

Prototipando Veículos Autônomos Submarinos usando o ns-3

Daniel R. Luna¹, Lívio C. Sousa², Mateus G. da S. Batista³,
Lucas I. C. Medeiros³, Vicente A. de Sousa Jr.^{1*}

¹Programa de Pós-Graduação em Engenharia Elétrica e de Computação (PPGEEC)
Universidade Federal do Rio Grande do Norte (UFRN) – Natal, RN, Brasil

danielro@ufrn.edu.br, vicente@ufrn.edu.br

²Departamento de Exploração e Produção
Petróleo Brasileiro S.A. (Petrobras) – Alto do Rodrigues – RN – Brasil

liviosousa@petrobras.com.br

³Departamento de Engenharia de Comunicações (DCO)
Universidade Federal do Rio Grande do Norte (UFRN) – Natal, RN, Brasil

mateus.gppcom@ufrn.edu.br, lucas.ismael@ufrn.edu.br

Abstract. *Specifically in telecommunications, the underwater environment is challenging and deprived of solutions from those related to the form of transmission, without cables or with cables, to those related to data routing (e.g., in a sensor network). This work proposes to make a collection of different prototyping tools for submarine network solutions involving autonomous submarine vehicles (AUV). A case study using ns-3 is presented. Although it is a mature tool, several adjustments (with one-time implementations) were required to allow a set of functionalities to be added to the same prototyping scenario, such as mobility of specific AUVs, AUV navigation and communication power consumption, TCP/IP model for communication and specific propagation model of the submarine acoustic channel. A performance analysis is presented in terms of useful throughput, packet loss, and power consumption in a simple scenario with two communicating nodes, one of which is an AUV.*

Resumo. *Especificamente em telecomunicações, o meio submarino é desafiador e carente de soluções desde àquelas relacionadas a forma de transmissão, sem cabos ou com cabos, até aqueles relacionados ao roteamento de informações (e.g., em uma rede de sensores). Este artigo se propõe a fazer um apanhado de diferentes ferramentas de prototipagem de soluções para redes submarinas que envolvem veículos autônomos submarinos (AUV). É apresentado um estudo de caso usando o ns-3. Mesmo sendo uma ferramenta madura, foram necessários vários ajustes (com implementações pontuais) para permitir que um conjunto de funcionalidades fossem agregadas a um mesmo cenário de prototipagem, entre elas: mobilidade de AUVs específicos, consumo de energia do AUV e da unidade de comunicação, modelo TCP/IP de comunicação e modelo de propagação específico do canal acústico submarino. Uma análise de desempenho é apresentada em termos de vazão útil, de perda de pacotes e de consumo de energia em um cenário simples com dois nós se comunicando, um deles sendo um AUV.*

*O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

1. Introdução

A exploração oceânica é reconhecida como um passo-chave no sentido de obter um profundo entendimento e sustentabilidade da vida por meio de observações dos diversos sistemas oceânicos [Heidemann 2009], além de servir como rota para o comércio internacional e a indústria de pesca. Isso tudo passa necessariamente pelo desenvolvimento de tecnologias de comunicação adaptadas ao meio em questão, visto que a aplicação das tecnologias já consolidadas podem apresentar desafios de implantação e desenvolvimento. Então, as redes de comunicação submarina, composta de um conjunto de dispositivos imersos, pode ter um papel importante na coleta dados do ambiente, para fins exploratórios e/ou comerciais.

Um dos desafios de telecomunicação é a forte atenuação das ondas eletromagnéticas (EM) em um meio submerso, o que possibilita, nas melhores situações, um alcance de apenas 100 m [Maher et al. 2019]. Por outro lado, a comunicação via ondas acústicas surgiu como solução para o alcance limitado das ondas de rádio-frequência, proporcionando alcance na ordem de alguns quilômetros [Qarabaqi and Stojanovic 2013], por isso, é a abordagem utilizada neste trabalho. Contudo, problemas não presentes nas ondas EM, por exemplo, a pequena largura de banda (na ordem de kHz), a baixa velocidade de propagação e abundância de fontes de ruído criam novos desafios para a comunicação no espectro acústico. Esses novos desafios se aliam a vulnerabilidades quanto os multipercursos e ao efeito Doppler [Qarabaqi and Stojanovic 2013], gerando um canal extremamente variante no tempo (tempo de coerência muito curto) [Qarabaqi and Stojanovic 2013].

