GABRIEL DURANTE

**Desenvolvimento De Uma Rede De Sensores Sem Fio
Para Processamento De Sinais Acústicos Marinhos**

São Paulo
2018

GABRIEL DURANTE

Desenvolvimento De Uma Rede De Sensores Sem Fio Para Processamento De Sinais Acústicos Marinhos

Versão Revisada

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do título de Mestre em Ciências

Área de Concentração: Sistemas Eletrônicos

Orientador: Dr. Henrique Estanislau Maldonado Peres

São Paulo

2018

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, _____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catálogo-na-publicação

Durante, Gabriel

Desenvolvimento De Uma Rede De Sensores Sem Fio Para
Processamento De Sinais Acústicos Marinhos / G. Durante -- versão corr. --
São Paulo, 2018.
90 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São
Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.redes de sensores sem fio 2.acústica submarina 3.instrumentação
4.sistemas operacionais em tempo real I.Universidade de São Paulo. Escola
Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t.

AGRADECIMENTOS

À Deus, sobre todas as coisas. Ao Prof. Henrique Estanislau Maldonado Peres, meu orientador e ao Prof. Wesley Beccaro pela dedicação, pela preocupação em me manter motivado, pelo profissionalismo, pelo grande exemplo de excelência acadêmica, pela amizade e reconhecimento desde os tempos de graduação.

Agradeço ao Prof. Francisco Javier Ramirez Fernandez pelo convite para minha participação no projeto de Auxílio Pesquisa FAPESP No. 2014/50174-8, suas contribuições em artigos em conjunto, críticas e ideias.

Agradeço ao Prof. Linilson Rodrigues Padovese, pela grande ajuda prestada no decorrer deste trabalho com sugestões valiosas. Sem isso, não teria sido possível concluir o trabalho desta forma.

Um agradecimento especial aos meus familiares pela grande ajuda ao longo dos anos e pela preocupação quanto a minha formação acadêmica e realização profissional.

RESUMO

DURANTE, Gabriel. **Desenvolvimento de Uma Rede de Sensores Sem Fio para Processamento de Sinais Acústicos Marinhos**. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2018.

O crescente interesse pela caracterização de parâmetros acústicos marinhos esbarra nos poucos dispositivos eletrônicos que permitem capturar, armazenar, processar e difundir dados. Os equipamentos atuais que possibilitam obter dados referentes à vida marinha e permitem estudar as influências humanas sobre ela possuem alto custo e funcionam de forma isolada, ou seja, normalmente não há comunicação com outros dispositivos, o que dificulta a monitorização de eventos em tempo real. Sendo assim, neste trabalho são revisadas as características necessárias para estabelecer uma rede de sensores sem fio (RSSF) de alta confiabilidade, utilizando-se o conceito de Internet das Coisas (IoT). São comparados os protocolos MQTT-SN e CoAP com ferramentas de simulação e implementada uma rede de sensoriamento acústico aquático constituída de três nós sensores autônomos e um *gateway* usando o protocolo 6LoWPAN. Também é implementada a tradução para o protocolo IPV4 no gateway, de modo que o funcionamento da rede de sensores pode ser monitorada num servidor remoto na Internet em tempo real. O desempenho da rede é avaliado com relação à autonomia energética, capacidade de memória, alcance e confiabilidade na entrega dos dados. As unidades autônomas desenvolvidas podem operar ininterruptamente, com uso de baterias, por aproximadamente 125 horas com memória de gravação de 48GB e alcance de 100m aproximadamente. Por fim, é implementado o pré-processamento dos dados sonoros visando a detecção de eventos acústicos aquáticos.

Palavras-chave: Redes de sensores sem fio, Internet das coisas, 6LoWPAN, Parâmetros acústicos marinhos

ABSTRACT

DURANTE, Gabriel. **Development of a Wireless Sensor Network for Marine Acoustic Signals Processing**. Dissertation (Master) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2018.

The increasing interest in the characterization of marine acoustic parameters needs electronic devices that allow capture, store, process, and diffuse data. Current equipment that provides data about the marine life and allow the study of human influences on it have high cost and work isolated, that is, without communication with other devices, which makes it difficult to monitor events in real time. Therefore, this work revises the necessary features to establish a high reliability wireless sensor network (WSN) by applying the concept of Internet of Things (IoT). The protocols MQTT-SN and CoAP are compared with simulation tools and a network is implemented to sense acoustic signals, composed by three autonomous sensor nodes and a gateway using the protocol 6LoWPAN. The translation for the IPV4 protocol in the gateway is also implemented, so the operation of the sensor network can be monitored by a remote server on the Internet in real time. The performance of the network is evaluated in relation to power autonomy, memory capacity, reach and reliability. The autonomous sensor nodes can operate uninterrupted, with use of batteries, for approximately 125 hours with 48GB storage for recording and approximately 100m range. Finally, data pre-processing is implemented in order to detect aquatic acoustic events.

Keywords: *Wireless sensor network, Internet of things, 6LoWPAN, Marine acoustic parameters*

LISTA DE FIGURAS

Figura 1 – Temperatura superficial global (°C) no mês de Agosto de 2011.	19
Figura 2 – Perfis médios típicos da variação de temperatura com a profundidade para diferentes latitudes em oceano aberto.	19
Figura 3 – Salinidade superficial média global (em PSU - <i>Practical Salinity Unit</i>). .	20
Figura 4 – Perfis típicos de salinidade em função da profundidade em latitudes polares e tropicais.	21
Figura 5 – Pilha do protocolo IEEE 802.15.4 e camadas superiores.	28
Figura 6 – Pilha de Protocolos e Serviços de cada nó sensor e da estação base. . .	28
Figura 7 – Padrão IEEE 802.15.4: Topologias de rede compatíveis.	30
Figura 8 – Arquitetura RPL.	35
Figura 9 – Arquitetura MQTT-SN.	37
Figura 10 – Arquitetura CoAP.	38
Figura 11 – Diagrama macro da solução de sensoriamento proposta.	40
Figura 12 – Arquitetura do sistema de sensoriamento acústico.	41
Figura 13 – Arquitetura da unidade autônoma de aquisição.	42
Figura 14 – Unidade autônoma construída.	44
Figura 15 – Arquitetura macro do <i>software</i> desenvolvido.	45
Figura 16 – Arquitetura do <i>Ring Buffer</i> de Áudio.	46
Figura 17 – Arquitetura de <i>software</i> do detector.	47
Figura 18 – Arquitetura do <i>hardware gateway</i> desenvolvido.	48
Figura 19 – Arquitetura de <i>software</i> do <i>gateway</i>	49
Figura 20 – Arquitetura de <i>software</i> do servidor.	50
Figura 21 – Proposta de interface no servidor remoto para apresentação dos dados recebidos da rede.	51
Figura 22 – Ambiente de Simulação COOJA.	53
Figura 23 – Resultados de latência utilizando os protocolos MQTT-SN e CoAP. . .	55
Figura 24 – Comparação entre os tempos de tráfego na rede das arquiteturas MQTT- SN e CoAP.	56
Figura 25 – Comparação entre as quantidades de pacotes trafegados utilizando os protocolos MQTT-SN e CoAP.	57
Figura 26 – Comparação entre as quantidades de bytes trafegados utilizando os protocolos MQTT-SN e CoAP.	57
Figura 27 – Comparação entre os consumos de energia dos nós sensores utilizando os protocolos MQTT-SN e CoAP.	58
Figura 28 – Testes conduzidos na Raia Olímpica da USP.	60
Figura 29 – Nível de Sinal RSSI em função da distância entre o nó sensor e o <i>gateway</i> . .	61

Figura 30 – Taxa de erro na recepção do sinal para uma distância fixa de 50 metros.	61
Figura 31 – Tempo de armazenamento em função da memória disponível, para diferentes taxas de amostragem do sinal.	62
Figura 32 – Cenários dos testes RPL.	63
Figura 33 – Sinais e Espectrogramas dos arquivos de áudio analisados.	64
Figura 34 – Sinais e Valores RMS dos sinais de áudio analisados.	65

LISTA DE TABELAS

Tabela 1	– Pressão acústica para diversos tipos de transdutores.	23
Tabela 2	– Comparação das principais características de monitores acústicos passivos.	25
Tabela 3	– Métodos de Detecção de eventos acústicos e características	26
Tabela 4	– Bandas de Frequência do IEEE 802.15.4.	27
Tabela 5	– Padrões IEEE 802.15.4 mais usados.	29
Tabela 6	– Análise comparativa dos protocolos CoAP e MQTT-SN.	39
Tabela 7	– Análise comparativa do desempenho dos protocolos CoAP e MQTT-SN a partir das simulações para uma rede com 40 nós.	59
Tabela 8	– <i>Rankings</i> e rotas RPL, cenários A e B	63

LISTA DE ABREVIATURAS E SIGLAS

6LoWPAN - *IPv6 over Low-Power Wireless Personal Area Networks*

ADC - *Analog to Digital Converter*

AODV - *Ad Hoc On-Demand Distance Vector*

BLE - *Bluetooth Low Energy*

BPSK - *Binary Phase Shift Keying*

CCA - *Clear Channel Assessment*

CoAP - *Constrained Application Protocol*

CODEC - *Encoder Decoder*

CPU - *Central Processing Unit*

CRC - *Cyclic Redundancy Check*

CSMA - *Carrier Sense Multiple Access*

DAG - *Directed Acyclic Graph*

DAO - *Destination Advertisement Object*

DIO - *DODAG Information Object*

DIS - *DODAG Information Solicitation*

DMA - *Direct Memory Access*

DODAG - *Destination-Oriented DAG*

DSP - *Digital Signal Processor*

DSSS - *Direct Sequence Spread Spectrum*

FDMA - *Frequency Division Multiple Access*

FLAC - *Free Lossless Audio Codec*

FPU - *Float Point Unit*

HAL - *Hardware Abstraction Layer*

IEEE - *Institute of Electrical and Electronics Engineers*

IETF - *Internet Engineering Task Force*

IoT - *Internet of Things*

I2C - *Inter-Integrated Circuit*

I2S - *Inter-IC Sound*

ISM - *Industrial, Scientific, and Medical*

JSON - *JavaScript Object Notation*

LLN - *Low-power and Lossy Networks*

LPL - *Low Power Listening*

M2M - *Machine-to-Machine*

MAC - *Medium Access Control*

MQTT - *Message Queue Telemetry Transport*

MQTT-SN - *Message Queue Telemetry Transport for Sensor Networks*

OSI - *Open Systems Interconnection*

PAM - *Passive Acoustic Monitor*

PHY - *Physical Layer*

PZT - *Lead Zirconate Titanate*

PSU - *Practical Salinity Unit*

QoS - *Quality of Service*

QPSK - *Quadrature Phase Shift Keying*

RAM - *Random Access Memory*

REST - *Representational State Transfer*

RF - *Radio Frequência*

RMS - *Root Mean Square*

ROM - *Read-only Memory*

RPL - *Routing Protocol for Low Power and Lossy Networks*

RSSI - *Received Signal Strength Indicator*

RSSF - Redes de Sensores Sem Fio

RTOS - *Real-Time Operating System*

SDRAM - *Synchronous Dynamic Random Access Memory*

SO - Sistema Operacional

TCP - *Transmission Control Protocol*

TDMA - *Time Division Multiple Access*

UDP - *User Datagram Protocol*

USB - *Universal Serial Bus*

WSN - *Wireless Sensor Networks*

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Objetivos	15
1.2	Estrutura do Trabalho	15
2	ACÚSTICA SUBMARINA E TECNOLOGIAS DE REDES DE SENSORES SEM FIO	17
2.1	Fundamentos de Acústica Submarina	17
2.1.1	Variação da temperatura	18
2.1.2	Variação da salinidade	20
2.1.3	Variação da densidade	21
2.1.4	O oceano como meio de propagação do som	22
2.1.5	Transdutores para aplicações submersas	22
2.1.6	Equipamentos de monitoramento acústico	24
2.1.7	Deteção de eventos acústicos	24
2.2	Redes de Sensores Sem Fio	26
2.2.1	Tipos de arquitetura	27
2.2.2	Sistemas operacionais para RSSF	30
2.2.3	Evolução da camada de controle de acesso ao meio (MAC) . .	33
2.2.4	Protocolos de roteamento	34
2.2.5	Protocolos de comunicação	36
2.2.5.1	Protocolo MQTT-SN (<i>Message Queuing Telemetry Transport for Sensor Networks</i>)	37
2.2.5.2	Protocolo CoAP (<i>Constrained Application Protocol</i>)	38
3	MATERIAIS E MÉTODOS: DESENVOLVIMENTO DA REDE DE SENSORES SEM FIO	40
3.1	Unidade Autônoma de Aquisição	41
3.1.1	<i>Hardware</i> da unidade autônoma de aquisição	42
3.1.2	<i>Software</i> da unidade autônoma de aquisição	44
3.1.3	Pré-processamento dos sinais acústicos	46
3.2	<i>Gateway</i>	47
3.2.1	<i>Hardware</i> do <i>gateway</i>	47
3.2.2	<i>Software</i> do <i>gateway</i>	48
3.3	Servidor Remoto	49
3.3.1	<i>Software</i> do servidor remoto	50

4	AVALIAÇÃO DO DESEMPENHO DA REDE DE SENSO- RES: RESULTADOS E ANÁLISE	53
4.1	Análise e Comparação dos Protocolos de Comunicação	53
4.2	Análise do Enlace de Rádio	58
4.3	Avaliação de Autonomia e Armazenamento	60
4.4	Avaliação de Roteamento da Rede	62
4.5	Avaliação do Pré-processamento de Sinais Acústicos	64
5	CONCLUSÕES	66
	REFERÊNCIAS BIBLIOGRÁFICAS	68
	ANEXO A – DATASHEETS	73
A.1	Transceiver RF AT86RF212B	74
A.2	Codec de Audio WM8731	75
A.3	Kit de Desenvolvimento 32F429IDISCOVERY	76
A.4	Banco de Baterias PN-968	77
	ANEXO B – TRABALHOS GERADOS	78
B.1	Trabalho Apresentado - CBO 2016	79
B.2	Trabalho Apresentado - MOMAG 2016	80
B.3	Trabalho Aceito para Publicação - IEEE LA 2017	84

1 INTRODUÇÃO

Fontes de som subaquático podem ser provenientes de inúmeros eventos, tais como atividades sísmicas, fenômenos de superfície do mar como ondas de superfície, turbulência em grande escala, atividades biológicas, atividades humanas *off-shore*, tráfego naval, pesca, explosões e acidentes subaquáticos, por exemplo (BJØRNØ, 1989). Tanto a detecção passiva quanto a ativa se tornaram interessantes por possibilitar o estudo desses eventos de interesse para avaliar o impacto ambiental e auxiliar o planejamento de atividades relacionadas ao estudo marinho. Dessa forma, a análise acústica marinha pode ser importante ferramenta nas áreas de engenharia naval industrial ou militar, pesca ou nas áreas de biologia marinha e ambiental.

Quando um objeto irradia ou reflete um sinal acústico em meio aquático, informações sobre a sua localização podem ser obtidas a partir do tempo de viagem do sinal para os sensores de uma matriz (array) de observação (SCHULTHEISS; WAGNER, 1989). Fisicamente essas matrizes estão limitadas em questão da sua localização, tornando assim o monitoramento em grandes distâncias complexo, caro e algumas vezes inviável uma vez que o espaçamento entre os sensores pode ser acima de 50 m (MISRA et al., 2017). Além disso, em geral, cada elemento da matriz é apenas responsável pela aquisição e armazenamento dos sinais capturados pelos transdutores.

Dentre os sistemas de detecção passiva aquáticos comparados por Sousa-Lima et al. (2013) quase nenhum deles apresenta conexão externa seja com ou sem fios para possibilitar o envio de informações da unidade ou dados de pré processamento dos sinais adquiridos. Neste caso os dados precisam ser coletados nos locais onde os equipamentos estão instalados, o que dificulta a análise de fenômenos aquáticos. Observando-se essa lacuna, pretende-se juntar as tecnologias emergentes de redes de sensoriamento sem fio e processamento de sinais, extendendo assim o uso de aplicações convencionais.

Redes de sensores sem fio são compostas por um grande número de sensores que possuem conectividade *wireless* e esses sensores formam uma rede autônoma para os dados que necessitam de transporte até outros dispositivos. Os recursos dos dispositivos sensores são frequentemente restringidos devido à limitação de energia, normalmente realizado por baterias e painéis solares. Outros fatores são limitantes como o tamanho físico e limite de custo por dispositivo, aumentando a complexidade do sistema. Também é importante considerar os protocolos de comunicação de dados adequados para a aplicação desejada (DUNKELS; GRONVALL; VOIGT, 2004). Os sistemas operacionais voltados para rede de sensores sem fio possuem características que facilitam o desenvolvimento e adaptação do código base para diversas plataformas e o modelo de programação também se adapta

para processar eventos em tempo real, bem como agendados, mesmo o processador tendo limitações de memória (RAM/ROM) e tipos de instruções (HAHM et al., 2016).

Como uma tendência atual, destaca-se o conceito de “Internet das Coisas” (IoT – *Internet of Things*) que pode ser definido como uma instância de uma classe de redes *low-power* and *lossy networks* (LLN’s) que permite que sensores, atuadores e objetos físicos sejam conectados à internet (WINTER, 2012). Exemplos incluem medidores de energia em redes inteligentes, sensores de estacionamento e sensores de temperatura. A informação deve ser transportada entre diversos dispositivos e integrar-se a serviços na nuvem, passando por uma rede de larga escala emergente de comunicação que permite interoperabilidade com os sistemas já existentes. Idealmente, para reduzir o custo e complexidade do sistema como um todo, dispositivos *proxy* ou *gateways* devem ser usados somente quando houver uma necessidade explícita (DUNKELS et al., 2012).

Dessa forma, ressalta-se a importância do desenvolvimento de redes de sensores sem fio que sejam adequadas às peculiaridades do monitoramento acústico subaquático.

1.1 OBJETIVOS

O objetivo principal deste trabalho é desenvolver uma rede de sensores sem fio que permita a aquisição de sinais acústicos em ambiente aquático, bem como o pré-processamento e envio dos dados processados em tempo real.

Entre os objetivos secundários estão:

- Viabilizar o uso da tecnologia de rede sem fio para pesquisa e desenvolvimento com baixo custo.
- Permitir transferência de mensagens em tempo real, relacionadas a eventos acústicos, entre os nós sensores e sistema de visualização de dados.
- Permitir a pré-análise de dados simultaneamente à aquisição dos sinais, sem a necessidade de interferir no funcionamento das unidades autônomas.
- Aplicar algoritmos de detecção de eventos acústicos em tempo real.

1.2 ESTRUTURA DO TRABALHO

Este trabalho está organizado da seguinte forma:

- No capítulo 2 é apresentada a revisão bibliográfica de parâmetros importantes para caracterização de ambientes marinhos, é feita uma breve descrição de equipamentos autônomos que fazem monitoramento acústico passivo (*Passive Acoustic Monitoring* - PAM), sistemas embarcados microprocessados, tecnologia de rede sem fio *IPv6 over*

low power wireless personal area networks (6LoWPAN) e sua evolução tecnológica, sistemas operacionais em tempo real e aspectos do processamento de sinais aplicado à acústica submarina.

- No capítulo 3 é apresentado o sistema desenvolvido e os protocolos implementados para transferência de dados, bem como o algoritmo para detecção de eventos acústicos.
- No capítulo 4 são apresentados e discutidos os resultados obtidos nos testes para caracterização do enlace de comunicação via rádio sub 1 GHz e testes de autonomia da unidade. Também são apresentados testes do pré-processamento dos dados acústicos.
- No capítulo 5 são apresentadas as conclusões do trabalho e perspectivas de continuidade de estudo.

2 ACÚSTICA SUBMARINA E TECNOLOGIAS DE REDES DE SENSO-RES SEM FIO

O segmento oceanográfico tem como variáveis de interesse relevantes propriedades da água do mar como por exemplo: temperatura, salinidade, pressão, nível do mar, correntes marítimas, marés, ondas, propriedades do ar sobre os oceanos e propagação do som submarino, dentre outras (GODOI, 2010).

Dentre as aplicações atuais de instrumentação acústica submarina, percebe-se uma lacuna em relação à autonomia e intercomunicação entre as unidades sensoras. Elas normalmente ficam isoladas e muitas vezes não levam em conta aspectos de tecnologias emergentes e por consequência, o consumo, complexidade e praticidade do uso nem sempre são abordados.

Neste contexto, a utilização de redes de sensores sem fio (RSSF ou WSN - *Wireless Sensors Network*) tem como objetivo habilitar a instalação de sensores em qualquer configuração de distribuição permitindo a interconexão entre equipamentos, aumentando assim as possibilidades de diferentes aplicações.

2.1 FUNDAMENTOS DE ACÚSTICA SUBMARINA

Ondas acústicas, também denominadas ondas sonoras, são ondas mecânicas que necessitam de um meio de propagação. São classificadas como ondas de pressão uma vez que seu mecanismo de propagação consiste na alternância regular de pressão em um meio, uma sucessão de compressões e rarefações que se propagam como onda (TIPLER, 2009). São caracterizadas pelo seu comprimento de onda, amplitude e frequência.

As ondas sonoras podem se propagar por meios sólidos, líquidos ou gasosos. É necessário considerar que o meio de propagação é o fator principal que condiciona tanto a velocidade quanto a propagação da onda sonora. Quanto maior for a densidade do meio, maior será a velocidade do som. Valores típicos de propagação apontam que no ar o som apresenta uma velocidade aproximada de 340 m/s (em temperatura ambiente), na água em torno de 1500 m/s e em sólidos de 5200 m/s (no caso do ferro) (TIPLER, 2009).

Tratando-se de um meio em aberto, o som é transmitido de forma muito mais eficiente na água do que em qualquer outro meio gasoso (TIPLER, 2009). Consequentemente utilizam-se ondas acústicas no mar porque as ondas eletromagnéticas sofrem muita atenuação nesse meio, sendo inviáveis para telecomunicações.

Apesar das ondas acústicas dominarem os sistemas de comunicação em ambiente marinho, esses são extremamente restritos se comparados aos sistemas de comunicação efetuados por ondas eletromagnéticas no ar, sendo que as limitações do método acústico

estão relacionadas com a baixa velocidade de propagação do som no ambiente marinho (em relação à velocidade de propagação da luz no ar), alta atenuação, ruído, interferência, largura de banda limitada e potência de transmissão (SANTOS; RUBINSTEIN, 2012).

Ao efetuar-se uma estimativa mais precisa do comportamento de ondas acústicas em ambiente marinho é necessário considerar parâmetros oceanográficos como as propriedades físico-químicas (temperatura, salinidade, densidade, etc.) sobrepostas umas às outras, definindo assim o perfil estratificado dos oceanos. Fronteiras entre tais camadas são geradas devido ao padrão de circulação das correntes marítimas, que contribuem para a variação tanto da composição quanto da localização e movimentação das massas d'água (FERREIRA, 2002).

