



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Implementação e avaliação do mecanismo TSCH do IEEE 802.15.4e no simulador ns-3

Luis Alberto Belem Pacheco

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Orientadora

Prof.^a Dr.^a Priscila América Solis M. Barreto

Brasília
2015

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Curso de Engenharia da Computação

Coordenadora: Prof. Dr. Ricardo Zelenovisky

Banca examinadora composta por:

Prof.^a Dr.^a Priscila América Solis M. Barreto (Orientadora) — CIC/UnB
Prof. Marcos F. Caetano — CIC/UnB
Prof. Dr. Jacir Bordim — CIC/UnB

CIP — Catalogação Internacional na Publicação

Pacheco, Luis Alberto Belem.

Implementação e avaliação do mecanismo TSCH do IEEE 802.15.4e no simulador ns-3 / Luis Alberto Belem Pacheco. Brasília : UnB, 2015.

78 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2015.

1. 802.15.4, 2. redes de sensores sem fio, 3. RSSF, 4. TSCH, 5. tdma, 6. ns-3, 7. simuladores de rede

CDU 004

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília-DF — Brasil



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Implementação e avaliação do mecanismo TSCH do IEEE 802.15.4e no simulador ns-3

Luis Alberto Belem Pacheco

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Prof.^a Dr.^a Priscila América Solis M. Barreto (Orientadora)
CIC/UnB

Prof. Marcos F. Caetano Prof. Dr. Jacir Bordim
CIC/UnB CIC/UnB

Prof. Dr. Ricardo Zelenovisky
Coordenadora do Curso de Engenharia da Computação

Brasília, 7 de agosto de 2015

Dedicatória

Dedico este trabalho à minha família, graças ao apoio e carinho que recebi tive o privilégio de ingressar na universidade e por fim realizar este trabalho.

Agradecimentos

Agradeço aos amigos e familiares que me ajudaram no desenvolvimento deste trabalho. Especialmente aos amigos do intercâmbio Leandro Vieira, Thales Bertaglia, João Martins e Ariane Vizzuso, e aos amigos de Brasília Lucas Saad e Arthur Rodrigues. Também agradeço aos professores que me instruíram e possibilitaram esse trabalho Tom Vermeulen e Priscila Solis.

Resumo

O IEEE 802.15.4e[6] é uma nova emenda ao protocolo 802.15.4[5]. Este trabalho estuda o desempenho do modo operacional TSCH em relação a mudanças no ambiente e necessidades da aplicação. Este trabalho também implementa o TSCH no simulador de redes de código livre ns-3[18].

O estudo do desempenho do TSCH em circunstâncias dinâmicas foi realizado através do desenvolvimento de um cenário de simulação, uma rede de sensores acústicos sem fio em uma residência doméstica é usada para automatizar tarefas do ambiente. Para simular um ambiente dinâmico um dispositivo Wi-Fi[2] causa interferência na rede de sensores sem fio. Mudanças na aplicação são simuladas através das diferentes cargas encontradas em um dia típico de uma residência doméstica.

Os resultados obtidos mostram claramente que a rede precisa se adaptar a mudanças no ambiente e na aplicação. Um estudo foi realizado para demonstrar que ao empregar técnicas de autoaprendizagem a rede pode alcançar seu desempenho esperado.

Palavras-chave: 802.15.4, redes de sensores sem fio, RSSF, TSCH, tdma, ns-3, simuladores de rede

Abstract

The IEEE 802.15.4e[6] is a new amendment for the 802.15.4[5] standard. This work studies the performance of the TSCH operational mode regarding changes in the environment and in the application's requirements. This work implement the TSCH in the well know open-source network simulator ns-3.

A simulation scenario is designed to verify TSCH's performance under dynamic circumstances, an acoustic wireless sensor network is used in a house to automate common tasks. In order to simulate changes in the environment an intermittent Wi-Fi[2] device causes interference in the wireless sensor network. The simulation of a common day's activity in a house achieves changes in the application's requirements.

The obtained results clearly shows that the network needs to adapt to changes in the environment and application. The employment of self-learning techniques enables the network to achieve its required performance.

Keywords: 802.15.4,network simulation,wireless sensor networks, wsn,TSCH,tdma,ns-3

Sumário

1	Introdução	1
2	O Protocolo IEEE 802.15.4	5
2.1	Topologias	5
2.2	Camada de enlace	6
2.2.1	Estrutura do quadro da camada de enlace	8
2.3	Camada física	8
2.4	Emenda IEEE 802.15.4e	9
2.5	Resumo do capítulo	15
3	Estado da arte	17
3.1	Simulação de redes	17
3.2	Comparação entre ns-2 e ns-3	18
3.3	Situação do modelo IEEE 802.15.4 no ns-3	20
3.4	Autoaprendizagem em redes de sensores sem fio	22
3.5	Resumo do capítulo	23
4	O Simulador ns-3	24
4.1	Arquitetura do ns-3	24
4.2	O modelo de espectro	26
4.3	Implementação do IEEE 802.15.4 no ns-3	30
4.4	Implementação do modo de operação TSCH no ns-3	32
4.5	Resumo do capítulo	41
5	Resultados	42
5.1	Simulação	42
5.2	Resultados da simulação	45
5.2.1	Autoaprendizagem	54
5.3	Resumo do capítulo	58

6 Conclusão	59
Referências	63

Lista de Figuras

1.1	Exemplo de rede de sensores sem fio	1
2.1	Camadas definidas pelo protocolo IEEE 802.15.4[5]	6
2.2	Topologias suportadas pelo protocolo IEEE 802.15.4[5]	7
2.3	Estrutura do <i>superframe</i> do protocolo IEEE 802.15.4[5]	7
2.4	Período ativo no IEEE 802.15.4[5]	8
2.5	Quadro definido pelo protocolo IEEE 802.15.4[5]	8
2.6	Frequências suportadas pelo protocolo IEEE 802.15.4[5]	10
2.7	Quadro da camada física do protocolo IEEE 802.15.4 para a modulação O-QPSK [5]	10
2.8	Estrutura do quadro da emenda IEEE 802.15.4e[6]	11
2.9	Formato dos Elementos de Informação[6]	11
2.10	Estrutura do <i>superframe</i> para redes LLDN[5]	12
2.11	Estrutura do DSME [5]	13
2.12	Estrutura do <i>slotframe</i> TSCH[5]	14
2.13	Estrutura do <i>timeslot</i> TSCH[5]	14
2.14	Salto entre canais[5]	15
2.15	Múltiplos <i>slotframes</i> [5]	15
3.1	Ferramentas de visualização do ns-2 e ns-3	19
4.1	Estrutura da classe Node do ns-3	25
4.2	Arquitetura do modelo <i>Spectrum</i> do ns-3	27
4.3	Arquiteturas do <i>SpectrumChannel</i> e camada física do ns-3	28
4.4	Máscara do espectro da transmissão do padrão IEEE 802.11[2]	29
4.5	Diagrama simplificado da implementação do padrão IEEE 802.15.4 no ns-3	30
4.6	Máscara do espectro do padrão IEEE 802.15.4 para o canal 16 (frequência central = 2430MHz) com uma potência de 0dBm	33
4.7	Adições realizadas na classe <i>LrWpanMac</i> para habilitar o modo TSCH	33
4.8	Fluxograma de uma rede em modo TSCH	35

4.9	<i>Timeslots</i> de transmissão e recepção	36
4.10	Diagrama simplificado das classes <i>LrWpanSpectrumInterfence</i> e <i>LrWpanSpectrumErrorModel</i>	37
4.11	Vazão de uma rede TSCH configurada com <i>timeslots</i> de 10ms com vários tamanhos de pacotes.	39
4.12	Validação do <i>SpectrumErrorModel</i>	39
5.1	Residência utilizada como cenário de simulação	43
5.2	Configuração do <i>slotframe</i> para o cenário proposto	43
5.3	Atividade da rede de sensors durante o dia	44
5.4	Comparação entre as mascaras de espectro do Wi-Fi e IEEE 802.15.4 . . .	46
5.5	BER de cada canal de todos os dispositivos	47
5.6	Vazão simulada, com e sem interferência	49
5.7	Atraso do dispositivo N	51
5.8	Consumo de energia, em J , dos dispositivos O, L e I em cada região e durante toda a simulação	53
5.9	Vazão com e sem técnicas de autoaprendizagem	56
5.10	Atraso do dispositivo O com e sem técnicas de autoaprendizagem	57

Lista de Tabelas

3.1	Lista dos serviços da camada de enlace do padrão IEEE 802.15.4-2006, a situação atual de suporte no ns-3 e os serviços utilizados pelo TSCH.	21
3.2	Lista dos serviços da camada física do padrão IEEE 802.15.4-2006, a situação atual de suporte no ns-3 e os serviços utilizados pelo TSCH.	21
4.1	Serviços TSCH	34
4.2	Tipos de <i>timeslot</i>	40
4.3	Configuração to <i>timeslot</i> do OpenWSN	40
4.4	Carga consumida por cada tipo de <i>timeslot</i> , medida e calculada	41
4.5	Energia total consumida em um <i>slotframe</i>	41
5.1	Níveis de atividade	45
5.2	Vazão máxima para cada nível de atividade	48
5.3	Corrente consumida, em μC , para cada tipo de <i>timeslot</i>	52
5.4	Média da corrente consumida por cada dispositivo durante a simulação	54
5.5	Tempo de vida de cada dispositivo considerando uma bateria de $2600mAh$	54

Lista de Abreviaturas e Siglas

A-TSCH Adaptive Time Slotted Channel Hopping. 3, 23

ASN Absolute Slot Number. 14, 15

AWGN Additive White Gaussian Noise. 28, 38

CAP Contention access period. 7, 15

CCA Clear channel assessment. 9, 13, 20, 21, 31, 35, 37, 46, 50, 61

CFP Contention Free Period. 6, 7, 15

CH Channel Hopping. 3, 9

CSMA/CA Carrier sense multiple access with collision avoidance. 6–9, 12, 14, 15, 20, 31, 38

DSME Deterministic and Synchronous Multi-channel Extension. x, 9, 12, 13, 15, 23, 59, 62

DSSS Direct-Sequence Spread Spectrum. 9, 26

ED Energy detection. 9, 20

FFD Full Function Device. 5, 6

GACK Group acknowledgment. 12

GTS Guaranteed time slot. 7, 21, 23, 59

GUI Graphical user interface. 18, 24

ID Identifier. 11, 12

IE Information element. 11

IEEE Institute of Electrical and Electronics Engineers. viii, x–xii, 2–12, 15, 17–23, 25, 26, 29, 30, 32–34, 36, 42, 43, 46, 52, 55, 59, 60, 62