Assim, enquanto a solução para comunicação submarina são as ondas acústicas, para monitoramento e exploração do meio oceânico é o desenvolvimento de redes de sensores. Além disso, devido as grandes distâncias e a dificuldade de acesso ao fundo do mar, pode ser necessário o uso de veículos autônomos submarinos, do inglês *Autonomous Underwater Vehicles* (AUV). Em um cenário heterogêneo em que se tem AUVs se comunicando com sensores fixos, e estes se comunicando entre si e com *gateways* na superfície, é possível, por exemplo, citar algumas atividades-fim: 1) monitoramento de fazendas de animais marinhos (peixes, camarões, lagostas, etc), com o intuito de garantir a qualidade dos produtos; 2) exploração de águas profundas para prospecção e monitoramento de poços de petróleo e; 3) aplicações militares e de segurança nacional, por meio do monitoramento das áreas oceânicas pertencentes a uma determinada região.

Por essas razões, o principal objetivo deste trabalho é montar um experimento de redes submarinas com AUVs, possibilitando a análise de estatísticas da rede de comunicação e do consumo de energia. O artigo é organizado com a seguir. A Seção 2 traz um apanhado das ferramentas de prototipagem para redes submarinas e a razão da escolha do ns-3 como ferramenta de prototipagem deste trabalho. Na Seção 3, é apresentado um histórico do módulo de comunicação submarina do ns-3 bem como os ajustes realizados para gerar os resultados deste trabalho. As Seções 4 e 5 apresentam e discutem o cenário de estudo e os resultados, respectivamente. As conclusões e os trabalhos futuros encerram o artigo na Seção 6.

2. Ferramentas de Prototipagem de Redes Submarinas

A prototipagem em software é uma etapa muito importante para concepção e teste de funcionalidades de uma rede de comunicação. A realização de testes em sistemas subaquáticos reais podem necessitar de muito recurso financeiro para que uma rede seja fisicamente implementada em um ambiente que experimente os efeitos do uso real do sistema. Isso dificulta o controle do ambiente de teste, limitando a identificação causa-efeito. Outro aspecto se refere aos testes de escalabilidade, tanto do tamanho da rede como da quantidade

de usuários. O recurso financeiro disponível para montar a rede real limita tais estudos a casos envolvendo poucos enlaces de comunicação.

Já na prototipagem em software de redes de comunicação é possível testar diferentes aspectos do comportamento do canal, modos de tráfego, mobilidade, tamanho da rede, modelos de consumo e fornecimento de energia. Modelar mais ou menos características está ligado a complexidade da ferramenta de simulação, não envolvendo alto custo financeiro para sua construção. Contudo, é fato que alguns testes de escalabilidade necessitam de alto poder computacional para realizar as simulações, mas é um recurso que pode ser reusado em vários testes, não se comparando a montagem física de cada versão do sistema de se deseja prototipar.

Existem várias ferramentas de prototipagem de redes de comunicação disponíveis, como mostra a Tabela 1 [Kabir et al. 2014, Issac et al. 2017]. Além das diferenças de funcionalidades entre os simuladores, como o suporte para redes submarinas, há também diferenças de custo comercial e requisitos mínimos de hardware. Grande parte dos simuladores são escritos em C ou C++, mas interface gráfica é um recurso mais aperfeiçoado em ferramentas comerciais. Mesmo considerando o OMNeT++ uma ferramenta muito atrativa (é *open source*, possui interface gráfica e é multiplataforma), infelizmente, ela não possui um módulo dedicado às redes submarinas.

Tabela 1. Principais simuladores de rede e suas características.