Além das correntes marítimas, outros fatores influenciam as propriedades físico-químicas da água no oceano, principalmente aqueles relacionados com a posição geográfica, as estações do ano, as condições atmosféricas e a profundidade. As fronteiras entre as camadas subaquáticas podem exercer profunda influência na propagação acústica devido a efeitos gerados pela reflexão, espalhamento e absorção das ondas sonoras.

Dentre as características de maior importância podem ser citadas a temperatura e salinidade e suas respectivas variações horizontal e vertical. Juntas, controlam a densidade da água do mar que por consequência influenciam nos movimentos verticais das águas oceânicas e na propagação do som.

2.1.1 Variação da temperatura

A distribuição da temperatura na superfície do oceano é aproximadamente zonal, como observado na fig. 1. No oceano aberto, os valores mais altos de temperatura superficial encontram-se em baixas latitudes (da ordem dos 28 °C na região equatorial), decrescendo em direção às altas latitudes, chegando até 2 °C negativos nas regiões polares (HARARI, 2007).

A temperatura também varia de acordo com a profundidade no oceano. Logo abaixo da camada superficial do oceano encontra-se a termoclina, camada marítima onde ocorre uma mudança brusca de temperatura. Sua existência e profundidade dependem da latitude e da estação do ano. Ela é permanente nas baixas latitudes (localizada entre 150 e 400 m abaixo da superfície) e nas médias latitudes é sazonal (só ocorre no verão) e encontra-se entre 400 e 1000 m de profundidade. Nas altas latitudes a termoclina é ausente, como indicado na fig. 2. Abaixo dos 1000 metros de profundidade, a temperatura oceânica tende a ser uniforme, independentemente da sua latitude.

Figura 1 – Temperatura superficial global (°C) no mês de Agosto de 2011.

Fonte: http://funnel.sfsu.edu/courses/gmo405/images/SST_Aug2011.gif.

Figura 2 – Perfis médios típicos da variação de temperatura com a profundidade para diferentes latitudes em oceano aberto.

Fonte: (HARARI, 2007).

2.1.2 Variação da salinidade

A salinidade corresponde à quantidade total de sólidos inorgânicos dissolvidos na água. Suas variações temporais e espaciais estão relacionadas a controles físicos, dependendo quase que inteiramente do balanço entre a evaporação, precipitação e capacidade de mistura entre as camadas de água da superfície e as demais (HARARI, 2007).

No oceano aberto, o balanço entre evaporação e precipitação é o principal fator de controle da salinidade e as taxas desses processos dependem da latitude. Já nas zonas costeiras, o aporte fluvial e os lençóis freáticos que são grandes fontes de água doce, influenciam os valores da salinidade local. O mesmo pode ser observado nas regiões polares, uma vez que o degelo diminui a salinidade devido à presença de geleiras e de neve. Em lagos de água salgada e em mares fechados situados em latitudes baixas a salinidade tende a ser maior, pois nestas regiões a evaporação é alta devido à influência dos ventos e o fluxo de água doce é relativamente pequeno.

A salinidade na superfície dos oceanos têm valores máximos em latitudes próximas a 20°. Essas regiões sofrem grandes influências dos ventos alísios e a evaporação excede os valores de precipitação atmosférica. A salinidade decresce à medida que a latitude aumenta, assim como também é menor na direção do Equador, como ilustrado na fig. 3.

Figura 3 – Salinidade superficial média global (em PSU - *Practical Salinity Unit*).

Fonte: <http://pages.vassar.edu/magnes/files/2010/01/world-salinity.jpg>.

Os valores de salinidade da superfície estão relacionados ao equilíbrio climático e são dependentes dos valores da precipitação e da evaporação. Desta forma, a salinidade é maior na superfície do que em maiores profundidades. Entre as camadas de superfície ou camada de mistura e as camadas de fundo oceânico, existem regiões onde a salinidade

decrece com a profundidade, denominadas haloclinas e são tipicamente encontradas em baixas e médias latitudes, como pode ser verificado na fig. 4.

Figura 4 – Perfis típicos de salinidade em função da profundidade em latitudes polares e tropicais.

Fonte: (SCHMIEGELOW, 2004).

Abaixo de 1000 m (camada de fundo), a salinidade é relativamente constante, pois nesta profundidade a influência dos fatores que alteram a salinidade na superfície é mínima. Nas camadas de fundo, as salinidades são regularmente encontradas entre 34,5 e 35 PSU em todas as latitudes.

2.1.3 Variação da densidade

A variação da densidade é uma função não linear com relação à temperatura e à salinidade. Os valores desta grandeza na água do mar variam entre 1,020 e 1,030 g cm⁻³. Já no oceano profundo, a pressão torna-se o principal fator que influencia a densidade.

O perfil estratificado do oceano em camadas está relacionado ao aumento da densidade da água com a profundidade. O oceano encontra-se geralmente em equilíbrio dinâmico, com camadas menos densas superficialmente e mais densas no fundo. Caso a densidade da água de superfície aumente, ela se tornará gravitacionalmente instável e afundará (circulação denominada termohalina) ou será transportada. Esses dois fenômenos determinam o padrão de circulação oceânica (SCHMIEGELOW, 2004).

A densidade da água não aumenta uniformemente com a profundidade. A camada oceânica onde ocorre um rápido aumento da densidade é denominada de picnoclina. Dependendo das condições locais, pode chegar até 1000 m de profundidade (nas regiões equatoriais e tropicais) ou pode ser absolutamente ausente (nos pólos). As picnoclinas correspondem a aproximadamente 18% do volume total da água do oceano.

Abaixo da picnoclina encontram-se as águas profundas, situadas abaixo dos 1000 m nas médias latitudes (entre os 40°), e representam cerca de 80% de toda água do oceano.

A densidade aumenta lentamente com a profundidade e sua variação varia de acordo com a latitude, porém a média da densidade nessas camadas em todas latitudes é $1,0279 \text{ g cm}^{-3}$.

2.1.4 O oceano como meio de propagação do som

A propagação do som nos oceanos não é trivial devido às suas características físico-químicas não homogêneas. Por menores que sejam as variações de temperatura, densidade ou salinidade, isso faz com que o som tenha diferentes velocidades de propagação ao longo das camadas oceânicas, promovendo refrações das ondas sonoras que se propagam pelo meio subaquático (RIBEIRO; PEDROZA; COSTA, 2011).

Uma forma simples de expressar a velocidade do som na água com precisão de até 1 m s^{-1} em profundidades de até 1 km foi descrita por Medwin (1975) segundo a expressão 2.1:

$$v = 1449.2 + 4.6T - 0.055T^2 + 0.00029T^3 + (1.34 - 0.010T)(S - 35) + 0.016z \quad (2.1)$$

Nesta expressão, a velocidade v é dada em m s^{-1} , a temperatura T é dada em graus celsius, a salinidade S é dada em partes por mil do peso dos sais dissolvidos e a profundidade z é dada em metros. Percebe-se que o efeito da salinidade é pequeno exceto próximo a estuários que são caracterizados por grandes transições de marés e gradientes ambientais ou em regiões polares onde a água de degelo se mistura com a água salina.

Descritas as características principais do oceano como meio de propagação, percebe-se a dificuldade de implementação de sensoriamento em águas profundas usando emissão de ondas eletromagnéticas e ondas mecânicas de forma ativa. Por estes motivos o foco do trabalho é monitorar ondas acústicas marinhas próximo à superfície e viabilizar um enlace aéreo usando ondas de rádio frequência para transmitir os sinais, visando o estudo de sons marinhos e suas causas.

2.1.5 Transdutores para aplicações submersas

Os estudos em eletroacústica começaram há mais de 200 anos com observações dos efeitos mecânicos associados com a eletricidade e magnetismo. Como consequência, foram desenvolvidas importantes aplicações em acústica marinha principalmente a partir do século 20. Pode ser verificado em (URICK, 1967) a relação desta evolução e sua ligação com aplicações subaquáticas.

A descoberta da magnetostricção por James Joule entre 1842 e 1847 e da piezoelectricidade, observada no quartzo e em outros cristais, por Jacques e Pierre Curie em 1880, tiveram grande importância para as aplicações subaquáticas, uma vez que os materiais

com tais propriedades são usados em larga escala nos transdutores atualmente (HUNT, 1954).

Segundo Butler e Sherman (2016) são cinco os principais mecanismos de transdução eletroacústica que relacionam força mecânica e corrente (ou tensão):

- Transdutores de campo elétrico.
 - Piezoelétrico
 - Eletrostático
- Transdutores de campo magnético.
 - Magnetostrictivo
 - Relutância variável
 - Bobina móvel

Na tabela 1 é apresentada uma comparação da força por unidade de área, ou pressão acústica de cada tipo de transdutor. Algumas aplicações requerem transdutores que suportam pressões acústicas da ordem de 1 - 10 atm (10^5 - 10^6 Pa).

Tabela 1 – Pressão acústica para diversos tipos de transdutores.

Tipo do Transdutor	Pressão acústica (Pa)
Eletrostático	3
Bobina móvel	8000
Relutância variável	10^5
Piezoelétrico (PZT)	8×10^6
Magnetostrictivo (Terfenol)	8×10^6

Fonte: (BUTLER; SHERMAN, 2016).

Segundo Au e Hastings (2008), a intensidade acústica é definida como a taxa de fluxo de energia através de uma área unitária normal à direção da propagação da onda. Sendo S a área de um pequeno volume de fluido e dE/dt sendo a razão do fluxo de energia acústica dentro da área S , a intensidade acústica da onda pode ser definida como:

$$I = \lim_{S \rightarrow 0} \frac{dE/dt}{S} \quad (2.2)$$

A energia E da onda acústica pode ser pensada em termos de trabalho em uma membrana pela pressão acústica. Como o trabalho é igual a Força $\times$ Distância, o fluxo de energia ou o trabalho por unidade de tempo causado pela onda acústica é dado por:

$$\frac{dE}{dt} = F \frac{dx}{dt} \quad (2.3)$$

Uma vez que a pressão acústica está relacionada à força no fluido ($p = F/S$) e $u = dx/dt$, a intensidade de uma onda acústica pode ser expressa também em termos de pressão acústica:

$$I = p.u \quad (2.4)$$

Segundo a literatura, o transdutor eletrostático não é adequado para aplicações de alta potência na água, o transdutor de bobina móvel se adapta com pouca eficiência para aplicações de baixa potência, o transdutor de relutância variável é atrativo para potências médias, e os transdutores piezoelétricos e magnetostrictivos são totalmente capazes de produzir as forças necessárias em alta potência (BUTLER; SHERMAN, 2016). A impedância característica de transdutores piezoelétricos, eletrostrictivos e magnetostrictivos também favorece a sua utilização na água, onde a impedância acústica é aproximadamente $22 \times 10^6 \text{ kg}^2 \text{ m}^{-1} \text{ s}$ para PZT, e, combinado com sua alta capacidade de força, o torna adequado para a maioria das aplicações de som subaquáticas.

2.1.6 Equipamentos de monitoramento acústico

Em torno de 40 equipamentos com as características de gravadores acústicos autônomos passivos foram comparados por Sousa-Lima et al. (2013). Dentre as características de armazenamento, tempo e formato de gravação, nada é citado em relação à possibilidade de verificação do funcionamento da unidade sem a necessidade de interrupção. As principais limitações que afetam a duração do funcionamento da unidade são as características de armazenamento e capacidade de bateria, que estão ligadas diretamente à taxa de amostragem dos sinais. Pode ser verificado na tabela 2 uma comparação das principais características dos equipamentos em questão.

Percebe-se também a necessidade de compactação sem perdas de informação (*lossless*) uma vez que pode ser aumentada a taxa de amostragem sem necessidade de aumento da capacidade de armazenamento, sendo o *codec* (codificador/decodificador) FLAC (*Free Lossless Audio Codec*) um forte candidato à implementação neste tipo de sistema.

2.1.7 Detecção de eventos acústicos

São diversos os tipos de algoritmos voltados para detecção, classificação, localização e rastreamento de sons dentro do oceano. O processamento acústico passivo desempenha um papel fundamental nesse cenário uma vez que um determinado sinal deve ser detectado, informações devem ser obtidas para uma classificação e, por fim, pode ser feito o tratamento nos dados para localização e rastreamento (ZIMMER, 2011).

Tabela 2 – Comparação das principais características de monitores acústicos passivos.

Instrumento	Tempo de Operação	Frequência de Amostragem (Hz)	Fonte de Energia	Armazenamento
AAR	70 horas	2000-20000	Baterias Alcalinas	1GB Memória Flash
AARS	4 a 5 horas	69000	Bateria de Lítio	HD externo
Acousonde	140 horas	232000	Bateria de Lítio	64GB MicroSD
AULS	60 horas	44000	Bateria de Gel de Chumbo-Ácido	HD externo
DASAR	1000 horas	2400	Baterias Alcalinas	HD externo
OceanPod	4300 horas	96000	Baterias Alcalinas	512GB MicroSD
DMON	180 horas	80000	Baterias Recarregáveis de Li-Ion	64GB Memória Flash
PANDA	240 horas	10000	Baterias Recarregáveis de Li-Ion	HD externo
SYLence	4300 horas	192000	Bateria Alcalina Tipo D	HD externo

Fonte: adaptado de (SOUSA-LIMA et al., 2013)

O método de detecção de eventos acústicos mais comum é baseado no estabelecimento de um nível *threshold* de intensidade sonora acima do qual considera-se a ocorrência de um evento. Este método pode ser implementado em diferentes variações, sendo a mais básica a detecção aplicando a média quadrática (*Root Mean Square* - RMS) com janelas entre 25 e 100 ms do sinal amostrado (ZIMMER, 2011).

Dentre outros métodos de detecção se destacam os detectores sequenciais de probabilidade (WALD; WOLFOWITZ, 1948), soma de energia no domínio da frequência, correlação de espectrograma (MELLINGER; NIEUKIRK; HEIMLICH, 2017) e técnicas de filtros que melhoram a relação sinal ruído visando evitar falsas detecções, aumentando assim a qualidade do detector. Uma breve comparação entre os métodos de detecção pode ser verificada na tabela 3.

Há uma necessidade do uso de algoritmos computacionais que trabalham no domínio da frequência ou no domínio do tempo. Dependendo do tipo de aplicação essa escolha pode impactar em diversos aspectos, desde a inserção de atrasos no processamento

Tabela 3 – Métodos de Detecção de eventos acústicos e características

Método	Entrada de Dados	Função
Clicks	Dado de áudio	Procura por transientes sonoros
Apito	Transformada Rápida de Fourier	Procura por ruídos tonais
Soma de Energia	Transformada Rápida de Fourier	Detecta sons com energia em uma banda de frequência específica
Correlação de Espectrograma	Transformada Rápida de Fourier	Detecta sons que correspondem a uma "forma" definida previamente em um espectrograma
Filtros Combinados	Dado de áudio	Detecta sons usando um filtro correspondente definido previamente
Probabilidade	Dado de áudio	Uma implementação de um teste de razão de verossimilhança

Fonte: adaptado de (MELLINGER; NIEUKIRK; HEIMLICH, 2017).

decorrente do cálculo das transformadas, até diminuição do tempo de operação de determinada instrumentação. Como alternativa pode-se implementar algoritmos mais intensos computacionalmente em outros estágios posteriores para classificação de eventos.

2.2 REDES DE SENSORES SEM FIO

Uma rede de sensores sem fio pode ser denominada como um aglomerado de nós sensores e pelo menos uma estação base, também nomeada de *sink*. O nó sensor é um pequeno dispositivo autônomo que possui quatro partes: detecção, processamento, comunicação e suprimento de energia para o sistema. Esses sensores são usados para coletar as informações do meio ambiente e transferi-las para a estação base, a qual fornece uma conexão local ou com outras redes, como por exemplo a Internet (através de um *gateway*) onde os dados coletados são processados, analisados e apresentados de forma útil (GUPTA; SINHA, 2014).

O enlace sem fio é normalmente implementado com base na banda não licenciada ISM - *Industrial, Scientific, and Medical*, usando diferentes tecnologias de modulação e padrão de comunicação, a exemplo: ZigBee, Bluetooth *Low Energy* (BLE) e 6LoWPAN. Dois padrões de modulação (BPSK e QPSK) especificam as camadas físicas que usam técnicas de sequência direta de espalhamento do espectro - *Direct Sequence Spread Spectrum*

(DSSS) em três bandas base (868-870MHz, 902-928MHz e 2.4GHz) (GUPTA; SINHA, 2014). A tabela 4 apresenta as diferentes bandas de frequência do padrão IEEE 802.15.4 e suas características de modulação, capacidade de canais e taxa de transmissão (BAMBER; SHARMA, 2010).

Tabela 4 – Bandas de Frequência do IEEE 802.15.4.

Camada Física	Banda	Núm. de Canais	Modulação	Taxa
868 MHz	868-870 MHz	1	BPSK	20kbps
868-915 MHz	902-928 MHz	1-10	BPSK	40kbps
2.4 GHz	2.4-2.4835 GHz	11-26	QPSK	250kbps

Fonte: (BAMBER; SHARMA, 2010)

O IEEE 802.15.4 é um padrão definido para dispositivos de comunicação que operam em redes pessoais sem fio de baixa taxa de transmissão de dados (LR-WPANs). Ele fornece comunicação de baixo custo, curto alcance, baixa potência e baixa taxa de dados para redes de sensores. Destina-se a aplicações de sensores sem fio, que exigem comunicação de curto e médio alcance para maximizar a vida da bateria. O padrão especifica as duas camadas mais baixas da pilha de protocolos: o controle de acesso físico (PHY - *Physical*) e acesso ao meio (MAC - *Medium Access Control*), com base no modelo OSI (*Open System Interconnection*) como mostrado na Fig. 5. As camadas superiores e as subcamadas de interoperabilidade da pilha de protocolos são definidas separadamente por outras arquiteturas, como 6LoWPAN, ZigBee, ISA100.11a e WirelessHART (RAWAT et al., 2014).

A tecnologia WSN (*Wireless Sensors Network*) oferece inúmeras vantagens quando comparada com as tecnologias convencionais de rede, como por exemplo, baixo custo, escalabilidade, estabilidade, precisão, flexibilidade e fácil instalação. Além disso, com o avanço tecnológico o sensoramento está se tornando mais inteligente, mais barato e automatizado (RAWAT et al., 2014).

2.2.1 Tipos de arquitetura

O consumo de energia de um nó sensor é influenciado pela estrutura das camadas de protocolos e também pelo comportamento de cada camada ao processar o dado coletado. As camadas de protocolo usadas pelos nós sensores e a estação base (*sink* ou sorvedouro) incluem aplicação, transporte, rede, acesso ao meio, *link* de dados, camada física e gerenciamento de energia. A fig. 6 apresenta a pilha de protocolos e serviços de camada cruzada (GUPTA; SINHA, 2014).

A camada de aplicação suporta *softwares* para diferentes aplicações dependendo da tarefa de detecção. Três tipos de protocolos são definidos para esta camada:

Figura 5 – Pilha do protocolo IEEE 802.15.4 e camadas superiores.

Fonte: adaptado de (BURATTI et al., 2009).

Figura 6 – Pilha de Protocolos e Serviços de cada nó sensor e da estação base.

Fonte: adaptado de (GUPTA; SINHA, 2014).

- SMP (*Sensors Management Protocol*) - Protocolo de Gerenciamento de Sensores.
- TADAP (*Task Assignment and Data Advertisement Protocol*) - Atribuição de tarefas e divulgação de dados.
- SQDDP (*Sensor Query and Data Dissemination Protocol*) - Consulta do sensor e protocolo de disseminação de dados.

A camada de transporte ajuda a manter o fluxo de dados quando a camada de aplicação está em processamento. O desenvolvimento do protocolo na camada de transporte é um verdadeiro desafio porque os sensores são influenciados por muitos parâmetros e restrições, como o poder de processamento limitado, pouco recurso de energia e memória.

A camada de rede permite o roteamento de dados através da rede sem fio usando um canal de comunicação. Existem várias estratégias para rotear dados com menor custo de energia, tais como: otimizar a rota do pacote, aplicar métricas corretas de tempo de desligamento do rádio e roteamento baseado em árvores de decisão com pesos atualizados em tempo real (RAWAT et al., 2014).

A camada de enlace de dados é responsável pela multiplexação de dados, detecção de quadro de dados, controle de acesso ao meio (MAC), detecção e correção de eventuais erros. Os problemas de projeto da camada de protocolo devem levar em consideração as diferentes restrições como conservação de energia, mobilidade dos nós sensores e estratégias de gerenciamento e recuperação de falhas.

A camada física é a camada mais baixa e é responsável pela seleção de frequência, geração de frequência portadora, detecção de sinal, modulação e criptografia de dados.

Pode ser verificado na tabela 5 os padrões IEEE 802.15.4 mais usados e suas características quanto à topologia, tempo útil de bateria, quantidade de nós dentro da mesma rede, taxa de transferência e alcance (RAWAT et al., 2014).

Tabela 5 – Padrões IEEE 802.15.4 mais usados.

Padrão	Topologia	Vida Útil da Bateria (horas)	Nós na Rede	Máxima Taxa de Transferência	Alcance (m)
ZigBee	Malha	100-1000	255	250 kbps	10-100
6LoWPAN	Malha	100-365	65536	250 kbps	1-100
WirelessHART	Malha	760	200	250 kbps	1-100
ISA 100.11a	Malha, Estrela	1000	-	250 kbps	100

Fonte: (RAWAT et al., 2014).

A topologia de rede em estrela é preferida quando a área de cobertura é pequena e baixa latência é requerida pela aplicação, enquanto que a topologia ponto-a-ponto ou malha é adequada para uma grande área de cobertura onde a latência não é um problema crítico uma vez que dados adicionais são transmitidos entre os nós para controle de rotas da rede.

A versão original da especificação do padrão de comunicação em 2003 suportava bandas baixas de 868/915 MHz com taxas de dados de 20 e 40 kbps e bandas altas de 2,4 GHz com uma taxa de 250 kbps. A versão de 2009 do padrão de comunicação (IEEE 802.15.4d-2009) melhora as taxas máximas de dados para até 100 e 250 kbps para as bandas 868/915 MHz e inclui mais algumas modulações referentes ao uso no Japão na

banda 950MHz.

O 6LoWPan extrapola o uso do protocolo IPv6 em redes sem fio com baixo consumo de energia, como é o caso de sistemas embarcados com processamento muitas vezes limitado (SHELBY; BORMANN, 2011). Com essa adaptação do IPv6, tornou-se possível o conceito de “Internet das Coisas”, que na prática pode ser definido como uma rede IP de sistemas embarcados que se interconectam através de serviços de controle e monitoramento.

Em contraste ao IPv6 tradicional, o 6LoWPan possui diversas melhorias que buscam o aproveitamento da potência de interfaces de rádio, redução do tamanho da palavra transmitida ou recebida e aproveitamento de banda, suportando dispositivos interconectados em topologias estrela e malha (DUNKELS; GRONVALL; VOIGT, 2004).