ISM Industrial, scientific and medical. 2

LLDN Low latency deterministic network. x, 9, 12, 15, 59, 62

LQI Link Quality Indicator. 13, 22, 23, 55

LR-WPAN Low-Rate Wireless Personal Area Networks. 5, 26, 32, 44

MAC Medium Access Control. 2, 25

MCPS MAC Data Service. 20, 21, 37

MLME MAC Management Service. 20, 21, 34

ns-2 Network-Simulator 2. viii, x, 4, 18, 19, 60

ns-3 Network-Simulator 3. viii, x, xii, 3, 4, 18–21, 23–32, 34, 38, 41, 42, 45, 59, 60, 62

O-QPSK Offset quadrature phase-shift keying. x, 9, 10

OSI Open Systems Interconnection. 2, 5, 22

PD PHY Data service. 20–22, 35, 37

PHR PHY header. 9

PHY Physical layer. 8

PLME PHY Management Service. 20, 21, 35, 37

PPDU PHY protocol data unit. 9

PSD Power Spectral Density. 26, 27, 29, 32, 38, 44, 45, 60

RFD Reduced Function Device. 5, 6

SAML Self-Adapting MAC Layer. 23

SHR Synchronization header. 9

SINR Signal-to-interference-plus-noise ratio. 28, 32, 38

TCP Transmission Control Protocol. 22

TG4 Task Group 4. 2

TSCH Timeslotted channel hopping. viii, x–xii, 2–4, 11, 13–15, 20–24, 32–43, 45, 46, 48, 52, 54, 58–62

WPAN Wireless Personal Area Network. 2

Capítulo 1

Introdução

Redes de sensores sem fio são compostas por diversos nós de sensores que ligam o mundo físico ao mundo digital, transformando informações analógicas em dados digitais. As funções de comunicação sem fio de tais dispositivos possibilitam a transmissão dos dados adquiridos para uma unidade de processamento central, que por sua vez toma decisões de acordo com a interpretação da informação recebida. A Figura 1.1 mostra um exemplo de redes de sensores sem fio em que vários dispositivos enviam dados para uma unidade central.

Redes de sensores sem fio possuem diversas aplicações, apesar de em seu início o foco estar direcionado principalmente ao uso na defesa e no espaço, atualmente a variedade de aplicações é ampla e abrange diversos campos. Redes de sensores sem fio são utilizadas para monitorar e controlar áreas industriais, automatizar residências, prover assistência médica monitorando pacientes, etc.

Apesar de permitir diversas novas possibilidades, redes de sensores sem fio também trazem muitos desafios, dispositivos com consumo de energia limitado demandam Sistemas Operacionais e pilhas de rede especialmente projetados. Para possibilitar comunicação sem fio em dispositivos com tais restrições o Instituto dos Engenheiros Elétricos

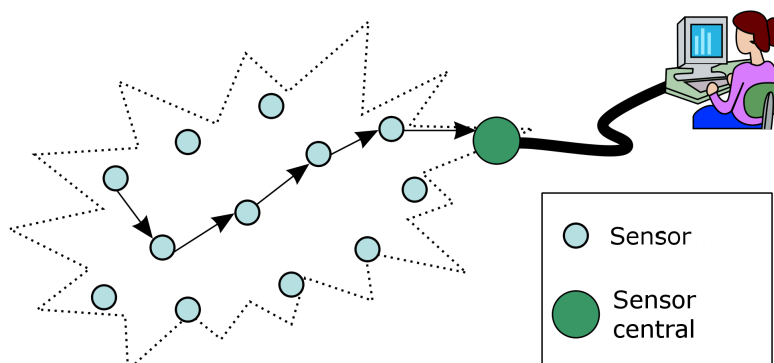


Figura 1.1: Exemplo de rede de sensores sem fio

e Eletrônicos (do inglês *Institute of Electrical and Electronics Engineers (IEEE)*) 802.15 WPANTM Task Group 4 (TG4) [13] projeta padrões que são utilizados em todo o mundo, esse trabalho tem como ponto central aprimorar o padrão IEEE 802.15.4[4] e sua emenda 802.15.4e[6]. Esse trabalho contribui para o estado da arte em relação ao desempenho do 802.15.4e em circunstâncias dinâmicas.

A primeira versão do protocolo IEEE 802.15.4 foi lançada em 2003[3], definindo as camadas 1 e 2 do modelo de Interconexão de Sistemas Abertos (do inglês *Open Systems Interconnection (OSI)*)[15] para Redes de Área Pessoal Sem Fio (do inglês *Wireless Personal Area Network (WPAN)*). Para possibilitar a comunicação em diversos países são utilizadas as faixas de rádio industriais, científicas e médicas (do inglês *Industrial, scientific and medical (ISM)*). A camada de acesso ao meio (do inglês *Medium Access Control (MAC)*) define dois modos de operação diferentes: um modo assíncrono, em que sensores tentam acessar o meio sempre que necessário e um modo síncrono, que é gerenciado por um dispositivo coordenador que estipula quando sensores podem acessar o meio. Em 2012 a emenda 802.15.4e foi lançada com a intenção de abranger as áreas de aplicações suportadas pelo padrão, mais 3 modos de operação foram definidos para principalmente possibilitar aplicações industriais.

Esse trabalho estuda o modo operacional *Timeslotted channel hopping (TSCH)* da emenda IEEE 802.15.4e, neste modo os dispositivos transmitem e recebem dados de acordo com uma agenda previamente determinada pelo coordenador da rede, que se repete no tempo. A emenda a técnica de salto entre canais, a qual consiste na mudança das faixas de frequência (ou canais) em que a comunicação ocorre para que haja um aumento na resiliência à interferência. Por exemplo, se um canal é afetado por interferência e a comunicação não é bem sucedida, na próxima tentativa outro canal será utilizado, aumentando as chances de uma comunicação bem sucedida.

Um problema comum em muitas áreas em que redes de sensores sem fio são utilizadas é como lidar com mudanças nos requisitos da aplicação e nas condições do ambiente. Tais mudanças podem causar impacto significativo no rendimento da rede, sendo assim imperativo que a rede tome ações para se adaptar em cenários dinâmicos. Sendo preferível a menor quantidade de intervenção humana durante o funcionamento da rede, uma abordagem de autoaprendizagem, ou uma adaptação automática de acordo com as mudanças percebidas, pode ser um aspecto necessário. O padrão IEEE 802.15.4 e sua emenda não oferecem nenhum tipo de autoaprendizagem, esse trabalho mostra os ganhos no desempenho do modo de operação TSCH quando sendo adaptado automaticamente às mudanças no ambiente e nos requisitos da aplicação.

Para estudar os efeitos que uma mudança no ambiente ou nos requisitos da aplicação teriam no desempenho de uma rede TSCH uma ferramenta de simulação é necessária.

Apesar de haver vários simuladores de rede que implementam o padrão IEEE 802.15.4, a situação é diferente para a sua emenda, nesse caso, no melhor do nosso conhecimento não está disponível uma ferramenta de simulação que suporta a emenda IEEE 802.15.4e, acredita-se que devido ao seu lançamento relativamente recente. Uma das principais realizações desse trabalho é a implementação e validação de um simulador para redes TSCH.

Diversos trabalhos foram propostos em relação à autoaprendizagem em redes de sensores sem fio. O Adaptive Time Slotted Channel Hopping (A-TSCH)[9] muda os canais disponíveis pelo sistema de *Channel Hopping (CH)* do TSCH para usar apenas canais com um mínimo de qualidade, dessa maneira aumentando as chances de uma comunicação bem sucedida, foram observadas melhoras entre 5% e 8% na relação entre pacotes enviados e recebidos. As melhorias são relativamente baixas devido às condições do meio não serem muito desfavoráveis, diminuindo assim as possibilidades de aperfeiçoamento. Apesar do A-TSCH melhorar o uso do meio de comunicação, uma redução no desempenho da rede é observada, pois os nós agora precisam de certo tempo para medir as condições do meio e sincronizar a nova lista de canais a serem utilizados, em redes pequenas esse tempo afeta o desempenho de forma significativa.

Neste trabalho foi realizado um estudo para avaliar as ferramentas de simulações atuais e decidir qual a melhor para a implementação do modo de operação TSCH. O desenvolvimento do modelo TSCH foi realizado com o simulador ns-3[18]. A arquitetura do simulador foi estudada em profundidade, e também uma implementação parcial do padrão IEEE 802.15.4. A implementação do modo TSCH foi feita modificando e adicionando funcionalidades na implementação parcial já presente.

Para exemplificar a necessidade de funcionalidades de autoaprendizagem um cenário de simulação foi projetado: uma residência doméstica que possui uma rede de sensores acústicos sem fio, utilizados para automatizar várias tarefas, tais como acender/apagar luzes conforme a presença dos moradores e detectar acidentes. Para simular um ambiente dinâmico uma rede Wi-Fi[2] também está presente na casa, sua atividade muda durante o dia, podendo causar grande interferência na rede de sensores. Para simular mudanças nos requisitos da aplicação a rede de sensores muda o número de dispositivos ativos de acordo com a atividade dos moradores na casa, por exemplo, se todos estão dormindo apenas alguns sensores estarão ativos.

Para demonstrar os benefícios da adaptação automática da rede na presença das mudanças descritas uma simulação é implementada. Os resultados foram comparados contra uma simulação em que não é realizada nenhuma adaptação. O impacto da aplicação de técnicas de autoaprendizagem em uma rede TSCH é medido em relação à vazão, atraso e consumo de energia.

Esse trabalho designa a responsabilidade da autoaprendizagem para as camadas superiores, o cenário proposto evidencia a perda no desempenho quando não são utilizadas técnicas de autoaprendizagem e o ganho obtido quando adaptações são feitas durante o tempo de execução.

Em resumo, esse trabalho apresenta um modelo do padrão IEEE 802.15.4e para o simulador de redes ns-3. A implementação é validada contra resultados conhecidos. Em seguida o desempenho do 802.15.4e é analisado em uma rede de sensores doméstica. Finalmente, são explorados os limites superiores e inferiores do ganho de desempenho ao se utilizar técnicas de autoaprendizagem.