Simulador	Linguagens Suportadas	Tipo de Licença	Interface Gráfica	Sistemas Operacionais	Módulos dedicados às redes submarinas
ns-2	C++ e OTcl	Open Source	Não	Linux	Sim
ns-3	C++ e Python	Open Source	Não	Linux e MAC OS	Sim
OMNeT++	C++	Open Source	Sim	Linux, MAC OS e Windows	Não
OPNET	C e C++	Comercial	Sim	Windows	Não
NetSim	C e Java	Comercial	Sim	Windows	Não

Então, por ser um simulador *open source*, o ns-3 foi escolhido como ferramenta de prototipagem deste trabalho. Ele é simulador destinado a fins educacionais e de pesquisas, que começou a ser desenvolvido em 2006 a partir do ns-2. Além de ser *open source*, sua popularidade se deve a boa documentação¹ e a grande comunidade de discussão e suporte técnico. O ns-3 possui um grande leque de modelos que simulam diferentes aspectos de uma rede baseada em pacotes, por exemplo:

- Vários modelos de tráfego, de mobilidade e de propagação;
- Modelagem das camadas PHY/MAC de sistemas LTE e Wi-Fi, seguindo as especificações técnicas dos seus respectivos padrões;
- Protocolos diversos como Aloha/CW-MAC, IPv4/IPv6, UDP/TCP, e vários algoritmos de roteamento, emulando uma pilha de protocolos real.

Além disso, outra vantagem do ns-3 é a possibilidade de adição de novos módulos, extensões do simulador feitos por terceiros, que acrescentam novas funcionalidades. A Tabela 2, montada a partir de informações presentes em [Das and Thampi 2017, Nayyar and Balas 2019], mostra os módulos de terceiros e oficiais disponíveis para o ns-2 e ns-3. Cada módulo possui sua peculiaridade, como modelos de propagação e protocolos de roteamento. A última coluna da tabela mostra o último ponto de atualização do repositório

Os módulos DESERT e SUNSET estendem o ns-2 com bibliotecas que auxiliam na implementação de protocolos de redes submarinas, mas apenas o primeiro está disponível para download no momento. Quanto ao ns-3, existem os módulos não oficiais: o WOSS (*World Ocean Simulator System*), que utiliza dados ambientais do oceano nos seus modelos, e

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

o Aqua-Sim NG, que é uma reestruturação da sua versão disponível para ns-2. Neste trabalho utilizamos o módulo UAN devido ao mesmo ser um módulo padrão e ser integrado ao código principal do ns-3.

Tabela 2. Módulos e suas Características.

Módulo	Simulador	Modelo de propagação	Protocolos	Última atualização
UAN Framework (oficial) ²	ns-3	Ideal, Thorp e Bellhop	CW-MAC, RC-MAC e ALOHA simples	Setembro de 2018
Aqua-Sim NG ³	ns-3	Thorp e Bellhop	R-MAC, UW-ALOHA, Broadcast-MAC, VBF e DBR	Fevereiro de 2019
WOSS ⁴	ns-3	Urick e Bellhop	ALOHA, T-Lohi e DACAP	Fevereiro de 2018
DESERT ⁵	ns-2	Bellhop	ALOHA, CSMA, T-Lohi e UW-Polling	Julho de 2017
SUNSET ⁶	ns-2	Urick e Bellhop	ALOHA, CSMA, CSMA-ALOHA, Slotted-CSMA, TDMA, T-Lohi e DACAP	-

3. Módulo UAN do ns-3

Nesta Seção são apresentados a evolução do módulo UAN de acordo com as versões do ns-3, o desenvolvimento dos módulos não-oficiais, além das alterações feitas na versão utilizada pelos autores deste trabalho.

O módulo UAN do ns-3 foi disponibilizado a partir da versão ns-3.11. Com apenas algumas funcionalidades implementadas, não houve mais atualizações oficiais desde essa versão. No que se refere a camada PHY, os modelos de propagação *UanPropModelIdeal* e *UanPropModelThorp* foram desenvolvidos. Ambos baseiam-se no modelo de múltiplos percursos TDL (*Tapped Delay Line*), tendo com entrada uma PDP (*Power Delay Profile*), representando a potência relativa e o atraso de propagação de cada percurso. Além disso, para o modelo *UanPropModelIdeal*, não há a perda de percurso (é considerada zero), enquanto o modelo *UanPropModelThorp* implementa a já consolidada aproximação de Thorp como perda de percurso [UAN 2019, Berkhovskikh and Lysanov 1982]. Vale salientar que não há algoritmos dinâmicos de controle de potência, mas existem os seguintes modos de transmissão: *TX*, *RX*, *IDLE* e *SLEEP*, sendo os dois últimos de menor gasto energético. Na documentação do módulo UAN no ns-3⁷, ainda é citado o modelo de propagação Bellhop (*UanPropModelBh*), que utiliza traçado de raios para gerar a resposta ao impulso do canal. Contudo, tal modelo não está integrado ao código do ns-3 e nem há documentação sobre como isso poderia ser feito.