Um diagrama das topologias de rede pode ser verificado na fig. 7.

Figura 7 – Padrão IEEE 802.15.4: Topologias de rede compatíveis.

Fonte: adaptado de (RAWAT et al., 2014).

2.2.2 Sistemas operacionais para RSSF

Do ponto de vista do *hardware*, a Internet das Coisas é composta de equipamentos heterogêneos, mais ainda que a própria *internet* tradicional. Os dispositivos IoT podem ser classificados em duas categorias com base na sua capacidade e desempenho. A primeira categoria consiste em dispositivos IoT *high-end*, que incluem computadores de placa única, como a plataforma Raspberry Pi (UPTON; HALFACREE, 2014) e smartphones. Estes dispositivos têm recursos suficientes e características adequadas para executar *software* baseado em sistemas operacionais (OS), como Linux ou BSD (HAHM et al., 2016).

A segunda categoria consiste em dispositivos IoT *low-end* que são limitados em recursos para executar esses sistemas operacionais tradicionais. Exemplos populares de dispositivos IoT *low-end* incluem Arduino (BLUM, 2013), Zolertia Z1 e SensorTag (ZHAO; GUIBAS, 2004).

Os dispositivos IoT *low-end* geralmente são muito restritos em termos de recursos, incluindo energia, capacidade de processamento e memória. Recentemente, a *Internet Engineering Task Force* (IETF) padronizou uma classificação destes dispositivos em três subcategorias com base na capacidade de memória (BORMANN; ERSUE; KERANEN, 2014).

- Classe 0: dispositivos com recursos menores de 10kB de RAM e 100kB de FLASH.
- Classe 1: dispositivos com recursos da ordem de 10kB de RAM e 100kB de FLASH.
- Classe 2: dispositivos com mais recursos que classes 0 e 1 porém ainda escassos se comparados com dispositivos *high-end*.

Ao selecionar um sistema operacional genérico para um dispositivo IoT *low-end*, diversos requisitos devem ser levados em conta.

- Baixo uso de memória.

Em comparação com outros dispositivos conectados, tratando-se de IoT há uma limitação maior em recursos, especialmente em termos de memória. Um dos requisitos para um sistema operacional genérico é caber dentro dessas restrições de memória. Enquanto PCs, smartphones, tablets ou laptops fornecem Gigabytes de memória, os dispositivos IoT normalmente fornecem apenas alguns kilobytes, tanto para memória volátil (RAM) quanto para não volátil (ROM) (HAHM et al., 2016). Desta forma os desenvolvedores devem ter um conjunto de bibliotecas otimizadas (potencialmente compiladas de forma cruzada) que habilitam funcionalidades IoT comuns e estruturas de dados eficientes.

- Suporte a *hardwares* heterogêneos.

Os dispositivos IoT são baseados em microcontroladores de várias arquiteturas e famílias, incluindo 8 bits (Intel 8051/52, Atmel AVR), 16 bits (TI MSP430) e 32 bits (ARM7, ARM Cortex-M). Embora a diversidade de *hardware* e protocolos utilizados na *internet* de hoje seja relativamente pequena, o grau de heterogeneidade em IoT é alto uma vez que a quantidade de casos de uso é elevada. Por essa razão, foi necessário o desenvolvimento de uma grande variedade de tecnologias de comunicação.

- Conectividade de rede.

A principal vantagem dos dispositivos IoT é que eles podem se interconectar e, através de um *gateway* ou roteador, podem alcançar a *internet*. Para isso, os dispositivos IoT podem ser equipados com uma ou mais interfaces de rede.

- Eficiência energética.

As técnicas de comunicação utilizadas em IoT englobam uma grande variedade de rádios de baixa potência. Muitos dispositivos IoT usam baterias ou outras fontes de energia restritas. A exemplo, medidores inteligentes e outros dispositivos de automação residencial devem trabalhar por anos com uma única carga de bateria (MIN et al., 2002). A eficiência energética é necessária devido ao grande número de dispositivos IoT que devem ser implantados no mundo (dezenas de bilhões). O *hardware* IoT em geral possui recursos para operar de forma eficiente porém isso exige também requisitos em *software* como, por exemplo, colocar os dispositivos no modo baixo consumo sempre que possível, usar técnicas como rádio *duty cycling*, minimizando o número de tarefas periódicas que precisam ser executadas, entre outras.

- Capacidades em tempo real.

A precisão em tempo de execução é importante em vários casos de uso em IoT como por exemplo aplicativos de saúde inteligentes que fornecem monitoramento wireless (MILENKOVIĆ; OTTO; JOVANOVIĆ, 2006), conectividade entre automóveis e estrada ou em outros cenários de automação que incluem atuadores e robôs dentro de uma indústria (HUGHES et al., 2004). Portanto um sistema operacional (SO) que pode cumprir requisitos de execução é chamado de Sistema Operacional em Tempo Real (*Real-Time Operating System* - RTOS) e é projetado para garantir tempos de execução e latências de interrupção pequenas.

- Segurança.

Em alguns cenários, os sistemas IoT fazem parte de uma infraestrutura crítica com implicações de segurança de vida (STOUFFER; FALCO; SCARFONE, 2011) e uma vez que estão conectados à *internet* é esperado que as características de segurança e padrões de privacidade sejam atendidas. Os desafios de segurança em IoT incluem a integridade de dados, autenticação e controle de acesso em várias partes da arquitetura. Levando em conta esses pontos um requisito desafiador para um SO IoT é fornecer os mecanismos necessários de segurança (bibliotecas criptográficas e protocolos de segurança) mantendo flexibilidade e usabilidade (HOEPMAN; JACOBS, 2007).

Segundo a comparação de (HAHM et al., 2016), um dos projetos emergentes que combina todos os aspectos citados previamente é o CONTIKI-OS, possuindo uma categoria de escalonamento de tarefas baseada em eventos externos, camada de abstração

de hardware, camada de controle de acesso ao meio (MAC) com alta eficiência energética e suporte a redes 6LoWPAN utilizando menos de 32kB em RAM do processador alvo.

2.2.3 Evolução da camada de controle de acesso ao meio (MAC)

A camada de controle de acesso ao meio é responsável por controlar como os nós sensores compartilham o meio sem fio. Para reduzir as chances de colisões esses protocolos usam técnicas para aumentar o rendimento energético da rede (HUANG et al., 2013). Pode-se relacionar os protocolos MAC em quatro categorias: assíncrono, síncrono, *frame-slotted* e multicanal. A comparação entre as diversas categorias é feita em termos da eficiência energética, desempenho de entrega de dados e sobrecarga necessária para manter os mecanismos de um protocolo funcionando.

Nos protocolos MAC assíncronos cada nó escolhe seu intervalo de atividade de forma autônoma sem sincronização com os vizinhos da rede, desta forma podem alcançar um ciclo de trabalho ultra-baixo. Em contra partida deve-se buscar maneiras eficientes para estabelecer a comunicação entre os nós.

A amostragem de preâmbulo pode ser usada para minimizar a duração ativa dos receptores, fazendo o nó "acordar" periodicamente por um curto intervalo de tempo para verificar o canal de rádio. Se o canal estiver ocioso, a melhor opção é colocar o nó em menor consumo possível imediatamente, caso haja atividade o nó continua em estado de recepção até que os dados subsequentes sejam recebidos com sucesso usando um tempo limite de duração (HILL; CULLER, 2002; EL-HOIYDI, 2002).

Para evitar colisões de preâmbulo uma técnica usada em CSMA (*Carrier Sense Multiple Access*) coloca o transmissor para executar uma avaliação do canal antes da transmissão (CCA - *Clear Channel Assessment*). O mesmo é válido para o receptor na tentativa de detecção do preâmbulo. Uma melhoria do CCA foi desenvolvida no protocolo B-MAC, conseguindo uma operação de baixa potência, empregando amostragem adaptativa de preâmbulo que reduz o ciclo de trabalho do nó e minimiza a escuta ociosa (POLASTRE; HILL; CULLER, 2004).

Nos protocolos MAC síncronos há uma troca de mensagens entre nós vizinhos para estabelecer um canal de comunicação, entretanto, isso se torna um custo adicional. O próprio nó escolhe seu tempo de ativação e transmite seu próximo tempo de ativação, tornando-se um sincronizador. Se um nó recebe uma programação de outro nó vizinho antes de configurar seu próprio tempo de ativação, torna-se um seguidor (HUANG et al., 2013).

Um protocolo MAC síncrono clássico é o S-MAC, no qual os nós são organizados em *clusters* por sincronização de tempo local. Cada cluster possui um cronograma independente composto de três períodos: SYNC, DATA e SLEEP. Todos os nós dentro de um mesmo

cluster entram em estado ativo no início do SYNC para sincronizar os relógios entre si. Os nós com pacotes disponíveis para envio trocam quadros RTS (*request-to-send*) e CTS (*clear-to-send*) dentro do período DATA. Nós que não estão envolvidos na comunicação retornam para modo de baixo consumo no início do período SLEEP e os outros nós retornam para o modo de baixo consumo depois de terminar a transmissão de pacotes de dados e quadros de reconhecimento (ACK). Pode-se observar que em cada ciclo um pacote só pode ser encaminhado de um nó para o próximo, ou seja, apenas um salto (YE; HEIDEMANN; ESTRIN, 2002).

Os protocolos MAC do tipo *frame-slotted* derivam as técnicas do TDMA (*Time Division Multiple Access*), onde a taxa de transferência é alta e a utilização de canal é maximizada sob alta contenção. Uma combinação entre CSMA e TDMA melhora consideravelmente a utilização do canal porém traz uma sobrecarga na detecção de slots e diminui consideravelmente a eficiência energética da rede (HUANG et al., 2013).

A transmissão de dados paralela é atraente uma vez que a largura de banda de rádio em RSSF é limitada. É desejável conceber protocolos MAC multicanais com suporte multitarefa. Uma maneira de conseguir isso é combinar TDMA com FDMA (*Frequency Division Multiple Access*), a exemplo do protocolo MC-LMAC que coordena transmissões em múltiplos canais de frequência tirando proveito de transmissões paralelas sem interferência e sem contenção em diferentes canais (INCEL et al., 2011).

Entre os tipos de protocolos MAC o padrão mais utilizado é o assíncrono com LPL (*Low Power Listening*). O ContikiMAC se destaca em economia de energia, quando comparado com outros protocolos RDC (*Radio Duty Cycle*) como o X-MAC, CXMAC e LPP (DUNKELS, 2011), (BUETTNER et al., 2006), (MUSALOIU-E; LIANG; TERZIS, 2008). Usando um método simples e elaborado de temporização, o ContikiMAC permite que seu mecanismo de entrada e saída do modo de economia energética seja altamente eficiente (UWASE et al., 2017).

2.2.4 Protocolos de roteamento

O RPL (*Routing Protocol for Low Power and Lossy Networks*) especifica um protocolo de roteamento bidirecional especialmente adaptado para as necessidades de comunicação IPv6 sobre redes de baixa potência e com perdas, suportando tráfego ponto-a-ponto e comunicação com um servidor central para múltiplos nós dentro da rede criando um grafo acíclico direto orientado a destino (DODAG) (HERSENT; BOSWARTHICK; ELLOUMI, 2011), (WINTER, 2012).

Protocolos orientados a estado de *link* tendem a ser mais eficientes energeticamente se a topologia da rede é conhecida. No caso do RPL a orientação pode ser por destino, dependendo de como os nós da rede estão conectados com o sorvedouro (*sink*), ou orientado

a vetores de distância, baseando-se na capacidade de memória e processamento de energia dos nós. Uma função objetiva é usada para calcular um *ranking* do nó com base na contagem de saltos necessários para o alcance do *sink* e contagem de transmissões esperadas (ETX) (CARELS et al., 2014).

A fig. 8 apresenta as mensagens trocadas na inicialização da rede para formar as rotas desde o nó até o sorvedouro (*sink*). Um nó ao se ligar na rede, deve fazer uma requisição aos nós ao seu redor para alcançar o *sink*. Neste momento o nó envia uma mensagem para a formação de um *cluster* DAG (*Directed Acyclic Graph*). Isso caracteriza um grafo ou vetor de rotas que garante que nenhum nó irá receber mensagens duplicadas de outro nó e todos estarão conectados a pelo menos um nó raiz. Dentro das variações de um DAG, pode existir DODAG (*Destination-Oriented DAG*) que implica um DAG com um único ponto de destino. Um DAG ainda sem *sink* deve enviar uma mensagem DIS (*DODAG Information Solicitation*) para um vizinho a fim de achar uma melhor rota e permitir assim o roteamento dos dados de outros nós anteriores conectados a ele. O fluxo bidirecional é alcançado através do uso das mensagens DIO (*DODAG Information Object*) e DAO (*Destination Advertisement Object*) montando assim as rotas dos nós para o *sink* bem como o inverso.

Figura 8 – Arquitetura RPL.

Fonte: próprio autor.

Outros protocolos de roteamento como o AODV (*Ad Hoc On-Demand Distance*

Vector) e DYMO (*Dynamic MANET On-demand Routing Protocol*) possuem técnicas de roteamento baseado em vetores presentes no RPL (MIAO; XU, 2010). Entretanto, comparando-os é possível notar maior tempo de convergência devido a características de sincronismo, sobrecarga de roteamento e atraso fim-a-fim (LI et al., 2015).

Em questão de segurança, o RPL define três modelos: inseguro, onde nenhuma técnica de segurança é implementada; pré-instalado, onde todos os nós da rede são previamente provisionados com chaves de segurança que decodificam as mensagens seguras do protocolo; e o autenticado, onde estende-se a metodologia pré-instalado para entrar na rede como um nó da árvore e caso o nó seja candidato a tornar-se um *sink* ele deve requisitar uma nova chave ou certificado de uma autoridade certificadora. Vale destacar que o último mecanismo é pouco documentado e utilizado uma vez que não é seguro deixar uma autoridade certificadora exposta fisicamente onde a rede estará em funcionamento (HERSENT; BOSWARTHICK; ELLOUMI, 2011).

Uma rede com múltiplos *sink*'s também é especificada no protocolo RPL e, dependendo da aplicação, essa necessidade é fundamental uma vez que uma quantidade grande de nós estará enviando mensagens de descoberta de rede para determinar a melhor rota a ser usada. Caso tenha apenas um *sink* presente é provável que o rendimento dos nós diminua em questão de eficiência energética e atrasos, entre outras características. É proposto em (CARELS et al., 2014) uma técnica de *sink* virtual de baixa complexidade que diminui a energia média de consumo em torno de 30% para o acréscimo de 1 a 4 *sink*'s. Essa técnica foi testada em ambiente de simulação e em dispositivos dentro de uma rede real, atestando assim a qualidade do protocolo RPL e possíveis implementações dentro das especificações originais.

2.2.5 Protocolos de comunicação

Existem diversos protocolos M2M (*Machine-to-Machine*) para entrega de mensagens, como por exemplo, o WebSocket, XMPP (*Extensible Messaging and Presence Protocol*) e AMQP (*Advanced Message Queuing Protocol*) (YASSEIN; SHATNAWI et al., 2016). Estes protocolos são comparados amplamente devido ao uso em diversas aplicações, entretanto suas características não se mostraram tão eficientes quanto os protocolos MQTT (STANDARD, 2014) e CoAP (SHELBY; HARTKE; BORMANN, 2014). O WebSocket, por exemplo, tende a substituir aplicações REST (*Representational State Transfer*) em serviços comuns WEB inserindo novas características em tempo real (HASIBUAN et al., 2015). A seguir são apresentados os conceitos do protocolo MQTT-SN (*Message Queue Telemetry Transport for Sensor Networks*), suas diferenças para a padronização base, MQTT e os conceitos do protocolo CoAP (*Constrained Application Protocol*).

2.2.5.1 Protocolo MQTT-SN (Message Queuing Telemetry Transport for Sensor Networks)

O MQTT-SN estende o uso do MQTT para rede de sensores sem fio usando o UDP (*User Datagram Protocol*) ao invés do TCP (*Transmission Control Protocol*). Tendo em vista que nessas redes é comum a falha do *link* devido à mobilidade, essa abordagem se torna eficaz no transporte de dados entre sensores e concentradores de dados. O principal conceito de funcionamento do MQTT é a comunicação centralizada de dados onde os recursos são transferidos via mensagens *publish/subscribe* e a largura de banda é extremamente limitada.

A fig. 9 apresenta a arquitetura do MQTT-SN onde os nós com baixo recurso rodam instâncias clientes e esses dados são publicados no MQTT *broker* através de um *gateway* de protocolos. A comunicação pode ser de forma transparente através do *gateway*, onde apenas algumas alterações de sintaxe são feitas, ou de forma agregada, quando apenas uma conexão é feita com o MQTT *broker* (STANDARD, 2014). As mensagens que são transportadas por MQTT têm um nível de qualidade de serviço associada (QoS) e um nome do tópico. O protocolo define três níveis de QoS: o nível 0, em que o remetente envia a mensagem apenas uma vez e nenhuma nova tentativa é executada, e os níveis 1 e 2, onde o protocolo garante que nenhuma mensagem de publicação será perdida, envolvendo, assim, um processo de reconhecimento de um ou dois passos, respectivamente.

Figura 9 – Arquitetura MQTT-SN.

Fonte: próprio autor.

Algumas diferenças estão presentes ao comparar a versão base do MQTT com a versão para rede de sensores. Como principais, pode-se destacar a quebra da mensagem *CONNECT* em mais mensagens, devido principalmente ao meio de transporte, envio do *topic id* ao invés de *topic name* nas mensagens *PUBLISH*, procedimento de *DISCOVERY* para os nós que não possuem configuração prévia de *gateway* e *keep alive offline* para permitir *sleep* dos nós. É importante também salientar que o processo de *DISCOVERY*

(*SEARCHGW*) suporta múltiplos *gateways*, função comumente usada em redes de sensores sem fio devido a possíveis falhas de *link*.

2.2.5.2 Protocolo CoAP (Constrained Application Protocol)

O CoAP é um protocolo de transferência WEB especializado para uso em nós sensores e redes de baixo recurso, desenvolvido para aplicações M2M pelo IETF (*Internet Engineering Task Force*). O modelo de protocolo e troca de mensagens foi idealizado para substituir o HTTP em redes de sensores sem fio. A organização é feita em duas camadas: Camada de Transição, responsável por gerenciar a troca de mensagens entre os dispositivos e Camada de Requisição/Resposta, responsável pelo envio, recebimento e gerenciamento de recursos. As duas camadas permitem que o CoAP forneça mecanismos de confiabilidade e controle de congestionamento, de forma assíncrona (SHELBY; HARTKE; BORMANN, 2014). A fig. 10 apresenta a arquitetura de uma rede com protocolo CoAP, onde os nós sensores enviam seus dados de forma configurável para o CoAP proxy, mediante requisições. Neste caso, a comunicação é feita de forma ‘um para um’, diferente do caso do MQTT-SN, onde a comunicação é feita em forma *multicast*.

Figura 10 – Arquitetura CoAP.

Fonte: próprio autor.

Pode ser verificado na tabela 6 uma comparação entre os protocolos descritos enfatizando o padrão de transmissão das mensagens, protocolo de transporte, confiabilidade e tamanho do cabeçalho das mensagens e vantagens.

Com base no levantamento feito ao longo deste capítulo, foi estabelecida como proposta deste trabalho o desenvolvimento de uma rede de sensores sem fio voltada para o monitoramento acústico subaquático que atenda às exigências e conveniências da aplicação pretendida, como robustês, boa autonomia energética, confiabilidade na transmissão dos dados e possibilidade de comunicação remota, via internet. Os protocolos e tecnologias que se encaixam nessas características são: ContikiMAC, RPL e MQTT-SN.

Tabela 6 – Análise comparativa dos protocolos CoAP e MQTT-SN.

Características do protocolo	CoAP	MQTT-SN
Protocolo de transporte	<i>User Datagram Protocol</i> (UDP)	<i>User Datagram Protocol</i> (UDP)
Padrão de envio de mensagens	Similar ao HTTP utilizando método <i>request/response</i> (<i>client/server</i>)	<i>publish/subscribe</i> utilizando <i>message broker</i> .
Confiabilidade	Possui recurso para verificar o transporte de mensagens utilizando mensagens do tipo CON.	Possui recurso para verificar o transporte de mensagens utilizando o recurso QoS 1 e 2.
Cabeçalho das mensagens	4 bytes, sendo os dois primeiros com indicação da versão do protocolo e os últimos ligados ao tipo de mensagem CON, NON, ACK ou RST.	2 bytes, apresentando as informações do tamanho e o tipo da mensagem.
Vantagens	Protocolo rápido e eficiente devido ao uso do UDP. Interoperável com o protocolo HTTP.	A arquitetura do tipo broker facilita o gerenciamento da rede.

Fonte: próprio autor.

3 MATERIAIS E MÉTODOS: DESENVOLVIMENTO DA REDE DE SENSORES SEM FIO

Foi desenvolvido um sistema com três classes de agentes para adquirir, converter e amostrar os dados. Na fig. 11 É apresentado o diagrama em blocos da solução proposta usando as tecnologias discutidas no capítulo 2.

Figura 11 – Diagrama macro da solução de sensoriamento proposta.

Fonte: próprio autor.

A fig. 12 apresenta de forma mais detalhada o sistema de sensoriamento acústico e comunicação remota.

Como se pode observar no diagrama da arquitetura, dois componentes principais são necessários para a criação da infraestrutura básica de rede: um nó sensor e um *gateway* conversor de protocolos.

Cada nó sensor é uma unidade autônoma de sensoriamento acústico com um transdutor hidrofone operando em taxas de 8kHz até 96kHz. Simultaneamente à coleta e armazenamento dos dados, as informações relacionadas aos eventos é enviada via rádio para outros nós sensores da rede usando o protocolo 6LowPAN até que cheguem ao *gateway*, de onde os dados podem ser transmitidos para a internet por uma rede IPv4.

Para o *gateway*, foi necessário implementar a tradução entre protocolos de rede usados convencionalmente para a tecnologia de rede de sensores sem fio.

Figura 12 – Arquitetura do sistema de sensoriamento acústico.

Fonte: próprio autor.

Visto que as taxas de aquisição do transdutor podem chegar até 96 kHz torna-se impraticável o envio direto dos sinais acústicos coletados via rede sem fio. Por este motivo os dados devem ser pré-processados localmente e apenas mensagens relacionadas a eventos são enviadas para o servidor remoto, otimizando o consumo de energia da rede autônoma e capacidade de banda, sem perda de informação relevante.

3.1 UNIDADE AUTÔNOMA DE AQUISIÇÃO

A finalidade básica de um monitor acústico passivo (PAM - *Passive Acoustic Monitor*) é a aquisição dos dados provenientes de um transdutor e a gravação destes dados em um espaço limitado em memória. Dependendo do nível de processamento que será feito posteriormente à aquisição dos dados, a quantidade de bits de aquisição e tamanho de palavra devem ser considerados. Outras características não triviais também devem ser levadas em consideração, como por exemplo o tipo de memória que será utilizada, o nível de compressão digital que o sistema deve ter e se isso pode refletir no tempo de processamento e possível interferência ou degradação do estágio de aquisição.