O capítulo 2 apresenta uma visão geral do protocolo IEEE 802.15.4 e da emenda 802.15.4e, as principais características necessárias para compreender esse trabalho são evidenciadas. O capítulo 3 introduz o estado da arte relevante a este trabalho, a primeira seção apresenta uma visão geral do cenário atual em ferramentas de simulação de redes, é feita uma classificação e avaliação para estipular o simulador mais adequado a este trabalho, a segunda seção compara os simuladores ns-2 e ns-3, a terceira seção apresenta a arquitetura do ns-3 e a atual implementação do IEEE 802.15.4, a última seção introduz o tópico de autoaprendizagem e o estado da arte envolvendo o TSCH. O capítulo 4 apresenta em mais detalhes a arquitetura do simulador ns-3, a primeira seção exibe as principais bibliotecas utilizadas nesse trabalho, a segunda seção mostra em detalhes a atual implementação do protocolo e a última seção apresenta a implementação do modo de operação TSCH realizada nesse trabalho. O capítulo 5 expõe os resultados desse trabalho, a primeira seção apresenta o cenário de simulação proposto usado para evidenciar a necessidade de técnicas de autoaprendizagem e a segunda seção apresenta os resultados obtidos ao executar o cenário proposto. Finalmente o capítulo 6 conclui esse trabalho ao sumarizar tudo que foi realizado e apresentar trabalhos futuros que podem ser realizados a partir desse estudo.

Capítulo 2

O Protocolo IEEE 802.15.4

A primeira versão do protocolo IEEE 802.15.4 concebida em 2003[3] pelo Grupo de Trabalho IEEE 802.15[13], como mostrado na Figura 2.1 ele especifica as camadas física e de enlace (camadas 1 e 2 do modelo OSI[15], respectivamente) para Redes Sem Fio Pessoais de Baixas Taxas de Transmissão (do inglês *Low-Rate Wireless Personal Area Networks (LR-WPAN)*). Mais duas versões foram lançadas, uma em 2006[4] e outra em 2011[5], visando melhorar e expandir suas características. É importante notar que as novas versões são compatíveis com versões anteriores, o que significa que um dispositivo compatível com o padrão 2006 pode trabalhar em uma rede em conformidade com a norma de 2003, o mesmo se aplica para a versão de 2011.

Várias emendas também foram lançadas, a emenda 802.15.4e[6], lançada em 2012, acrescentou três diferentes modos de operação da camada de enlace, com foco em melhor desempenho para ambientes industriais. Esta seção primeiro descreve as topologias do protocolo, em seguida, descreve brevemente as camadas física e de enlace, finalmente, a emenda 802.15.4e é descrita.

2.1 Topologias

O protocolo define dois tipos de dispositivos diferentes, um aparelho com funções completas (do inglês *Full Function Device (FFD)*) e um dispositivo de função reduzida (do inglês *Reduced Function Device (RFD)*), FFDs implementam todos os recursos necessários, podem atuar como dispositivos comuns como coordenadores de rede, enquanto RFDs são dispositivos simples, com ainda mais restrições e só pode falar com coordenadores, eles são destinados a aplicações extremamente leves.

Duas topologias diferentes são suportadas pelo protocolo, dependendo do tipo de aplicação, a rede pode ser executada em uma topologia de estrela ou em uma topologia ponto-a-ponto. A Figura 2.2 mostra ambas as topologias suportadas, a topologia de es-

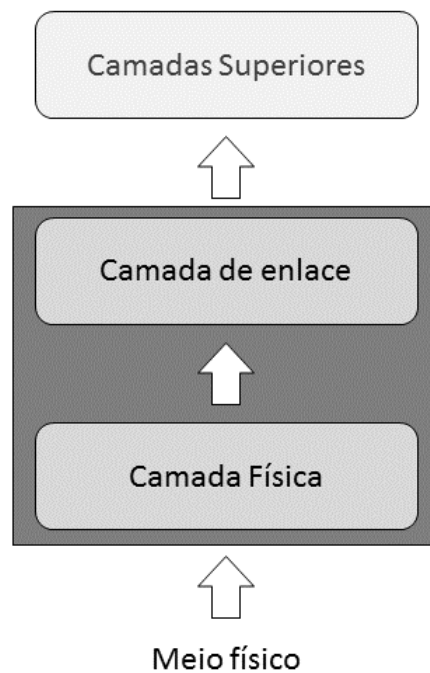


Figura 2.1: Camadas definidas pelo protocolo IEEE 802.15.4[5]

trela contém um Coordenador PAN e todos os dispositivos se conectam a ele, eles podem ser RFDs ou FFDs. Em uma topologia de ponto-a-ponto todos nós se comunicam uns com os outros, não só com o coordenador PAN. Através da topologia ponto-a-ponto também é possível implementar outras formações de rede, como redes em árvore agrupadas, com vários coordenadores se comunicando com um coordenador PAN principal.

2.2 Camada de enlace

A camada de enlace tem como responsabilidade proporcionar a comunicação entre dois dispositivos, uma vez que uma rede sem fios utiliza um meio compartilhado (espectro) para propagar as suas mensagens, é necessário regular o acesso a esse meio. Isso é feito por meio de um mecanismo de regulação, o protocolo IEEE 802.15.4 usa o acesso múltiplo com verificação de portadora com anulação/prevenção de colisão (do inglês *Carrier sense multiple access with collision avoidance (CSMA/CA)*), que investiga o canal para detectar se ele está disponível antes de iniciar uma comunicação, mas também oferece suporte a agendamento determinístico em vez de CSMA/CA, neste esquema os dispositivos têm um período predeterminado para transferir uma mensagem, que é chamado período sem contenção (do inglês *Contention Free Period (CFP)*).

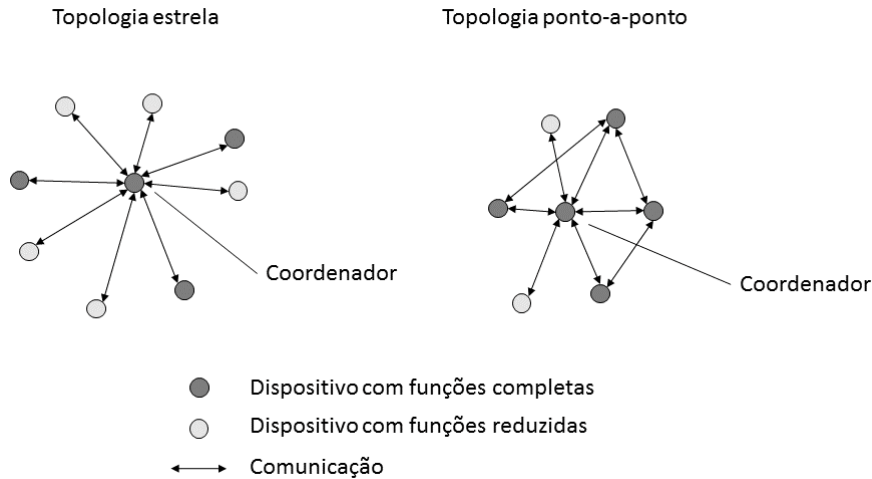


Figura 2.2: Topologias suportadas pelo protocolo IEEE 802.15.4[5]

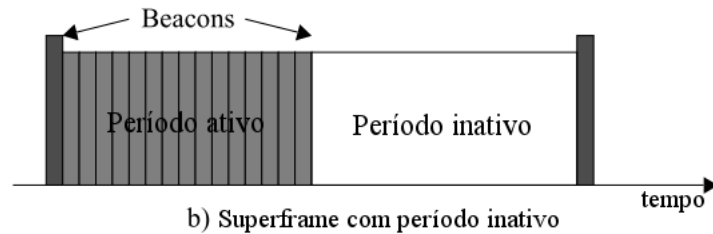


Figura 2.3: Estrutura do *superframe* do protocolo IEEE 802.15.4[5]

A fim de permitir uma ampla gama de aplicações, o protocolo define dois modos operacionais diferentes: *slotted*, em que os dispositivos são sincronizados com o coordenador e podem ter períodos pré-determinados para enviar dados, e não *slotted*, em que os dispositivos simplesmente enviam os dados quando é necessário, sem nenhum tipo de sincronização. Três outros modos de funcionamento são definidos na emenda 802.15.4e, estes são descritos em suas respectivas seções.

O modo de operação *slotted* é baseado em pacotes sinalizadores que são emitidos pelo coordenador, a fim de sincronizar todos os dispositivos associados a rede. Esse pacote define a estrutura da rede (*superframe*) que, por sua vez, define períodos ativos e inativos. Como visto na Figura 2.3 o *superframe* pode ter um período ativo e inativo, no período inativo não há atividades na rede para que haja economia de energia. Como mostrado na Figura 2.4 o período ativo pode ter um período de acesso com contenção (do inglês *Contention access period (CAP)*) e um período livre de contenção. No período CAP dispositivos utilizam um esquema CSMA/CA para acessar o meio, no período CFP é atribuído um intervalo de tempo garantido (do inglês *Guaranteed time slot (GTS)*) pelo coordenador e dispositivos designados possuem acesso exclusivo ao meio.

Também é possível que a rede não possua sincronização, em um *unslotted*, sempre que

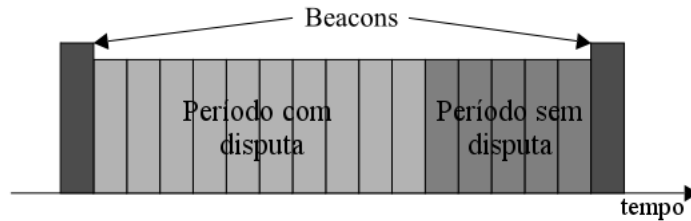


Figura 2.4: Período ativo no IEEE 802.15.4[5]

Octetos: 2	1	0/2	0/2/8	0/2	0/2/8	0/5/6/10/14	variável	2
Controle de Quadro	Número sequencial	Identificador PAN de destino	Endereço de destino	Identificador PAN de origem	Endereço de origem	Cabeçalho auxiliar de segurança	Carga do quadro	FCS
		Campos de endereçamento						
MHR							Carga da camada de enlace	MFR

Figura 2.5: Quadro definido pelo protocolo IEEE 802.15.4[5]

for necessário enviar uma mensagem, dispositivos disputam o meio usando o CSMA/CA, porém o protocolo não define quando um dispositivo deve entrar em modo de recepção, esta é uma responsabilidade camada superior.