Em relação a camada MAC, os protocolos implementados são o Aloha, *Reservation* (RC-MAC) e o *Contention Window* (CW-MAC). No Aloha, os pacotes são transmitidos assim que são gerados pelo nó, seguindo a implementação tradicional do protocolo [Abramson 1970]. O protocolo RC-MAC implementa reserva de canal, em que a largura de banda é dinamicamente dividida em um canal de controle e de dados [UAN 2019]. Já o CW-MAC utiliza janelas de contenção de forma similar ao *Distributed coordination function* (DCF) do IEEE 802.11 (Wi-Fi). Se o canal estiver ocupado, o nó entra em um estado de espera (*backoff time*), escolhendo um *slot* de tempo futuro para tentar nova transmissão [P.; and Fox 2008].

² <https://www.nsnam.org/wiki/GSOC2010UANFramework>.

³ <https://github.com/rmartin5/aqua-sim-ng>.

⁴ <http://telecom.dei.unipd.it/ns/woss/>.

⁵ <http://desert-underwater.dei.unipd.it/>.

⁶ http://reti.dsi.uniroma1.it/UWSN_Group/index.php?page=sunse.

⁷ <https://www.nsnam.org/docs/models/html/uan.html>.

Sobre exemplos de uso do UAN, há referência ao exemplo *uan-energy-auv.cc* que, infelizmente, não está integrado ao simulador, porém pode ser encontrado em um repositório do *Google Summer of Code (GSoC)*⁸. Neste exemplo, um *SeaGlider* AUV transmite pacotes genéricos de 17 bytes para um *gateway* na superfície a cada 10 s, enquanto submerge verticalmente a 1000 m, e em seguida emerge na superfície. Provavelmente, o motivo pelo qual esse exemplo nunca foi integrado ao código principal do ns-3 é a ausência do módulo de AUV, ainda não presente no simulador e necessário para o exemplo.

Sobre esse assunto, na documentação há ainda referências a modelos de AUVs que não estão oficialmente no ns-3, pois foram desenvolvidos em um módulo separado em 2017, resultando na publicação [Franchi et al. 2017]⁹. Contudo, a compatibilidade do módulo AUV com o ns-3 só é possível com a versão ns-3.27. Para versões mais recentes, ajustes no código são necessárias para compilação dos módulos. Por exemplo, para a versão mais recente do simulador (ns-3.29), é necessário declarar a função *HandleEnergyChanged* no modelo de energia de cada AUV (*SeaGlider*, *Remus* e *Typhoon*).

Por fim, o maior motivo do uso do ns-3.29 para este trabalho foi o suporte da pilha de protocolos TCP/IP completa para redes UAN. Esse suporte foi adicionado a partir da versão ns-3.28 segundo *release*¹⁰. Até então, investigações sobre roteamento e estudos com estatísticas de rede mais detalhadas não poderiam ser realizados. Este é outro bom motivo em pagar o preço de alterar diretamente o código do módulo AUV para integrá-lo ao ns-3.29.

4. Cenário de Estudo

Este trabalho se propõe a reproduzir uma versão modificada do experimento *uan-energy-auv.cc* na versão mais recente do ns-3.29. Como descrito na Seção 3, esse experimento nunca foi integrado ao ns-3 e não funciona em versões mais recentes do simulador. O exemplo original serviu como base para a definição dos melhores parâmetros para o cenário de teste, como mostrado na Tabela 3.

Tabela 3. Principais parâmetros de rede.

