1: Funcionamento do ALOHA exemplificando o conceito de colisão e retransmissão. ....	46
Figura 3.2: Funcionamento do <i>Slotted</i> ALOHA. ....	47
Figura 3.3: Fluxograma do algoritmo de funcionamento do CSMA/CD. ....	49
Figura 3.4: Estrutura de funcionamento do CSMA/CA.....	51
Figura 3.5: Funcionamento do CSMA exemplificando o problema dos nós expostos e escondido..	52
Figura 3.6: Estrutura de funcionamento do protocolo MACA. ....	53
Figura 3.7: Protocolo <i>MACA-Wireless</i> : (a) funcionamento, (b) sem resposta CTS e (c) pacote <i>WAIT</i> . .....	54
Figura 3.8: Teste de potências durante troca de tramas RTS e CTS.....	55
Figura 3.9: Estrutura de funcionamento do protocolo <i>Slotted</i> FAMA.....	58
Figura 3.10: Estrutura de funcionamento do protocolo baseado na configuração do <i>handshake</i> .....	61



Figura 3.11: Estrutura de funcionamento do protocolo <i>T-Lohi</i> .	63
Figura 3.12: Estrutura de funcionamento do protocolo DACAP.	66
Figura 4.1: Diagrama de sequência dos pacotes para eventos TX/RX no WOSS.	75
Figura 4.2: Diagrama de sequência dos pacotes para eventos TX/RX no WOSS para camada física e MAC.	77
Figura 4.3: Arquitetura da camada MAC utilizada para implementação do DACAP.	81
Figura 4.4: Utilização do módulo gestão do estado de <i>backoff</i> pelo módulo gestão do estado de transmissão.	84
Figura 4.5: Módulo gestão do estado de <i>backoff</i> .	84
Figura 4.6: Diagrama de sequência de pacotes e primitivas do protocolo DACAP com <i>warning</i> .	92
Figura 5.1: Estrutura da simulação no simulador de rede acústico WOSS.	96
Figura 5.2: Cenário utilizado na simulação para análise comparativa dos protocolos para tamanhos de pacotes de 50 e 600 bytes.	100
Figura 5.3: Velocidade de propagação do som em relação a profundidade, utilizado no cenário apresentado na figura 5.2.	102
Figura 5.4: Mapa do perfil batimétrico utilizado no cenário apresentado na figura 5.2.	103
Figura 5.5: <i>Average Throughput per Node</i> em função do <i>Data Generation Rate per Node</i> dos protocolos para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.	104
Figura 5.6: <i>Normalized Throughput</i> em função do <i>Data Generation Rate per Node</i> dos protocolos para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.	105
Figura 5.7: <i>Success Ratio</i> em função do <i>Data Generation Rate per Node</i> dos protocolos para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.	106
Figura 5.8: <i>Application Success Ratio</i> em função do <i>Data Generation Rate per Node</i> dos protocolos para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.	107
Figura 5.9: Tempo gasto em mecanismos de controlo ( <i>Overhead</i> ), em função do <i>Data Generation Rate per Node</i> dos protocolos para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.	108
Figura 5.10: Cenário utilizado na simulação quando o nó 2 desloca para distância de 25 metros na horizontal em relação ao nó 0.	109
Figura 5.11 : <i>Average Throughput per Node</i> em função do <i>Data Generation Rate per Node</i> dos protocolos utilizando o cenário da figura 5.10 para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.	110
Figura 5.12: <i>Normalized Throughput</i> em função do <i>Data Generation Rate per Node</i> dos protocolos utilizando o cenário da figura 5.10 para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.	111
Figura 5.13 : <i>Success Ratio</i> em função do <i>Data Generation Rate per Node</i> dos protocolos utilizando o cenário da figura 5.10 para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.	111
Figura 5.14: <i>Application Success Ratio</i> em função do <i>Data Generation Rate per Node</i> dos protocolos utilizando o cenário da figura 5.10 para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.	112

Figura 5.15: Tempo gasto em mecanismos de controlo ( <i>Overhead</i> ) em função do <i>Data Generation Rate per Node</i> dos protocolos utilizando o cenário da figura 5.10 para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.....	113
Figura 5.16: Cenário da simulação para análise comparativa dos protocolos considerando o nó 4 como nó recetor para tamanho do pacote de 600 bytes. ....	115
Figura 5.17: Velocidade de propagação do som em relação a profundidade utilizando o cenário da simulação apresentado na figura 5.16. ....	116
Figura 5.18: Mapa do perfil batimétrico utilizado no cenário apresentado na figura 5.16. ....	116
Figura 5.19: <i>Average Throughput per Node</i> em função do <i>Data Generation Rate per Node</i> dos protocolos considerando (a) Nó 4 como recetor (b) Nó 2 como recetor.....	117
Figura 5.20: <i>Normalized Throughput</i> em função do <i>Data Generation Rate per Node</i> dos protocolos considerando (a) Nó 4 como recetor (b) Nó 2 como recetor. ....	118
Figura 5.21: <i>Success Ratio</i> em função do <i>Data Generation Rate per Node</i> dos protocolos considerando (a) Nó 4 como recetor (b) Nó 2 como recetor. ....	118
Figura 5.22: <i>Application Success Ratio</i> em função do <i>Data Generation Rate per Node</i> dos protocolos considerando (a) Nó 4 como recetor (b) Nó 2 como recetor.....	119
Figura 5.23: Tempo gasto em mecanismos de controlo ( <i>Overhead</i> ) em função do <i>Data Generation Rate per Node</i> dos protocolos considerando (a) Nó 4 como recetor (b) Nó 2 como recetor...	120
Figura 5.24: Cenário utilizado na simulação para análise comparativa dos protocolos considerando dois nós recetores (nó 4 e nó 5) para tamanho de pacote de 600 bytes. ....	121
Figura 5.25: <i>Average Throughput per Node</i> em função do <i>Data Generation Rate per Node</i> dos protocolos para tamanho do pacote de 600 bytes. ....	123
Figura 5.26: <i>Normalized Throughput</i> em função do <i>Data Generation Rate per Node</i> dos protocolos para tamanho do pacote de 600 bytes. ....	123
Figura 5.27: <i>Success Ratio</i> em função do <i>Data Generation Rate per Node</i> dos protocolos para tamanho do pacote de 600 bytes. ....	124
Figura 5.28: <i>Application Success Ratio</i> em função do <i>Data Generation Rate per Node</i> dos protocolos para tamanho do pacote de 600 bytes. ....	124
Figura 5.29: Tempo gasto em mecanismos de controlo ( <i>Overhead</i> ) em função do <i>Data Generation Rate per Node</i> dos protocolos para tamanho do pacote de 600 bytes.....	125





# Índice de Tabelas

Tabela 2.1: Largura de banda disponível para as diferentes distâncias em canais submarinos .....	15
Tabela 5.1: Tabela com pequena descrição dos protocolos utilizados na simulação .....	98
Tabela 5.2: Coordenadas e profundidades dos nós utilizados na simulação .....	101
Tabela 5.3: Coordenadas e profundidades dos nós utilizados na simulação do cenário da figura 5.16 .....	115
Tabela 5.4: Coordenadas e profundidades dos nós utilizados na simulação do cenário da figura 5.24 .....	122
Tabela 5.5: Resumo sobre o desempenho dos protocolos de acesso ao meio para o cenário da figura 5.2.....	128
Tabela 5.6: Resumo sobre o desempenho dos protocolos de acesso ao meio para o cenário da figura 5.16.....	129
Tabela 5.7: Resumo sobre o desempenho dos protocolos de acesso ao meio para o cenário da figura 5.24.....	129

# Lista de Abreviaturas

ACK	Acknowledgment
AUV	Autonomous Underwater Vehicles
API	Application Programming Interface
ALAN	Rede Área Local Acústica
BODC	British Oceanographic Data Centre
BPSK	Binary Phase Shift Keying
CSMA	Carrier Sense Multiple Access
CDMA	Code Division Multiple Access
CTS	Clear to Send
CD	Collision Detection
CA	Collision Avoidance
CW	Congestion Window
CBR	Constant Bit Rate
DACAP	Distance Aware Collision Avoidance Protocol
DSP	Digital Signal Processors
DCF	Distributed Coordination Function
DIFS	DCF Inter Frame Spaces
FSK	Frequency Shift Keying
FHSS	Frequency Hopping Spread Spectrum
FDMA	Frequency Division Multiple Access
FAMA	Floor Acquisition Multiple Access
FH-BFSK	Frequency Hopping - Binary Frequency Shift Keying
GEBCO	General Bathymetric Chart of the Oceans
GPS	Global Positioning System
ISI	Interferência Intersimbólica
LWT	Listen While Talk
MAC	Medium Access Control
MACA	Multiple Access with Collision Avoidance
MACA-W	Multiple Access with Collision Avoidance - Wireless
NAV	Network Allocation Vector
NS-2	Network Simulator, versão 2
NTU	Nephelometric Turbidity Unit
PSK	Phase Shift Keying
PCF	Point Coordination Function
PIFS	PCF Inter Frame Spaces
PDAP	Propagation Delay Aware Protocol
PRN	Rede de Pacotes Rádio
RAS	Redes Acústicas Submarinas
RF	Rádio Frequência
RTS	Request to Send
SOFAR	Sound Fixing And Ranging
SNR	Signal-to-Noise Ratio
SINR	Interferência da Relação Sinal Ruído
SSP	Sound Speed Profile
SIFS	Short Inter Frame Spaces
SIR	Interferência Sinal Ruído
TTP	Tempo de Transmissão do Pacote

TDMA	Time Division Multiple Access
TGI	Time Guard Interval
WOSS	World Ocean Simulation System
WHOI	Woods Hole Oceanographic Institution
WOA	World Ocean Atlas





# Capítulo 1

## Introdução

As redes acústicas submarinas contêm grande potencial para suportar uma grande variedade de aplicações, incluindo entre outras a navegação, a pesca, a ecologia, a monitorização ambiental, a vigilância submarina, a prevenção de catástrofes, a exploração de petróleo, a deteção de intrusos e a monitorização de poluição [1].

Nos últimos anos tem-se assistido a um crescimento na importância das comunicações submarinas [1]. Neste âmbito a comunicação entre sistemas autónomos, constituídos por nós fixos e móveis, tornar-se-ão uma importante ferramenta com repercussões directas em inúmeras aplicações [2]. Na sua maioria, tais aplicações requerem a transmissão de sinais de sensores numa estrutura ad-hoc, envolvendo protocolos de Controlo de Acesso ao Meio (MAC) e de encaminhamento [2].

No entanto as comunicações submarinas diferem das comunicações em rádio frequência clássicas no que toca às características do canal onde os sinais de acústica são transmitidos. Para além da velocidade de propagação mais baixa dos sinais acústicos, as características do canal são intrinsecamente dependentes das características da água como meio de propagação, que dependem da sua temperatura, salinidade e pressão. Outros factores como a profundidade da água, o tipo de sedimento e perfil batimétrico do fundo, ou a posição dos pontos emissores em relação à superfície e fundo, influenciam o débito de transmissão. Em termos de ruído, a turbulência, os ventos, os animais e as condições atmosféricas adicionam-se ao ruído térmico.

Em tais condições, o estabelecimento de comunicações pode tornar-se complexo, havendo mesmo uma ausência de um *link budget* que consiga caracterizar tais comunicações [3].

Neste contexto os simuladores de redes tornam-se como ferramentas fundamentais para avaliação do desempenho das redes acústicas submarinas e um grande auxiliar no projecto de implementação dos protocolos em aplicações onde os cenários são complexos, devido ao elevado custo e inviabilidade em pôr na prática.

Embora têm sido feitos alguns esforços para construir nós mais baratos com menor intervalo de transmissão e reduzindo as experiências para redes com menor complexidade, a simulação continua a ser uma ferramenta importante na avaliação do desempenho da rede.

Actualmente não existe nenhuma ferramenta de simulação padrão específica para redes acústicas submarinas. Uma abordagem mais amplamente adotada é a reutilização dos simuladores terrestres das redes sem fios, alterando o modelo de propagação sem fios de forma a aproximar-se ao modelo de propagação acústico submarino.

Isso normalmente é feito definindo o valor da velocidade do som um valor constante de 1,5 km / s, utilizando fórmulas empíricas para atenuação e densidade espectral de ruído.

Esta abordagem, no entanto, torna-se bastante limitada, devido a que as fórmulas empíricas geralmente possuem uma aproximação, cuja precisão depende muito do cenário e dos parâmetros de implementação (isto é águas com pequenas profundidades em relação a águas com grandes profundidades, água quente em relação a água fria, fundo do mar e etc...). Além disso, a variação dos factores ambientais como a temperatura da água e da morfologia do fundo do mar poderá não ter um impacto desprezível sobre a propagação.

Com o objectivo de proporcionar um ambiente fiável de simulação de redes acústicas submarinas, neste trabalho utilizaremos o simulador de rede WOSS, de forma a fornecer uma reprodução mais detalhada da propagação acústica (isto é, através do modelo de propagação *belhop* “*ray tracing*” em vez de fórmulas empíricas).

O simulador WOSS será integrado com outras ferramentas de simulação disponíveis como as extensões do MIRACLE para o NS-2 (*Network Simulator -2*) de forma a fornecer uma ferramenta completa e personalizada, incluindo a propagação acústica, o modelo da camada física e *cross-layer* para especificação dos protocolos de encaminhamento e protocolos de controlo de acesso ao meio.

## 1.1. Motivação

Na década passada as redes de comunicação sem fios sofreram um desenvolvimento extraordinário tendo havido desde então uma necessidade crescente de comunicar a débitos mais elevados, de forma a acomodar novos serviços. Pesquisas recentes apontam para um crescente interesse em ampliar redes de sensores sem fios terrestres ao meio submarino [4]. Aplicações submarinas, tais como controlo de dados oceanográficos, assistência na navegação, vigilância submarina, antecipação de informações sobre catástrofes, exploração submarinas e controlo de poluição podem ser contornados através de implementação de redes acústicas submarinas [4].

Neste âmbito o laboratório de processamento de sinal da Universidade o Algarve em parceria com os parceiros europeus incluindo cinco centros de investigação e empresas da Itália, Noruega e Suécia pretendem criar uma rede sem fios para comunicar através do som dentro de água. Este projecto que recebeu o nome de UAN (*Underwater Acoustic Network*), consiste em desenvolver metodologias e tecnologias operacionais para implementação de redes de comunicação acústica submarina a serem integradas como componentes de uma rede multimedia para vigilância e monitorização de infra-estruturas críticas no mar.

A UAN será composta por nós móveis, fixos, e uma estação de base que assumirá de cérebro de toda a operação. Esta rede terá capacidade para cobrir de forma dinâmica um perímetro subaquático de cem quilómetros quadrados com os nós da rede a transmitirem mensagens entre si através de uma espécie de correio electrónico: enviando texto e imagens uns aos outros usando ondas sonoras, que se propagam facilmente debaixo de água. Todos os nós desta rede estarão equipados com vários sensores, as armas necessárias para detetar potenciais ameaças a infra-estruturas estratégicas como plataformas petrolíferas ou instalações de geração de energia, tanto em alto mar como em zonas costeiras.

Os diversos sensores captarão informação sobre, a temperatura da água, a velocidade das correntes ou a oscilação da coluna de água, monitorando a todo o tempo o perímetro que defendem. Os nós móveis, ou veículos autónomos subaquáticos, serão cruciais no processo de recolha de informação, uma vez que podem ser destacados para ir a uma determinada localização confirmar dados ou recolher imagens cruciais para tomar uma decisão face a uma potencial ameaça. Os nós fixos têm como função participar na recolha de informação ambiental estratégica para o processo de vigilância e deteção.

A controlar as operações estará uma cadeia de sensores, a estação de base, que terá uma antena, uma unidade de telemetria, um modem e vários sensores ao longo de um cabo que termina num flutuador abaixo da superfície. Um exemplo de uma configuração típica da rede acústica submarina é apresentado na figura 1.1 (Adaptada a referência [6]).

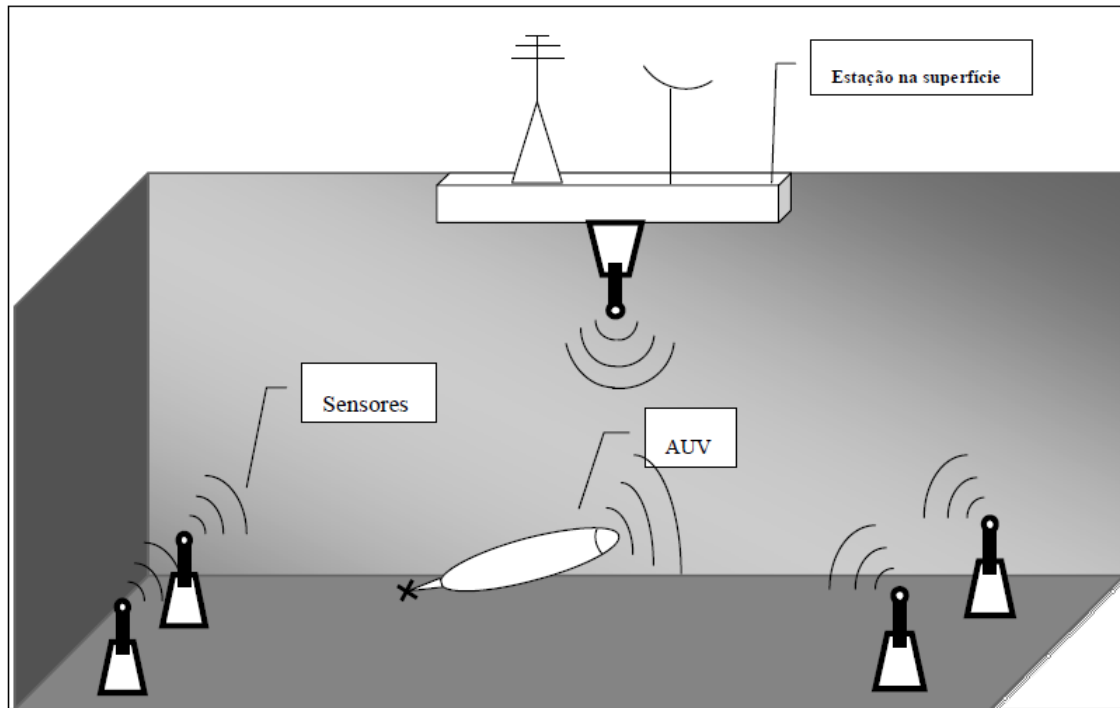


Figura 1.1: Configuração típica de redes acústicas submarinas.

Apesar de poderem potencialmente vir a ser úteis em diversas aplicações reais, as redes acústicas submarinas colocam diversos desafios e limitações que derivam do ambiente submarino [7][8][9]. A variabilidade do meio submarino é tal, e a velocidade, que as condições mudam a cada segundo. Em termos práticos a transmissão de uma mensagem poderá levar vários segundos, ficando assim sujeita à variabilidade do meio de propagação.

É à luz deste problema que o conceito de rede, foi introduzido de forma prioritária no projecto UAN no contexto da comunicação debaixo de água. Até agora a comunicação submarina tem vindo a ser feita ponto a ponto, o que cria grandes problemas devido a variabilidade do meio. O conceito de rede que vamos trabalhar vem permitir fazer uma gestão mais eficaz de todo o processo comunicacional, permitindo-nos escolher qual é o melhor caminho em dado momento para colocar os nós da rede em contacto, tornando a troca de informação mais rápida e efetiva.

As razões atrás apresentadas motivaram a escolha do estudo das redes de transmissão de dados em meios submarinos, como tema desta dissertação.

## 1.2. Objectivo da dissertação

Tendo em consideração as limitações do canal acústico e as particularidades do ambiente submarino quanto à propagação tais como largura de banda, atenuações, interferências, ruído, alcance, velocidade de propagação do som, perdas de transmissão, trajetórias de propagação e potências de transmissão pretende-se considerar a comunicação entre vários nós da rede e analisar os efeitos que tais comunicações têm no aumento da interferência mútua, de acordo com vários mecanismos de controlo de acesso ao meio.

Para tal, pretende-se melhorar as ferramentas existentes de simulação dos meios de transmissão submarinas para que traduzam a complexidade da transmissão submarina. Neste âmbito no presente trabalho pretende-se analisar, implementar e propor melhorias nos principais protocolos de controlo de acesso ao meio para redes acústicas submarinas. Tendo como ponto de partida a análise do princípio de funcionamento de vários mecanismos de acesso ao meio existentes pretende-se implementar um mecanismo de controlo de acesso ao meio designado por DACAP (*Distance Aware Collision Avoidance Protocol*) e testá-lo no simulador de rede WOSS utilizando diversas configurações.

Nesse sentido, após a implementação do referido mecanismo de controlo de acesso ao meio (DACAP) serão realizados diversos testes de forma a verificar a melhoria introduzida por este mecanismo de controlo de acesso ao meio em diversos contextos, incluindo variações no tamanho de trama, do número de nós e na sua disposição. Os resultados obtidos pelo protocolo DACAP serão comparados com os resultados dos protocolos como o ALOHA com e sem *Acknowledgment* [10], ALOHA com escuta ao meio e sem *Acknowledgment* (sem ACK) e o CSMA (*Carrier Sense Multiple Access*). O objetivo deste estudo é o de analisar em que circunstâncias cada um dos protocolos acima referidos apresentam melhor desempenho nas redes acústicas submarinas.

### 1.3. Organização da dissertação

Esta dissertação está organizada em seis capítulos. Neste capítulo (capítulo 1) é apresentada uma visão geral sobre redes acústicas submarinas, o objectivo da dissertação e a motivação para a realização deste estudo.

No capítulo 2 será realizada uma revisão bibliográfica sobre redes acústicas submarinas, apresentando: as suas características; as principais aplicações; a sua arquitetura; o conceito de canal acústico com as suas particularidades tais como latência, largura de banda, ruído, alcance, velocidade de propagação do som, perdas de transmissão, trajectórias de propagação e potências de transmissões. Também neste capítulo será apresentada o modelo do canal acústico, com vista ao desenvolvimento de simulações no NS-2 (*Network Simulator*, versão 2), assim como também será apresentado o simulador de rede WOSS utilizado posteriormente.

No capítulo 3 será feita a descrição dos principais protocolos de controlo de acesso ao meio, analisando-se os seus mecanismos de funcionamento que permitem efetuar o acesso ao canal e melhorar o desempenho da rede. Também serão apresentados os protocolos de encaminhamento para redes acústicas submarinas. O capítulo será encerrado apresentando os principais desenvolvimentos das redes acústicas submarinas.

No capítulo 4, descreve-se os passos que conduziram à implementação do mecanismo de controlo de acesso ao meio DACAP.

No capítulos 5, serão apresentados os resultados comparativos entre os diversos mecanismos MAC, obtidos para um conjunto de topologias diferentes e para diferentes tamanhos de tramas. No capítulo 6, apresentaremos as conclusões, sugerindo alguns trabalhos futuros.

## Capítulo 2

# Conceitos gerais sobre propagação acústica e Simulador de redes submarinas

O desenvolvimento da acústica submarina teve grande impulso na segunda guerra mundial, devido à necessidade militar em identificar navegações inimigas e à necessidade de comunicação entre submarinos. No entanto, a teoria de comunicação acústica submarina desenvolveu-se apenas nas décadas seguintes, com os trabalhos de Wilson [11] (1960), Urick *et al.* [12] (1983), Coates [13] (1989) e Berkhovskikh *et al.* [14] (1991). Em paralelo com o desenvolvimento das técnicas de comunicação digital de sinais e os avanços da microeletrónica, incluindo os DSPs (*Digital Signal Processors*), permitiram o fabrico de modems acústicos submarinos compactos, de baixo consumo e com grande capacidade de processamento, permitindo a implementação de sistemas de modulação e codificação para transmissão digital de sinais acústicos que proporcionam bons débitos de comunicação [15]. Este desenvolvimento tem suscitado estudos e propostas para implementação de redes acústicas submarinas para diversas aplicações. Neste âmbito, as principais aplicações descritas na literatura são:

- **Controlo de dados oceanográficos:** Existe um interesse crescente por dados oceanográficos, quer por parte da comunidade científica, quer para fins comerciais. Tais dados oceanográficos podem auxiliar nas previsões climáticas e na antecipação de catástrofes. As redes acústicas submarinas podem neste âmbito proporcionar um meio adequado e eficiente para recolha dessas informações. Entre os dados de interesse para a oceanografia e passíveis de monitorização e registo podem citar-se: a temperatura, a salinidade, a condutividade, as correntes marítimas (superficiais, intermediárias e profundas), a pressão, dados sobre ondas e marés, penetração e distribuição da luz na água do mar, turbulência, concentração de produtos químicos, poluentes etc;

- **Assistência na Navegação:** Os sensores podem ser usados para identificar os perigos no fundo do mar, localizar rochas perigosas ou cardumes em águas rasas, destroços submersos, ou traçar um perfil batimétrico.
- **Vigilância submarina:** esta aplicação foi impulsionada durante a segunda guerra mundial, com o objectivo de detetar inimigos e proporcionar comunicação com submarinos. A utilização de redes acústicas submarinas para essa finalidade tem o potencial de incrementar a eficiência desses sistemas de defesa, ampliando sua área de cobertura e melhorando sua resolução. Uma rede autónoma para essa finalidade pode ser composta por nós fixos e móveis, o que proporciona flexibilidade e agilidade na detecção de intrusos;
- **Controlo de poluição:** o controlo de poluição ambiental tem-se tornado cada vez mais necessário e de interesse para os órgãos fiscalizadores. As redes acústicas submarinas têm flexibilidade e potencial para implementar um sistema de controlo em tempo real que efectivamente identifique fontes poluidoras com a agilidade necessária para minimizar os seus efeitos;
- **Antecipação de informações sobre catástrofes:** uma rede acústica submarina com sensores apropriados pode ser utilizada para antecipar informações sobre possíveis catástrofes originadas nos oceanos, tais como os tsunamis. Tal permite a criação de um sistema de segurança para alertar as populações sobre a geração desses eventos, com o objectivo de minimizar seus os efeitos.
- **Exploração submarina:** As redes acústicas submarinas podem ajudar a detetar campos petrolíferos submarinos ou reservatórios, determinar as rotas para colocação de cabos submarinos, e auxiliar na exploração de minérios valiosos.

No decorrer deste capítulo abordaremos ao longo de 6 seções as redes de acústica submarina. Na próxima seção (seção 2.1) apresentaremos as principais características das redes acústicas submarinas. A seção 2.2 apresentará as principais arquitecturas utilizadas nas redes acústicas submarinas. Nas seções 2.3 e 2.4 apresentam-se os conceitos da comunicação acústica, canal acústico e os principais fatores que influenciam as comunicações acústicas. Na seção 2.5 apresenta-se o simulador de rede. Esta seção (seção 2.5) descreverá o modelo acústico no simulador de NS-2 [16], assim como também apresentará uma pequena descrição do princípio de funcionamento do simulador WOSS [17]. O capítulo é encerrado na (seção 2.6) apresentando as respectivas conclusões do referido capítulo.



## 2.1. Principais características das redes acústicas submarinas

Como forma de descrever as redes acústicas submarinas é comum caracterizar-se essas redes com base em diversos critérios [4][5][6], sendo os mais comuns os seguintes:

- **Mobilidade:** uma rede acústica submarina pode ser composta exclusivamente por nós fixos, móveis, ou por mistura de ambos os tipos. Devido às limitações da comunicação submarina, os nós fixos são mais fáceis de implementar, mas a flexibilidade que os nós móveis proporcionam é um atrativo cada vez mais considerado na configuração de redes acústicas submarinas [5]. Um factor que tem contribuído para isso é o desenvolvimento dos AUVs (*Autonomous Underwater Vehicles*), que operam como nós móveis para recolha de informações, utilizando para tal sensores próprios [5];
- **Topologia:** podem ser centralizadas ou distribuídas. A centralizada é utilizada em redes com os nós fixos ao redor de um nó denominado de *Sink*, onde os nós da rede encaminham os seus dados e dele recebem comandos. Esse nó *Sink* necessita de um canal de comunicação com uma estação terrestre para encaminhar os dados da rede e dela receber comandos para configuração dos nós sensores [5].  
A topologia distribuída pode ser utilizada em redes acústicas submarinas com nós móveis, onde não há hierarquia. Os nós móveis podem navegar independentemente com auxílio de referências acústicas submersas [5];
- **Distribuição espacial:** pode ser bidimensional, quando todos os nós da rede estão fixados no leito oceânico, ou tridimensional, quando os nós da rede ficam submersos, flutuando em diferentes profundidades, presos ao fundo do mar por cabos com comprimentos variados;
- **Utilização do canal de comunicação:** depende da densidade dos nós e da extensão geográfica, a utilização do canal de comunicação pode ser comum ou em cluster. Na utilização comum o canal é partilhado por todos os nós da rede ao mesmo tempo, sendo necessário um protocolo de controlo de acesso ao meio para organizar a utilização do canal. Na forma de utilização em clusters a arquitetura prevê a utilização de diversos grupos (ditos “clusters”), dentro dos quais os nós se submetem a um mecanismo interno de acesso ao meio, conforme descrevem Doukkali *et al.* [11] e Ma *et al.* [12]. Essa “clusterização” tem como objetivo permitir o reaproveitamento do canal de comunicação em células onde o afastamento impede a interferência mútua de transmissões simultâneas.

## 2.2. Arquitecturas de comunicação em redes acústicas submarinas

A arquitetura da rede é um factor crucial na determinação do consumo de energia, da capacidade e da confiabilidade de uma rede. Assim, a arquitetura da rede deve ser projetada com garantia de cuidado e após a sua implementação e otimização a arquitetura deve ser sempre que possível reavaliada [5].

As missões de acompanhamento submarino são extremamente caras devido ao alto custo dos aparelhos submarinos. Por isso, é importante que a rede implementada seja altamente confiável, de modo a evitar o fracasso das missões de acompanhamento que derivam da falha de um ou vários dispositivos.

A capacidade da rede também é influenciada pela arquitetura da rede. Como a capacidade do canal submarino é bastante limitada, é muito importante organizar a arquitetura de rede de forma que não haja nenhuma falha de comunicação.

Em seguida analisaremos as principais arquiteturas utilizadas no contexto das redes acústicas submarinas.

### 2.2.1. Arquitectura bidimensional

A arquitetura bidimensional para redes acústicas submarinas é apresentada na figura 2.1 (adaptada a partir de [5]). Na figura pode observar-se que os nós sensores estão ancorados no fundo do oceano. Os nós sensores submarinos comunicam com um ou mais nós ditos *Sink*, por meio de ligações acústicas sem fios. Os nós *Sink*, como se pode observar na figura 2.1, são dispositivos da rede responsáveis pela retransmissão dos dados da rede do fundo do oceano para uma estação da superfície terrestre. Para retransmitir os dados da rede do fundo do oceano para estação da superfície terrestre, os nós *Sink* são equipados com dois transdutores acústicos, um transdutor vertical e um transdutor horizontal [5].

O transdutor horizontal é utilizado pelo nó *Sink* para comunicar com os nós sensores, com o objetivo de: enviar comandos e configurar os sensores (no sentido *Sink* para nós sensores); recolher dados monitorizados (no sentido sensores para nós *Sink*). O transdutor vertical é utilizado pelo nó *Sink* para transmitir dados para uma estação da superfície.

A estação da superfície é equipada com um transdutor acústico que é capaz de receber múltiplas comunicações paralelas a partir de diversos nós *Sink*. Também é dotado de uma ligação rádio

frequência (RF) de longo alcance e/ou transmissor via satélite para comunicar com um *Sink* em terra e/ou um *Sink* de superfície.

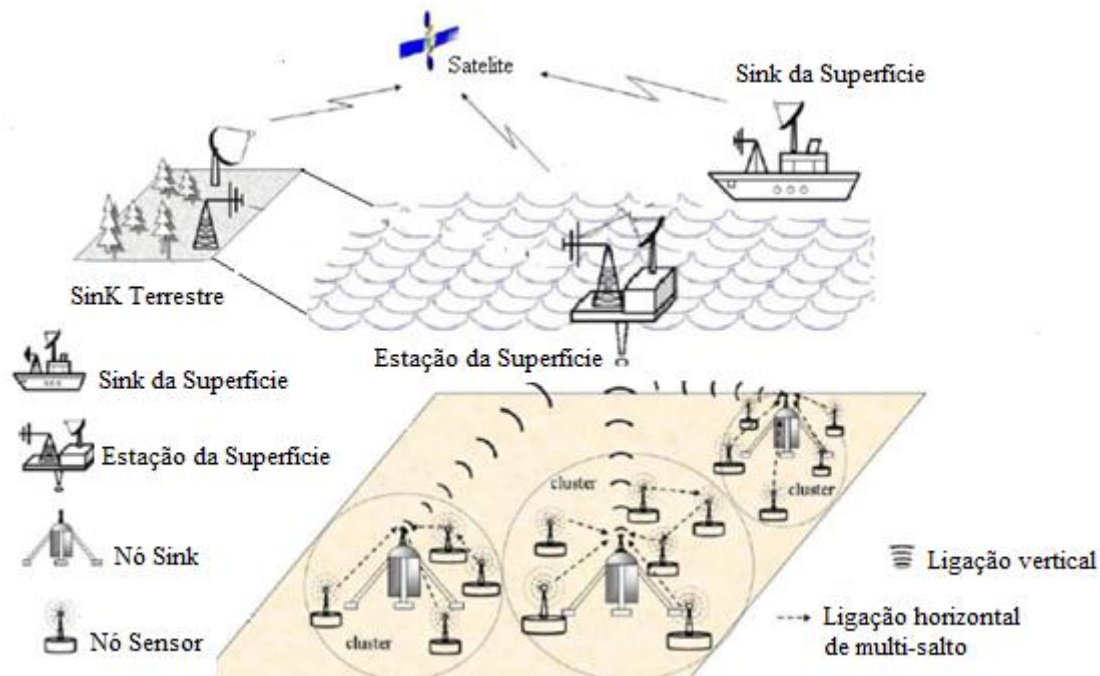


Figura 2.1: Arquitetura bidimensional das redes acústicas submarinas.

Os nós sensores podem comunicar com os nós *Sink* diretamente ou através de múltiplos-saltos. Nas ligações diretas, cada nó sensor envia diretamente os dados para o nó *Sink* selecionado [10]. Nas ligações de múltiplos saltos, à semelhança das redes de sensores terrestres [7], os dados produzidos por um nó sensor da fonte é retransmitida por nós intermédios até atingir o nó *Sink* do destino. Isso pode implicar um aumento do consumo de energia e aumento da capacidade da rede, mas também aumenta a complexidade em termos de encaminhamento.

### 2.2.2. Arquitectura tridimensional

A arquitetura tridimensional para redes acústicas submarinas é apresentada na figura 2.2 (adaptada de [5]). As arquiteturas tridimensionais são utilizadas para detetar e observar fenómenos que não podem ser observados adequadamente por sensores colocados no fundo do oceano. Estas arquiteturas permitem realizar uma amostragem tridimensional do ambiente oceânico[5].

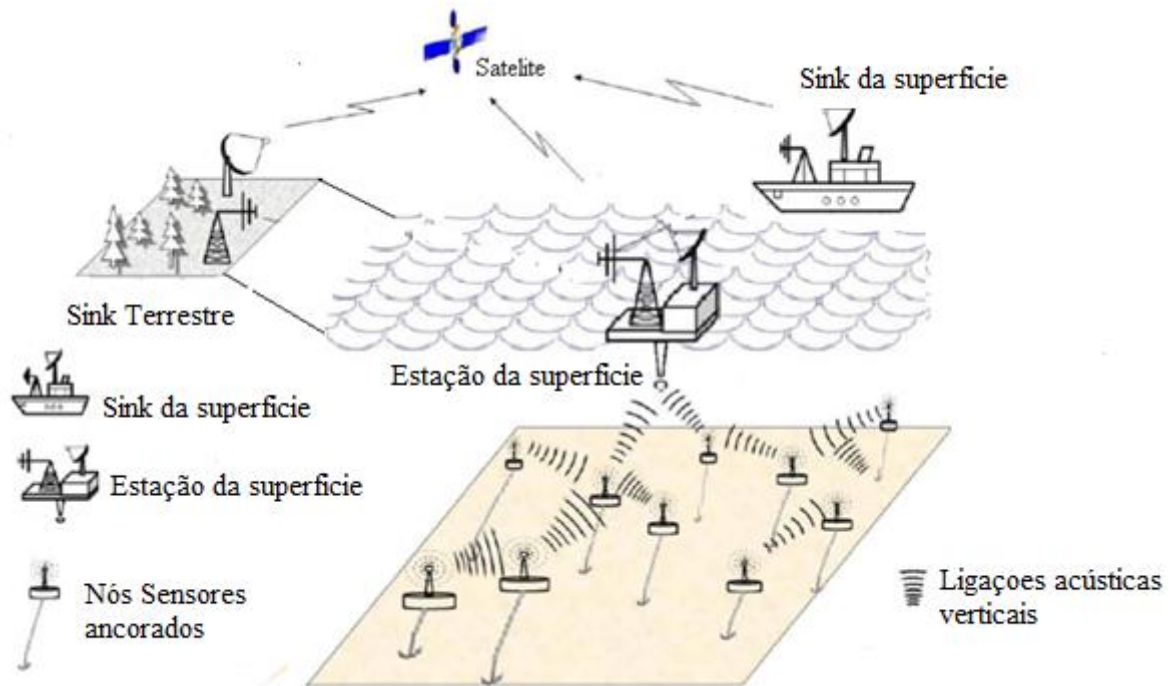


Figura 2.2: Arquitetura tridimensional das redes acústicas submarinas.

Nas redes acústicas submarinas tridimensionais, os nós sensores flutuam em diferentes profundidades, com o objetivo de observar um determinado fenómeno [5]. Uma solução possível seria conectar cada nó sensor a uma boia de superfície, por meio de cabos, onde o comprimento pode ser regulado de modo a ajustar a profundidade de cada nó sensor [8]. Embora esta solução permita uma implementação rápida e fácil da rede de sensores, diversas boias podem obstruir navios que naveguem à superfície, ou em caso de cenários militares, podem ser facilmente detetadas e desativadas por inimigos. Além disso, as boias flutuantes são vulneráveis às condições meteorológicas. Por estas razões, uma solução diferente consiste em ancorar dispositivos sensores no fundo do oceano.

Na arquitetura tridimensional, apresentada na figura 2.2, cada nó sensor é representado ancorado ao fundo do oceano e equipado com uma boia flutuante que pode ser enchida utilizando uma bomba, e que puxa o sensor em direção à superfície do oceano. A profundidade do nó sensor pode ser regulada, ajustando o comprimento do cabo que interliga o sensor à âncora, utilizando para tal um motor que reside no sensor e que é controlado eletronicamente [5].

Na arquitetura tridimensional das redes acústicas submarinas não se utiliza o nó *Sink* de profundidade tal como representado na figura 2.1. Os nós sensores são capazes de transmitir informações para a estação de superfície através de múltiplos saltos [5]. Assim, os dispositivos de rede devem coordenar as suas profundidades de tal forma que a arquitetura da rede esteja sempre

conectada, ou seja deve existir pelo menos um caminho que permita que todos os nós sensores consigam comunicar para a estação de superfície [5].

## 2.3. Comunicação submarina

Apesar do uso de cabos elétricos ou de fibras ópticas no leito oceânico ser uma operação viável, esses meios físicos para comunicação entre nós de uma rede submarina são restritos a aplicações muito específicas. A agressividade do ambiente acústico submarino dificulta a viabilidade para conexões eficientes e fiáveis.

A criação de redes submarinas com cabos pode ser útil apenas para certas aplicações onde não se espera mudança de cenário ao longo do tempo, ou como parte de uma infra-estrutura de apoio na implementação de redes mistas [9].

Em termos práticos, uma rede acústica submarina deve dispor de meios de comunicação sem fios entre os nós. A comunicação sem fios submarina proporciona mobilidade, dinamismo na composição da rede (inclusão e exclusão de nós) e contorna os inconvenientes das limitações das conexões submarinas (resistência a corrosão, resistência a pressões elevadas, etc).

### 2.3.1. Propagação de sinais no ambiente acústico submarino

A comunicação acústica submarina sem fios pode ser feita através de transmissões eletromagnéticas, óticas ou acústicas. No entanto, embora possíveis, as comunicações submarinas através de transmissões óticas ou eletromagnéticas apresentam várias limitações que as tornam inadequadas para implementação de redes acústicas submarinas. Lanbo *et al.* [4] organizaram uma boa descrição e comparação das comunicações submarinas empregando essas três tecnologias.

#### Transmissão eletromagnética

A principal desvantagem das transmissões eletromagnéticas no ambiente acústico submarino é sua atenuação devido à absorção do meio. Essa absorção fica condicionado ao uso de baixas frequências. A comunicação a baixas frequências requer grandes antenas, o que torna a comunicação eletromagnética inadequada para implementação de redes acústicas submarinas [4][5][6]. Além disso, mesmo em baixas frequências, a propagação a alcances úteis requer potências elevadas, o que também é um fator que limita a sua utilização, já que os nós dessas redes são alimentados por baterias.

### **Transmissão ótica**

A transmissão ótica na água apesar de proporcionar altas taxas de transmissão com um consumo de energia muito baixo, têm a desvantagem de propagar-se apenas a curtas distâncias, em consequência da absorção e do espalhamento. A luz é fortemente absorvida pela água e sofre grande espalhamento devido às partículas em suspensão, dificultando a comunicação e limitando a sua aplicação a distâncias muito pequenas. Dependendo da turbidez na água, essa distância pode ser tão pequena quanto alguns centímetros e, mesmo em água cristalina, o alcance prático para essas transmissões não excede alguns metros. O espalhamento da luz na água é maior quanto maior for sua turbidez, que no caso da água do mar pode variar de poucas dezenas a alguns milhares de NTUs (*Nephelometric Turbidity Unit*). Além disso, mesmo para comunicação a pequenas distâncias, é necessário garantir alinhamento estável entre transmissor e recetor durante a transmissão para garantir a eficiência da comunicação, o que no caso da dinâmica do ambiente acústico submarino é difícil de obter [5].

### **Transmissão acústica**

A transmissão acústica submarina apesar das suas limitações, é a mais versátil e a que é efetivamente utilizada para comunicação no ambiente submarino, devido principalmente às limitações das transmissões eletromagnéticas e óticas. No entanto, o canal acústico submarino tem características que restringem a sua aplicabilidade em redes acústicas submarinas, tais como: velocidade de propagação baixa e variável; banda estreita e dependente da distância entre transmissor e recetor; grande consumo de energia nas transmissões [18].

## **2.4. Canal acústico submarino**

Os principais fatores que influenciam as comunicações acústicas submarinas são: perdas de transmissão, trajetória de propagação, potência de transmissão, ruído, multi-percurso e atraso de propagação elevado e variável. Todos estes fatores determinam a variabilidade temporal e espacial do canal acústico submarino e fazem com que a largura de banda disponível do canal acústico submarino seja limitada e dependente tanto do alcance como da frequência. Nas comunicações acústicas submarinas as ligações podem ser classificadas de acordo com sua distância em: muito longo, longo, médio, curta e muito curta [19]. A tabela 2.1 apresenta larguras de banda típicas do canal acústico submarino para diferentes distâncias [19]. Nesta tabela pode-se verificar que quanto maior forem as ligações, isto é maior forem as distâncias nas comunicações acústicas menor será a

largura de banda. As ligações acústicas submarinas também são classificadas como vertical e horizontal, de acordo com a direção do raio de som em relação ao fundo do oceano.

Sistema	Distância [km]	Largura de Banda [kHz]
Muito longo	1000	<1
Longo	10-100	2-5
Médio	1-10	~10
Curto	0.1-1	20-50
Muito Curto	<0.1	>100

Tabela 2.1: Largura de banda disponível para as diferentes distâncias em canais submarinos

Em seguida são apresentados os principais fatores que influenciam as comunicações acústicas submarinas:

- **Perdas de transmissão:** as perdas de transmissão são calculadas considerando dois fatores: a atenuação e a geometria da propagação. A atenuação entre outros fatores é provocada pela absorção devido à conversão de energia acústica em calor. A atenuação aumenta com a distância e frequência. A atenuação também é causada pelo espalhamento e reflexão (na superfície e no fundo do oceano), refração e dispersão (devido ao deslocamento do ponto de reflexão causada pelo vento na superfície). A profundidade da água também desempenha um papel fundamental na determinação da atenuação [5].  
A geometria da propagação também é um fator importante no cálculo das perdas de transmissão. Existem dois tipos de geometria de propagação: esférica, que caracteriza a comunicação em águas profundas, e cilíndricas, o que caracteriza comunicações em águas rasas [5].
- **Ruído:** no canal acústico submarino existe dois tipos de ruído:
  - o ruído causado pelas máquinas (bombas), e pelas actividades de navegação, especialmente em áreas com tráfego de navios, e
  - o ruído ambiental que está relacionado com a hidrodinâmica (movimento das águas, incluindo as marés, correntes, tempestades, vento e chuva), e de fenómenos sísmicos e biológicos [5].
- **Multi-Percurso:** o multi-percurso pode ser responsável pela degradação severa do sinal na comunicação acústica submarina, uma vez que gera interferência intersimbólica (ISI) [5].

A geometria do multi-percurso depende da configuração da ligação. Os canais verticais são caracterizadas pelo pouco tempo de dispersão, enquanto que os canais horizontais têm propagação de multi-percurso extremamente longo.

- **Atraso elevado e variável:** a velocidade de propagação no canal acústico submarino é de cinco ordens de magnitude menor do que no canal de rádio terrestre. O atraso de propagação muito elevado (0,67 s/km) pode reduzir o rendimento do sistema consideravelmente [5]. Para além disso, a elevada variação do atraso é muito prejudicial na implementação dos protocolos de controlo de acesso ao meio, pois não permite uma estimativa precisa do tempo de ida e volta [5].

Os fatores descritos acima são causados pelas propriedades físico-químicas do meio da água, tais como temperatura, salinidade e pressão, e com as suas variações temporais no espaço [5]. Essas variações, em conjunto com a natureza da guia de onda do canal, fazem com que o canal acústico submarino seja altamente temporal e espacialmente variável.

### 2.4.1. Velocidade de propagação do som

A velocidade de propagação do som na água é de aproximadamente 1500 m/s. Isso é mais de 4 vezes a velocidade de propagação do som no ar, mas é cinco ordens de grandeza menor que a propagação eletromagnética no ar. Essa baixa velocidade provoca grande latência nas transmissões acústicas submarinas (aproximadamente 0.67 s/km, ou 1 segundo a cada 1.500 metros), que é um dos maiores problemas no desenvolvimento de redes acústicas submarinas, porque dificulta o controlo de acesso ao meio, localização e sincronismo. Para além de baixa, a velocidade de propagação do som na água é variável, dependendo da profundidade, da temperatura e da salinidade do meio. A fórmula da velocidade de propagação do som na água do mar descrita por Wilson em [11], e expressa em m/s, pode ser representada usando uma aproximação simplificada, com os termos mais significativos através da expressão (2.1).

$$c = 1449 + 4.6t - 0.055t^2 + 1.39(s - 35) + 0.017d \quad (2.1)$$

Em (2.1),  $c$  representa a velocidade em m/s,  $t$  a temperatura da água em graus célsius,  $d$  a profundidade em metros e  $s$  a salinidade, em partes por mil (ppm) (varia de 32 a 37, sendo 35 um valor típico). A equação (2.1) dá origem à curva conhecida como perfil de velocidade, que é uma curva que representa a variação da velocidade do som em função da profundidade.



A temperatura da água nos oceanos, a partir de uma pequena camada superficial, diminui com o aumento da profundidade, mas atinge um mínimo de aproximadamente 4 graus Celsius a uma profundidade que varia em função da latitude, permanecendo constante a partir desta profundidade. Desprezando o efeito da variação da salinidade, que é desprezível em mar aberto, chega-se ao perfil de velocidade, apresentado na figura 2.3(a), (adaptada a referência [14]). Nesta figura o eixo horizontal ( $c$ ) representa a velocidade do som e o eixo vertical ( $z$ ) a profundidade.

O ponto  $C_h$  é a velocidade do som na profundidade do leito oceânico e  $Z_c$  a profundidade na qual a velocidade se iguala à da superfície.

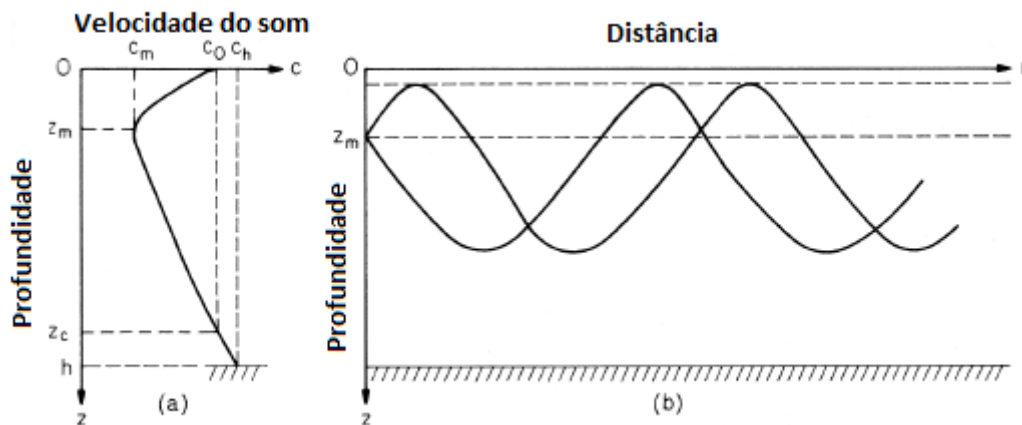


Figura 2.3: Variação da velocidade de propagação do som no mar (a) e canal SOFAR (*Sound Fixing And Ranging*) (b).

O ponto de inflexão dessa curva,  $Z_m$ , delimita duas regiões: acima, a termoclina principal e abaixo, a isoterma profunda, origem dos efeitos da refração que criam o canal SOFAR (*Sound Fixing And Ranging*), tal como apresentado na figura 2.3 (b) (adaptada a referência [14]). A termoclina é a faixa de profundidade no mar que apresenta temperatura decrescente, compreendida entre a camada superficial e a profundidade onde a temperatura atinge seu valor mínimo (4 graus Celsius) [14]. Nesta figura o eixo horizontal de (b) ( $r$ ) representa distância horizontal e o eixo vertical ( $z$ ) é a profundidade [14]. Este canal forma uma guia horizontal de ondas acústicas cujo eixo central se localiza na separação dessas duas regiões [14]. Esse efeito ocorre por causa da refração que as ondas acústicas sofrem nas camadas adjacentes, com velocidades diferentes [14].

## 2.4.2. Trajetórias de propagação

O fenômeno da refração ocorre quando as ondas atravessam interfaces entre meios que possuam diferentes velocidades de propagação. O que relaciona as velocidades nos dois meios é o índice de refração relativo  $n_{21}$ . O desvio que a trajetória do som na água sofre quando passa de um meio para outro depende das diferenças de velocidade entre esses meios. A lei de Snell relaciona os ângulos de incidência ( $\theta_i$ ) e de refração ( $\theta_r$ ) com os índices de refração dos meios e, conseqüentemente, com suas respectivas velocidades ( $V1$  e  $V2$ ) através da seguinte relação [14].

$$\frac{\sin(\theta_i)}{\sin(\theta_r)} = \frac{n_2}{n_1} = n_{21} = \frac{V1}{V2} \quad (2.2)$$

A figura 2.4 (a) mostra esse efeito para o som quando atravessa a interface entre dois meios com velocidades distintas e a figura 2.4 (b) mostra o efeito curvilíneo de uma onda sonora propagando-se de uma região de menor velocidade para uma de maior velocidade com variações graduais. O raio acústico tende a curvar-se na direção da região de menor velocidade.

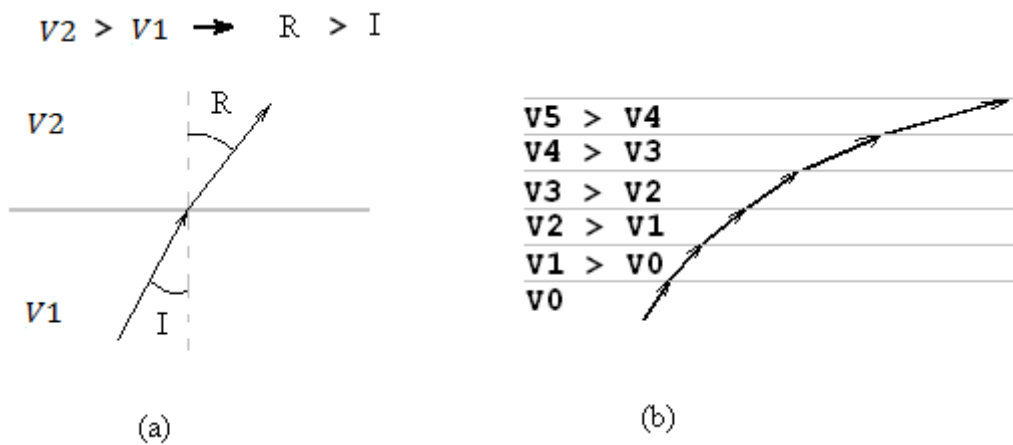


Figura 2.4: Refração do som entre dois meios com velocidades distintas (a) e em meio com variação gradual de velocidade (b).

Pode concluir-se que o som no mar pode propagar-se em trajetórias curvilíneas e, em alguns casos, nunca atingir determinadas regiões, conhecidas como zonas de sombra. Essas zonas são locais inalcançáveis por transmissões acústicas, como apresentado na figura 2.5, (adaptada de referência [18]). O mesmo fenômeno é responsável por formação de zonas de convergência, locais onde, devido às trajetórias cíclicas, há uma convergência de transmissões, porém provenientes de

trajetórias diferentes, o que as faz chegar ao ponto de convergências em tempos diferentes. No caso de transmissões digitais esse efeito causa o aumento de interferências intersímbolica.

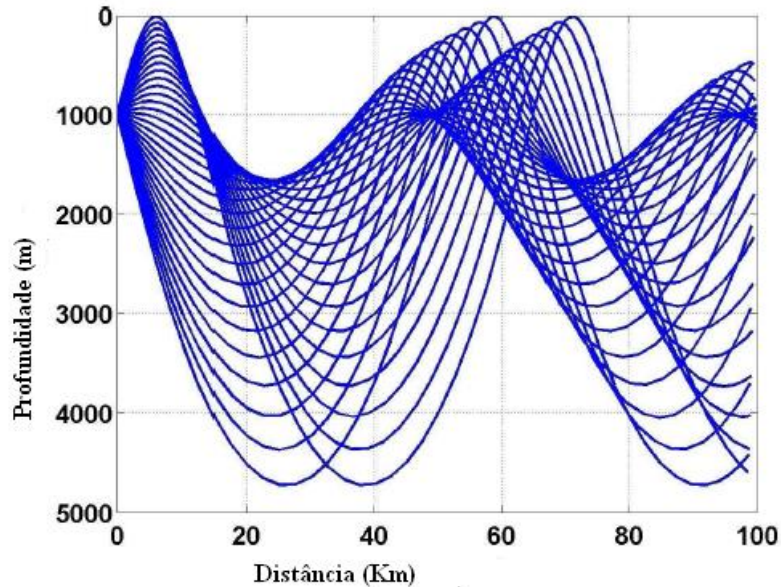


Figura 2.5: Zonas de sombra e de convergência.

Esses efeitos ocorrem principalmente em águas profundas, porque em águas rasas a velocidade não chega a sofrer a inversão apresentada na figura 2.3 (a). No entanto, em águas rasas, outro problema que ocorre na propagação de sinais acústicos é o da reflexão provocada pelas múltiplas reflexões do sinal transmitido, na superfície e no leito oceânico, como mostrado na figura 2.6. Esse efeito também provoca aumento da interferência intersímbolica, devido à chegada no recetor de sinais provenientes da mesma fonte, mas com trajetórias distintas, e atrasos distintos, o que é agravado pela baixa velocidade de propagação.