2.2.1 Estrutura do quadro da camada de enlace

Como exibido na Figura 2.5 o quadro da camada de enlace possui os seguintes campos:

- Controle do quadro: informa o tipo de quadro, além de outros detalhes como segurança ou pedido de confirmação. Quatro tipos de quadro são suportados: Quadro de Dados, Quadro sinalizador, Quadro de confirmação e Quadro de comando;
- Número sequencial: é um identificador único do quadro;
- Campos de endereçamento: contém o endereço da origem e destino e os identificadores de rede;
- Cabeçalho auxiliar de segurança: contém informações relacionadas a segurança;
- Carga do quadro: contém informações específicas a cada tipo diferente de quadro;
- Rodapé: contém o delimitador para indicar o fim do quadro.

2.3 Camada física

A responsabilidade da camada física (do inglês *Physical layer (PHY)*) é a de transmitir o quadro da camada de enlace através do meio, isto é realizado transmitindo o pacote como

um sinal através do espectro. A fim de lidar com regulações diferentes entre países, o protocolo IEEE 802.15.4 define várias frequências diferentes (Figura 2.6), um dispositivo compatível deve suportar pelo menos uma delas. Este trabalho se concentra na banda de 2,4 GHz, utilizando sequência direta de espectro de dispersão (do inglês *direct sequence spread spectrum* - *DSSS*), DSSS é uma técnica de modulação em que o sinal gerado é espalhado através de todo o espectro utilizado pelo dispositivo [12, Chapter 13.2], ele tem algumas vantagens, tais como resistência à interferência. A banda de frequência de 2,4 GHz é dividida em 16 canais de 5 MHz cada, a norma define a possibilidade de mudar de canal, mas não especifica um esquema para fazê-lo, isso é feito pela emenda de 2012, em que é definido o salto de canais (do inglês *Channel Hopping* (*CH*)).

O padrão define também Detecção de Energia (do inglês *Energy detection* (*ED*)), que proporciona uma interface para detectar a energia presente no canal. Também proporciona a detecção de canal disponível (do inglês *Clear channel assessment* (*CCA*)), que verifica a disponibilidade do canal, ambos os recursos são utilizados pelo sistema CSMA/CA.

A camada física encapsula o quadro que vem da camada de enlace, como a unidade de dados de protocolo da camada física (do inglês *PHY protocol data unit* (*PPDU*)), a Figura 2.7 mostra o PPDU para a Modulação por deslocamento de fase em quadratura (do inglês *Offset quadrature phase-shift keying* (*O-QPSK*)), que é a utilizada na frequência de 2,4 GHz.

Os campos do PPDU são definidos da seguinte maneira:

- SHR: cabeçalho de sincronização (do inglês *Synchronization header* (*SHR*)), possibilita a identificação do fluxo de dados como um pacote 802.15.4;
- PHR: cabeçalho da camada física (do inglês *PHY header* (*PHR*)), contém o tamanho da carga;
- Carga: possui contém o quadro da camada superior, pode possuir até 127 bytes.

2.4 Emenda IEEE 802.15.4e

Em 2012 uma emenda foi lançada com a intenção de tornar o padrão adequado para mais áreas de aplicação, tais como aplicações industriais e comerciais. A fim de manter a compatibilidade com os dispositivos atuais foram propostos apenas modificações na camada de enlace.

Três novos modos operacionais são definidos nesta alteração: Redes Determinísticas de Baixa Latência (do inglês *Low latency deterministic network* (*LLDN*)), Extensão Determinística e Síncrona com Múltiplos Canais (do inglês *Deterministic and Synchronous Multi-channel Extension* (*DSME*)) e Segmentação no Tempo com Salto entre Canais (do

PHY (MHz)	Faixas de frequência (MHz)	Parâmetros de dissipação		Parâmetros de dados		
		Taxa do chip (kchip/s)	Modulação	Taxa de bits (kb/s)	Taxa de símbolos (ksymbol/s)	Símbolos
780	779–787	1000	O-QPSK	250	62.5	16-ary ortogonal
780	779–787	1000	MPSK	250	62.5	16-ary ortogonal
868/915	868–868.6	300	BPSK	20	20	Binário
	902–928	600	BPSK	40	40	Binário
868/915 (Opcional)	868–868.6	400	ASK	250	12.5	20-bit PSSS
	902–928	1600	ASK	250	50	5-bit PSSS
868/915 (Opcional)	868–868.6	400	O-QPSK	100	25	16-ary ortogonal
	902–928	1000	O-QPSK	250	62.5	16-ary ortogonal
950	950–956	—	GFSK	100	100	
950	950–956	300	BPSK	20	20	
2450 DSSS	2400–2483.5	2000	O-QPSK	250	62.5	16-ary ortogonal
UWB sub-gigahertz (Opcional)	250–750	Como definido em 14.1.1				
2450 CSS (Opcional)	2400–2483.5	Como definido em 13.2		250	167 (como definido em 13.4.2)	
		Como definido em 13.2		1000	167 (como definido em 13.4.2)	
UWB low band (Opcional)	3244–4742	Como definido em 14.1.1				
UWB high band (Opcional)	5944–10 234	Como definido em 14.1.1				

Figura 2.6: Frequências suportadas pelo protocolo IEEE 802.15.4[5]

Octetos				
1			variável	
Preâmbulo	SFD	Tamanho do Quadro (7 bits)	Reservado (1 bit)	PSDU
SHR		PHR		Carga da camada física

Figura 2.7: Quadro da camada física do protocolo IEEE 802.15.4 para a modulação O-QPSK [5]

Octetos: 1/2	0/1	0/2	0/1/2/8	0/2	0/1/2/8	0/1/5/6/1 0/14	variável		variável	2
Controle do Quadro	Número Sequencial	Identificador PAN de destino	Endereço de destino	Identificador PAN de origem	Endereço de origem	Cabeçalho auxiliar de segurança	Elementos de informação		Carga do quadro	FCS
		Campos de endereçamento					Cabe- çalhos	Carga		
MHR								Carga camada de enlace		MFR

Figura 2.8: Estrutura do quadro da emenda IEEE 802.15.4e[6]

Cabeçalho do IE	Bit: 0-6	7-14	15	Octetos: 0 ... 127
	Tamanho	ID do Elemento	Tipo = 0	Conteúdo do IE
Carga do IE	Bit: 0-10	11-14	15	Octetos: 0 ... 2047
	Tamanho	ID do Elemento	Tipo = 0	Conteúdo do IE

Figura 2.9: Formato dos Elementos de Informação[6]

inglês *Timeslotted channel hopping (TSCH)*). A estrutura do quadro (Figura 2.8) também é atualizada para suportar os novos modos operacionais e proporcionar um maior nível de flexibilidade para encapsular informações.

As próximas subseções descrevem vários novos recursos definidos pela emenda 802.15.4e.

Elemento de informação

Elementos de Informação (do inglês *Information element (IE)*) oferecem um método para encapsular informação dentro do quadro da camada de enlace, foi introduzido a fim de suportar todas as novas características de uma maneira flexível, mas também é possível incluir qualquer outra informação.

Um elemento de informação consiste de: identificador (do inglês *Identifier (ID)*), comprimento, tipo e conteúdo. Mais do que uma IE pode ser enviada e um único quadro, tanto no cabeçalho e nos campos de carga. Alguns identificadores já estão definidos pela emenda, eles são usados pelos novos recursos introduzidos. A Figura 2.9 mostra o formato de ambos os IEs de carga e de cabeçalho.

Salto de canais

Salto de canais é uma técnica usada para aumentar a resiliência da rede, dispositivos saltam entre bandas de frequência predefinidas (canais), dessa forma se há alguma interferência em um canal a próxima pode estar em melhores condições, aumentando as chances de uma transmissão bem sucedida. Para certificar-se de que os dispositivos se comunicam no mesmo canal uma lista de canais é utilizada.



Figura 2.10: Estrutura do *superframe* para redes LLDN[5]

O protocolo IEEE 802.15.4e define a lista de sequência de saltos consiste com um ID (*macHoppingSequenceID*), campos com a informação relevante à camada física, o comprimento da sequência (*macHoppingSequenceLength*), a lista atual dos canais (*macHoppingSequenceList*), e o tempo de permanência para salto sem segmentação, quando necessário. A lista de sequência pode ser configurada anteriormente, de forma aleatória ou enviada pelo coordenador. O padrão deixa a responsabilidade da mudança da lista de canais em tempo de execução para uma camada superior.

Redes Determinísticas de Baixa Latência

Redes LLDN operam em topologia de estrela. No modo LLDN um *superframes* são utilizados, como mostra a Figura 2.10, consistem de:

- LL-Beacon: usado para sincronização;
- Segmentos de gerenciamento: opcional, utilizado para configurações;
- Segmentos base: segmentos designados aos dispositivos da rede.

Em um *superframe* LLDN todos os dispositivos enviam e recebem informações apenas através do coordenador. Intervalos de tempo atribuídos a mais de um dispositivo são chamados intervalos de tempo compartilhado, dispositivos em um intervalo de tempo compartilhado disputam o canal utilizando um algoritmo especial CSMA/CA.

O LLDN também suporta Confirmação em Grupo (do inglês *Group acknowledgment (GACK)*), que permite a utilização de apenas um pacote de confirmação para indicar o resultado de todas as transmissões do *superframe*. Após o coordenador enviar o GACK todos os dispositivos com transmissões mal sucedidas realizam o reenvio no mesmo *superframe*.

Extensão Determinística e Síncrona com Múltiplos Canais

O modo DSME foi concebido como uma melhoria do modo *slotted*, melhora várias questões previamente apresentadas. É possível ter mais de sete intervalos de tempo garantidos em



Figura 2.11: Estrutura do DSME [5]

um *superframe*, comunicação com vários também é possível em intervalos garantidos, além de mudar de canal entre transmissões, todos esses recursos não estavam presentes no 802.15.4-2011.

Múltiplos *superframes* também são definidos, com este recurso o *beacon* presente entre *superframes* pode ser omitido, sendo necessário apenas uma vez, como mostrado na Figura 2.11.

Dois tipos diferentes de diversidade canal são suportados, o já discutido Salto entre Canais e Adaptação de Canal, que define o canal baseado na qualidade do mesmo (do inglês *Link Quality Indicator (LQI)*) durante o período de alocação.