Parâmetros de Simulação	
Canal	UanPropModelThorp
Distância horizontal (d) entre o <i>gateway</i> e o AUV	10, 100, 1000, 2000, 3000, 4000, 5000 m
Deslocamento vertical	1000 m
Modelo de AUV	SeaGlider
Modulação	FH-FSK
Protocolo MAC	Aloha
Protocolo de Rede	IPv4
Protocolo de Transporte	UDP
Modelo de erro	Mapeamento de SINR em PER
Tamanho dos Pacotes	29 bytes
Distribuição <i>inter-arrival</i> de pacotes	Poisson ($\lambda = 4.3$)
Tempo de Simulação	86400s (1 dia)
Repetições	1000

Em resumo, o cenário de simulação proposto procura seguir o mais fielmente a descrição do exemplo *uan-energy-auv.cc*¹¹, contudo com as seguintes modificações:

- Inserção de todas as camadas do protocolo TCP/IP (o experimento roda com IPv4);

⁸ <https://codereview.appspot.com/1743057/>.

⁹ Código disponível em <https://codereview.appspot.com/312460043>

¹⁰ Código disponível em <https://www.nsnam.org/releases/ns-3-28/>.

¹¹ Descrição pode ser encontrada na seção *Mobility Model Examples* em <https://www.nsnam.org/docs/release/3.13/models/html/uan.html>.

- O tamanho do pacote de 17 bytes para 29 bytes (novo padrão do ns-3);
- A taxa de geração dos pacotes, anteriormente constante a cada 10s, segue agora um processo poisson de $\lambda = 4.3$;
- Variação da distância horizontal entre transceptores. Mesmo mantendo somente dois nós se comunicando, um *gateway* localizado na superfície e um AUV (SeaGlider), eles são posicionados a uma distância horizontal d um do outro. O AUV movimenta-se, submergindo 1000 m e emergindo à superfície. Enquanto isso, o AUV transmite pacotes para o *gateway* durante 1 dia, considerando o modelo de propagação *UanPropModelThorp* implementado no ns-3.

Para cada *setup* de simulação são feitas 1000 repetições com sementes do gerador de números aleatórios diferentes, objetivando garantir confiabilidade estatística nos resultados apresentados a seguir.

5. Resultados e Discussões

A análise de desempenho apresentada neste artigo é simples e baseada em três indicadores de desempenho: 1) vazão útil (*Throughput*); 2) Taxa de Perda de Pacote (PLR) e; 3) Consumo de Energia do AUV. Todas as curvas de desempenho são feitas em função do aumento da distância horizontal entre os dois nós para o modelo de propagação de *Thorp*. Cada ponto das curvas representa a média da métrica nas 1000 repetições da simulação, em que o intervalo de confiança de 95% também é mostrado.

A Figura 1 mostra o comportamento da vazão útil. Note que a vazão útil é de aproximadamente 16,8 bps nos primeiros mil metros. Com o aumento da distância, a taxa sofre uma redução, variando entre 16,5 bps e 16,22 bps entre 1 km e 4 km, devido a menor potência recebida. Vale salientar que é nos primeiros mil metros que a vazão útil tem uma queda mais acentuada. Ao chegar em 5 km, não há pacotes sendo recebidos.

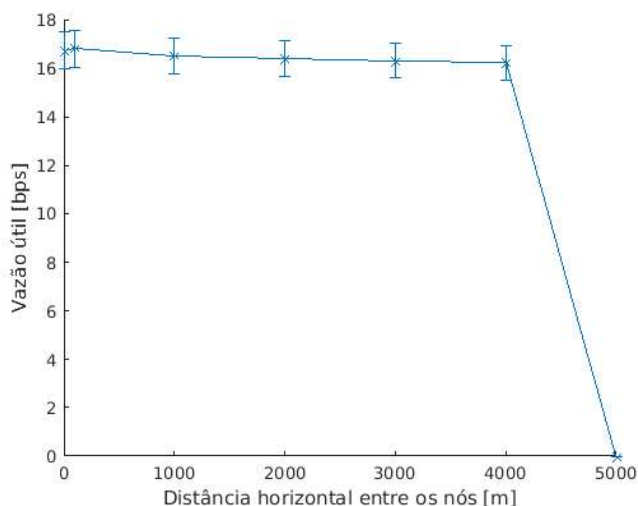


Figura 1. Vazão útil com aumento da distância entre os nós.