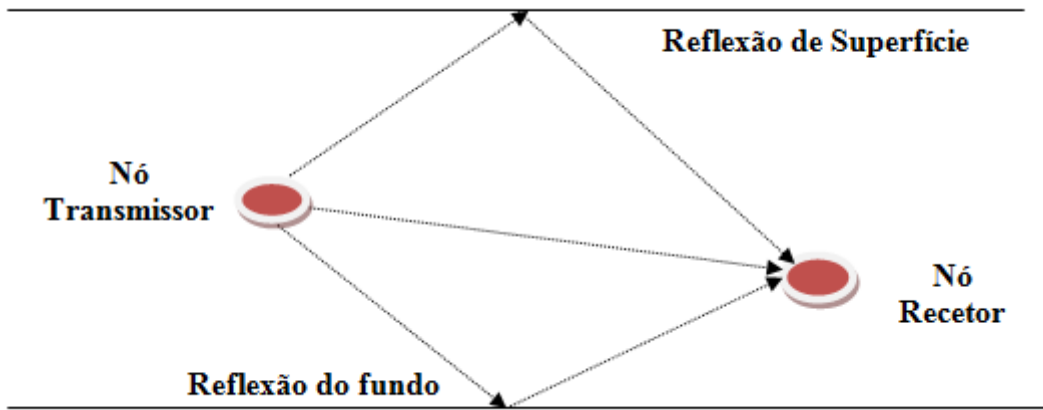


Figura 2.6: Reflexão do sinal acústico na superfície e no fundo.

### 2.4.3. Perdas de transmissão

As perdas de transmissão no canal acústico submarino dependem tanto da distância transmissor-receptor quanto da frequência do sinal transmitido, pela seguinte relação [14]:

$$A(l, f) = l^k a(f)^l \quad (2.3)$$

A expressão (2.3) pode ser reescrita da seguinte forma:

$$10 \log_{10} A(l, f) = k \times 10 \log_{10}(l) + l \times 10 \log_{10}(a(f)) \quad (2.4)$$

Em (2.4),  $A(l, f)$  representa a atenuação em função da distância  $l$  e da frequência  $f$ ,  $k$  é o factor de espalhamento e  $a(f)$  é o coeficiente de absorção. O factor  $k$  determina a geometria da propagação, quando para  $k = 1$  o espalhamento é cilíndrico e quando  $k = 2$  para espalhamento é esférico. Normalmente, em termos práticos, é adotado um valor de  $k = 1,5$  [14].

O coeficiente de absorção é expresso em dB/km através da fórmula empírica de Thorp e a frequência é dada em kHz para comunicações acústicas submarinas [14]:

$$10 \log a(f) = \begin{cases} 0.11 \frac{f^2}{1+f^2} + 44 \frac{f^2}{4100+f} + 2.75 \cdot 10^{-4} f^2 + 0.003, & f \geq 0.4 \\ 0.002 + 0.01 \left( \frac{f}{1+f} \right) + 0.011f, & f \leq 0.4 \end{cases} \quad (2.5)$$

As perdas por absorção são divididas em duas partes de frequências, uma frequência abaixo dos 400 Hz e uma outra acima dos 400 Hz.

O ruído no canal acústico submarino tem sido estudado e modelado em função de 4 componentes principais: turbulência, ruído da navegação, vento e ruído térmico. A densidade espectral de potência desses ruídos, em função da frequência (em kHz) é expressa nas seguintes fórmulas [13]:

$$\log N_t(f) = 1.7 - 3 \log f \quad (2.6)$$

$$\log N_s(f) = 4 + 2(s - 0.5) + 2.6 \log f - 6 \log(f + 0.03) \quad (2.7)$$

$$\log N_w(f) = 5 + 0.75w^{\frac{1}{2}} + 2 \log f - 4 \log(f + 0.4) \quad (2.8)$$

$$\log N_{th}(f) = -1.5 + 2 \log f \quad (2.9)$$

Nestas expressões,  $N_t$  representa o ruído por turbulências,  $N_s$  o ruído de navegações,  $N_w$  o ruído dos ventos,  $N_{th}$  o ruído térmico,  $s$  varia de 0 a 1, e  $w$  é a velocidade do vento em m/s. O ruído total,  $N(f)$  é a soma desses quatro componentes [13]:

$$N(f) = N_t(f) + N_s(f) + N_w(f) + N_{th}(f) \quad (2.10)$$

A combinação dos efeitos de atenuação, provocados pelo espalhamento e pela absorção, apresentada pela expressão (2.4), com o ruído total apresentado pela expressão (2.10), determina o gráfico da Relação Sinal Ruído (*SNR*) em relação a frequência, para cada distância emissor/recetor, como mostra a figura 2.7, (adaptada de [19]).

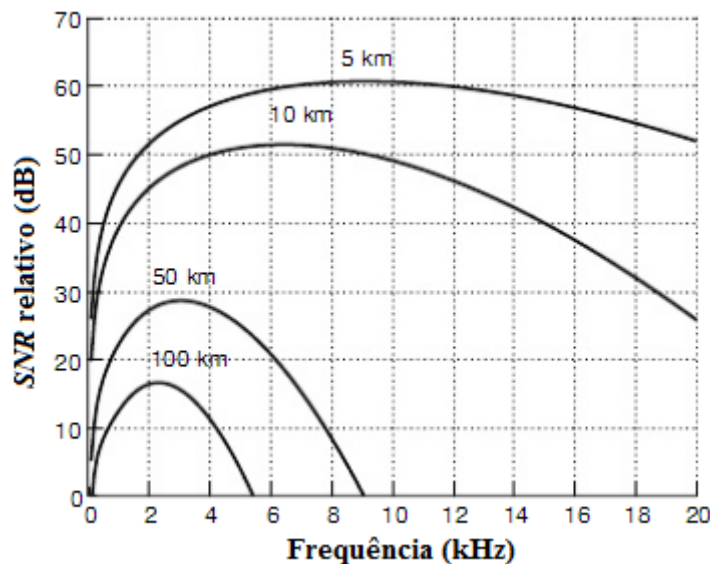


Figura 2.7: Relação Sinal Ruído em função da frequência.

A partir deste gráfico pode concluir-se que a densidade espectral de potência do ruído não depende apenas da distância, mas também das frequências usadas na comunicação. A melhor frequência para

transmissões acústicas submarinas, a que apresenta maior Relação Sinal Ruído (*SNR*) depende da distância emissor/recetor, sendo tanto menor quanto maior for essa distância. Portanto, o ruído total afecta a comunicação entre os nós quando a distância variar.

A banda disponível para as transmissões também varia com essa distância, sendo maior quando menor for as distâncias entre emissor/recetor. Portanto a banda de transmissão disponível no canal acústico, além de estreita em relação quando comparada com a banda do canal eletromagnético terrestre, é variável com a distância emissor/recetor, uma característica importante do canal acústico submarino que precisa ser tido em consideração na implementação das redes de acústica submarina.

## 2.5. Simulador de rede acústica submarina

A maior parte dos trabalhos publicados na área de redes acústicas submarinas apresentam resultados analíticos havendo poucos com resultados experimentais. Os que apresentam resultados de simulação, nem todos informam os detalhes da ferramenta e a configuração de simulação usada. Os que descrevem um pouco melhor a simulação, fazem-no com dados obtidos em simuladores desenvolvidos pelos próprios pesquisadores, ou em adaptações no NS-2, sem muitos detalhes sobre o modelo da comunicação acústica submarina (de [23] a [34]).

Nas redes acústicas submarinas ainda não há um consenso sobre um modelo padrão de simulação. A principal dificuldade consiste na definição do modelo da comunicação acústica submarina, que tem características absolutamente distintas do canal eletromagnético terrestre. Essa lacuna dificulta o desenvolvimento de protocolos por não permitir uma avaliação comparativa fiável da performance, especialmente para os protocolos de acesso ao meio e de rede, por não haver ainda uma referência de simulação consolidada e amplamente aceite.

Uma iniciativa no sentido de fornecer módulos para simulação de redes acústicas submarinas no NS-2 tem sido desenvolvida pelo departamento de engenharia eléctrica da Universidade de Pádova, Itália [22]. Essa iniciativa baseia-se na definição do modelo do canal acústico submarino descrito por Harris e Zorzi, na qual os autores dividem as funções e características do canal acústico submarino e da camada física em 4 componentes do NS-2: propagação, canal, meio físico e modulação [22].

O simulador WOSS é outra proposta mais recente, apresentada por Guerra *et al.* [17], onde os autores descrevem simulação baseada no NS-2 e no NS2-*MIRACLE*, utilizando o modelo de propagação do *software Bellhop ray tracing* [35]. Esta proposta agrega conhecimentos de perfis de velocidade do som, dados de batimetria e informações sobre os sedimentos do leito oceânico,

obtidos a partir de base de dados livres na *internet*, para proporcionar ao modelo da comunicação acústica submarina maior fiabilidade na tradução das condições ambientais [35].

### **2.5.1. Modelo físico acústico submarino no simulador NS-2**

No simulador de rede NS-2 o modelo físico acústico submarino está dividido em 4 componentes ou modelos como é apresentado na figura 2.8 (adaptada de referência [22]): o modelo de propagação, o modelo do canal, o modelo da camada física, e o modelo de modulação. Na figura A.1 (ver anexo A) são apresentadas de forma mais detalhada as características de cada um dos componentes do modelo físico acústico submarino do simulador de rede NS-2.

O modelo de propagação calcula a Relação Sinal Ruído no recetor e a abrangência da interferência das transmissões com base nas características da atenuação do sinal acústico e no ruído ambiental. A atenuação total é obtida com base nas perdas por espalhamento e nas perdas por absorção, segundo a aproximação de Thorp descrito na seção 2.4.3 pela expressão (2.5) [14].

O modelo de propagação no simulador de rede considera 4 factores tradicionais de ruído: turbulências, navegações, vento e ruído térmico [13], permitindo atribuições de pesos diferentes a cada uma dessas fontes para adequarem o modelo aos diversos cenários possíveis do ambiente submarino.

Como a atenuação do sinal acústico submarino depende da frequência, o modelo calcula a frequência cuja atenuação é menor, em função da distância transmissor-recetor, para poder calcular a Relação Sinal Ruído (*SNR*) no recetor. Calcula também a banda disponível para as distâncias transmissor-recetor, identificando as frequências onde a atenuação é 3 dB maior que a mínima.

O modelo do canal calcula os atrasos de propagação, com base nas características de velocidade variável do canal acústico submarino. Leva em consideração profundidades, temperaturas e salinidade do meio.

O modelo físico calcula estatísticas de recepção de pacotes, tempos de transmissão do pacote, erro do pacote e a capacidade em bit/s. O modelo requer informações sobre potência de transmissão, consumo de energia e definição da potência limiar de recepção. Esses valores variam de equipamento para equipamento e os valores padrão do modelo são estabelecidos com base no modem acústico WHOI [20].

O modelo de modulação fornece o tipo de modulação, sistema de codificação, nível de ruído, sendo também responsável pelo cálculo da taxa de transmissão binária (*R*), taxa de erro do bit (*BER*) e a taxa de erro de pacote (*PER*), obtidos através da relação sinal ruído.

Na próxima subseção apresentaremos as funções de transferência de cada um dos modelos descritos em cima.

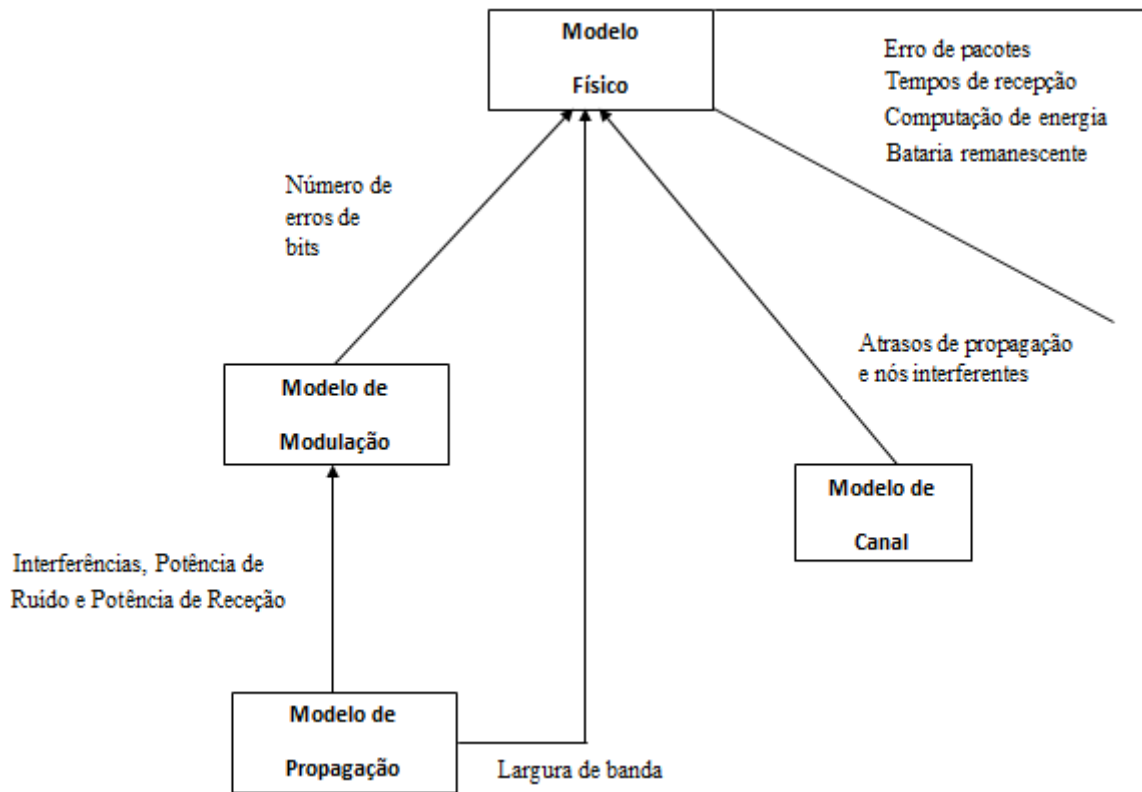


Figura 2.8: Modelo do canal e da camada física acústica submarina no NS-2.

### 2.5.1.1. Função de transferência do modelo de propagação

A figura 2.9 apresenta a função de transferência do modelo de propagação.

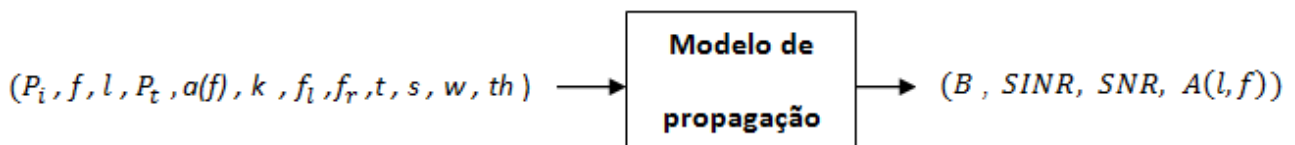


Figura 2.9: Função de transferência do modelo de propagação.

O modelo de propagação no simulador de rede NS-2 recebe os seguintes parâmetros de entrada: potência de interferência ( $P_i$ ), frequência ( $f$  em kHz), a distância ( $l$ ), o coeficiente de atenuação  $a(f)$ , os coeficientes de propagação ( $k$ ) para determinar a geometria da propagação tal como descrito na equação em [14]. Também no modelo de propagação temos que ter em consideração as faixas ou bandas de frequências superiores e inferiores ( $f_l, f_r$ ) que são utilizadas no cálculo da largura de banda, a potência de transmissão  $P_t$  e alguns fatores como a temperatura ( $th$ ),



o vento ( $w$ ), a turbulência ( $t$ ), e a navegação ( $s$ ) para determinar a densidade espectral de potência total do ruído.

Assim de acordo com os parâmetros de entrada referidos, o modelo de propagação devolve os seguintes parâmetros de saída: atenuação total  $A(l, f)$ , a Relação Sinal Ruído no recetor ( $SNR$ ), a Interferência da Relação Sinal Ruído ( $SINR$ ) e a largura de banda ( $B$ ) em Hz.

A atenuação total é obtida a partir da relação [14] descrita na secção 2.4.3.

No modelo propagação o cálculo da Relação Sinal Ruído ( $SNR$ ) no recetor é feito tendo em consideração a atenuação total  $A(l, f)$ , a potência da transmissão  $P_t$ , e a densidade espectral de potência total do ruído  $N(f)$ . O cálculo da Relação Sinal Ruído no recetor é dado pela seguinte expressão [22]:

$$SNR = \frac{P_t/A(l,f)}{N(f)} \quad (2.11)$$

No modelo de propagação para além do cálculo da Relação Sinal Ruído no recetor ( $SNR$ ), também se tem que ter em conta a Interferência da Relação Sinal Ruído ( $SINR$ ) que pode ser calculada pela seguinte expressão [22]:

$$SINR = \frac{P_t/A(l,f)}{N(f)+P_i} \quad (2.12)$$

A potência de interferência,  $P_i$ , é dada por [35]:

$$P_i = \sum_{n \neq 0} |g^2(n)| \quad (2.13)$$

Esta expressão traduz o somatório das respostas impulsivas do sinal dado pela interferência intersimbólica. Em que  $g$  representa a resposta impulsiva.

No simulador de rede o valor da potência de interferência ( $P_i$ ) é calculada usando a função *MInterference*. Essa função implementa um modelo de interferência responsável por verificar algum tipo de interferência causado na transmissão do pacote. Esse modelo de interferência pode ser substituído por outro modelo utilizando os seguintes métodos: *addToInterference()* e *getInterferencePower()* [35].

### 2.5.1.2. Função de transferência do modelo do canal

A figura 2.10 apresenta a função de transferência do modelo do canal no simulador de rede NS-2. O modelo do canal é responsável por calcular os atrasos de propagação, e determinar as colisões entre nós vizinhos.



Figura 2.10: Função de transferência do modelo canal.

O modelo do canal recebe os seguintes parâmetros de entrada: a temperatura ( $t$ ) em graus celsius ( $^{\circ}\text{C}$ ), a salinidade ( $S$ ) na água em ppm (partes por mil), a profundidade ( $Z$ ) em metros, as coordenadas ( $T_x, T_y, T_z$ ) do transmissor, e as coordenadas ( $R_x, R_y, R_z$ ) do recetor em metros.

O modelo do canal como parâmetro de saída fornece o cálculo do atraso de propagação ( $Pdelay$ ). O cálculo do atraso de propagação é dado pela seguinte expressão [35]:

$$Pdelay = \frac{Totaldist}{c} \quad (2.14)$$

Onde  $Totaldist$  representa a distância total, e  $c$  a velocidade de propagação do som em acústica submarina (em m/s).

A distância total ( $Totaldist$ ) pode ser calculada em função das coordenadas  $T_x, T_y, T_z$  do transmissor e coordenadas,  $R_x, R_y, R_z$  do recetor através da seguinte expressão [35]:

$$Totaldist = \sqrt{((R_x - T_x)^2 + (R_y - T_y)^2 + (R_z - T_z)^2)} \quad (2.15)$$

A velocidade de propagação do som na água ( $c$ ) depende da temperatura na água que por sua vez depende na profundidade, em consequência de alguns efeitos de não linearidade. Para além da temperatura e profundidade também devemos considerar o valor da salinidade na água (em ppm) para fazer o cálculo da velocidade de propagação. Normalmente o valor padrão para a salinidade nos oceanos do mundo é (35 ppm), mas também pode tomar outros valores. Por exemplo, no mar báltico o valor da salinidade é de (9 ppm).

No modelo do canal acústico submarino do simulador de rede NS-2, o valor da velocidade de propagação do som na água é calculada utilizando a expressão [35].

$$c = 1448.5 + 45.7t - 5.2t^2 + 0.23t^2 + (1.333 - 0.126t + 0.009t^2)(s - 35) + 16.3z + 0.18z^2 \quad (2.16)$$

Onde ( $t$ ) é a temperatura, ( $z$ ) é a profundidade e ( $s$ ) é a salinidade na água.

### 2.5.1.3. Função de transferência do modelo de modulação

A figura 2.11 apresenta a função de transferência do modelo de modulação. No simulador de rede NS-2 o modelo de modulação apresenta características sobre o tipo de modulação, sistema de codificação e o nível do ruído para comunicações acústicas submarinas. O tipo de sistema de modulação usado no simulador de rede NS-2 é o BPSK (*Binary Phase Shift Keying*), modulação coerente onde nas transições de um bit 0 para um bit 1, ou de um bit 1 para um bit 0, a portadora sofre uma alteração de fase.

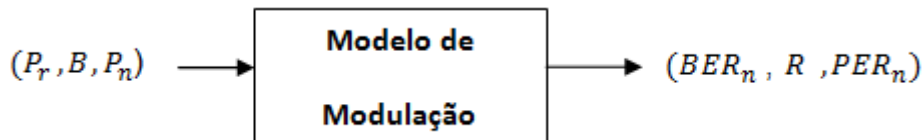


Figura 2.11: Função de transferência do modelo de modulação.

No modelo de modulação obtêm-se o débito binário ( $R$ ), taxa de erros de bit ( $BER_n$ ) e a taxa de erro do pacote ( $PER_n$ ), a partir da relação sinal ruído. Para efectuar os cálculos dos valores dos parâmetros citados, o modelo de modulação recebe os seguintes parâmetros de entrada: potência do sinal recebido ( $P_r$ ), potência do ruído ( $P_n$ ), e a largura de banda ( $B$ ) dada pelo sistema de modulação.

No sistema de modulação desenvolvido no simulador de rede NS-2 não se define diretamente a taxa de transmissão, mas sim a banda utilizada na modulação do sinal transmitido. Internamente a banda é convertida em taxa de transmissão na proporção de 1bps para cada 2Hz disponível na banda do canal acústico. O débito binário pode ser calculado usando a seguinte expressão [35]:

$$R = \frac{1}{T} = \frac{B}{2} \quad (2.17)$$

Em (2.17),  $T$  representa a duração do símbolo. A taxa de erro de bit  $BER_n$  é por sua vez calculada pela seguinte expressão [35]:

$$BER_n = \frac{1}{2} \operatorname{erfc} \left( \sqrt{\frac{P_r}{P_n}} \right) \quad (2.18)$$

Em (2.18),  $erfc(x)$  é uma função complementar do erro. Essa função complementar do erro é obtida pela seguinte expressão [35]:

$$erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^{+\infty} e^{-\gamma^2} d\gamma \quad (2.19)$$

A taxa de erro de bit também pode ser utilizada para determinar a taxa de erro do pacote pela seguinte expressão [35]:

$$PER_n = 1 - (1 - BER_n)^{size \times 8.0} \quad (2.20)$$

A expressão (2.20) traduz a probabilidade de um pacote com um tamanho de  $size$  bytes conter um bit errado.

### 2.5.1.4. Função de transferência do modelo físico

A figura 2.12 apresenta a função de transferência do modelo físico. Tal como já referido no simulador de rede NS-2 supõe-se que é utilizado um hardware específico para comunicações acústicas submarinas designado por modelo micromodem WHOI [20]. Neste modelo são definidos alguns parâmetros de entrada como a potência máxima de transmissão ( $P_t$ ), potência limiar recebido ( $P_r$ ), e alguns parâmetros de interface de consumo de energia.

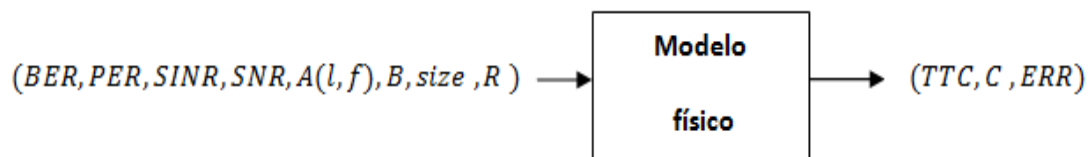


Figura 2.12: Função de transferência do modelo físico.

O modelo físico recebe os seguintes parâmetros de entrada: taxa de erros de bit ( $BER$ ), taxa de erros do pacote ( $PER$ ), relação sinal ruído interferente ( $SINR$ ), relação sinal ruído no recetor ( $SNR$ ), atenuação total ( $A(l, f)$ ), largura de banda, ( $B$ ), tamanho do pacote em bytes ( $size$ ) e o débito binário ( $R$ ).

Como parâmetros de saída o modelo físico fornece: tempo de transmissão do pacote ( $TTP$ ), erro do pacote ( $ERR$ ), e a capacidade do canal em bit/s ( $C$ ).

O tempo de transmissão do pacote em bit/s é calculado utilizando a seguinte expressão [35]:

$$TTP = \frac{8 \times size}{R} \quad (2.21)$$

Onde  $size$  é o tamanho do pacote em bytes e  $R$  representa o débito binário (em bit/s).

O débito máximo teórico permitido pelo canal (em bit/s) é obtido a partir do teorema de Shannon-Hartley para canais reais:

$$C = B \times \log_2(1 + SNR) \quad (2.22)$$

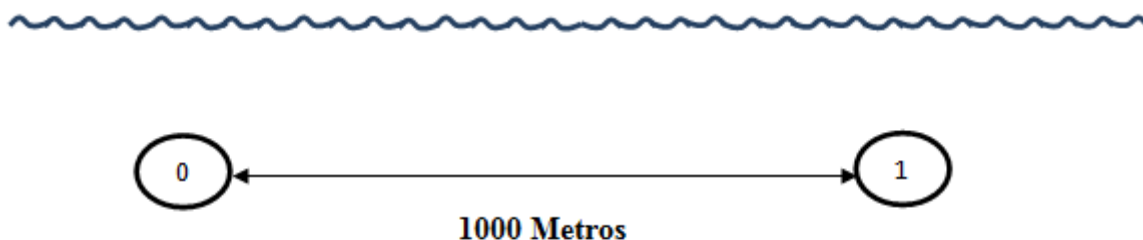
Onde  $B$  é a largura de banda do canal, é definido em 3 dB para faixas ou bandas de frequências superiores e inferiores ( $f_r$ ,  $f_l$ ) em Hz e  $SNR$  representa a Relação Sinal Ruído no recetor (representado como um fração de potências em unidades lineares, i.e: não em decibéis (dB)).

O modelo físico para além de fornecer os cálculos dos tempos de transmissões dos pacotes e as capacidades em bit/s fornece também o cálculo do erro do pacote e o atraso total.

### 2.5.1.5. Análise do modelo de propagação do NS-2

Por forma a avaliar o modelo do canal acústico submarino implementado no simulador NS-2, foram realizadas várias simulações, comparando os resultados dos testes com os resultados do artigo dos autores Harris e Zorzi [22]. Para tal, foram avaliadas as principais características do modelo, tais como: o atraso de propagação e potência de receção em função da profundidade e da distância entre os nós.

O resultado do atraso de propagação em relação a profundidade da água foi obtido fazendo varias simulações uma vez que este resultado depende da profundidade a que se efetua a comunicação. Assim foram realizados diversos testes, variando a profundidade e as distâncias entre nós e utilizando um cenário simples, em que ambos os nós, emissor e recetor, se encontravam à mesma profundidade, tal como mostra a figura 2.13.



---

Figura 2.13: Cenário da simulação para avaliar o modelo físico no NS-2.

A figura 2.14 apresenta o gráfico do atraso de propagação para uma comunicação entre dois nós a uma distância de 1 km, variando-se a profundidade conjunta dos dois nós, desde os 0 m até aos 4500 m. Desse gráfico pode concluir-se que o atraso de propagação vai diminuindo com aumento da profundidade desde os 1000 m até 4500 m de profundidade devido ao aumento da velocidade de propagação do som para esses valores de profundidade quando a distância entre os dois nós é de 1 km.

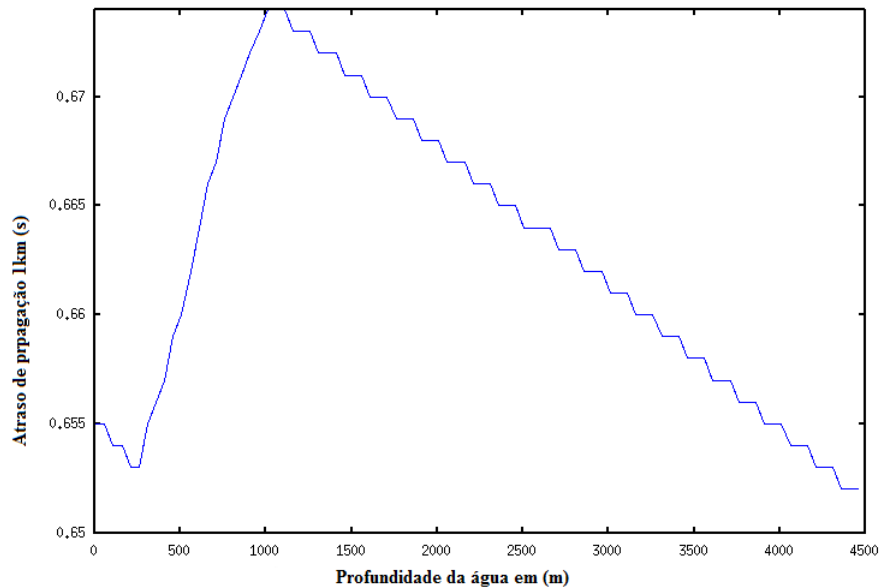


Figura 2.14: Efeito da profundidade em relação ao atraso de propagação.

Para o mesmo cenário da figura 2.13 colocaram-se em seguida ambos os nós a 10 m de profundidade, fazendo variar distância entre nós, dos 2000 m iniciais até aos 0 m, através da deslocação do nó 1 a uma velocidade de 2 m/s no sentido do nó 0. A duração do tempo da simulação foi de 1000 segundos.

A figura 2.15 apresenta o gráfico com o resultado do atraso de propagação em relação a distância entre os nós. Os resultados confirmam o esperado, ou seja pode concluir-se que para a mesma profundidade o atraso de propagação aumenta de forma linear com a distância entre os nós.

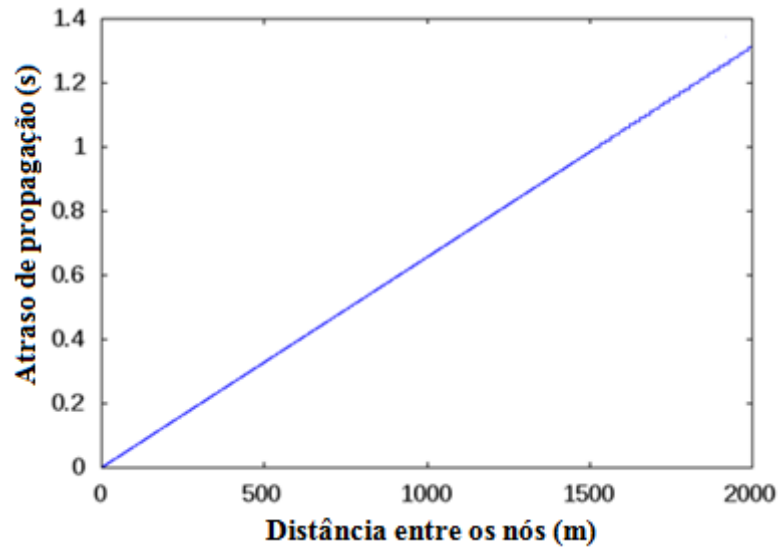


Figura 2.15: Atraso de propagação em relação a distância entre os nós.

Em seguida para o mesmo cenário e utilizando os mesmos dados da simulação anterior, foi obtido o gráfico que relaciona a potência de recepção em (dB re  $\mu\text{Pa}$ ) com a distância entre os nós. Desse gráfico, representado na figura 2.16, pode concluir-se que a potência de recepção decresce de forma acentuada nos primeiros 500 m de distância, descendo de forma menos acentuadamente para distâncias superiores a 500 m.

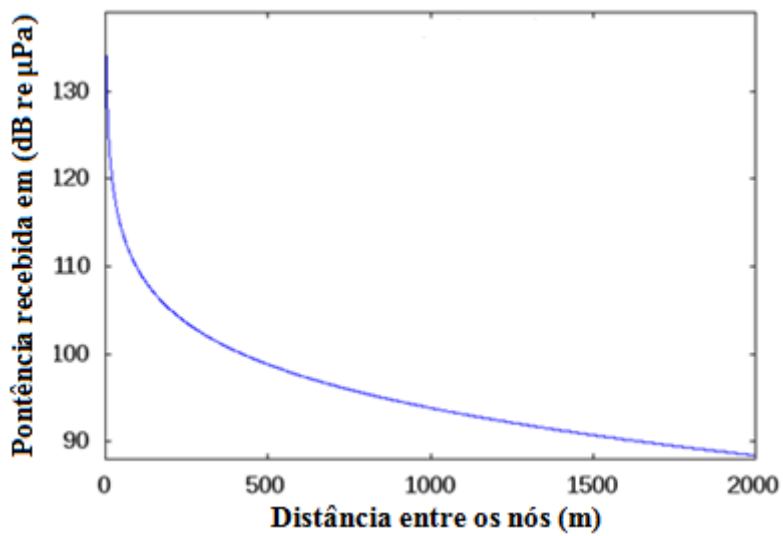


Figura 2.16: Potência de recepção em relação a distância entre os nós.

Em cada caso, os resultados simulados corresponderam exatamente ao modelo apresentado pelos autores Harris e Zorzi [22]. No entanto, dos resultados obtidos e da análise do modelo físico acústico submarino implementado por Harris e Zorzi no simulador de rede NS-2 através de fórmulas empíricas verificou-se a necessidade de um modelo acústico submarino mais completo [22].

## 2.5.2. Simulador de rede WOSS

O simulador de rede acústico submarino WOSS foi proposto por Frederico Guerra, é baseado no NS-2 [16] e nas extensões do NS-2-MIRACLE [35]. O simulador WOSS com integração de novas extensões do NS-2-MIRACLE fornece novas e melhores funcionalidades incluindo comunicações baseadas em pacotes, controle de acesso ao meio, encaminhamento, protocolos de camada superior, mobilidade, modelos de tráfego, e as métricas de desempenho. Esses recursos são integrados com uma reprodução mais realista dos efeitos de propagação acústica por meio do programa *Bellhop ray tracing* [36]. Isso substitui e melhora o modelo acústico submarino descrito anteriormente, incluindo os modelos de propagação.

Para calcular a solução das equações de propagação entre um transmissor e um recetor, *Bellhop* exige o conhecimento do perfil da velocidade do som SSP (*Sound speed profile*), o perfil batimétrico e o tipo de sedimentos de fundo oceânico necessário para modelar as perdas de potência acústica devido às reflexões de fundo, obtidos através de bases de dados gratuitamente disponíveis na *internet*.

O SSP é calculado, utilizando a base de dados mundial dos oceanos [37], uma coleção de SSPs medidos durante uma série de experiências em todo o mundo. As medições são divididas pela localização e dia ou estação do ano, em que a medição foi realizada.

Como complemento, são também utilizados dados espacialmente mais finos disponíveis em GLINT'08 [38], a partir de testes no mar (não foram utilizados na presente dissertação).

Os dados batimétricos são fornecidos através do GEBCO (*General Bathymetric Chart of the Oceans*) [39], uma base de dados de amostras publicamente disponível da profundidade do fundo do mar.

O tipo de sedimentos de fundo oceânico é retirado da base de dados do *National Geophysical Data Center Deck41* [40].

Para tornar a interface da base de dados mais fácil, o simulador WOSS especifica uma tecnologia de base de dados (SQLite, netcdf) através de um programa de interface simples, com isto, o utilizador



somente escreve um objeto que executa a interface e o simulador WOSS faz as ligações adequadas para as bases de dados.

A união de todos os esses componentes necessários para executar o *Bellhop* permite que o utilizador somente tenha que especificar em que lugar ou zona as experiências simuladas ocorrem. Isto é feito definindo a latitude e longitude, assim como o tamanho da área de rede e o simulador automaticamente trata do resto.

No simulador de rede WOSS todo o processo de simulação do modelo físico acústico submarino torna-se transparente para o utilizador, de modo que é possível concentrar-se apenas no projeto de implantação de rede e protocolos, e não nas questões de propagação.

Em seguida apresentaremos e descreveremos os principais blocos do simulador WOSS, exemplificando o seu funcionamento.

### **2.5.2.1. Diagrama de blocos do simulador WOSS**

O diagrama de blocos do simulador acústico submarino WOSS é apresentado na figura 2.17.

Este diagrama representa a API (*Application Programming Interface*) do simulador, detalhando o conjunto de rotinas e normas estabelecidas para usar os seus serviços, sem se envolver em detalhes da implementação do *software*.

O *WossManager* API tem como tarefa de responder, controlar, manipular e criar os objetos do simulador WOSS. Ele oferece também a capacidade de planejar uma estratégia para variação do tempo e / ou simulações multifrequência do canal, usar batimetria, sedimentos ou dados do perfil da velocidade do som para qualquer par transmissor-recetor.

O *WossCreator* API fornece uma interface para criação e inicialização dos objetos do WOSS, aliviando o utilizador dessa tarefa, fornecendo-lhe uma instância da base de dados pronta a utilizar.

O *Woss* API tem como função integrar um simulador de canal, fornecendo para tal interfaces para:

- Configuração do ambiente da base de dados ou de dados personalizados através da API *DbManager*.

- Leitura dos resultados do simulador do canal através da API *ResReader*.

A definição da batimetria, o perfil da velocidade do som (SSP), a base de dados dos sedimentos, o cálculo da pressão e o cálculo do atraso de potência do canal são acedidas através da base de dados ( *WossBathymetryDb*, *WossSedimentDb*, *WossSSPDb*, *WossResPressDb* e *WossResTimeArrDb*) fornecidas pela API *DbManager*.

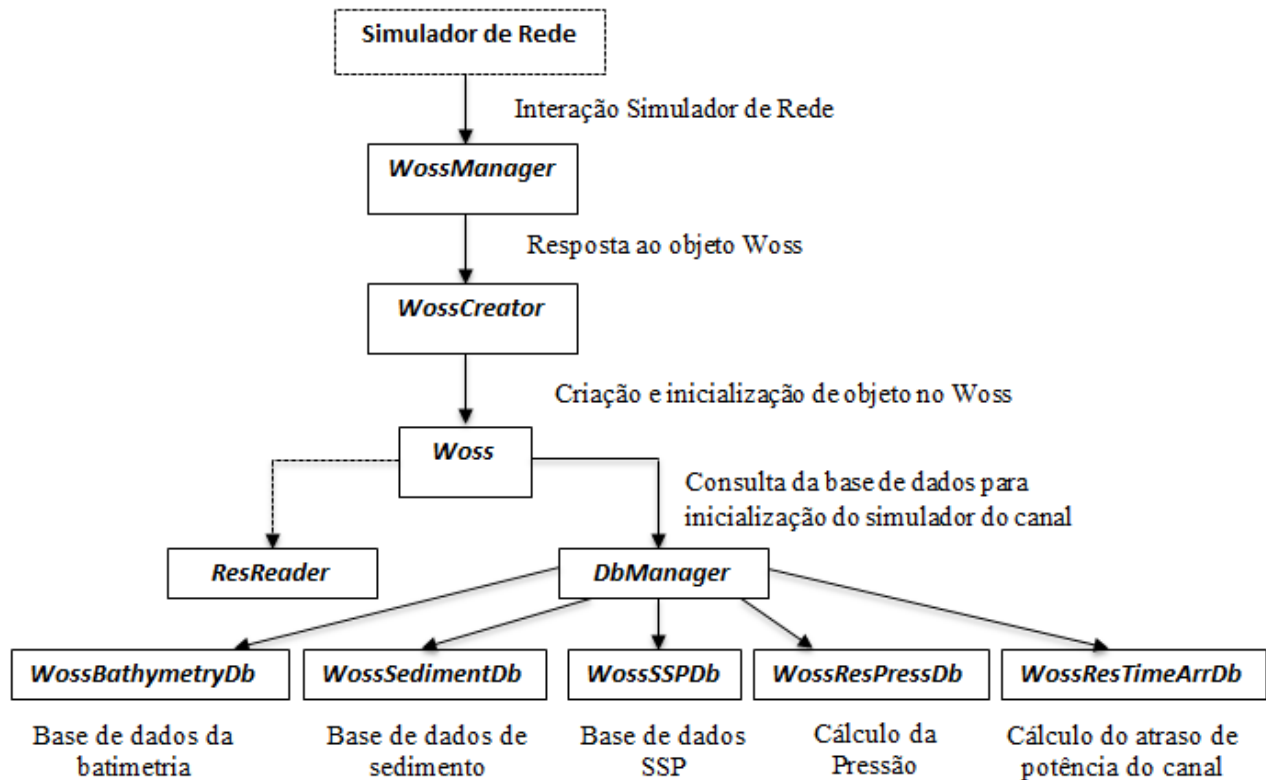


Figura 2.17: Diagrama de blocos do simulador de rede WOSS.

## 2.5.2.2. Definição das classes no simulador WOSS

No simulador WOSS são definidos um conjunto de classes. Das principais classes existentes podemos destacar os seguintes:

- A classe das coordenadas geográficas definida pelas classes (*Woss :: Coord*) e (*Woss :: Coord Z*): esta classe oferece informações sobre a distância e as coordenadas horizontais, verticais e profundidades assim como também apresenta outras fórmulas matemáticas como a raiz quadrada;
- Data e tempo definidas pela classe (*Woss :: Time*): esta classe define aritméticas sobre data e hora;
- Tempo de referência da simulação definida pela classe (*Woss :: TimeReference*): esta classe importa o tempo de referência da simulação externa ao do simulador de rede;
- Geração de números aleatórios definida pela classe (*Woss:: randomGenerator*): esta classe faz importação de um gerador de números aleatório personalizado a partir de outro simulador de rede externo;

- Perfil da velocidade do som definida pela classe (*Woss :: SSP*): esta classe faz cálculos sobre velocidade de som e manipulações das equações de propagações do som retiradas através da literatura científica;
- Sedimento da superfície definida pela classe (*Woss:: Sediment*): esta classe faz definições e manipulações dos parâmetros dos sedimentos geoacústicos;
- Perfil do atraso de potência do canal definida pela classe (*Woss :: TimeArr*): esta classe faz o cálculo do atraso.
- Pressão definida pela classe (*Woss :: Pressure*): esta classe faz operações aritméticas e conversões sobre pressão;
- Transdutor definida pela classe (*Woss :: Transducer*): esta classe faz cálculos precisos sobre o consumo de energia e capacidades do feixe dos transdutores reais;
- Classe manipulador de todas as classes definidas acima através da classe (*Woss :: DefHandler*): esta classe serve para "conetar" em tempo de execução qualquer classe que herda as classes descritas acima.

A versão actual do simulador WOSS para além das classes definidas acima também tem capacidade de oferecer:

- Implementação de uma interface para o programa *Bellhop ray tracing* [36], através das classes (*Woss :: BellhopWoss*) e (*Woss :: BellhopCreator*);
- Implementação de uma interface da base de dados *NetCDF* do perfil das velocidades dos sons médias mensais retiradas das base de dados do WOA (*World Ocean Atlas*) [37] através das classes (*Woss :: SspWoa2005Db*) e (*Woss :: SspWoa2005DbCreator*);
- Implementação de uma interface para a base de dados da batimetria da GEBCO [39], através das classes (*Woss :: BathyGebcoDb*) e (*Woss :: BathyGebcoDbCreator*);
- Implementação de uma interface personalizada da *NetCDF* para análise dos dados extraídos das bases de dados DECK41, para a composição dos sedimentos de fundo [40], através das classes (*Woss :: SedimDeck41Db*) e (*Woss :: SedimDeck41DbCreator*);
- Uma base de dados dos transdutores acústicos reais feitos a partir de arquivos de dados disponíveis publicamente na *Internet*.

### 2.5.2.3. Base de dados de batimetria do GEBCO

O GEBCO consiste num grupo de especialistas internacionais que trabalham para o desenvolvimento de um conjunto de dados batimétricos globais para os oceanos, incluindo um

conjunto de dados batimétricos em grelha, o GEBCO Altas Digital, o GEBCO da mapa do mundo e o GEBCO com dados geográficos para recursos submarinos [39].

No simulador WOSS os dados batimétricos da GEBCO são apresentados em forma de grelha, com uma representação computacional dos dados em 3D tridimensional ( $x$ ,  $y$ ,  $z$ ). O valor de  $z$  é a profundidade do fundo do mar e é dado por incrementos uniformemente espaçados de longitude ( $x$ ) e latitude ( $y$ ). O simulador WOSS disponibiliza as seguintes grelhas para dados batimétricos: a grelha GEBCO\_08 *Grid* e a grelha GEBCO *one Minute*. Essas duas grelhas estão disponíveis para *download* através página *web* da BODC (*British Oceanographic Data Centre*) centro de dados oceanográficos britânicos. Utilizando esta aplicação *web*, o utilizador pode seleccionar os ficheiros completos da grelha que pretende extrair ou definir uma área através de um mapa interativo, digitando numa caixa de diálogo as coordenadas geográficas da zona ou área que pretende simular.

A grelha GEBCO\_08 é apresentada na figura 2.18. Esta grelha consiste numa grelha mundial de 30 arco-segundo gerada na sua grande parte pela combinação das análises das profundidades dos navios controlados pela interpolação dos pontos de sons orientados por dados derivados do satélite. A grelha GEBCO\_08 é acompanhada pela grade SID “ indentificador da fonte” que também esta disponível para *download*. Este identificador mostra como é que as células da grelha serão limitadas por dados batimétricos durante o processo. A grelha GEBCO *One Minute*, grelha global de um arco-minuto de resolução baseia-se na sua grande parte em contornos batimétricos contidos no GEBCO Digital Atlas. Os seus dados estão disponíveis para *download*, visualização e acesso a partir dos ficheiros das grades Mundiais em ASCII, assim como também através da *NetCDF* mostrando a forma do fundo do mar.

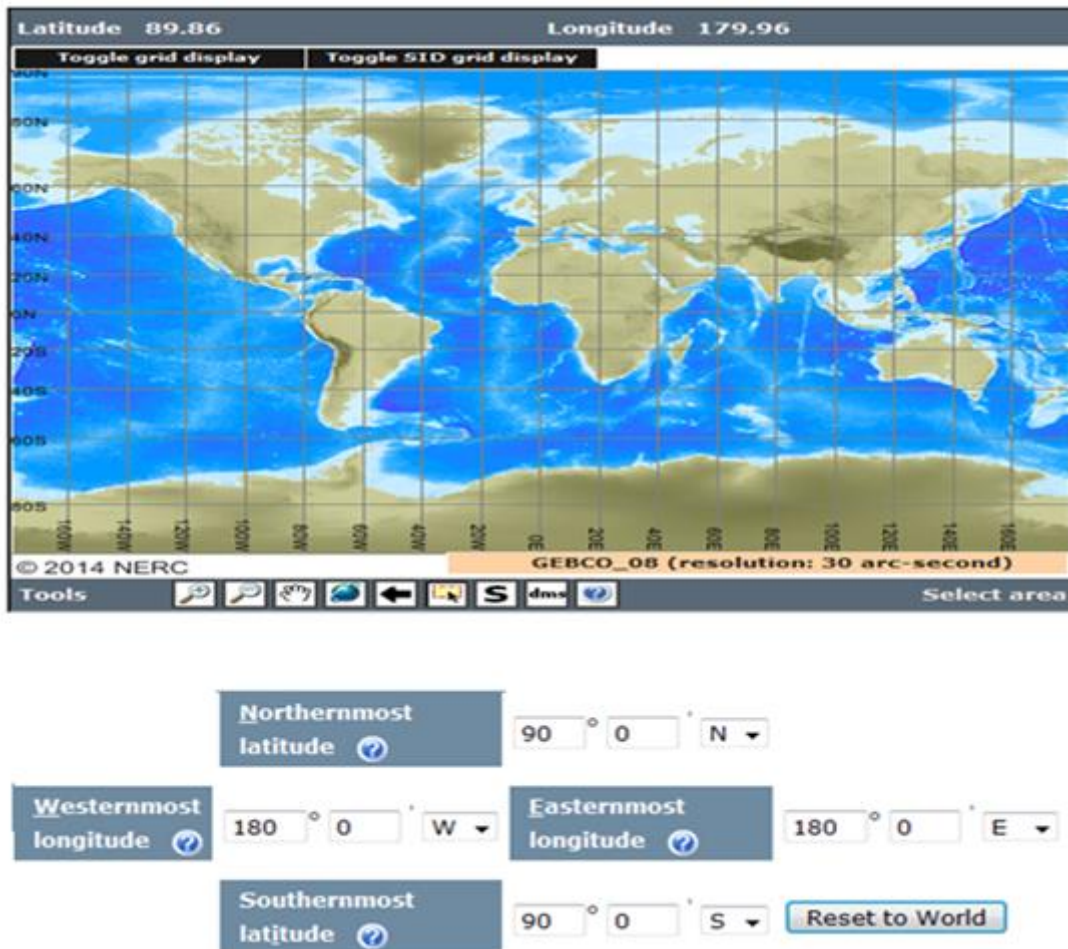


Figura 2.18: Mapa do perfil batimétrico utilizando a GEBCO\_08 *grid*.

### 2.5.2.4. Base de dados do perfil da velocidade propagação do som

No simulador WOSS o cálculo do perfil da velocidade propagação do som (SSP) é feito utilizando os dados retirados da base de dados da *World Ocean Atlas* para implementação de uma base de dados a partir do ficheiro da *NetCDF* através da classe (*Woss :: SSP*). Esses dados contêm médias mensais, sazonais e anuais dos perfis de velocidade do som. Os diferentes perfis conduzem a efeitos de propagação potencialmente muito diferentes, incluindo os canais de superfície do som, canais de sons profundo, zonas de convergência e zonas de sombra [41].

A classe (*Woss :: SSP*) oferece múltiplas capacidades para criação e manipulação dos perfis de velocidade do som. A figura 2.19 apresenta o diagrama utilizado pela classe (*Woss :: SSP*) no simulador WOSS para cálculo do perfil da velocidade de propagação do som.

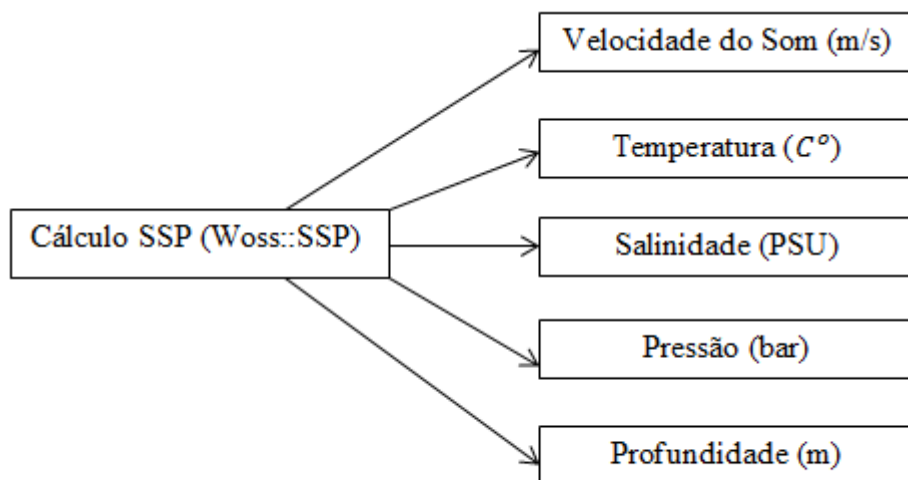


Figura 2.19: Diagrama para o cálculo do SSP.

Como pode se observar neste diagrama a classe (*Woss :: SSP*) armazena as seguintes informações relacionadas com um perfil de velocidade de propagação do som: temperatura (em  $C^{\circ}$ ), a pressão (em bar), salinidade (em PSU), profundidade (em metros) e velocidade do som (em m/s). Ela também oferece recursos para cálculos aritméticos computacionais, cálculos de velocidade de som utilizando equações de (Chen e Milerio) com atualizações de (Wong e Zu), conversão da profundidade para pressão (e vice-versa) com correções nas coordenadas, transformações do perfil da velocidade de propagação e perturbações aleatórias.

### 2.5.2.5. Base de dados de sedimentos DECK41

Os dados sobre os sedimentos do fundo oceânico são retirados da base de dados do *National Geophysical Data Center Deck41* [40]. O tipo de sedimento do fundo é necessário para caracterizar as reflexões de fundo. A base de dados DECK41 contém mais de 36.000 descrições dos sedimentos superficiais de amostras do fundo do mar em todo o mundo. Esses dados incluem fonte de recolha navio, cruzeiro, id da amostra, latitude, longitude, a data da recolha, profundidade da água, dispositivo de amostragem, litologia dominante, litologia secundário e descrição da superfície do local.

Os índices da litologia dominante e secundária para fins geoacústicos são fornecidos pelo simulador WOSS através da base de dados NetCDF para o sedimento do fundo. Estes valores são representados por um número inteiro sem sinal entre 1 e 11, segundo:

1- grãos de areia, 2- areia, 3- limos, 4- argila, 5- lama, 6- barros, 7- pedras, 8- material orgânico (granada, turfa, coral e madeira), 9- nódulos, concreções (fosfato, ferro e glauconite), 10- fundo, 11- nenhum valor. Esses valores podem ser consultados utilizando a classe (*woss:: sedimDeck41Db*). Esta classe fornece uma interface para criação e manipulação dos parâmetros geocústicos superficiais.

A classe (*woss:: sedimDeck41Db*) para além dos parâmetros dos sedimentos geocústicos superficiais referidos acima, fornece os seguintes parâmetros para simulação: nome do sedimento em ficheiro texto, velocidade de comprimento de onda em (m/s), *shear wave velocity* em (m/s), densidade do sedimento em (g/cm<sup>3</sup> ou definido pelo utilizador), atenuação do comprimento de onda em (dB/comprimento de onda ou definida pelo utilizador), *shear wave attenuation* em (dB/comprimento de onda ou definido pelo utilizador) e profundidade do fundo em metros.

No Capítulo 5, apresentaremos os resultados das simulações utilizando o simulador WOSS e os respetivos parâmetros físicos.

## 2.6. Conclusões do capítulo

Neste capítulo foram apresentados os conceitos base e revisão bibliográfica sobre acústicas submarina, com foco nas características de propagação acústica do canal, já que, embora outras soluções de comunicação sejam possíveis (tais como a eletromagnética e a ótica) é a transmissão acústica a que é efetivamente empregue.

O capítulo descreveu as classificações das redes acústicas submarinas, principais características, aplicações e arquitetura para comunicações acústicas submarinas.

Da propagação acústica submarina foram apresentados os principais fatores que influenciam as comunicações acústicas submarinas como: perdas de transmissão; ruído; para além de atraso de propagação elevado, de multi-percurso e variável. Para além das características de propagação acústica, também foram enfatizadas as características do canal acústico como: velocidade de propagação do som, trajetórias de propagação, reflexões do fundo e potências de transmissão.

Foi abordado também estudo do modelo da comunicação acústica submarina no simulador de rede NS-2, assim como também foi apresentado um simulador de rede mais recente designado por WOSS baseado no NS-2 e nas extensões do NS-2-*MIRACLE* com modelo de propagação do software *Bellhop ray tracing*.

No Capítulo 3 abordaremos os principais protocolos de acesso ao meio MAC, descrevendo as suas principais características.





## Capítulo 3

# Mecanismos de controlo de acesso ao meio em redes acústicas submarinas

Neste capítulo pretende-se descrever os principais protocolos de controlo de acesso ao meio MAC para redes sem fios terrestres assim como apresentar algumas propostas desses protocolos para ambientes acústicas submarinas, analisando os seus mecanismos de funcionamento que permitem efetuar o acesso ao canal e melhorar o desempenho da rede através do controlo do número de colisões. Também apresentaremos os protocolos de encaminhamento para redes acústicas submarinas. O capítulo termina apresentando os principais desenvolvimentos das redes acústicas submarinas.