Segmentação no Tempo com Salto entre Canais

TSCH é o objeto de estudo deste trabalho, ele foi implementado em uma ferramenta de simulação de rede a fim de permitir um cenário de simulação que destaque os benefícios da aplicação de técnicas de autoaprendizagem. TSCH é um modo de operação determinístico, todas as comunicações são previamente agendadas durante a formação da rede ou configuradas anteriormente nos dispositivos, a emenda 802.15.4e não define um esquema para alterar sua configuração em tempo de execução, deixando essa responsabilidade para camadas superiores. Um *slotframe* é a agenda de transmissões de todos os dispositivos da rede que se repete no tempo, cada *slotframe* é composto por intervalos de tempo, que contém uma ligação entre dois dispositivos. A Figura 2.12 ilustra um *slotframe* configurado com três intervalos de tempo: o primeiro intervalo de tempo tem uma ligação entre os nós A e C, o segundo intervalo de tempo tem uma ligação entre os nós B e C e o terceiro intervalo de tempo está vazio.

Um único intervalo de tempo (do inglês *timeslot*) deve ser longo o suficiente para garantir uma transmissão e uma confirmação. A Figura 2.13 mostra os componentes de *timeslot*. O transmissor aguarda *macTsCCAOffset*, e realiza o CCA (se ativo). Se o CCA retorna um meio ocupado o nó encerra suas atividades durante o *timeslot* atual e tenta enviar o mesmo pacote no próximo *timeslot*. Se o CCA retorna um meio livre ou o CCA está desabilitado, o dispositivo começa a transmissão do quadro. Se uma confirmação é solicitada o nó aguarda *macTsAckDelay* e liga o rádio em modo de escuta, se um quadro

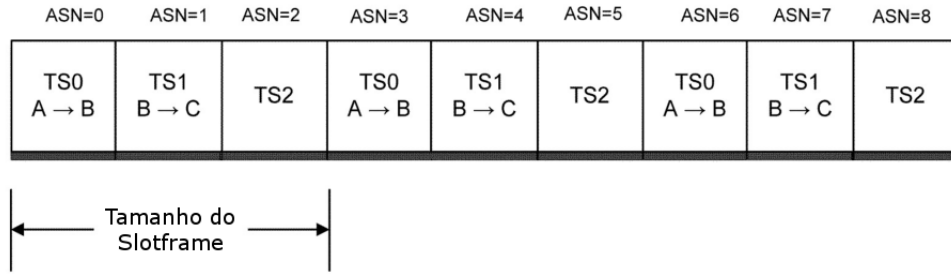


Figura 2.12: Estrutura do *slotframe* TSCH[5]

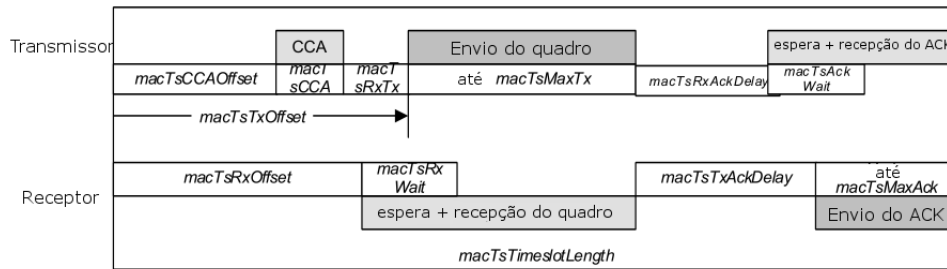


Figura 2.13: Estrutura do *timeslot* TSCH[5]

de confirmação é recebido durante o intervalo *macTsAckWait* então a comunicação é considerada bem sucedida, se nenhuma confirmação é recebida a transmissão é considerada falha. O dispositivo receptor espera o intervalo *macTxRxOffset* e, em seguida, liga o seu rádio em modo de escuta por um intervalo *macTxRxWait*, se um quadro não detectado durante este período o receptor fica inativo até seu próximo *timeslot*. Caso contrário, depois de receber o quadro o dispositivo aguarda *macTsTxAckDelay* e envia a confirmação, caso necessário.

Ligações (do inglês *links*) compartilhadas possuem mais de um dispositivo designado como transmissor, neste caso, podem ocorrer colisões. Para evitar colisões, os nós em um *link* compartilhado executam um algoritmo CSMA/CA. A Figura 2.14 mostra uma configuração de *slotframe* possuindo um *link* compartilhado no primeiro *timeslot*, neste caso, os nós A e C irão competir pelo canal antes da transmissão.

Uma lista de canais a serem utilizados para mudar de canal entre comunicações, de acordo com um contador global chamado Número Absoluto do Segmento (do inglês *Absolute Slot Number (ASN)*):

$$CH = Lista[(ASN + CO) \% TamanhoDaLista] \quad (2.1)$$

Onde *CH* é o canal atual, *CO* é o deslocamento definido durante a configuração e *TamanhoDaLista* é tamanho da lista de canais a serem utilizados.

Channel offset	0	A → B C → E	C → D			A → B C → E	C → D		
	1			E → A				E → A	
	2				B → D				B → D
	3								
		0	1	2	3	4	5	6	7
		ASN							

Figura 2.14: Salto entre canais[5]

	ASN=0	ASN=1	ASN=2	ASN=3	ASN=4	ASN=5	ASN=6	ASN=7	
Slotframe 1 5 slots	TS 0	TS 1	TS 2	TS 3	TS 4	TS 0	TS 1	TS 2	...
Slotframe 2 3 slots	TS 0	TS 1	TS 2	TS 0	TS 1	TS 2	TS 0	TS 1	...

Figura 2.15: Múltiplos *slotframes* [5]

A utilização do salto de canais faz com que seja possível a coexistência de simultâneas configurações de *slotframe*, cada nó com um deslocamento de canal diferente. A Figura 2.15 mostra uma configuração simples de múltiplos *slotframes*, neste exemplo o *slotframe* 1 possui 5 *timeslots*, e o *slotframe* 2 possui 3 *timeslots*. Múltiplos *slotframes* podem ser utilizados para definir diferentes esquemas de comunicação para diferentes grupos de dispositivos. O uso do deslocamento de canal permite que diferentes canais possam ser utilizados no mesmo ASN, isto permite que múltiplos *slotframes* ocorram simultaneamente sem colisões ou interferências. Embora cada *slotframe* possua diferentes ligações e quantidade de *timeslots*, todos eles possuem a mesma configuração de temporização. A Figura 2.15 mostra como todos os *timeslots* são sincronizados.

2.5 Resumo do capítulo

Neste capítulo o protocolo IEEE 802.15.4 é introduzido, os modos de operação *slotted* e *unslotted* são apresentados. O modo *slotted* sincroniza todos os nós da rede, oferecendo acesso exclusivo ao meio (CFP) ou acesso livre (CAP), em que os nós disputam pelo meio utilizando um algoritmo CSMA/CA. No modo *unslotted* os nós disputam pelo meio sempre que necessário utilizando um algoritmo CSMA/CA.

Os modos de operação incluídos pela emenda 802.15.4e também são apresentados: o DSME atualiza o modo *slotted*; o LLDN possui baixa latência através do uso de *superframes* em que os nós se comunicam apenas com a central; e por fim o TSCH, objeto

de estudo deste trabalho, apresenta um funcionamento de alta confiabilidade através da definição de *timeslots* com *links* de transmissão ou recepção previamente definidos.

Capítulo 3

Estado da arte

Neste capítulo, o estado da arte para esta tese será abordado, a primeira seção apresenta simulação de rede e abrange simuladores de rede que suportam o padrão 802.15.4, a segunda seção compara dois dos principais simuladores de rede usados para a pesquisa e a terceira seção apresenta autoaprendizagem em redes de sensores sem fio.

3.1 Simulação de redes

A fim de assegurar um funcionamento correto, é essencial a qualificação e avaliação de uma rede. As redes se tornaram altamente complexas e cresceram muito ao longo das últimas décadas e se tornou muito difícil de analisá-las em um ambiente real [10], é aí onde simuladores de rede se tornam uma grande ajuda. A simulação de rede permite a análise de uma rede sem ter de implantá-la em um ambiente real, reduzindo custos e aumentando os cenários possíveis.

Os ambientes mais comuns de simulação de rede usam métodos de evento discreto para simular uma rede. Evento discreto significa que o simulador apenas processa o tempo quando existe alguma atividade de interesse acontecendo, por exemplo, na transmissão de um simples pacote o simulador considerará apenas o tempo de transmissão dos pacotes quando começa e termina, o tempo entre este intervalo não é processado, o nível de detalhe do que foi simulado muda de acordo com a finalidade da simulação.

Simulações de rede podem ser usadas para vários aspectos de pesquisa e desenvolvimento, tais como a análise do ambiente real ou avaliação de uma proposta de um novo modelo de rede. Nesta tese a finalidade do uso de simulações de rede é a melhoria e análise de protocolos.

A primeira versão do IEEE 802.15.4 foi lançada em 2003, desde então ele tem sido coberto por uma grande quantidade de simuladores de rede com diferentes níveis de co-

bertura. Simuladores de rede podem ser classificados em duas categorias: proprietários ou de código aberto.

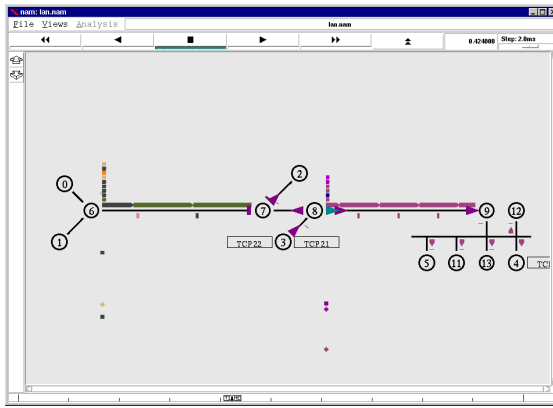
Simuladores proprietários muitas vezes oferecem suporte completo ao padrão e uma interface gráfica de usuário (do inglês *Graphical user interface (GUI)*), eles também oferecem suporte através de uma taxa e são direcionados para uso comercial. A sua natureza de código fechado muitas vezes dificulta modificações. Alguns exemplos de simuladores proprietários bem conhecidos que suportam o padrão 802.15.4 padrão são Qualnet[19] e NetSim[22].