A Figura 2 complementa a análise anterior, mostrando a PLR. Assim como no gráfico anterior, nos primeiros mil metros é onde sofre um maior aumento. Entre 1 km 4 km, ela varia entre 26,5 % e 26,81% do total de pacotes transmitidos. Por fim, aos 5 km de distância, a taxa zera, pois não há pacotes sendo recebidos, conforme mostrado na Figura 1.

Na Figura 3 é ilustrado o comportamento do consumo de energia tanto da navegação do AUV quanto do modem de transmissão mais a navegação. Na Figura 3(a), com o aumento

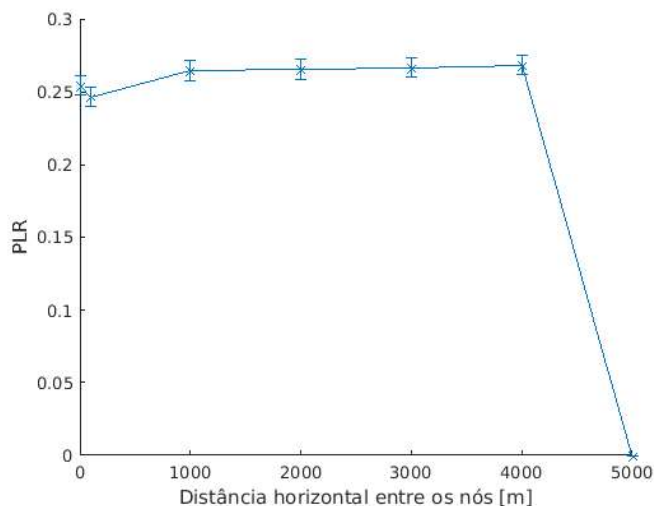
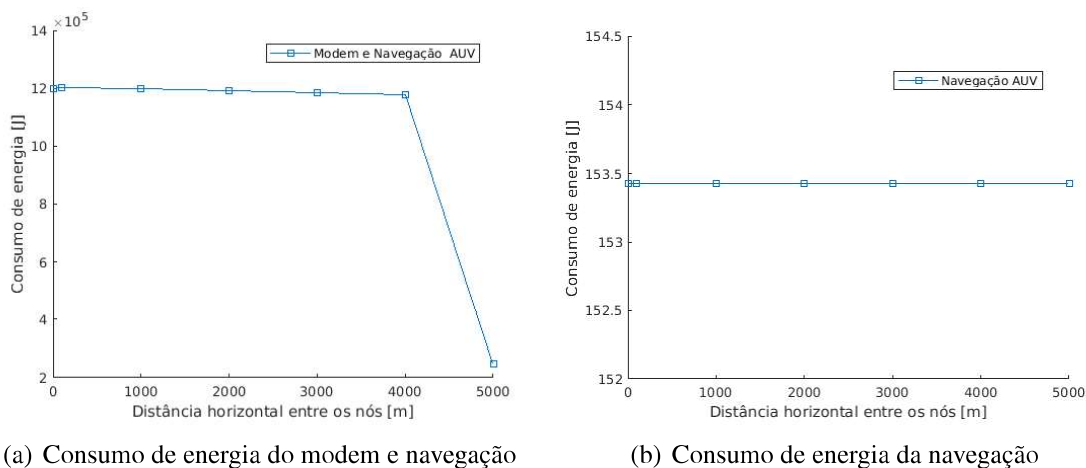


Figura 2. PLR com aumento da distância horizontal entre os nós.



(a) Consumo de energia do modem e navegação

(b) Consumo de energia da navegação

Figura 3. Consumo de energia com aumento da distância horizontal entre os nós

da distância até 4 km, a energia consumida sofre uma leve queda devido a diminuição da vazão útil, isso leva a diminuição do consumo de energia proporcional. Em 5 km, há um decaimento brusco de potência, devido, principalmente, a mudança do modo de transmissão para *IDLE* que possui gasto menor de potência. Embora, a comunicação seja *uplink* (no sentido AUV para *gateway*), ao começar a transmissão e apenas nesse momento, o AUV manda uma requisição ARP para o *gateway*, e este entra em modo *RX*, esperando a resposta do *gateway* com as informações pedidas. Como a distância entre os nós é grande e o canal mais severo, o *gateway* não recebe a requisição ARP e, portanto, não responde. O AUV durante toda a simulação fica alternando entre os estados de transmissão, *TX* (envio do pacote de ARP), *RX* (espera da resposta ao ARP) e *IDLE* (estado de transição, após o fim do tempo de espera de *RX*). Em outras palavras, o consumo de energia mostrado na distância de 5 km refere-se apenas ao consumo de energia do envio de pacotes ARP e espera da resposta pelo *gateway*. Finalizando, na Figura 3(b) é mostrado o gasto com navegação do AUV, como esse movimento independe do envio de pacotes, há contabilização de gasto energético durante toda a simulação e manutenção do valor encontrado.