### 3.1. Introdução

Numa rede acústica submarina com múltiplos terminais, o controlo de acesso ao meio é um dos pontos mais importantes e complexos devido ao elevado atraso de propagação das comunicações. A não utilização de um mecanismo adequado implica a existência de colisões entre transmissões que por conseguinte obrigam a sucessivas retransmissões dos mesmos dados.

Uma das diferenças existentes entre as transmissões eletromagnéticas terrestres e as transmissões acústicas submarinas reside na possibilidade ou não de troca de bateria ou a sua recarga. Em terra, quer a troca de bateria quer a sua carga através de energia fotovoltaica, são actividades possíveis e até triviais. No entanto, no caso das redes acústicas submarinas essas tarefas são muito difíceis e dispendiosas. A conversão fotovoltaica tem uma eficiência modesta, mesmo à superfície. A luz solar penetra pouco no mar, tornando essa alternativa desprezável como fonte de energia submarina.

Por outro lado, trocar uma bateria de um nó da rede acústica submarina, mesmo a pequenas profundidades, é uma operação com elevado custo e operacionalmente complicado que pode inviabilizar o projeto das redes acústicas submarinas.

Para além disso, nas redes acústicas submarinas as colisões no controlo de acesso ao meio, não só são muito mais frequentes do que nas redes terrestres em consequência dos atrasos elevados, mas têm ainda um efeito de reduzir o tempo de carga das baterias, dificultando assim a sua operação. Devido às diferenças de energia e latência entre transmissões acústicas submarinas e eletromagnéticas terrestres, as soluções encontradas para minimizar as colisões nas comunicações eletromagnéticas terrestres não podem, sem alterações, ser aproveitadas nas comunicações acústicas submarinas. Assim, têm-se vindo a desenvolver estudos e projetos nesta área, considerando protocolos de controlo de acesso ao meio especialmente definidos para as redes acústicas submarinas (de [42] a [50]).

### **3.2. Métodos de acesso múltiplo ao meio**

Nesta seção, apresentaremos três dos principais métodos de acessos múltiplos em ambientes acústicos submarinos: Acesso Múltiplo por Divisão de Frequência (FDMA), Acesso Múltiplo por Divisão de Tempo (TDMA), e Acesso Múltiplo por Divisão de Código (CDMA).

O método FDMA (*Frequency Division Multiple Access*) é o mais simples de todos os métodos de acesso múltiplo e baseia-se no seguinte: divide a banda de frequências disponível em sub-bandas e atribui cada sub-banda a utilizadores individuais. O canal é utilizado apenas por esse utilizador até que ele seja libertado. A largura de banda dos canais FDMA é menor que a largura de banda do canal de transmissão. Devido ao desvanecimento do sinal presente nos canais acústicos submarinos, que afecta de forma diferenciada diferentes frequências, os sistemas baseados em FDMA são dificilmente utilizados nestes ambientes [10]. Nas experiências do *SeaWeb* descritas por RICE *et al.* [51] foi utilizado esse método em consequência das limitações dos modems da época. No entanto, devido às limitações de banda disponíveis no canal acústico, este método não foi muito eficiente para comunicação nas redes acústicas submarinas.

O método TDMA (*Time Division Multiple Access*) em vez de dividir a banda de frequências, divide o tempo disponível para comunicação em intervalos de tempo (*time slots*), permitindo a transmissão sequencial dos nós num esquema de multiplexagem no tempo. Durante cada *time slot* o canal acústico apenas fica reservado para transmissão de dados de um dos nós da rede. Para garantir ausência de colisões há necessidade de uma margem de segurança TGI (*Time Guard Interval*) entre

os *time slots* sucessivos. Essa margem de segurança é proporcional ao atraso de propagação existente no canal. Para melhorar o funcionamento deste método de acesso múltiplo, há necessidade de um mínimo de sincronismo entre os nós da rede [52]. A elevada latência presente na comunicação acústica submarina implica a necessidade de margens de segurança muito grandes, o que limita a eficiência da comunicação utilizando esse método de acesso múltiplo nas redes acústicas submarinas. Além disso, a forma de garantir um mínimo de sincronismo entre os nós submarinos ainda é uma questão em investigação. Desse modo, o método TDMA só é seguro se a aplicação tolerar longos *Time Guard Intervals*, compatíveis com a precisão do sincronismo disponível.

O método CDMA (*Code Division Multiple Access*) permite uma transmissão simultânea de vários utilizadores na mesma gama de frequências. Os sinais são distribuídos através de códigos de pseudo-ruído que são utilizados para espalhar, por todo o espectro disponível, as mensagens de cada utilizador. Este método é resistente ao desvanecimento seletivo de frequências que ocorre nas redes acústicas submarinas, devido à propagação por múltiplos percursos. As técnicas de espalhamento no espectro podem ser: o FHSS (*Frequency Hopping Spread Spectrum*), usando FSK (*Frequency Shift Keying*) para baixas taxas de transmissão e o DSSS (*Direct Sequence Spread Spectrum*), usando PSK (*Phase Shift Keying*) para altas taxas de transmissão [53], [54].

O canal acústico submarino tem recursos muito limitados. Estes recursos escassos devem ser compartilhados de uma forma justa e eficiente por meio de protocolos de controlo de acesso ao meio simples e eficientes. Apesar dos métodos de acesso múltiplo ao meio aqui descritos serem bastante utilizados em redes sem fios, em redes de acústica submarina os principais esforços têm-se centrado em mecanismos não determinísticos de controlo de acesso ao meio.

Em seguida iremos analisar os protocolos de acesso ao meio utilizados em redes sem fios, para posteriormente descrever alguns protocolos propostos para meios submarinos.

### **3.3. Mecanismo de controlo de acesso ao meio para redes sem fios**

Em Sozer *et al.* [10] faz-se uma revisão dos principais protocolos de controlo de acesso ao meio para redes sem fios, resumidamente descritos a seguir.

O protocolo de acesso ao meio mais simples é o ALOHA, cujo funcionamento é baseado na transmissão de um pacote assim que os dados estiverem disponíveis para envio. Em caso de sucesso na transmissão o emissor receberá um ACK (*Acknowledgment*) emitido pelo nó de destino

confirmando a receção. Caso contrário, o emissor tentará retransmitir o pacote. Devido à independência e aleatoriedade das transmissões verificam-se muitas colisões, obtendo-se uma eficiência máxima na utilização do canal de aproximadamente 18% [55]. Para melhorar o desempenho do ALOHA foi introduzido o conceito de *slots* temporais, diminuindo assim a probabilidade de haver colisões a meio dos pacotes, que por sua vez podem danificar vários pacotes de uma só vez. Este método denominado de *slotted ALOHA*, permitiu o aumento da eficiência do ALOHA para 36 % [56].

Nos protocolos ALOHA, os utilizadores não tomam em consideração o estado do canal, ou seja antes de transmitir não escutam para ver se alguma transmissão está em curso resultando por isso numa elevada taxa de colisões.

O protocolo de acesso ao meio CSMA (*Carrier Sense Multiple Access*) foi uma forma encontrada para minimizar as colisões. Consiste em escutar o meio antes de transmitir, e durante a mesma. Em redes sem fios IEEE 802.11 o CSMA é complementado com técnicas que previnem as colisões (ditas de *Collision Avoidance*) através de um elaborado sistema de tempos entre tramas. Mais detalhes e várias versões do método CSMA podem ser encontradas em [57][58][59] e são apresentadas no decorrer deste capítulo.

Diversos protocolos MAC para redes sem fios são derivados do MACA (*Multiple Access with Collision Avoidance*). Esse protocolo, foi proposto por Karn em [60] e usa dois pacotes de sinalização, o RTS (*Request to Send*) e o CTS (*Clear to Send*). Quando um nó A deseja enviar uma mensagem ao nó B, inicia o protocolo com o envio de um RTS ao nó B, que ao recebê-lo responde com um CTS. Ao receber o CTS o nó A inicia então a transmissão dos dados.

O protocolo MACAW (*MACA-Wireless*), constitui uma melhoria do MACA. Foi proposto por Bharghavan *et al.* [61] com a transmissão de uma trama de ACK (*Acknowledgment*) após cada receção bem-sucedida, o que possibilita à origem retransmitir pacotes que não tenham sido corretamente recebidos pelo destino.

Em seguida analisaremos com mais detalhe cada um destes mecanismos.

### **3.3.1. Protocolo ALOHA**

O protocolo de acesso ao meio ALOHA foi desenvolvido por N. Abramson em 1970 na universidade de Havai [62]. O funcionamento do protocolo ALOHA é simples, ou seja, sempre que

um utilizador tem algum pacote de dados para enviar, esse pacote é enviado imediatamente [10]. Se a transmissão do pacote de dados for bem-sucedido, isto é, se nenhum pacote for recebido com erro, o recetor envia uma confirmação ao emissor [10]. Se nenhum ACK for recebido num intervalo de tempo definido entre zero e  $2 \times Delay + ackTime$ , onde *Delay* representa o atraso de propagação entre o nó emissor e recetor e *ackTime* é o tempo de transmissão de um ACK, o emissor considera que ocorreu a perda do pacote na rede, possivelmente devido a uma colisão no canal. Nesse caso o pacote será enviado novamente depois de um tempo aleatório, para evitar que haja uma nova colisão no caso de algum outro nó tentar fazer o mesmo [10].

Assim, sempre que dois terminais tentam aceder ao canal ao mesmo tempo [17], dá-se uma colisão e é necessário haver um mecanismo de *backoff* [17] que evite que volte a ocorrer. O mecanismo de *backoff* funciona da seguinte forma: depois de enviar um pacote de dados, o emissor espera durante período de tempo (entre zero e o dobro do atraso máximo de propagação). No caso de não chegar uma confirmação da correta receção do pacote, cada terminal escolhe aleatoriamente e uniformemente o tempo de *backoff*, num intervalo que varia entre  $[0, 2^{txRetry} \times (2 \times maxDelay + dataTime)]$ , em que *dataTime* representa o tempo de transmissão do pacote, *maxDelay* o atraso máximo de propagação de um pacote na rede, e *txRetry* é o número de retransmissões do pacote. Assim, por cada transmissão falhada o valor máximo do intervalo vai duplicando, até um máximo de 5 tentativas falhadas.

A figura 3.1 mostra um exemplo simples do funcionamento do protocolo ALOHA, em que três terminais tentam enviar as suas tramas para um terminal recetor, originando uma colisão entre os terminais 2 e 3.

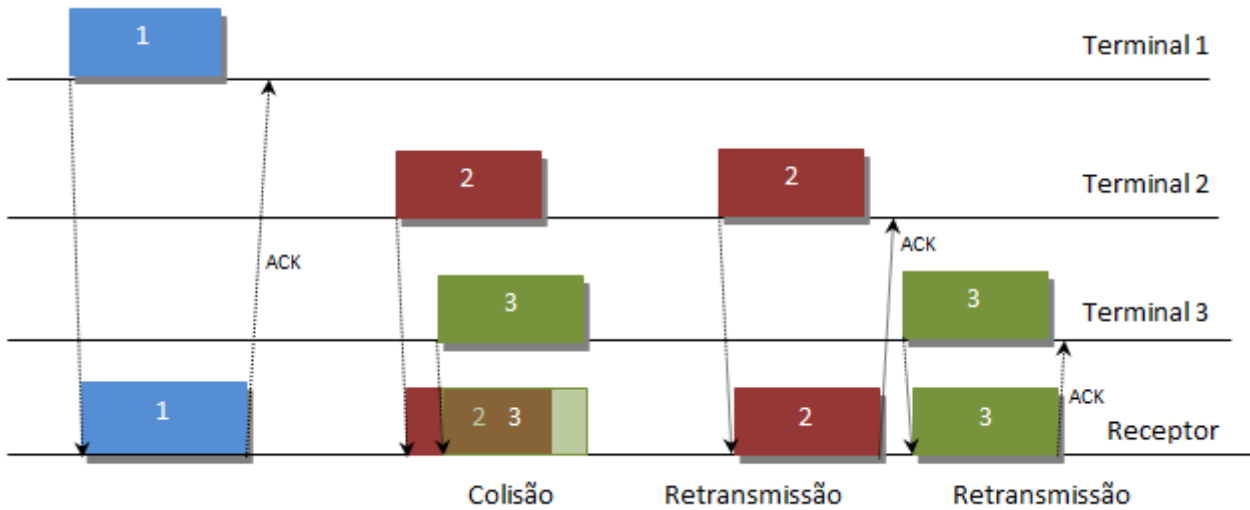


Figura 3.1: Funcionamento do ALOHA exemplificando o conceito de colisão e retransmissão.

O protocolo ALOHA, segundo os autores em [10], apresenta um desempenho muito fraco [10]. Desta forma para melhorar o desempenho da rede no sentido de se atingir uma eficácia mais elevada Lawrence G. Roberts propôs o protocolo de acesso ao meio *slotted ALOHA* [56].

### 3.3.2. Protocolo *Slotted ALOHA*

O protocolo *Slotted ALOHA* foi proposto em [56] para aumentar a eficiência do método proposto por N. Abramson no ALOHA [61]. No protocolo *Slotted ALOHA* o tempo é dividido em intervalos (*time slots*) com durações iguais ao tempo necessário para transmitir um pacote de dados no canal [61]. Cada terminal da rede é sincronizado de acordo com esses slots, de modo que, quando um nó quer enviar um pacote de dados, o pacote é transmitido no intervalo de tempo (*time slot*) seguinte [61].

A duração de um *slot*  $\sigma$  depende do atraso máximo de propagação e do tempo de transmissão do pacote [56], sendo definido como  $\beta \times \maxDelay + dataTime$ . Normalmente é escolhido o valor de  $\beta = 1$  para garantir que não haja nenhuma colisão, a menos que os pacotes de dados sejam transmitidos ao mesmo tempo, sobrepondo-se no receptor [56].

Na versão do protocolo *Slotted ALOHA* com ACK, a duração do *slot*  $\sigma$  é definida como  $2\beta \times \maxDelay + dataTime + ackTime$ . Neste caso, se durante a transmissão de um pacote de dados um nó emissor não receber um aviso de ACK ele tenta transmitir novamente, até um determinado número de *slots*, implementando também o mecanismo de *backoff*. O número de *slots* que um nó espera antes de retransmitir um pacote é escolhido aleatoriamente e uniformemente em  $[0, T]$ , onde  $T = \sigma 2^{txRetry}$ .

No protocolo *slotted* ALOHA, o intervalo de vulnerabilidade onde ocorrem colisões para qualquer pacote é reduzido a um tempo de pacote duas vezes menor do que o tempo do pacote no ALOHA, como mostra a figura 3.2.

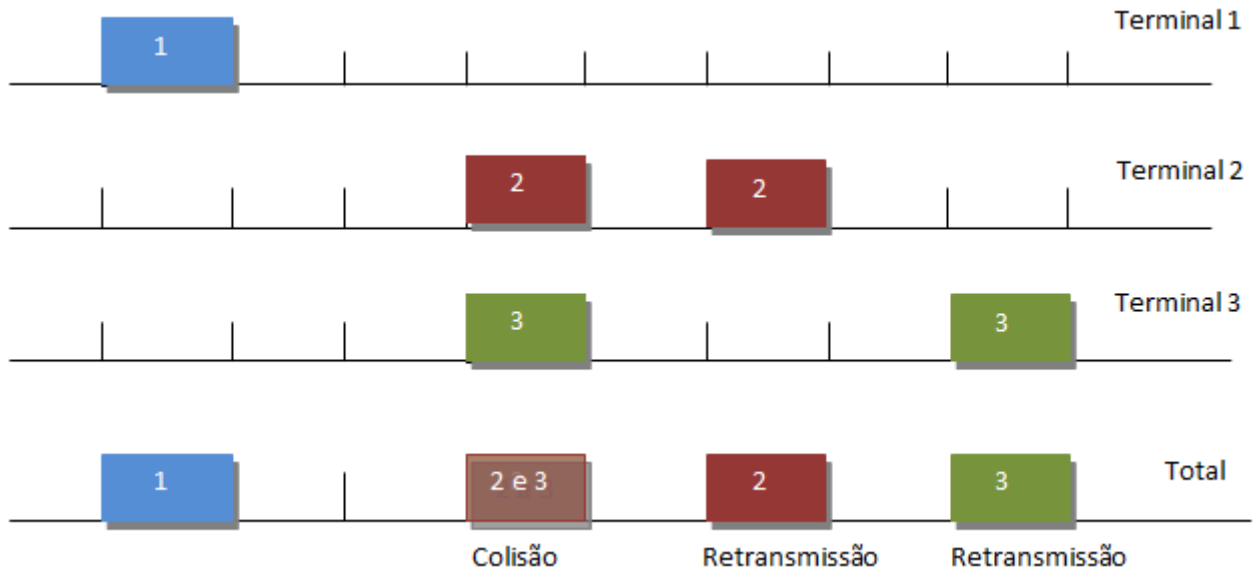


Figura 3.2: Funcionamento do *Slotted* ALOHA.

Segundo os autores em [10], o método usado pelo protocolo *Slotted* ALOHA, permite alcançar um débito na rede mais elevado do que o método usado pelo protocolo ALOHA [10]. Apesar disso o protocolo *Slotted* ALOHA demonstrou ser muito ineficiente para resolver problemas de congestionamento que podem surgir nas redes sem fios. Também não foi capaz de resolver problemas de colisões durante retransmissões [10].

### 3.3.3. Protocolo *Carrier Sense Multiple Access* (CSMA)

As baixas eficiências dos protocolos ALOHA levaram a que os investigadores procurassem soluções que reduzissem as probabilidades de colisões.

Os escassos recursos do canal podem ser melhor utilizados se os terminais implementarem um mecanismo eficiente de escuta do canal antes de transmitir o pacote. O protocolo Carrier Sense Multiple Access (CSMA) consiste em escutar o meio antes de tentar uma transmissão [63]. No que respeita às estratégias de escuta os protocolos CSMA podem ser classificados em:

- Não Persistente – Se o meio não está ocupado, transmite. Caso contrário se o meio estiver ocupado, espera um tempo aleatório e volta a escutar.

- 1 Persistente (ou apenas persistente) – Se o meio não está ocupado, transmite. Se o meio está ocupado, continua a escutar até o meio deixar de estar ocupado. Então transmite imediatamente.
- P Persistente – Se o meio não está ocupado, transmite com probabilidade P ou atrasa transmissão com probabilidade (1-P). Caso contrário se o meio estiver ocupado, continua a escutar até o meio deixar de estar ocupado e então transmite com probabilidade P ou atrasa transmissão com probabilidade (1-P).

O protocolo CSMA não elimina por si só as colisões, mas redu-las consideravelmente. Muitas variações do protocolo CSMA foram propostas e são hoje utilizadas nas redes sem fios para aumentar a eficiência do seu método, como por exemplo: CSMA/CD, CSMA/CA e o CSMA *with* ACK.

Por exemplo, o CSMA/CD ("*Carrier-Sense Multiple Access with Collision Detection*", com deteção de colisão) é usado em redes *ethernet*. Por sua vez, o CSMA/CA ("*Carrier-Sense Multiple Access with Collision Avoidance*", com prevenção de colisão) é popular em redes locais sem fios.

### 3.3.3.1. Protocolo CSMA/CD

O CSMA/CD (em que CD significa *Collision Detection*) é o método utilizado para controlar o acesso ao meio em redes *ethernet half duplex*.

O CSMA/CD funciona da seguinte forma:

- Se não detetar sinal no meio antes de transmitir começa a transmitir. Durante a transmissão verifica se houve colisão.
- Se detetar um sinal no meio antes de transmitir, espera que este sinal termine e começa a transmitir logo a seguir.

O mecanismo CD obriga os nós a escutarem a rede enquanto emitem dados, razão pela qual o CSMA/CD é também conhecido por "*Listen While Talk*" (LWT). Neste mecanismo, se for detetada uma colisão, deixa-se de transmitir e é emitido um sinal ("*jam*") com uma sequência de 48 bits para garantir que os outros terminais também a detetam. Posteriormente os terminais envolvidos esperam um determinado tempo aleatório antes de voltar a tentar utilizando um algoritmo de *binary exponential backoff* em que o tempo de espera para novas tentativas (até 15) vai duplicando em cada tentativa falhada. Após 15 tentativa admite haver falha na rede. Neste caso, existe um aspeto



importante a considerar para que as colisões sejam detetadas com sucesso: o tamanho mínimo dos pacotes deve ser tal que o seu tempo de transmissão seja superior ao dobro do atraso de propagação. Se isto não acontecer uma estação pode completar a emissão do pacote sem que o sinal produzido pela colisão chegue a tempo. Ou seja se  $t_p$  é o tempo de propagação,  $C$  a taxa de transmissão e  $M$  o tamanho mínimo do pacote tem-se:

$$M \geq 2 C t_p \quad (3.1)$$

O método CSMA/CD utiliza o algoritmo 1-Persistente que é o mais eficiente para utilização do meio de transmissão, ou seja o terminal transmite logo que o meio seja livre, como é apresentado na figura 3.3.

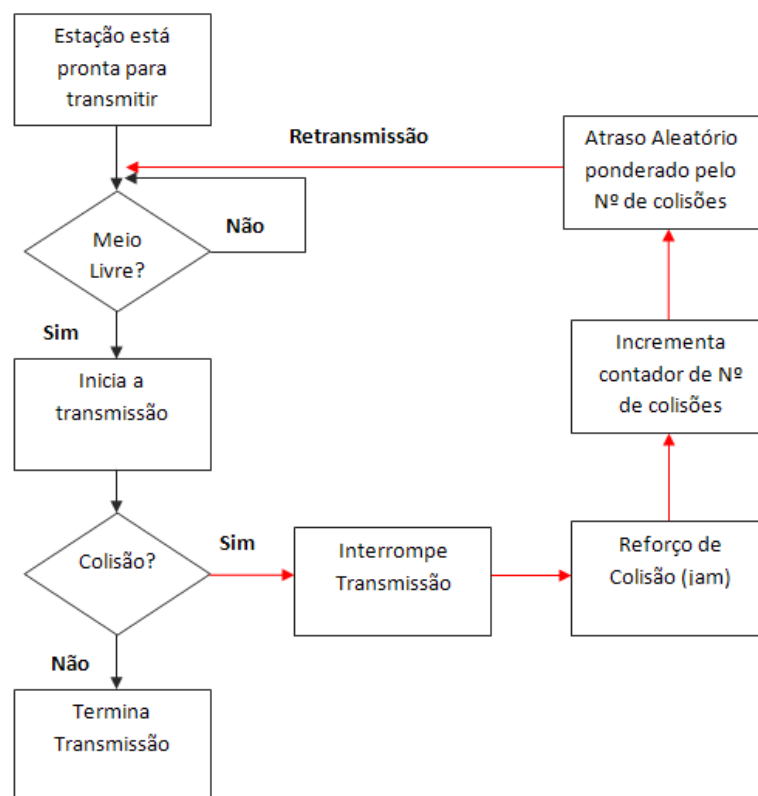


Figura 3.3: Fluxograma do algoritmo de funcionamento do CSMA/CD.

### 3.3.3.2. Protocolo CSMA/CA

O protocolo CSMA/CA (em que CA significa *Collision Avoidance*) é um protocolo utilizado nas redes sem fios [64]. O CSMA/CA difere do método CSMA/CD porque o seu método de transmissão possui um grau de ordenação maior que o seu antecessor (CSMA/CD) e também pelo facto de, no meio sem fios as colisões serem difíceis de detetar, uma vez que nem todos os terminais podem escutar-se uns aos outros [64].

O CSMA utiliza dois métodos básicos para acesso ao meio: o DCF (*Distributed Coordination Function*) e o PCF (*Point Coordination Function*) [64]. O método DCF fornece mecanismos às estações para negociar o acesso e para fornecer uma entrega fiável dos dados [64]. O método PCF fornece um serviço “*Connection oriented*” para tráfego com requisitos de largura de banda e atrasos tais como a VoIP. O PCF funciona com base em *Polling* em que uma estação coordenadora (*Point Coordinator*) vai perguntando a cada uma das outras se quer transmitir [64].

Em termos de acesso ao meio as estações a funcionar no modo PCF têm prioridade em relação ao terminais DCF. Essa prioridade é escalonada com base em intervalos de tempo diferentes que cada estação tem que esperar antes de tentar aceder ao meio [64].

São definidos os seguintes intervalos por ordem crescente de duração:

- *Short InterFrame Spaces* (SIFS) - é o intervalo de espera mais curto e é utilizado por todas as estações para enviar pacotes de pedido ou de confirmação (ACK, RTS, CTS, etc),
- *PCF InterFrame Spaces* (PIFS) - é o intervalo de espera intermédio que permite às estações PCF transmitir.
- *DCF InterFrame Spaces* (DIFS) - é o intervalo de espera máximo associado ao modo de DCF, dando-lhes a menor prioridade.

Assim, o CA significa evitar colisões onde elas são mais prováveis, ou seja depois de uma transmissão é provável que duas ou mais estações tentem transmitir ao mesmo tempo. O seu princípio de funcionamento é apresentado na figura 3.4 (adaptada a partir de [64]) e baseia-se no seguinte:

- Se o meio está livre quando inicia a escuta, continua a escutar durante um intervalo DIFS e se o meio continuar livre transmite,
- Quando o meio está ocupado espera que este fique livre e:
  - 1- espera um intervalo DIFS
  - 2- durante o período de *backoff* as estações que querem transmitir vão decrementando o *slot time* associado aos seus pacotes por cada *slot* escutado vazio,
  - 3- se *slot time* associado aos seus pacotes ficar igual a 0, transmite,

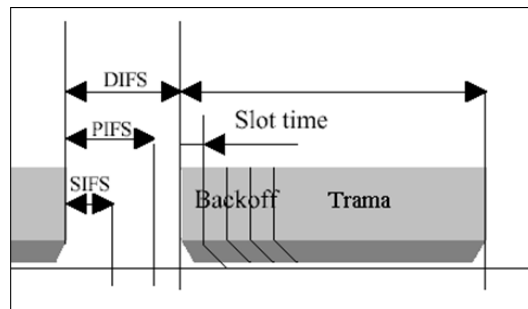


Figura 3.4: Estrutura de funcionamento do CSMA/CA.

O número de *slots* de espera por cada pacote é definido aleatoriamente (e uniformemente) logo à partida, de acordo com o congestionamento verificado entre 0 (zero) e  $CW$  (*Congestion Window*) *slots* [64]. O algoritmo de cálculo da  $CW$  funciona da seguinte forma:

O valor inicial da  $CW$  é inicialmente definido como  $CW_{min}$  (pré-determinado).

Por cada retransmissão  $n$  verificada o valor de  $CW$  é definido como:

$$2^n(CW_{min} + 1) - 1 \quad (3.2)$$

Até que  $CW$  atinja o valor pré-determinado  $CW_{max}$ , mantendo-se igual a  $CW_{max}$ .

O número de *slots* de espera por cada pacote vai sendo decrementado por cada *slot* que é escutado vazio. Assim, os pacotes que estão à espera mais tempo têm mais probabilidade de serem os seguintes.

Uma estação que receba bem um pacote envia um ACK no sentido inverso. Esse ACK tem mais prioridade uma vez que apenas tem que esperar um intervalo de tempo SIFS.

As colisões são deduzidas pela ausência de ACK e não através da escuta, tal como acontecia no *Collision Detection*.

O protocolo CSMA em redes sem fios utiliza habitualmente o pacote de ACK [65] para verificar a correta receção dos dados transmitidos. Se nenhum ACK for recebido dentro de um determinado tempo ( $\tau = 2T_p + T_{ack}$ , onde  $T_p$  é o atraso de propagação e  $T_{ack}$ , é o tempo de transmissão do ACK), o pacote de dados é retransmitido. Este processo é repetido até que, a receção seja bem-sucedida ou até que o limite máximo de tentativas (*maxRetries*) seja atingido [65]. O valor do atraso de propagação  $T_p$  é inicialmente ajustado com um valor superior ao atraso máximo de propagação (*maxDelay*) [65]. O tempo de *backoff* é escolhido de forma aleatória e uniformemente no intervalo  $[0, T]$  onde  $T = 2^{txRetry} (2maxDelay + T_{data} + T_{ack})$ . Onde  $T_{data}$  é o tempo necessário para transmitir um pacote de dados (na versão do CSMA sem aderência a ACK é assumido ser zero) [65].

### 3.3.3.3. Problemas do nó escondido e exposto

Na figura 3.5, podemos observar uma rede formada por três terminais, onde os círculos em torno de cada nó mostram o alcance da comunicação de cada nó.

O nó escondido ocorre quando numa transmissão de A para B como mostra a figura 3.5, um terceiro nó C estiver no alcance de transmissão de B mas não ao alcance de A. Por não ouvir a transmissão de A para B, C pode transmitir ao mesmo tempo, causando colisão em B (nó A oculto ou escondido a C) [10].

O CSMA é também suscetível ao problema do nó exposto, que ocorre quando B transmite para A na presença de C; como C está ao alcance de B, terá sua transmissão adiada, mesmo se quiser transmitir a nós diferentes de A e B (nó C exposto a B) [10].

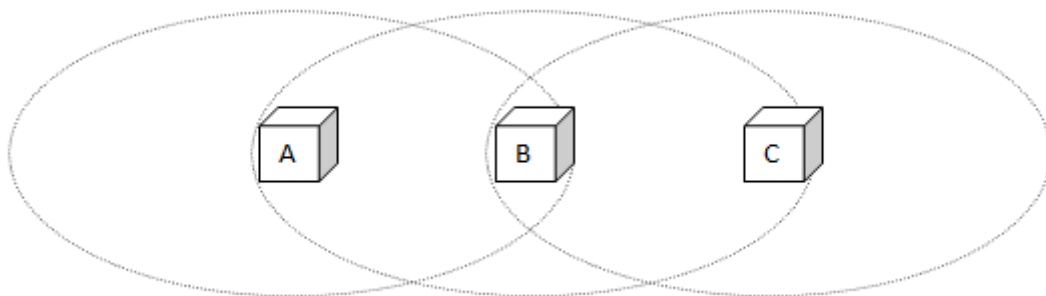


Figura 3.5: Funcionamento do CSMA exemplificando o problema dos nós expostos e escondido.

Para resolver estes problemas foram propostos mecanismos de envio de pacotes de controlo RTS e CTS, que passaremos a analisar em seguida.

### 3.3.4. Protocolo *Multiple Access with Collision Avoidance*

O protocolo MACA foi proposto por Karn em [60] para evitar colisões no recetor, como alternativa para o CSMA. Este protocolo usa dois pacotes de sinalização, o Request-to-Send (RTS) e o Clear-to-Send (CTS). Quando um nó A deseja enviar uma mensagem ao nó B, ele primeiro inicia o protocolo com o envio de um pequeno pacote de RTS, que contém o comprimento da mensagem que se pretende enviar ao nó B. Se o nó B receber o RTS e puder receber o pacote de A, ele responde com um CTS, que também contém o comprimento da mensagem. Ao receber o CTS o nó A inicia então a transmissão dos dados. A figura 3.6 apresenta a estrutura do funcionamento do protocolo MACA.

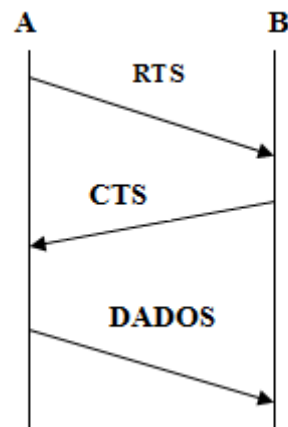


Figura 3.6: Estrutura de funcionamento do protocolo MACA.

Se um terceiro nó ouvir um RTS, mas não ouvir um CTS, decide que ele está fora do alcance do recetor e por isso pode transmitir o seu próprio pacote. Neste caso o protocolo MACA pode resolver tanto os problemas de nó escondido como o do nó exposto [10] que não foram resolvidos pelo protocolo CSMA.

No protocolo de acesso ao meio MACA o controlo de potência automático pode ser incorporado de uma forma simples [60]. Durante a mudança RTS-CTS, os nós da rede percebem o nível de potência mínimo necessário para uma comunicação confiável por tentativa e erro. Com controlo de potência, um nó que ouve um CTS não precisa terminar as suas transmissões. De forma a limitar a sua potência de saída a um nível que não vai ser ouvido pelo nó que enviou o CTS, pode continuar as suas transmissões com os nós que requerem menos potência de saída [60].

O protocolo MACA depende fortemente da simetria do canal, isto é, um CTS deve ser ouvido por todos os nós da rede dentro da faixa do nó recetor. No entanto, caso se utilize o controlo de

potência, esta simetria pode ser perdida [10]. Portanto, é melhor enviar sinais CTS a um nível de potência mais elevado para garantir que todos os nós dentro deste alcance possam ouvi-lo [10].

O protocolo MACA pode ser utilizado como base de um protocolo de acesso ao meio para redes acústicas submarinas. Os pacotes extra de RTS-CTS adicionam uma sobrecarga, mas a redução de retransmissões podem compensar esse aumento [10]. Para melhorar o protocolo MACA, Bharghavan *et al.* [61] propuseram o MACAW (*MACA-Wireless*) introduzindo um pacote do ACK e escalonamento.

### 3.3.5. Protocolo *MACA-Wireless*

O protocolo MACAW é uma versão modificada do protocolo MACA. Este protocolo foi proposto por Bharghavan *et al.* [61] para melhorar o desempenho e a fiabilidade do protocolo MACA com a transmissão de uma trama de ACK após cada receção bem-sucedida, possibilitando ao emissor a retransmissão de pacotes que não tenham sido corretamente recebidos pelo recetor. O seu princípio de funcionamento é apresentado na figura 3.7 (a).

Quando um nó A deseja enviar uma mensagem ao nó B, ele primeiro inicia o protocolo com o envio de um RTS ao nó B. Se o nó B receber o RTS, ele responde com um CTS. Ao receber o CTS o nó A transmite um pacote de dados ao nó B. O nó B ao receber o pacote de dados com sucesso transmite um ACK ao nó A após cada receção bem-sucedida.

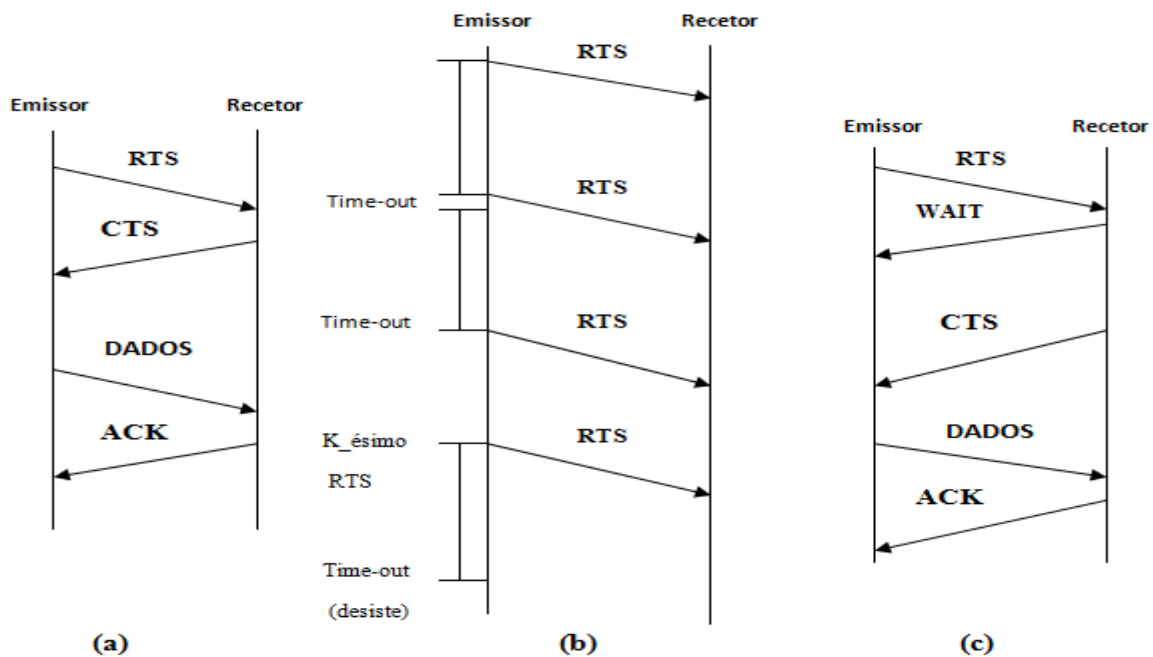


Figura 3.7: Protocolo *MACA-Wireless*: (a) funcionamento, (b) sem resposta CTS e (c) pacote *WAIT*.

Apesar da sobrecarga introduzida, o ganho obtido na eficiência é compensador. No entanto, a não recepção de CTS em resposta a um RTS leva o emissor a repetir o RTS por um número de vezes predeterminado, após o que, sem sucesso, desiste da comunicação como mostra a figura 3.7 (b). Esse esquema é inconveniente pois desperdiça energia quando o nó destino, apesar de ativo, estiver temporariamente impossibilitado de responder.

Para resolver esse problema em [66] Doukkali e Nuaymi propõem o acréscimo de um pacote de *WAIT*, a ser usado pelo destino quando estiver temporariamente impossibilitado de atender a um RTS. Um exemplo é como apresentado na figura 3.7 (c).

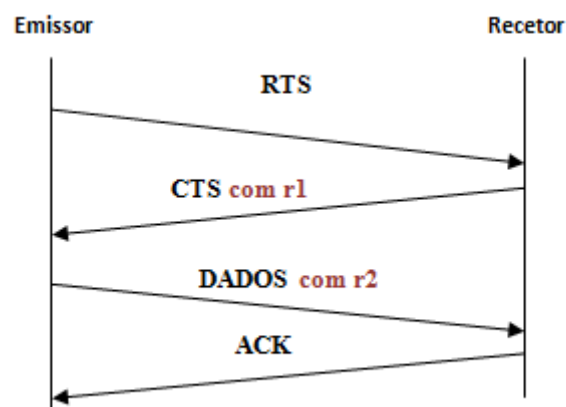


Figura 3.8: Teste de potências durante troca de tramas RTS e CTS.

Uma característica importante dos métodos que usam RTS e CTS consiste em que os nós da rede podem testar o canal durante essa troca de RTS e CTS. Apesar de ter sido especificado para redes sem fios, esta solução é muito conveniente para as redes acústicas submarinas, por possibilitar a adequação das potências de transmissão ao mínimo necessário para obter comunicação fiável. Em [67] Doukalli *et al.* descrevem esse esquema apresentado na figura 3.8. Os nós da rede transmitem o RTS e CTS, sempre com a potência máxima. Ao receber um RTS o nó de destino envia, junto com o CTS, uma informação sobre a relação entre a potência recebida e a potência mínima que necessita para decodificar corretamente os sinais recebidos ( $r1$ ). Através dessa informação o nó que enviou o RTS pode então combinar a potência necessária e suficiente para enviar dados de forma fiável aquele nó. Ao enviar o pacote de dados, o nó que iniciou a comunicação envia também a informação da relação entre a potência do CTS recebido e a potência mínima de que necessita para decodificar corretamente os sinais ( $r2$ ). Dessa forma o nó destino também pode adequar a potência das suas transmissões ao nó que iniciou a comunicação. Assim os nós da rede podem gerar uma tabela dinâmica com as potências necessárias às transmissões aos seus vizinhos.

O protocolo FAMA (*Floor Acquisition Multiple Access*) [68] aumenta a duração dos pacotes de RTS e CTS para além do tempo máximo de propagação da rede para prevenir as suas colisões com pacotes de dados. Foi mostrado pelos autores em [68] que a ausência de colisões é garantida se a duração do RTS for maior que o maior atraso de propagação e se a duração do CTS for maior que a duração do RTS, somado com o atraso na comutação do *hardware* entre TX (Transmissor) e RX (Recetor) e mais duas vezes o maior atraso de propagação [27].

Segundo os autores em [61], a adição de ACK em ambientes de grandes latências não foram considerados. Embora este esquema garanta ausência de colisões, a sua aplicação a redes acústicas submarinas representa um desperdício de energia devido aos elevados atrasos de propagação.

### **3.4. Mecanismos de controlo de acesso ao meio propostos para redes de acústicas submarina**

Baseando nos três protocolos de acesso ao meio para redes sem fios (MACA, MACAW e FAMA) foram apresentadas diversas propostas de adaptação para uso nas redes acústicas submarinas. Nesta seção, apresentaremos os mecanismos de funcionamento dos principais protocolos de acesso ao meio existentes nas redes acústicas submarinas.

#### **3.4.1. Protocolo *Slotted* FAMA**

O protocolo *Slotted* FAMA foi proposto por Molins e Stojanovic em [27], acrescentando *slots* de tempo ao esquema proposto no FAMA original, com o objetivo de limitar o impacto dos longos atrasos de propagação do canal acústico submarino. As transmissões só ocorrem dentro desses *slots* de tempo caso o canal esteja livre. Quando o emissor envia um RTS e não recebe o respetivo CTS no *slot* de tempo seguinte, assume que houve uma colisão, voltando a enviar um novo RTS após um período aleatório de *backoff* [27].

O princípio de funcionamento do protocolo *Slotted* FAMA baseia-se no seguinte: quando um nó tem um pacote de dados para enviar ele espera até o início do próximo *slot* e transmite um pacote de RTS. Este pacote é recebido pelo nó destino e todos os nós vizinhos do nó de origem dentro do *slot* de tempo [27]. O nó destino se não estiver ocupado com outras comunicações, responde com envio de um pacote CTS no início do *slot* seguinte. O pacote de CTS é recebido dentro do *slot* de tempo pelo nó de origem e por todos os nós no alcance do nó destino. Quando o nó de origem receber corretamente o CTS ele fica à espera até o início do próximo *slot* e, em seguida, começa a enviar o



pacote de dados. Quando o nó de destino receber o pacote de dados, ele envia um pacote de ACK para indicar que a transmissão foi bem-sucedida [27].

A figura 3.9 apresenta a estrutura do funcionamento do protocolo *Slotted* FAMA no caso de uma transmissão bem-sucedida.

O protocolo *Slotted* FAMA baseia-se na detecção de portadora. Isto significa que os nós da rede estão constantemente a ouvir o canal. Todos os nós da rede ficam em estado de repouso até que sintam a portadora no canal ou até que tenham dados para transmitir [27]. Se um pacote estiver pronto para ser transmitido no início de um *slot* e não for detetada nenhuma portadora, o nó envia um RTS e fica à espera durante os dois *slots* de tempo (o corrente e o próximo *slot*) por um pacote CTS. Se nenhum pacote de CTS for recebido durante este tempo, é assumido que houve uma colisão e o nó entra no estado *backoff*. A duração do tempo *backoff* é determinada aleatoriamente e uniformemente pelo número de *slots* escolhido no intervalo  $[1, S_{max}]$  [27], onde  $S_{max}$  representa o número médio de *slots* necessários para transmitir corretamente o pacote. Depois disso, o pacote de RTS é reenviado se não for detetado nenhuma portadora durante o tempo de *backoff*. Quando um CTS é recebido com sucesso, o nó começa a enviar o pacote de dados no próximo *slot*.

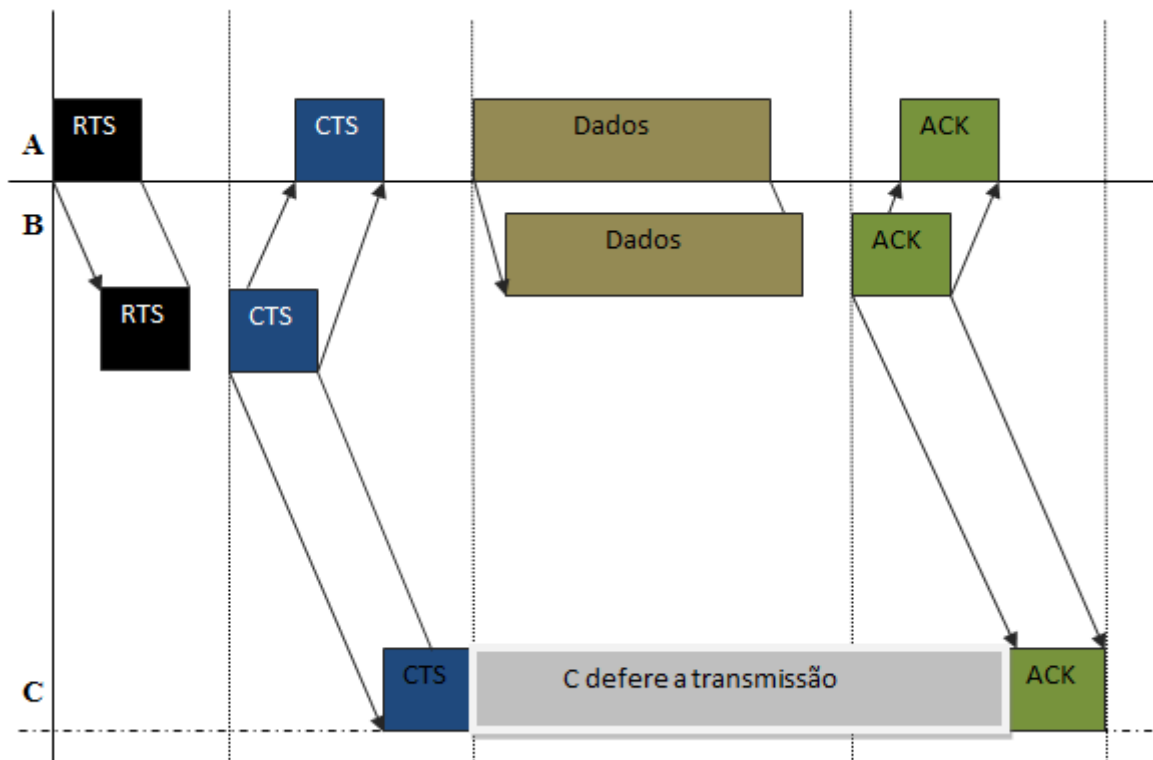


Figura 3.9: Estrutura de funcionamento do protocolo *Slotted FAMA*.

Quando um nó detetar uma portadora no canal entra no estado de receção, (enquanto estiver a receber o pacote). O tipo de pacote recebido irá determinar as ações do recetor, da seguinte forma:

- Depois de receber um pacote RTS destinado a outra estação o nó deve esperar dois *slots* de tempo (tempo necessário para que o recetor envie um CTS e o tempo necessário para emissor iniciar a transmissão de dados). Se após este período, nenhuma portadora for detetada, o nó volta para o estado de repouso [27].
- Depois de receber um pacote CTS destinado a outra estação um nó deve esperar o tempo suficiente de forma a permitir que a outra estação transmita o pacote de dados e receba o ACK correspondente. Como o nó recebeu o pacote de CTS, também irá receber o pacote ACK e, assim, saberá se a transmissão de dados terminou com êxito [27].
- Depois de receber um pacote de dados destinado a outra estação o nó deve esperar o tempo suficiente de forma a permitir a receção do pacote de ACK subsequente. Caso um nó não consiga ouvir o ACK deve esperar um *slot* de tempo adicional para detetar se o pacote de dados foi reenviado ou não [27].
- Depois de ouvir um ACK destinado a outra estação o nó apenas tem de esperar até ao fim do *slot* uma vez que a transmissão de dados terminou com sucesso [27].

- Se algum nó sentir alguma interferência no canal, é assumido que houve uma colisão. Se ele não souber quais foram os pacotes que colidiram é considerada a pior suposição, e ele age como se tivesse recebido um pacote de CTS que é uma situação que requer um longo tempo de espera [27].

### 3.4.1.1. Mecanismo de backoff utilizado pelo protocolo *Slotted*

#### FAMA

No protocolo FAMA original, quando um nó não recebe um CTS em resposta ao seu pacote RTS, ele entra no estado *backoff*. O tempo de *backoff* é definido aleatoriamente num intervalo entre um mínimo e um máximo, e depois do *backoff*, o pacote de RTS é reenviado [27]. Se for detetada uma portadora no canal enquanto um nó estiver no estado de *backoff*, o nó entra no estado de recepção, executando todas as operações requeridas pelo pacote recebido. Quando todas estas operações forem efetuadas, o nó retorna ao estado *backoff* e repõe um novo tempo de *backoff* [27].

Em ambientes onde há muito tráfego (ou seja, numa rede com um nó com muitos vizinhos) o tempo necessário para obter acesso ao canal pode ser muito longo em consequência dos *backoffs* [27]. Isto verifica-se em consequência do nó deixar o estado *backoff* devido a entrada de um pacote, e voltar a repô-lo no final [27].

Para atenuar este problema, o tempo de *backoff* pode ser reduzido, mas isso implicaria um outro problema: com tempos de *backoffs* reduzidos, a probabilidade de dois ou mais nós vizinhos escolherem o mesmo tempo aumenta e, portanto aumentaria consideravelmente a probabilidade do reenvio dos seus RTS no mesmo instante.

Neste sentido a solução proposta consistiu em redefinir o tempo de *backoff* cada vez que o nó retorne ao estado de *backoff* [27].

Isto significa que o tempo de *backoff* deve ser escolhido quando não há resposta a um RTS enviado. Neste sentido, é definido um temporizador (*timer*) com o tempo de *backoff* aleatório [27]. Se durante este tempo de *backoff*, o nó tiver que sair deste estado devido a uma nova transmissão, o tempo continuará a contar [27]. Quando o nó volta ao estado *backoff*, o tempo de *backoff* não é repostado [27]. Quando terminar o tempo, o pacote RTS é reenviado. Isto permite o protocolo melhorar o seu desempenho em situações de elevado tráfego sem ter que diminuir o tempo de *backoff* [27].

### 3.4.2. Protocolo de configuração do *handshake*

O protocolo de acesso ao meio baseado na configuração do *handshake* foi apresentado por Peleato e Stojanovic em [45], para adequar-se as redes acústicas submarinas com energia limitada e caracterizada por longo atraso de propagação. Este protocolo tem como objetivo limitar os efeitos da longa duração da troca dos pacotes de RTS e CTS, onde a temporização do *handshake* é proporcional à separação dos nós envolvidos na comunicação [45]. Para fazer isso, o protocolo determina a capacidade de um recetor em tolerar um certo nível de interferência [45]. Ao adaptar o processo de *handshake* à distância entre os nós consegue-se aumentar o rendimento e a eficiência, enquanto se evitam colisões o que por sua vez minimiza o consumo de energia [45].

Para evitar colisões, o protocolo alonga o processo de *handshake* para além do mínimo necessário, definindo os parâmetros de transmissão de acordo com o atraso máximo de propagação [45]. Para isso é definido um  $t_{min}$  (comprimento mínimo de *handshake*) que permitirá otimizar o protocolo para uma determinada rede.

Numa rede onde a maioria das ligações estão próximas do alcance de transmissão, deve ser atribuído ao  $t_{min}$  um valor tão longo como o tempo de ida e volta. Quando as ligações são mais curtas, pode ser reduzido [45].

O princípio de funcionamento do protocolo de configuração do *handshake* baseia-se no seguinte: ao receber um RTS, o nó recetor responde imediatamente com um CTS, e em seguida, escuta o canal e fica à espera do pacote de dados [45]. Se durante este período de escuta ele ouvir um pacote RTS para outro nó, ele envia um pequeno pacote de aviso para o seu vizinho (o nó a quem tinha enviado o CTS) [45]. Ao receber um CTS, o nó emissor espera algum tempo antes de transmitir o pacote de dados [45]. Se ouvir um outro CTS ou um aviso do seu vizinho durante este tempo, o nó interrompe a transmissão. A duração do período de espera depende da distância entre os nós e do tempo de ida e volta que o emissor utiliza para medição do RTS / CTS [45].

A figura 3.10 apresenta a estrutura de funcionamento do protocolo baseado na configuração do *handshake*.

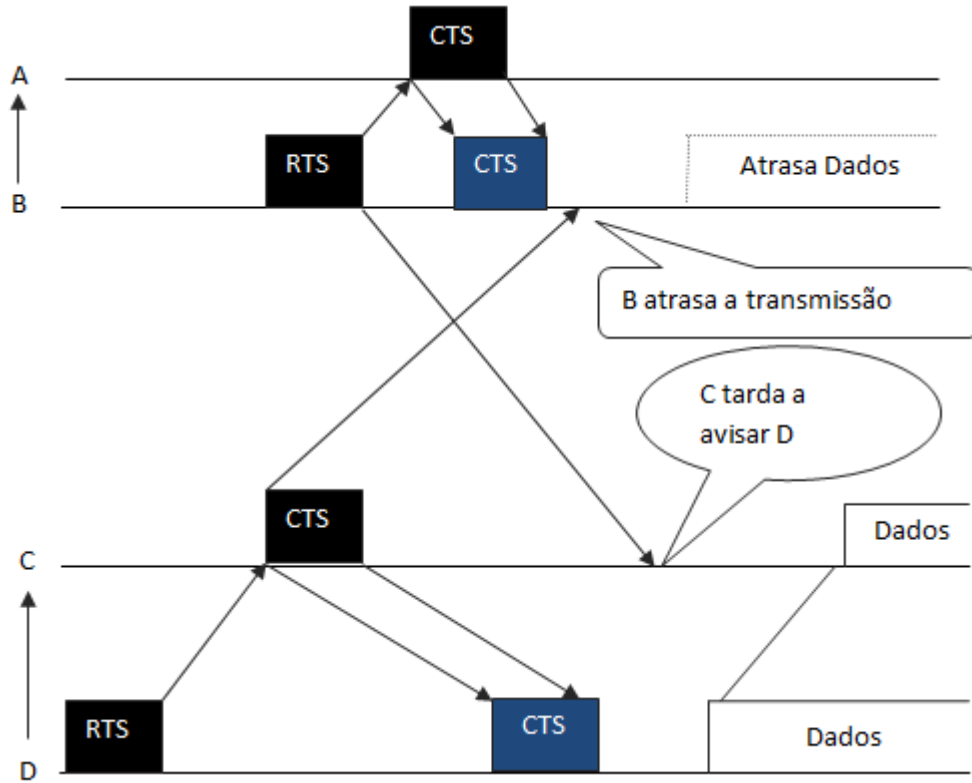


Figura 3.10: Estrutura de funcionamento do protocolo baseado na configuração do *handshake*.

Na figura 3.10, o nó B quer transmitir um pacote para o nó A, e o nó D quer transmitir um pacote para o nó C. Os nós B e A estão dentro do outro intervalo diferente dos nós D e C. No entanto, os nós B e C podem ouvir um ao outro, onde há uma grande probabilidade de interferência. Neste sentido a colisão pode ser evitada utilizando um pequeno aviso.

Considerando que  $U$  é a distância entre dois nós, e  $U + D$  é a distância mínima de interferência para que uma recepção seja bem-sucedida, os tempos de propagação correspondentes são obtidos dividindo as distâncias pela velocidade de som submarino. Assim, se  $t_{data}$  é a duração de um pacote de dados, o período de espera ( $T_W$ ) é determinado pela seguinte expressão [45]:

$$T_W = \begin{cases} t_{min} - \frac{2U}{c}, & \frac{U}{c} < t_1 \\ \frac{2(U + D)}{c} - t_{min}, & U/c \in (t_1, t_2) \\ \frac{2D}{c} + t_{data}, & U/c > t_2 \end{cases} \quad (3.3)$$

Onde [45]

$$t_1 = \frac{t_{min} - \min(\frac{D}{c}, t_{data})}{2} \quad (3.4)$$

$$t_2 = \frac{t_{data} + t_{min}}{2} \quad (3.5)$$

Existe uma restrição adicional de que  $T_w > 2D/c$  para evitar colisões de pacotes de dados com pacotes de controlo.