Nas unidades autônomas foi embarcado o sistema operacional em tempo real Contiki que possui um *kernel* baseado em interrupções e gerenciadores de eventos, disparando-os de forma assíncrona e síncrona. Além disso, o *kernel* suporta mecanismos de varredura que podem ser configurados entre os processos assíncronos (DUNKELS; GRONVALL; VOIGT, 2004).

Para controlar as informações em uma rede 6LoWPan, o Contiki utiliza o RPL (*Routing Protocol for Low-Power and Lossy Networks*) para formar um gráfico de roteamento desde o nó principal até seus vizinhos mais distantes, trocando mensagens em forma de *broadcast*. Além disso, o sistema operacional possibilita economia de banda e energia nos nós. O uso de módulos MAC específicos evitam colisão de pacotes e diminuem o tempo de recepção e transmissão dos nós. O protocolo implementado foi o MQTT-SN. Esta escolha foi feita com base na comparação entre os protocolos MQTT-SN e o CoAP, como descrito no item 4.1

3.1.1 Hardware da unidade autônoma de aquisição

Para a captação do sinal acústico foi utilizado um hidrofone piezoelétrico PZT-5 desenvolvido no laboratório de dinâmica e instrumentação da USP (LADIN) com formato cilíndrico de dimensões: diâmetro de 2,5 mm e comprimento de 32 mm, sensibilidade de -198 dB re 1V μ Pa e resposta em frequência plana (± 2 dB) até a frequência de 30 kHz.

O *hardware* proposto tem como base o *kit* de desenvolvimento de baixo custo STM32F429I-DISCO que utiliza um microprocessador CortexM4F, uma placa extensora desenvolvida para comportar o codificador/decodificar (CODEC) de áudio WM8731 de 24 bits, um módulo transceptor RF (rádio-frequência) sub 1 GHz AT86RF212B e um banco de baterias com capacidade total de 10Ah. O diagrama em blocos do sistema projetado pode ser verificado na fig. 13.

Figura 13 – Arquitetura da unidade autônoma de aquisição.

Fonte: próprio autor.

O codec de áudio selecionado é usado em larga escala e foi desenvolvido para aplicações portáteis que requerem pouca energia e alta qualidade de áudio. A frequência de amostragem pode ser configurada de 8 kHz à 96 kHz via I2C e os dados coletados via

I2S podem ser de 16 a 32 bits. O Anexo A.2 apresenta as especificações detalhadas do codec WM8731.

O *transceiver* RF é uma solução integrada que suporta diversos protocolos de rede como o Zigbee, 802.15.4 e o 6LoWPAN e trabalha no range de frequências de 769 a 935 MHz usando modulação QPSK, podendo alcançar uma taxa de transmissão de até 200kbps de acordo com a norma IEEE 802.15.4-2003/2006/2011. O consumo de energia do rádio chega em 0.0594W em transmissão (+5 dBm), 0.03036W em recepção (sensibilidade de -110 dBm) e 0.66 μ W em modo *sleep*, tornando-o atrativo para a aplicação em questão.

As antenas cerâmicas presentes nos módulos AT86RF212B possuem baixo custo, pois são conectadas diretamente ao circuito integrado da placa de circuito impresso. Essas antenas são omnidirecionais, característica desejada para a rede de sensores sem fio, embora não possuam ganho de potência algum. Especificações mais detalhadas do *transceiver* AT86RF212B são apresentadas no Anexo A.1.

Visando a versatilidade nos testes iniciais com a placa protótipo, no codec WM8731 foram conectados os sinais LineIn, LineOut e o sinal MicIn. Vale destacar que as saídas LLineOut e RLineOut possuem baixa impedância, o que permite conexão com cargas de impedância de 10k ohms e capacitâncias de 50 pF.

Levando em consideração a taxa de amostragem máxima de 96 kHz e o ADC configurado para 24 bits, o equivalente em memória é de aproximadamente 576 kB de dados para 1 segundo de sinal. Com isso a placa STM32F429I-DISCO se tornou uma opção válida e barata pois já possui uma memória SDRAM de 64-MBit externa, uma CPU de alta performance ARM Cortex-M4 com FPU para instruções DSP e possui uma porta USB para gravar os dados em um *pendrive*. O Anexo A.3 apresenta as especificações detalhadas do *kit* STM32F429I-DISCO.

O controle do codec de áudio é feito por comandos via interface serial a 2 fios (I2C) e os dados são trafegados via barramento de dados serial a 4 fios e sincronismo (I2S). A interface analógica-digital possui componentes passivos com características para garantir a qualidade da aquisição, removendo os níveis DC dos sinais e posteriormente filtros digitais do tipo passa-alta podem ser habilitados.

A interface e controle do barramento de dados e comunicação do codec de áudio é feita utilizando um microprocessador de 32 bits baseado na arquitetura CortexM4 que pode trabalhar em até 180MHz, além de possuir interfaces para armazenamento dinâmico e estático dos dados adquiridos, visando assim a diminuição de possíveis perdas e degradação de qualidade do sinal. Alguns cuidados foram tomados para garantir a integridade da informação e são apresentados no item 3.1.2 correspondente à arquitetura de *software*.

A fig. 14 apresenta fotos da unidade sensora autônoma construída, incluindo o banco de baterias, hidrofone piezoelétrico e condicionador de sinal.

Figura 14 – Unidade autônoma construída.

Fonte: próprio autor.

3.1.2 Software da unidade autônoma de aquisição

O *software* em sistemas embarcados é uma parte crucial do equipamento e a sua arquitetura determina a qualidade final do projeto. Por este motivo foi implementada uma arquitetura de *software* em camadas ou “*top-down*”, deixando cada módulo de *software* o mais independente possível um do outro, mas ligados diretamente a um bloco de controle de *hardware*. Neste caso, independente da mudança de arquitetura de processador, o módulo pode ser reutilizado sem grandes alterações. Normalmente devem-se mudar apenas as chamadas de baixo nível de drivers para que o módulo continue exercendo sua função principal.

Para gerenciamento das funções do gravador, controle da camada de rede e protocolo de transmissão de dados, foi utilizado o sistema operacional em tempo real CONTIKI-OS e uma série de drivers para controle de periféricos escritos em linguagem de programação ANSI C.

Na fig. 15 pode ser verificado o diagrama macro da arquitetura de *software* que controla o *streaming* de áudio e funcionalidades da unidade autônoma de aquisição.

É possível observar uma camada de abstração de *hardware* (HAL) que tem a finalidade de abstrair ao máximo o modelo do processador que está sendo usado. Para as camadas superiores, simples chamadas de função estão disponíveis para inicialização, escrita e leitura do periférico usado para uma função determinada. Seguindo um pouco mais neste contexto, além das funções básicas de leitura e escrita, foram implementadas

funções de alta velocidade que usam acesso direto à memória (DMA), sem o intermédio do processador, sendo assim há mais tempo livre para o processador fazer outras tarefas concorrentes de forma satisfatória. Como blocos principais do gravador acústico estão os blocos de aquisição via I2S (*streaming* de dados), armazenamento dinâmico via SDRAM (*ring buffer* externo) e armazenamento/controle de dados (gerenciamento de mídia).

Figura 15 – Arquitetura macro do *software* desenvolvido.

Fonte: próprio autor.

O *software* foi dimensionado para armazenar até alguns segundos de dados em banco de memória dinâmica (SDRAM) a fim de possibilitar atrasos nos periféricos de escrita, elevando assim a qualidade do áudio gravado. Esta arquitetura é baseada em uma técnica usada em larga escala em programas de processamento de áudio, nomeada “*Audio Ring Buffer*” e pode ser verificada na fig. 16.

O bloco *StorageManager* é responsável por fazer o gerenciamento das mídias instaladas no sistema, prioritariamente via USB devido à velocidade atual dos periféricos, e gerenciar a gravação de arquivos .WAV no sistema de arquivos FAT32. No momento que o bloco *StreamManager* inicializa o codec de áudio, o *master clock out* começa a sincronizar os dados e, via DMA, estarão disponíveis 2048 bytes a cada 125ms no caso de uma taxa de amostragem de 8kHz. Este processo é gerenciado por um bloco de aquisição interno ao processador mediante interrupções, após o processamento do *buffer* selecionado. Seguido desta interrupção, imediatamente os dados são enviados para outro bloco DMA que grava os dados na SDRAM utilizando o controle circular. Caso o *ring buffer* lote, ele volta a

Figura 16 – Arquitetura do *Ring Buffer* de Áudio.

Fonte: próprio autor.

preencher do início, para assim evitar embaralhamento dos dados de áudio.

No *software* da unidade autônoma foi ainda previsto um bloco de comunicação de rádio, implementado usando o protocolo 6LoWPAN e com possibilidade de adequar outras tecnologias, a exemplo ZigBee, LoRA (BOR; VIDLER; ROEDIG, 2016), etc.

3.1.3 Pré-processamento dos sinais acústicos

Para o pré-processamento dos sinais foi utilizada a técnica de detecção de eventos acústicos baseada em níveis de *threshold* usando cálculo RMS para uma janela de 100 ms do sinal amostrado, de forma contínua. Para tanto fez-se o uso da biblioteca *Cortex Microcontroller Software Interface Standard* (CMSIS), presente na arquitetura de software dos nós sensores (WICKERT, 2015).

O diagrama do *software* detector proposto pode ser verificado na fig. 17.

É possível observar que o nível de *threshold* é configurável, assim como o tamanho do *buffer* de dados, este mediante taxa de aquisição, para garantir, assim, a janela de tempo desejada do sinal.

Uma vez detectado o nível de RMS que satisfaça o parâmetro de *threshold* (V_{Th}) configurado, uma chamada de função é enviada ao módulo de software responsável pela notificação deste evento. É possível usar esse parâmetro para iniciar gravações diminuindo assim o espaço utilizado para armazenamento dos dados PCM (*Pulse-code modulation*). O tempo de gravação também pode ser configurado e faz com que não seja necessário mais

Figura 17 – Arquitetura de *software* do detector.

Fonte: próprio autor.

de um nível de *threshold*, diminuindo assim o uso do processamento dentro das janelas de aquisição.

Como parâmetros da mensagem de notificação de evento estão presentes: *timestamp*, que corresponde ao horário configurado no relógio interno da unidade, indicador de início ou final da gravação e espaço disponível na memória de armazenamento.

3.2 GATEWAY

Na tecnologia de transmissão IP os pacotes devem ser encaminhados dentro de uma mesma rede local por *switches*, *hubs* ou *bridges*. Se o endereço de destino não está presente na rede local, o pacote deve ser encaminhado para um *gateway*, que é responsável por determinar como o pacote será entregue para o destino (DONAHUE, 2007).

No contexto da Internet das Coisas (IoT) os nós sensores se comportam como *switches* para encaminhar os dados até o *gateway* que por sua vez encaminha os dados para fora da rede de sensores. Dentro do *gateway* normalmente há recursos suficientes que permitem liberdade no desenvolvimento, que podem instanciar um servidor intermediário que faz um pré-processamento dos dados ou apenas podem converter e retransmitir os protocolos nativos.

3.2.1 Hardware do gateway

O *hardware* proposto tem como base uma placa de baixo custo Raspberry Pi Zero W, com processador ARM de 1GHz e 512MB de RAM, além de possuir conectividade Wifi e Bluetooth (PI, 2017) e um módulo transceptor RF sub 1 GHz AT86RF212B. O diagrama em blocos do sistema projetado e uma imagem do sistema montado podem ser verificados na fig. 18.

Figura 18 – Arquitetura do *hardware gateway* desenvolvido.

Fonte: próprio autor.

A alimentação é ligada via conector microUSB, o que facilita a compatibilidade com banco de baterias ou fontes AC/DC comuns em carregadores celulares.

Comparando com *gateways* de mercado, poucas soluções robustas são encontradas. Apenas duas empresas de fácil acesso oferecem equipamentos completos para tal finalidade, *Saker Gateway* (WEPTech, 2018) e *Waspmote Mote Runner* (LIBELIUM, 2018) que contam com processadores limitados em RAM e conectividade. Grandes empresas como a CISCO também fornecem *gateway's* porém o custo é muito elevado (RAMÍREZ, 2014). A arquitetura de *hardware* da OpenLabs (OPENLABS, 2014) é bem próxima do que foi selecionado, com mudanças apenas em frequência, usando o transceptor AT86RF233 que trabalha em banda de 2.4GHz. Em questão do *software* há diversas limitações relacionadas à maturidade do desenvolvimento do SO Linux, uma vez que as ferramentas disponíveis ainda não possuem suporte a diversos protocolos 6LoWPAN.

3.2.2 *Software* do gateway

A Raspberry Pi Zero W roda o sistema operacional Linux (SHACKLETTE, 1995) e o suporte a dispositivos e tecnologias IoT ainda não é oferecida por completo. Percebe-se que o principal motivo disso é a inviabilidade de suportar dispositivos e protocolos heterogêneos, além disso, o próprio sistema operacional requer equipamentos robustos e capazes de alocar grandes quantidades de memória para diversas tarefas, sejam mecanismos internos de controle e sincronismo, ou recursos em aplicações do usuário.

Como alternativa a essas limitações, foi desenvolvida uma camada de aplicação que roda o sistema operacional Contiki como um processo dentro do Linux, permitindo assim

o uso de todas as características de baixo consumo e protocolos já desenvolvidos e estáveis.

O diagrama do *software* projetado pode ser verificado na fig. 19.

Figura 19 – Arquitetura de *software* do *gateway*.

Fonte: próprio autor.

Pode-se compreender as seguintes camadas: 1, acesso direto aos recursos de *hardware*; 2, camada de *software* que gerencia os recursos abaixo dela e provê uma interface de programação para as aplicações; 3, camadas que gerenciam recursos de aplicação; (apenas essa camada possui acesso direto aos dispositivos); 4, camada de aplicação onde é executada toda a pilha de controle da rede 6LoWPAN e instanciada a aplicação do *gateway*.

Com essa arquitetura, os recursos e possibilidades tornam-se virtualmente ilimitadas na camada de aplicação do *gateway*, uma vez que estão disponíveis todos os recursos dos sistemas operacionais Linux e Contiki simultaneamente.

3.3 SERVIDOR REMOTO

O método mais comum de administração de sistemas é acessar um sistema remoto através da rede por meio de protocolos cliente-servidor que permitam acesso privilegiado a um ou mais serviços em execução no sistema de destino (RAMILLI; PRANDINI, 2009). A orquestração de serviços WEB tornou-se uma tarefa difícil com o surgimento de diversos protocolos no decorrer da história, cada um com seus pontos fortes e fracos. Ainda há fatores latentes de interoperabilidade e segurança que nem sempre podem ser alcançados facilmente (SEELY; BY-SHARKEY, 2001).

Tendo conhecimento dos desafios naturais do mundo WEB, foi necessária uma adaptação dos protocolos provenientes do cenário IoT a fim de garantir a qualidade e integridade dos dados trafegados a partir dos nós sensores até o servidor remoto. É então proposto um modelo de arquitetura de serviços que permite escalabilidade e fácil

manutenção. As tecnologias sugeridas são conhecidas por milhares de desenvolvedores e usadas em diversos serviços de alta demanda.

3.3.1 *Software* do servidor remoto

A base de *software* que roda no servidor foi dividida em duas partes, *backend* e *frontend*. Os tipos de linguagem de programação e ferramentas podem ser distintas ou não. Ainda sendo a mesma linguagem em ambas as partes, deve haver um protocolo específico de comunicação entre elas, como por exemplo o REST e WebSocket (RICHARDSON; RUBY, 2008; FETTE, 2011). Desta forma o *frontend* pode ser desacoplado da arquitetura e ser executado em qualquer dispositivo. O diagrama do *software* proposto pode ser verificado na fig. 20.

Figura 20 – Arquitetura de *software* do servidor.

Fonte: próprio autor.

Como estrutura base do *backend* foi selecionado o NODE.js, um *framework* em ambiente de programação Javascript, baseado no motor interpretador V8 da Google, implementado em C e C++, com foco no desempenho e baixo uso de memória (TILKOV; VINOSKI, 2010). A modularidade do NODE.js permite que módulos sejam carregados dinamicamente antes de inicializar o serviço.

Para a comunicação cliente-servidor foi selecionada a tecnologia Websocket que permite comunicação por duas vias e pode encapsular qualquer frame de dados dentro dele como, por exemplo, o protocolo MQTT ou JSON (*JavaScript Object Notation* - Notação de Objetos JavaScript). É possível o uso desta mesma tecnologia para receber e enviar dados do *gateway*.

Para a base de *software* do *frontend* foi selecionado o Angular.js, um *framework* em ambiente de programação Javascript que aplica conceitos inovadores em controle assíncrono de chamadas e validação de dados nativa, além de possuir uma arquitetura modular (DARWIN; KOZLOWSKI, 2013).

Com estas camadas de *software* é possível uma implementação segura dos serviços, garantindo as características desejadas.

A fig. 21 apresenta uma proposta de interface no servidor remoto contendo as informações enviadas pelas unidades autônomas.

Figura 21 – Proposta de interface no servidor remoto para apresentação dos dados recebidos da rede.

PAMApp ☰ Menu				
MAC Unidade	Espaço Disponível (%)	Tipo do Evento	Timestamp Evento (s)	Hora do Registro
:46:e3	95	Início Gravação	1532368930	2018-07-23 18:02:10
:57:b1	87	Início Gravação	1532371256	2018-07-23 18:40:56
:97:c4	70	Início Gravação	1532372205	2018-07-23 18:56:45
:46:e3	93	Término Gravação	1532369230	2018-07-23 18:07:10
:57:b1	85	Término Gravação	1532371556	2018-07-23 18:45:56
:97:c4	68	Término Gravação	1532372505	2018-07-23 19:01:45

Fonte: próprio autor.

Como parâmetros de informação da rede estão presentes: *mac address*, que é a identificação da unidade autônoma, espaço de gravação disponível em memória, tipo do evento e *timestamp* recebidos da unidade autônoma. O *timestamp* é a contagem interna de tempo interno da unidade (*epoch time*), em segundos, correspondente à data/horário do registro e pode ser comparado com a data/hora de gravação do evento pelo servidor. O espaço disponível em memória pode servir como estimativa de funcionamento, uma vez que sem espaço disponível, os eventos serão enviados, porém não serão gravados.

Através dessa interface um usuário pode identificar as ocorrências de eventos acústicos na rede ao longo do tempo, o que permite avaliar, em tempo real, que regiões e horários apresentam maior atividade acústica.

Fechando a metodologia utilizada neste trabalho, o desempenho da rede de sensores desenvolvida foi caracterizado considerando-se os seguintes aspectos: Comparação de protocolos de comunicação, avaliação do link de rádio, avaliação da autonomia energética

e armazenamento da unidade autônoma e avaliação da comunicação entre os dispositivos da rede. Esses aspectos estão descritos com maiores detalhes no capítulo seguinte.

4 AVALIAÇÃO DO DESEMPENHO DA REDE DE SENSORES: RESULTADOS E ANÁLISE

Foi criada uma rede para monitoramento acústico passivo composta por três unidades sensoras autônomas e que podem ser conectadas a outra rede através de um *gateway*. As unidades de monitoramento são compostas por um cilindro estanque, um hidrofone externo e uma eletrônica embarcada para condicionamento, digitalização e armazenamento de sinais acústicos. Também pode executar pré-processamento dos sinais e transmissão sem fio em tempo real das condições de funcionamento da unidade sensora e detecção de eventos acústicos. A unidade fixa ou *gateway* é responsável por gerenciar uma topologia ad-hoc da rede de sensores.

4.1 ANÁLISE E COMPARAÇÃO DOS PROTOCOLOS DE COMUNICAÇÃO

É imprescindível verificar o comportamento de uma rede de sensores sem fio e os meios de transporte de dados nela presentes, garantindo a entrega das características monitoradas. Para isso, foram comparados os protocolos MQTT-SN e CoAP.

Foi utilizada a ferramentas de simulação COOJA, que faz parte da distribuição do pacote CONTIKI-OS. A fig. 22 ilustra o ambiente de simulação onde são instanciados os nós sensores usando características de *hardware* também simuladas. A camada de roteamento usada foi a RPL(WINTER, 2012), responsável pelo gerenciamento de rotas da rede 6LoWPAN (MONTENEGRO et al., 2007). Para evitar qualquer interferência no envio dos dados, não foi configurada no CONTIKI nenhuma camada RDC (*Radio Duty Cycle*).

Figura 22 – Ambiente de Simulação COOJA.

Fonte: próprio autor.

A execução da aplicação, comunicação de rede e periféricos podem ser simulados em tempo real para uma grande quantidade de nós. Entre os tipos de simulação estão presentes quatro tipos de modelos de propagação de rádio: *Unit Disk Graph Medium* (UDGM) *Distance Loss* no qual a interferência é considerada, porém os pacotes são transmitidos e recebidos com probabilidade de sucesso, UDGM *Constant Loss*, onde o alcance de transmissão é determinado por uma circunferência ideal e o sinal alcança todo o perímetro, *Directed Graph Radio Medium* (DGRM) no qual a razão de sucesso é determinada de forma assimétrica em cada *link* por meio de parametrização dos *delays* de propagação e *Multi-path Ray-tracer Medium* (MRM) no qual a potência de recepção é determinada usando a fórmula de Friis e os obstáculos impõem atenuações nos sinais dos sensores (AL-FUQAHA et al., 2015).

O software desenvolvido para esta etapa foi dividido em três partes: MQTT-SN *Client*, MQTT-SN *Gateway* e MQTT *Broker*. As pilhas *CLIENT* e *GATEWAY* rodam no sistema operacional CONTIKI-OS e a pilha *BROKER* roda nativamente no sistema *host* Linux, onde a plataforma de testes foi instalada. Como ponte entre o ambiente de simulação COOJA e o *host* Linux, foi necessário usar os utilitários TUNSLIP, que faz um tunelamento IP via serial e o TAYGA, que faz a tradução do IPv6 para IPv4, uma vez que o *broker* MQTT instanciado no *host* roda apenas com *stack* IPv4. Para os testes do CoAP, todos os utilitários estavam presentes dentro do ambiente COOJA.

Os testes foram realizados no simulador COOJA usando dois tipos de nós criados com a base no *hardware* WISMOTE. O primeiro tipo de nó simula as unidades sensoras na rede e o segundo simula o *border router*, que fará a conexão com outra rede para envio dos dados. Para os testes do protocolo MQTT-SN, foram usados os parâmetros QoS = 1, porta 1884 UDP, envio de mensagens de 10 em 10 segundos com até 4 tentativas de registro de tópico entre os nós sensores e o MQTT *broker*. Para os testes do protocolo CoAP, foram usados os parâmetros de porta 61616 UDP e requisições de 10 em 10 segundos.