Simuladores de rede de código aberto são geralmente mantidos pela comunidade acadêmica e não oferecem um apoio profissional, mas sim um fórum de discussão onde a comunidade pode perguntar e responder as questões uns dos outros. Sua natureza aberta facilita modificações e, portanto, é uma opção adequada de simulador para pesquisadores. Um lado negativo é que as contribuições estão frequentemente espalhadas nos próprios repositórios do pesquisador e é penoso reutilizar essas contribuições. Um simulador de rede de código aberto bem conhecido e que suporta 802.15.4 é o *Network-Simulator 3 (ns-3)*[18], um simulador de rede mantido pela comunidade acadêmica que suporta vários padrões de rede. As primeiras versões do simulador de rede datam de 1995, sua segunda versão, chamada ns-2 (do inglês *Network-Simulator 2*)[17], foi lançada em 1996. Mesmo contendo uma arquitetura considerada defasada ele ainda é utilizado por muitos pesquisadores devido ao seu suporte à diversos padrões de rede. O ns-3[18] é a versão mais recente desse simulador e foi criada para atualizar a estrutura do ns-2. O ns-3 introduz uma nova interface, quebrando a compatibilidade com os modelos do ns-2 e portanto ele não herdou a biblioteca de modelos. O modelo do IEEE 802.15.4 no ns-3 está atualmente em desenvolvimento. Embora o simulador de rede ns-3 seja bem conhecido dentre a comunidade acadêmica, ele é relativamente novo para um projeto de código aberto, mas seu uso ainda está ganhando força. Pesquisadores estão parando de usar sua versão anterior, ns-2, e começando a adotar esta nova plataforma com vários benefícios, como uma arquitetura atualizada.

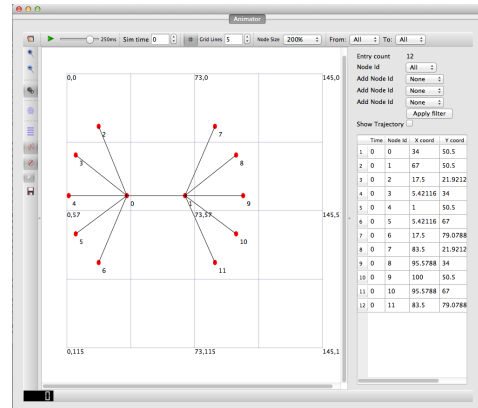
Em relação à emenda IEEE 802.15.4e, não há nenhum simulador de rede conhecido que o suporte no momento, isso é devido a sua data de lançamento recente. Qualquer estudo que precisa de uma ferramenta de simulação tem que executar o seu próprio simulador.

3.2 Comparação entre ns-2 e ns-3

O ns-2 é amplamente utilizado pela comunidade acadêmica, devido à sua ampla gama de protocolos suportados. Sua última versão, ns-3, foi criada para atualizar a arquitetura antiga do ns-2, que apresenta várias desvantagens, tais como o uso da linguagem OTcl[21]



ferramenta de visualização
do ns-2 (nam)



ferramenta de visualização
do ns-3 (NetAnim)

Figura 3.1: Ferramentas de visualização do ns-2 e ns-3

para descrever simulações de rede e C++ para implementação do modelo. Este esquema foi escolhido para evitar a recompilação de código C++. A interface entre essas duas línguas é complexa e a sobrecarga traz problemas de desempenho em simulações em grande escala. Já que o tempo de compilação não é um problema em computadores modernos, o ns-3 utiliza apenas C++ tanto para o desenvolvimento de modelos quanto para a descrição de simulações. Este esquema facilita o desenvolvimento de simulações e oferece mais liberdade para o desenvolvedor. Além de eliminar o impacto de desempenho decorrentes do OTcl, o ns-3 também suporta *python* para escrever *scripts* de simulação e protótipos de modelos.

O ns-3 utiliza intensivamente recursos de orientação a objetos do C++ como herança e polimorfismo. Os modelos básicos podem ser facilmente modificados e ampliados. O gerenciamento de memória é muito melhor no ns-3 uma vez que ele automaticamente desaloca objetos como pacotes, o que também melhora o seu desempenho para simulações de grande escala. O ns-2, por outro lado, não libera a memória usada por pacotes anteriores, apenas à reutiliza.

Como pode ser visto na Figura 3.1, ambos oferecem uma interface gráfica simples, que suporta apenas a visualização. Simulações são construídas usando editores de texto comuns.

Embora existam vários simuladores com suporte para o IEEE 802.15.4, o ns-3 é uma excelente opção para o uso nesta tese. A sua inclinação acadêmica, com foco na modificação do modelo, o torna muito atraente para os pesquisadores. Uma vez que as duas versões não têm qualquer suporte para a emenda IEEE 802.15.4e, o estado prematuro do 802.15.4 no ns-3 tem pouco impacto na escolha de um simulador uma vez que apenas a camada física do 802.15.4 é utilizada. A estrutura moderna do ns-3 faz com que seja uma excelente escolha para este trabalho.

3.3 Situação do modelo IEEE 802.15.4 no ns-3

A principal razão que impede que os pesquisadores adotem o ns-3 é a sua falta de padrões compatíveis. Entre os modelos não totalmente suportados pelo ns-3 está o IEEE 802.15.4. Seu desenvolvimento data de 2006, mas ele ainda não foi concluído. Este trabalho tenta melhorar o estado de implementação atual no ns-3, incluindo suporte para o modo operacional TSCH na emenda IEEE 802.15.4e.

Um entendimento mais profundo sobre a implementação atual é essencial para permitir o desenvolvimento do modo operacional TSCH. A camada de enlace, além de oferecer serviços específicos para os modos *slotted* e *unslotted*, também fornece serviços que são utilizados por todos os modos, incluindo os definidos pela emenda. Esses serviços precisam ser modificados a fim de suportar o TSCH.

O IEEE 802.15.4 define dois modos de funcionamento: *slotted* e *unslotted*. A implementação atual suporta apenas o modo operacional *unslotted*. Embora suportando apenas o modo *unslotted*, ambos os esquemas de CSMA/CA já estão implementados. Suporte para a formação da rede também é parcial, uma vez que a funcionalidade de sinalizador não está presente. Os recursos de segurança também não são suportados.

Dois tipos de serviços são definidos pela camada de enlace: Serviço de Dados (do inglês *MAC Data Service (MCPS)*) e Serviço de Gerenciamento (do inglês *MAC Management Service (MLME)*). O MCPS é a interface com a camada superior que faz a transmissão, recepção e expurgo de pacotes. O MLME fornece comandos relacionados ao gerenciamento de recursos, como associação, a mudança de parâmetros de configuração, sincronização, etc. A Tabela 3.1 lista todos os serviços definidos pelo padrão IEEE 802.15.4 e seu nível de suporte na implementação atual.

Como se pode verificar na Tabela 3.1 maioria dos serviços não são necessários em uma rede TSCH, o que torna o impacto atual da implementação no desenvolvimento TSCH menos importante. No entanto, existem algumas características da camada de enlace que são usados em TSCH que possuem alguns problemas, tais como o sistema de reconhecimento que não apresenta um comportamento adequado. Neste trabalho os serviços necessários ao TSCH foram atualizados para suportar tal modo e os novos serviços introduzidos na emenda 802.15.4e foram implementadas conforme necessário para permitir o cenário de simulação proposto.

Na camada física apenas a banda de frequência de 2,4 GHz é totalmente suportada. Assim como a camada de enlace, a camada física define um serviço para transmitir e receber dados (do inglês *PHY Data service (PD)*) e um serviço de gestão (do inglês *PHY Management Service (PLME)*) responsável pelo CCA e esquemas de ED, configurar e obter atributos da camada física e alterar o estado do rádio. A Tabela 3.2 lista todos os serviços prestados pela camada física e seu atual nível de suporte.

Nome	Situação	Utilizado pelo TSCH
MCPS-DATA	Parcial	Sim
MCPS-PURGE	Inexistente	Sim
MLME-ASSOCIATE	Parcial	Não
MLME-DISASSOCIATE	Parcial	Não
MLME-BEACON-NOTIFY	Inexistente	Sim
MLME-GET	Completo	Não
MLME-GTS	Inexistente	Não
MLME-ORPHAN	Parcial	Não
MLME-RESET	Completo	Não
MLME-RX-ENABLE	Completo	Não
MLME-SCAN	Parcial	Sim
MLME-COMM-STATUS	Completo	Não
MLME-SET	Completo	Não
MLME-START	Parcial	Não
MLME-SYNC	Inexistente	Não
MLME-SYNC-LOSS	Inexistente	Não
MLME-POLL	Parcial	Não

Tabela 3.1: Lista dos serviços da camada de enlace do padrão IEEE 802.15.4-2006, a situação atual de suporte no ns-3 e os serviços utilizados pelo TSCH.

Nome	Situação	Utilizado pelo TSCH
PD-DATA	Parcial	Sim
PLME-CCA	Completo	Sim
PLME-ED	Completo	Sim
PLME-GET	Completo	Sim
PLME-SET-TRX-STATE	Completo	Sim
PLME-SET	Completo	Sim

Tabela 3.2: Lista dos serviços da camada física do padrão IEEE 802.15.4-2006, a situação atual de suporte no ns-3 e os serviços utilizados pelo TSCH.

Como pode ser visto na Tabela 3.2 a situação atual de implementação da camada física é muito mais avançada do que da camada de enlace. Uma vez que a alteração não atualiza a camada física, todos os seus serviços são utilizados pelo TSCH e nenhuma modificação é necessária para suportá-lo. Embora a maioria dos serviços sejam suportados completamente, faltam testes para assegurar o correto funcionamento e a completude de funcionalidades. O recurso ausente no serviço PD-DATA é o LQI, ou seja, não há uma maneira de medir a qualidade do canal.

O suporte à interferência é muito primitivo na implementação atual. Tal característica é essencial para o cenário de simulação proposto e teve de ser atualizado neste trabalho. Durante a realização de testes, problemas foram encontrados em receber dois pacotes ao mesmo tempo e alterar o estado do rádio. Também não há suporte para a mudança do canal de comunicação em tempo de execução.

A implementação atual do padrão 802.15.4, embora ainda em desenvolvimento, fornece um grande embasamento para este trabalho.

3.4 Autoaprendizagem em redes de sensores sem fio

Redes de sensores sem fio têm uma gama de aplicação significativa. Um problema comum em muitas áreas é como lidar com as mudanças que ocorrem no ambiente ou na própria aplicação. Tais alterações podem causar um grande impacto sobre o desempenho da rede, por exemplo, uma alteração no ambiente muito comum é a interferência no canal, variando de muito baixa, sem qualquer impacto aparente, a muito elevada, causando a perda completa da comunicação.