Devido a tamanhos de *handshakes* diferentes, podem colocar-se problemas de justiça, que são resolvidos atribuindo a  $t_{min}$  um valor de  $2T$  (o dobro do atraso máximo de propagação). Com esse valor, todos os *handshakes* têm a mesma duração e não são necessários avisos, mas o rendimento global é reduzido.

### 3.4.3. Protocolo *Tone-Lohi* (*T-Lohi*)

O protocolo *T-Lohi* é um protocolo de acesso ao meio adaptado para redes acústicas submarinas. Este protocolo foi proposto por Syed e Heidemann em [69] e baseia-se em contenção, onde os nós da rede concorrem no acesso ao meio para transmissão de seus dados.

O protocolo *T-Lohi* utiliza duas formas para poupar energia: alocação do canal para evitar colisão no campo de dados e utilização de um recetor de baixíssimo consumo para identificar tom de despertar [69]. Os nós da rede passam a maior parte do tempo em repouso, apenas ouvindo o meio com um recetor de baixo consumo. A alocação do canal é feita durante um Período de Reserva (PR) composto por Períodos de Disputa pelo meio (PD), tantos quantos necessários para eleger o vencedor [69]. Durante os Períodos de Disputa os nós da rede enviam curtos tons de disputa pelo meio [69]. Se um nó emitir um tom durante um desses períodos e não receber outros tons de disputa até o final do período de disputa em curso então ganhou o meio. Pode nesse caso encerrar o Período de Reserva e iniciar sua transmissão [69]. Se durante o Período de disputa ao meio houver tons de disputa de outros candidatos, todos os nós desistem da transmissão e executam individualmente um procedimento de *backoff* aleatório baseando no número de candidatos à transmissão naquele momento, contados pelo número de tons recebidos e tentam novamente no Período de Disputa seguinte, indicado pelo procedimento automático de *backoff* [69]. O nó que ganhar acesso ao canal envia um tom de despertar a todos os nós da rede, que despertam para verificar, pelo preâmbulo, se a mensagem no meio é a ele destinada. Se assim for processam-na, caso contrário voltam ao estado de repouso [69]. Ao nó que ganhou o meio e que acabou de transmitir seus dados é suprimida temporariamente a possibilidade de nova transmissão, com o objetivo de tornar o protocolo mais eficiente [69].

A figura 3.11 apresenta a estrutura de funcionamento do protocolo *T-Lohi* (adaptada a partir de [70]). O protocolo de acesso ao meio *T-Lohi* possui duas versões: uma versão mais agressiva designada de *Tone-Lohi* agressivo (*aT-Lohi*), e outra versão mais conservadora designada de *Tone-Lohi* conservador (*cT-Lohi*) [70]. Na versão agressiva do protocolo o comprimento do tempo de espera é igual à duração dos tons mais o atraso máximo de propagação enquanto que na versão conservadora do protocolo o tempo de espera é a duração dos tons mais duas vezes o atraso máximo de propagação [70].

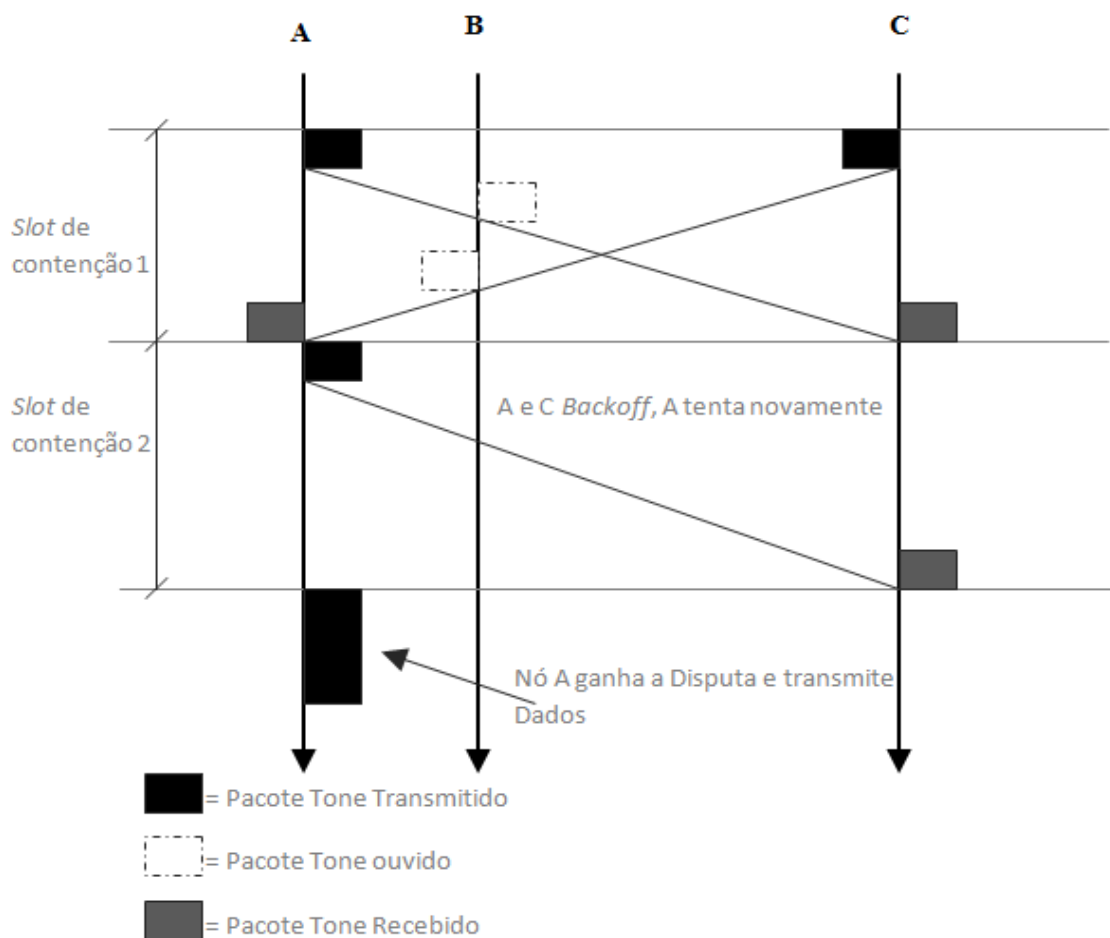


Figura 3.11: Estrutura de funcionamento do protocolo *T-Lohi*.

### 3.4.4. Protocolo PDAP (*Propagation Delay Aware Protocol*)

O protocolo PDAP é um protocolo de acesso ao meio proposto por Milica Stojanovic *et al* em [71] baseando no protocolo de acesso ao meio CSMA para redes sem fios que foi adaptado para redes acústicas submarinas utilizando o mecanismo de RTS/CTS para reserva do canal e para transmissão de dados [71]. O protocolo PDAP tem como objetivo maximizar a utilização da largura de banda,

mantendo o controlo das transmissões dos vizinhos de forma a evitar colisões e retransmissões, permitindo a transmissão de pacotes intercalados entre diferentes pares de nós [71].

Em seguida analisaremos o funcionamento do protocolo PDAP.

Quando um nó tiver um pacote de dados para transmitir, primeiro verifica o seu cronograma com o objetivo de encontrar tempos livres para troca de toda a informação, isto é, para enviar o RTS, para receber o CTS e enviar o pacote de dados [71].

O tempo de receção do CTS deve iniciar-se após a soma do tempo de chegada do RTS ao destino com o tempo de volta do CTS à origem. Do mesmo modo, o tempo de transmissão do pacote de dados deve ser programado imediatamente após o tempo de receção do CTS. Antes de enviar o RTS, o emissor fica à espera por um tempo aleatório de forma a evitar a sincronização com outros emissores [71]. Este tempo de espera é escolhido aleatoriamente e uniformemente no intervalo entre  $[0, T]$ , onde  $T$  é determinado pela seguinte expressão [71]:

$$T = 2^{txRetry} (2maxDelay + rtsTime) \quad (3.6)$$

em que  $txRetry$  indica o número de retransmissões de um pacote,  $rtsTime$  é o tempo que um nó demora para transmitir um pacote de RTS, e o  $maxDelay$  é o atraso máximo de propagação de um pacote na rede, calculado com base no alcance máximo de transmissão e na velocidade de propagação do som debaixo de água [71].

Quando um nó quer enviar um RTS, primeiro verifica o estado do canal. Se estiver no estado de repouso, o RTS é enviado, ficando à espera do CTS. Caso contrário, o emissor autoriza a programação (dos tempos do RTS, CTS e os tempos dos pacotes de dados), aumentando o contador de retransmissão e seleccionando novos tempos para transmissão do RTS, para receção do CTS e para transmissão de dados [71]. Ambos os pacotes de RTS e CTS contêm informações sobre a distância entre a origem e o destino. Esta distância é calculada com base no atraso de propagação do canal e no tempo do pacote de controlo [71].

Quando um potencial nó interferente receber um pacote de controlo de um nó vizinho, e no caso de ser agendada uma transmissão que poderia colidir com a sua própria comunicação, ele atualiza o seu cronograma, atrasando a sua transmissão [71]. Assim, a receção de RTS e CTS por um possível nó interferente é tratado da seguinte forma: quando o nó interferente receber um pacote RTS, ele define o seu NAV (*Network Allocation Vector*) de modo a que não sejam transmitidas informações de controlo destinadas ao emissor durante o tempo agendado para receção do CTS [71]. Esse nó também não irá transmitir um pacote de controlo destinado ao recetor, enquanto o recetor estiver a



enviar o CTS. Da mesma forma, o nó interferente não irá transmitir a informação de controlo destinado ao emissor durante a transmissão do pacote de dados [71]. Quando o nó interferente receber o CTS, ele atualiza o seu NAV de acordo com o tempo de receção do CTS. Como consequência, o nó interferente não transmitirá a informação de controlo destinada ao recetor durante a receção do pacote de dados, e também não transmitirá a informação de controlo ao emissor durante a transmissão de dados [71].

O protocolo PDAP não utiliza uma política agressiva para acesso ao canal, mas sim tenta agendar o maior número possível de transmissões paralelas. Os nós da rede tentam evitar colisões e os que são emissores são sincronizados durante a transmissão dos pacotes [71].

### **3.4.5. Protocolo DACAP (*Distance Aware-Collision Avoidance Protocol*)**

O Protocolo DACAP é um protocolo de controlo de acesso ao meio baseado no protocolo MACA, definido para redes sem fios. O DACAP foi proposto recentemente por Borja Peleato e Milica Stojanovic para redes acústicas submarinas [46]. Este protocolo implementa mecanismos para evitar colisões de pacotes maximizando o débito na rede, não exigindo que os nós da rede sejam sincronizados [46]. O protocolo DACAP também utiliza os pacotes sinalização RTS/CTS, e o pacote de aviso (*WARNING*) para transmissão de dados.

Em seguida passaremos a explicar o funcionamento do protocolo DACAP. Ao receber um RTS, o nó recetor responde imediatamente enviando um CTS e fica à espera de um pacote de dados. Se durante a espera, ele receber pacote proveniente de um terceiro nó, o recetor envia um pequeno aviso (*WARNING*) ao seu vizinho (o nó a quem tinha enviado o CTS). Ao receber um CTS o nó emissor espera durante algum tempo antes de transmitir o pacote de dados. Por sua vez, o nó que pretende enviar, se durante o período de espera escutar um pacote externo ou ouvir algum aviso (*WARNING*) então adia a sua transmissão.

A figura 3.12 apresenta um exemplo do mecanismo do funcionamento do protocolo DACAP.

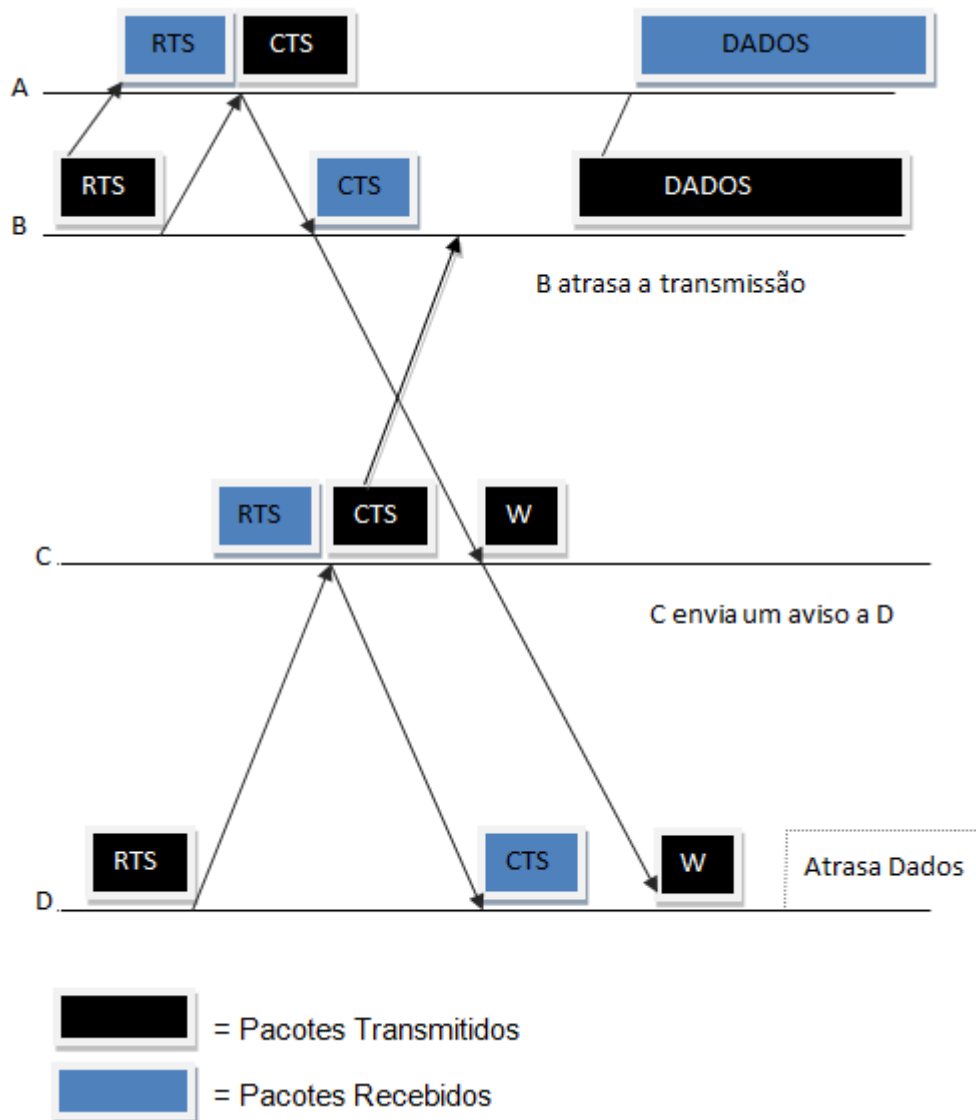


Figura 3.12: Estrutura de funcionamento do protocolo DACAP.

A duração do período de espera é concebido quando se verifica que a relação Interferência Sinal Ruído (SIR) é muito elevada para continuar a transmissão [46]. A duração desse período depende da distância entre os nós ( $U$ ) e o tempo de ida e volta dos pacotes RTS/CTS (i.e.  $2U/c$ , onde  $c$  é a velocidade de propagação do som). Normalmente é escolhido um período de espera que permita garantir ausência de colisões [46]. Também são utilizados os protocolos de sinalização *handshake* para evitar colisões entre nós a uma determinada distância, isto é, as confirmações de *handshake* entre nós vizinhos são feitas rapidamente, enquanto que entre nós distantes são feitas lentamente.

Para maximizar o débito da rede, é definido o comprimento mínimo de *handshake* ( $t_{min}$ ) para todos os nós. Numa rede onde a maioria das ligações estão próximas da faixa de transmissão,  $t_{min}$  deve ser igual ao dobro do atraso máximo de propagação ( $2T$ ), podendo ser reduzido quando as ligações são mais curtas.

O protocolo de acesso ao meio DACAP possui duas versões: uma versão com ACK, e outra versão sem ACK. Na versão com ACK, o protocolo exige sincronismos ligeiramente diferentes em relação à versão sem ACK, a fim de acomodar a mensagem de ACK [46].

### 3.4.5.1. Versão do protocolo DACAP sem ACK

Quando não é utilizado o ACK, o recetor só envia um aviso se ocorrer um RTS entre  $2T - t_{min}$  segundos depois de enviar um CTS. O emissor só adia a transmissão se ele receber um aviso do seu vizinho, ou ocorrer um CTS nos próximos  $t_{min}$  segundos depois de enviar um RTS.

Considerando  $U + \Delta D$  a distância mínima de interferência do nó em caso de sucesso na receção e  $t_{data}$  a duração do pacote de dados transmitidos, o período de espera ( $T_w$ ) é determinado pela seguinte expressão [46]:

$$T_w\left(\frac{U}{c}\right) = \begin{cases} t_{min} - \frac{2U}{c}, & \frac{U}{c} < t_1 \\ \frac{2(U + \Delta D)}{c} - t_{min}, & \frac{U}{c} > t_1 \end{cases} \quad (3.7)$$

Em que [46]:

$$t_1 = \frac{t_{min} - \min\left(\frac{\Delta D}{c}, t_{data}, 2T - t_{min}\right)}{2} \quad (3.8)$$

Há ainda uma restrição adicional que determina que  $T_w > 2\Delta D/c$ .

### 3.4.5.2. Versão do protocolo DACAP com ACK

Na versão do protocolo DACAP com ACK são introduzidos dois novos parâmetros:  $\Delta t_{data}$ , que é a diferença máxima entre duração de dois pacotes de dados e  $T_{w-min}$ , um tempo mínimo de espera predeterminado  $T_w$ . Deve ser escolhido um  $t_{min}$  superior a  $\min\left(T, 2\max_U\left(\frac{\Delta D}{c}\right) + \Delta t_{data}\right)$  e um  $T_{w-min}$  inferior a  $\min(T, t_{min})$ .

O recetor envia um aviso quando ocorrer um CTS entre  $2T - T_{w-min}$  segundos, ou um RTS entre  $2T - t_{min}$  segundos após transmitir um CTS. O emissor adia a transmissão se receber um aviso, ou ocorrer um CTS entre  $t_{min}$  segundos, ou um RTS em  $T$  segundos após o envio de RTS. O período de espera ( $T_w$ ) é determinado pela seguinte expressão [46]:

$$T_w\left(\frac{U}{c}\right) = \begin{cases} \frac{2(U + \Delta D)}{c} - t_{min}, & \frac{U}{c} \in (t_1, t_2) \\ \frac{2(U + \Delta D)}{c} - T_{w-min} & \frac{U}{c} > \max(t_2, t_3) \\ t_{min} - \frac{2U}{c} & \text{outros} \end{cases} \quad (3.9)$$

Onde [46]

$$t_1 = \frac{t_{min} - \min\left(\frac{\Delta D}{c}, t_{data}, 2T - t_{min}\right)}{2} \quad (3.10)$$

$$t_2 = \frac{t_{min} - \Delta t_{data}}{2} \quad (3.11)$$

$$t_3 = \min\left(t_1, \frac{t_{min} + T_{w-min} - \frac{2\Delta D}{c}}{4}\right) \quad (3.12)$$

Há uma restrição adicional que é expressa por [46]:

$$T_w\left(\frac{U}{c}\right) > \max\left(\frac{2\Delta D}{c}, T_{w-min}\right)$$

### 3.4.5.3. Mecanismo de *backoff* utilizado pelo protocolo DACAP

Quando houver uma confirmação de ocorrência de um RTS ou CTS, entre nós no mesmo intervalo de transmissão, é adicionado um mecanismo de *backoff* para evitar colisões [46]. Enquanto não forem conhecidas as distâncias entre os nós, ao tempo do *backoff* é atribuído um valor acima do que o necessário.

Para minimizar as colisões, o mecanismo de *backoff* propõe duas estratégias [46]:

- Os nós da rede continuam a escutar o canal durante o estado de *backoff*, e voltam ao estado de repouso ou inativo se for detetado o fim da transmissão (pacote de dados ou ACK).
- Se um nó receber um RTS de um ou mais nós envolvidos na transmissão ele responde com um CTS e sai do estado de *backoff*.

### 3.5. Protocolos de encaminhamento

A camada de rede é responsável por determinar o caminho entre a fonte e um nó de destino (geralmente na estação de superfície).

Nos últimos anos tem havido um estudo intensivo em protocolos de encaminhamento para redes ad hoc sem fio e redes acústicas submarinas.

Os protocolos de encaminhamento são responsáveis por decidir qual o caminho que os pacotes terão que fazer para viajar entre dois nós que não estejam diretamente alcançáveis. Os protocolos de encaminhamento podem ser genericamente classificados em quatro grupos, como pró-ativos, reativos, híbridos e geográficos.

Os pró-ativos são baseados na atualização e divulgação periódica das tabelas de encaminhamento, com rotas definidas entre todos os nós da rede. Assim as rotas estão sempre atualizadas, mas esse procedimento consome banda e energia, para manter rotas que por vezes nunca são usadas. Estes tipos de protocolos não são adequados às redes acústicas submarinas por implicarem uma grande sobrecarga devido à contínua atualização das tabelas de encaminhamento, desperdiçando energia.

No caso dos protocolos reativos a fonte inicia a descoberta da rota para o destino, quando, e se, necessitar de enviar um pacote para esse nó. Nos protocolos reativos a sobrecarga é menor ou seja as rotas são determinadas apenas quando for necessário, mas há um atraso significativo imposto pela determinação da rota. Entretanto, também requerem o inundar da rede com pacotes de controlo para descoberta das rotas, o que também os tornam inadequados para uso nas redes acústicas submarinas.

Os protocolos híbridos baseiam-se na combinação dos dois métodos pró-ativos e reativos. O objetivo principal é reduzir transmissões *broadcasts* e latência, mas também melhorar o impacto e o dinamismo. Toda a rede será separada em zonas, e cada zona terá um *gateway*. Dentro de cada zona, um protocolo reativo é considerado. Para o encaminhamento entre zonas diferentes, um protocolo dinâmico ou pró-ativo é usado.

Os protocolos de encaminhamento geográficos baseiam-se nas coordenadas dos nós para definição das rotas e necessitam de poucos pacotes de controlo, sendo muito eficientes e de baixo consumo. Entretanto, a localização dos nós nas redes acústicas submarinas ainda é uma questão a ser resolvida, já que no ambiente submarino não há um sistema de posicionamento global, como o GPS para ambientes terrestres (de [72] a [75]).

Neste sentido, nenhum dos quatro tipos de protocolos de encaminhamento são adequados às características e limitações das transmissões acústicas submarinas.

De qualquer forma, devido à diminuição da banda do canal acústico com a distância transmissor-recetor, Stojanovic et al. demonstraram em [10] que teoricamente o consumo total de energia de uma rede, quando os nós encaminham os seus dados ao nó recetor (*Sink*) através de múltiplos e pequenos saltos, é menor do que se todos os nós transmitissem os seus dados diretamente ao nó *Sink*. Neste sentido é necessário a pesquisa e desenvolvimento de protocolos de encaminhamento para este tipo de redes. No entanto, dado que o maior problema ainda é o de controlo de acesso ao meio, poucas propostas têm sido apresentadas para encaminhamento em redes acústicas submarinas devido às características adversas do canal acústico submarino.

### **3.6. Desenvolvimento das redes acústicas submarinas**

Atualmente poucas aplicações das redes acústicas submarinas foram desenvolvidas. A Rede Área Local Acústica (ALAN) para águas profundas foi desenvolvida por Monterey Canyon, CA [76]. A rede foi concebida para transferir dados dos nós de fundo do oceano para uma estação de superfície implementando comando e controlo de sinais na direção inversa. Sempre que um nó pretende enviar uma determinada quantidade de dados, emite uma solicitação ao recetor da superfície relativa ao tamanho de dados que vai ser transmitida. Após a receção do pedido, o recetor da superfície programa o tempo de transmissão (que depende do atraso de propagação) e notifica o nó de origem com um ACK. O nó de origem imediatamente envia o pacote de dados depois de receber o ACK. O pedido, o ACK, e as transmissões dos dados são realizados em faixas de frequências diferentes. Se ocorrer uma colisão no canal onde foi feito o pedido, o recetor tenta decifrar a colisão através do uso de uma estrutura com vários utilizadores, aumentando o débito da rede, eliminando a necessidade de retransmissões. Uma vez que este protocolo baseia-se numa avaliação correta dos tempos de propagação, qualquer erro no processo de avaliação diminui a taxa de transmissão, causando retransmissões. No entanto, uma alternativa para esta solução é fixar o tempo entre transmissões e o tempo máximo de propagação.

O protocolo *store-and-forward* foi proposto em [77] para ser utilizado nas Redes de Área Local Acústicas (ALAN) para águas com poucas profundidades. Esse protocolo é uma versão modificada do protocolo da Rede de Pacotes Rádio (PRN) [78]. No protocolo *store-and-forward* cada nó na rede usa três canais diferentes (bandas de frequência): para transmitir, para receber pacotes do seu antecessor, e para receber as confirmações do ACK do seu sucessor.

Em [79] foi desenvolvido um protocolo de comunicação ponto a ponto para controlar AUV (*Autonomous Underwater Vehicles*). O protocolo de acesso ao meio utilizado por esse protocolo é o

CSMA. Quando uma mensagem estiver disponível para ser transmitido por um nó, ele começa a escutar o canal. Se não receber um ACK do destino, o emissor envia novamente o pacote após um período de tempo, calculado utilizando o tempo máximo de propagação.

### **3.7. Conclusões do capítulo**

Neste capítulo foram apresentadas os principais protocolos de acesso múltiplo ao meio, protocolos de acesso ao meio para redes sem fios em redes terrestres. Foram também apresentadas as principais propostas de adaptações dos protocolos de acesso ao meio para redes acústicas submarinas, incluindo soluções para minimizar as colisões nas comunicações acústicas submarinas devido a enorme latência. O capítulo também apresentou os mecanismos de funcionamento dos principais protocolos considerados clássicos para redes acústicas submarinas. Os protocolos de encaminhamento para redes acústicas submarinas foram apresentados de forma resumida já que ainda estão numa fase inicial de desenvolvimento, devido às características adversas do canal acústico submarino.

Após análise dos mecanismos de funcionamento dos principais protocolos de acesso ao meio clássicos para redes acústicas submarinas, propomos a implementação do mecanismo de funcionamento do protocolo DACAP por ser um dos protocolos mais recentes e por apresentar melhor desempenho nas redes acústicas submarinas de acordo com os autores em [46].





## Capítulo 4

# Implementação do Mecanismo DACAP

Neste capítulo irá ser apresentado um mecanismo de acesso ao meio proposto para resolver os problemas de colisões existente nas redes acústicas submarinas, caracterizada por longos atrasos de propagação e diferentes requisitos de potência de transmissão/receção. O princípio de funcionamento em que se baseia o protocolo foi descrito no capítulo 3 na seção 3.4.5, apresentando as fases crucias do seu funcionamento. Neste sentido baseando no seu mecanismo e fazendo um estudo prévio dos mecanismos de funcionamento dos protocolos existentes no simulador de rede propomos a implementação do protocolo DACAP, apresentando propostas para melhorar o seu desempenho de forma a obter ótimos resultados nas redes acústicas submarinas.

### 4.1. Visão geral

O constante avanço nas tecnologias de *modems* acústicos, as pesquisas em comunicações acústicas submarinas estão atualmente muito focadas no desenho dos principais protocolos de acesso ao meio para redes sem fios em ambientes submarinos [10]. Os desafios no ambiente acústico submarino incluem largura de banda limitada (tipicamente em poucos kHz a distância de vários quilómetros), má qualidade da ligação física e alta latência causada pela baixa velocidade de propagação do som ( $c = 1500$  m/s). Além disso, atualmente os modems acústicos submarinos disponíveis operaram em half-duplex e a potência necessária para a transmissão é tipicamente muito maior do que a que se encontra na receção.

Como forma de resolver o problema, a aplicação directa do controlo de acesso a meio baseando nos seus princípios de funcionamento torna-se necessária. Neste sentido em vez de modificar os protocolos existentes, optou-se por uma abordagem diferente e pela implementação de um protocolo que aproveita os atrasos de propagação e perda de energia do sinal de forma a maximizar o *throughput*. Para além disso, é importante evitar colisões de pacotes no recetor quando os pacotes

de dados estão a ser transmitidos. Este facto foi reconhecido por diversos autores, o que resultou na definição e implementação de protocolos MAC que são particularmente adequados para o canal acústico submarino. Refira-se a este nível os trabalhos [27], [45] e [80], onde os autores propuseram procedimentos baseados em *handshakes* que garantam a ausência de colisões. No entanto, estes protocolos requerem longos períodos de *handshake* e um elevado nível de sincronização entre os nós. Ao definir os parâmetros de transmissão de acordo com o atraso máximo de propagação, o processo de *handshake* é prolongado para além do mínimo necessário, o que torna estes protocolos ineficientes em ligações muito mais curtas [46].

Desta forma para resolver este problema, na seção 4.3 será apresentada a implementação do mecanismo do protocolo de acesso ao meio DACAP no simulador de rede, um protocolo não sincronizado que permite a um nó usar diferentes tamanhos de *handshakes* para diferentes recetores, de modo a minimizar a duração média do *handshake*.

## 4.2. Funcionamento da camada MAC no simulador de rede

Com intuito de testar o protocolo de acesso ao meio analisado nesta dissertação, foi feita a sua implementação no simulador de rede WOSS (apresentado no capítulo 2).

Uma das razões pela qual são desenvolvidos APIs, conjunto de rotinas, programas e normas estabelecidas pelo programa para utilização das funcionalidades da camada física (PHY) para o desenvolvimento da camada MAC é que o simulador WOSS não fornece suporte nativo para simular diferentes fases de transmissão e receção de pacotes, nem no MAC nem na camada física.

A figura 4.1 apresenta um diagrama de sequência de pacotes e eventos de Transmissão e Receção (TX/RX) no simulador de rede WOSS. Como se pode observar neste diagrama, a transmissão do pacote passa por rotinas que gerem a camada física e MAC. No recetor, a camada física (PHY) só tem conhecimento do início da transmissão, enquanto que a camada MAC tem noção do início e final da receção. Inicialmente sempre que a camada MAC e física necessitam executar uma operação (após a conclusão do pacote) eventos dedicados precisam ser gerados. Seguidamente a duração de uma transmissão é determinada na camada física, uma vez que depende do tamanho do pacote, do esquema de codificação, modulação e de outros aspetos específicos tais como o comprimento do preâmbulo de sincronização. Assim a camada MAC deve agendar os eventos necessários que envolvem duplicação de atributos da camada física na camada MAC e funcionalidades que podem levar a inconsistências, fraca legibilidade e manutenção do código.

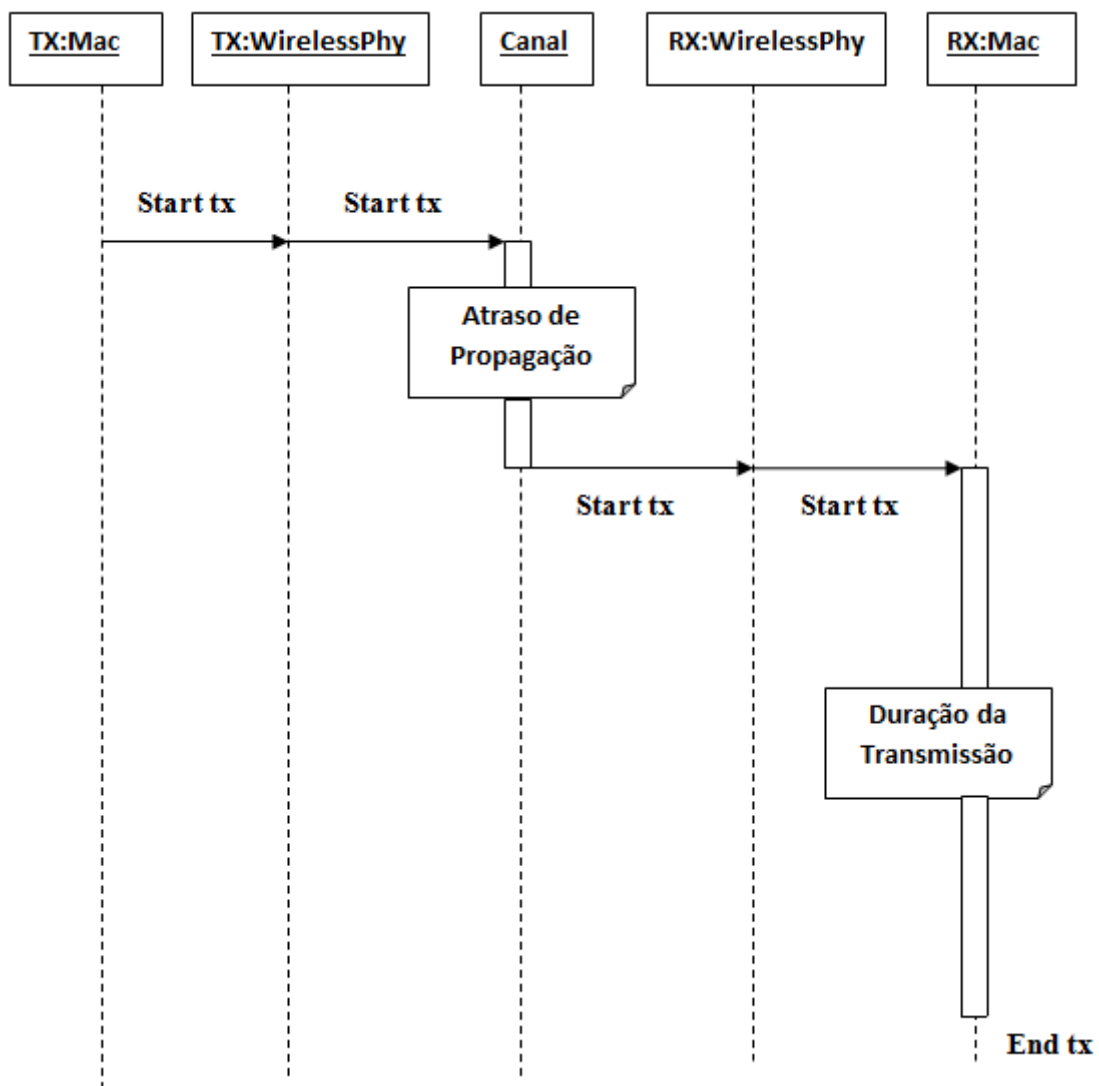


Figura 4.1: Diagrama de sequência dos pacotes para eventos TX/RX no WOSS.

Para resolver estes problemas foi concebido em alternativa o diagrama de sequência dos pacotes com eventos para TX/RX no simulador WOSS como apresentado na figura 4.2 com ferramentas que suportam elevado grau de complexidade e flexibilidade para camada física e camada MAC. Este diagrama traz vantagens para o simulador de rede primeiro de tudo, porque a duração da transmissão de um pacote neste caso será determinada pela camada física, além disso, a programação dos eventos do início/fim das transmissões e receções serão fornecidas pela classe do modelo físico (MPhy), que tem como função chamar os parâmetros de entrada para o código com funcionalidades específicas para camada física. Várias mensagens de camada cruzada com funcionalidades fornecidas pela estrutura do simulador WOSS são definidas para que as classes derivadas do modelo físico possam desencadear os eventos de transmissão/receção ou início/fim na camada MAC (MMac). A classe do modelo MMac define os seguintes métodos: *Phy2MacEndTx* (),

*Phy2MacStartRx()* e *Phy2MacEndRx()* que são chamados para recepção das mensagens acima mencionadas e podem ser usados pelas classes que herdam o modelo MAC (MMAC) com objetivo de facilitar a implementação do código do protocolo especificado e serem executados em resposta aos eventos correspondentes. O modelo da camada MAC para além dos métodos referenciados acima também possui o método *Mac2PhyStartTx* fornecido pela classe *Mac2PhyStartTx()*. Este método fornece parâmetro de entrada para o código para ser executado no início de uma transmissão. Este método é responsável por iniciar uma transmissão do pacote da camada MAC para camada física. O método *Phy2MacEndTx* fornece parâmetros de entrada para o código para ser executado no final da transmissão do pacote. Esse método é também utilizado para finalizar uma transmissão da camada física para camada MAC. Enquanto que esses dois métodos são utilizados apenas para desencadear as transmissões, os métodos *Phy2MacStartRx* e *Phy2MacEndRx* são utilizados para desencadear as recepções dos pacotes.

O método *Phy2MacStartRx* fornece parâmetros de entrada para o código para ser executado no início da recepção. Este método deve saber lidar com o processo de aquisição na transmissão do pacote e detecção do preâmbulo de sincronização. Deve também responsabilizar-se por todas as ações relativas ao início da recepção da camada física para camada MAC.

O método *Phy2MacEndRx* fornece parâmetros de entrada para o código para ser executado no final da recepção. Este método é responsável por determinar a presença de erros no pacote, usando um modelo de erro apropriado para camada física e para uma eventual ocorrência de encaminhamento dos pacotes para as camadas superiores. Esse método também é utilizado para finalizar a recepção de um pacote, da camada física para a camada MAC.

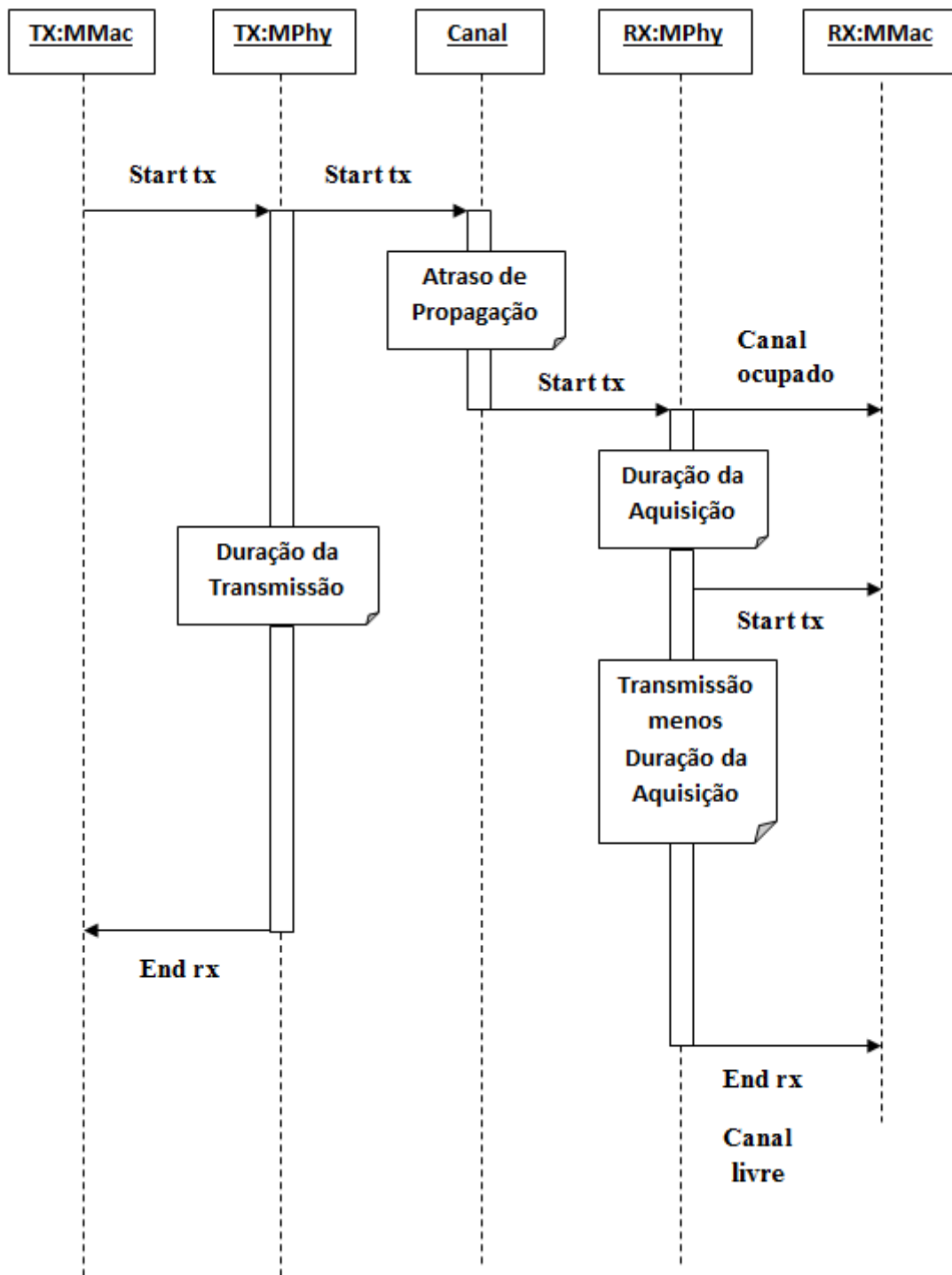


Figura 4.2: Diagrama de sequência dos pacotes para eventos TX/RX no WOSS para camada física e MAC.

Na próxima seção apresenta-se o mecanismo de implementação do protocolo DACAP onde será apresentado com mais detalhe o funcionamento desses métodos no simulador de rede.

## 4.3. Implementação do protocolo DACAP proposto

### 4.3.1. Procedimentos e testes iniciais

Nesta seção apresenta-se as principais fases e etapas estabelecidas durante o processo de implementação do protocolo DACAP. Numa primeira fase, após o estudo do modelo da camada MAC no simulador de rede e compreensão do funcionamento dos principais mecanismos dos protocolos de acesso ao meio propostos para rede acústica submarina começou-se por instalar o simulador WOSS e todas as ferramentas para o real funcionamento da camada MAC no simulador de rede. Em seguida deu-se início à fase de exploração dos ficheiros com exemplos de implementação dos protocolos MAC existentes no simulador de rede.

Neste sentido, começou-se por perceber o funcionamento do protocolo ALOHA sem ACK (ALOHA *no* ACK). Este protocolo não considera o estado do canal resultando em altas taxas de colisões, não havendo transmissão de ACKs. Ou seja quando um emissor tiver um pacote de dados pronto para transmitir ele envia-o imediatamente sem escuta prévia causando elevadas taxas de colisões. Desta forma para minimizar o efeito de colisões no protocolo ALOHA sem ACK foi implementado uma alteração no código do protocolo de forma a escutar o meio antes de tentar uma transmissão, ou seja quando um emissor tiver um pacote de dados para enviar primeiro escuta o canal e só depois envia o pacote ao recetor. Este protocolo alterado foi designado de ALOHANEW sem ACK (ALOHANEW *no* ACK). De seguida por forma a testar estes dois protocolos ALOHA foram feitas vários testes variando o número da *seed* (semente) do gerador pseudo-aleatório por forma a garantir pequenas variações no período entre pacotes. Os resultados obtidos com estes dois protocolos são apresentados no (anexo C) da presente dissertação. Dos dados obtidos por simulações verificou-se que o protocolo ALOHA *no* ACK (Sem escuta) apresenta uma perda ligeiramente superior ao ALOHANEW *no* ACK (Com escuta). Nesse sentido, os resultados obtidos indicaram que a solução de escuta por si só não é suficientemente dado os elevados atrasos, requerendo a utilização combinado de outros mecanismos.

A solução encontrada para resolver este problema foi fazer o estudo de um protocolo que apresenta um mecanismo com maior grau de complexidade que não só apresentasse a solução de escuta mas também mecanismos de *backoff*, transmissão de ACK e retransmissão de pacotes.

Começou-se então por analisar a implementação do mecanismo do protocolo ALOHA com ACK (ALOHA *With* ACK) existente no simulador de rede. No protocolo ALOHA *With* ACK [10] se a

transmissão do pacote de dados for bem-sucedida, isto é, se nenhum pacote for recebido com erro, o recetor envia um aviso de ACK ao emissor num intervalo de tempo igual ao tempo de propagação [10]. Se nenhum ACK for recebido num intervalo de tempo definido significa que houve perda de pacote na rede devido a uma colisão no canal, o pacote será enviado novamente com um atraso aleatório para evitar que haja uma outra colisão entre vários terminais que estejam a concorrer entre si [10]. Assim, para resolver os problemas de sucessivas colisões, o protocolo utiliza um mecanismo de *backoff* [17].

O protocolo ALOHA com ACK foi testado no simulador apresentando melhores resultados que o ALOHA com e sem escuta. Tais resultados serão analisados em detalhe no capítulo 5. No entanto, apesar dessa melhoria, o protocolo ALOHA com ACK não conseguiu resolver todos os problemas de colisões nas redes acústicas submarinas.

Para resolver essas lacunas e ineficiências dos protocolos anteriores propôs-se a implementação do protocolo de acesso ao meio DACAP incluindo propostas para melhorar o seu funcionamento e a sua complexidade em ambientes acústicos submarinos. Nas próximas subsecções serão apresentadas o algoritmo do protocolo DACAP e várias etapas seguidas durante o processo de implementação do referido protocolo.

### 4.3.2. Algoritmo do protocolo DACAP

Como já foi referido o protocolo DACAP utiliza um mecanismo para resolver problemas de colisões nas redes acústicas submarinas. Este método proposto utiliza um algoritmo que é decomponível nos seguintes passos:

- i) Quando um nó tiver um pacote de dados para enviar, primeiro ele verifica o canal. Se o canal estiver inativo ele transmite um RTS. Caso contrário se o canal estiver ocupado, o emissor calcula um tempo de *backoff* e após este tempo verifica o canal novamente. Ao receber corretamente um pacote RTS, o nó recetor responde enviando um pacote CTS e fica à espera de um pacote de dados. Se enquanto espera por um pacote de dados, um nó de destino ouvir um pacote de controlo destinado a um outro nó, ele envia um pequeno pacote de aviso (*Warning*) para o emissor a quem tinha enviado um CTS para alertá-lo sobre possíveis interferências que possam afetar a próxima comunicação.
- ii) Ao receber um pacote de CTS, o nó emissor fica à espera durante um tempo *Twarning* antes de transmitir o pacote de dados. O valor de *Twarning* é definido como o tempo

mínimo que permite que os nós vizinhos não colidam. Se enquanto espera por um CTS o emissor ouvir um pacote de controlo, ele adia a transmissão de dados. Ele também adia a transmissão de dados se durante o período de espera  $T_{warning}$  receber um pacote de aviso do nó destino ou ouvir um pacote de controlo de outros nós. Neste caso, o emissor calcula o tempo de *backoff* e faz nova tentativa (para um número predeterminado de vezes). O nó de destino que enviou um pacote de *Warning* não se sabe se este pacote foi recebido pelo nó emissor a tempo de voltar a adiar a transmissão de dados e continuar a escutar o canal. Ainda assim, o pacote de dados pode ser recebido corretamente.

- iii) Ao receber o pacote de dados corretamente o nó recetor transmite um pacote de ACK ao nó emissor. Se nenhum ACK for recebido após transmissão de dados dentro de um tempo especificado o emissor considera a que a transmissão foi mal sucedida. O mesmo acontece quando o emissor estiver à espera de um ACK e receber um RTS, um CTS ou um pacote de dados de outros nós. Neste caso as potenciais interferências serão bloqueadas com esquemas RTS/CTS.

Para implementar o algoritmo DACAP no simulador de rede foram consideradas várias etapas a ter em conta na camada MAC. Essas etapas foram divididas em 6 módulos como é apresentado na figura 4.3. Esses módulos são: 1) Transmissão, 2) Receção, 3) Gestão do estado do canal, 4) Gestão estado do *backoff*, 5) Gestão do estado de transmissão e 6) Gestão do estado de receção. Como se pode observar na figura 4.3 as linhas a cheio na figura mostram os caminhos de passagem de pacote de dados e pacotes de controlo enquanto que as linhas a tracejado indicam as interfaces de sinalização ativa entre os módulos.



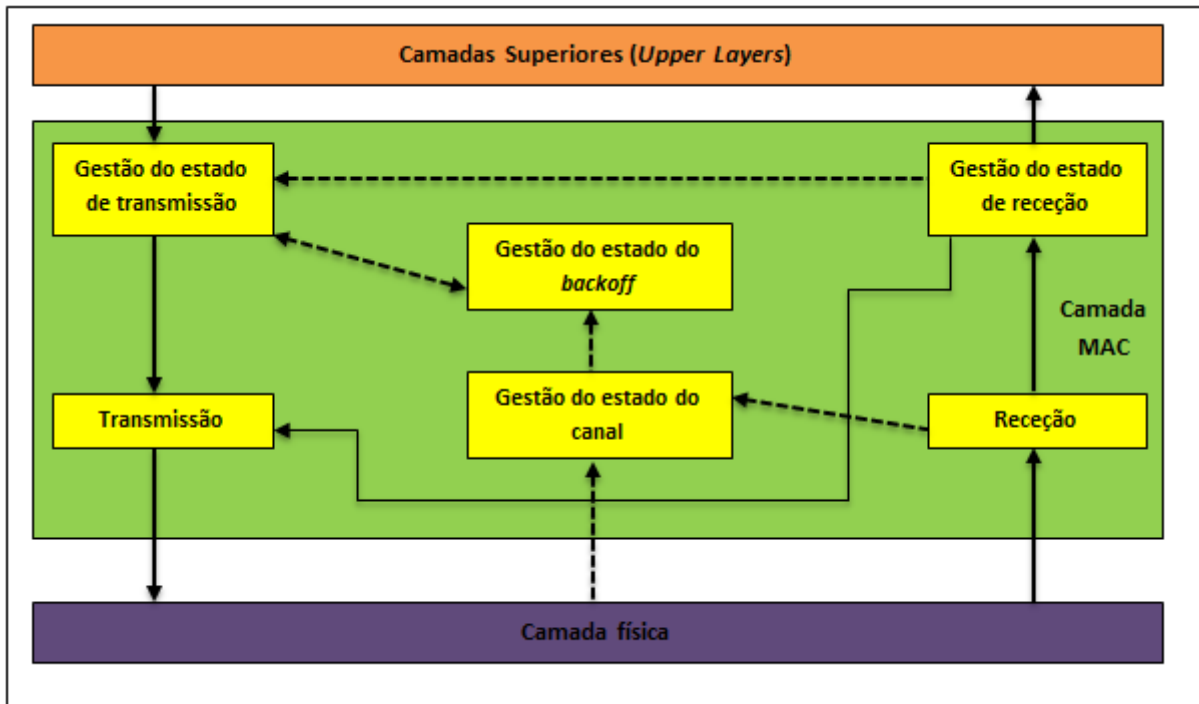


Figura 4.3: Arquitetura da camada MAC utilizada para implementação do DACAP.

### 4.3.2.1. Módulo de transmissão

O módulo de transmissão possui uma interface para camada física. Este módulo passa todos os tipos de tramas para camada física. O módulo de transmissão aceita pacotes de: RTS e dados a partir do módulo de gestão do estado de transmissão; e pacotes de ACK e CTS a partir do módulo de gestão do estado de receção. Quando são incorporadas outras características de gestão das camadas superiores do MAC, este módulo também irá aceitar tramas a partir dos novos módulos.

O módulo de transmissão possui uma máquina de estados simples, que consiste em 4 estados: TX RTS, TX CTS, TX DATA e TX ACK.

O estado TX RTS é responsável pela transmissão do pacote RTS enquanto que os outros estados TX CTS, TX DATA e TX ACK são responsáveis pelas transmissões dos seus respetivos pacotes.

### 4.3.2.2. Módulo de recepção

O módulo de recepção completa todo o processo de recepção de um pacote de entrada, que é iniciado na camada física. É utilizado para filtragem de pacotes recebidos com êxito antes de serem passados para outros módulos. O módulo de recepção faz uma verificação da trama em cima da camada física para ver se um pacote foi recebido com sucesso. Isto é, se um nó receber um pacote com erro o módulo de recepção informará o módulo de gestão do estado do canal que houve um erro na verificação da trama. Caso contrário se um pacote for recebido com sucesso o módulo de recepção faz uma filtragem de todos os endereços dos pacotes recebidos com sucesso e entra no módulo de gestão do estado de recepção apropriado que se enquadra esse nó. Se algum pacote não passar no processo de filtragem de endereços será examinado para ver se ele contém um NAV (*Network Allocation Vector*) antes de ser descartado. Os pacotes de RTS, CTS e às vezes os pacotes de dados contêm informações do NAV. Se for encontrado, o módulo de recepção passa o valor do NAV para o módulo gestão do estado do canal.

O módulo de recepção também possui uma máquina de estados simples. Estes estados são: RX RTS, RX CTS, RX DATA e RX ACK. O estado RX RTS é responsável pela recepção de um pacote RTS. Os outros estados RX CTS, RX DATA e RX ACK são responsáveis pela recepção dos seus respectivos pacotes.

### 4.3.2.3. Módulo gestão do estado do canal

O módulo gestão do estado do canal é responsável por manter o estado do suporte físico para o mecanismo de acesso ao meio DACAP.

O módulo gestão do estado do canal depende da camada física para atualizar o estado físico do canal acústico. Ele espera que a camada física sinalize que está ocupado, quando a força total do sinal recebido aumenta acima do limite da detecção da portadora do sinal no canal acústico ou quando a camada física está a transmitir. Da mesma forma o módulo gestão do estado do canal espera que a camada física sinalize canal disponível quando as duas condições terminarem de acontecer.

O módulo gestão do estado do canal fica à espera da sinalização para atualizar a portadora do canal acústico a partir do módulo de recepção. O módulo gestão do estado do canal possui 2 estados: *State Idle* (Estado de repouso), *State Listen* (Estado de escuta).

No estado de repouso, todos os temporizadores estão inicialmente parados e só depois é reiniciada a sessão. No momento em que é reiniciada a sessão verifica-se o estado da fila do *buffer* de transmissão do pacote. Se a fila de transmissão do pacote estiver vazia atualiza o *State Listen* e depois entra no estado de escuta ao meio. Quando estiver no estado de escuta é calculado um tempo que depende do tempo de escuta. Esse tempo depois é escalonado pelo temporizador de escuta para verificar se esse tempo expirou ou ainda está ativo.

O módulo gestão do estado do canal informa o estado da camada física em resposta à consultas de qualquer outro módulo. Ou seja ele informa se está no estado de repouso (se estiver em repouso) ou no estado de escuta (caso esteja nesse estado). Ele também informa o módulo gestão do estado de receção quando for consultado, para auxiliar a decisão de envio de um pacote de controlo. O módulo gestão do estado do canal sinaliza ativamente o módulo gestão do estado de *backoff* sempre que ele se move para dentro ou fora do estado de repouso indicando as mudanças nos estados do canal. Por sua vez o módulo gestão do estado de *backoff* retoma ou interrompe o seu processo de *backoff* se respetivamente estiver num desses estados.

#### **4.3.2.4. Módulo gestão do estado do *backoff***

O módulo gestão do estado de *backoff* utiliza o temporizador de *backoff* para auxiliar o mecanismo de prevenção de colisões do protocolo DACAP. O módulo gestão do estado de *backoff* também auxilia o módulo gestão do estado de transmissão a executar tanto o *backoff* regular e o *backoff* pós-transmissão, mas não estará ciente da diferença entre os dois. A figura 4.4 apresenta uma visão simplificada de como tal é feito. O módulo gestão do estado de *backoff* é constituído por 3 estados como é apresentado na figura 4.5. Esses estados são: Estado de saída do *backoff* (*Exit Backoff*), Estado de pausa do *backoff* (*Backoff Pause*) e Estado de execução do *backoff* (*Backoff Running*). O módulo gestão do estado de *backoff* depende do módulo gestão do estado do canal para sinalizar o estado de deteção da portadora do canal. Quando o temporizador do *backoff* chega a zero o módulo gestão do estado de *backoff* muda de estado sinalizando o módulo de gestão do estado de transmissão.