Foram testados 1, 5, 10, 15, 20, 30 e 40 nós com alcance direto ao *gateway* e distribuídos ao seu redor em forma de elipse a fim de verificar a entrega dos dados para os servidores MQTT e CoAP bem como outras características de transmissão/recepção. A topologia estrela foi escolhida, pois é tipicamente encontrada em rede de sensores para monitoramento de parâmetros marinhos (ALBALADEJO et al., 2010). Foi utilizado o comando CoAP POST e MQTT-SN PUBACK para realizar o envio das mensagens de texto.

Os dois protocolos foram avaliados comparando-se os seguintes indicadores de desempenho em função da quantidade de nós sensores: atraso (*delay*), latência, quantidade de pacotes trafegados, quantidade de bytes trafegados e consumo de energia.

A fig. 23 apresenta as curvas de latência para cada protocolo. Compreende-se a

latência, *Round Trip Time* (RTT), como o tempo entre o início da comunicação até o momento de recepção da mensagem pelo servidor de aplicação, considerando o processamento do protocolo (CARO et al., 2013).

Figura 23 – Resultados de latência utilizando os protocolos MQTT-SN e CoAP.

Fonte: próprio autor.

O protocolo MQTT-SN apresentou latência de 36 ms na entrega da mensagem para um nó. Levou-se em consideração a latência entre as mensagens PUBLISH e PUBACK. Pode-se verificar um acréscimo de, aproximadamente, 2 ms a cada nó adicionado, levando em consideração o processamento do servidor MQTT.

Para os testes do protocolo CoAP, a latência foi de, aproximadamente, 231 ms para a resposta do dispositivo considerando novamente um único nó após a requisição do servidor. Dessa maneira, o protocolo CoAP é 195 ms mais lento que o MQTT-SN para as condições de simulação aplicadas. Essa diferença de valores se mantém aproximadamente a mesma para qualquer quantidade de nós na rede. Sendo assim, foi possível verificar que o MQTT-SN é viável para a aplicação proposta e apresenta menor latência na entrega dos dados. Pode-se sugerir que o menor tempo de latência do protocolo MQTT-SN ocorre devido ao menor tamanho do pacote necessário para enviar mensagens entre os nós e o *gateway*.

A fig. 24 mostra o tempo de tráfego na rede para os dois protocolos. Pode-se notar que ambos os protocolos apresentam praticamente o mesmo desempenho com relação ao tempo de tráfego na rede. Isso ocorre, pois ambas as pilhas de protocolos utilizam a mesma camada de roteamento (RPL), demonstrando que o tempo de atraso se deve

exclusivamente ao processamento dos protocolos tanto nos nós transmissores como no *gateway*.

Figura 24 – Comparação entre os tempos de tráfego na rede das arquiteturas MQTT-SN e CoAP.

Fonte: próprio autor.

Durante os testes uma mesma mensagem foi enviada dos nós sensores para o *gateway*. A mesma mensagem consiste numa frase concatenada com os últimos dois bytes do endereço IPv6 do nó sensor. Quando se compara a quantidade de pacotes trafegados (fig. 25) o MQTT-SN apresenta um tráfego 2,15 vezes maior que o CoAP, para qualquer quantidade de nós. Isso ocorre devido à fragmentação de mensagem realizada pelo MQTT-SN. O protocolo CoAP, por ser baseado no protocolo HTTP, encaminha mensagens mais longas, pois possui métodos que garantem a entrega. O MQTT-SN encaminha mensagens mais curtas, utilizando menor banda de comunicação e também pode garantir a entrega dos dados, desde que configurado o QOS.

A fig. 26 apresenta a quantidade de bytes trafegados. O protocolo MQTT-SN transmite a mesma mensagem utilizando somente 70 % da quantidade de bytes do protocolo CoAP. Isso é esperado, pois o CoAP possui maior número de bytes para encaminhar os cabeçalhos e as mensagens. Dessa forma, o MQTT-SN realiza o envio da mesma informação com menor quantidade de dados.

Com relação ao consumo de energia, a fig. 27 mostra que o consumo do MQTT-SN é levemente melhor, sendo aproximadamente 10 % menor que o consumo do CoAP. Como mencionado anteriormente, as características do MQTT-SN, entre elas menor tráfego de bytes e menor latência, resultam no menor consumo de energia dos nós.

Figura 25 – Comparação entre as quantidades de pacotes trafegados utilizando os protocolos MQTT-SN e CoAP.

Fonte: próprio autor.

Figura 26 – Comparação entre as quantidades de bytes trafegados utilizando os protocolos MQTT-SN e CoAP.

Fonte: próprio autor.

A comparação entre os protocolos pode ser sintetizada na tabela 7. Nessa tabela são detalhados os resultados obtidos para 40 nós.

Figura 27 – Comparação entre os consumos de energia dos nós sensores utilizando os protocolos MQTT-SN e CoAP.

Fonte: próprio autor.

Dessa forma, o protocolo MQTT-SN se mostra como uma alternativa mais adequada para implementação em aplicações IoT para redes de sensores sem fio em ambientes hostis, em especial, para monitoramento acústico marinho, sendo o protocolo escolhido para implementação no sistema de monitoração proposto neste trabalho.

4.2 ANÁLISE DO ENLACE DE RÁDIO

Foram realizados testes do *link* de rádio na Raia Olímpica da USP, com uma unidade fixa, fora da água, e uma unidade autônoma, na água, como ilustrado na fig. 28.

A caracterização do enlace de RF foi realizada através do envio de mensagens UDP (*User Datagram Protocol*) *multicast* entre dois rádios na frequência de 902 MHz com intervalos de 5 segundos.

Os testes de caracterização do enlace foram divididos em duas fases. Na primeira fase foi analisado o nível de sinal RSSI (*Received Signal Strength Indication*) do receptor mediante o aumento da distância entre o nó autônomo e a estação base. Além disso, foi avaliada a perda de pacotes quando a intensidade do sinal estava menor que a sensibilidade do receptor. As métricas utilizadas neste teste foram: quantidade de pacotes perdidos e os erros de CRC (*Cyclic Redundancy Check*) de pacotes recebidos em comparação com a quantidade de pacotes enviados em determinado intervalo de tempo. Com o nível de potência recebida foi possível determinar a distância máxima aceitável entre os dois

Tabela 7 – Análise comparativa do desempenho dos protocolos CoAP e MQTT-SN a partir das simulações para uma rede com 40 nós.

Parâmetros do Protocolo	MQTT-SN	CoAP	Comparativo
Latência da rede (ms)	271	473	MQTT-SN é 200 ms mais rápido comparado ao CoAP.
Tempo de tráfego na rede (ms)	20	20	Ambos os protocolos possuem aproximadamente o mesmo tempo de tráfego na rede.
Pacotes trafegados	366	170	O MQTT-SN fragmenta a informação em maior número de pacotes, cerca de 2,15 vezes.
Bytes trafegados	1464	2040	O MQTT-SN transmite a mesma mensagem com apenas 70 % dos bytes utilizados no CoAP.
Consumo de energia dos nós (mW)	42,4	46,7	O MQTT-SN apresenta consumo de energia 10 % menor do que o CoAP.

Fonte: próprio autor.

transceptores.

A fig. 29 apresenta a intensidade do sinal de recepção conforme o aumento da distância entre a unidade autônoma e estação base. O transmissor foi configurado com ganho de 10dBm. Nota-se que o receptor foi incapaz de receber pacotes para distâncias acima de 100m. O limite de sensibilidade medido foi de -95dBm, embora o receptor seja capaz de medir sinais com potência até -110dBm, segundo características informadas pelo fabricante.

A análise de perda de pacotes e erros foi feita de forma conjunta. Com um intervalo fixo de espera de pacotes na recepção, foi possível verificar as perdas de pacotes e erros de CRC (pacotes recebidos e descartados devido à má integridade). A fig. 30 demonstra essas medidas, bem como a taxa de erro na recepção (pacotes perdidos e a quantidade de erros CRC em relação ao número de pacotes enviados) para uma distância de 50m.

É possível observar que a quantidade de pacotes perdidos e o número de erros CRC aumentam com a quantidade de pacotes enviados. Entretanto a taxa de erros ficou abaixo de 9.3 % e tende a se estabilizar para uma quantidade maior de pacotes enviados, flutuando em torno de 8 %. Isso indica que a qualidade da rede é boa e a degradação ocorre apenas se um dos nós sofre uma interferência brusca no sinal.

Figura 28 – Testes conduzidos na Raia Olímpica da USP.

Fonte: próprio autor.

4.3 AVALIAÇÃO DE AUTONOMIA E ARMAZENAMENTO

Foi caracterizado o consumo de potência da unidade autônoma. Nesta fase foi observada a potência usada pelo transceptor de rádio no pico de transmissão e a partir deste parâmetro foi estimada a autonomia do equipamento, em termos de horas de funcionamento.

Nos picos de corrente do transceptor de RF foi possível medir um consumo de 81.2 mA com polarização de 3,3 V (o que resulta em 268 mW), totalizando, aproximadamente, 125 horas de funcionamento sem interrupções. Dessa forma, o sistema se mostra viável para o uso em ambientes marítimos e expedições de caracterização de paisagem acústica. É interessante mencionar a necessidade do uso de técnicas de *duty cycle* do *link* RF com o intuito de otimizar o uso da fonte de energia.

Com relação à capacidade de armazenamento, ela pode ser estimada mediante a taxa de aquisição e quantidade de bits do canal de áudio do CODEC. Pode ser verificado na fig. 31 uma comparação entre as taxas de aquisição possíveis e quantidade de espaço necessário para um determinado tempo de gravação.

Para a finalidade de análise dos sinais em tempo real e maiores tempos de arma-

Figura 29 – Nível de Sinal RSSI em função da distância entre o nó sensor e o gateway.

Fonte: próprio autor.

Figura 30 – Taxa de erro na recepção do sinal para uma distância fixa de 50 metros.

Obs.: Taxa de erro = (Erros CRC + Pacotes Perdidos)/Pacotes Enviados.

Fonte: próprio autor.

Figura 31 – Tempo de armazenamento em função da memória disponível, para diferentes taxas de amostragem do sinal.

Fonte: próprio autor.

zenamento, pode-se atestar que as taxas de aquisição menores tornam-se atraentes. Por exemplo, para 125 horas de registro, gravadas a 48kHz, seria necessária uma memória de aproximadamente 48GB.

4.4 AVALIAÇÃO DE ROTEAMENTO DA REDE

Para a caracterização do funcionamento das unidades como rede, foi verificado o fluxo de mensagens RPL em dois cenários: todos os nós com alcance ao *gateway* e um nó sem alcance direto. Com isso pode ser analisada a capacidade da rede em propagar os dados e se os nós escolhem a melhor rota para envio dos dados, como planejado. Os arranjos das redes testadas podem ser verificados na fig. 32.

No primeiro arranjo (Fig. 32a), todos os nós tinham alcance direto ao *gateway*, já no segundo arranjo (Fig. 32b), as mensagens do nó 3 podem trafegar por duas rotas, de menor e maior custo, através do nó 1 ou nó 2, respectivamente. Os parâmetros usados que caracterizam o custo da rota são denominados LQI (*Link Quality Indication*) e RSSI (*Received Signal Strength Indication*), que variam de 0 a 255 e podem ser recebidos diretamente no protocolo de comunicação do transceptor de rádio utilizado, de forma síncrona com o recebimento dos dados da rede. Estes parâmetros também podem ser chamados de metadados dos pacotes que trafegam entre os nós e o *gateway*. A partir dos metadados é então calculado o *rank* de cada nó sensor usando uma função objetiva (FO)

Figura 32 – Cenários dos testes RPL.

Fonte: próprio autor.

especificada no protocolo de roteamento RPL (THUBERT, 2012). Quanto menor o rank, melhor a qualidade do roteamento.

Os *rankings* e rotas coletados durante os testes foram relacionadas na tabela 8.

Tabela 8 – *Rankings* e rotas RPL, cenários A e B

Cenário	Dispositivo	<i>rank</i>	Rotas
A	GW	128	N1, N2, N3
A	N1	277	-
A	N2	273	-
A	N3	277	-
B	GW	128	N1, N2, N3
B	N1	261	N3
B	N2	277	-
B	N3	434	-

Fonte: próprio autor.

Verificando o fluxo do protocolo RPL, as mensagens DIO (*DODAG Information Object*) contém as informações gerais que permitem o nó descobrir uma instância RPL, são enviadas por todos os nós para seus vizinhos via *broadcast*. As mensagens DIS (*DODAG Information Solicitation*) são usadas para solicitar informações DIO dos nós vizinhos também via *broadcast*. Por fim, as mensagens DAO (*Destination Advertisement Object*) são usadas para propagar as rotas a serem usadas até o nó principal (*root*).

Pode ser verificado que tanto para o cenário A quanto para o cenário B o nó 1 envia uma solicitação DIS para todos os nós começarem a se reportar. As informações de cada nó são então propagadas na rede e de acordo com as métricas estabelecidas do enlace, as mensagens DAO são publicadas e as rotas são criadas consequentemente. Apenas no

cenário B foi criada a rota do nó 3 para o nó 1, uma vez que as métricas do nó 2 não favoreciam o roteamento dos dados do nó 3.

Ficou atestada então a capacidade de roteamento e disseminação dos dados entre os nós sensores tanto de forma direta, como por necessidade de roteamento usando as métricas do RPL.

4.5 AVALIAÇÃO DO PRÉ-PROCESSAMENTO DE SINAIS ACÚSTICOS

A avaliação do pré-processamento de sinais foi realizada a fim de verificar o funcionamento do bloco de detecção RMS implementado. Para isso foram utilizados dois arquivos WAVE de referência, um com vocalizações de golfinho e outro com ruído de motor de barco.

Com o intuito de caracterizar a não ocorrência de eventos foi adicionado um ruído de fundo do oceano nos primeiros segundos dos arquivos, usando o *software* de áudio Audacity (TEAM, 2015).

Os sinais e respectivos espectrogramas podem ser verificados na fig. 33. Ambos arquivos foram baixados de uma base de dados acústicos disponível em (OCEAN, 2018) e possuem um canal PCM de 16 bits e taxa de amostragem 44 kHz.

Figura 33 – Sinais e Espectrogramas dos arquivos de áudio analisados.

(a) Sinal golfinho (b) Espectrograma golfinho (c) Sinal barco (d) Espectrograma barco

Fonte: próprio autor.

Para os cálculos RMS ficou definido um tamanho de *buffer* com 4096 bytes, o que equivale a uma duração de janela de aproximadamente 100 ms para a taxa de aquisição dos arquivos. Com isso, os arquivos de áudio foram reproduzidos usando a saída *line out* do computador ligada diretamente ao *line in* da unidade sensora. Pode ser verificado na fig. 34 o sinal original e os níveis RMS calculados.

Para esses exemplos de sinais, pode-se estabelecer como *threshold* (limiar de disparo)

Figura 34 – Sinais e Valores RMS dos sinais de áudio analisados.

(a) Golfinho

(b) Barco

Fonte: próprio autor.

o valor RMS $V_{th} = 3000$ unidades PCM, acima do qual deve ser iniciada a gravação pelo tempo definido e gerada uma notificação de evento acústico para transporte até o servidor remoto.

O nível de *threshold* para o início da gravação, bem como o intervalo de tempo mínimo de gravação são parâmetros que podem ser definidos previamente pelo usuário, de acordo com o estudo acústico que se pretende realizar. Vale destacar que este recurso de pré-tratamento dos sinais para gravação automática apenas acima de determinado limiar é uma proposta original deste trabalho, que contribui para a economia de memória e energia das unidades autônomas.

5 CONCLUSÕES

Foi proposta uma rede de sensores sem fio para captação de sons aquáticos, pré-tratamento dos sinais e envio de notificações de eventos por rádio em tempo real para um servidor na internet. Foram feitas simulações para comparar o desempenho de protocolos de comunicação e foram construídos três nós sensores e o *gateway* da rede.

Nas simulações foram comparados os protocolos MQTT-SN e CoAP onde o MQTT-SN mostrou latência da ordem de 200 ms menor que o CoAP, mesmo trafegando uma quantidade maior de pacotes. Também apresentou consumo energético 10 % menor que o CoAP. Dessa forma, o protocolo MQTT-SN se mostrou como uma alternativa mais adequada para implementação em aplicações IoT para redes de sensores sem fio para monitoramento acústico marinho.

Por meio de testes em campo foi avaliada a qualidade do *link* de rádio, consumo de energia, capacidade de armazenamento necessária para diferentes taxas de aquisição e eventuais limitações de distância e interferências, bem como os atrasos para a entrega dos protocolos M2M analisados e capacidade de roteamento da rede.

O *link* de rádio alcançou em torno de 100 metros. Para 50 metros foi verificada taxa de erro abaixo de 9.3 %. Com relação à autonomia, pode-se estimar que o nó sensor desenvolvido pode operar por aproximadamente 125 horas ininterruptamente.

As mensagens enviadas entre os nós e o servidor durante a etapa de simulação foram delimitadas a 4 bytes, portanto é interessante a realização de mais testes em ambiente real aumentando a quantidade de dados enviados a fim de analisar a capacidade de transmissão e fragmentação dos protocolos e não apenas o atraso de uma única mensagem.

Nenhum amplificador de potência foi utilizado no estágio de transmissão RF, o que limita a distância entre nós em 100 metros aproximadamente. Mesmo assim a rede pode ser estendida com múltiplos saltos, dependendo apenas da distribuição física dos nós. Usando-se 3 unidades sensoras e um *gateway* pode-se alcançar em teoria uma cobertura de até 400 metros.

Usando-se o RPL, existe ainda a possibilidade de um nó encontrar uma rota alternativa caso haja interrupção do sinal direto, garantindo a entrega dos dados mesmo com obstáculos móveis. Outras características como configurações de ciclo de envio e recepção de mensagens devem ser bem definidas dentro do sistema operacional (CONTIKI-OS) a fim de garantir a melhor taxa de economia de energia e entrega de dados.

Foi possível implementar e testar o funcionamento de um detector RMS para pré-processamento dos dados acústicos usando os recursos de DSP da unidade autônoma,

o que pode resultar em economia de armazenamento no processo de gravação e economia de energia da unidade autônoma, uma vez que a gravação só ocorre para sinais com valor RMS acima de certo valor de *threshold* escolhido.

De forma geral, é possível destacar que o sistema desenvolvido não está limitado apenas à detecção de eventos, pois, uma vez possuindo um processador de alta capacidade e um sistema de aquisição robusto e configurável, implementações mais sofisticadas de classificação podem ser alcançadas.

Como melhorias e possibilidades de continuidade deste trabalho podem ser destacadas: análise da propagação do sinal de rádio e desempenho da rede em função de fatores ambientais (ondulação do mar, névoa, chuva, etc.), implementação completa do servidor remoto, desenvolvimento de uma metodologia de agregação de dados no *gateway* com a finalidade de diminuição do uso de banda, estudo e validação de métodos de classificação na unidade autônoma, métodos de compactação de arquivos de áudio gravados e estudo de possíveis integrações com softwares de monitoramento acústico existentes.

Nesse contexto, o presente trabalho aborda diferentes áreas que se inter-relacionam e aponta para possibilidades de aplicações mais abrangentes sob o conceito de Internet das Coisas.

REFERÊNCIAS BIBLIOGRÁFICAS

- AL-FUQAHA, A. et al. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, IEEE, v. 17, n. 4, p. 2347–2376, 2015.
- ALBALADEJO, C. et al. Wireless sensor networks for oceanographic monitoring: A systematic review. *Sensors*, Molecular Diversity Preservation International, v. 10, n. 7, p. 6948–6968, 2010.
- AU, W. W.; HASTINGS, M. C. *Principles of marine bioacoustics*. [S.l.]: Springer, 2008.
- BAMBER, S. S.; SHARMA, A. K. Comparative performance investigations of different scenarios for 802.15. 4 wpan. *arXiv preprint arXiv:1004.3555*, 2010.
- BJØRNØ, L. Sources of low-frequency sound in the sea. In: *Underwater Acoustic Data Processing*. [S.l.]: Springer, 1989. p. 25–37.
- BLUM, J. *Exploring Arduino: tools and techniques for engineering wizardry*. [S.l.]: John Wiley & Sons, 2013.
- BOR, M.; VIDLER, J. E.; ROEDIG, U. Lora for the internet of things. Junction Publishing, 2016.
- BORMANN, C.; ERSUE, M.; KERANEN, A. *Terminology for constrained-node networks*. 2014.
- BUETTNER, M. et al. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In: ACM. *Proceedings of the 4th international conference on Embedded networked sensor systems*. [S.l.], 2006. p. 307–320.
- BURATTI, C. et al. An overview on wireless sensor networks technology and evolution. *Sensors*, Molecular Diversity Preservation International, v. 9, n. 9, p. 6869–6896, 2009.
- BUTLER, J. L.; SHERMAN, C. H. *Transducers and arrays for underwater sound*. [S.l.]: Springer, 2016.
- CARELS, D. et al. Support of multiple sinks via a virtual root for the rpl routing protocol. *EURASIP Journal on Wireless Communications and Networking*, Springer, v. 2014, n. 1, p. 91, 2014.
- CARO, N. D. et al. Comparison of two lightweight protocols for smartphone-based sensing. In: IEEE. *Communications and Vehicular Technology in the Benelux (SCVT), 2013 IEEE 20th Symposium on*. [S.l.], 2013. p. 1–6.
- DARWIN, P. B.; KOZLOWSKI, P. *AngularJS web application development*. [S.l.]: Packt Publ., 2013.
- DONAHUE, G. *Network warrior*. [S.l.]: "O'Reilly Media, Inc.", 2007.
- DUNKELS, A. The contikimac radio duty cycling protocol. Swedish Institute of Computer Science, 2011.