Em situações dinâmicas é importante que a rede possa se adaptar às variações apresentadas. Com autoaprendizagem a rede adapta-se automaticamente, sem intervenção de um administrador de rede. De acordo com o modelo OSI[15], a pilha de rede tem sete camadas. A autoaprendizagem pode ser aplicada em qualquer camada, e de fato é, em diversos padrões, como o Protocolo de Controle e Transmissão (do inglês *Transmission Control Protocol (TCP)*)[1], a camada de transporte da Internet, que adapta o fluxo de comunicação de acordo com a velocidade de transmissão dos dispositivos.

O padrão 802.15.4 não define nenhum tipo de autoaprendizagem, embora o LQI possa ser utilizado para tal tarefa, o padrão não faz qualquer uso do mesmo, apenas fornece o valor à camada superior. A emenda IEEE 802.15.4e define dois tipos diferentes de autoaprendizagem: adaptação de acordo com a qualidade do espectro, saltos de canal, em que os dispositivos usam uma lista predefinida de canais e "saltam" entre os canais de comunicação (neste caso, não há adaptação, porque a lista de canais é pré-definida, mas ocorre uma melhora na qualidade da comunicação, pois ocorrendo degradação em

algum canal outros serão utilizados), e adaptação de canal, disponível apenas para o modo DSME, em que os canais usados em segmentos GTS podem ser definidos de acordo com o LQI.

Vários trabalhos tentaram resolver esta questão. O Self-Adapting MAC Layer (SAML)[20] propõe um esquema para mudar entre protocolos da camada de enlace em tempo de execução, de acordo com as prioridades definidas pelo estado da aplicação e do ambiente. Tal esquema tem a vantagem de se adaptar a cenários completamente diferentes, desde que a camada de enlace mude completamente. As desvantagens são que o esquema de mudar a camada de enlace adiciona alguma sobrecarga e dispositivos consomem mais memória e processamento. O Adaptive Time Slotted Channel Hopping (A-TSCH)[9] estende o modo TSCH do 802.15.4e para alterar a lista de canais que são utilizados de acordo com a qualidade dos canais. Este trabalho mostrou melhorias que variam de 5,3 % para 8,1 % na taxa de entrega de pacotes.

3.5 Resumo do capítulo

Neste capítulo o estado da arte avançado por este trabalho é apresentado, utilizando critérios como facilidade de implementação e adesão pela comunidade acadêmica o simulador de redes ns-3 foi escolhido para implementar o modo TSCH. A implementação atual do protocolo 802.15.4 no ns-3 é apresentada, é constatado que ela está em estado de desenvolvimento, não oferecendo suporte a muitas funcionalidades.

A situação atual de técnicas de autoaprendizagem em redes de sensores também é explorada, é observado que o protocolo IEEE 802.15.4 não introduz qualquer tipo de autoaprendizagem. Dois trabalhos que tentam resolver este problema são apresentados, o SAML propõe um esquema em que de acordo com mudanças no ambiente os dispositivos da rede mudam completamente sua forma de funcionamento, já o A-TSCH propõe uma técnica para gerenciar a lista de salto de canais de forma a utilizar apenas canais com qualidade aceitável para comunicação.

Capítulo 4

O Simulador ns-3

Primeiramente nesse capítulo teremos uma visão geral da arquitetura do ns-3, focando nos recursos utilizados na implementação do TSCH. A segunda seção explica o modelo *Spectrum*, usado pelo ns-3 para representar o meio. A terceira seção detalha a atual implementação do 802.15.4, suas limitações e funcionalidades presentes, finalmente, a quarta seção apresenta o principal produto deste trabalho, o modelo TSCH para ns-3, incluindo a descrição do modelo de energia usado para prever o consumo de energia de uma rede TSCH.

4.1 Arquitetura do ns-3

A arquitetura do ns-3 é altamente flexível e modular, o que difere de outros simuladores que oferecem um ambiente de simulação fechado, em que tudo é feito em uma GUI. O ns-3 oferece um conjunto de modelos que podem ser combinados com outras bibliotecas e softwares externos para visualização de dados ou análise. O ns-3 é usado principalmente com Linux e Windows (através do Cygwin[14], que é um ambiente Unix para Windows), mas existem maneiras de executá-lo em Mac OS.

O simulador ns-3 fornece uma biblioteca central que oferece várias classes usadas extensivamente em outros modelos, esta biblioteca central fornece recursos como gerador de números pseudoaleatórios, o que é crucial para a simulação da rede, registros de saída em vários formatos e muitos outros recursos.

Como mostrado na Figura 4.1, uma abstração chave usada pelo ns-3 é a classe *Node*, que representa um dispositivo comum. Semelhante a dispositivos reais, um *Node* pode ter várias aplicações em placas de rede que suportam diferentes protocolos, tais como um laptop que normalmente fornece redes com fio e sem fio.

Um *Node* pode conter uma ou mais aplicações. A classe *Applications* é utilizada para gerar tráfego, é uma classe abstrata que tem de ser especializada para simular uma apli-

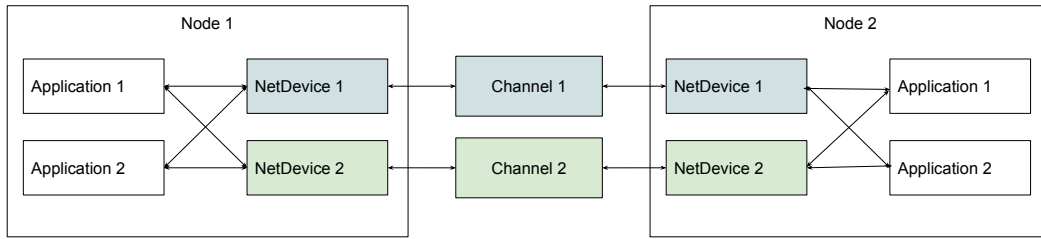


Figura 4.1: Estrutura da classe Node do ns-3

cação, no entanto, o ns-3 já proporciona algumas aplicações que são comuns na simulação de redes, tais como *UdpEchoClient* e *UdpEchoServer*, que simulam um servidor e cliente, transferindo dados entre eles. Neste trabalho aplicações não são utilizadas uma vez que o foco é nas camadas mais baixas, ao invés disso os pacotes são criados manualmente.

A ligação entre os dispositivos é modelada pela classe *Channel*, que cria o meio no qual a informação é transmitida. Esta classe pode ser especializada de acordo com a tecnologia a ser utilizada. O ns-3 já oferece algumas especializações, tais como o *WifiChannel*, que simula o espectro de comum de um dispositivo Wi-Fi e o *PointToPointChannel*, que simula um cabo. O modelo *SpectrumChannel* é o utilizado neste trabalho, ele modela um espectro dependente de frequência. A seção 4.2 explica em detalhes como ele funciona. Os canais são criados em modo global, assim todos os dispositivos utilizam o mesmo canal ao transferir dados.

A classe *NetDevice* faz a abstração de uma placa de rede. Um Nó pode ter mais do que um *NetDevice*, tal como um computador pode ter mais do que uma placa de rede. Cada *NetDevice* também pode ser ligado a um canal diferente. A classe *NetDevice* pode ser especializada de acordo com a tecnologia. O ns-3 já oferece algumas especializações que são comumente usadas em simulações de rede, como *WifiNetDevice*, que implementa as camadas física e de enlace de um dispositivo Wi-Fi. *NetDevices* são geralmente compostas pela camada física e camadas de enlace, mas tal implementação não é imposta pelo ns-3, são os desenvolvedores que escolhem como implementar sua *NetDevice*. Este trabalho utiliza a classe *LrWpanNetDevice*, que implementa um dispositivo que suporta o padrão IEEE 802.15.4, este modelo é descrito na seção 4.3.

De acordo com a modelagem do ns-3, três modelos principais são usados para dar forma a uma topologia de rede, *Nodes*, *NetDevices* e *Channels*. O trabalho de criação e configuração de uma simulação com muitos *Nodes* e *NetDevices* pode exigir várias operações. Para facilitar tais tarefas, estão disponíveis classes que auxiliam a criação de topologias de rede. Essas classes costumam realizar tarefas básicas que são necessárias para criar uma rede, tornando a simulação mais simples, por exemplo, classes que auxiliam a criação da camada de enlace costumam prover um método para criação automática de endereços MAC para todos os dispositivos.

4.2 O modelo de espectro

O padrão IEEE 802.15.4 adota Sequência Direta de Espalhamento do Espectro (do inglês *Direct-Sequence Spread Spectrum (DSSS)*), o qual espalha o sinal através de várias faixas de frequências, cada faixa pode sofrer impactos diferentes pelo canal[12]. Para simular esse comportamento é necessário representar o canal como dependente da frequência. O modelo *Spectrum* do ns-3 [7] foi desenvolvido para permitir uma representação do espectro com diferenciação das frequências.

Um sinal eletromagnético pode ser representado como uma função de tempo ou frequência, simuladores frequentemente representam um sinal com um único pulso que é constante durante toda a duração da transmissão. O modelo de espectro do ns-3 faz o mesmo, mas o pulso não é representado como um valor único, mas como um vetor, cada componente com o seu próprio valor. A densidade espectral (do inglês *Power Spectral Density (PSD)*) [8] representa a distribuição de energia em uma banda de frequência. Geralmente em *Watts*, é utilizada como a representação do domínio da frequência, isso permite uma representação do canal razoavelmente precisa com baixa complexidade. Neste modelo, cada banda de frequência tem um PSD constante. O tamanho de cada banda, ou o número de bandas usadas para representar uma gama de frequências está diretamente relacionada com a precisão de representação do modelo. Quanto maior o número de bandas, mais precisamente o espectro é representado, no entanto também é maior a carga computacional exigida, impactando diretamente o tempo de simulação.