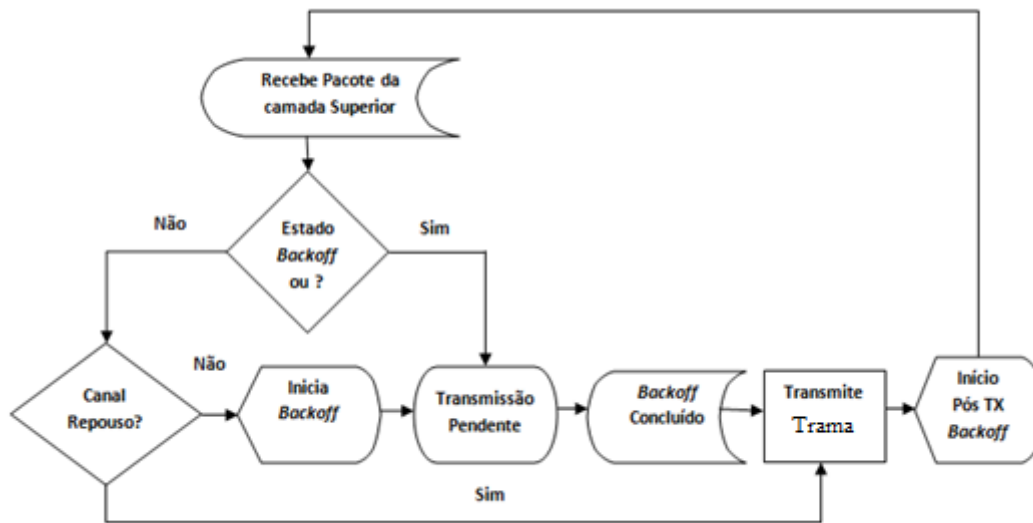


Figura 4.4: Utilização do módulo gestão do estado de *backoff* pelo módulo gestão do estado de transmissão.

Como se pode verificar na figura 4.5 o estado de saída do *backoff* só é realizado quando for sinalizado que o *backoff* foi concluído com sucesso pelo módulo gestão do estado de transmissão. O *backoff* deve ser iniciado se o canal estiver em repouso e em seguida deve-se inicializar o temporizador de *backoff*, após o que entra no estado de pausa do *backoff* se o canal continuar em repouso.

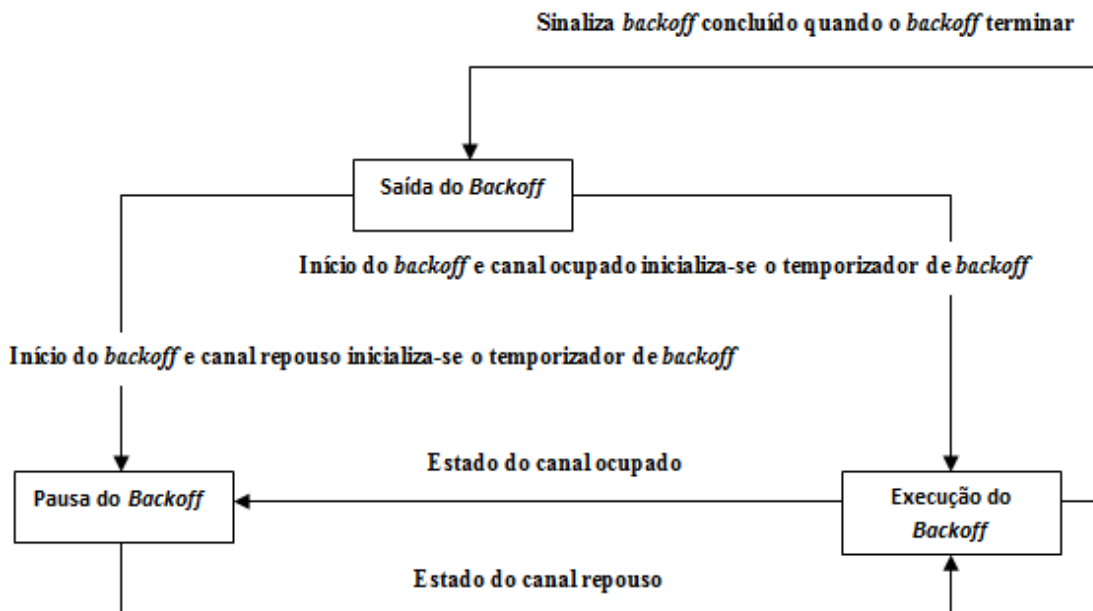


Figura 4.5: Módulo gestão do estado de *backoff*.

### 4.3.2.5. Módulo gestão do estado de transmissão

O módulo gestão do estado de transmissão gere todo o acesso ao meio dos pacotes transmitidos a partir da camada superior. O diagrama de estado do módulo gestão do estado de transmissão é apresentado na figura D.1 (ver anexo D). Este diagrama é dividido em 4 partes, consoante for necessário troca de mensagens RTS / CTS / DADOS / ACK. Pode-se observar que nesta fase não foi tido em conta o mecanismo de *Warning*, por isso não foi considerado nenhum estado para transmissão do pacote de *Warning*.

Se o módulo gestão do estado de transmissão não estiver no estado de repouso devido a um pacote vindo da camada superior, ele primeiro verifica o tamanho da fila do *buffer*, os pacotes que devem ser gerados e em seguida entra no estado de escuta ao meio (*State listen*). Enquanto estiver no estado de escuta ao meio é reescalonado o tempo de escuta através do temporizador de escuta (*Listen Timer*). Se o temporizador de escuta expirar, entra no estado de transmissão do pacote RTS (TX RTS) e quando terminar de escutar inicia a transmissão do pacote RTS. Caso contrário se o temporizador de escuta estiver activo volta novamente a escutar os pacotes de dados pendentes no canal e em seguida entra no estado de transmissão dos pacotes de dados (TX DATA). Quando estiver concluída a transmissão do RTS, o módulo gestão de estado de transmissão muda para o estado de espera do CTS (Wait CTS) e inicia um temporizador de CTS (CTS Timer). Se o módulo gestão do estado de receção sinalizar que não recebeu um pacote de CTS antes do tempo expirar, entra no estado de repouso e depois volta novamente para o estado de transmissão dos pacotes RTS pendentes. Este processo é repetido até que o número máximo de tentativas seja atingido. Caso contrário se o módulo gestão do estado de receção sinalizar que ouviu uma resposta a um RTS, (isto é, que recebeu um pacote CTS) o módulo gestão do estado de transmissão dá instrução ao módulo de transmissão para transmitir o pacote de dados. Após a transmissão do pacote de dados (TX DATA), o módulo gestão do estado de transmissão entra no estado de espera do pacote ACK (Wait ACK) e em seguida inicia o temporizador do ACK (ACK Timer). Se ele não conseguir ouvir nenhuma resposta confirmando a receção do pacote ACK pelo módulo gestão do estado de receção (antes do temporizador do ACK expirar) inicia o processo de *backoff* e logo de seguida volta a transmitir os respectivos pacotes RTS e dados pendentes. Quando entra no estado do *backoff* após o temporizador do ACK expirar é iniciado o reescalonamento do tempo do *backoff* pelo temporizador do *backoff* (*Backoff Timer*). Se o temporizador do *backoff* expirar e se o *backoff* for executado com sucesso o módulo gestão do estado do *backoff* sinaliza que o *backoff* foi concluído e sai do estado do *backoff* (*Exit backoff*) e volta para o estado de repouso. Se um nó enquanto estiver no estado de

repouso, ouvir um pacote CTS dirigido para outro nó, ele muda para o estado de espera do ACK (*Wait ACK*) e fica à espera do respetivo ACK. Se não for recebido nenhum pacote de ACK, significa que ocorreu um erro do pacote. O nó, permanece nesse estado até que a transmissão de dados (TX DATA) tenha sido concluída corretamente, isto é até que o módulo gestão do estado de receção sinalize que recebeu pacotes de dados com sucesso e entre no estado de transmissão do ACK (TX ACK). Assim que o módulo gestão do estado de receção sinalizar que o nó recebeu o pacote de ACK, ele volta novamente para o estado de repouso. Depois disso, se enquanto estiver no estado de repouso, houver um pacote na fila de espera isto é se a fila do *buffer* não estiver vazia volta para o estado de escuta ao meio (*State listen*) e reinicia novamente todo o processo de transmissão, caso contrário, volta para o estado de repouso.

#### 4.3.2.6. Módulo gestão do estado de receção

O módulo gestão do estado de receção recebe do módulo de receção todos os pacotes de controlo e pacotes de dados destinados ao respetivo nó. Ele sinaliza o módulo gestão do estado de transmissão, quando receber um CTS e um pacote de ACK. É responsável pelo tratamento das respostas dos CTS e respostas dos pacotes de ACK quando forem recebidos os pacotes de RTS e Dados. O módulo gestão do estado de receção também filtra todos os pacotes de dados antes de serem transmitidos para a camada superior. Como se pode verificar na figura D.2 (ver anexo D), o diagrama de estado do módulo gestão do estado de receção possui os seguintes estados: *RX Backoff*, *RX Idle*, *RX Listen*, *RX Wait CTS*, *RX Wait ACK*. O módulo gestão de estado de receção passa a maior parte do tempo no estado de repouso (*State Idle*) e fica à espera dos pacotes de controlo e de dados a partir do módulo de receção.

Quando iniciada a receção (START RX) começa-se por verificar o estado atual da transmissão entre os seguintes possíveis: estado escuta na receção (*RX Listen*); *backoff* na receção (*RX Backoff*); estado de repouso na receção (*RX Idle*); CTS na receção (*RX Wait CTS*); estado de espera do ACK na receção (*RX Wait ACK*). Quando um pacote RTS for recebido, o módulo gestão do estado de receção consulta o gestor do estado do canal para verificar o estado do NAV. Se a resposta indicar um NAV ativo, o pacote de RTS é simplesmente descartado. Caso contrário, o módulo gestão do estado de receção cria um pacote CTS e muda-se para o estado de espera do CTS na receção (*RX Wait CTS*) com um temporizador do CTS (*CTS Timer*). Na fase final de receção (END RX) dos pacotes RTS são verificados todos os endereços MAC dos pacotes RTS. Se o endereço MAC do terminal for igual ao recebido no pacote RTS, entra no estado de receção do pacote RTS (*RX RTS*)

e logo de seguida dá instrução ao módulo gestão do estado de transmissão para transmitir o pacote CTS, entrando no estado de espera do pacote CTS na receção (*RX Wait CTS*).

Se enquanto estiver à espera de receber um pacote de CTS for ouvido um pacote de controlo, ele adia a transmissão de dados. Quando for recebido um pacote CTS com o seu MAC como destino, muda para o estado de receção do pacote CTS (*RX CTS*) e logo de seguida o módulo gestão do estado de transmissão dá instrução ao módulo de transmissão para transmitir os pacotes de dados.

No recetor, se o pacote de dados for recebido (*RX DATA*), o módulo gestão do estado de receção inicia o processo ACK (de forma semelhante como ele lidou com as respostas do CTS). No entanto, ele não consulta o módulo gestão do estado do canal para verificar o estado do NAV. Isto é, se um pacote de CTS ou ACK chega ao destinatário, o módulo gestão do estado de receção simplesmente sinaliza o módulo gestão do estado de transmissão correspondente.

Na fase final da receção do ACK (*RX ACK*) será feita uma verificação do endereço MAC do pacote. Se o endereço MAC de destino no pacote de ACK for o do próprio nó, então a transmissão foi feita com sucesso. Volta-se nesse caso para o estado de repouso na transmissão. Caso não seja o temporizador do ACK (*ACK Timer*) entra no estado que verifica se o pacote de ACK expirou (*Check ACK Timer*). Se o temporizador de ACK estiver expirado entra-se no estado de *backoff*. Caso contrario se o temporizador estiver ativo volta novamente para o estado de repouso.

Na fase final da receção (*END RX*) do protocolo DACAP poderão surgir casos em que um nó receba um pacote que não lhe seja destinado (*RX Packet Not For Me*). Nesses casos serão verificados os estados de todos os temporizadores (*Backoof Timer*, *CTS Timer*, *Listen Timer* e *ACK Timer*) de forma a analisar onde ocorreu uma colisão no canal.

O módulo gestão do estado de receção envia todos pacotes de dados para a camada superior. Neste processo, será responsável por descartar todos pacotes de dados duplicados vindo do canal, que podem ser causados pelo mecanismo de retransmissão.

Nesta fase de implementação do algoritmo do protocolo DACAP baseado na arquitetura da camada MAC apresentada na figura 4.3 tendo em conta os 6 módulos apresentados anteriormente não foi considerado o mecanismo de *Warning*. Isto é não foi utilizada a transmissão do pacote de *Warning* como foi exemplificado no procedimento de implementação do protocolo DACAP apresentado no algoritmo 4.3.1. Apenas foram utilizados os esquemas de RTS/CTS/DATA/ACK referenciado no capítulo 3 da presente dissertação. Esta primeira implementação do protocolo DACAP foi designada *DACAP no Warning*. A diferença entre esse protocolo DACAP sem envio do pacote de *Warning* e o protocolo DACAP apresentado pelo algoritmo 4.3.1 consiste em que no protocolo

DACAP *no Warning* ao receber um CTS, o nó emissor envia imediatamente o pacote de dados. Por outro lado no protocolo DACAP com envio do pacote de *Warning* ao receber um CTS o nó emissor tem que esperar durante algum tempo antes de transmitir o pacote de dados.

Na próxima seção será apresentada a versão DACAP com *warning*.

## 4.4. Protocolo DACAP *with Warning*

A proposta seguinte consiste em utilizar um pacote de *Warning* para reduzir o número de colisões entre pacotes de controlo e dados, especialmente pensada para quando for aumentada a complexidade da rede no ambiente acústico submarino. Esta redução será feita introduzindo um tempo adicional (*Twarning*) de espera ao lado do emissor antes de transmitir os pacotes de dados, complementada pela introdução de um temporizador de espera (*Wait Timer*) na receção para quando receber um pacote de *Warning*. A utilização do pacote de *Warning* justifica-se sempre que os nós terminais devam evitar colisões, por exemplo por uma questão de gestão de energia, mesmo que tal possa acarretar uma diminuição de dados recebidos.

Foi utilizada a mesma arquitetura da camada MAC apresentada na figura 4.3 para implementar o mecanismo de *Warning*, partindo dos mesmos módulos utilizados na implementação do protocolo DACAP sem envio de *warning*. Foi acrescentado o estado TX *Warning* no módulo de transmissão e o estado RX *Warning* no módulo de receção. Não foi feita nenhuma alteração nos módulos gestores do estado do canal, *Backoff* e módulo gestão do estado de transmissão. Apenas foi alterada a implementação do módulo gestão do estado de receção. De seguida será apresentada de forma mais detalhada essas alterações apresentando o diagrama de estado do protocolo DACAP com envio do pacote de *Warning* na receção.

### 4.4.1. Módulo gestão do estado de receção do protocolo DACAP com *warning*

O diagrama de estado do módulo gestão do estado de receção do protocolo DACAP com envio de *Warning* é apresentado na figura D.3 (Ver anexo D). Este diagrama foi dividido em 5 partes, consoante for necessária a troca de RTS / CTS / DADOS / ACK/ WARNING. Pode-se observar que nesta fase foi incluído o mecanismo de *Warning*, tendo sido adicionado o estado de receção do pacote de *Warning* (RX *Warning*) neste diagrama. No diagrama de estado do módulo gestão do estado de receção, do protocolo DACAP com envio do pacote de *Warning*, foi considerado na fase



inicial da recepção (START RX) os mesmos estados utilizados no módulo de gestão do estado de recepção no protocolo DACAP sem envio de *Warning* acrescentado apenas um novo estado, designado de *RX Wait Warning*. O módulo gestão do estado de recepção do protocolo DACAP com envio de *warning* sinaliza o módulo de gestão do estado de transmissão, quando receber um pacote de CTS, *Warning* e pacote de ACK, assim como também filtra todos os pacotes de dados antes de serem transmitidos para a camada superior.

No instante em que se inicia a recepção (START RX) começa-se por verificar o estado atual na transmissão entre os seguintes: estado de espera do pacote de *Warning* na recepção (*RX Wait Warning*) ou estado de espera do CTS na recepção (*RX Wait CTS*). Na fase final de recepção (*END RX*) são verificados os endereços MAC de todos os pacotes. Mas antes dessa verificação começa-se por consultar o módulo gestão do estado do canal para verificar o estado do canal. Se o canal estiver inativo é dado uma instrução ao módulo gestão do estado de transmissão para transmitir o pacote de RTS. Caso contrário se o canal estiver ocupado, o módulo gestão do estado do *backoff* calcula um tempo de *backoff* e após o cálculo deste tempo verifica-se novamente o estado do canal.

Se for recebido um pacote de RTS (tendo como MAC de destino o seu próprio MAC), então o módulo gestão do estado de recepção dá instrução ao módulo de transmissão para transmitir o pacote de CTS e muda para o estado de espera do pacote de dados, inicializando um temporizador de espera (*Wait Timer*). Se enquanto estiver no estado de espera de dados for detetado um pacote de controlo para outro nó, será dada uma instrução ao módulo Gestão de estado recepção para transmitir o pacote de *warning* (*TX Warning*) de forma a alerta-lo sobre possíveis colisões que podem afetar a próxima comunicação.

Se enquanto estiver à espera de receber um pacote CTS, for ouvido um pacote de controlo, adia-se a transmissão de dados. Quando for recebido o pacote CTS (com o seu MAC como destino), então transita-se para o estado de recepção do pacote CTS (*RX CTS*) e logo de seguida fica-se à espera por um tempo *Twarning* antes de transmitir o pacote de dados através do temporizador de espera (*Wait Timer*). O tempo *Twarning* foi implementado com base na expressão do protocolo DACAP com ACK, apresentada no capítulo 3 (Ver seção 3.4.5.2) com o intuito de evitar colisões entre nós vizinhos. Caso não seja recebido o pacote CTS, será verificado se o temporizador do CTS expirou (*Check CTS Timer*). Se o temporizador CTS expirar, são atualizados todos os pacotes CTS expirados e muda-se para o estado de repouso. Caso contrário se o temporizador do CTS estiver ativo contínua-se à espera dos pacotes de CTS pendentes.

Caso seja recebido o pacote de *warning*, entra-se no estado de recepção do pacote de *warning* (RX *Warning*) e atrasa-se a transmissão dos pacotes de dados através do temporizador de espera (*Wait Timer*). Se o temporizador de espera estiver ativo fica-se à espera de todos os pacotes de *warning* pendentes. Caso o temporizador de espera expire, o módulo gestão de estado de recepção dá instrução ao módulo de transmissão para transmitir os pacotes de dados. Após transmissão dos pacotes de dados fica-se a aguardar pela chegada dos pacotes de dados na recepção (RX *Wait DATA*). Se o pacote de dados for recebido (RX *DATA*), o módulo gestão do estado de recepção inicia o processo ACK. Na fase final da recepção do ACK (RX ACK) será feita uma verificação do endereço MAC do pacote, assim como foi feito anteriormente com os outros pacotes. Se o endereço MAC de destino no pacote de ACK for o do próprio nó, então a transmissão é considerada bem-sucedida.

Os restantes mecanismos são iguais aos do protocolo DACAP sem *warning*.

Para testar a implementação do protocolo DACAP com envio do pacote de *warning* foi proposto um diagrama que verifica a sequência dos pacotes para eventos TX/RX no simulador de rede WOSS de forma a poder descobrir onde poderá ocorrer uma colisão. Com este diagrama (ver figura 4.6) pode-se verificar a sequência de todos os pacotes e as camadas envolvidas na transmissão/recepção de cada um dos pacotes, de forma a poder identificar onde poderão ocorrer colisões entre nós. De seguida será apresentado esse diagrama exemplificando todos os eventos de transmissão e recepção do protocolo DACAP com envio de *warning* no simulador de rede.

#### **4.4.2. Diagrama de sequência do protocolo DACAP com *warning***

O diagrama de sequência dos pacotes para eventos TX/RX foi proposto como uma alternativa para testar a implementação do protocolo DACAP com envio de *warning*. Este diagrama utiliza os pacotes de sinalização, o RTS e o CTS, pacotes de dados, ACK e pacotes de *Warning* para comunicação entre os nós. Para além dos referidos pacotes o diagrama de sequência também apresenta as camadas física, MAC e superiores e os eventos de transmissão/recepção entre as camadas física para camada MAC e vice-versa. A figura 4.6 apresenta a estrutura de funcionamento do diagrama. Quando um nó A deseja enviar uma mensagem ao nó B, ele primeiro inicia o protocolo com o envio de um pacote RTS da camada MAC para camada física através da primitiva (MAC2PHYSTART TX) com informações sobre o tamanho do pacote, endereço MAC e o identificador do pacote (ID do pacote) que se pretende enviar ao nó B. Na fase final de transmissão

do pacote RTS da camada física para camada MAC (primitiva PHY2MACEND TX) o nó A fica à espera do pacote CTS. Na fase final da recepção do pacote RTS (utilizando a primitiva PHY2MACEND RX), se o nó B receber o referido pacote, ele responde transmitindo um CTS da camada MAC para física (MAC2PHYSTART TX) para o nó A, que também contém informações sobre o seu endereço MAC, ID do pacote e o tamanho do pacote. No fim da transmissão do pacote CTS (PHY2MACEND TX) o nó B fica a aguardar por um pacote de dados. Se enquanto espera por um pacote de dados, o nó B ouvir um pacote de controlo destinado a um outro nó, por exemplo um nó C, ele envia o pacote de *warning* da camada MAC para camada física (MAC2PHYSTART TX) para o nó A de forma a alertá-lo sobre uma possível colisão que possa afetar a próxima comunicação.

Ao receber o pacote CTS da camada física para MAC (PHY2MACEND RX) o nó A fica à espera durante um tempo  $T_{warning}$  antes de transmitir o pacote de dados para o nó B. Esse tempo de espera  $T_{warning}$  definido no código da implementação é que vai evitar a colisão entre os nós vizinhos. Se enquanto espera por um CTS, o nó A ouvir um pacote de controlo, ele adia a transmissão do pacote de dados da camada MAC para camada física (MAC2PHYSTART TX). O nó A também vai adiar a transmissão de dados se durante o período de espera  $t_{warning}$  ele receber um pacote de *warning* do nó B (através da camada física para camada MAC) ou ouvir um pacote de controlo de outros nós. O nó B transmite o pacote de *warning* da camada mas não irá saber se este pacote foi recebido pelo nó A a tempo de voltar a adiar a transmissão de dados, porque o pacote de dados pode ter sido recebido corretamente. Na fase final de transmissão dos pacotes de dados o nó A fica à espera do pacote de ACK. Ao receber o pacote de dados corretamente da camada o nó B irá transmitir um pacote de ACK da camada MAC para camada física para o nó A. Se nenhum ACK for recebido após transmissão de dados dentro do tempo  $t_{warning}$  especificado o nó A considera que a houve um erro na transmissão. O mesmo acontece durante o periodo em que o nó A estiver a espera de um ACK e receber um RTS, um CTS ou um pacote de dados de outros nós. Neste caso as potenciais colisões que possam surgir serão evitadas com o envio do pacote de *warning*.

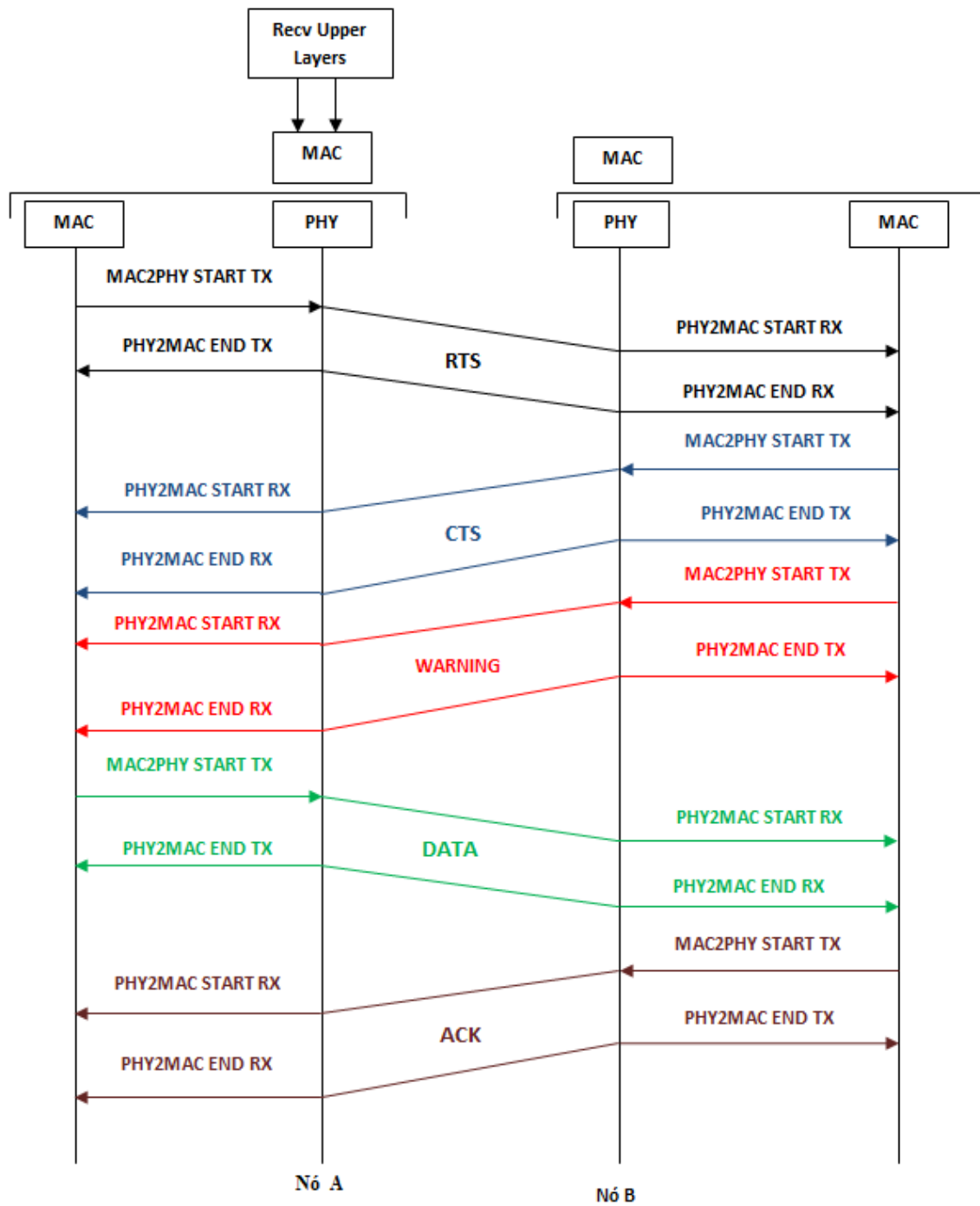


Figura 4.6: Diagrama de sequência de pacotes e primitivas do protocolo DACAP com *warning*.

## 4.5. Conclusões do capítulo

Neste capítulo foi apresentada uma visão geral sobre a implementação do mecanismo de acesso ao meio DACAP. Neste sentido primeiro fez-se um estudo do funcionamento da camada MAC no simulador de rede baseando no diagrama de sequência dos pacotes para eventos transmissão/recepção. De seguida foi apresentada uma arquitetura da camada MAC utilizada para implementação do algoritmo do protocolo DACAP. Foi dividida a implementação do referido protocolo em duas fases. Na primeira fase de implementação do protocolo DACAP não foi tida em consideração o mecanismo de *warning*. Foi feita a descrição dos seguintes módulos: Transmissão, recepção, módulo gestão do estado do canal, módulo gestão do estado do *backoff*, módulo gestão do estado transmissão e módulo gestão do estado de recepção. Na fase seguinte foi proposto implementação do mecanismo de *warning* no protocolo DACAP, como alternativa para melhorar a primeira implementação do protocolo, alterando o módulo de gestão de estado de recepção.

No capítulo 5, apresentaremos os resultados das simulações das duas implementações do protocolo DACAP apresentado resultados comparativos entre os diversos mecanismos MAC existentes no simulador de rede.



## Capítulo 5

### Simulações

Neste capítulo pretendem-se analisar os resultados da simulação dos protocolos de controlo de acesso ao meio (MAC) ALOHA com ACK, ALOHA sem ACK, ALOHANEW sem ACK, CSMA, DACAP com *Warning* e DACAP sem *Warning*, com o objetivo de determinar qual deles apresenta melhor desempenho nas redes acústicas submarinas. O estudo basear-se-á numa análise comparativa de desempenho desses mecanismos em vários cenários, de forma a determinar qual deles se comporta melhor em determinadas condições de rede.

Numa primeira fase, antes de obter os resultados, foram definidos os seguintes passos para analisar o desempenho desses seis protocolos de camada MAC para redes acústicas submarinas no simulador de rede WOSS:

- 1- Definição de uma estrutura no simulador de rede de forma a perceber as camadas físicas e de rede para comunicações acústicas submarinas;
- 2- Definição e análise dos seis protocolos de acesso ao meio a utilizar na simulação;
- 3- Definição dos parâmetros para análise do desempenho;
- 4- Definição de um cenário da rede e respetivos parâmetros físicos e de rede;
- 5- Análise dos resultados obtidos de forma a comparar o desempenho dos protocolos para vários cenários da rede de acordo com os parâmetros definidos.

No resto deste capítulo esses passos serão descritos em várias seções.

#### 5.1. Estrutura da simulação no WOSS

A figura 5.1 apresenta as camadas envolvidas na simulação efetuada. Nessa estrutura da simulação foi definido o canal acústico submarino no simulador de rede WOSS. Esse modelo do canal acústico foi integrado no simulador de rede acústico submarino com uma reprodução mais realista

de propagação acústica através do modelo de propagação do *software Bellhop ray tracing* [36]. O modelo de propagação *Bellhop* agrega informações sobre o perfil de velocidade do som (i.e., a variação da velocidade do som com a profundidade, o que influencia a quantidade de flexão incorridos pelas ondas), assim como dados de batimetria e informações sobre os sedimentos do fundo, ambas necessárias para caracterizar as reflexões do fundo [36]. Esses recursos do simulador de rede serão interligados com bases de dados livres disponíveis na *Internet*, como o WOA (Base de Dados Mundial dos Oceanos) [37], GEBCO (Quadro Batimétrico Geral dos Oceanos) [39] e *National Geophysical Data Center Deck41* (Base de dados do tipo de sedimentos de fundo oceânico) [40] de forma a proporcionar ao modelo da comunicação acústica submarina maior fidelidade às condições ambientais.

Quando se utiliza o *Bellhop* o utilizador apenas tem que especificar as coordenadas geográficas de onde a simulação deve ser feita e deixar o simulador executar o *Bellhop*. Os efeitos de propagação estimados para cada par de nós da rede incluindo todos os cálculos das perdas de caminho acústico de cada par origem-destino de um determinado local, bem como a interferência, são repetidos se os nós se movimentarem.

A figura 5.1 apresenta o modelo protocolar considerado nos nós. A camada física foi definida com base no sistema de modulação BPSK, suportando vários protocolos de acesso ao meio para redes acústicas submarinas. Para tal consideraram-se diversos nós da rede a tentarem enviar tráfego do tipo de débito constante (CBR) para o qual se definiu o intervalo de geração de pacotes e tamanho.

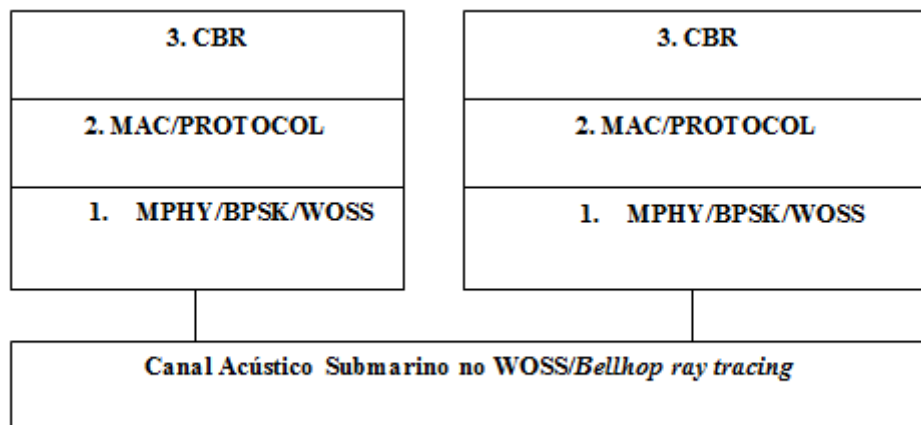


Figura 5.1: Estrutura da simulação no simulador de rede acústico WOSS.



Depois da estruturação da simulação no simulador de rede acústico WOSS foram definidos alguns cenários durante o processo de simulação, de forma a avaliar os resultados comparativos dos protocolos que serão apresentadas nas próximas seções.

## 5.2. Descrição dos protocolos utilizados na simulação

Nesta seção apresenta-se uma breve descrição dos seis protocolos de acesso ao meio para redes acústicas submarinas (ALOHA com ACK, ALOHA sem ACK, ALOHANEW sem ACK, CSMA, DACAP com *Warning* e DACAP sem *Warning*), utilizados nas simulações.

No protocolo ALOHA sem ACK [17], não é utilizado o pacote de ACK, nem é considerado o estado do canal antes da transmissão de um pacote de dados. Quando um emissor tiver um pacote de dados para enviar, ele envia imediatamente o pacote ao recetor. Não utiliza mecanismo de *backoff*, nem utiliza os pacotes de sinalização RTS, CTS e pacotes de *warning* para resolver eventuais problemas de colisões que possam surgir nas redes acústicas submarinas.

O protocolo ALOHANEW sem ACK, também não utiliza o pacote de ACK na retransmissão. O protocolo ALOHANEW foi implementado no simulador de rede, alterando o código de implementação de forma a escutar o meio. Este protocolo também não utiliza o mecanismo de *backoff*, *warning*, nem pacotes de sinalização.

No protocolo ALOHA com ACK, é utilizado o pacote de ACK quando o recetor receber corretamente um pacote de dados do emissor confirmando que a transmissão foi bem-sucedida. Também utiliza o mecanismo de *backoff* para resolver o problema de colisão em casos onde o emissor não consegue receber um ACK num intervalo de tempo predefinido. Nestes casos admite-se ter havido perda de pacote devido a uma colisão no canal. O protocolo ALOHA com ACK efectua escuta ao meio mas não utiliza os pacotes de sinalização RTS, CTS e mecanismo de *warning*.

A descrição do protocolo CSMA e o DACAP foram apresentados no (capítulo 3) da presente dissertação, enquanto que os detalhes da implementação e descrição do protocolo DACAP com *warning* foi apresentado no capítulo anterior (capítulo 4).

Na tabela 5.1 é apresentado um breve resumo sobre a descrição desses protocolos de acesso ao meio utilizados na simulação.

Protocolo	Tem ACK/Retransmissão	Tem Mecanismo <i>Backoff</i>	Efectua escuta ao meio	Tem Mecanismo <i>Warning</i>
ALOHA sem ACK	Não	Não	Não	Não
ALOHA NEW sem ACK	Não	Não	Sim	Não
ALOHA com ACK	Sim	Sim	Sim	Não
CSMA	Sim	Sim	Sim	Não
DACAP sem <i>Warning</i>	Sim	Sim	Sim	Não
DACAP com <i>Warning</i>	Sim	Sim	Sim	Sim

Tabela 5.1: Tabela com pequena descrição dos protocolos utilizados na simulação

### 5.3. Definição dos parâmetros de avaliação de desempenho

Para avaliação dos resultados foram definidos os seguintes parâmetros de forma a verificar o desempenho dos protocolos de acesso ao meio nas redes acústicas submarinas: o *Average Throughput per Node* (Débito recebido por nó), *Normalized Throughput* (Throughput Normalizado), *Success Ratio* (Índice de Sucesso), *Application Success Ratio* (Índice Sucesso a nível de Aplicação), *Data Generation Rate per Node* (Débito de envio por nó) e o *Overhead* (O tempo gasto em mecanismos de controlo).

O *Average Throughput per Node* é definido como débito recebido em bps, isto é o número de bits de dados por segundo que consegue chegar ao destino pretendido, neste caso ao nó *Sink* [17].

O *Throughput* Normalizado é definido como o rácio entre o débito recebido versus débito de pacotes gerados, a nível de aplicação [49]. O *Throughput* Normalizado é determinado pela seguinte expressão:

$$\text{Normalized Throughput} = \frac{\text{Throughput}}{\text{pktsize} \times 8 / \text{cbrperiod}} \quad (5.1)$$

Onde *pktsize* é o tamanho do pacote em bytes e o *cbrperiod* é o período entre pacotes em milissegundos.

O Índice de Sucesso é definido como rácio entre número de pacotes recebidos corretamente pelo nó *Sink* versus número total de pacotes enviados, enquanto que o Índice Sucesso a nível de Aplicação é necessário a fim de compreender o nível de desempenho da rede percebido a partir de uma fonte de

informação que funcione nas camadas de rede superiores [17]. Em casos em que os protocolos MAC, a um nível mais baixo, não permitam a transmissão a um débito igual à da fonte, os dados da mesma acabam por permanecer no *buffer* ou fila de transmissão, podendo mesmo ser descartados sem chegarem a ser transmitidos [17]. O índice de sucesso a nível de aplicação é então definido como o rácio entre número de pacotes recebidos corretamente versus número total de pacotes gerados pela fonte [49].

O Índice de Sucesso e o Índice Sucesso a nível de Aplicação são determinados pelas seguintes expressões:

$$\mathbf{Success\ Ratio} = \frac{RxPkts}{TXPkts} \quad (5.2)$$

$$\mathbf{Application\ Sucess\ Ratio} = \frac{RxPkts}{stopTime/cbrperiod} \quad (5.3)$$

Onde *RxPkts* é o número de pacotes CBR recebidos e o *TXPkts* é o número de pacotes CBR enviados. Em que *stopTime* corresponde ao tempo total da simulação.

O débito de envio por nó é obtido pela divisão entre tamanho de pacotes CBR em bytes e o intervalo entre pacotes CBR [17], mais concretamente utilizando a seguinte expressão:

$$\mathbf{Data\ Generation\ Rate\ per\ Node} = \frac{pktsize \times 8}{cbrperiod} \quad (5.4)$$

O tempo gasto em mecanismos de controlo (ou *overhead*) mede o tempo que é dedicado à sinalização, i.e., à transmissão de pacotes de ACK, RTS, CTS e *Warning* (DACAP). No caso das versões com ACK tal não inclui a retransmissão de pacotes errados.

Com base nestes parâmetros, nas próximas seções serão apresentadas as análises comparativas dos resultados obtidos da simulação dos protocolos de acesso ao meio em redes acústicas submarinas.

## 5.4. Simulações efectuadas e resultados

Nesta seção apresenta-se os resultados comparativos dos seis protocolos de acesso ao meio referenciados anteriormente para vários cenários da rede analisando os parâmetros de desempenho citados na seção anterior de forma a determinar qual deles apresenta melhor desempenho nas redes acústicas submarinas.

Para teste e comparação dos protocolos de controlo de aceso ao meio consideraram-se vários cenários em que dois ou mais nós tentavam comunicar com um ou mais recetores. Nesses cenários

assumiu-se que os pacotes foram gerados de acordo com um processo de *Poisson* determinando o débito de transmissão da fonte, e considerando tamanhos de pacotes de 50 e 600 bytes. Os pacotes de sinalização (RTS, CTS, *Warning*) e os pacotes de ACK foram definidos como tendo um tamanho de 4 Bytes.

Nos referidos cenários também se assumiu que os nós da rede foram equipados com *hardware* de modem padrão, como o *modem* WHOI micromodem (ver capítulo 2 da presente dissertação), permitindo transmissões utilizando o sistema de modulação coerente BPSK em diferentes bandas, com um débito binário de 4800 bps e uma frequência da portadora de 11,5 kHz. Assumiu-se um período entre pacotes do fluxo CBR de 400 milissegundos e uma largura de banda do canal de 5kHz. O período entre geração de pacotes faz com que os nós emissores estejam constantemente a tentar enviar dados.

### 5.4.1. Análise comparativa dos protocolos MAC para 2 nós emissores e 1 recetor

#### 5.4.1.1. Arquitetura simétrica

Inicialmente começou-se por analisar os resultados obtidos do cenário apresentado na figura 5.2, com 2 nós emissores (nó 0 e nó 1) e um nó recetor (nó 2), designado de *Sink* e localizado no centro da área da rede em que os nós (0 e 1). A distância horizontal entre os nós emissores (nó 0 e o nó 1) foi definida como sendo 500 metros, sendo a distância entre o nó 0 e nó 2 e entre nó 1 e nó 2 de 258,3 metros, de acordo com o esquema da figura seguinte.

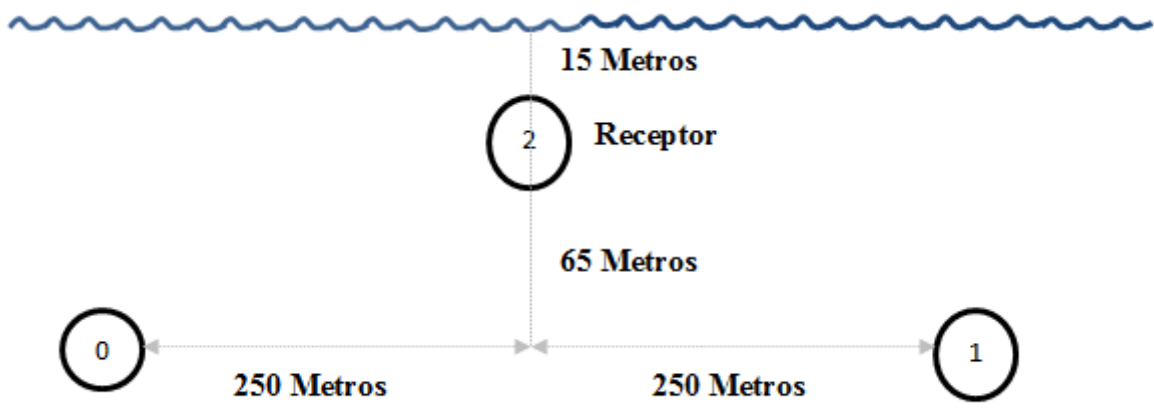


Figura 5.2: Cenário utilizado na simulação para análise comparativa dos protocolos para tamanhos de pacotes de 50 e 600 bytes.

Foram definidos os seguintes valores para a latitude, longitude e profundidade dos nós da rede como mostra a tabela 5.2.

Nó	Latitude	Longitude	Profundidade (metros)	Distância
0	42,59°N	10,125°E	-80 Metros	0 Metros
1	42,59°N	10,131°E	-80 Metros	500 Metros
2	42,59°N	10,128°E	-15 Metros	250 Metros

Tabela 5.2: Coordenadas e profundidades dos nós utilizados na simulação

Em seguida foram definidas as coordenadas geográficas como 42.59° N 10.125° E, para o SSP (Perfil de Velocidade do Som), impondo um conjunto de parâmetros ambientais para avaliação dos resultados.

Foi assumido que o ruído ambiental foi modelado, por meio de 4 fontes de ruído: turbulência, *shipping* (navegação), vento e o ruído térmico [13]. Esses factores de ruídos foram utilizados para determinar a densidade espectral de potência desses ruídos em dB re  $\mu\text{Pa}/\text{Hz}$ , em função da frequência (em kHz) pelas seguintes fórmulas empíricas [13]:

$$10 \log N_t(f) = 17 - 30 \log f \quad (5.5)$$

$$10 \log N_s(f) = 40 + 20(s - 0.5) + 26 \log(f) - 60 \log(f + 0.03) \quad (5.6)$$

$$10 \log N_w(f) = 50 + 7.5w^{\frac{1}{2}} + 20 \log f - 40 \log(f + 0.4) \quad (5.7)$$

$$10 \log N_{th}(f) = -15 + 20 \log f \quad (5.8)$$

O impacto de cada componente da densidade espectral de potência do ruído é diferente para frequências diferentes. Por exemplo, nas faixas de frequências encontradas para distâncias de transmissão de mais de 10 metros, a turbulência e as componentes de navegação têm muito pouco efeito, enquanto que o vento e o ruído térmico têm efeitos mais significativos [22]. No referido cenário foram definidos os seguintes valores para os parâmetros dos ruídos e salinidade da água: a navegação foi atribuída o valor de 0.5, o vento foi considerado um valor nulo (0 metros por segundos) e salinidade foi atribuída um valor de 35 ppm [22].

O perfil da velocidade do som utilizado na simulação é apresentado na figura 5.3 com um conjunto de velocidades de propagação do som para várias profundidades. Sendo um valor típico para

velocidades de propagação do som para comunicações acústicas submarinas um valor de aproximadamente 1500 metros por segundos.

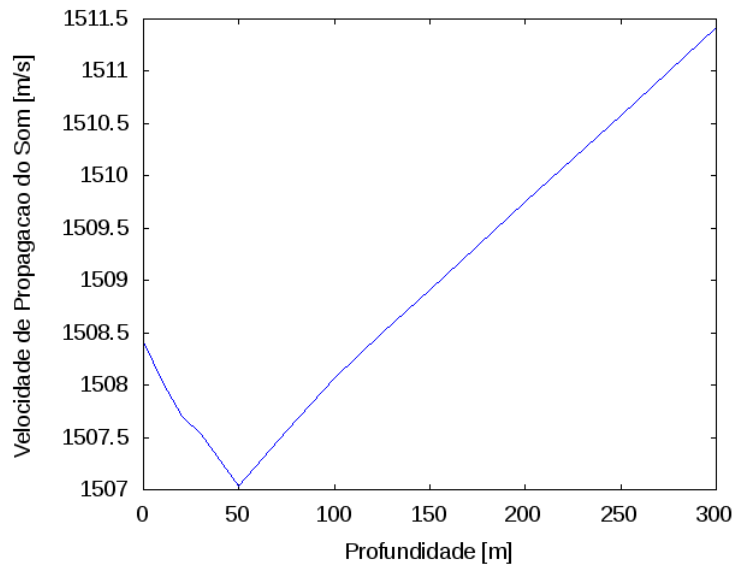


Figura 5.3: Velocidade de propagação do som em relação a profundidade, utilizado no cenário apresentado na figura 5.2.

O perfil batimétrico utilizado na simulação foi definido através do GEBCO que tem como objetivo proporcionar um quadro batimétrico geral dos oceanos livremente disponíveis na *Internet* [39]. No referido cenário foi utilizado um conjunto de grade de dados batimétricos usando o GEBCO *One Minute Grid* (grade de um minuto arco-resolução) [39]. Na figura 5.4 apresenta-se o mapa do perfil batimétrico utilizado na simulação.

O comando seguinte apresenta um exemplo de um script de definição do perfil batimétrico no simulador de rede:

```
SetCustomBathymetry $opt (start_lat) $opt (start_long) -500.0 4 0.0 100.0 500.0 200.0 1500.0
200.0 2500.0 100.0
```

Foram definidos os valores da latitude e longitude iniciais e um conjunto de pontos de intervalos de profundidade e distância. Em **\$opt (start\_lat) \$opt (start\_long)** são definidas as latitudes e longitudes iniciais e os valores **(-500.0 4 0.0 100.0 500.0 200.0 1500.0 200.0 2500.0 100.0)** correspondem a conjuntos de pontos que definem as profundidades e distâncias.

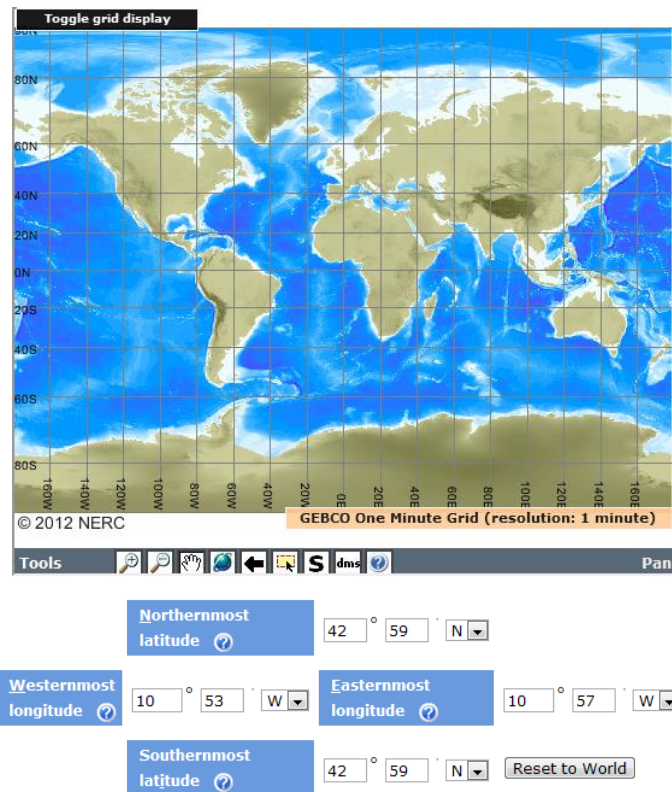


Figura 5.4: Mapa do perfil batimétrico utilizado no cenário apresentado na figura 5.2.

O perfil do sedimento do fundo utilizado na simulação foi definido através da *National Geophysical Data Center Deck41* uma base de dados sobre os tipos de sedimentos do fundo oceânico [40]. Os parâmetros do perfil do sedimento do fundo foram definidos através do seguinte comando no simulador de rede:

```
setCustomSediment "Test Sediment" 1560 200 1.5 0.9 0.8 1.0
```

Onde *Test Sediment* indica o nome do tipo de sedimento e os valores (1560 200 1.5 0.9 0.8 1.0) correspondem aos parâmetros do perfil do sedimento, nomeadamente:

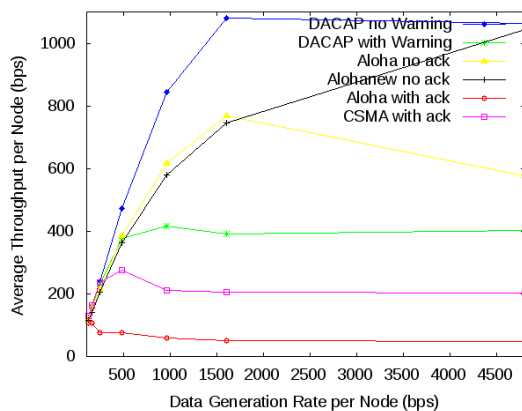
- Velocidade do comprimento de onda = 1560 m/s;
- *Shear wave velocity* = 200 m/s;
- Densidade do sedimento = 1,5 g/cm<sup>3</sup>;
- Atenuação do comprimento = 0,9 dB/comprimento de onda;
- *Shear wave attenuation* = 0,8 dB/comprimento de onda;
- Profundidade do fundo = 1,0 m.

Foram realizadas 40 simulações, em cada uma delas variando o número da *Seed* (semente) do gerador pseudo-aleatório, por forma a garantir pequenas variações nos períodos entre pacotes CBR.

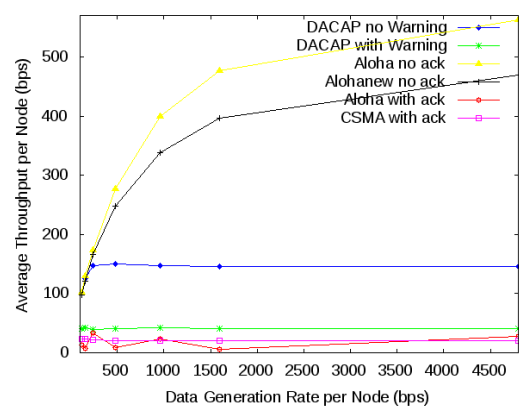
Seguidamente foram analisados os resultados dos débito recebido por nó em função do débito de envio por nó dos protocolos para tamanho do pacote de 600 e 50 bytes como mostra a figura 5.5.

Na figura 5.5 (a) pode-se verificar que para tamanhos de pacotes de 600 bytes, o protocolo DACAP sem *Warning* apresentou maior débito recebido por nó para elevados débitos de envio quando comparado com os outros protocolos. Em contrapartida, o protocolo ALOHA com ACK apresentou o pior resultado.

Para tamanhos de pacotes de 50 bytes, apresentado na figura 5.5 (b) o protocolo ALOHA sem ACK apresentou maiores débitos recebido por nó. A sobrecarga que os ACKs introduzem nos protocolos ALOHA com ACK e CSMA assim como a sobrecarga que os *handshakings* (RTS/CTS/*Warning*) introduzem nos protocolos DACAP, causam ineficiências quando são transmitidos pacotes pequenos, uma vez que os tempos de ida e volta são maiores do que os tempos de transmissão.



(a) Tamanho do pacote 600 bytes.



(b) Tamanho do pacote 50 bytes.

Figura 5.5: *Average Throughput per Node* em função do *Data Generation Rate per Node* dos protocolos para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.

A figura 5.6 apresenta os resultados do *Throughput* Normalizado em função do débito de envio por nó dos protocolos para tamanhos do pacote de 600 e 50 bytes. O *Throughput* Normalizado corresponde ao número de bits de dados por segundo que consegue chegar ao destino pretendido, dividido pelo número de bits de dados gerados.

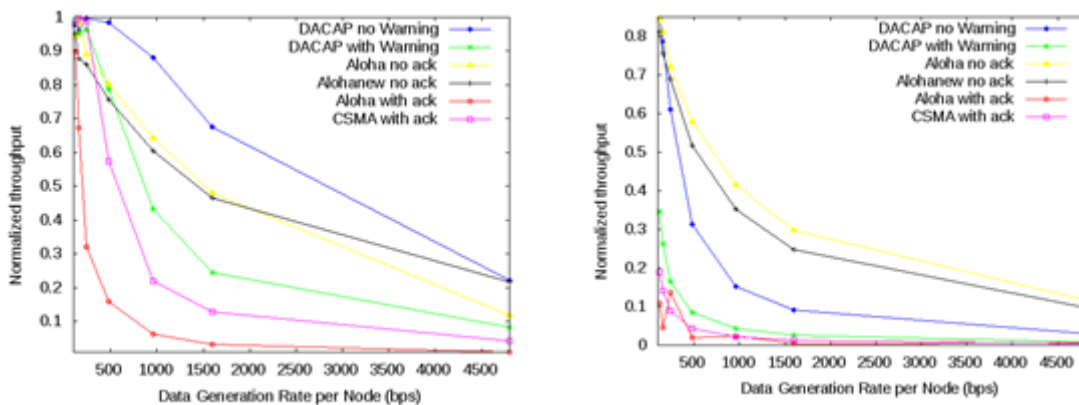
Como pode ser verificado na figura 5.6 (a), o protocolo DACAP sem *Warning* apresentou um melhor desempenho no *Throughput* normalizado para baixos valores de débito de envio por nó em relação aos outros protocolos para tamanho do pacote de 600 bytes, enquanto que o protocolo ALOHA com ACK apresentou o pior resultado. Nessa figura também se pode observar que o protocolo DACAP sem *Warning* alcança o *Throughput* máximo de 1, e além disso é capaz de manter esse nível até uma determinada taxa de tráfego gerado, após o qual o desempenho começa a



cair devido as perdas dos pacotes que são gerados. Este comportamento deve-se a maior nível de coordenação aplicado entre os nós no protocolo DACAP.

Por outro lado, para tamanhos de pacotes de 50 bytes o *Throughput* Normalizado do ALOHA sem ACK decresce mais lentamente que o dos outros protocolos, mesmo com baixo tráfego, devido à colisão cada vez mais frequente nos protocolos de acesso aleatório.

Em termos gerais, pode-se observar que em ambas as figuras 5.6 (a) e 5.6 (b) o *Throughput* normalizado dos protocolos ALOHA sem ACK (ALOHA sem ACK e ALOHANEW sem ACK) diminui mais levemente do que os outros protocolos com ACK (ALOHA com ACK e CSMA), graças à ausência do tempo de espera antes da transmissão dos pacotes de dados.



(a) Tamanho do pacote 600 bytes.