- DUNKELS, A. et al. Low-power ipv6 for the internet of things. In: IEEE. *Networked Sensing Systems (INSS), 2012 Ninth International Conference on*. [S.l.], 2012. p. 1–6.
- DUNKELS, A.; GRONVALL, B.; VOIGT, T. Contiki-a lightweight and flexible operating system for tiny networked sensors. In: IEEE. *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*. [S.l.], 2004. p. 455–462.
- EL-HOIYDI, A. Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks. In: IEEE. *Communications, 2002. ICC 2002. IEEE International Conference on*. [S.l.], 2002. v. 5, p. 3418–3423.
- FERREIRA, R. *Modelagem da Propagação do Som em águas Rasas*. 2002.
- FETTE, I. The websocket protocol. 2011.
- GODOI, S. S. d. *Oceanografia física observacional*. 2010. Curso de pós-graduação em oceanografia física.
- GUPTA, S. K.; SINHA, P. Overview of wireless sensor network: a survey. *Telos*, v. 3, n. 15, p. 38mW, 2014.
- HAHM, O. et al. Operating systems for low-end devices in the internet of things: a survey. *IEEE Internet of Things Journal*, IEEE, v. 3, n. 5, p. 720–734, 2016.
- HARARI, J. *Oceanografia Física Descritiva, Instituto Oceanográfico da Universidade de São Paulo*. 2007.
- HASIBUAN, A. et al. Design and implementation of modular home automation based on wireless network, rest api, and websocket. In: IEEE. *Intelligent Signal Processing and Communication Systems (ISPACS), 2015 International Symposium on*. [S.l.], 2015. p. 362–367.
- HERSENT, O.; BOSWARTHICK, D.; ELLOUMI, O. 6lowpan and rpl. *The Internet of Things: Key Applications and Protocols*, Wiley Online Library, p. 193–208, 2011.
- HILL, J. L.; CULLER, D. E. Mica: A wireless platform for deeply embedded networks. *IEEE micro*, IEEE, v. 22, n. 6, p. 12–24, 2002.
- HOEPFMAN, J.-H.; JACOBS, B. Increased security through open source. *Communications of the ACM*, ACM, v. 50, n. 1, p. 79–83, 2007.
- HUANG, P. et al. The evolution of mac protocols in wireless sensor networks: A survey. *IEEE communications surveys & tutorials*, IEEE, v. 15, n. 1, p. 101–120, 2013.
- HUGHES, B. et al. Towards real-time middleware for vehicular ad hoc networks. In: ACM. *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*. [S.l.], 2004. p. 95–96.
- HUNT, F. V. *Electroacoustics: the analysis of transduction, and its historical background*. [S.l.]: Acoustical Society of America, 1954.
- INCEL, O. D. et al. Mc-lmac: A multi-channel mac protocol for wireless sensor networks. *Ad Hoc Networks*, Elsevier, v. 9, n. 1, p. 73–94, 2011.

- LI, L. et al. Improvement and implementation of rpl routing protocol in wireless sensor networks. IET, 2015.
- LIBELIUM, S. *Wasp mote, Mote Runner*. 2018. Disponível em: <<http://www.libelium.com/products/waspmote-mote-runner-6lowpan/>>.
- MEDWIN, H. Speed of sound in water: A simple equation for realistic parameters. *The Journal of the Acoustical Society of America*, ASA, v. 58, n. 6, p. 1318–1319, 1975.
- MELLINGER, D. K.; NIEUKIRK, S. L.; HEIMLICH, S. L. Ishmael 3. o user guide. 2017.
- MIAO, Q.-x.; XU, L. Dymo routing protocol research and simulation based on ns2. In: IEEE. *Computer Application and System Modeling (ICCASM), 2010 International Conference on*. [S.l.], 2010. v. 14, p. V14–41.
- MILENKOVIĆ, A.; OTTO, C.; JOVANOVIĆ, E. Wireless sensor networks for personal health monitoring: Issues and an implementation. *Computer communications*, Elsevier, v. 29, n. 13-14, p. 2521–2533, 2006.
- MIN, R. et al. Energy-centric enabling technologies for wireless sensor networks. *IEEE wireless communications*, IEEE, v. 9, n. 4, p. 28–39, 2002.
- MISRA, S. et al. u-ocean: An underwater omnidirectional communication environment using acoustic sensor nodes. In: IEEE. *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. [S.l.], 2017. p. 1–6.
- MONTENEGRO, G. et al. *Transmission of IPv6 packets over IEEE 802.15. 4 networks*. 2007.
- MUSALOIU-E, R.; LIANG, C.-J. M.; TERZIS, A. Koala: Ultra-low power data retrieval in wireless sensor networks. In: IEEE. *Information Processing in Sensor Networks, 2008. IPSN'08. International Conference on*. [S.l.], 2008. p. 421–432.
- OCEAN, S. *Hydrophone Recordings*. 2018. Disponível em: <<http://oceansonics.com/sound-gallery/>>.
- OPENLABS. *Raspberry Pi 802.15.4 radio*. 2014. Disponível em: <<http://openlabs.co/OSHW/Raspberry-Pi-802.15.4-radio>>.
- PI, R. *Raspberry Pi Model Zero W*. 2017. Disponível em: <<https://www.raspberrypi.org/products/raspberry-pi-zero-w/>>.
- POLASTRE, J.; HILL, J.; CULLER, D. Versatile low power media access for wireless sensor networks. In: ACM. *Proceedings of the 2nd international conference on Embedded networked sensor systems*. [S.l.], 2004. p. 95–107.
- RAMILLI, M.; PRANDINI, M. A messaging-based system for remote server administration. In: IEEE. *Network and System Security, 2009. NSS'09. Third International Conference on*. [S.l.], 2009. p. 262–269.
- RAMÍREZ, A. N. *Implementació d'Internet de les coses mitjançant Raspberry Pi*. Dissertação (Mestrado) — Universitat Politècnica de Catalunya, 2014.

- RAWAT, P. et al. Wireless sensor networks: a survey on recent developments and potential synergies. *The Journal of supercomputing*, Springer, v. 68, n. 1, p. 1–48, 2014.
- RIBEIRO, F. J. L.; PEDROZA, A. C.; COSTA, L. H. M. Sistema de monitoramento subaquático usando navios de suporte logístico em uma rede de sensores acústicos tolerante a atrasos e interrupções. *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos-SBRC*, v. 2011, p. 133–146, 2011.
- RICHARDSON, L.; RUBY, S. *RESTful web services*. [S.l.]: "O'Reilly Media, Inc.", 2008.
- SANTOS, F. D. dos; RUBINSTEIN, M. G. *Análise da viabilidade de aplicações de monitoramento e controle utilizando redes acústicas submarinas*. 2012.
- SCHMIEGELOW, J. *O planeta azul: uma introdução às ciências marinhas*. [S.l.: s.n.], 2004.
- SCHULTHEISS, P. M.; WAGNER, K. Active and passive localization: Similarities and differences. In: *Underwater Acoustic Data Processing*. [S.l.]: Springer, 1989. p. 215–232.
- SEELY, S.; BY-SHARKEY, K. F. *SOAP: Cross platform web service development using XML*. [S.l.]: Prentice Hall PTR, 2001.
- SHACKLETTE, M. Linux operating system. *Handbook of Computer Networks: LANs, MANs, WANs, the Internet, and Global, Cellular, and Wireless Networks, Volume 2*, Wiley Online Library, p. 78–90, 1995.
- SHELBY, Z.; BORMANN, C. *6LoWPAN: The wireless embedded Internet*. [S.l.]: John Wiley & Sons, 2011. v. 43.
- SHELBY, Z.; HARTKE, K.; BORMANN, C. *The constrained application protocol (CoAP)*. 2014.
- SOUSA-LIMA, R. S. et al. A review and inventory of fixed autonomous recorders for passive acoustic monitoring of marine mammals: 2013 state-of-the-industry. In: *IEEE. Acoustics in Underwater Geosciences Symposium (RIO Acoustics), 2013 IEEE/OES*. [S.l.], 2013. p. 1–9.
- STANDARD, O. *MQTT version 3.1. 1*. 2014.
- STOUFFER, K.; FALCO, J.; SCARFONE, K. Guide to industrial control systems (ics) security. *NIST special publication*, v. 800, n. 82, p. 16–16, 2011.
- TEAM, A. D. Audacity (version 2.1.2)[computer software]. Available: *audacity.sourceforge.net/download*, 2015.
- THUBERT, P. *Objective function zero for the routing protocol for low-power and lossy networks (RPL)*. [S.l.], 2012.
- TILKOV, S.; VINOSKI, S. Node.js: Using javascript to build high-performance network programs. *IEEE Internet Computing*, IEEE, v. 14, n. 6, p. 80–83, 2010.
- TIPLER, P. A. *Física para Cientistas e Engenheiros: Eletricidade e Magnetismo, Óptica*. 6ª edição. [S.l.]: LTC, 2009.

- UPTON, E.; HALFACREE, G. *Raspberry Pi user guide*. [S.l.]: John Wiley & Sons, 2014.
- URICK, R. J. *Principles of underwater sound for engineers*. [S.l.]: Tata McGraw-Hill Education, 1967.
- UWASE, M. et al. Experimental comparison of radio duty cycling protocols for wireless sensor networks. *Sensors*, v. 18247, p. 1, 2017.
- WALD, A.; WOLFOWITZ, J. Optimum character of the sequential probability ratio test. *The Annals of Mathematical Statistics*, JSTOR, p. 326–339, 1948.
- WEPTech, E. *Saker, 6LoWPAN Gateway*. 2018. Disponível em: <<https://www.weptech.de/en/6lowpan/gateway-saker.html>>.
- WICKERT, M. A. Using the arm cortex-m4 and the cmsis-dsp library for teaching real-time dsp. In: IEEE. *Signal Processing and Signal Processing Education Workshop (SP/SPE), 2015 IEEE*. [S.l.], 2015. p. 283–288.
- WINTER, T. *RPL: IPv6 routing protocol for low-power and lossy networks*. 2012.
- YASSEIN, M. B.; SHATNAWI, M. Q. et al. Application layer protocols for the internet of things: A survey. In: IEEE. *Engineering & MIS (ICEMIS), International Conference on*. [S.l.], 2016. p. 1–4.
- YE, W.; HEIDEMANN, J.; ESTRIN, D. An energy-efficient mac protocol for wireless sensor networks. In: IEEE. *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. [S.l.], 2002. v. 3, p. 1567–1576.
- ZHAO, F.; GUIBAS, L. J. *Wireless sensor networks: an information processing approach*. [S.l.]: Morgan Kaufmann, 2004.
- ZIMMER, W. M. *Passive acoustic monitoring of cetaceans*. [S.l.]: Cambridge University Press, 2011.

ANEXO A – DATASHEETS

Features

- Fully integrated 769 – 935MHz transceiver including:
 - Chinese WPAN band from 779 to 787MHz
 - European SRD band from 863 to 870MHz
 - North American ISM band from 902 to 928MHz
 - Japanese band from 915 to 930MHz
- Direct Sequence Spread Spectrum with different modulation and data rates:
 - BPSK with 20 and 40kb/s, compliant to IEEE[®] 802.15.4-2003/2006/2011
 - O-QPSK with 100 and 250kb/s, compliant to IEEE 802.15.4-2006/2011
 - O-QPSK with 250kb/s, compliant to IEEE 802.15.4-2011
 - O-QPSK with 200, 400, 500, and 1000kb/s PSDU data rate
- Flexible combination of frequency bands and data rates
- Industry leading link budget:
 - Receiver sensitivity up to -110dBm
 - Programmable TX output power up to +11dBm
- Ultra-low current consumption:
 - SLEEP = 0.2µA
 - TRX_OFF = 450µA
 - RX_ON = 9.2mA
 - BUSY_TX = 18.0mA at TX output power +5dBm
- Ultra-low supply voltage (1.8V to 3.6V) with internal regulator
- Easy to use interface:
 - Registers, frame buffer, and AES accessible through fast SPI
 - Clock output with prescaler from radio transceiver
- Radio transceiver features:
 - 128-byte FIFO (SRAM) for data buffering
 - Fully integrated, fast settling PLL to support Frequency Hopping
 - Battery monitor
 - Adjustable receiver sensitivity
 - Integrated TX/RX switch, LNA, and PLL loop filter
 - Automatic VCO and filter calibration
 - Integrated 16MHz crystal oscillator
- Special IEEE 802.15.4[™]-2011 hardware support:
 - FCS computation and Clear Channel Assessment
 - RSSI measurement, Energy Detection and Link Quality Indication
- MAC hardware accelerator:
 - Automated acknowledgement and retransmission
 - CSMA-CA and Listen Before Talk (LBT)
 - Automatic address filtering and automated FCS check
- Extended feature set hardware support:
 - AES 128-bit hardware accelerator
 - Antenna Diversity
 - RX/TX indication for external RF front end control
 - True Random Number Generation for security application
- Optimized for low BoM Cost and ease of production:
 - Few external components necessary (crystal, capacitors and antenna)
 - Excellent ESD robustness
 - Industrial temperature range from -40°C to +85°C
- I/O and packages:
 - 32-pin Low-Profile QFN Package 5 x 5 x 0.9mm³
 - RoHS/Fully Green
- Compliant to IEEE 802.15.4-2003/2006/2011
- Compliant to ETSI EN 300 220-1, and FCC 47 CFR Section 15.247

**Low Power,
700/800/900MHz
Transceiver for
ZigBee,
IEEE 802.15.4,
6LoWPAN, and
ISM
Applications**

AT86RF212B

A.2 CODEC DE AUDIO WM8731

WM8731 / WM8731L

Portable Internet Audio CODEC with Headphone Driver and Programmable Sample Rates

DESCRIPTION

The WM8731 or WM8731L (WM8731/L) are low power stereo CODECs with an integrated headphone driver. The WM8731/L is designed specifically for portable MP3 audio and speech players and recorders. The WM8731 is also ideal for MD, CD-RW machines and DAT recorders.

Stereo line and mono microphone level audio inputs are provided, along with a mute function, programmable line level volume control and a bias voltage output suitable for an electret type microphone.

Stereo 24-bit multi-bit sigma delta ADCs and DACs are used with oversampling digital interpolation and decimation filters. Digital audio input word lengths from 16-32 bits and sampling rates from 8kHz to 96kHz are supported.

Stereo audio outputs are buffered for driving headphones from a programmable volume control, line level outputs are also provided along with anti-thump mute and power up/down circuitry.

The device is controlled via a 2 or 3 wire serial interface. The interface provides access to all features including volume controls, mutes, de-emphasis and extensive power management facilities. The device is available in a small 28-pin SSOP package or the smaller 28 lead quad flat leadless package (QFN).

FEATURES

- Highly Efficient Headphone Driver
- Audio Performance
 - ADC SNR 90dB ('A' weighted) at 3.3V, 85dB at 1.8V
 - DAC SNR 100dB ('A' weighted) at 3.3V, 95dB at 1.8V
- Low Power
 - Playback only 22mW, 8mW ('L' Variant)
 - Analogue Pass Through 12mW, 3.5mW ('L' variant)
 - 1.42 – 3.6V Digital Supply Operation
 - 2.7 – 3.6V Analogue Supply Operation
 - 1.8 – 3.6V Analogue Supply Operation ('L' Variant)
- ADC and DAC Sampling Frequency: 8kHz – 96kHz
- Selectable ADC High Pass Filter
- 2 or 3-Wire MPU Serial Control Interface
- Programmable Audio Data Interface Modes
 - I²S, Left, Right Justified or DSP
 - 16/20/24/32 bit Word Lengths
 - Master or Slave Clocking Mode
- Microphone Input and Electret Bias with Side Tone Mixer
- Available in 28-Pin SSOP or 28-pin QFN package

APPLICATIONS

- Portable MP3 Players and Recorders
- CD and Minidisc Recorders
- PDAs / smartphones

BLOCK DIAGRAM

32F429IDISCOVERY

Discovery kit with STM32F429ZI MCU

Data brief

Features

- STM32F429ZIT6 microcontroller featuring 2 Mbytes of Flash memory, 256 Kbytes of RAM in an LQFP144 package
- On-board ST-LINK/V2 on STM32F429I-DISCO (old order code) or ST-LINK/V2-B on STM32F429I-DISC1 (new order code)
- mbed™ -enabled (see <http://mbed.org>) with the new order code only
- USB functions:
 - Debug port
 - Virtual COM port (with new order code only)
 - Mass storage (with new order code only)
- Board power supply: through the USB bus or from an external 3 V or 5 V supply voltage
- 2.4" QVGA TFT LCD
- 64-Mbit SDRAM
- L3GD20, ST-MEMS motion sensor 3-axis digital output gyroscope
- Six LEDs:
 - LD1 (red/green) for USB communication
 - LD2 (red) for 3.3 V power-on
 - Two user LEDs: LD3 (green), LD4 (red)
 - Two USB OTG LEDs: LD5 (green) VBUS and LD6 (red) OC (over-current)
- Two push-buttons (user and reset)
- USB OTG with micro-AB connector
- Extension header for LQFP144 I/Os for a quick connection to the prototyping board and an easy probing
- Comprehensive free software including a variety of examples, part of STM32CubeF4 package or STSW-STM32138, for using legacy standard libraries

1. Picture is not contractual.

Description

The 32F429IDISCOVERY kit leverages the capabilities of the STM32F429 high-performance microcontrollers, to allow users to easily develop rich applications with advanced Graphic User interfaces.

With the latest board enhancement, the new STM32F429I-DISC1 order code has replaced the old STM32F429I-DISCO order code.

A.4 BANCO DE BATERIAS PN-968

PINENG PN-968 POWER BANK

Pineng PN-968 10,000mAh PowerBank

PINENG®
Universal USB Backup Power
PN-968 10,000mAh Power Bank

PINENG®
Universal USB Backup Power
PN-968 10,000mAh Power Bank

- Pineng PN-968 10,000mAh PowerBank
- Dual USB Output 2.1A/1A
- LCD Display for battery level
- Built-In Battery Protection IC
- Prevent Over-Charging, Over-Discharging, Short-circuit
- Intelligent voltage recognition chip
- 10,000mAh Battery Capacity
- 7 Days Satisfactory Guarantee
- 90 Days Onsite Warranty
- 12 Months Product Warranty
- Enjoy FREE SHIPPING when you spend more than RM100

Rating: Not Rated Yet

Price

Base price with tax: RM 108.10

Price with discount: RM 53.49

Sales price: RM 56.70

Discount: RM -51.40

Tax amount: RM 3.21

[Ask a question about this product](#)

Description

ANEXO B – TRABALHOS GERADOS

São relacionados os trabalhos gerados a partir deste trabalho de mestrado:

- Durante, G.; Becari, W.; Peres, H.E.M.; Ramirez-Fernandez, F.J. - Rede de Sensores Sem Fio sub 1 GHz para Caracterização de Paisagens Acústicas Marinhas. In: 17º SBMO - Simpósio Brasileiro de Micro-ondas e Optoeletrônica (MOMAG 2016) Porto Alegre, 2016.
- Durante, G.; Becari, W.; Oliveira, A.M.; Peres, H.E.M.; Galeazzo, E.; Ramirez-Fernandez, F.J. - Desenvolvimento de Sistema Embarcado Autônomo para Caracterização de Paisagem Acústica Marinha. In: VII CONGRESSO BRASILEIRO DE OCEANOGRAFIA (CBO 2016), Salvador, 2016, Anais, p. 584.
- Durante, G.; Beccaro, W.; Peres, H.E.M.- IoT Protocols Comparison for Wireless Sensors Network Applied to Marine Environment Acoustic Monitoring. IEEE Latin America (submetido em dez. 2017).

A reprodução dos trabalhos listados pode ser vista nos Anexos B.1, B.2 e B.3.

B.1 TRABALHO APRESENTADO - CBO 2016

DESENVOLVIMENTO DE SISTEMA EMBARCADO AUTÔNOMO PARA CARACTERIZAÇÃO DA PAISAGEM ACÚSTICA MARINHA

Durante, G.; Becari, W.; Oliveira, A.M.; Peres, H.E.M.; Galeazzo, E.; Ramirez-Fernandez, F.J.

Escola Politécnica da Universidade de São Paulo

Oceanos são ecossistemas repletos de sons naturais (biológicos ou ambientais), como também de sons não-naturais (decorrentes de atividades humanas). Esses sons são indicadores de condições adequadas para o crescimento, alimentação e orientação espacial de diversas espécies marinhas. A biodiversidade marinha pode ser monitorada através do emprego do mapeamento da diversidade acústica. Contudo, apesar de sua importância para elucidar processos ecológicos marinhos e a ação do homem no meio ambiente, essas paisagens ainda não foram caracterizadas para a maioria dos ambientes costeiros e estuarinos, devido à lacuna de dispositivos eletrônicos dedicados, que permitam monitorar o ruído ambiental durante um período de tempo prolongado. Dessa forma, há a necessidade de desenvolver sistemas embarcados, capazes de caracterizar a paisagem acústica marinha, com boa autonomia energética e possibilidade de registro e transmissão remota dos dados acústicos coletados. Este trabalho apresenta o desenvolvimento de um sistema embarcado microcontrolado, de baixo custo, para aplicações em meios aquáticos. O sistema é composto por uma unidade autônoma e uma unidade fixa que se comunicam através de um enlace de rádio frequência (RF). A unidade de monitoramento é composta por um cilindro estanque, um hidrofone externo e uma eletrônica embarcada para condicionamento, digitalização, armazenamento e transmissão sem fio dos sinais acústicos amostrados. Para a captação do sinal acústico foi utilizado um hidrofone piezoelétrico PZT-5 com sensibilidade de $-198 \text{ dB re } 1 \text{ V}/\mu\text{Pa}$ e resposta em frequência até 30 kHz. A aquisição dos dados do hidrofone foi realizada utilizando-se um CODEC (*Coder/Decoder*) WM8731 de áudio estéreo de 24 bits, com frequência de amostragem 44.000 amostras/segundos. Esse circuito integrado é conectado a um microcontrolador ARM Cortex M4 (STM32F429) através de uma interface I2S (*Integrated Interchip Sound*). Os dados são armazenados em uma memória *flash* externa. O enlace RF é composto por módulos sub 1 GHz AT86RF212B que possuem baixo consumo de energia, sensibilidade de -110 dBm e potência de transmissão configurável de até $+11 \text{ dBm}$. Além disso, os módulos de RF suportam diversos protocolos de rede, em especial o IEEE 802.15.4. A unidade de monitoramento é alimentada por um *powerbank* com capacidade de 10.000 mAh. A unidade fixa é conectada diretamente a um computador para controlar parâmetros da unidade autônoma e permitir o monitoramento das medidas. A caracterização do sistema em ambiente aquático foi realizada na raia olímpica da USP através do nível de sinal RSSI (*Received Signal Strength Indication*) do receptor, e da quantidade de perda de pacotes. O alcance entre nós do sistema foi de 100 metros. A taxa de erros de pacotes de dados ficou abaixo de 9,3%. A autonomia do nó autônomo foi estimada em 125 horas sem interrupções. A infraestrutura desenvolvida possibilitou a interconexão entre os nós em tempo real através de uma topologia *ad-hoc*. Testes de captação de sons foram realizados na base oceanográfica da USP em Cananéia (litoral sul de São Paulo), o que permitiu verificar a funcionalidade do sensor piezoelétrico especificado. Dessa forma, o sistema proposto apresenta boa autonomia e confiabilidade para a captação e transmissão de sinais para estudos de bioacústica submarina.

Palavras-Chave: PAISAGEM ACÚSTICA MARINHA; REDE DE SENSORES; IMPACTOS AMBIENTAIS.