6. Conclusões

Este trabalho se propôs a: (i) fazer um apanhado de diferentes ferramentas de prototipagem de soluções para redes submarinas que envolvem veículos autônomos submarinos (AUV); e (ii) juntar alguns módulos do UAN concebidos para diferentes versões do ns-3, e ajustar seus códigos para reproduzir o exemplo *uan-energy-auv.cc* com sucesso em uma rede com a pilha de protocolos TCP/IP completa. Uma análise de desempenho simples mostrou que o ns-3 é uma boa opção para testar a concepção de funcionalidades em redes submarinas. Contudo, algum trabalho de ajuste dos modelos precisa ser realizado.

Em futuros trabalhos, serão explorados o comportamento da rede em um cenário heterogêneo, com nós fixos e um número maior de dispositivos AUV, objetivando verificar o desempenho de novos protocolos da camada MAC e roteamento.

Referências

- [Abramson 1970] Abramson, N. (1970). The ALOHA System: Another Alternative for Computer Communications. In *Proceedings of the November 17-19, 1970, Fall Joint Computer Conference, AFIPS '70 (Fall)*, pages 281–285, New York, NY, USA. ACM.
- [Berkhovskikh and Lysanov 1982] Berkhovskikh, L. and Lysanov, Y. (1982). *Fundamentals of Ocean Acoustics*. Springer.
- [Das and Thampi 2017] Das, A. and Thampi, S. (2017). Simulation tools for underwater sensor networks: A survey. *Network Protocols and Algorithms*.
- [Franchi et al. 2017] Franchi, M., Pecorella, T., Ridolfi, A., Fantacci, R., and Allotta, B. (2017). Kinematic Constraints and ns-3 Mobility Models: the AUV Issue. pages 103–109.
- [Heidemann 2009] Heidemann, J., M. U. P. J. S. M. Z. M. . C. L. (2009). Guest editorial - underwater wireless communication networks. *IEEE Journal on Selected Areas in Communications (Volume: 26 , Issue: 9 , December 2008)*.
- [Issac et al. 2017] Issac, A., A. Samad, S., and S. Jereesh, A. (2017). Software tools for simulation and realization of underwater networks. pages 0457–0461.
- [Kabir et al. 2014] Kabir, M., Islam, S., Hossain, M., and Hossain, S. (2014). *Detail Comparison of Network Simulators*. PhD thesis.
- [Maher et al. 2019] Maher, S. M., Ali, Z. M., Mahmoud, H. H., Abdellatif, S. O., and Abdellatif, M. M. (2019). Performance of RF underwater communications operating at 433 MHz and 2.4 GHz. In *2019 International Conference on Innovative Trends in Computer Engineering (ITCE)*, pages 334–339.
- [Nayyar and Balas 2019] Nayyar, A. and Balas, V. (2019). *Analysis of Simulation Tools for Underwater Sensor Networks (UWSNs): Proceedings of ICICC 2018, Volume 1*, pages 165–180.
- [P.; and Fox 2008] P., P. N. T. L. R. S. A. and Fox, W. (2008). System design considerations for undersea networks: Link and multiple access protocols. *IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Underwater Wireless Communications and Networks*.
- [Qarabaqi and Stojanovic 2013] Qarabaqi, P. and Stojanovic, M. (2013). Statistical characterization and computationally efficient modeling of a class of underwater acoustic communication channels. *IEEE Journal of Oceanic Engineering*, 38(4).
- [UAN 2019] UAN, F. (2019). UAN Framework - Model Library. <https://www.nsnam.org/docs/release/3.29/models/html/uan.html>.