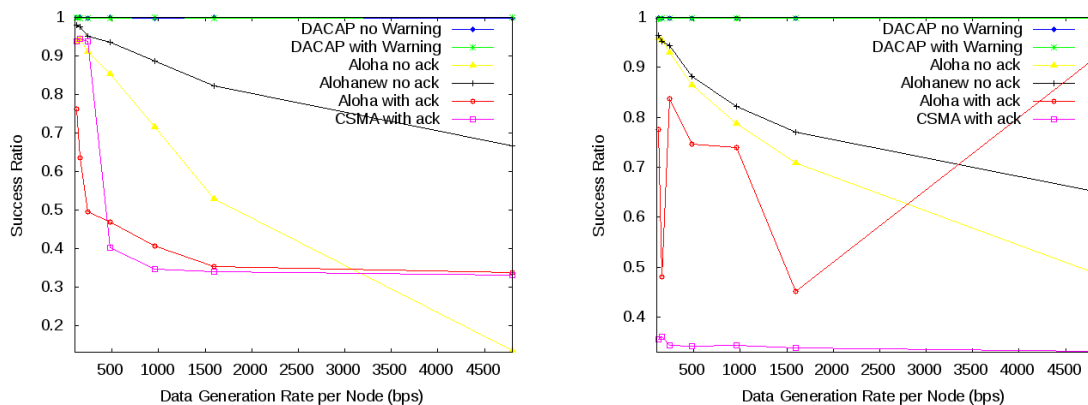
(b) Tamanho do pacote 50 bytes.

Figura 5.6: *Normalized Throughput* em função do *Data Generation Rate per Node* dos protocolos para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.

Na figura 5.7 apresentam-se os resultados do índice de sucesso dos protocolos para tamanhos de pacotes de 600 e 50 bytes respetivamente. Pode-se verificar nas figuras 5.7 (a) e 5.7 (b) que o índice de sucesso dos protocolos ALOHA em ambas as versões com e sem ACK e CSMA diminui consistentemente com o tráfego oferecido, o que traduz que são necessárias mais transmissões por cada receção bem-sucedida. Pode-se também observar que o protocolo CSMA apresentou um fraco desempenho no índice de sucesso mostrando que não se adequa aos atrasos envolvidos. Por outro lado, nos protocolos DACAP em ambas as versões com *Warning* e sem *Warning* o índice de sucesso está sempre próximo de 1. Se por um lado isso significa que todos os pacotes transmitidos são recebidos, isso não significa que todos os pacotes gerados chegam a ser transmitidos. Isso é mostrado pela figura 5.6 onde se apresenta o *Throughput* Normalizado, que tem como referência o débito da fonte, ou seja que é definido como o rácio entre o débito recebido versus débito de

pacotes gerados, a nível de aplicação. Nestes casos alguns desses pacotes são descartados devido às filas de espera estarem cheias ou completas.

Dos resultados apresentados nas figuras 5.7 (a) e 5.7 (b) pode-se concluir que os protocolos DACAPs utilizados na simulação apresentaram maior índice de sucesso em relação aos outros protocolos devido a ausência de colisões nos protocolos DACAPs devido ao mecanismo de *backoff* que obriga os nós a estar em longos períodos de silêncio, diminuindo o nível de interferência e as probabilidades de colisões nos protocolos DACAP.



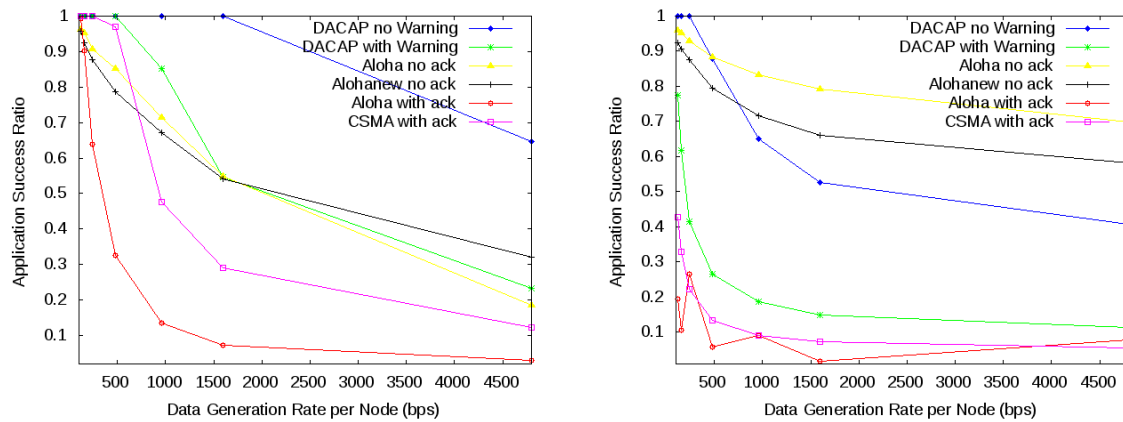
(a) Tamanho do pacote 600 bytes.

(b) Tamanho do pacote 50 bytes.

Figura 5.7: *Success Ratio* em função do *Data Generation Rate per Node* dos protocolos para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.

Os resultados do índice sucesso a nível de aplicação em função do débito de envio por nó dos protocolos para tamanho do pacote 600 e 50 bytes são apresentados nas figuras 5.8 (a) e 5.8 (b). Na figura 5.8 (a) pode-se verificar que, para tamanhos de pacotes mais elevados o índice de sucesso a nível de aplicação do protocolo DACAP sem *warning* mantém-se até um determinado nível de tráfego e diminui com o aumento do débito de envio por nó devido à saturação dos nós da rede, conduzindo a que quaisquer novos pacotes sejam descartados. No entanto pode-se observar nos resultados anteriores que o *Throughput* e o índice de sucesso consideram apenas os pacotes que são transmitidos, desprezando os pacotes gerados que são descartados devido aos atrasos excessivos nos nós. Ao contrário na figura 5.8 (b) para tamanho do pacote de 50 bytes pode-se verificar que todos os protocolos coordenados (DACAP sem *Warning*, DACAP com *Warning*) e os protocolos com ACK (ALOHA com ACK e CSMA) sofrem perdas de pacotes a nível de aplicação, não por causa de erros de transmissão, mas sim por causa de pacotes que são descartados devido as filas estarem cheias ou completas. O pior desempenho nesse sentido é apresentado pelo protocolo ALOHA com ACK, onde os efeitos de colisões e operações do protocolo determinam significativas perdas de pacotes. Os protocolos ALOHA sem ACK apresentaram melhores resultados a nível do índice

sucesso a nível de aplicação em relação aos outros protocolos para tamanhos de pacotes de 50 bytes.

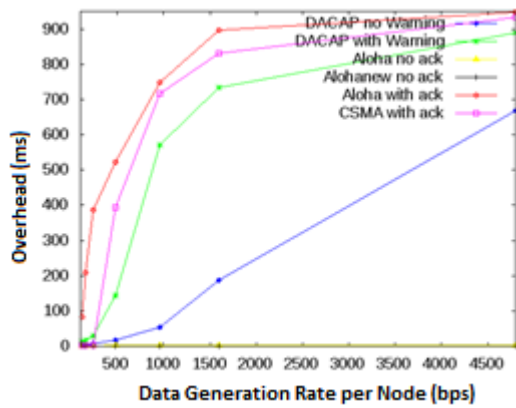


(a) Tamanho do pacote 600 bytes.

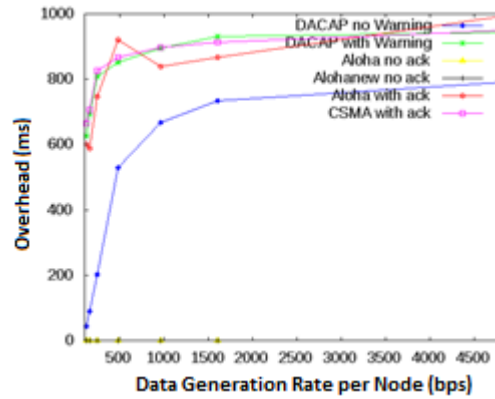
(b) Tamanho do pacote 50 bytes.

Figura 5.8: *Application Success Ratio* em função do *Data Generation Rate per Node* dos protocolos para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.

Em seguida, para finalizar a análise comparativa do desempenho dos protocolos para o cenário apresentado na figura 5.2 são apresentados os resultados dos tempos gastos em mecanismos de controlo (*Overhead*) em função do débito de envio por nó, para tamanhos do pacote de 600 e 50 bytes nas figuras 5.9 (a) e 5.9 (b). Como se pode verificar os protocolos ALOHA com ACK, CSMA e os protocolos de sinalização (DACAP sem *Warning*, DACAP com *Warning*) apresentaram maiores tempos gastos em mecanismos de controlo, para elevados débitos de envio por nó, em relação aos protocolos sem ACK (ALOHA sem ACK e ALOHANEW sem ACK). O maior tempo gasto em mecanismo de controlo desses protocolos deve-se a maior sobrecarga introduzido pelos pacotes de ACK e pacotes de sinalização (RTS / CTS/ *Warning*) aumentando o tempo de espera nas transmissões. Por outro lado, os protocolos ALOHA sem ACK e ALOHANEW sem ACK apresentaram um menor tempo gasto em mecanismo de controlo devido a sobrecarga nesses protocolos ser praticamente nulo diminuindo os tempos de ida e volta.



(a) Tamanho do pacote 600 bytes.



(b) Tamanho do pacote 50 bytes.

Figura 5.9: Tempo gasto em mecanismos de controlo (*Overhead*), em função do *Data Generation Rate per Node* dos protocolos para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.

Dos resultados obtidos do cenário apresentado na figura 5.2 utilizando a arquitetura simétrica para tamanhos de pacotes de 600 e 50 bytes conclui-se que as duas versões dos protocolos DACAP apresentaram melhores desempenhos no índice de sucesso quer para elevados ou baixos tráfegos. Também se constatou que apesar do índice de sucesso ser bastante bom nos protocolos DACAP, verificou-se que para tamanho do pacote de 50 bytes os protocolos DACAPs apresentaram um fraco desempenho do débito recebido por nó em relação aos protocolos ALOHA sem ACK e em especial o ALOHA sem ACK que apresentou um elevado *Throughput*. Em relação aos protocolos ALOHA com ACK e CSMA pode-se verificar que há uma degradação acentuada dos parâmetros testados à medida que se aumenta o débito de geração de tráfego nos nós.

### 5.4.1.2. Arquitetura assimétrica

De forma a testar se o desempenho desses protocolos varia consoante o cenário da rede que estiver a ser utilizado, foi alterado o cenário da figura 5.2, aproximando o nó 2 do nó 0, como é apresentado figura 5.10. No referido cenário foi definido a distância horizontal entre o nó 0 e nó 1 como 500 metros, sendo a distância entre o nó 0 e o nó 2 de 69,6 metros e entre nó 1 e nó 2 de 479,4 metros de distância. Tal como nos testes anteriores, foram considerados tamanhos de pacotes de 600 e 50 bytes. Também foram atribuídos os mesmos valores para os parâmetros ambientais e físicos, tais como velocidade de propagação do som, batimetria, tipo de sedimento do fundo e modulação como apresentado no cenário anterior para nossa avaliação de forma a comparar o desempenho dos protocolos de acesso meio em cenários diferentes.

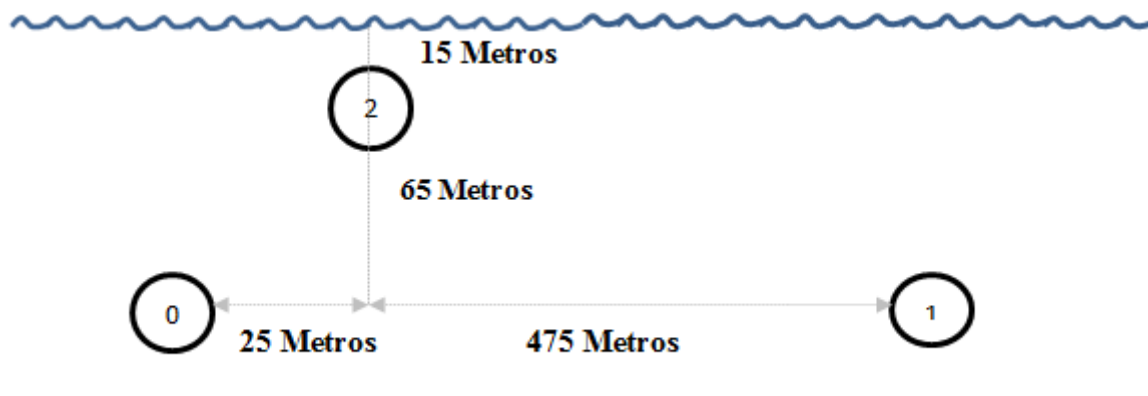
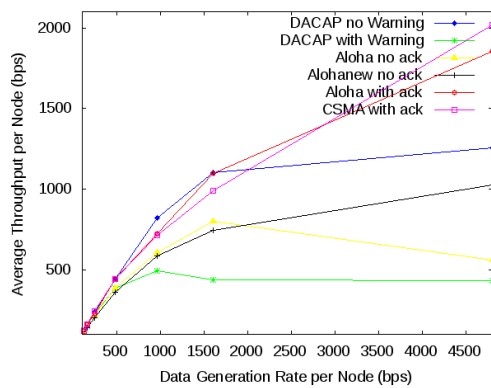


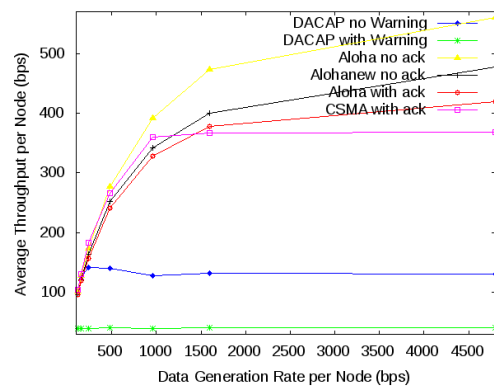
Figura 5.10: Cenário utilizado na simulação quando o nó 2 desloca para distância de 25 metros na horizontal em relação ao nó 0.

Seguidamente apresentam-se os resultados obtidos das 40 simulações efectuadas no cenário apresentado na figura 5.10, fazendo uma análise comparativa desses resultados com os resultados obtidos no cenário anterior para tamanhos de pacote de 600 e 50 bytes.

Como se pode verificar na figura 5.11 (a) o resultado do débito recebido por nó em função do débito de envio por nó dos protocolos CSMA e ALOHA com ACK para tamanho do pacote de 600 byte apresentaram melhorias em relação ao resultado do cenário anterior apresentado na figura 5.5 (a) onde os protocolos CSMA e ALOHA com ACK apresentavam piores resultados. Por outro lado quando se diminui o tamanho do pacote para 50 bytes pode-se verificar na figura 5.11 (b) que apesar de haver uma ligeira melhoria nos protocolos CSMA e ALOHA com ACK em relação ao resultado do cenário anterior apresentado na figura 5.5 (b) continua o protocolo ALOHA sem ACK a apresentar o melhor resultado a nível do débito recebido por nó. Também pode-se constatar na figura 5.11 (b) para tamanho do pacote de 50 bytes que os protocolos DACAPs utilizados na simulação apresentaram piores resultados a nível do débito recebido por nó devido à sobrecarga que o *handshaking* (RTS/CTS/Warning) introduz causando ineficiências quando os pacotes são pequenos. Tal deve-se ao facto do tempo de ida e volta ser maior do que os tempos de transmissão.



(a) Tamanho do pacote 600 bytes.

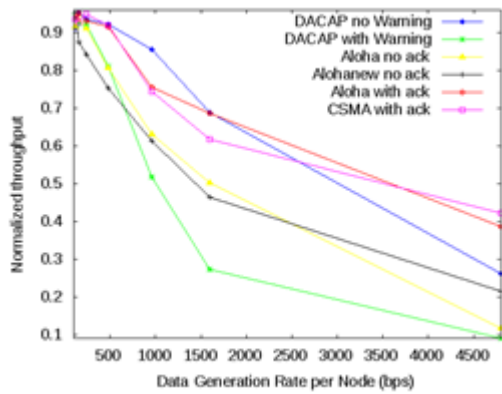


(b) Tamanho do pacote 50 bytes.

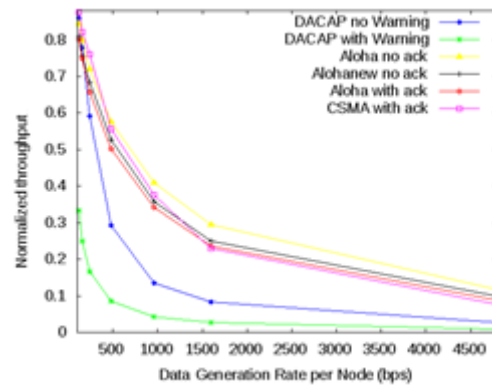
Figura 5.11 : *Average Throughput per Node* em função do *Data Generation Rate per Node* dos protocolos utilizando o cenário da figura 5.10 para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.

Os resultados do *Throughput* Normalizado em função do débito de envio por nó para tamanho de pacotes de 600 e 50 bytes respetivamente são apresentadas nas figuras 5.12 (a) e 5.12 (b). Como se pode observar na figura 5.12 (a) há uma grande quebra ou diminuição nos *Throughputs* normalizados de todos os protocolos com o aumento do débito de envio por nó para tamanho de pacotes de 600 bytes. Também na figura 5.12 (a) se pode verificar que os protocolos ALOHA com ACK e CSMA a partir de um determinado nível de débito de envio por nó apresentaram melhores resultados no *Throughput* Normalizado em relação aos outros protocolos, ao contrário dos resultados da figura 5.6 (a) onde o protocolo DACAP sem *warning* apresentava melhor resultado no *Throughput* Normalizado.

Em relação ao *Throughput* Normalizado representado na figura 5.12 (b), o protocolo ALOHA sem ACK apresentou o melhor resultado para tamanhos de pacotes de 50 bytes em relação aos outros protocolos quando aproximamos o nó 2 do nó emissor. Ao contrário do protocolo ALOHA sem ACK, o protocolo DACAP com *warning* apresentou o pior desempenho para tamanhos de pacotes de 50 bytes. Este fraco desempenho no *Throughput* Normalizado do protocolo DACAP com *warning* deve-se ao facto não haver um incremento no débito de emissão (e por conseguinte de receção) que responda ao débito de geração, acabando muitos dos pacotes gerados por ficar nas filas de espera de emissão, ou acabando por ser descartados.



(a) Tamanho do pacote 600 bytes.

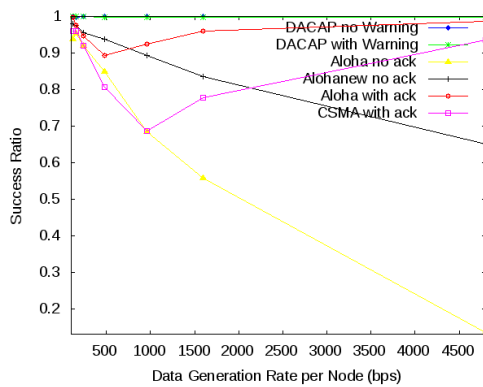


(b) Tamanho do pacote 50 bytes.

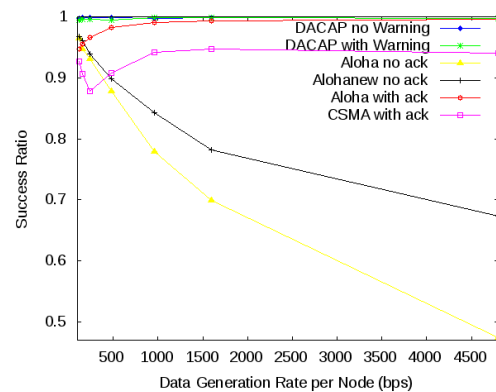
Figura 5.12: *Normalized Throughput* em função do *Data Generation Rate per Node* dos protocolos utilizando o cenário da figura 5.10 para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.

Nas figuras 5.13 (a) e 5.13 (b) são apresentadas os resultados do índice de sucesso dos protocolos em função do débito de envio por nó para tamanhos do pacote de 600 e 50 bytes. Dos resultados apresentados pode-se verificar que os protocolos DACAP com e sem *Warning* apresentaram melhores desempenhos nos índices de sucessos em relação aos outros protocolos atingindo um valor máximo muito próximo dos 100 %, em consequência da maior prevenção dos protocolos DACAPs em relação às colisões.

O protocolo ALOHA sem ACK apresentou um resultado muito fraco ao nível do índice de sucesso quando aproximamos o nó 2 em relação ao nó 0 para tamanhos de pacotes de 600 e 50 bytes, que compare com os resultados anteriores apresentados nas figuras 5.7 (a) e 5.7 (b), onde o protocolo CSMA apresentava o pior resultado.



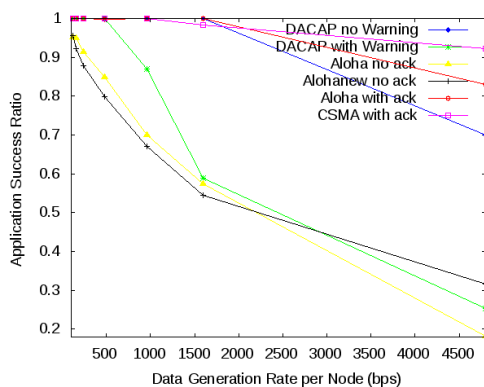
(a) Tamanho do pacote 600 bytes.



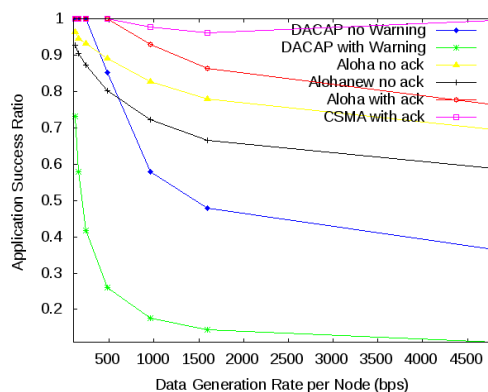
(b) Tamanho do pacote 50 bytes.

Figura 5.13 : *Success Ratio* em função do *Data Generation Rate per Node* dos protocolos utilizando o cenário da figura 5.10 para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.

Os resultados do índice sucesso a nível de aplicação em função do débito de envio por nó dos protocolos para tamanhos do pacote de 600 e 50 bytes são apresentadas nas figuras 5.14 (a) e 5.14 (b). Pode-se verificar que para tamanhos de pacote de 600 bytes o índice sucesso a nível de aplicação dos protocolos CSMA e ALOHA com ACK mantém-se até um determinado nível de débito de envio e depois começa a diminuir. Devido à saturação dos nós da rede quaisquer pacotes adicionais são descartados. No entanto pode-se observar nos resultados obtidos anteriormente, apresentados na figura 5.8 (a) para tamanho do pacote de 600 byte, que os resultados do índice de sucesso a nível de aplicação dos protocolos CSMA e ALOHA com ACK sofreram uma grande queda. Ao contrário, para tamanhos de pacote de 50 bytes (apresentado na figura 5.14 (b)) pode-se verificar que todos os protocolos com *handshake* (DACAP sem *Warning* e DACAP com *Warning*) apresentaram péssimos resultados a nível do índice sucesso a nível de aplicação porque sofreram perdas de pacotes ao nível de aplicação, não por causa dos erros de transmissão, mas sim por causa dos pacotes que são descartados devido as filas estarem cheias ou completas. O pior desempenho nesse sentido é apresentado pelo protocolo DACAP com *Warning*. Estes resultados no entanto seriam evitáveis se se utilizassem outros mecanismos de controlo de fluxo de nível de aplicação.



(a) Tamanho do pacote 600 bytes.



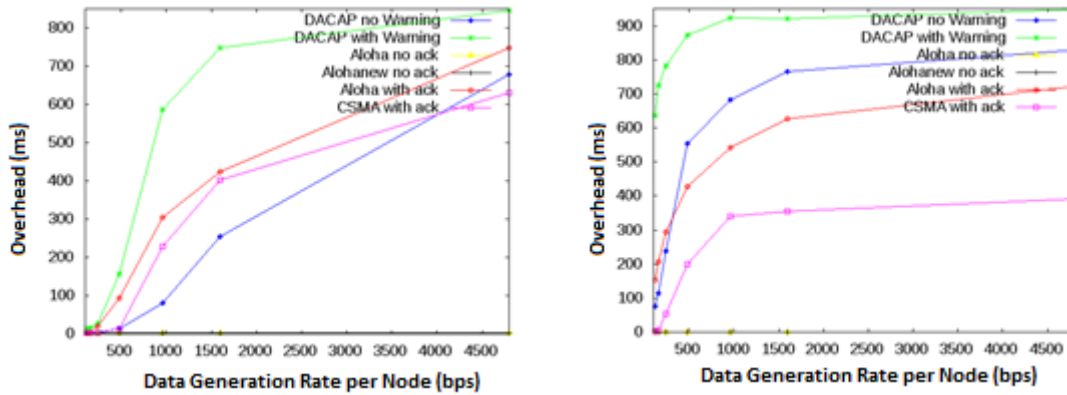
(b) Tamanho do pacote 50 bytes.

Figura 5.14: *Application Success Ratio* em função do *Data Generation Rate per Node* dos protocolos utilizando o cenário da figura 5.10 para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.

Em seguida foram apresentados os resultados dos tempos gastos em mecanismos de controlo (*Overhead*) em função dos débitos de envio por nó para tamanhos do pacote de 600 e 50 bytes nas figuras 5.15 (a) e 5.15 (b). Pode-se verificar nesses resultados obtidos que o protocolo DACAP com *Warning* apresentou um elevado tempo gasto em mecanismos de controlo para maiores débito de envio por nó em relação aos outros protocolos. O maior tempo gasto em mecanismos de controlo deve-se aos pacotes de sinalização (RTS/CTS/*Warning*) que provocam um aumento do tempo de espera nas transmissões. No sentido contrário, os protocolos ALOHA sem ACK e ALOHANEW



sem ACK apresentaram menores tempos gastos em mecanismos de controlo devido à ausência de sobrecarga nesses protocolos, diminuindo os tempos de ida e volta.



(a) Tamanho do pacote 600 bytes.

(b) Tamanho do pacote 50 bytes.

Figura 5.15: Tempo gasto em mecanismos de controlo (*Overhead*) em função do *Data Generation Rate per Node* dos protocolos utilizando o cenário da figura 5.10 para (a) Tamanho do pacote 600 bytes (b) Tamanho do pacote 50 bytes.

### 5.4.1.3. Sumário

Em termos globais considerando os dois cenários apresentados anteriormente com três nós na rede para tamanhos de pacotes de 600 e 50 bytes para os diversos parâmetros testados, pode-se concluir que os protocolos DACAP apresentaram melhores desempenhos a nível dos índices de sucessos dos pacotes, justificando pela menor probabilidade de colisões que induzem. Também se verificou que o pior desempenho no índice de sucesso dos pacotes depende do cenário que estiver a ser utilizado. Como se pode observar no cenário apresentado na figura 5.2 o protocolo CSMA apresentou um fraco desempenho quando o nó recetor (nó 2) se encontra no centro da rede, quer para tamanhos de pacote de 600 ou 50 bytes, enquanto que o protocolo ALOHA sem ACK apresentou o pior desempenho para a configuração assimétrica, em termos do índice de sucesso em função do débito de envio por nó. O protocolo DACAP, em particular o DACAP com *warning*, apesar de apresentar um bom desempenho no índice de sucesso, apresentou um fraco desempenho a nível do débito recebido versus débito de envio, assim como apresentou um elevado tempo gasto em mecanismos de controlo (*Overhead*) em relação aos protocolos ALOHA sem ACK e ALOHANEW sem ACK devido à sobrecarga que os pacotes de sinalização (RTS/CTS/*Warning*) introduzem aumentando o tempo de espera nas transmissões.

Os resultados obtidos aqui foram comparados com os resultados do artigo apresentado por Federico Guerra *et al.* [17], onde os autores fizeram uma análise comparativa de três protocolos de acesso ao

meio (DACAP, ALOHA e T-Lohi) para dois diferentes tipos de padrões de tráfego para geração de pacotes, isto é, quer de acordo com um tráfego de *Poisson* utilizando o parâmetro pacotes por segundo por nó e o tráfego orientado a eventos (*event-driven traffic*).

A conclusão que se pode tirar dos resultados do artigo em comparação com os nossos resultados foi que os protocolos com menores sobrecarga tendem a alcançar um maior *Throughput* em detrimento de um menor índice de sucesso, enquanto que os protocolos baseados em *handshakes* robustos, tais como o DACAP podem alavancar por usa vez maior organização nas transmissões, a fim de atingir um índice de sucesso muito elevado à custa de menor *Throughput*.

Em termos gerais, os resultados obtidos nas nossas simulações e no artigo anterior indicam que o melhor protocolo num contexto, não é necessariamente melhor noutro contexto. Também se concluiu que devem ser considerados outros efeitos tais como efeitos dos parâmetros ambientais e maior complexidade da rede com mais nós, de forma a verificar o comportamento desses protocolos em cenários diferentes. Esse será o âmbito das seções seguintes.

### **5.4.2. Análise comparativa dos protocolos MAC para 4 nós emissores e 1 recetor**

Nesta seção será apresentada a análise dos resultados obtidos dos protocolos de acesso ao meio para o cenário apresentado na figura 5.16. No referido cenário foi aumentada a complexidade da rede por incremento do número de nós da rede. Assim, consideraram-se 4 nós emissores e um nó recetor. Considerou-se que os nós emissores da rede pretendem transmitir tráfego do tipo CBR.

Fizeram-se duas análises comparativas, uma utilizando uma arquitetura simétrica em que o nó 4 foi o recetor (e os nós 0, 1, 2 e 3 emissores) e outra assimétrica considerando o nó 2 como recetor (com os nós 0, 1, 3 e 4 como emissores).

No referido cenário da simulação, assumiu-se que os nós da rede foram equipados com *hardware* de *modem* padrão, como o modem WHOI micromodem referenciado no cenário da seção anterior. Considerou-se o sistema de modulação coerente BPSK, com um débito binário de 4800 bps e uma frequência da portadora de 11,5 kHz. Assumiu-se nesse cenário apenas o tamanho do pacote de 600 bytes, com o período entre pacotes do fluxo CBR de 400 milissegundos e uma largura de banda do canal de 5kHz.

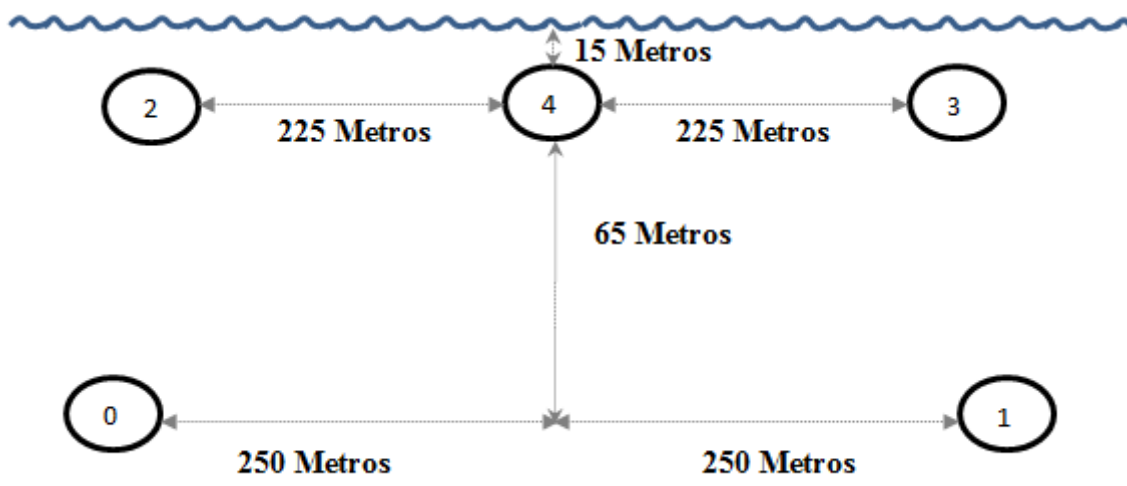


Figura 5.16: Cenário da simulação para análise comparativa dos protocolos considerando o nó 4 como nó recetor para tamanho do pacote de 600 bytes.

A distância horizontal entre o nó 0 e nó 1 foi definida como 500 metros, sendo que a distância horizontal entre o nó 2 e o nó 4 e entre nó 3 e nó 4 de 225 metros. A distância entre o nó 0 e o nó 4 e entre nó 1 e nó 4 de 258,3 metros. Foram definidos os valores para a latitude, longitude e profundidade dos nós da rede, de acordo a tabela 5.3.

Nó	Latitude	Longitude	Profundidade (metros)	Distância
0	42,59°N	10,125°E	-80 Metros	0 Metros
1	42,59°N	10,131°E	-80 Metros	500 Metros
2	42,59°N	10,125°E	-15 Metros	25 Metros
3	42,59°N	10,130°E	-15 Metros	475 Metros
4	42,59°N	10,128°E	-15 Metros	250 Metros

Tabela 5.3: Coordenadas e profundidades dos nós utilizados na simulação do cenário da figura 5.16

Em seguida foram definidas as coordenadas geográficas, o perfil de velocidade do som, assim como foram especificados um conjunto de parâmetros ambientais para nossa avaliação. O valor do perfil da velocidade do som foi obtido através de um ficheiro texto indicando os valores das latitudes e longitudes iniciais e um conjunto de velocidades de propagação do som para várias profundidades como é apresentado na figura 5.17. Foi considerado que o ruído ambiental foi modelado por meio de 4 fontes de ruído: turbulência, a navegação (*shipping*), vento e o ruído térmico. Foram atribuídos os seguintes valores para os parâmetros dos ruídos e salinidade da água: a navegação foi atribuída o valor de 1, o vento foi considerado um valor de 10 metros por segundos e salinidade foi atribuída um valor de 33 ppm.

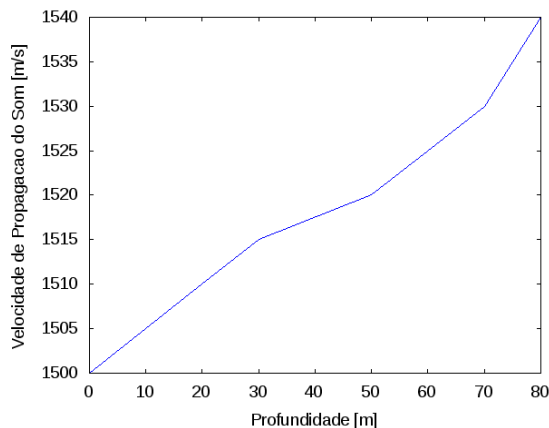


Figura 5.17: Velocidade de propagação do som em relação a profundidade utilizando o cenário da simulação apresentado na figura 5.16.

Diferente dos cenários apresentados anteriormente, o perfil batimétrico utilizado no cenário apresentado na figura 5.16, foi definido através do GEBCO\_08 *Grid* (grade de 30 arco-segundos de resolução). O mapa do perfil batimétrico utilizado nessa simulação é apresentado na figura 5.18.

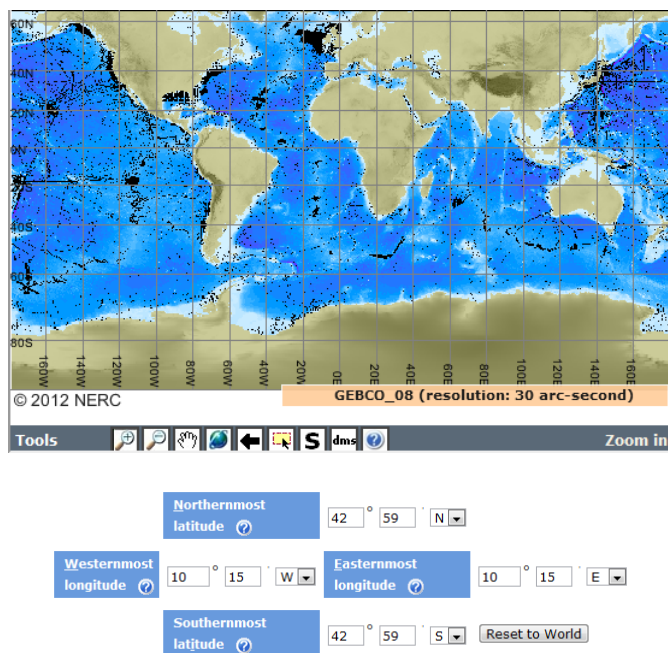


Figura 5.18: Mapa do perfil batimétrico utilizado no cenário apresentado na figura 5.16.

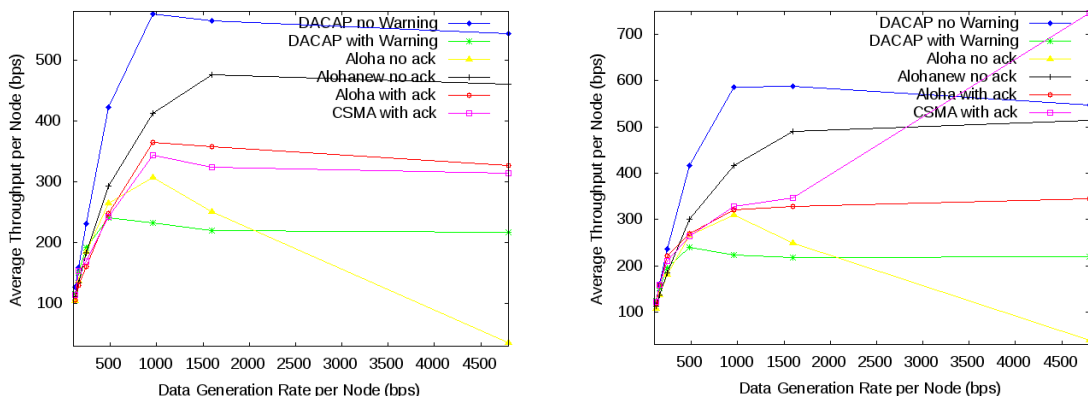
Foram definidos os seguintes parâmetros do perfil do sedimento do fundo:

- Velocidade do comprimento de onda = 1540 m/s;
- *Shear wave velocity* = 180 m/s;
- Densidade do sedimento = 1,0 g/cm<sup>3</sup>;
- Atenuação do comprimento = 0,7 dB/comprimento de onda;
- *Shear wave attenuation* = 0,6 dB/comprimento de onda;

- Profundidade do fundo = 1,5 m.

Seguidamente apresentam-se os resultados obtidos das 40 simulações efetuadas com o referido cenário, fazendo uma análise comparativa desses resultados utilizando as duas arquiteturas: uma arquitetura simétrica em que o nó 4 foi considerado como recetor e outra assimétrica considerando o nó 2 como recetor.

A figura 5.19 apresenta o resultado do débito recebido por nó em função do débito de envio por nó dos protocolos utilizando as duas arquiteturas. Como se pode verificar nesses resultados o protocolo DACAP sem *Warning* apresentou maiores débitos recebidos por nó, para elevados débitos de envio em relação aos outros protocolos, sendo que os protocolos DACAP com *warning* e ALOHA sem ACK a apresentarem piores desempenhos no débito recebido por nó.

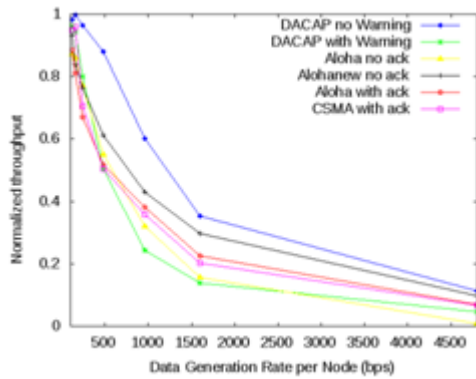


(a) Nó 4 recetor.

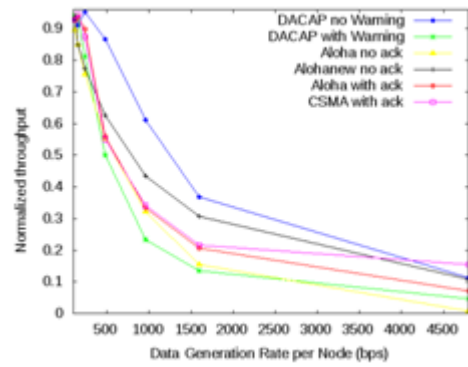
(b) Nó 2 recetor.

Figura 5.19: *Average Throughput per Node* em função do *Data Generation Rate per Node* dos protocolos considerando (a) Nó 4 como recetor (b) Nó 2 como recetor.

Na figura 5.20 apresenta-se o resultado do *Throughput* Normalizado em função do débito de envio por nó dos protocolos para as duas arquiteturas: (a) Nó 4 como recetor (b) Nó 2 como recetor. Pode-se verificar que há uma diminuição dos *Throughput* normalizados com um aumento dos débito de envio por nó em todos os protocolos. O protocolo DACAP sem *Warning* apresenta um melhor desempenho no *Throughput* normalizado em relação aos outros protocolos nas duas arquiteturas apresentadas devido ao aumento do débito na receção em comparação com os débitos recebidos nos outros protocolos. Esses resultados mostram que o protocolo DACAP sem *Warning* alcança o *Throughput* máximo de 1, e além disso é capaz de manter esse nível até uma determinada taxa de tráfego gerado, após o qual o desempenho começa a cair devido as perdas dos pacotes que são gerados.



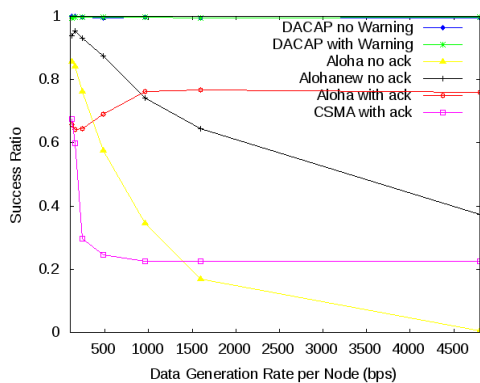
(a) Nó 4 recetor.



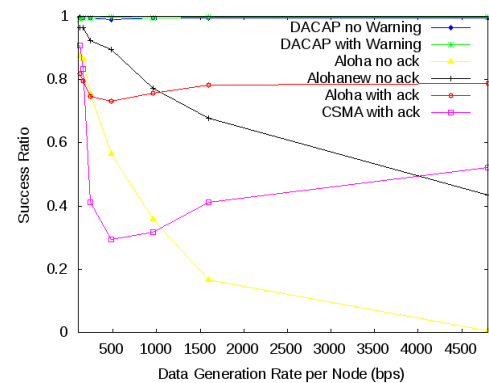
(b) Nó 2 recetor.

Figura 5.20: *Normalized Throughput* em função do *Data Generation Rate per Node* dos protocolos considerando (a) Nó 4 como recetor (b) Nó 2 como recetor.

Os resultados do índice de sucesso dos pacotes considerando as duas arquiteturas são apresentadas nas figuras 5.21 (a) e 5.21 (b). Pode-se verificar que os índices de sucessos dos protocolos ALOHA sem ACK e ALOHANEW sem ACK diminuem consistentemente com o aumento tráfego oferecido. Por outro lado, nos protocolos DACAP sem *Warning* e DACAP com *Warning* o índice de sucesso está sempre próximo dos 100%, justificando que as duas versões dos protocolos DACAP utilizados na simulação apresentam maior índice de sucesso que os outros protocolos devido a ausência de colisões nos protocolos DACAP devido ao mecanismos de *backoff* que obriga os nós estarem em longos períodos de silêncio diminuindo o nível de interferência e as probabilidades de colisões nos protocolos DACAPs.



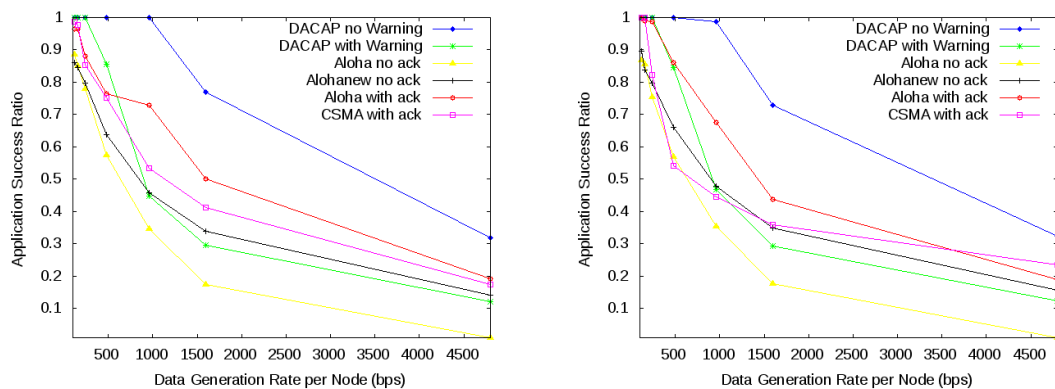
(a) Nó 4 recetor.



(b) Nó 2 recetor.

Figura 5.21: *Success Ratio* em função do *Data Generation Rate per Node* dos protocolos considerando (a) Nó 4 como recetor (b) Nó 2 como recetor.

A figura 5.22 apresenta os resultados do índice de sucesso a nível de aplicação em relação ao débito do envio por nó para as duas arquiteturas: (a) Nó 4 como recetor (b) Nó 2 como recetor. Nesses resultados verifica-se que no protocolo DACAP sem *warning* o índice de sucesso a nível de aplicação mantém-se até um determinado nível de débito de envio por nó e diminui com o aumento do mesmo em consequência da saturação dos nós da rede. No entanto também se pode observar que os índices de sucesso a nível de aplicação dos outros protocolos sofreram uma grande queda uma vez que os pacotes gerados são descartados devido aos atrasos excessivos nos nós. O pior desempenho nesse sentido é apresentado pelo protocolo ALOHA sem ACK.

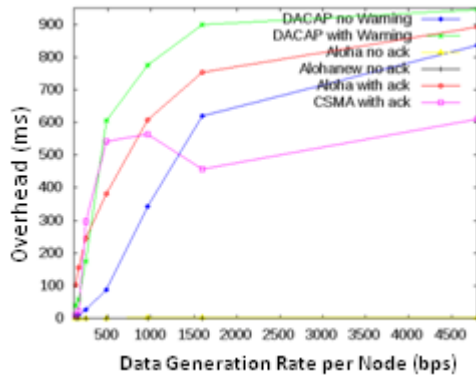


(a) Nó 4 recetor.

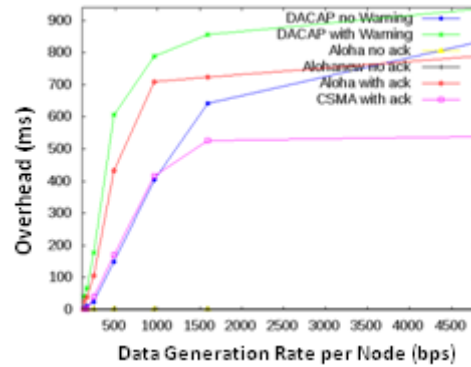
(b) Nó 2 recetor.

Figura 5.22: *Application Success Ratio* em função do *Data Generation Rate per Node* dos protocolos considerando (a) Nó 4 como recetor (b) Nó 2 como recetor.

Em seguida, para finalizar a análise comparativa do desempenho dos protocolos pode-se observar nas figuras 5.23 (a) e 5.23 (b) os resultados dos tempos gastos em mecanismos de controlo (*Overhead*) em função dos débitos do envio por nó. Como se pode verificar, o protocolo DACAP com *Warning* apresentou o maior tempo gasto em mecanismos de controlo em relação aos outros protocolos. O maior tempo gasto em mecanismos de controlo (*Overhead*) deve-se aos pacotes de sinalização (RTS/CTS/*Warning*) aumentando o tempo de espera nas transmissões. Por outro lado, os protocolos sem ACK apresentaram menor tempo gasto em mecanismos de controlo devido a ausência de sobrecarga nesses protocolos diminuindo os tempos de ida e volta.



(a) Nó 4 receptor.



(b) Nó 2 receptor.

Figura 5.23: Tempo gasto em mecanismos de controlo (*Overhead*) em função do *Data Generation Rate per Node* dos protocolos considerando (a) Nó 4 como receptor (b) Nó 2 como receptor.

Pela análise dos resultados dos dados anteriores, considerando as duas arquiteturas apresentadas para tamanho de pacote de 600 bytes utilizando o tráfego de *Poisson* para diversos parâmetros testados pode-se concluir o seguinte:

- O protocolo ALOHA sem ACK apresentou menor desempenho no débito recebido por nó, baixo *Throughput* Normalizado, menor índice de sucesso, mas um baixo *Overhead*;
- O protocolo ALOHANEW sem ACK apresentou um bom débito recebido por nó, bom *Throughput* Normalizado, índice de sucesso limitado e um menor tempo gasto em mecanismos de controlo;
- O protocolo ALOHA com ACK apresentou um Débito recebido por nó razoável, *Throughput* Normalizado limitado, bom índice de sucesso apesar de um tempo gasto em mecanismos de controlo (*Overhead*) elevado;
- O protocolo CSMA apresentou um débito recebido por nó razoável, baixo *Throughput* Normalizado, fraco desempenho no índice de sucesso e um *Overhead* razoável;
- O protocolo DACAP sem *Warning* apresentou o melhor desempenho no débito recebido por nó, bom *Throughput* Normalizado, alta taxa de sucesso e um *Overhead* elevado;
- O protocolo DACAP com *Warning* apresentou um fraco desempenho no débito recebido por nó, baixo *Throughput* Normalizado, maior índice de sucesso de pacote apesar de um maior tempo gasto em mecanismos de controlo (*Overhead*).

Em termos globais pode-se constatar que a utilização do *Warning* no protocolo DACAP com *warning* tende a reduzir a quantidade de pacotes perdidos. Também se verificou que há uma diminuição do número de pacotes recebidos fazendo com que o débito recebido por nó e o *Throughput* Normalizado diminuam. Esta redução deriva da introdução do tempo adicional de



espera no lado do emissor antes de poder enviar o pacote de dados. Assim, a utilização do pacote de *Warning* justifica-se sempre que os nós terminais devam evitar colisões, por exemplo por uma questão de gestão de energia, mesmo que tal acarrete uma diminuição da quantidade de dados recebidos.

### 5.4.3. Análise comparativa dos protocolos MAC para 4 nós emissores e 2 recetores

Nesta seção apresenta-se uma análise comparativa dos resultados obtidos nos cenários apresentados anteriormente com os resultados obtidos do cenário apresentado na figura 5.24. No referido cenário foi aumentado o número de nós da rede para 6 considerando 4 nós emissores (nó 0, nó 1, nó 2 e nó 3) e dois nós recetores (nó 4 e nó 5) sobrepostos com as mesmas distâncias localizados no centro da área da rede. Em seguida definiram-se os mesmos valores para os parâmetros ambientais e físicos, tais como fontes do ruído, salinidade da água, velocidade de propagação do som, batimetria, tipo de sedimento do fundo como foi definido na seção anterior para nossa avaliação. Assumiu-se também que os nós da rede foram equipados com hardware WHOI micromodem, considerando o mesmo tipo de modulação e os mesmos valores para o débito binário, frequência da portadora, tamanho do pacote e o mesmo valor para largura de banda do canal.

No referido cenário utilizado na simulação foram-se aumentados as distâncias entre os nós da rede em comparação com as distâncias consideradas nas seções anteriores. A distância horizontal entre o nó 0 e nó 3 foi definida como 3000 metros, sendo que a distância horizontal entre o nó 0 e o nó 1 foi considerado de 1000 metros e entre nó 0 e nó 2 de 2000 metros. A distância horizontal entre o nó 0 e os nós (4 e 5) e entre o no 3 e nós (4 e 5) foi definida como 1500 metros.

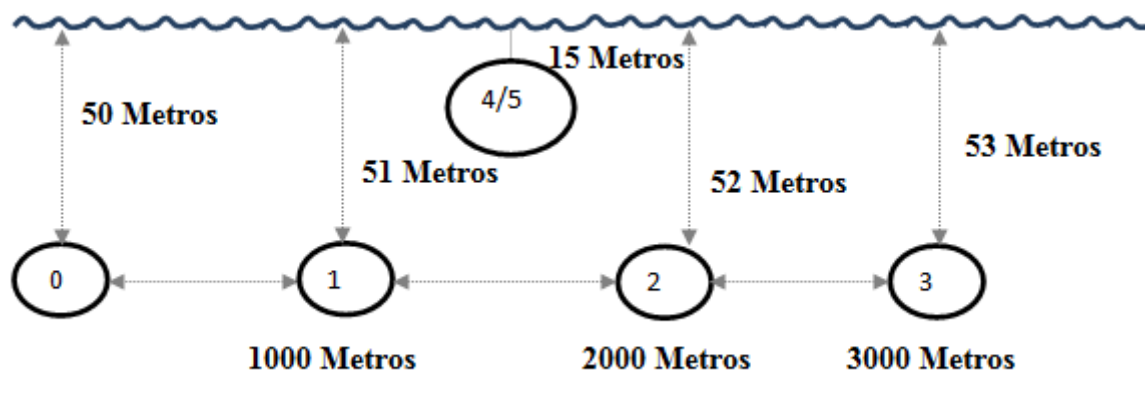


Figura 5.24: Cenário utilizado na simulação para análise comparativa dos protocolos considerando dois nós recetores (nó 4 e nó 5) para tamanho de pacote de 600 bytes.

De forma diferente do cenário apresentado na seção anterior, aqui foram definidos os seguintes valores para a latitude, longitude, profundidade dos nós em metros e distância entre os nós da rede como mostra a tabela 5.4.

Nó	Latitude	Longitude	Profundidade (metros)	Distância
0	42,49°N	10,125°E	-50 Metros	0 Metros
1	42,49°N	10,137°E	-51 Metros	1000 Metros
2	42,49°N	10,149°E	-52 Metros	2000 Metros
3	42,49°N	10,161°E	-53 Metros	3000 Metros
4	42,49°N	10,143°E	-15 Metros	1500 Metros
5	42,49°N	10,143°E	-15 Metros	1500 Metros

Tabela 5.4: Coordenadas e profundidades dos nós utilizados na simulação do cenário da figura 5.24

Em seguida será apresentada a análise comparativa dos resultados dos protocolos utilizando o cenário apresentado na figura 5.24. Esses resultados foram obtidos efetuando 40 simulações como nos cenários anteriores.

A figura 5.25 apresenta o resultado do débito recebido por nó em função do débito de envio por nó dos protocolos para tamanho do pacote 600 bytes. Como se pode verificar os protocolos DACAP apresentaram piores resultados no débito recebido por nó. O protocolo DACAP sem *warning* apresentou um fraco desempenho. Estes resultados contrastam com os resultados obtidos nos cenários apresentados anteriormente que apresentavam maiores valores para o débito recebido por nó para elevados débito de envio por nó. Esse fraco desempenho dos protocolos DACAP deve-se ao fato do referido cenário com dois nós recetores (nó 4 e nó 5). Também se pode observar que o protocolo CSMA apresentou melhorias no débito recebido por nó em comparação com os resultados obtidos anteriormente.

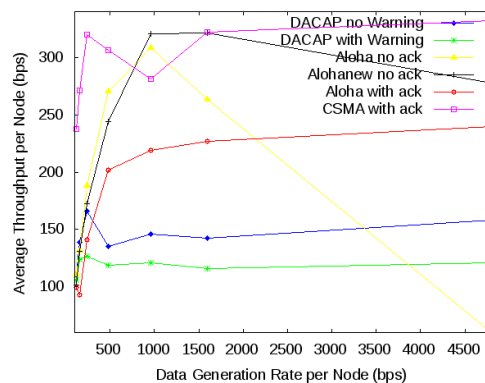


Figura 5.25: *Average Throughput per Node* em função do *Data Generation Rate per Node* dos protocolos para tamanho do pacote de 600 bytes.