Rede de Sensores Sem Fio sub 1 GHz para Caracterização de Paisagens Acústicas Marinhas

Gabriel Durante, Wesley Becari, Henrique E. M. Peres, Francisco Javier Ramirez-Fernandez

Departamento de Engenharia de Sistemas Eletrônicos

Escola Politécnica da Universidade de São Paulo

São Paulo, Brasil

gabriel.durante@usp.br, wesley@lme.usp.br, hperes@lme.usp.br, jramirez@lme.usp.br

Resumo— O crescente interesse pela caracterização de paisagens acústicas marinhas esbarra nos poucos dispositivos eletrônicos que permitem armazenar, processar e difundir os parâmetros marinhos. Os equipamentos atuais, que possibilitam obter dados referentes à vida marinha e as influências humanas sobre ela, possuem alto custo e alta complexidade. Além disso, muitos equipamentos funcionam autonomamente, sem que haja comunicação com outros dispositivos, o que impossibilita a monitorização em tempo real. Nesse sentido o desenvolvimento de uma rede de sensores capaz de obter dados de hidrofones e realizar a comunicação através de um enlace por rádio frequência é fundamental para os estudos envolvendo paisagem acústica marinha. Esse trabalho apresenta a análise do desempenho de uma rede de sensores sub 1 GHz em um ambiente aquático, demonstrando a potencialidade do uso dessa tecnologia. O sistema testado é capaz de fazer a aquisição dos dados de sensores acústicos de forma ininterrupta por até 125 horas com um bom balanceamento entre consumo e distância de enlace.

Palavras-chave— Paisagem Acústica Marinha, Rede de Sensores sem Fio, Protocolo 6LoWPan.

I. INTRODUÇÃO

Oceanos são ambientes repletos de sons naturais, biológicos ou ambientais, e não-naturais, decorrentes de atividades humanas. Esses sons são indicadores de condições adequadas para o crescimento, alimentação e orientação espacial de diversas espécies marinhas [1]. Existem evidências crescentes de que sons muito intensos associados ao aumento das atividades antropogênicas prejudiquem a comunicação e o reconhecimento de ambientes favoráveis à manutenção do ciclo de vida dos organismos [2].

Sabe-se que diferentes ambientes marinhos e animais possuem assinaturas sonoras específicas. Essas diferentes assinaturas permitem definir o conceito de paisagem acústica marinha. Além disso, existem estudos que indicam a possibilidade de monitorar e mensurar a biodiversidade marinha em diferentes escalas temporais e espaciais, através do emprego de índices de diversidade acústica [3, 4].

Contudo, apesar de sua potencial importância para elucidar processos ecológicos marinhos, essas paisagens ainda não puderam ser plenamente caracterizadas para a maioria dos ambientes costeiros e estuarinos, devido a lacuna de

dispositivos eletrônicos de baixo custo, que permitam acompanhar durante um tempo prolongado o comportamento de uma região [5].

Dessa forma, verificar o comportamento de uma rede de sensores sem fio capaz em um ambiente aquático é fundamental para o desenvolvimento de sistemas eletrônicos de baixo custo capazes de caracterizar a paisagem acústica submarina. Esse trabalho propõe uma rede de sensores sem fio sub 1 GHz para aplicações em meios aquáticos. Na seção II são apresentados os conceitos do protocolo de comunicação 6LoWPan e do sistema operacional em tempo real Contiki. A seção III descreve os elementos constitutivos da rede de sensores desenvolvida. A seção IV apresenta a caracterização do sistema com análise do desempenho da rede em um cenário aquático. Por fim, é apresentada a conclusão e perspectivas na seção V.

II. ASPECTOS TEÓRICOS

A. Protocolo de Comunicação 6LoWPan

O 6LoWPan extrapola o uso do protocolo IPv6 em redes sem fio com baixo consumo de energia, como é o caso de sistemas embarcados com processamento muitas vezes limitado [6]. Com essa adaptação do IPv6, tornou-se possível o conceito de “Internet das Coisas”, que na prática pode ser definido como uma rede IP de sistemas embarcados que se interconectam através de serviços de controle e monitoramento.

Em contraste ao IPv6 tradicional, o 6LoWPan possui diversas melhorias que buscam o aproveitamento da potência de interfaces de rádio, redução do tamanho da palavra transmitida ou recebida e aproveitamento de banda, suportando dispositivos interconectados em topologias estrela e Mesh [7]. A Fig. 1 apresenta as diferentes arquiteturas de rede 6LoWPan.

Nessas redes a comunicação é feita utilizando-se o padrão IEEE 802.15.4, que define o transceptor de rádio para operar em frequências de 868 MHz, 915 MHz ou 2.4 GHz. Para este trabalho em específico, foi criada uma rede *ad-hoc* operando sem infraestrutura e que pode ser conectada à outra rede IP através de um *edge router*.

Fig. 1. Comparação entre arquiteturas de rede de sensores sem fio 6LoWPan[7].

B. Sistema Operacional em Tempo Real Contiki

O sistema operacional em tempo real *Contiki* possui um *kernel* baseado em interrupções e gerenciadores de eventos, disparando-os de forma assíncrona e síncrona. Além disso, o *kernel* suporta mecanismos de varredura que podem ser configurados entre os processos assíncronos [7].

Para controlar as informações em uma rede 6LoWPan, o *Contiki* utiliza o RPL (*Routing protocol for low-power and lossy networks*) para formar um gráfico de roteamento desde o nó principal até seus vizinhos mais distantes, trocando mensagens em forma de *broadcast*. Além disso, sistemas operacionais possibilitam economia de banda e energia nos nós; uso de módulos MAC (*Media Access Control*) específicos que evitam colisão de pacotes e diminuem o tempo de recepção e transmissão dos nós.

III. MATERIAIS E MÉTODOS

A. Sistemas embarcados microprocessados

A rede de sensores é composta por unidades autônomas distribuídas e uma unidade fixa. Cada unidade de monitoramento é composta por um cilindro estanque, um hidrofone externo e uma eletrônica embarcada para condicionamento, digitalização, armazenamento e transmissão sem fio do sinal acústico.

A Fig. 2 apresenta um diagrama simplificado dos componentes eletrônicos usados para realizar a aquisição dos sinais provenientes do meio marinho.

Para a captação do sinal acústico foi utilizado um hidrofone piezoelétrico PZT-5 com formato cilíndrico de dimensões diâmetro de 2,5 mm e comprimento de 32 mm, sensibilidade de -198 dB re 1V/ μ Pa e resposta em frequência plana (± 2 dB) até a frequência de 30 kHz.

A aquisição dos dados do hidrofone foi realizada utilizando um CODEC (*Coder/Decoder*) WM8731 de áudio estéreo de 24 bits, com frequência de amostragem

44.000 amostras/segundos, e relação sinal ruído de 90 dB. Esse circuito integrado é conectado a um microcontrolador ARM Cortex M4 através de uma interface I²C (*Inter-Integrated Circuit*). Os dados são armazenados em uma memória *flash* externa via USB (*Universal Serial Bus*).

O módulo de gerenciamento é composto pelo kit de desenvolvimento STM32F429Disco e um módulo sub 1 GHz de radiofrequência (RF) AT86RF212B *ZigBit Xplained PRO Extension*. Esse transceptor RF possui baixo consumo de energia, sensibilidade de -110 dBm e potência de transmissão configurável de até +11 dBm. Além disso, possibilita taxas de transmissão até 250 kbps, modulação QPSK (*Quadrature Phase Shift Keying modulation*) e suporta diversos protocolos de rede, em especial o IEEE 802.15.4.

Um *powerbank*, modelo PN-968, com a capacidade de 10000 mAh, foi utilizado para alimentar o circuito eletrônico. A Fig. 3 apresenta a unidade autônoma com os circuitos eletrônicos e o encapsulamento à prova d'água.

Fig. 2. Diagrama em blocos de cada unidade autônoma de monitoramento acústico aquático.

Fig. 3. Unidade autônoma para coleta de dados: detalhe da eletrônica de embarcada e do encapsulamento à prova d'água.

Nas unidades autônomas foi embarcado o sistema operacional *Contiki*. A unidade fixa é constituída por um Atmel ATmega256RFR2 *Xplained Pro Evaluation Kit* conectado ao módulo AT86RF212B *ZigBit Xplained PRO Extension*. Essa unidade fixa é responsável por gerenciar uma topologia *ad-hoc* com o objetivo de criar uma rede de sensores para monitoramento acústico passivo.

As antenas cerâmicas utilizadas nos módulos AT86RF212B possuem baixo custo e boa sensibilidade, pois são conectadas diretamente ao circuito integrado da placa de circuito impresso. Essas antenas são omnidirecionais, característica desejada para a rede de sensores sem fio, contudo não possuem ganho.

B. Instrumentação de Caracterização da Rede de Sensores: Cenários e Métricas de Avaliação

Os testes foram realizados na Raia Olímpica da USP, com uma unidade fixa, fora da água, e uma unidade autônoma, na água, como pode ser verificado na Fig. 4.

Fig. 4. Fig 4. Testes conduzidos na Raia Olímpica da USP.

A caracterização do enlace de RF foi realizada através do envio de mensagens UDP (*User Datagram Protocol*) *multicast* entre dois rádios na frequência de 902 MHz com intervalos de 5 segundos.

Os testes de caracterização do enlace foram divididos em duas fases. Na primeira fase, foram analisados o nível de sinal RSSI (*Received Signal Strength Indication*) do receptor mediante o aumento da distância entre o nó autônomo e a estação base. Além disso, foi avaliada a quantidade de perda de pacotes quando o nível de sinal estava menor que a sensibilidade do receptor. As métricas utilizadas neste teste foram: a quantidade de pacotes enviados em comparação com a quantidade de pacotes perdidos e os erros de CRC (*Cyclic Redundancy Check*) de pacotes recebidos em determinado intervalo de tempo. Com o nível de potência recebida foi possível determinar a distância máxima entre os dois transceptores.

Em um segundo momento, foi caracterizado o consumo de potência da unidade autônoma. Nesta fase foi observada a potência usada pelo transceptor de rádio no pico de transmissão e a partir deste parâmetro foi estimada a autonomia do equipamento, em termos de horas de funcionamento.

Também foram coletados e analisados sons aquáticos na base de pesquisa da Universidade de São Paulo (USP), situada na cidade de Cananéia, litoral sul de São Paulo. A título de exemplo, são apresentados os sons acústicos amostrados do motor da embarcação “Albacora” usando uma interface de áudio Behringer UCA222.

IV. RESULTADOS E DISCUSSÕES

A. Caracterização do Desempenho da Unidade Autônoma

A Fig. 5 apresenta a intensidade do sinal de recepção conforme o aumento da distância entre a unidade autônoma e estação base. O transmissor foi configurado com ganho de 10 dBm. Nota-se que o receptor foi incapaz de receber pacotes em uma distância acima de 100 m. O limite de sensibilidade medida foi de -95 dBm, embora o receptor seja capaz de medir, sinais com potência até -110 dBm, segundo características informadas pelo fabricante.

Fig. 5. Nível de sinal RSSI em função da distância.

A análise de perda de pacotes e erros foi feita de forma conjunta. Com um intervalo fixo de espera de pacotes na recepção, foi possível verificar as perdas de pacotes e erros de CRC (pacotes recebidos e descartados devido à má integridade). A taxa de perda de pacotes ficou abaixo de 9.3% para distâncias de até 100 metros. Observa-se que até a distância máxima testada, para uma quantidade maior de pacotes enviados, o erro tende a se estabilizar. A Fig. 6 demonstra os pacotes perdidos e a quantidade de erros CRC mediante o número de pacotes recebidos na unidade fixa.

Nos picos de corrente do transceptor de RF foi possível medir um consumo de 81,2 mA com polarização de 3,3 V,

totalizando, aproximadamente, 125 horas de funcionamento sem interrupções. Dessa forma, o sistema se mostra viável para o uso em ambientes marítimos e expedições de caracterização de paisagem acústica.

Fig. 6. Taxa de erro durante ensaio.

B. Aquisição dos Dados

Para a caracterização e testes do hidrofone PZT-5, foram coletadas mais de 20 horas de dados, com taxa de amostragem de 44.000 amostras/s, durante quatro dias. Selecionando-se trechos, pode-se apresentar como resultado a coleta do sinal acústico gerado por um motor de embarcação a 20 metros do hidrofone, conforme Fig. 7.

Do sinal captado é possível analisar pequenos trechos de (6 s) através do seu espectrograma. Essa ferramenta possibilita a inspeção visual do perfil da densidade de potência ao longo do tempo [8, 9]. A análise do espectrograma foi realizada através da FFT (*Fast Fourier Transform*) de janelas de curto tempo (da ordem de 12 ms), com 512 amostras, 50% de sobreposição e janelamento *Hamming*.

Fig. 7. Amplitude do sinal captado do motor de embarcação e espectrograma.

No espectrograma da Fig. 7, nota-se elevada potência sonora de frequências até 3 kHz e em torno de 5,5 kHz durante todo o intervalo, uma vez que o motor permaneceu ligado. Ou

seja, o sistema desenvolvido permite realizar medidas da paisagem acústica, incluindo sons de animais, embarcações, plataformas, ou outros ruídos antropogênicos.

V. CONCLUSÃO

Foi desenvolvida uma rede de sensores para caracterização da paisagem acústica marinha através de um enlace sub 1 GHz. O alcance entre nós do sistema, sem uso de amplificadores de potência externos, foi de 100 metros usando +10 dBm de potência no transmissor. A taxa de erros de pacotes ficou abaixo de 9,3% com a tendência a diminuir mediante ajuste da distância do enlace. A autonomia dos nós sensores foi estimada em 125 horas sem interrupções. A infraestrutura possibilitou a interconexão entre os nós em tempo real através de uma topologia *ad-hoc*. Foram coletados dados aquáticos na base de pesquisa da USP em Cananéia, que permitiram demonstrar a potencialidade do sistema para detecção, transmissão e análise de ruídos antrópicos (tráfego naval, pesca, nível de ruído ambiente) e ruídos naturais (prospecções sísmicas, cetáceos, peixes, crustáceos e outros animais marinhos). Dessa forma, o sistema proposto se mostra como interessante ferramenta para estudos de bioacústica submarina.

AGRADECIMENTOS

Os autores agradecem o Instituto Oceanográfico da USP por disponibilizar infraestrutura logística junto à base de pesquisa “Dr. João de Paiva Carvalho”, situada na cidade de Cananéia em São Paulo.

REFERÊNCIAS

- [1] E. R. Vivas, B. L. López. “Construction, calibration, and field test of a home-made, low-cost hydrophone system for cetacean acoustic research,” *Proceedings of Meetings on Acoustics*, 2012.
- [2] M. O. Lammers, R. E. Brainard, W. W. L. Au, T. A. Mooney, K. B. Wong. “An ecological acoustic recorder (EAR) for long-term monitoring of biological and anthropogenic sounds on coral reefs and other marine habitats,” *The Journal of the Acoustical Society of America*, 2008.
- [3] R. S. Sousa-Lima, T. F. Norris, J. N. Oswald, D. P. Fernandes, “A Review and Inventory of Fixed Autonomous Recorders for Passive Acoustic Monitoring of Marine Mammals,” *Aquat Mamm*, 2013.
- [4] S. M. Wiggins, J. A. Hildebrand. “High-frequency Acoustic Recording Package (HARP) for broadband, long-term marine mammal monitoring,” *Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies*, 2007.
- [5] J. J. Martinez, J. R. Myers, T. J. Carlson, Z. D. Deng, J. S. Rohrer, K. A. Caviggia. “Design and Implementation of an Underwater Sound Recording Device,” *Sensors*, 2011.
- [6] Z. Shelby, C. Bormann. “6LoWPan The Wireless Embedded Internet,” Wiley, 2010.
- [7] A. Dunkels, B. Gronvall, T. Voigt. “Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors. Swedish Institute of Computer Science,” Sweden, 2004.
- [8] T. D. Rossing “Springer handbook of acoustics,” Springer: New York, 2007.
- [9] R. K. Shastri, Y. V. Joshi and A. Das, “Time Frequency Analysis of underwater ambient noise in tropical littoral waters,” *Oceans - San Diego*, 2013, San Diego, CA, pp. 1-4, 2013.

IoT Protocols Comparison for Wireless Sensors Network Applied to Marine Environment Acoustic Monitoring

G. Durante, W. Beccaro, H. E. M. Peres

Abstract— There is a growing interest to analyze acoustic marine environment. Underwater sounds can indicate the natural marine activity, as well as the influence of human activities in the aquatic ecosystems. Several kinds of electronic devices can be used to monitor the marine sound, but normally these devices work autonomously, without communication with other equipment, which makes difficult the real-time monitoring. In addition, the usual equipment is normally expensive and highly complex. Thus, for the design of wireless sensor network (WSN) applied to marine monitoring, it is important to develop systems with good energy autonomy and efficient data processing and communication. However, it is not well established the best WSN communication protocol for applications in hostile environments. Some protocols used for Internet of Things (IoT) applications have emerged recently as alternative for efficient communication between nodes and gateways. This work presents a comparison between two IoT application protocols: CoAP and MQTT-SN. We have verified that the MQTT-SN protocol has some advantages, such as: its latency is 30% lower compared with CoAP, the power consumption is 10% lower than CoAP, and the traffic flow is 2.15 times larger than CoAP, for an architecture with 40 wireless nodes.

Keywords— Internet of Things, Underwater technology, Machine-to-machine, Acoustic sensors, Wireless sensor networks.

I. INTRODUÇÃO

OCEANOS são ambientes repletos de sons naturais, biológicos ou ambientais, e não-naturais, decorrentes de atividades humanas [1], [2]. Diferentes animais e ambientes marinhos apresentam, assim, “assinaturas” sonoras específicas que permitem definir o conceito de paisagem acústica marinha [3], [4]. Esses sons são indicadores de condições adequadas para o crescimento, alimentação e orientação espacial de diversas espécies marinhas [1]. Existem evidências crescentes de que sons muito intensos associados ao aumento das atividades antropogênicas prejudiquem a comunicação e o reconhecimento de ambientes favoráveis à manutenção do ciclo de vida dos organismos [2].

Apesar de sua importância para elucidar processos ecológicos marinhos, essas paisagens ainda não puderam ser plenamente caracterizadas para a maioria dos ambientes costeiros e estuarinos devido à lacuna de sistemas eletrônicos adequados às peculiaridades do ambiente marinho.

Para o monitoramento do som nesse ambiente, alguns requisitos são essenciais. O sistema necessita de módulos sensores distribuídos e que não afetem o tráfego de embarcações e de animais. Assim, é necessária a implementação de uma rede de sensores sem fio (RSSF). Além disso, os módulos devem possuir boa autonomia e eficiência energética para garantir sua operação ao longo de dias ou até meses [4], [5]. Em específico, os protocolos de comunicação de dados devem ser robustos para viabilizar a inclusão do contexto de Internet das Coisas (IoT) [6-9] permitindo a intercomunicação entre os dispositivos do sistema não só para o envio em tempo real dos dados sonoros coletados, mas também para possibilitar análises dos dados coletados ou reconfigurações nos dispositivos do sistema. Para esse caso, convém utilizar protocolos que transmitam a informação coletada sem que haja o controle de erros, ou seja, protocolos não orientados para a conexão.

Existem diversos protocolos para aplicações em IoT que possam atender às necessidades do desenvolvimento de uma rede de sensores em ambiente marinho, destacando-se o *Message Queue Telemetry Transport for Sensor Networks* (MQTT-SN) [10] e o *Constrained Application Protocol* (CoAP) [11]-[13]. Embora alguns trabalhos tenham discutido as diferenças entre os protocolos MQTT e CoAP [14], [15] e, sucintamente, entre o MQTT-SN e CoAP [16], não há uma análise dos principais parâmetros de comunicação e de outras características aplicadas a nós sensores. Para isso, é necessário realizar uma comparação entre seus desempenhos em termos de latência, tempo de tráfego na rede, quantidade de dados transmitidos, consumo de energia dos nós, entre outras características.

Dessa forma, é imprescindível verificar o comportamento de uma rede de sensores sem fio e os meios de transporte de dados nela presentes, garantindo a entrega das características monitoradas. Este trabalho apresenta uma análise comparativa entre os protocolos MQTT-SN e CoAP utilizando o simulador COOJA, para um cenário de monitoração de paisagens acústicas marinhas. Para tanto, foi utilizada a implementação nativa do protocolo CoAP, mas foi necessário implementar o

G. Durante, Escola Politécnica da Universidade de São Paulo (USP), São Paulo, Brasil, gbdurante@gmail.com.

W. Beccaro, Escola Politécnica da Universidade de São Paulo (USP), São Paulo, Brasil, wesley@lme.usp.br.

H. E. M. Peres, Escola Politécnica da Universidade de São Paulo (USP), São Paulo, Brasil, hperes@lme.usp.br.

(Corresponding author: H. E. M. Peres)

protocolo MQTT-SN e suas camadas no simulador. Dado que no sistema operacional de rede de sensores Contiki-OS, amplamente utilizado em nós sensores, não há implementação nativa do protocolo MQTT-SN.

II. ARQUITETURA IoT E PROTOCOLOS DE COMUNICAÇÃO

A. Arquitetura IoT

Existem, na atualidade, diferentes modelos de arquitetura de IoT [17]-[20]. Desde os modelos mais simplificados, com três camadas: camada de aplicação (*Application Layer*), camada de rede (*Network Layer*), e camada de percepção (*Perception Layer*), até modelos com cinco camadas [20], [21]: camada de negociação (*Business Layer*), camada de aplicação, camada de gerenciamento de processos (*Service Management*), camada de abstração de objetos (*Object Abstraction*) e camada de objetos (*Objects*), comparados na Fig.1

A camada *Perception Layer* ou *Objects* inclui os sensores e os atuadores. A parte física dos transdutores é responsável por coletar e processar os dados. Essa camada também contempla protocolos de rede de sensores para garantir a padronização através do conceito de transdutores *plug and play* [20], [21].

A camada *Network Layer* ou *Object Abstraction* é responsável por transferir os dados utilizando canais de comunicação seguros. Dentre as tecnologias dessa camada estão: 3G, *Group Special Mobile* (GSM), *Universal Mobile Telecommunications System* (UMTS), WiFi, Bluetooth, *Radio-Frequency Identification* (RFID), etc.

A camada *Application Layer* na arquitetura de 3 camadas abstrai três blocos na arquitetura de 5 camadas. A camada de *Service Management* é responsável pelo serviço de requisição de dados baseado em endereçamentos. Nessa camada, os pacotes são direcionados de acordo com a demanda de serviços dos nós IoT ou do servidor. Na arquitetura de 5 camadas, a camada de aplicação é responsável por prover serviços de acesso aos dados ao usuário final.

Por último, a camada de *Business Layer* gerencia todas as atividades e serviços do sistema IoT, dentre ela a construção de modelos, gráficos, fluxos de dados, banco de dados, entre outros. Essa camada facilita a construção de sistemas de tomada de decisão baseados na análise e processamento do *Big Data*, uma das principais fronteiras de pesquisa de IoT.

Figura. 1. Comparação entre arquiteturas IoT de 3 e 5 camadas.