O resultado do *Throughput Normalizado* em função do débito de envio por nó dos protocolos para tamanho do pacote 600 bytes é apresentado na figura 5.26. Como pode-se verificar há uma diminuição no desempenho dos *throughputs* normalizados quando aumentamos os débitos de envio por nó em todos os protocolos. Os protocolos DACAPs apresentaram piores desempenhos nos *Throughputs* normalizados quando aumentamos os débitos de envio por nó em relação aos outros protocolos comparando com os resultados obtidos nos cenários anteriores onde o DACAP sem *Warning* apresentava melhores resultados. O fraco desempenho do *Throughput Normalizado* do protocolo DACAP sem *warning* deve-se à diminuição do débito na recepção assim como perdas de pacotes gerados a nível de aplicação.

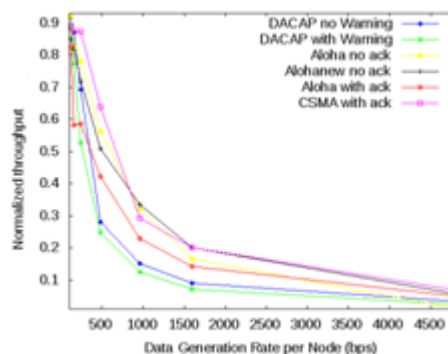


Figura 5.26: *Normalized Throughput* em função do *Data Generation Rate per Node* dos protocolos para tamanho do pacote de 600 bytes.

Na figura 5.27 apresenta-se os resultados do índice de sucesso em função do débito de envio por nó dos protocolos para tamanho do pacote de 600 bytes. Como se pode verificar, o índice de sucesso dos protocolos ALOHA sem ACK e CSMA diminuem consistentemente com o aumento do tráfego oferecido. Podemos observar que o protocolo CSMA apresenta pior desempenho nas redes acústicas submarinas. Por outro lado, os protocolos DACAP sem *Warning* e DACAP com *Warning* apresentaram um índice de sucesso bastante razoável com um valor um pouco abaixo dos 80%

comparando com os resultados anteriores onde o índice do sucesso dos protocolos apresentava resultados bastante bons com um valor muito próximo dos 100%, justificando que as duas versões dos protocolos DACAPs utilizados na simulação apresentam maior índice de sucesso que os outros protocolos devido ao melhor mecanismo de prevenção as colisões apresentado pelos protocolos DACAPs.

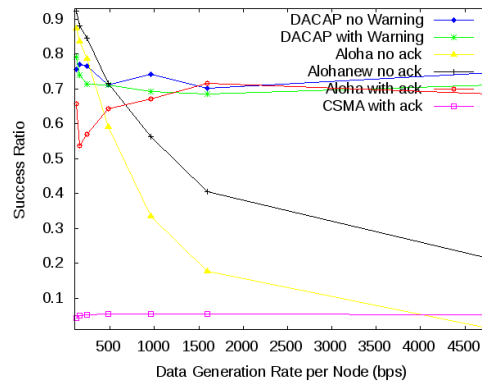


Figura 5.27: *Success Ratio* em função do *Data Generation Rate per Node* dos protocolos para tamanho do pacote de 600 bytes.

A figura 5.28 apresenta o resultado do índice sucesso a nível de aplicação em função do débito de envio por nó dos protocolos para tamanho do pacote 600 bytes. Nessa figura pode-se verificar que os protocolos DACAP atingem o índice de sucesso a nível de aplicação um valor máximo para baixos valores do débito de envio por nó e depois começa a diminuir com o aumento taxa de dados gerados por nó devido a saturação dos nós da rede os novos pacotes serão descartados. Também pode ser observado que os índices sucesso a nível de aplicação dos outros protocolos sofreram uma grande queda devido aos pacotes gerados que são descartados devidos aos atrasos excessivos nos nós. O pior desempenho nesse sentido é apresentado pelo protocolo CSMA. Nos resultados anteriores o ALOHA sem ACK apresentava o pior desempenho.

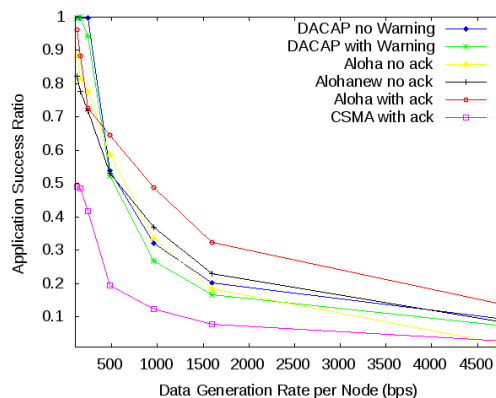


Figura 5.28: *Application Success Ratio* em função do *Data Generation Rate per Node* dos protocolos para tamanho do pacote de 600 bytes.

Finalmente para terminar a análise dos resultados é apresentado na figura 5.29 o resultado do tempo gasto em mecanismos de controlo (*Overhead*) em função do débito de envio por nó dos protocolos para tamanhos do pacote de 600 bytes. Como se pode verificar os protocolos de sinalização DACAP sem *Warning* e DACAP com *Warning* apresentaram maiores tempos gastos em mecanismos de controlo para elevados débitos de envio por nó em relação aos protocolos ALOHA sem ACK e ALOHANEW sem ACK. O maior tempo gasto em mecanismos de controlo desses protocolos deve-se à maior sobrecarga introduzida pelos pacotes de ACK e pacotes de sinalização, aumentando o tempo de espera nas transmissões.

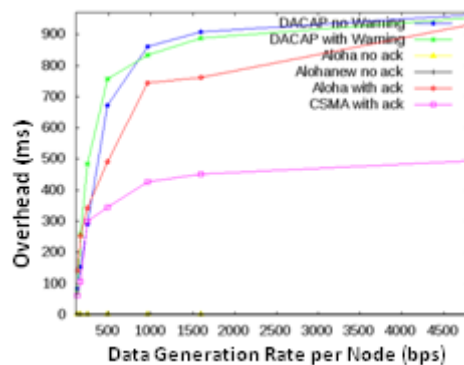


Figura 5.29: Tempo gasto em mecanismos de controlo (*Overhead*) em função do *Data Generation Rate per Node* dos protocolos para tamanho do pacote de 600 bytes.

Dos dados anteriores considerando a arquitetura apresentada na figura 5.24, onde foram aumentadas a complexidade da rede, o número dos nós, as distâncias entre os nós e alterando as coordenadas das latitudes, longitudes e profundidades dos nós para tamanho do pacote de 600 bytes pode-se concluir que os protocolos DACAPs apresentaram um decréscimo a nível dos índices de sucesso dos pacotes em comparação com os resultados anteriores. Também se verificou uma melhoria no débito recebido por nó do protocolo CSMA assim como uma diminuição no desempenho do débito recebido por nó do protocolo DACAP sem *warning* em comparação com os dados anteriores. Em relação aos protocolos ALOHA sem ACK verificou-se que este protocolo apresentou perdas de pacotes ligeiramente superiores ao ALOHANEW sem ACK (com escuta), justificando que a diferença entre as soluções com e sem escuta é pouco significativa, em comparação com a introdução do ACK onde podemos verificar que o protocolo ALOHA com ACK apresentou melhorias a nível do desempenho do índice de sucesso dos pacotes em comparação com os dados anteriores.

Os nossos resultados foram comparados com os resultados do artigo apresentado por Federico Guerra *et al.* [49], onde os autores fizeram uma análise comparativa de três protocolos de acesso ao

meio ALOHA, T-Lohi (em ambas as suas versões, a versão agressiva e a versão conservadora) e DACAP para tamanhos de pacotes de 600 bytes.

As conclusões que os autores em [49] tiraram dos resultados que obtiveram é que os protocolos T-Lohi e DACAP, alcançaram um valor máximo de *Throughput*, e além disso são capazes de manter esse nível até uma determinada taxa de tráfego gerado, após o qual o desempenho começa a cair. Nessa situação, os protocolos ALOHAs (com e sem ACK) apresentaram melhores resultados no *Throughput*. Pode-se verificar que os resultados que os autores obtiveram a nível *Throughput* são semelhantes aos resultados que obtivemos no nosso cenário, onde os protocolos ALOHA na versão com ACK e sem ACK obtiveram melhores resultados que os protocolos DACAP na versão com *Warning* e sem *Warning*. Por outro lado o índice de sucesso do protocolo ALOHA diminui consistentemente com o tráfego oferecido em comparação com os protocolos DACAP onde índice de sucesso está sempre próximo de 1. Em comparação com o nosso resultado pode-se verificar que o DACAP continua a apresentar melhores resultados a nível do índice sucesso, que os protocolos ALOHA, apesar de um ligeiro decréscimo no seu desempenho comparado com os resultados que os autores em [49] obtiveram.

## 5.5. Conclusões

Dos dados obtidos dos vários cenários apresentados neste capítulo pode-se concluir que os protocolos DACAP apresentaram melhores desempenhos nos índices de sucessos dos pacotes em função do débito de envio por nó, quando comparado com os outros protocolos, atingindo valores próximos dos 100 % em quase todos os casos, quer para tamanho de pacotes de 600 quer para 50 bytes. Exceptuam-se as situações em que aumentamos as distâncias entre os nós. Nestes casos pode-se constatar que apesar do protocolo DACAP ainda apresentar desempenho razoável no índice sucesso, houve um decréscimo no seu desempenho atingindo valores um pouco abaixo dos 80% em comparação com os resultados anteriores. Também se conclui que os protocolos DACAP apresentaram um fraco desempenho a nível do débito recebido por nó e *Throughput* Normalizado em quase todos os casos, com poucas excepções. Por exemplo para tamanhos de pacotes de 50 bytes o protocolo DACAP sem *warning* apresentou um bom desempenho (no débito recebido por nó) em comparação com o protocolo DACAP com *warning*, mostrando que a utilização do pacote de *warning* tende a reduzir os pacotes perdidos e diminuir a quantidade de dados recebidos em consequência da introdução do tempo de espera no lado do emissor antes de enviar o pacote de dados. O pacote de *warning* apesar da vantagem de evitar colisões tende a aumentar (juntamente

com os pacotes de sinalização RTS e CTS) a sobrecarga dos protocolos DACAP, aumentando os tempos gastos em mecanismos de controlo (*Overhead*).

Em relação aos protocolos ALOHA pode-se concluir que o protocolo ALOHA sem ACK apresentou melhor desempenho no débito recebido por nó, por comparação com o protocolo ALOHA com ACK, devido à ausência do tempo de espera antes de transmissão dos pacotes de dados. Também se constatou que o protocolo ALOHA sem ACK (sem escuta) apresentou perdas ligeiramente superiores ao ALOHANEW sem ACK (com escuta) tendo menores desempenhos no índice de sucesso quando comparado com o ALOHA com ACK.

O protocolo CSMA apresentou o pior desempenho na maior parte dos cenários apresentados verificando uma degradação acentuada dos parâmetros testados à medida que se aumenta o débito de geração de tráfego nos nós, justificando que o protocolo CSMA utilizado na simulação ainda necessita de algumas alterações de forma adaptar-se melhor em ambientes acústicos submarinos.

Nas tabelas (5.5, 5.6 e 5.7) são apresentados de forma resumida os resultados obtidos pelos protocolos de controlo de acesso ao meio.

Na tabela 5.5, pode-se verificar que, nas duas arquiteturas assimétrica e simétrica os protocolos DACAP (com *warning* e sem *warning*) apresentaram melhor desempenho no índice de sucesso dos pacotes. Também se conclui que a nível do *Throughput* Normalizado para tamanhos de pacotes pequenos (50 bytes) o protocolo ALOHA sem ACK apresentou melhor desempenho, para ambas as arquiteturas. Para tamanhos de pacotes grandes (600 bytes) este protocolo foi superado pelo protocolo DACAP sem *warning* na arquitetura simétrica e pelo ALOHA com ACK na arquitetura assimétrica. Os protocolos ALOHA (sem ACK) apresentaram melhores desempenhos nos tempos gastos em mecanismos de controlo (*Overhead*) utilizando as duas arquiteturas.

Na tabela 5.6 aumentando o número de nós (para 4 nós emissores e um nó recetor) utilizando o cenário da figura 5.16 para tamanhos de pacotes grandes (600 bytes) concluiu-se que os protocolos DACAP (com *warning* e sem *warning*) continuam a apresentar melhor desempenho a nível do índice de sucesso dos pacotes para as duas arquiteturas. Também se constatou uma evolução no desempenho do *Throughput* Normalizado do protocolo DACAP sem *warning* para tamanho de pacotes de 600 bytes apresentado melhor desempenho utilizando as duas arquiteturas. A nível do tempo gasto em mecanismos de controlo (*Overhead*) continuam os protocolos ALOHAs (sem ACK) a apresentar melhor desempenho.

Para tabela 5.7 utilizando o cenário da figura 5.24 (4 nós emissores e 2 nós recetores) para tamanhos de pacote grande (600 bytes) concluiu-se que os protocolos CSMA e ALOHANEW sem

ACK tiveram grandes evoluções ao nível do desempenho no *Throughput* Normalizado, apresentando melhores resultados comparando com os resultados dos cenários obtidos anteriormente, onde o DACAP sem *warning* apresentou melhor desempenho. A nível do índice de sucesso dos pacotes, os protocolos DACAP (sem *warning* e com *warning*) apresentaram melhores desempenhos, mantendo os resultados obtidos nos cenários anteriores. Também se concluiu que os protocolos ALOHA (sem ACK) obtiveram melhor desempenho no tempo gasto em mecanismos de controlo (*Overhead*), quando comparado com os outros protocolos utilizados na simulação.

No capítulo 6 após análise concreta dos resultados obtidos pelos protocolos de acesso ao meio utilizados na simulação (apresentadas no capítulo 5) serão apresentadas as conclusões finais da dissertação e trabalhos futuros na área das redes acústicas submarinas.

	<b>Simétrica</b>		<b>Assimétrica</b>	
<b>Tamanho Pacote</b>	<b>Pequeno (50 bytes)</b>	<b>Grande (600 bytes)</b>	<b>Pequeno (50 bytes)</b>	<b>Grande (600 bytes)</b>
<b>Normalize Throughput</b>	ALOHA sem ACK	DACAP sem <i>Warning</i>	ALOHA sem ACK	ALOHA com ACK
<b>Success Ratio</b>	DACAP sem <i>Warning</i> e DACAP com <i>Warning</i>	DACAP sem <i>Warning</i> e DACAP com <i>Warning</i>	DACAP sem <i>Warning</i> e DACAP com <i>Warning</i>	DACAP sem <i>Warning</i> e DACAP com <i>Warning</i>
<b>Time spent in Control Mechanisms (Overhead)</b>	ALOHA sem ACK e ALOHANEW sem ACK	ALOHA sem ACK e ALOHANEW sem ACK	ALOHA sem ACK e ALOHANEW sem ACK	ALOHA sem ACK e ALOHANEW sem ACK

Tabela 5.5: Resumo sobre o desempenho dos protocolos de acesso ao meio para o cenário da figura 5.2



	<b>Simétrica</b>	<b>Assimétrica</b>
<b>Tamanho Pacote</b>	<b>Grande (600 bytes)</b>	<b>Grande (600 bytes)</b>
<b><i>Normalize Throughput</i></b>	DACAP sem <i>Warning</i>	DACAP sem <i>Warning</i>
<b><i>Success Ratio</i></b>	DACAP sem <i>Warning</i> e DACAP com <i>Warning</i>	DACAP sem <i>Warning</i> e DACAP com <i>Warning</i>
<b><i>Time spent in Control Mechanisms (Overhead)</i></b>	ALOHA sem ACK e ALOHANEW sem ACK	ALOHA sem ACK e ALOHANEW sem ACK

Tabela 5.6: Resumo sobre o desempenho dos protocolos de acesso ao meio para o cenário da figura 5.16

<b>Simétrica</b>	
<b>Tamanho Pacote</b>	<b>Grande (600 bytes)</b>
<b><i>Normalize Throughput</i></b>	CSMA e ALOHANEW sem ACK
<b><i>Success Ratio</i></b>	DACAP sem <i>Warning</i> e DACAP com <i>Warning</i>
<b><i>Time spent in Control Mechanisms (Overhead)</i></b>	ALOHA sem ACK e ALOHANEW sem ACK

Tabela 5.7: Resumo sobre o desempenho dos protocolos de acesso ao meio para o cenário da figura 5.24



## Capítulo 6

# Conclusões e trabalhos futuros

### 6.1. Conclusão

A importância da implementação e teste dos protocolos para o ambiente acústico submarino é cada vez maior devido ao desejo de controlar e explorar os oceanos do mundo. No entanto, nenhum modelo padrão se encontra disponível para os investigadores utilizarem, na fase de simulação do seu projeto. Os trabalhos desenvolvidos no âmbito desta dissertação incidiram fundamentalmente na comparação dos vários protocolos de acesso ao meio (MAC) e da sua avaliação de desempenho em redes acústicas submarinas.

Foram apresentados estudos sobre as características da propagação acústica do canal, tendo na comunicação acústica submarina a sua principal tecnologia. Neste sentido foram apresentadas e discutidas as limitações do canal acústico e as dificuldades que essas limitações implicam na implementação dos protocolos de acesso ao meio nas redes acústicas submarinas. Do canal acústico foi concluído que:

- A velocidade de propagação é baixa provocando grande latência na comunicação. Para além disso, varia com a profundidade;
- A largura de banda além de estreita, depende da distância Emissor/Recetor, limitando as taxas de transmissão a poucos kbps;
- Devido aos efeitos da refração provocados pelo perfil de velocidade do som no mar, as trajetórias dos sinais acústicos podem ser curvilíneas, podendo gerar efeitos de zonas de sombra e zonas de convergência;
- Em águas rasas, devido a múltiplas reflexões na superfície e no leito oceânico, existe um aumento da interferência intersímbolica nas transmissões digitais;

- O consumo de energia para transmissões acústicas é muito maior do que para as transmissões eletromagnéticas terrestres, porque os sinais acústicos são gerados através de vibrações mecânicas das partículas do meio.

Essas limitações foram comprovadas com os resultados das simulações do modelo acústico submarino no simulador de rede NS-2. O modelo foi dividido em 4 camadas: propagação, canal, física e modulação. Foram descritas as funcionalidades críticas de cada uma dessas camadas, descrevendo detalhadamente as funções de transferências utilizadas em cada camada. Além disso, foram apresentados alguns resultados das simulações, comparando os nossos resultados com os resultados do artigo dos autores Harris e Zorzi [22].

Com base nas limitações do canal acústico, comprovadas pelas simulações feitas no NS-2, verificou-se a necessidade de um modelo acústico submarino mais completo para estudos de simulação. Neste sentido propôs-se utilizar o simulador WOSS que integra novas extensões do NS-2/MIRACLE para criar um modelo mais preciso dos efeitos de desvanecimento e multi-percurso no ambiente acústico submarino, baseado no modelo de propagação *Bellhop ray tracing*. Este modelo fornece novas e melhores funcionalidades, incluindo comunicações baseadas no controle de acesso ao meio, protocolos de camada superior, mobilidade, modelos de tráfego e as métricas de desempenho.

O nosso contributo na implementação do protocolo DACAP considerou três fases. Na primeira fase de implementação foi apresentada uma arquitetura da camada MAC para o protocolo DACAP implementando os principais módulos que compõem a referida arquitetura, sem ter em consideração o mecanismo de *warning*. Neste âmbito foram implementados os seguintes módulos: transmissão, recepção, módulo gestão do estado do canal, módulo gestão do estado do *backoff*, módulo gestão do estado transmissão e módulo gestão do estado de recepção. Na segunda fase de implementação foi introduzido o mecanismo de *warning* alterando a implementação do módulo de gestão de estado de recepção. Na terceira e última fase, foram realizadas testes de implementação do protocolo DACAP. Conclui-se que com introdução desse mecanismo de *warning* reduz-se o número de colisões entre pacotes de controlo e dados.

Em seguida realizaram-se testes de simulação para analisar o desempenho dos mecanismos de acesso ao meio nas redes acústicas submarinas para um conjunto de topologias diferentes e diferentes tamanhos de tramas.

O protocolo de acesso ao meio DACAP proposto, mostrou-se eficiente a nível de resolução de problemas de colisões nas redes acústicas submarinas. Os testes demonstraram que os protocolos

DACAP em ambas as versões apresentaram melhores resultados nos índices de sucessos dos pacotes quer para tamanhos de pacotes de 600 e 50 bytes utilizando o tráfego de *Poisson*, justificando pela menor probabilidade de colisões. No entanto a grande desvantagem do protocolo DACAP proposto em relação aos protocolos ALOHA existentes no simulador de rede está no facto de apresentarem um fraco desempenho a nível do *Throughput*. Tal deve-se à introdução dos mecanismos de *warning* e pacotes RTS e CTS, que causam uma maior sobrecarga e aumentam o tempo gasto em mecanismos de controlo (*Overhead*).

De um modo geral podemos dizer que foram atingidos todos os objetivos do trabalho, não tendo apenas sido satisfeitas em alguns parâmetros de desempenho, conforme o argumento acima.

Considera-se assim que a contribuição principal deste trabalho está na implementação do algoritmo do protocolo de acesso ao meio DACAP e que, este pode ser aplicado em qualquer arquitetura para redes sem fios nos ambientes acústicos submarinos.

## 6.2. Trabalhos futuros

Como proposta para trabalho futuros, gostaríamos de aprofundar estudos sobre a modelagem da comunicação acústica com objetivo de aprimorar os módulos para simulação de redes acústicas submarinas no simulador WOSS. Além disso pretendemos estudar o impacto que os parâmetros ambientais têm no desempenho dos protocolos de acesso ao meio, uma vez que tais parâmetros levam em consideração o modelo de propagação que também é uma ferramenta útil para ajudar a avaliar o desempenho da rede.

Finalmente, poucas propostas têm sido apresentadas para o encaminhamento em redes acústicas submarinas devido às características adversas do canal acústico submarino referenciadas nesta dissertação. Neste sentido gostaríamos de explorar a possibilidade de encaminhar os pacotes na rede, implementando os mecanismos que permitam comunicação entre vários nós fixos e móveis na rede, utilizando ligações que disponibilizassem largura de banda suficiente para responder aos seus ritmos de transmissão.



# Bibliografia

- [1] Mandar Chitre, Shiraz Shahabudeen, Lee Freitag, Milica Stojanovic. “Underwater Acoustic Communications and Networking: Recent Advances and Future Challenges”. Sep 2008. doi:10.1109/48.820738.
- [2] Partan, J., Kurose, J., Levine, B. N. “A Survey of Practical Issues in Underwater Networks”. In: *WUWNet '06: Proceedings of the 1st ACM International Workshop on Underwater Networks*, pp. 17–24, New York, NY, USA, 2006. ACM. ISBN: 1-59593-484-7. Available: <http://doi.acm.org/10.1145/1161039.1161045>.
- [3] Akyildiz, I. F., Pompilli, D., Melodia, T. “State-of-the-art in Protocol Research for Underwater Acoustic Sensor Networks”. In: *WUWNet'06: Proceedings of the 1st ACM International Workshop on Underwater*, pp. 7–16, Los Angeles, CA, USA, Sep. 2006. ACM. ISBN:1-59593-484-7. Available: <http://doi.acm.org/10.1145/1161039.1161045>.
- [4] Lanbo, L., Shengli, Z., Jun-Hong, C. “Prospects and Problems of Wireless Communication for Underwater Sensor Networks”, *Wirel. Commun.Mob. Comput.*, v. 8, n. 8, pp. 977–994, 2008. ISSN: 1530-8669. Available: <http://dx.doi.org/10.1002/wcm.v8:8>.
- [5] Akyildiz, I. F., Pompilli, D., Melodia, T. “Underwater Acoustic Sensor Networks: Research Challenges”, *Ad Hoc Networks*, pp. 257–279, 2005. doi: 10.101/j.adhoc.2005.01.004.
- [6] Heidemann, J., Ye, W., Wills, J., et al. “Research Challenges and Applications for Underwater Sensor Networking”, *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, v. 1, pp. 228–235, 2006. ISSN: 1525-3511. doi: 10.1109/WCNC.2006.1683469.
- [7] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: A survey, *Computer Networks* 38 (4) (2002) 393–422.
- [8] E. Cayirci, H. Tezcan, Y. Dogan, V. Coskun, Wireless sensor networks for underwater surveillance systems, *Ad Hoc Networks*, in press; doi:10.1016/j.adhoc.2004.10.008.
- [9] Freitag, L., Stojanovic, M., Grund, M., et al. “Acoustic Communication for Regional Undersea Observatories”. In: *Proceedings of Oceanology International*, 2002.
- [10] Sozer, E., Stojanovic, M., Proakis, J. “Underwater Acoustic Networks”, *Oceanic Engineering, IEEE Journal of*, v. 25, n. 1, pp. 72–83, Jan 2000. ISSN: 0364-9059. doi: 10.1109/48.820738.
- [11] Wilson, W. D. “Speed of Sound in Sea Water as a Function of Temperature, Pressure, and Salinity”, *Journal of the Acoustic Society of America*, v. 32, n. 6, pp. 641–644, 1960.
- [12] Urick, R. J. *Principles of Underwater Sound*. McGraw-Hill, 1983. ISBN:0-07-066087-5.
- [13] Coates, R. F. W. *Underwater Acoustic Systems*. John Wiley and Sons, 1990. ISBN: 0-333-42542-1.
- [14] Brekhovskikh, L. M., Lysanov, Y. P. *Fundamentals of Ocean Acoustics*. Springer-Verlag, 1991.
- [15] Kilfoyle, D., Baggeroer, A. “The State of the Art in Underwater Acoustic Telemetry”, *Oceanic Engineering, IEEE Journal of*, v. 25, n. 1, pp. 4–27, Jan 2000. ISSN: 0364-9059. doi: 10.1109/48.820733.
- [16] ns2 Network Simulator. Available: <http://www.isi.edu/nsnam/ns/>.
- [17] Guerra, F., Casari, P., Zorzi, M. “World ocean simulation system (WOSS): a simulation tool for underwater networks with realistic propagation modeling”. In: *WUWNet '09: Proceedings of the Fourth ACM International Workshop on UnderWater Networks*, pp. 1–8, New

York, NY, USA, 2009. ACM. ISBN: 978-1-60558-821-6. doi: <http://doi.acm.org/10.1145/1654130.1654134>.

- [18] Preisig, J. “Acoustic Propagation Considerations for Underwater Acoustic Communications Network Development”, *SIGMOBILE Mob. Comput. Commun. Rev.*, v. 11, n. 4, pp. 2–10, 2007. ISSN: 1559-1662. doi: <http://doi.acm.org/10.1145/1347364.1347370>.
- [19] M. Stojanovic, “Underwater Acoustic communications”. In: *Encyclopedia of Electrical and Electronics Engineering*, v.22, pp. 688-698, John Wiley and Sons, 2003.
- [20] Freitag, L., Grund, M., Singh, S., et al. “The WHOI Micro-modem: an Acoustic Communications and Navigation System for Multiple Platforms”, *OCEANS, 2005. Proceedings of MTS/IEEE*, v. 2, pp. 1086–1092, 2005. doi: 10.1109/OCEANS.2005.1639901.
- [21] Dokkali, H., Nuaymi, L., Houcke, S. “Power and Distance Based MAC Algorithms for Underwater Acoustic Networks”, *OCEANS 2006*, pp. 1–5, Sept 2006. doi: 10.1109/OCEANS.2006.307004.
- [22] Albert F. Harris, I., Zorzi, M. “Modeling the Underwater Acoustic Channel in ns2”. In: *ValueTools '07: Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools*, pp. 1–8, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). ISBN: 978-963-9799-00-4.
- [23] Albert F. Harris, I., Stojanovic, M., Zorzi, M. “When Underwater Acoustic Nodes Should Sleep with one Eye Open: Idle-time Power Management in Underwater Sensor Networks”. In: *WUWNet '06: Proceedings of the 1st ACM International Workshop on Underwater Networks*, pp. 105–108, New York, NY, USA, 2006. ACM. ISBN: 1-59593-484-7. doi: <http://doi.acm.org/10.1145/1161039.1161061>.
- [24] Chirdchoo, N., Soh, W.-S., Chua, K. C. “Aloha-Based MAC Protocols with Collision Avoidance for Underwater Acoustic Networks”, *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 2271–2275, May 2007. ISSN: 0743-166X. doi: 10.1109/INFCOM.2007.263.
- [25] Pompili, D., Melodia, T., Akyildis, I. F. “A Distributed CDMA Medium Access Control in Underwater Acoustic Sensor Networks”. In: *Proceedings of Med Hoc-Net*, Corfu, Greece, June 2007.
- [26] Chirdchoo, N., Soh, W.-S., Chua, K. C. “MACA-MN: A MACA Based MAC Protocol for Underwater Acoustic Networks with Packet Train for Multiple Neighbors”, *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pp. 46–50, May 2008. ISSN: 1550-2252. doi: 10.1109/VETECS.2008.22.
- [27] Molins, M., Stojanovic, M. “Slotted FAMA: a MAC Protocol for Underwater Acoustic Networks”, *OCEANS 2006 - Asia Pacific*, pp. 1–7, May 2006. doi: 10.1109/OCEANSAP.2006.4393832.
- [28] Lee, J.-W., Kim, J.-P., Lee, J.-H., et al. “An Improved ARQ Scheme in Underwater Acoustic Sensor Networks”, *OCEANS 2008 - MTS/IEEE Kobe Techno-Ocean*, pp. 1–5, April 2008. doi: 10.1109/OCEANSKOB.2008.4530937.
- [29] Xie, P., Hong Cui, J., Lao, L. “VBF: Vector-Based Forwarding Protocol for Underwater Sensor Networks”. In: *in Proceedings of IFIP Networking '06*, pp. 1216–1221, 2006.
- [30] Nicolaou, N., See, A., Xie, P., et al. “Improving the Robustness of Location-Based Routing for Underwater Sensor Networks”, *OCEANS 2007 - Europe*, pp. 1–6, June 2007. doi: 10.1109/OCEANSE.2007.4302470.
- [31] Xie, P., Cui, J.-H. “R-MAC: An Energy-Efficient MAC Protocol for Underwater Sensor Networks”, *Wireless Algorithms, Systems and Applications, 2007. WASA 2007. International*



- Conference on, pp. 187–198, Aug.2007.
- [32] Doukkali, H., Nuaymi, L., Houcke, S. “Distributed MAC Protocols for Underwater Acoustic Data Networks”, *Vehicular Technology Conference, 2006. VTC-2006 Fall. 2006 IEEE 64th*, pp. 1–5, Sept. 2006. doi: 10.1109/VTCF.2006.428.
- [33] Seah, W., Tan, H.-X. “Multipath Virtual Sink Architecture for Underwater Sensor Networks”, *OCEANS 2006 - Asia Pacific*, pp. 1–6, May 2006. doi: 10.1109/OCEANSAP.2006.4393933.
- [34] Morns, I., Hinton, O., Adams, A., et al. “Protocols for Sub-sea Communication Networks”, *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, v.4, pp. 2076–2082 vol.4, 2001. doi: 10.1109/OCEANS.2001.968318.
- [35] Baldo, N., Maguolo, F., Miozzo, M., et al. “ns2-MIRACLE: a modular framework for multi-technology and cross-layer support in network simulator 2”. In: *ValueTools '07: Proceedings of the 2nd international conference on Performance evaluation methodologies and tools*, pp. 1–8, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). ISBN:978-963-9799-00-4.
- [36] M. Porter *et al.*, “Bellhop code.” [Online]. Available: <http://oalib.hlsresearch.com/Rays/index.html>.
- [37] “World ocean atlas.” [Online]. Available: [www.nodc.noaa.gov/OC5/WOA05/pr\\_woa05.html](http://www.nodc.noaa.gov/OC5/WOA05/pr_woa05.html).
- [38] H. Schmidt *et al.*, “GOATS 2008: autonomous, adaptive multistatic acoustic sensing,” Massachusetts Institute of Technology, Tech. Rep., 2008. [Online]. Available: [www.onr.navy.mil/sci\\_tech/32/reports/docs/08/oaschmi3.pdf](http://www.onr.navy.mil/sci_tech/32/reports/docs/08/oaschmi3.pdf)
- [39] “General bathymetric chart of the oceans.” [Online]. Available: [www.gebco.net](http://www.gebco.net)
- [40] “National geophysical data center, seafloor surficial sediment descriptions.” [Online]. Available: <http://www.ngdc.noaa.gov/mgg/geology/deck41.html>
- [41] F. Jensen, W. Kuperman, M. Porter, and H. Schmidt, *Computational Ocean Acoustics*. Springer-Verlag, 2000.
- [42] Tan, H.-P., Sean, W., Doyle, L. “A Multi-hop ARQ Protocol for Underwater Acoustic Networks”, *OCEANS 2007 - Europe*, pp. 1–6, June 2007. doi:10.1109/OCEANSE.2007.4302195.
- [43] Park, M. K., Rodoplu, V. “UWAN-MAC: An Energy-Efficient MAC Protocol for Underwater Acoustic Wireless Sensor Networks”, *Oceanic Engineering, IEEE Journal of*, v. 32, n. 3, pp. 710–720, July 2007. ISSN:0364-9059. doi: 10.1109/JOE.2007.899277.
- [44] Chirdchoo, N., Soh, W.-S., Chua, K. C. “Aloha-Based MAC Protocols with Collision Avoidance for Underwater Acoustic Networks”, *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 2271–2275, May 2007. ISSN: 0743-166X. doi:10.1109/INFCOM.2007.263.
- [45] Peleato, B., Stojanovic, M. “A MAC Protocol for Ad-hoc Underwater Acoustic Sensor Networks”. In: *WUWNet '06: Proceedings of the 1st ACM International Workshop on Underwater Networks*, pp. 113–115, New York, NY, USA, 2006. ACM. ISBN: 1-59593-484-7. doi: <http://doi.acm.org/10.1145/1161039.1161063>.
- [46] Peleato, B., Stojanovic, M. “Distance Aware Collision Avoidance Protocol for Ad-hoc Underwater Acoustic Sensor Networks”, *Communications Letters, IEEE*, v. 11, n. 12, pp. 1025–1027, December 2007. ISSN:1089-7798. doi: 10.1109/LCOMM.2007.071160.
- [47] Makh.A, D., Kumaraswamy, P., Roy, R. “Challenges and Design of MAC Protocol for Underwater Acoustic Sensor Networks”, *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*, pp. 1–6, April 2006.
- [48] Stojanovic, M. “Optimization of a Data Link Protocol for an Underwater Acoustic Channel”,

*Oceans 2005 - Europe*, v. 1, pp. 68–73 Vol. 1, June 2005. doi: 10.1109/OCEANSE.2005.1511686.

- [49] Federico Guerra, Paolo Casari, Michele Zorzi, “A performance comparison of MAC protocols for underwater networks using a realistic channel simulator,” in *Proc. of MTS/IEEE Oceans*, Biloxi, MS, Oct. 2009.
- [50] F. Guerra, P. Casari, and M. Zorzi, “MAC protocols for monitoring and event detection in underwater networks employing a FH-BFSK physical layer,” in *Proc. of IACM UAM*, Nafplion, Greece, Jun. 2009.
- [51] Rice, J., Creber, B., Fletcher, C., et al. “Evolution of Seaweb Underwater Acoustic Networking”, *OCEANS 2000 MTS/IEEE Conference and Exhibition*, v. 3, pp. 2007–2017 vol.3, 2000. doi: 10.1109/OCEANS.2000.882235.
- [52] D. D. Falconer, F. Adachi, and B. Gudmundson, “Time division multiple access methods for wireless personal communications,” *IEEE Commun.Mag.*, pp. 33–36, Jan. 1995.
- [53] T. S. Rappaport, *Wireless Communications*. Englewood Cliffs, NJ:Prentice Hall, 1996.
- [54] R. Kohno, R. Meidan, and L. B. Milstein, “Spread spectrum methods for wireless communications,” *IEEE Commun. Mag.*, pp. 33–36, Jan. 1995.
- [55] D. Bertsekas and R. Gallager, *Data Networks*, NJ: Prentice Hall, 1992.
- [56] L. G. Roberts, “ALOHA packet system with and without slots and capture,” *ACMSIGCOM Comput. Commun. Rev.*, pp. 28–42, 1975.
- [57] L. Kleinrock and F. A. Tobagi, “Carrier sense multiple access for packet switched radio channels,” in *Proc. ICC’74*, June 1974, pp.21B-1–21B-7.
- [58] F. A. Tobagi and L. Kleinrock, “Packet switching in radio channels, part II: The hidden terminal problem in carrier sense multiple access and busy tone solution,” *IEEE Trans. Commun.*, pp. 1417–1433, 1975.
- [59] H. Takagi and L. Kleinrock, “Correction to ‘throughput analysis for persistent CSMA systems’,” *IEEE Trans. Commun.*, pp. 243–245, 1987.
- [60] Karn, P. “MACA, A New Channel Access Method for Packet Radio”. ARRL/CRRL Amateur Radio 9th Computer Networking Conf., 1990.
- [61] Bharghavan, V., Demers, A., Shenker, S., et al. “MACAW: a Media Access Protocol for Wireless LAN’s”, *SIGCOMM Comput. Commun. Rev.*,v. 24, n. 4, pp. 212–225, 199. ISSN: 0146-4833. doi: <http://doi.acm.org/10.1145/190809.190334>.
- [62] N. Abramson, “The ALOHA systems – Another Alternative for computer communications,” *AFIPS Conf. Proc., Fall Joint Comput. Conf.*, 37, 281-285, 1970.
- [63] K. Pahlavan and A. H. Levesque, *Wireless Information Networks*, New York: Wiley, 1995.
- [64] Monteiro, Jânio. 2011. *Sebenta de Redes de Comunicação*. Capítulo 4. Available: <http://w3.ualg.pt/~jmmonte/RC.htm/>.
- [65] S. Basagni, C. Petrioli, R. Petroccia, and M. Stojanovic, “Choosing the packet size in multi-hop underwater networks,” in *Proceedings of IEEE OCEANS 2010*, Sydney, Australia, May, 24–27 2010, pp. 1–9.
- [66] Doukkali, H., Nuaymi, L. “Analysis of MAC Protocols for Underwater Acoustic Data Networks”, *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, v. 2, pp. 1307–1311 Vol. 2, May-1 June 2005. ISSN: 1550-2252. doi: 10.1109/VETECS.2005.1543520.
- [67] Doukkali, H., Nuaymi, L., Houcke, S. “Power and Distance Based MAC Algorithms for Underwater Acoustic Networks”, *OCEANS 2006*, pp. 1–5, Sept 2006. doi: 10.1109/OCEANS.2006.307004.

- [68] Doukkali, H., Nuaymi, L., Houcke, S. "Power and Distance Based MAC Algorithms for Underwater Acoustic Networks", *OCEANS 2006*, pp. 1–5, Sept 2006. doi: 10.1109/OCEANS.2006.307004.
- [69] Syed, A., Ye, W., Heidemann, J. "T-Lohi: A New Class of MAC Protocols for Underwater Acoustic Sensor Networks", *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pp. 231–235, April 2008. ISSN: 0743-166X. doi: 10.1109/INFOCOM.2008.55.
- [70] A. Syed, W. Ye, and J. Heidemann, "Comparison and Evaluation of the T-Lohi MAC for Underwater Acoustic Sensor Networks," *IEEE J. Select. Areas Commun.*, vol. 26, pp. 1731–1743, Dec. 2008.
- [71] C. Petrioli, R. Petroccia, and M. Stojanovic, "A comparative performance evaluation of MAC protocols for underwater sensor networks," in *Proc. MTS/IEEE Oceans*, Québec City, Canada, 2008.
- [72] Tian, C., Liu, W., Jin, J., et al. "Localization and Synchronization for 3D Underwater Acoustic Sensor Networks", *Lecture Notes in Computer Science.*, n. 4611, pp. 622–631, 2007. ISSN: 0302-9743.
- [73] K. Pahlavan and A. H. Levesque, *Wireless Information Networks*, New York: Wiley, 1995.
- [74] Ko, J., Shin, J., Kwon, S., et al. "Localization of Sensor Nodes in Underwater Acoustic Sensor Networks Using Two Reference Points". In: *Information Networking, 2008. ICOIN 2008. International Conference on*, pp. 1–5, 2008.
- [75] Chandrasekhar, V., Seah, W. K., Choo, Y. S., et al. "Localization in Underwater Sensor Networks: Survey and Challenges". In: *WUWNet'06: Proceedings of the 1st ACM International Workshop on Underwater Networks*, pp. 33–40, New York, NY, USA, 2006. ACM. ISBN: 1-59593-484-7. doi: <http://doi.acm.org/10.1145/1161039.1161047>.
- [76] D. B. Johnson and D. A. Maltz, "Protocols for adaptive wireless and mobile networking," *IEEE Personal Commun.*, 1996.
- [77] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *ACM/IEEE Int. Conf. Mobile Computing and Networking*, Oct. 1998.
- [78] J. Jubin and J. D. Tornow, "The DARPA packet radio network protocols," in *Proc. IEEE*, vol. 75, Jan. 1987, pp. 21–32.
- [79] C. E. Perkins, "Ad hoc on demand distance vector (AODV) routing," Internet-Draft, draft-ietf-manet-aodv-00.txt, 1997.
- [80] X. Guo, M. Frater, and M. Ryan, "A propagation-delay-tolerant collision avoidance protocol for underwater acoustic sensor networks," in *Proc. IEEE Oceans Conf. 2006*.
- [81] Othmam, A.-K. "GPS-less Localization Protocol for Underwater Acoustic Networks". In: *Wireless and Optical Communications Networks, 2008. WOCN '08. 5th IFIP International Conference*, pp. 1 –6, May 2008. doi:10.1109/WOCN.2008.4542532.



# Anexo

## Anexo A. Modelo de canal e da camada física no NS-2

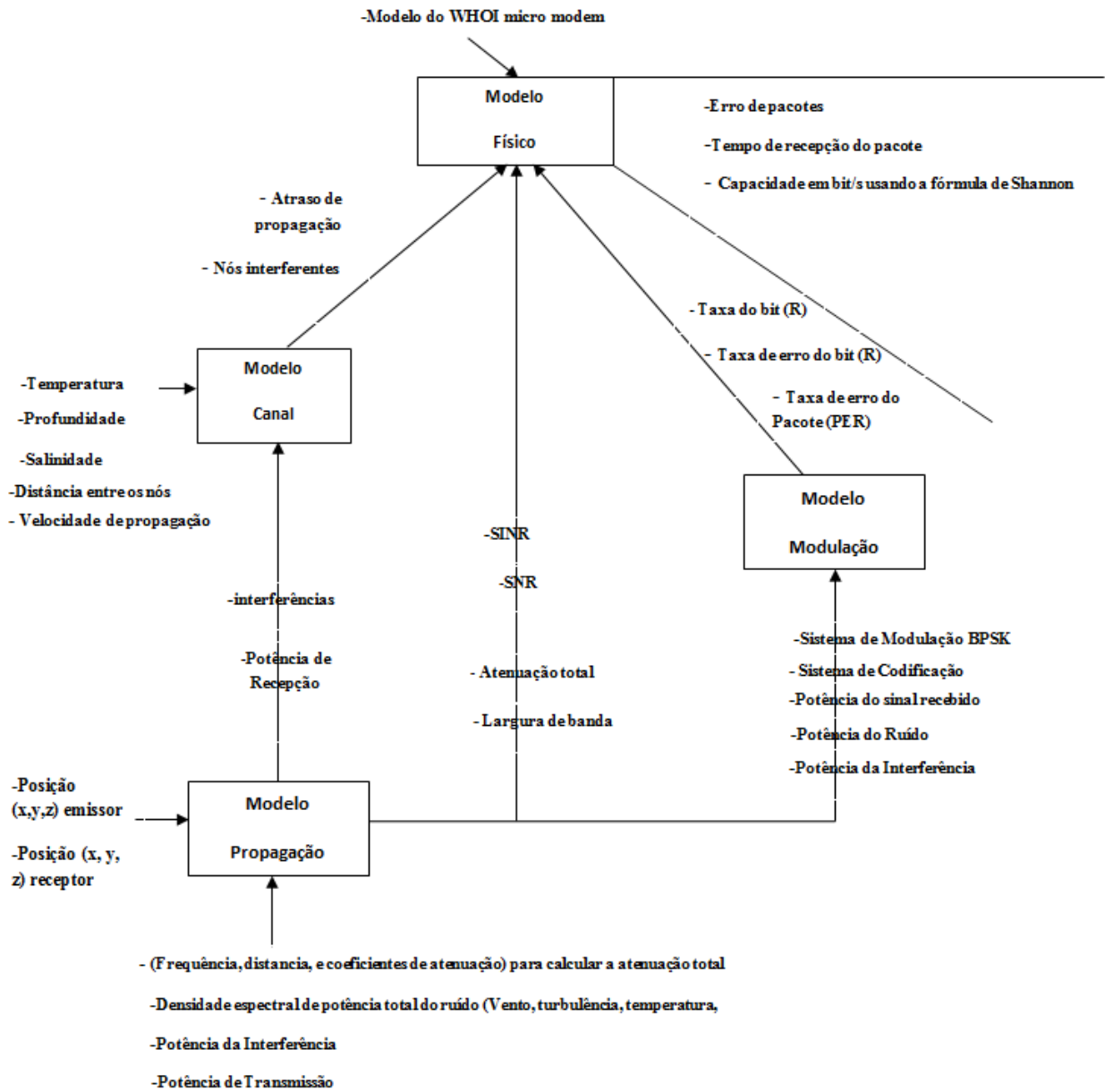


Figura A.1: Modelo de canal e da camada física no NS-2.

## Anexo B. Função de transferência de cada um dos modelos

### B.1. Função de transferência do modelo físico

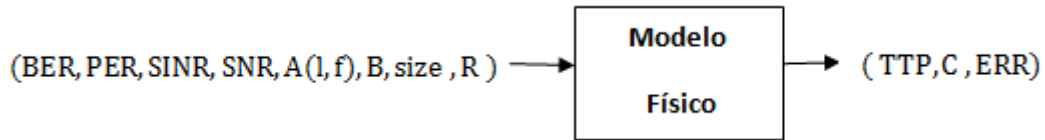


Figura B.1: Função de transferência do modelo físico.

$$TTP = \frac{8 \times \text{size}}{R}$$

$$C = B \times \log_2(1 + SNR)$$

$$ERR = 1 - PER$$

*SINR*=Interferência da relação sinal ruído em dB

*SNR*=Relação sinal ruído no receptor em dB

*Size*= Tamanho do pacote em bytes

*B*=Largura de banda (Hz)

*C*=cálculo da capacidade em bit/s

*A* (*l*, *f*) =Atenuação total (dB)

*R*=Taxa do bit (bit/s)

*PER*=Taxa do erro do pacote

*BER*=Taxa de erro do bit

*ERR*=Erro do pacote

*TTC*=Tempo de transmissão do pacote

## B.2. Função de transferência do modelo canal

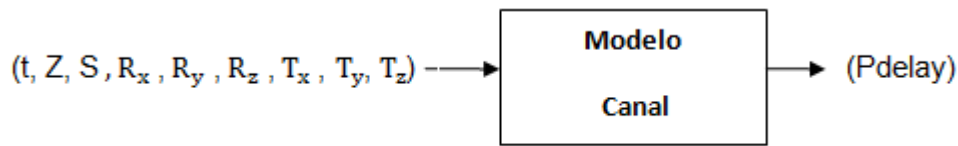


Figura B.2: Função de transferência do modelo canal.

$$\text{Totaldist} = \sqrt{((R_x - T_x)^2 + (R_y - T_y)^2 + (R_z - T_z)^2)}$$

$$C = 1448.5 + 45.7t - 5.2t^2 + 0.23t^2 + (1.333 - 0.126t + 0.009t^2)(S - 35) + 16.3Z + 0.18Z^2$$

$$Pdelay = \frac{\text{Totaldist}}{c}$$

t = Temperatura ( °C )

Z = Profundidade (m)

S = Salinidade (ppm)

Totaldist = Distância total (m)

R<sub>x</sub> = distância x do receptor (m)

R<sub>y</sub> = distância y do receptor (m)

R<sub>z</sub> = distância z do receptor (m)

T<sub>x</sub> = distância x do transmissor (m)

T<sub>y</sub> = distância y do transmissor (m)

T<sub>z</sub> = distância z do transmissor (m)

c = velocidade da propagação acústica submarina em (m/s)

Pdelay = Atraso de propagação (s)

### B.3. Função de transferência do modelo modulação

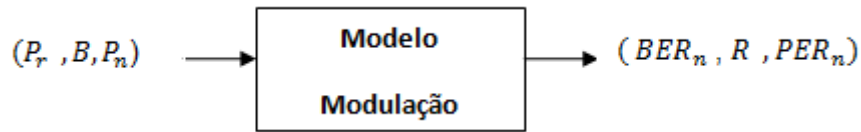


Figura B.3: Função de transferência do modelo modulação.

$$R = \frac{1}{T} = \frac{B}{2}$$

$$BER_n = \frac{1}{2} \operatorname{erfc} \sqrt{\frac{P_r}{P_n}}$$

$$PER_n = 1 - (1 - BER_n)^{\text{size} \times 8.0}$$

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-\gamma^2} d\gamma$$

$P_n$ =Potência do Ruído (dBm ou linear)

$P_r$ =Potência Recebida (dBm ou linear)

$BER_n$ =Taxa de erro do bit

$PER_n$ =Taxa de erro do pacote

$Size$ =Tamanho do pacote (bytes)

$R$ =Taxa do bit (bit/s)

$\operatorname{erfc}$ =função complementar de erro



## B.4. Função de transferência do modelo propagação



Figura B.4: Função de transferência do modelo propagação.

$$SNR = \frac{P_t/A(l, f)}{N(f)}$$

$$SINR = \frac{P_t/A(l, f)}{N(f) + P_i}$$

$$10 \log a(f) = \begin{cases} 0.11 \frac{f^2}{1+f^2} + 44 \frac{f^2}{4100+f} + 2.75 \cdot 10^{-4} f^2 + 0.003, & f \geq 0.4 \\ 0.002 + 0.01 \left( \frac{f}{1+f} \right) + 0.011f, & f \leq 0.4 \end{cases}$$

$$N(f) = N_t(f) + N_s(f) + N_w(f) + N_{th}(f)$$

$$10 \log A(l, f) = k \cdot 10 \log l + l \cdot 10 \log a(f)$$

$$10 \log N_t(f) = 17 - 30 \log f$$

$$10 \log N_s(f) = 40 + 20(s - 0.5) + 26 \log f - 60 \log(f + 0.03)$$

$$10 \log N_w(f) = 50 + 7.5w^{\frac{1}{2}} + 20 \log f - 40 \log(f + 0.4)$$

$$10 \log N_{th}(f) = -15 + 20 \log f$$

$$P_i = \sum_{n \neq 0} |g^2(n)|$$

$g_n$  = Resposta impulsiva

$S$  = navegação (*shipping* [0 1])

$Th$  = Temperatura ( $^{\circ}C$ )

$w$  = Vento (wind in m/s)

$t$  = turbulência

$f$  = Frequência ( $kHz$ )

$l$  = Distância (m)

$P_i$  = Potência de interferência (dBm ou linear)

$P_T$  = Potência de transmissão (dBm ou linear)

$K$  = coeficiente de propagação

$a(f)$  = Coeficiente de atenuação (absorption loss) (dB/m ou (dB re  $\mu Pa$ ))

$N(f)$  = densidade espectral de potência total do ruído (dB re  $\mu Pa$  per)

SNR = Relação sinal ruído no receptor (dB)

SINR=Interferência da relação sinal ruído (dB)

$A(l, f)$  =Atenuação total (dB)

$(f_l, f_r)$  = bandas de frequências usadas na comunicação

$B$  = largura de Banda (Hz)

$N_t$  =Ruído dado pela turbulência (dB)

$N_{th}$  =Ruído Térmico (dB)

$N_w$  = Ruído dado pelo vento (dB)

$N_s$  = Ruído dado pela navegação (dB)

## Anexo C. Análise comparativa entre protocolo ALOHA sem ACK e ALOHANEW sem ACK

ALOHA sem ACK (sem escuta)				ALOHANEW sem ACK (com escuta)			
Tipo de pacote	Nº de pacote enviado	Nº de Pacote Recebido	Nº de simulação	Tipo de pacote	Nº de pacote enviado	Nº de Pacote Recebido	Nº de simulação
CBR(0)	466	462	1	CBR(0)	502	495	1
CBR(1)	471	467	1	CBR(1)	461	457	1
CBR(0)	491	488	2	CBR(0)	514	507	2
CBR(1)	458	455	2	CBR(1)	465	457	2
CBR(0)	519	514	3	CBR(0)	478	472	3
CBR(1)	538	534	3	CBR(1)	528	520	3
CBR(0)	509	507	4	CBR(0)	524	520	4
CBR(1)	513	511	4	CBR(1)	506	501	4
CBR(0)	501	497	5	CBR(0)	489	484	5
CBR(1)	522	518	5	CBR(1)	500	493	5
CBR(0)	518	517	6	CBR(0)	497	491	6
CBR(1)	499	498	6	CBR(1)	517	508	6
CBR(0)	522	521	7	CBR(0)	487	481	7
CBR(1)	491	489	7	CBR(1)	451	443	7
CBR(0)	488	484	8	CBR(0)	511	507	8
CBR(1)	486	482	8	CBR(1)	513	508	8
CBR(0)	464	459	9	CBR(0)	516	511	9
CBR(1)	432	428	9	CBR(1)	511	504	9
CBR(0)	535	428	10	CBR(0)	509	495	10
CBR(1)	517	508	10	CBR(1)	530	520	10
CBR(0)	480	476	11	CBR(0)	450	447	11
CBR(1)	485	482	11	CBR(1)	493	488	11
CBR(0)	482	478	12	CBR(0)	471	463	12
CBR(1)	505	502	12	CBR(1)	517	506	12
CBR(0)	525	523	13	CBR(0)	521	518	13
CBR(1)	471	469	13	CBR(1)	499	495	13
CBR(0)	475	475	14	CBR(0)	500	496	14
CBR(1)	503	503	14	CBR(1)	487	484	14
CBR(0)	472	470	15	CBR(0)	479	471	15
CBR(1)	484	483	15	CBR(1)	505	499	15
CBR(0)	493	492	16	CBR(0)	501	497	16
CBR(1)	475	474	16	CBR(1)	466	460	16
CBR(0)	438	436	17	CBR(0)	510	502	17
CBR(1)	516	514	17	CBR(1)	486	474	17
CBR(0)	479	474	18	CBR(0)	524	518	18
CBR(1)	510	504	18	CBR(1)	504	490	18
CBR(0)	538	529	19	CBR(0)	547	539	19
CBR(1)	507	498	19	CBR(1)	515	509	19
CBR(0)	517	510	20	CBR(0)	507	494	20
CBR(1)	484	478	20	CBR(1)	497	487	20

Tabela C.1: Tabela comparativa de 2 versões do protocolo ALOHA sem escuta e com escuta durante 20 simulações.

<b>Tipo de Pacote</b>	<b>Média Pacotes Enviados</b>	<b>Média Pacotes Recebidos</b>
CBR(0)	500,60	487,00
CBR (1)	488,35	484,85

Tabela C.2: Média dos pacotes CBR enviados e recebidos pelo protocolo ALOHA sem ACK (sem escuta).

<b>Tipo de Pacote</b>	<b>Média Pacotes Enviados</b>	<b>Média Pacotes Recebidos</b>
CBR (0)	501,85	495,40
CBR (1)	497,55	490,15

Tabela C.3: Média dos pacotes CBR enviados e recebidos pelo protocolo ALOHANEW sem ACK (com escuta).

## Anexo D. Diagrama de estado dos módulos utilizados na implementação do DACAP

### D.1. Diagrama de estado do módulo gestão do estado de transmissão

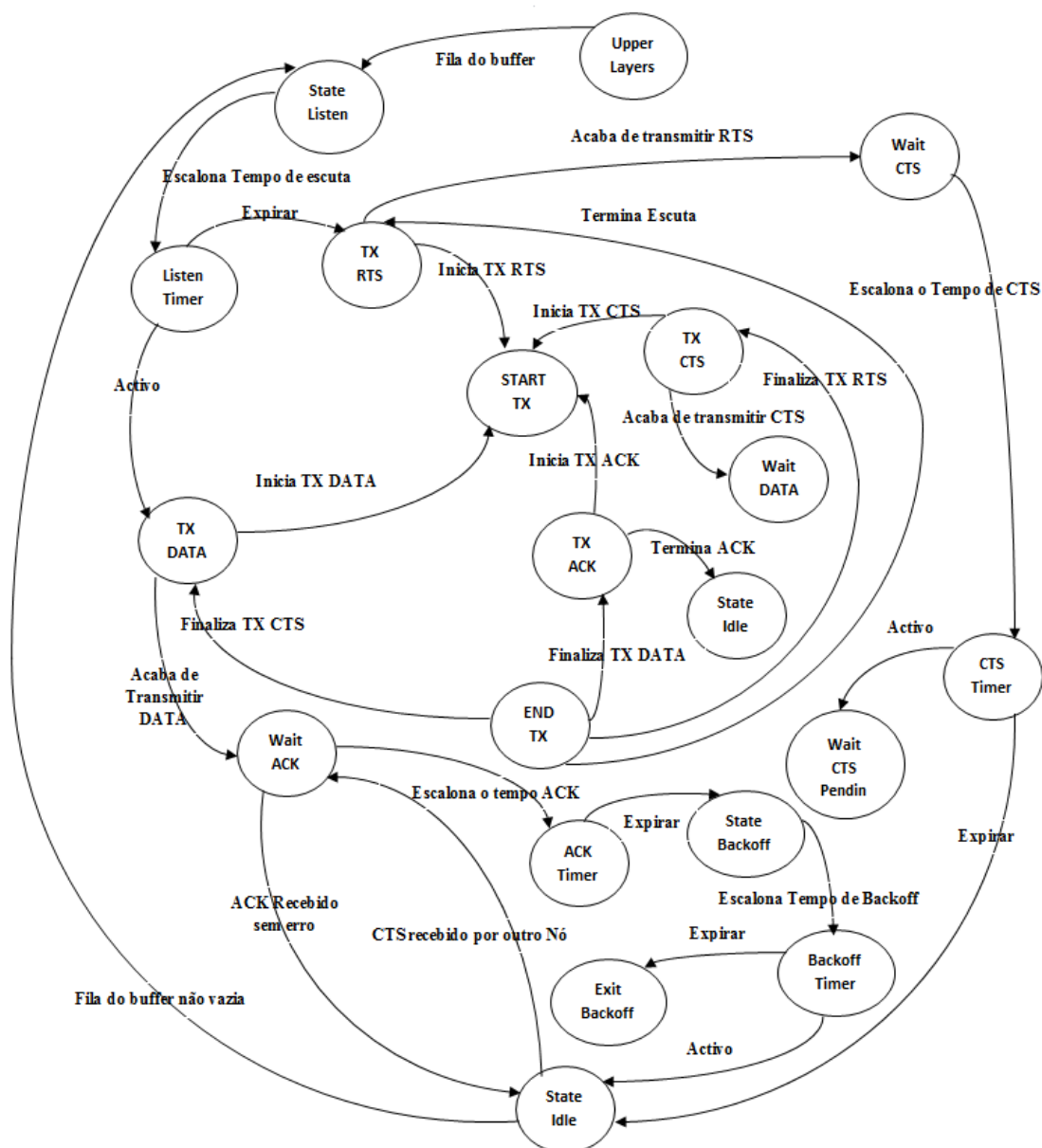


Figura D.1: Diagrama de estado do módulo gestão do estado de transmissão.

## D.2. Diagrama de estado do módulo gestão do estado de receção

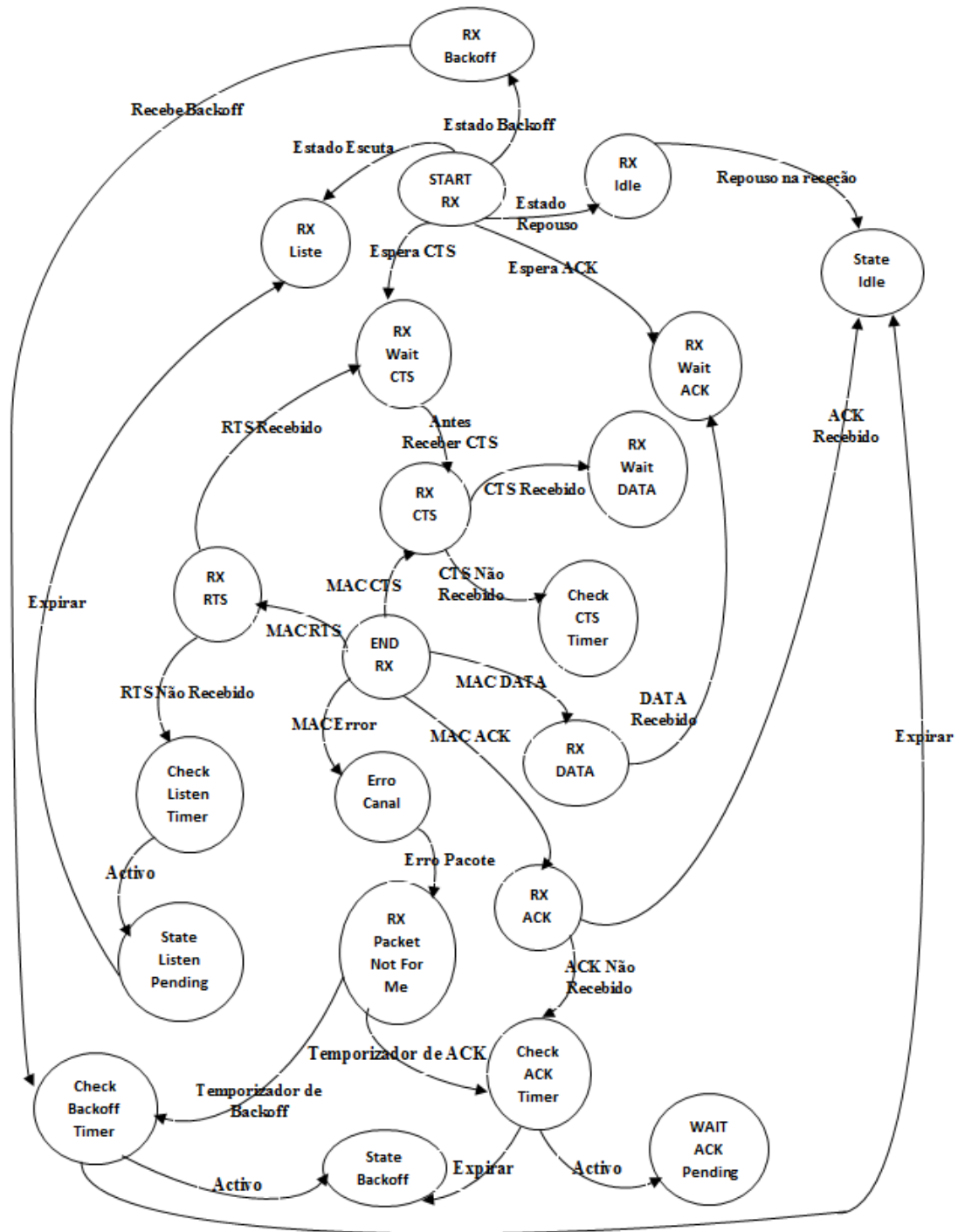


Figura D.2: Diagrama de estado do módulo gestão do estado de receção.

### D.3. Diagrama de estado do módulo gestão do estado de receção com *warning*

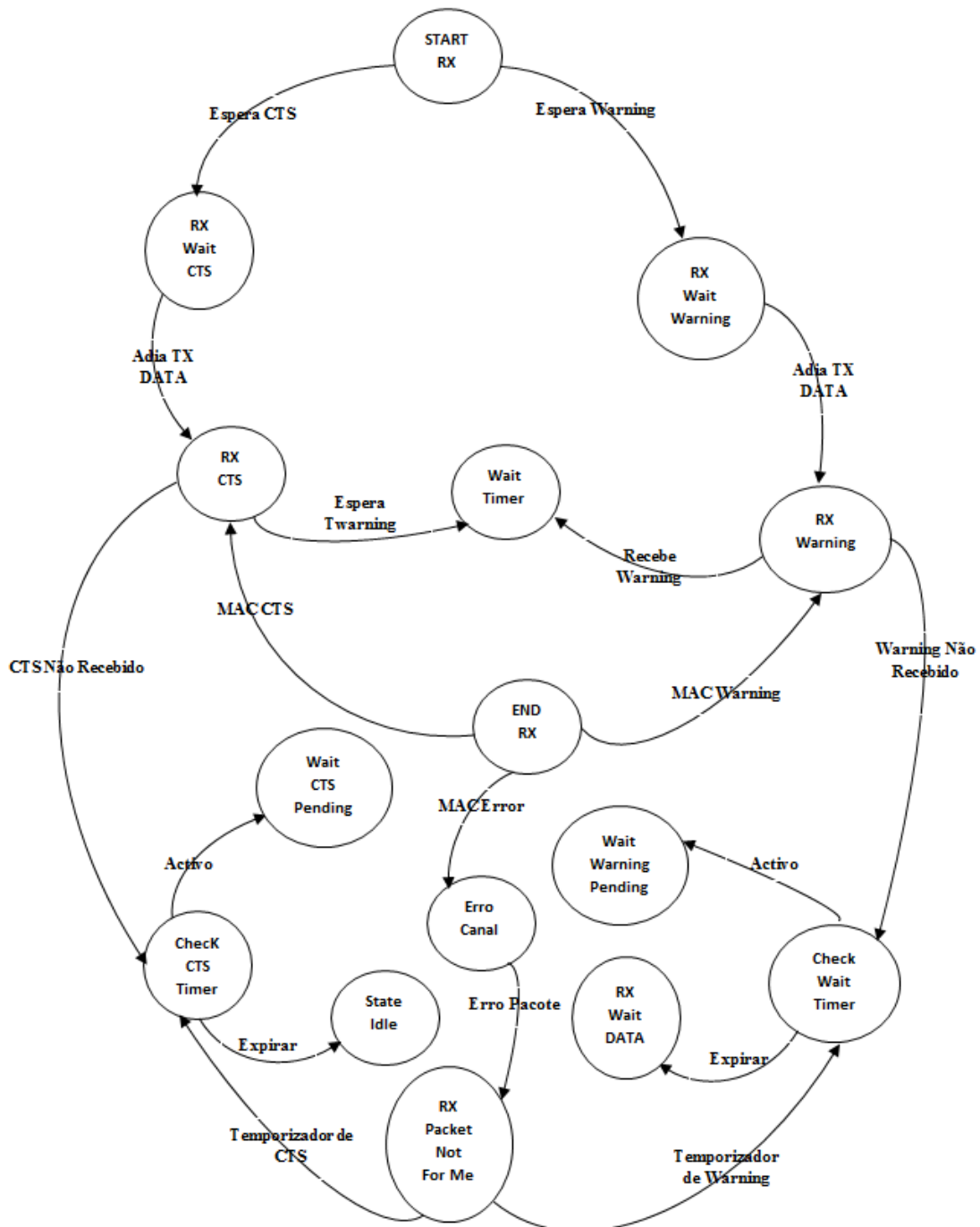


Figura D.3: Diagrama de estado do módulo gestão do estado de receção com *Warning*.

## Anexo E. Resultados dos testes

### E.1. Análise comparativa dos protocolos MAC para 2 nós emissores e 1 recetor

#### E.1.1. Arquitetura simétrica para tamanho pacote 600 byte

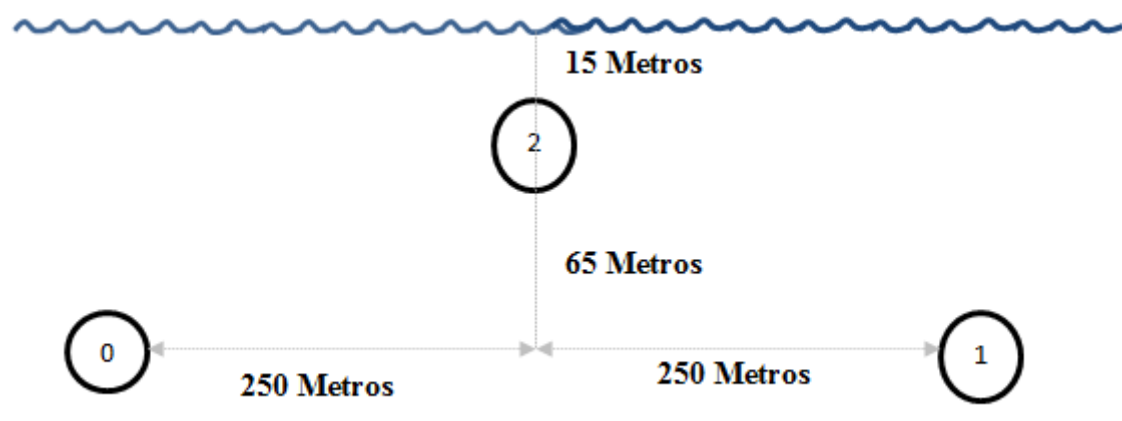


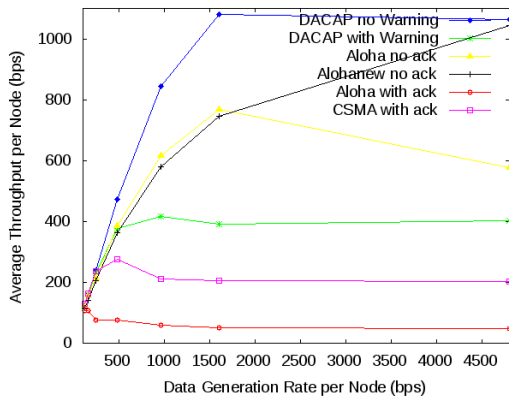
Figura E.1: Cenário utilizado na simulação dos protocolos para tamanho de pacote 600 byte.

Nó	Latitude	Longitude	Profundidade (metros)	Distância
0	42,59°N	10,125°E	-80 Metros	0 Metros
1	42,59°N	10,131°E	-80 Metros	500 Metros
2	42,59°N	10,128°E	-15 Metros	250 Metros

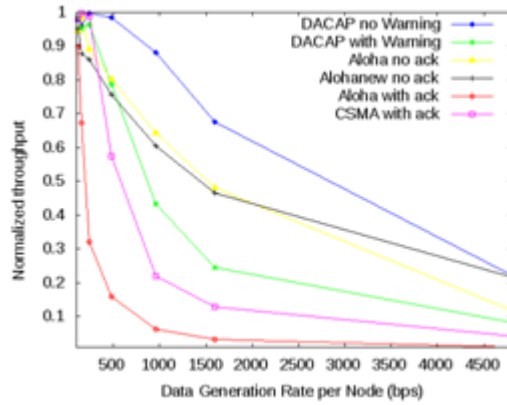
Tabela E.1: Coordenadas e profundidades dos nós utilizados na simulação utilizando o cenário apresentado na figura E.1.



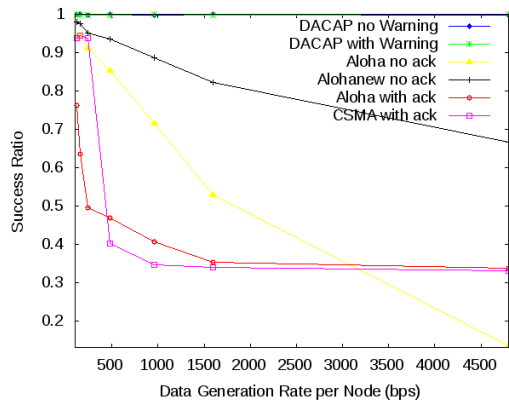
**Teste Average Throughput**



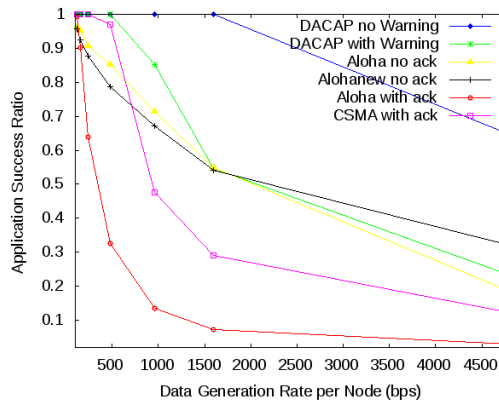
**Teste Normalized Throughput**



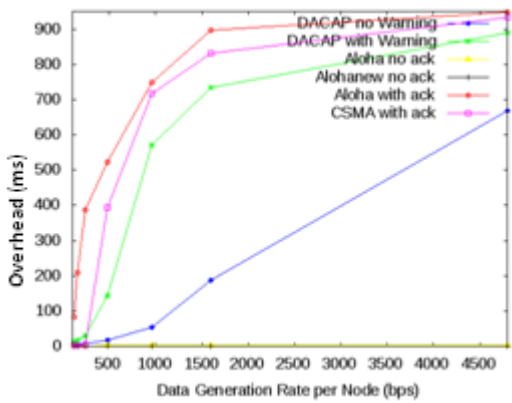
**Teste Success Ratio**



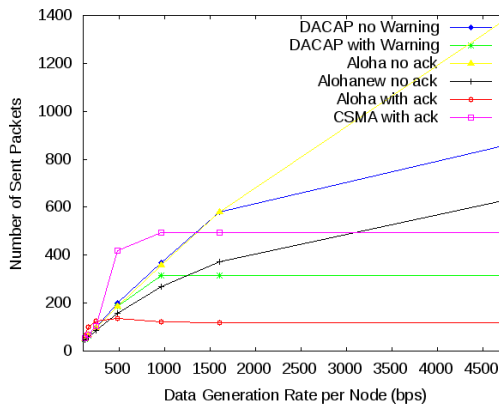
**Teste Applicattion Success Ratio**



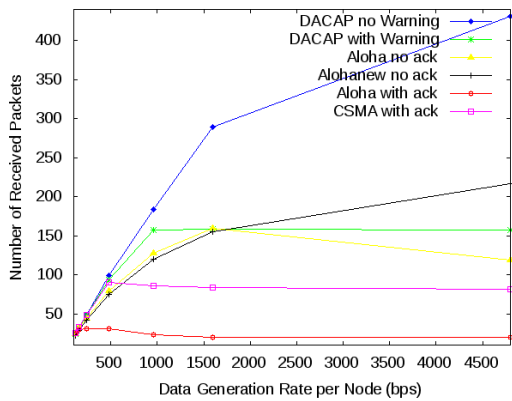
**Teste Overhead**



**Teste Number of Sent Packets**



### Teste Number of Received Packets



### Teste Application Layer Packet Loss Ratio

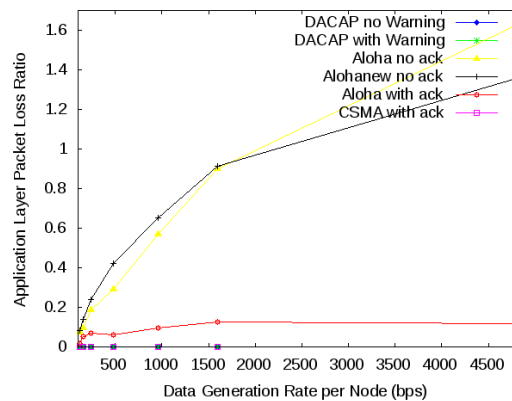


Figura E.2: Resultados obtidos do cenário apresentado na figura E.1.

## E.1.2. Arquitetura simétrica para tamanho pacote 50 byte

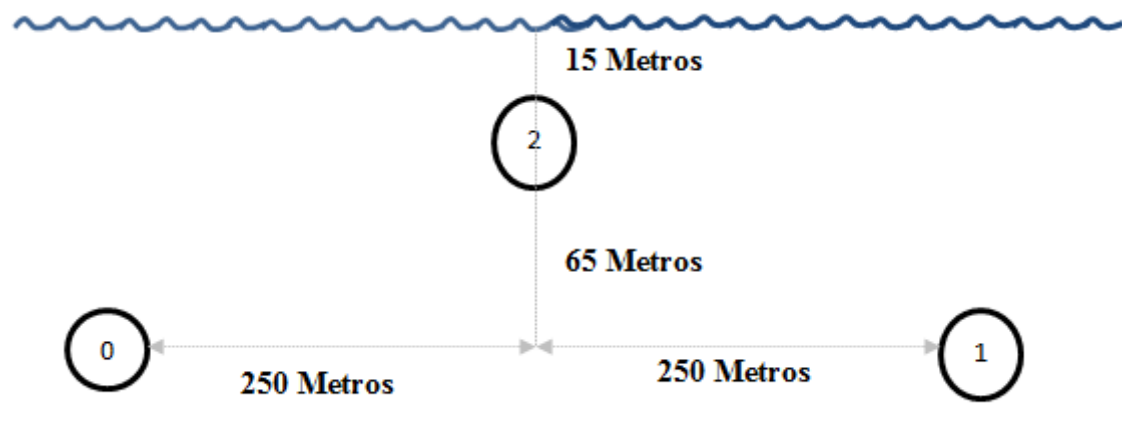
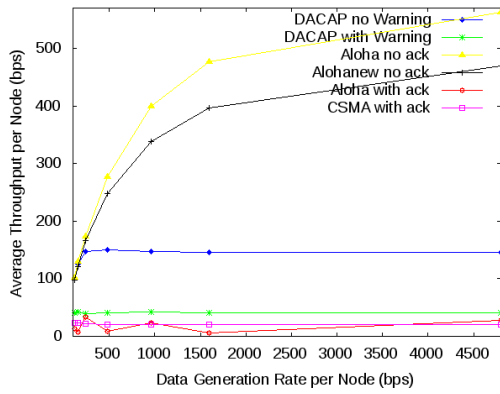


Figura E.3: Cenário utilizado na simulação dos protocolos para tamanho de pacote 50 byte.

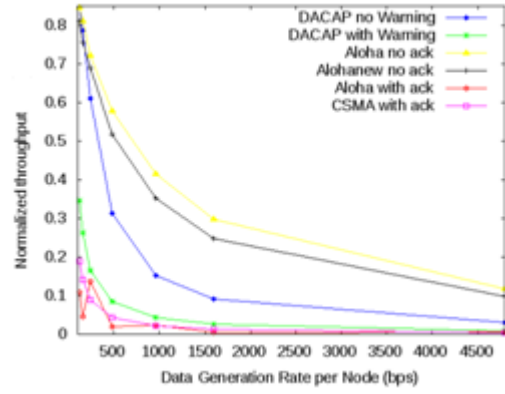
Nó	Latitude	Longitude	Profundidade (metros)	Distância
0	42,59°N	10,125°E	-80 Metros	0 Metros
1	42,59°N	10,131°E	-80 Metros	500 Metros
2	42,59°N	10,128°E	-15 Metros	250 Metros

Tabela E.2: Coordenadas e profundidades dos nós utilizados na simulação utilizando o cenário apresentado na figura E.3.

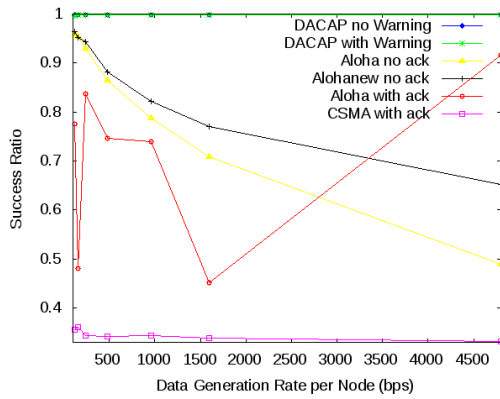
**Teste Average Throughput**



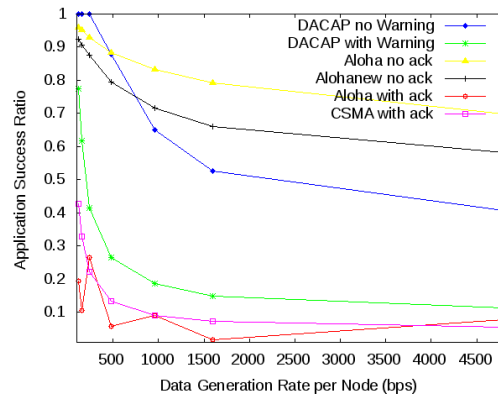
**Teste Normalized Throughput**



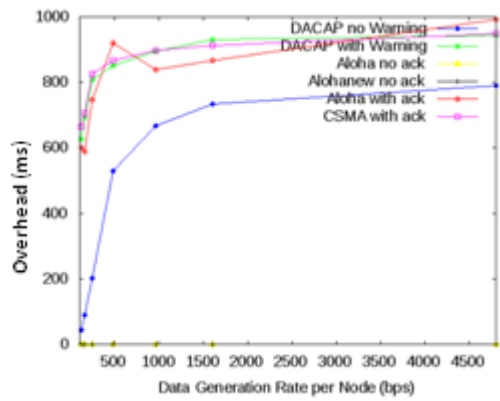
**Teste Sucess Ratio**



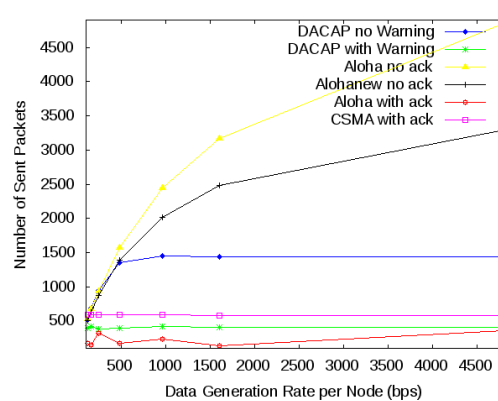
**Teste Aplictattion Success Ratio**



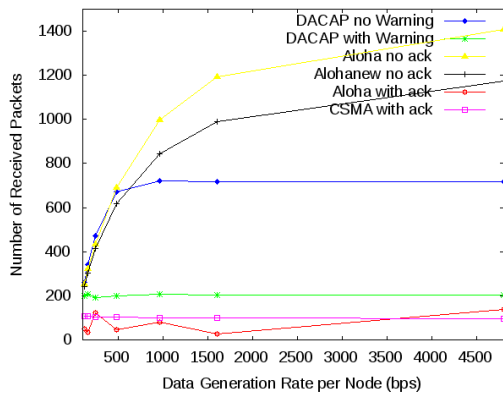
**Teste Overhead**



**Teste Number of Sent Packets**



**Teste Number of Received Packets**



**Teste Application Layer Packet Loss Ratio**

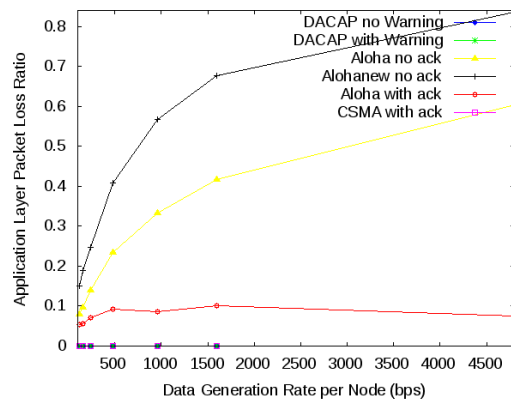


Figura E.4: Resultados obtidos do cenário apresentado na figura E.3.

### E.1.3. Arquitetura assimétrica para tamanho pacote 600 byte

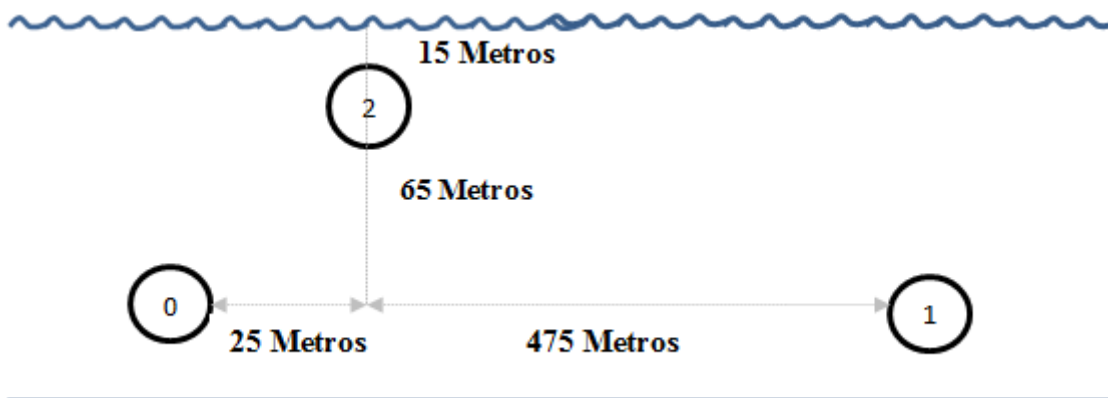
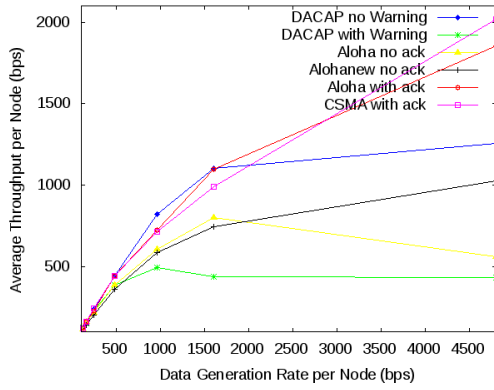


Figura E.5: Cenário utilizado na simulação dos protocolos para tamanho de pacote 600 byte.

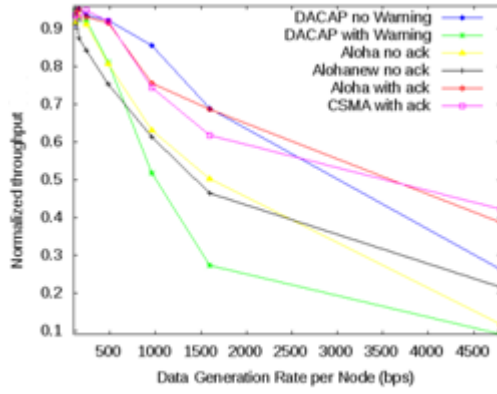
Nó	Latitude	Longitude	Profundidade (metros)	Distância
0	42,59°N	10,125°E	-80 Metros	0 Metros
1	42,59°N	10,131°E	-80 Metros	500 Metros
2	42,59°N	10,125°E	-15 Metros	25 Metros

Tabela E.3: Coordenadas e profundidades dos nós utilizados na simulação utilizando o cenário apresentado na figura E.5.

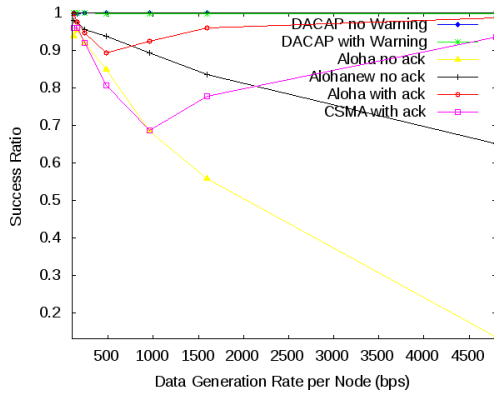
**Teste Average Throughput**



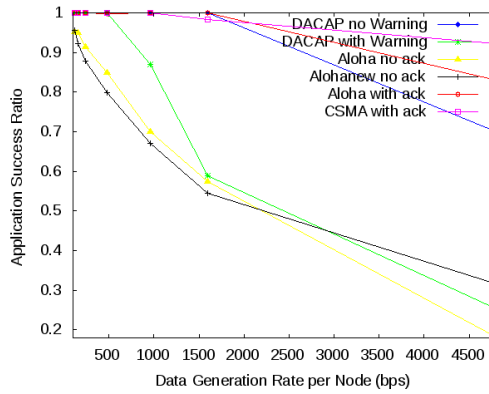
**Teste Normalized Throughput**



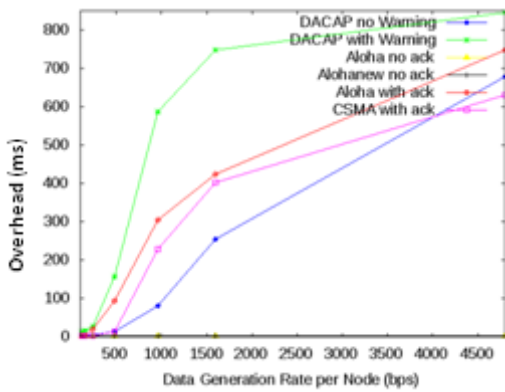
**Teste Sucess Ratio**



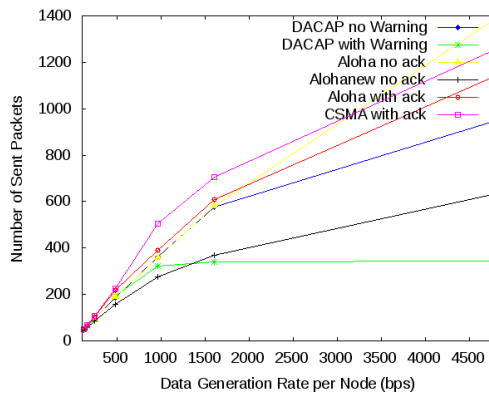
**Teste Applicattion Success Ratio**



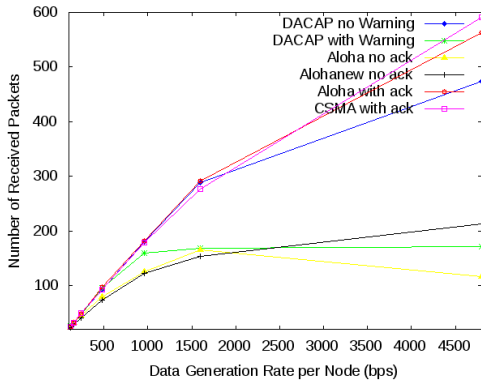
**Teste Overhead**



**Teste Number of Sent Packets**



### Teste Number of Received Packets



### Teste Application Layer Packet Loss Ratio

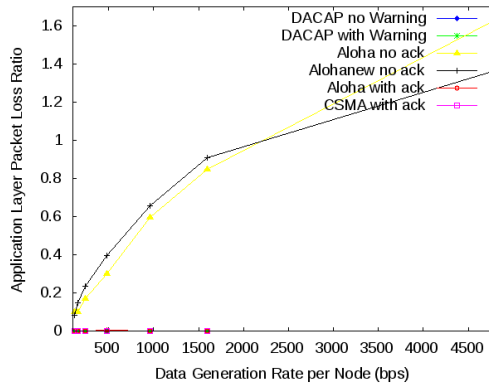


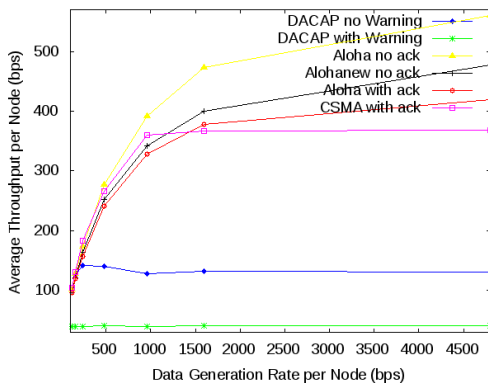
Figura E.6: Resultados obtidos do cenário apresentado na figura E.5

## E.1.4. Arquitetura assimétrica para tamanho pacote 50 byte

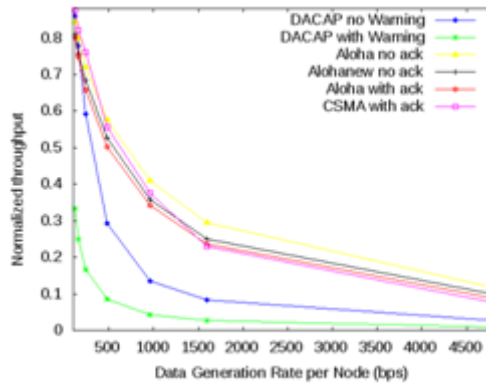
Nó	Latitude	Longitude	Profundidade (metros)	Distância
0	42,59°N	10,125°E	-80 Metros	0 Metros
1	42,59°N	10,131°E	-80 Metros	500 Metros
2	42,59°N	10,125°E	-15 Metros	25 Metros

Tabela E.4: Coordenadas e profundidades dos nós utilizados na simulação utilizando o cenário apresentado na figura E.5.

### Teste Average Throughput



### Teste Normalized Throughput



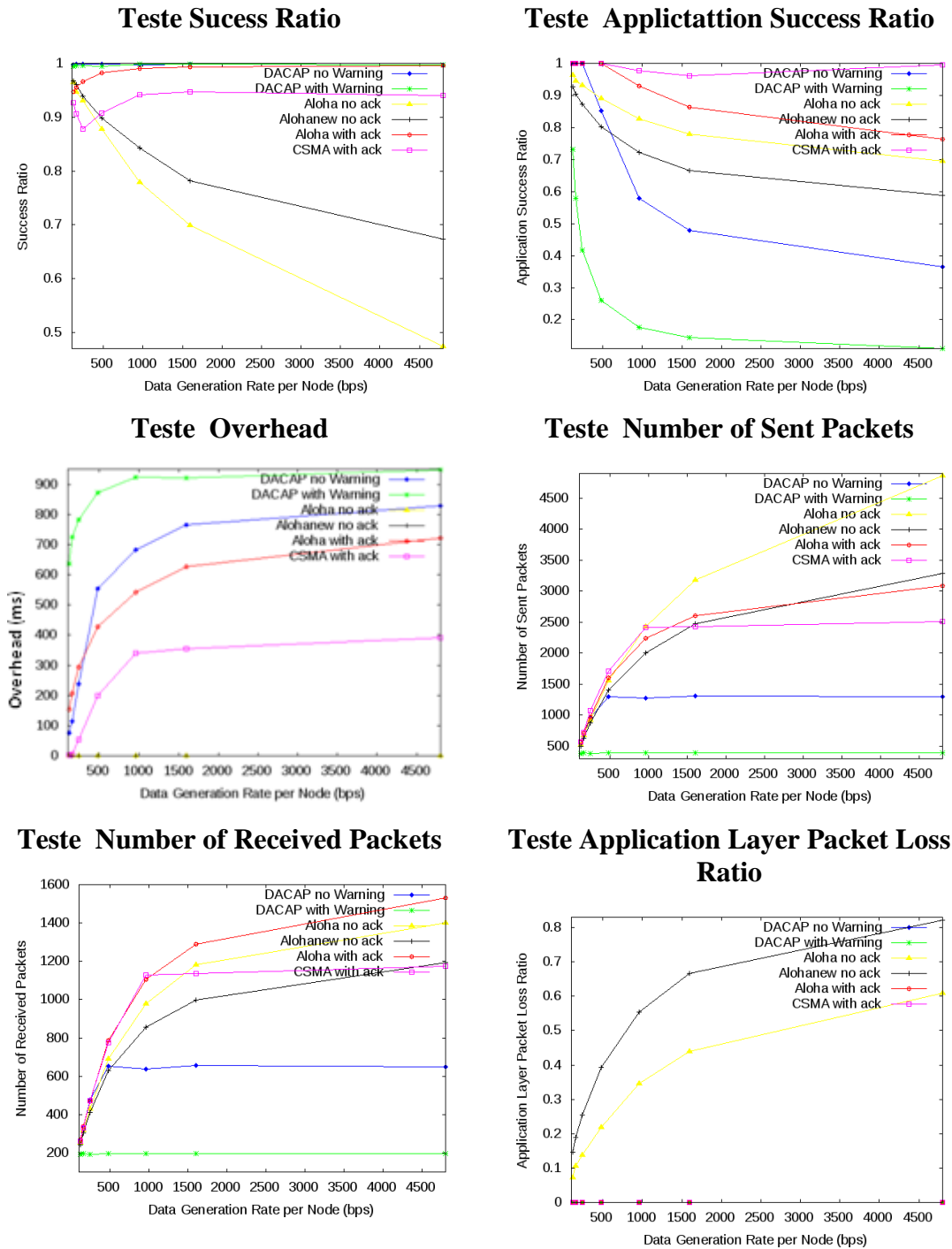


Figura E.7: Resultados obtidos do cenário apresentado figura E.5.

## E.2. Análise comparativa dos protocolos MAC para 4 nós emissores e 1 recetor

### E.2.1. Arquitetura simétrica para tamanho pacote 600 byte

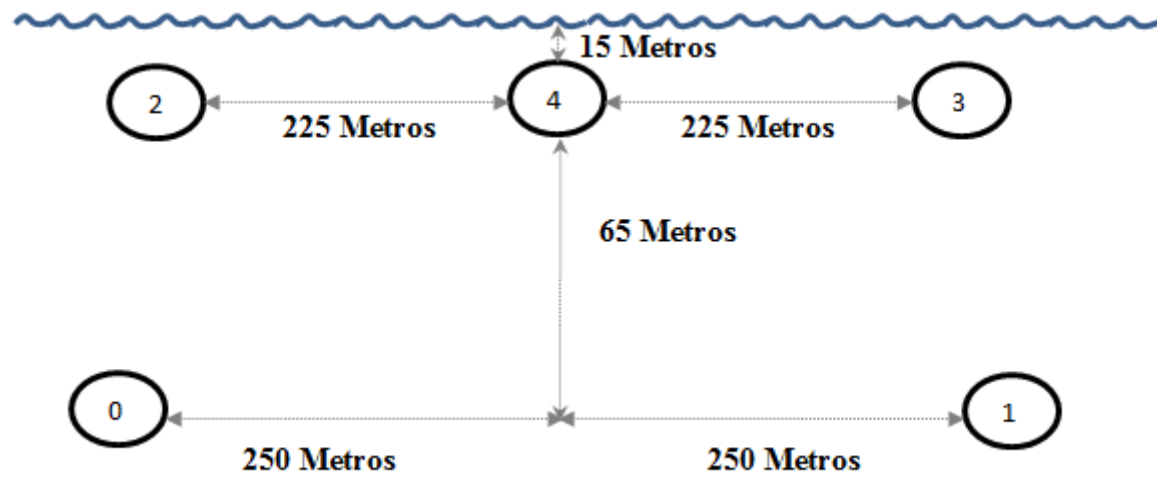


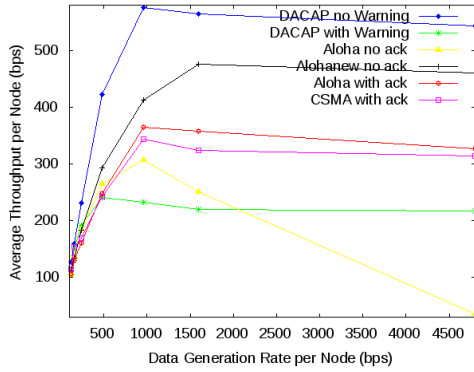
Figura E.8: Cenário utilizado na simulação dos protocolos considerando o nó 4 como recetor.

Nó	Latitude	Longitude	Profundidade (metros)	Distância
0	42,59°N	10,125°E	-80 Metros	0 Metros
1	42,59°N	10,131°E	-80 Metros	500 Metros
2	42,59°N	10,125°E	-15 Metros	25 Metros
3	42,59°N	10,130°E	-15 Metros	475 Metros
4	42,59°N	10,128°E	-15 Metros	250 Metros

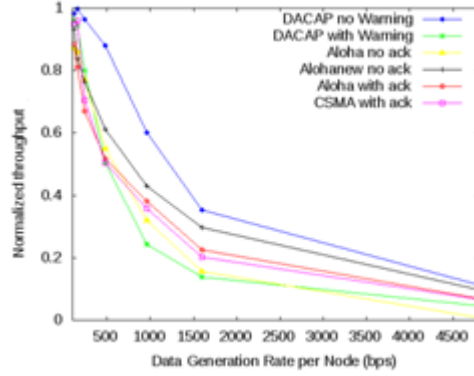
Tabela E.5: Coordenadas e profundidades dos nós utilizados na simulação utilizando o cenário apresentado na figura E.8.



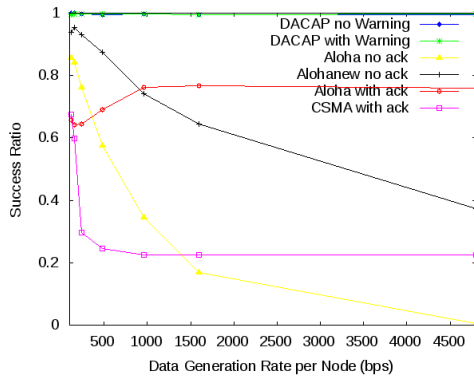
**Teste Average Throughput**



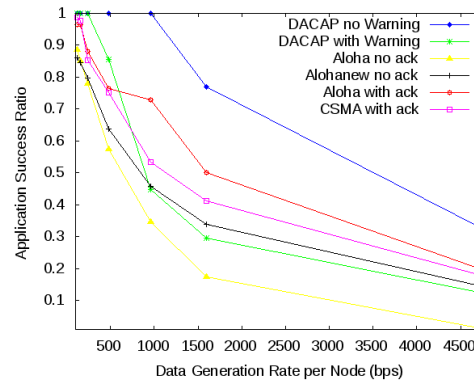
**Teste Normalized Throughput**



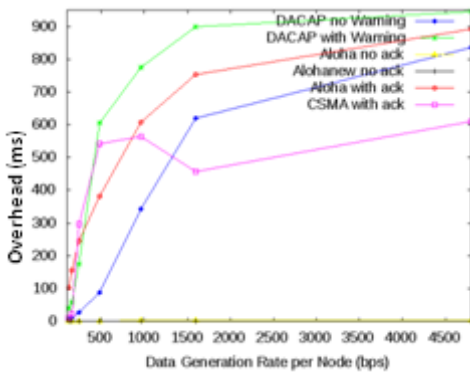
**Teste Sucess Ratio**



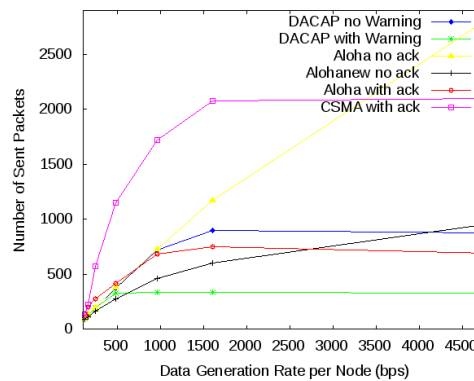
**Teste Applicattion Success Ratio**



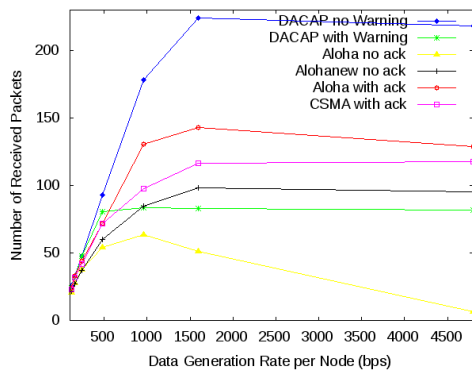
**Teste Overhead**



**Teste Number of Sent Packets**



**Teste Number of Received Packets**



**Teste Application Layer Packet Loss Ratio**

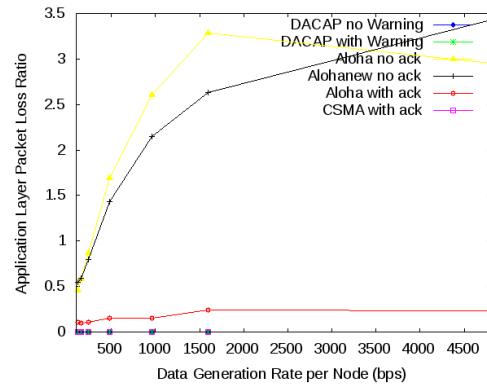


Figura E.9: Resultados obtidos do cenário apresentado na figura E.8.

### E.2.2. Arquitetura assimétrica para tamanho pacote 600 byte

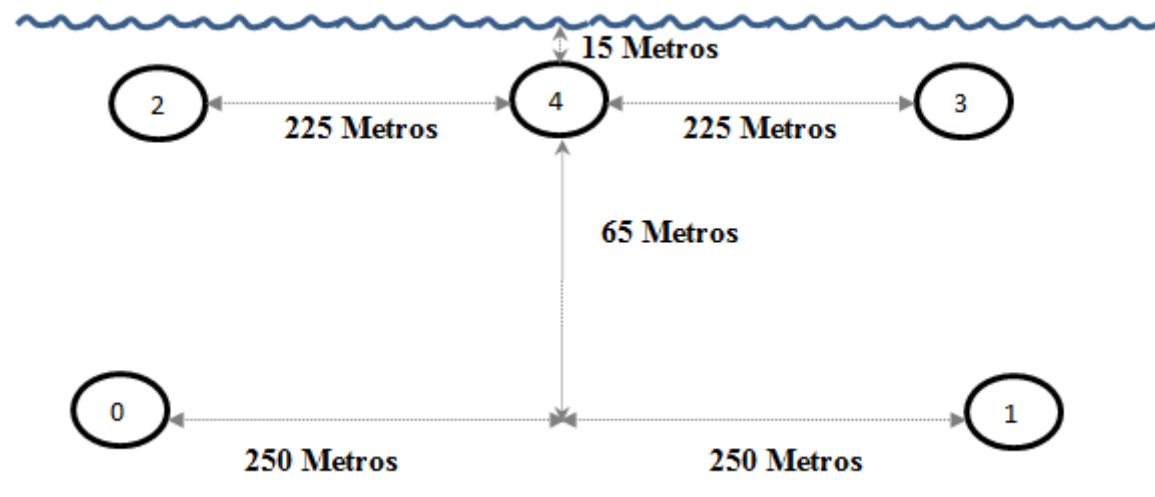
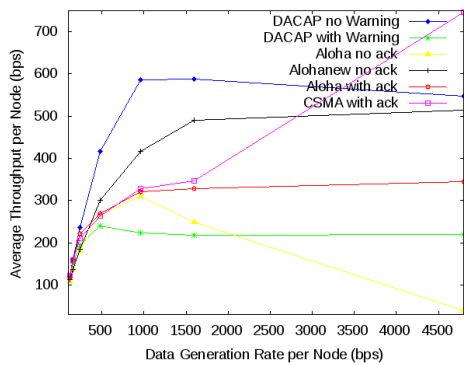


Figura E.10: Cenário utilizado na simulação dos protocolos considerando o nó 2 como recetor.

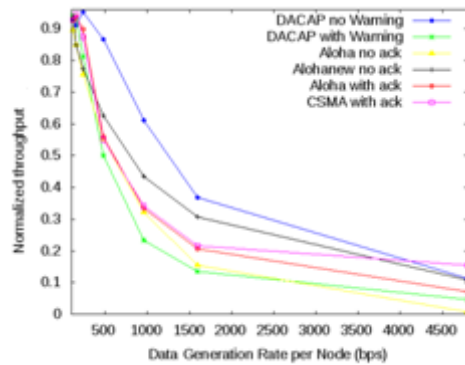
Nó	Latitude	Longitude	Profundidade (metros)	Distância
0	42,59°N	10,125°E	-80 Metros	0 Metros
1	42,59°N	10,131°E	-80 Metros	500 Metros
2	42,59°N	10,128°E	-15 Metros	25 Metros
3	42,59°N	10,130°E	-15 Metros	475 Metros
4	42,59°N	10,125°E	-15 Metros	250 Metros

Tabela E.6: Coordenadas e profundidades dos nós utilizados na simulação utilizando o cenário apresentado na figura E.10.

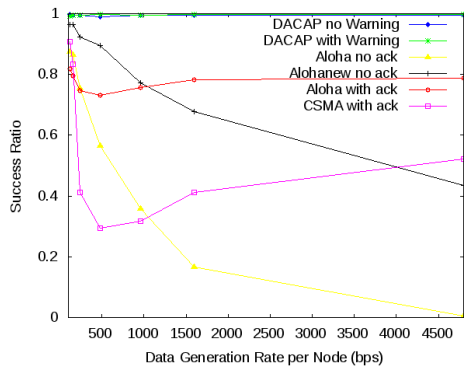
### Teste Average Throughput



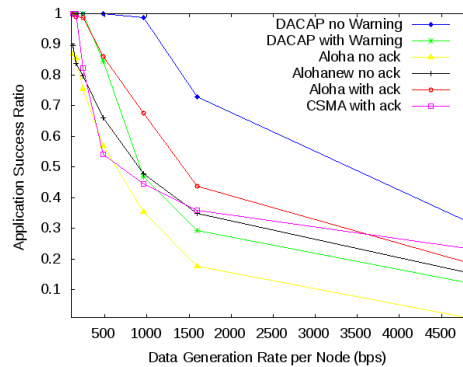
### Teste Normalized Throughput



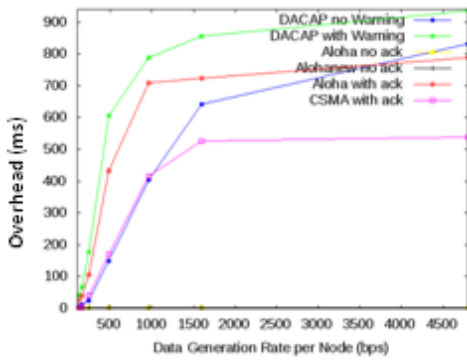
### Teste Sucess Ratio



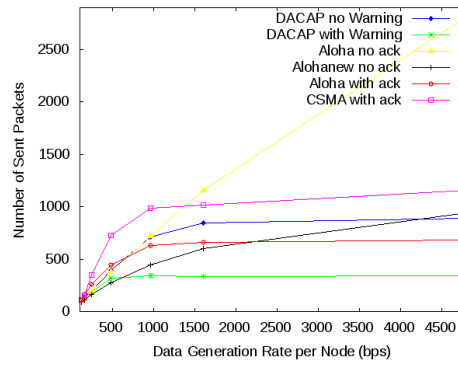
### Teste Applictattion Success Ratio



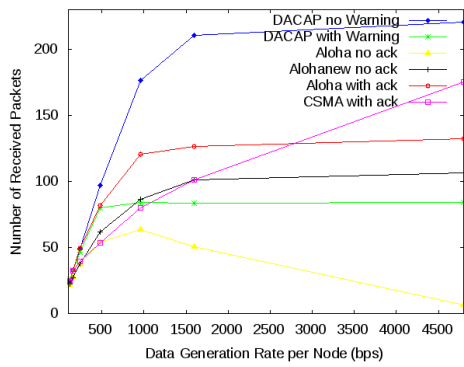
**Teste Overhead**



**Teste Number of Sent Packets**



**Teste Number of Received Packets**



**Teste Application Layer Packet Loss Ratio**

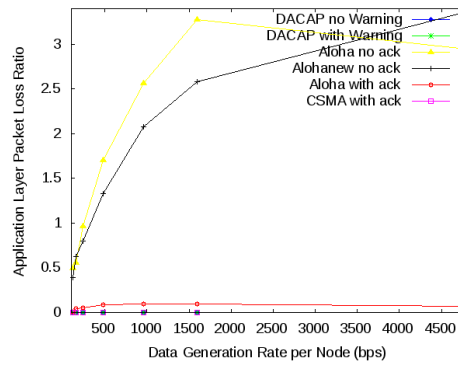


Figura E.11: Resultados obtidos do cenário apresentado na figura E.10.