Existem diferentes protocolos que são utilizados na camada *Application Layer* da arquitetura de IoT. Dentre os principais protocolos estão: *Data Distribution Service* (DDS), CoAP, *Advanced Message Queuing Protocol* (AMQP), MQTT, MQTT-SN, *Extensible Messaging and Presence Protocol* (XMPP) e *Hypertext Transfer Protocol* (HTTP). Como dito anteriormente, os protocolos CoAP e MQTT-SN têm se destacado por suas características, descritas a seguir:

A. Constrained Application Protocol (CoAP)

O CoAP define um protocolo de transmissão *web* baseado no conceito de *REpresentational State Transfer* (REST), que facilita a transferência de dados entre clientes e servidores através de HTTP. Esse protocolo foi desenvolvido pelo *Internet Engineering Task Force* (IETF) [12], [13] para aplicações M2M, permitindo a transferência de dados para uso em nós sensores e redes de baixo recurso.

O modelo de protocolo e troca de mensagens foi idealizado para aperfeiçoar o uso de HTTP em redes de sensores sem fio. A organização é feita em duas camadas: camada de transição (*messaging sub-layer*), responsável por gerenciar a troca de mensagens entre os dispositivos e camada de requisição/resposta (*request/response sub-layer*), responsável pelo envio, recebimento e gerenciamento de recursos. As duas camadas permitem que o CoAP forneça mecanismos de confiabilidade e controle de fluxo de dados de forma assíncrona [20]-[24]. O protocolo também possui quatro tipos distintos de mensagens: confirmáveis (*confirmable*), não confirmáveis (*non-confirmable*), reinicializar (*reset*), e reconhecimento (*acknowledgement*).

A Fig. 2 apresenta a arquitetura de uma rede com protocolo CoAP, onde os nós sensores e dispositivos clientes concentram os dados no servidor CoAP e, posteriormente, enviam seus dados para o CoAP *proxy*, mediante requisições.

B. Message Queuing Telemetry Transport for Sensor Networks (MQTT-SN)

O MQTT-SN estende o uso do MQTT para rede de sensores sem fio usando o *User Datagram Protocol* (UDP) ao invés do *Transmission Control Protocol* (TCP), tendo em vista que nessas redes é comum a falha do enlace devido à mobilidade dos nós sensores ou dos dispositivos clientes. Esta abordagem se torna mais eficaz no transporte de dados entre sensores e concentradores de dados.

Figura. 2. Ambiente do protocolo CoAP.

O principal conceito de funcionamento do MQTT e do MQTT-SN é a comunicação centralizada de dados onde os recursos são transferidos via mensagens publicação/assinatura (*publish/subscribe*), possibilitando maior simplicidade, flexibilidade e menor banda de transmissão de dados.

A Fig. 3 apresenta a arquitetura do MQTT-SN onde os nós com baixo recurso rodam instâncias clientes e seu dados são publicados no MQTT *broker*.

A comunicação pode ser feita de forma transparente através do uso adicional de um nó MQTT-SN *gateway*. Nessa condição, apenas algumas alterações de sintaxe são feitas e a conexão entre o MQTT-SN *gateway* e o *broker* é realizada através do protocolo TCP. Outra forma de realizar a comunicação é a forma agregada, quando apenas uma conexão é feita com o MQTT *broker* [11].

As mensagens que são transportadas pelo protocolo MQTT-SN possuem um nível de qualidade de serviço associada (*QoS*) e uma identificação (*topic id*). O protocolo define três níveis de *QoS*: o nível 0, em que o remetente envia a mensagem apenas uma vez e nenhuma nova tentativa é executada, e os níveis 1 e 2, nos quais o protocolo garante que nenhuma mensagem de publicação será perdida, envolvendo, assim, um processo de reconhecimento de um ou dois passos, respectivamente.

Algumas diferenças estão presentes entre a versão do MQTT em comparação com o MQTT-SN, dedicada à rede de sensores e dispositivos com baixo consumo de energia e pequena banda de transmissão. A principal diferença é o uso de UDP, o que retira qualquer necessidade de recebimento de mensagens, típicas no TCP. Além disso, podem ser destacadas a quebra da mensagem CONNECT em mais mensagens, o envio do *topic id* ao invés de *topic name* nas mensagens PUBLISH, o procedimento de DISCOVERY para os nós que não possuem configuração prévia de *gateway*, e, por último, um procedimento para manter os nós em estado de dormência (*sleep mode*).

Destaca-se que o MQTT-SN suporta múltiplos *gateways*, função comumente usada em redes de sensores sem fio devido à possíveis falhas de *link*. Diferentemente do CoAP, no qual a comunicação é feita de forma ponto a ponto (“um para um”), a comunicação no MQTT-SN é feita em forma *multicast*. A Tabela 1 apresenta outras comparações entre esses protocolos enfatizando os parâmetros: padrão de envio de mensagens, protocolo de transporte, confiabilidade, tamanho do cabeçalho das mensagens e vantagens.

Figura 3. Funcionamento do protocolo MQTT-SN.

TABELA I
ANÁLISE COMPARATIVA DOS PROTOCOLOS
CoAP E MQTT-SN [21]- [22], [26]

PARÂMETROS DO PROTOCOLO	CoAP	MQTT-SN
Padrão de envio de mensagens	Similar ao HTTP utilizando método Request/response (client-server)	Publish/subscribe utilizando message broker
Protocolo de Transporte	UDP	UDP
Confiabilidade	Possui recurso para verificar o transporte de mensagens utilizando mensagens do tipo CON.	Possui recurso para verificar o transporte de mensagens utilizando o recurso <i>QoS</i> 1 e 2.
Cabeçalho das mensagens	4 bytes, sendo os dois primeiros com indicação da versão do protocolo e os últimos ligados ao tipo de mensagem CON, NON, ACK ou RST.	2 bytes, apresentando as informações do tamanho e o tipo da mensagem.
Vantagens	Protocolo rápido e eficiente devido ao uso do UDP. Interoperável com o protocolo HTTP.	A arquitetura do tipo <i>broker</i> facilita o gerenciamento da rede.

III. MATERIAIS E MÉTODOS

A. Arquitetura de Rede Proposta

A rede de sensores proposta usa a camada PHY e MAC do IEEE802.15.4, o sistema operacional adotado foi o CONTIKI-OS [24] e a ferramenta de simulação foi o *software* COOJA [25] que acelera o desenvolvimento e testes de aplicativos, permitindo a simulação de um único nó sensor sem fio ou uma rede completa desses nós em um computador comum sem o *hardware* do nó do sensor. A execução da aplicação, comunicação de rede e periféricos podem ser simulados em tempo real para uma grande quantidade de nós. Entre os tipos de simulação estão presentes quatro tipos de modelos de propagação de rádio: *Unit Disk Graph Medium* (UDGM) *Distance Loss* onde a interferência é considerada porém os pacotes são transmitidos e recebidos com probabilidade de sucesso, UDGM *Constant Loss*, onde o alcance de transmissão é determinado por uma circunferência ideal e o sinal alcança todo o perímetro, *Directed Graph Radio Medium* (DGRM) onde a razão de sucesso é determinada de forma assimétrica em cada link por meio de parametrização dos delays de propagação e *Multi-path Ray-tracer Medium* (MRM), no qual a potência de recepção é determinada usando a fórmula de Friis e os obstáculos impõem atenuações nos sinais dos sensores [26].

O usuário pode interagir com os nós usando uma interface gráfica de usuário (GUI), que permite a inserção de nós de sensores, modificar entradas de sensores, inserir perturbações e mobilidade na rede. A simulação do conjunto de instruções executa os códigos binários de *firmware* compilados usando ferramentas de compilação cruzada e um simulador de microcontrolador externo. O método de simulação usado foi UDGM *Distance Loss*, a camada de roteamento usada foi a *Routing Protocol for Low power and Lossy Networks* (RPL) [12], que faz o gerenciamento de rotas da rede 6LoWPAN [13]. Para evitar qualquer interferência no envio dos dados, não foi configurada nenhuma camada *Radio Duty Cycle* (RDC) no CONTIKI.

O *software* desenvolvido foi dividido em três partes: MQTT-SN *Client*, MQTT-SN *Gateway* e MQTT *Broker*. A *camada de aplicação Client* e *Gateway* rodam no sistema operacional CONTIKI-OS e a *camada de aplicação Broker* roda nativamente no sistema operacional Linux, onde a plataforma de testes foi instalada. Como ponte entre o ambiente de simulação COOJA e o sistema operacional Linux, foi necessário usar os utilitários TUNSLIP, que faz um tunelamento IP via serial e o TAYGA, que faz a tradução do IPv6 para IPv4, uma vez que o *serviço de gerenciamento* MQTT instanciado no servidor utiliza apenas a pilha de protocolos IPv4. Para os testes do CoAP, todos os utilitários estavam presentes dentro do ambiente COOJA.

B. Descrição das Simulações Realizadas

Os testes foram realizados no simulador COOJA usando dois tipos de nós, criados com a base de *hardware* WISMOTE. O primeiro tipo de nó simula os sensores acústicos na rede e o segundo simula o *gateway*, que fará a conexão com outra rede para envio dos dados. Para os testes do protocolo MQTT-SN, foram usados os parâmetros $QoS = 1$ (comunicação consiste em duas mensagens (PUBLISH e PUBACK), porta 1884 UDP, envio de mensagens de 10 em 10 segundos com até 4 tentativas de registro de tópico entre os nós sensores e o servidor MQTT. Para os testes do protocolo CoAP, foram usados os parâmetros de porta 61616 UDP e requisições de 10 em 10 segundos, considerando mensagens que garantam confirmação (CoAP CON e ACK).

Foram testados 1, 5, 10, 15, 20, 30 e 40 nós com alcance direto ao *gateway* e distribuídos ao seu redor em forma de elipse a fim de verificar a entrega dos dados para os servidores MQTT e CoAP bem como outras características de transmissão/recepção.

A topologia estrela foi escolhida, pois é tipicamente encontrada em rede de sensores para monitoramento de parâmetros marinhos [26], [27]. Como exemplo de aplicação, pode-se implementar o *gateway* em uma plataforma de petróleo, cuja a fonte de energia não se esgota, e os nós sensores possuem alcance direto ao sistema de monitoramento. Foi utilizado o comando CoAP POST e MQTT-SN PUBACK para realizar o envio das mensagens de texto. A Fig. 4 mostra a interface gráfica do simulador COOJA. Os resultados foram coletados através da própria interface gráfica que disponibiliza as informações de cada nó instanciado na rede com precisão de microssegundos.

Figura. 4. Interface gráfica da ferramenta de simulação COOJA com a arquitetura de rede de sensores proposta para aplicações marinhas.

IV. RESULTADOS E ANÁLISE

A. Análise de Desempenho dos Protocolos

Os dois protocolos foram avaliados comparando-se os seguintes indicadores de desempenho em função da quantidade de nós sensores: latência, tempo de tráfego na rede, quantidade de pacotes trafegados, quantidade de *bytes* trafegados e consumo de energia.

A Fig. 5 apresenta as curvas de latência para cada protocolo. Compreende-se a latência [15], *Round Trip Time* (RTT), como o tempo entre o início da comunicação até o momento de recepção da mensagem pelo servidor de aplicação, considerando o processamento do protocolo.

O protocolo MQTT-SN apresentou latência de 36 ms na entrega da mensagem para um nó. Levou-se em consideração a latência entre as mensagens PUBLISH e PUBACK. Pode-se verificar um acréscimo de, aproximadamente, 2 ms a cada nó adicionado, levando em consideração o processamento do servidor MQTT.

Para os testes do protocolo CoAP, a latência foi de, aproximadamente, 231 ms para a resposta do dispositivo considerando novamente um único nó após a requisição do servidor. Dessa maneira, o protocolo CoAP é 195 ms mais lento que o MQTT-SN. Essa diferença de valores se mantém aproximadamente a mesma para qualquer quantidade de nós na rede. Sendo assim, foi possível verificar que o MQTT-SN é viável para a aplicação proposta e apresenta menor latência na entrega dos dados. Pode-se sugerir que o menor tempo de latência do protocolo MQTT-SN ocorre devido ao menor tamanho do pacote necessário para enviar mensagens entre os nós e o *gateway*.

A Fig. 6 mostra o tempo de tráfego na rede para os dois protocolos. Pode-se notar que ambos os protocolos apresentam praticamente o mesmo desempenho com relação ao tempo de tráfego na rede. Isso ocorre, pois ambas as pilhas de protocolos utilizam a mesma camada de roteamento (RPL), demonstrando que o tempo de atraso se deve exclusivamente ao processamento dos protocolos tanto nos nós transmissores como no *gateway*.

Durante os testes uma mesma mensagem foi enviada dos nós sensores para o *gateway*. A mesma mensagem consiste numa frase concatenada com os últimos dois *bytes* do endereço IPv6 do nó sensor. Quando se compara a quantidade de pacotes trafegados (Fig. 7) o MQTT-SN apresenta um tráfego 2,15 vezes maior que o CoAP, para qualquer quantidade de nós. Isso se deve à fragmentação de mensagem realizada pelo MQTT-SN. O protocolo CoAP, por ser baseado no protocolo HTTP, encaminha mensagens mais longas, pois possui métodos que garantem a entrega. O MQTT-SN encaminha mensagens mais curtas, utilizando menor banda.

A Fig. 8 apresenta a quantidade de *bytes* trafegados. O protocolo MQTT-SN transmite a mesma mensagem utilizando somente 70% da quantidade de *bytes* do protocolo CoAP. Isso é esperado, pois o CoAP possui maior número de *bytes* para encaminhar os cabeçalhos e as mensagens. Dessa forma, o MQTT-SN realiza o envio da mesma informação com menor quantidade de dados.

Com relação ao consumo de energia, a Fig. 9 mostra que o consumo do MQTT-SN é levemente melhor, sendo aproximadamente 10% menor que o consumo do CoAP. Como mencionado anteriormente, as características do MQTT-SN, entre elas menor tráfego de *bytes* e menor latência, resultam no menor consumo de energia dos nós.

Figura 5. Resultados de latência utilizando os protocolos MQTT-SN e CoAP.

Figura 6. Comparação entre o tempo de tráfego na rede (ms) da arquitetura MQTT-SN e CoAP.

Figura 7. Comparação entre a quantidade de pacotes trafegados utilizando os protocolos MQTT-SN e CoAP.

Figura 8. Quantidade de *bytes* trafegados utilizando os protocolos MQTT-SN e CoAP.

Figura 9. Comparação entre o consumo de energia dos nós sensores utilizando os protocolos MQTT-SN e CoAP.

A comparação entre os protocolos pode ser sintetizada na Tabela II. Nessa tabela são detalhados os resultados obtidos para 40 nós.

PARÂMETROS DO PROTOCOLO	MQTT-SN	CoAP	COMPARATIVO
Latência da rede (ms)	271	473	MQTT-SN é 200 ms mais rápido comparado ao CoAP.
Tempo de tráfego na rede (ms)	20	20	Ambos os protocolos possuem aproximadamente o mesmo tempo de tráfego na rede.
Pacotes trafegados	366	170	O MQTT-SN fragmenta a informação em maior número de pacotes, cerca de 2,15 vezes.
Bytes trafegados	1464	2040	O MQTT-SN transmite a mesma mensagem com apenas 70% dos <i>bytes</i> utilizados no CoAP.
Consumo de energia dos nós (mW)	42,4	46,7	O MQTT-SN apresenta consumo de energia 10% menor do que o CoAP.

V. CONCLUSÕES

Por meio de testes de simulação foi possível comparar os protocolos MQTT-SN e CoAP com relação a: latência para a entrega dos dados, tempo de tráfego na rede, tráfego de pacotes e de *bytes* e consumo energético. Para realizar essa simulação, foi implementado o protocolo MQTT-SN dentro do sistema operacional Contiki e do simulador COOJA. Conforme os resultados obtidos, o protocolo MQTT-SN apresentou, em geral, desempenho melhor que o CoAP. O MQTT-SN mostrou latência da ordem de 200 ms menor que o CoAP, mesmo trafegando uma quantidade maior de pacotes. Também apresentou consumo energético 10% menor que o CoAP. Dessa forma, o protocolo MQTT-SN se mostra como uma alternativa mais adequada para implementação em aplicações IoT para redes de sensores sem fio em ambientes hostis, em especial, para monitoramento acústico marinho.

REFERÊNCIAS

- [1] E. R. Vivas, B. L. Lopez, "Construction, calibration, and field test of a home-made, low-cost hydrophone system for cetacean acoustic research," *Proceedings of Meetings on Acoustics*, vol. 11, 2011.
- [2] M. O. Lammers, R. E. Brainard, W. W. L. Au, T. A. Mooney, K. B. Wong, "An ecological acoustic recorder (EAR) for long-term monitoring of biological and anthropogenic sounds on coral reefs and other marine habitats," *The Journal of the Acoustical Society of America*, vol. 123, 2008.
- [3] R. S. Sousa-Lima, T. F. Norris, J. N. Oswald, D. P. Fernandes, "A review and inventory of fixed autonomous recorders for passive acoustic monitoring of marine mammals," *Acoustics in Underwater Geosciences Symposium (RIO Acoustics)*, 2013.
- [4] S. M. Wiggins, J. A. Hildebrand, "High-frequency Acoustic Recording Package (HARP) for broadband, long-term marine mammal monitoring," *Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies*, 2007.
- [5] J. J. Martinez, J. R. Myers, T. J. Carlson, Z. D. Deng, J. S. Rohrer, K. A. Caviggia, "Design and Implementation of an Underwater Sound Recording Device," *Sensors (Basel)*, vol. 11, no. 9, pp. 8519-35, 2011.
- [6] C. Alexandre Gouveia da Silva, E. Leonardo dos Santos, A. Christian Krainski Ferrari and H. Tertuliano dos Santos Filho, "A Study of the Mesh Topology in a ZigBee Network for Home Automation Applications," *IEEE Latin America Transactions*, vol. 15, no. 5, pp. 935-942, May 2017.
- [7] D. C. Yacchirema Vargas and C. E. Palau Salvador, "Smart IoT Gateway for Heterogeneous Devices Interoperability," *IEEE Latin America Transactions*, vol. 14, no. 8, pp. 3900-3906, Aug. 2016.
- [8] P. D. Godoy, R. Cayssials and C. Garcia Garino, "A WSN Testbed for Teaching Purposes," *IEEE Latin America Transactions*, vol. 14, no. 7, pp. 3351-3357, July 2016.
- [9] D. Adorno Gomes and D. Bianchini, "Interconnecting Wireless Sensor Networks with the Internet Using Web Services," *IEEE Latin America Transactions*, vol. 14, no. 4, pp. 1937-1942, April 2016.
- [10] Z. Shelby, K. Hartke, C. Bormann, "Constrained Application Protocol (CoAP)," Internet Engineering Task Force (IETF), "Internet-Draft, draft-ietf-core-coap-18, 2013.
- [11] OASIS Standard, "MQTT v3.1.1 Protocol Specification," 2014.
- [12] A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur, R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," *Internet Engineering Task Force (IETF)*, "Internet-Draft, draft-ietf-roll-rpl, 2012.
- [13] N. Kushalnagar, J. Hui, D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," *Internet Engineering Task Force (IETF)*, "Internet-Draft, draft-ietf-6lowpan-format, 2007.
- [14] D. Thangavel, X. Ma, A. Valera, H. X. Tan and C. K. Y. Tan, "Performance evaluation of MQTT and CoAP via a common middleware," *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, Singapore, 2014, pp. 1-6.
- [15] N. De Caro, W. Colitti, K. Steenhaut, G. Mangino and G. Reali, "Comparison of two lightweight protocols for smartphone-based sensing," *2013 IEEE 20th Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*, Namur, 2013, pp. 1-6.
- [16] M. H. Amaran, N. A. M. Noh, M. S. Rohmad, H. Hashim, "A Comparison of Lightweight Communication Protocols in Robotic Applications," *Procedia Computer Science*, vol. 76, pp. 400-405, 2015.
- [17] L. Atzori, A. Iera and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, pp. 2787-2805, 2010.
- [18] R. Khan, S. U. Khan, R. Zaheer and S. Khan, "Future internet: The internet of things architecture, possible applications and key challenges," *Frontiers of Information Technology (FIT)*, 2012 10th International Conference On, 2012, pp. 257-260.
- [19] P. Lopez, D. Fernandez, A. J. Jara and A. F. Skarmeta, "Survey of internet of things technologies for clinical environments," *Advanced Information Networking and Applications Workshops (WAINA)*, pp. 1349-1354, 2013.
- [20] D. Yang, F. Liu and Y. Liang, "A survey of the internet of things," *Proceedings of the 1st International Conference on EBusiness Intelligence (ICEBI2010)*, 2010, pp. 358-366.
- [21] Z. Sheng, S. Yang, Y. Yu, A. V. Vasilakos, J. A. McCann and K. K. Leung, "A survey on the IETF protocol suite for the internet of things: standards, challenges, and opportunities," *Wireless Communications, IEEE*, vol. 20, pp. 91-98, 2013.
- [22] Z. Yang, Y. Yue, Y. Yang, Y. Peng, X. Wang and W. Liu, "Study and application on the architecture and key technologies for IOT," *2011 International Conference on Multimedia Technology (ICMT2011)*, Hangzhou, 2011, pp. 747-751.
- [23] M. Wu, T. J. Lu, F. Y. Ling, J. Sun and H. Y. Du, "Research on the architecture of internet of things," *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, Chengdu, 2010, pp. V5-484-V5-487.
- [24] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," *29th Annual IEEE International Conference on Local Computer Networks*, 2004, pp. 455-462.
- [25] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "CrossLevel Sensor Network Simulation with COOJA," *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, Tampa, FL, 2006, pp. 641-648.
- [26] D. Fanton, "Connecting the IoT MQTT and CoAP protocols," *IO Hub*, [Online]. Available: <http://www.logicsupply.com>. [Accessed: 17- Oct-2017]
- [27] C. Albaladejo, P. Sánchez, A. Iborra, F. Soto, J.A. López, R. Torres. "Wireless Sensor Networks for Oceanographic Monitoring: A Systematic Review," *Sensors (Basel)*, vol. 10, no. 7, pp. 6948-6968, 2010.
- [28] G. Xu, W. Shen, X. Wang, "Applications of Wireless Sensor Networks in Marine Environment Monitoring: A Survey," *Sensors (Basel)*, vol. 14, no. 9, pp. 16932-16954, 2014.

Gabriel Durante received his B.Sc. degree in Electrical Engineering from Fundação Instituto Tecnológico de Osasco (FAC-FITO). His main research interests are related to IoT, sensors networks, digital signal processing, embedded systems, and machine learning.

Wesley Beccaro received his B.Sc., M.Sc. and Ph.D. degrees in Electrical Engineering from the Escola Politécnica da Universidade de São Paulo (USP). He is currently a researcher at the Department of Electronic Systems Engineering in the same institution and his research interests are related to control systems, embedded systems and digital signal processing.

Henrique E. M. Peres received his M.Sc. and Ph.D. degrees both in Electrical Engineering from the Escola Politécnica - Universidade de São Paulo, Brazil. He is currently a researcher at the Department of Electronic Systems Engineering in the same institution, and his research interest is related to MEMS, integrable sensors and nanomaterials.