Um conversor de modelo de espectro está disponível para permitir que cada modelo implemente seu próprio modelo de espectro e use a configuração de distribuição de frequência mais adequada. Em uma simulação em que é usada mais de uma tecnologia, por exemplo, Wi-Fi e LR-WPAN, o modelo de espectro é convertido facilmente entre ambas as configurações. A operação de conversão entre dois modelos diferentes de espectro é definida como: seja $[f_{i,l}, f_{i,h}]$ e $[g_{j,l}, g_{j,h}]$ duas bandas de frequência de dois diferentes modelos *Spectrum* F e G, respectivamente, a matriz de conversão $S = s_{i,j}$ em que os coeficientes $s_{i,j}$ são determinados como:

$$s_{i,j} = \min \left(1, \frac{\max(0, \min(f_{i,h}, g_{j,h}) - \max(f_{i,l}, f_{j,l}))}{f_{j,h} - g_{j,l}} \right) \quad (4.1)$$

Essa definição permite a conversão de qualquer modelo de espectro $a \in S1$ para uma representação correspondente $b \in S2$ como a matriz transposta de $a * C$.

A Figura 4.2 mostra a implementação do modelo de espectro. A representação das bandas de frequência é feita pela classe *BandInfo*, a qual contém o limite superior, o limite inferior e a frequência central. Um modelo de espectro contém um vetor de *BandInfo* (*SetOfBands*), e vários outros métodos para facilitar o seu uso, como construtores que

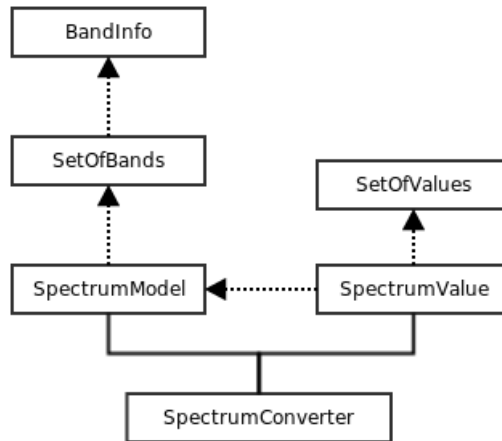


Figura 4.2: Arquitetura do modelo *Spectrum* do ns-3

precisam de um *SetOfBands* ou um vetor de frequências centrais como argumento para instanciá-lo.

A classe *SpectrumValue* representa um espaço funcional num modelo de espectro, que contém uma referência para o modelo de espectro e um vetor de valores que representa o domínio de uma função em particular em função do espaço. *SpectrumValues* são usados para representar PSDs, para permitir sua manipulação várias operações matemáticas de vetor são definidas. Um objeto *SpectrumValue* é gerado para cada transmissão e é passado para todos os dispositivos conectados ao mesmo canal, que por sua vez, devem empregar um tratamento específico para determinar se o PSD recebido representa uma transmissão bem-sucedida ou não.

A Classe *SpectrumConverter* implementa a equação 4.1, são necessárias duas instâncias do modelo de espectro, após sua inicialização o método *convert* converte um *SpectrumValue* de um determinado modelo de espectro em um *SpectrumValue* de um outro modelo de espectro.

A Figura 4.3 mostra os modelos *SpectrumPhy* e *SpectrumChannel*, juntamente com *SpectrumModel* é realizada a representação completa de um canal sem fios dependente da frequência e da sua interação com dispositivos físicos. As classes mostradas são abstratas, o que significa que só oferecem uma interface, sua aplicação efetiva deve ser feita pelo modelo que utiliza essas classes. Se um modelo tem um dispositivo que interage com um *SpectrumChannel* ele precisa herdar a classe *SpectrumPhy* e implementar todos os seus métodos, o mesmo acontece se um modelo precisa implementar um *SpectrumChannel*.

Um *SpectrumChannel* deve ser instanciado apenas uma vez, e todos os dispositivos que desejam usar este canal devem se registrar usando o método *AddRx*. Uma vez que o dispositivo está associado, o método *SpectrumChannel::StartTx* deve ser utilizado para

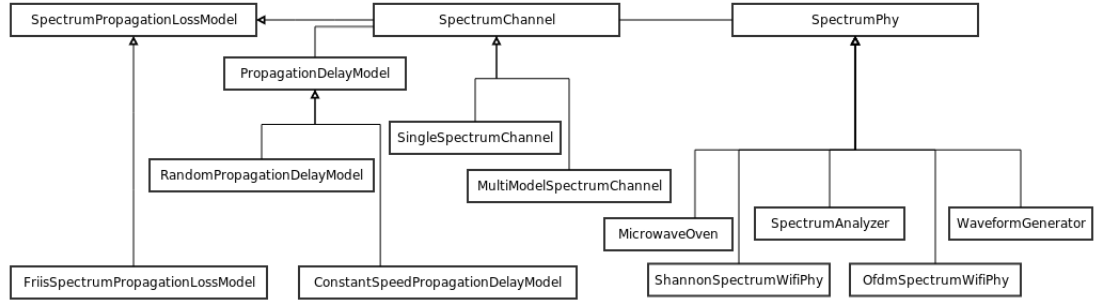


Figura 4.3: Arquiteturas do *SpectrumChannel* e camada física do ns-3

enviar um sinal através do novo canal, este método tem como argumentos um objeto *SpectrumValue*, que representa o sinal, a duração da transmissão e o pacote que está sendo transmitido. Após a recepção de um sinal o *SpectrumChannel* notifica todos os outros dispositivos associados, através do método *SpectrumPhy::StartRx*, utilizando os mesmos parâmetros associados com a transmissão. A implementação do método *SpectrumPhy::StartRx* é específica ao dispositivo, geralmente possui a tarefa de definir se o *SpectrumValue* recebido representa uma transmissão bem-sucedida ou não.

A classe *SpectrumInterference* é fornecida para modelar a interferência, juntamente com a classe *SpectrumErrorModel* uma modelagem de interferência completa é alcançada. A classe *SpectrumInterference* modela um canal com ruído gaussiano branco aditivo (do inglês *Additive White Gaussian Noise (AWGN)*), o qual define que todos os outros sinais são vistos como ruído. A relação entre sinal e interferência com ruído (do inglês *Signal-to-interference-plus-noise ratio (SINR)*) é definida como:

$$SINR = \frac{SinalRx}{TodosSinais} \quad (4.2)$$

Onde *SinalRx* é o sinal de recepção e *TodosSinais* é a somatória de todos os outros sinais.

Quando um sinal é recebido pela camada física ele é armazenado. Através da sua duração, juntamente com todos os outros sinais recebidos, quando um sinal é identificado como um pacote válido o método *StartRx* inicia o processo de avaliação do canal quando houver mudança (um sinal deixa de ser transmitido ou um novo sinal aparece). Essa avaliação é realizada pela classe *SpectrumErrorModel*, que implementa um modelo de erro específico de acordo com a tecnologia utilizada.

A classe *SpectrumModel* possui duas variantes: *SingleSpectrumModel*, que usa apenas um modelo de espectro, e *MultiModelSpectrumChannel*, o que permite a utilização de mais de um modelo de espectro, a conversão é feita continuamente sempre que for necessária.

A classe *SpectrumChannel* também tem modelos de propagação para simular a perda

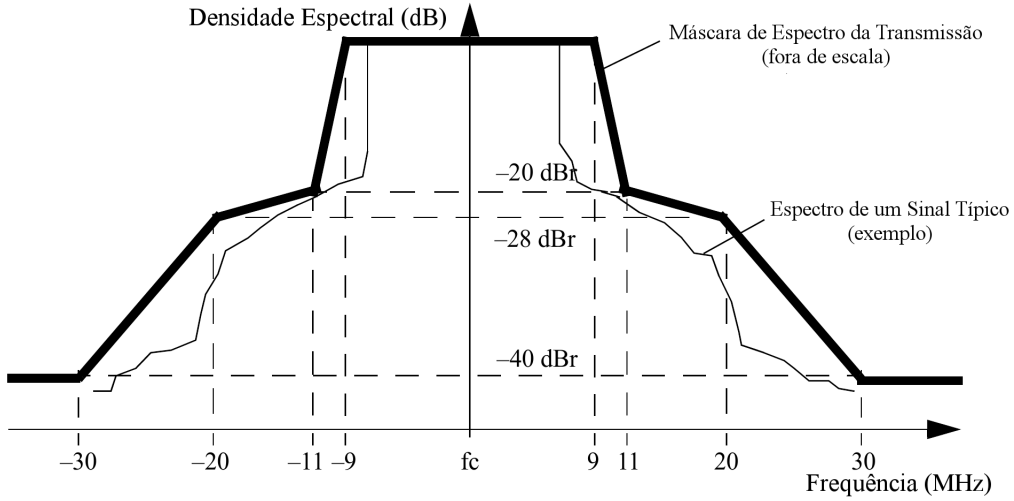


Figura 4.4: Máscara do espectro da transmissão do padrão IEEE 802.11[2]

e atraso de um sinal durante a sua transmissão, que é fornecida pelas classes *SpectrumPropagationLossModel* e *PropagationDelayModel*, respectivamente. A classe *PropagationDelayModel* pertence ao núcleo do ns-3 e vários modelos de atraso de propagação já estão disponíveis, tais como *RandomPropagationDelayModel*, que retorna um valor aleatório de um intervalo predefinido, e *ConstantSpeedPropagationDelayModel*, que calcula o atraso de acordo com a distância entre os dispositivos e uma velocidade pré-definida.

O *SpectrumPropagationLossModel* fornece uma interface para permitir modelos de perda de propagação dependente da frequência, o modelo de propagação Friis[16] já está implementado, e define que para a frequência de uma determinada transmissão f , em Hz, e a distância d , em metros, a razão entre a potência do receptor, P_r , e do transmissor P_t é,

$$\frac{P_r}{P_t} = \left(\frac{4 * \pi * f * d}{c} \right)^2, \quad (4.3)$$

Onde c é a velocidade da luz, utilizada como $3 * 10^8$ m/s.

Algumas especializações muito úteis do *SpectrumPhy* já são fornecidas pelo ns-3: A classe *WaveGenerator* fornece uma interface para transmitir um determinado PSD através de um canal, com um período predefinido e ciclo. É muito útil para simular interferência. A classe *SpectrumAnalyzer* sonda o canal em uma determinada frequência e salva o PSD do canal para um arquivo, muito útil para analisar os resultados da simulação.

Para simular a interferência de um dispositivo Wi-Fi a classe *WifiSpectrumValue5MhzFactory* é utilizada neste trabalho, ela gera um PSD de acordo com a máscara de espectro definida de um dispositivo 802.11g[2]. A largura de um canal de banda Wi-Fi é de 22 MHz, tem 11 canais na faixa de frequência de 2.4 GHz, mas, como mostrado na Figura 4.4, sua

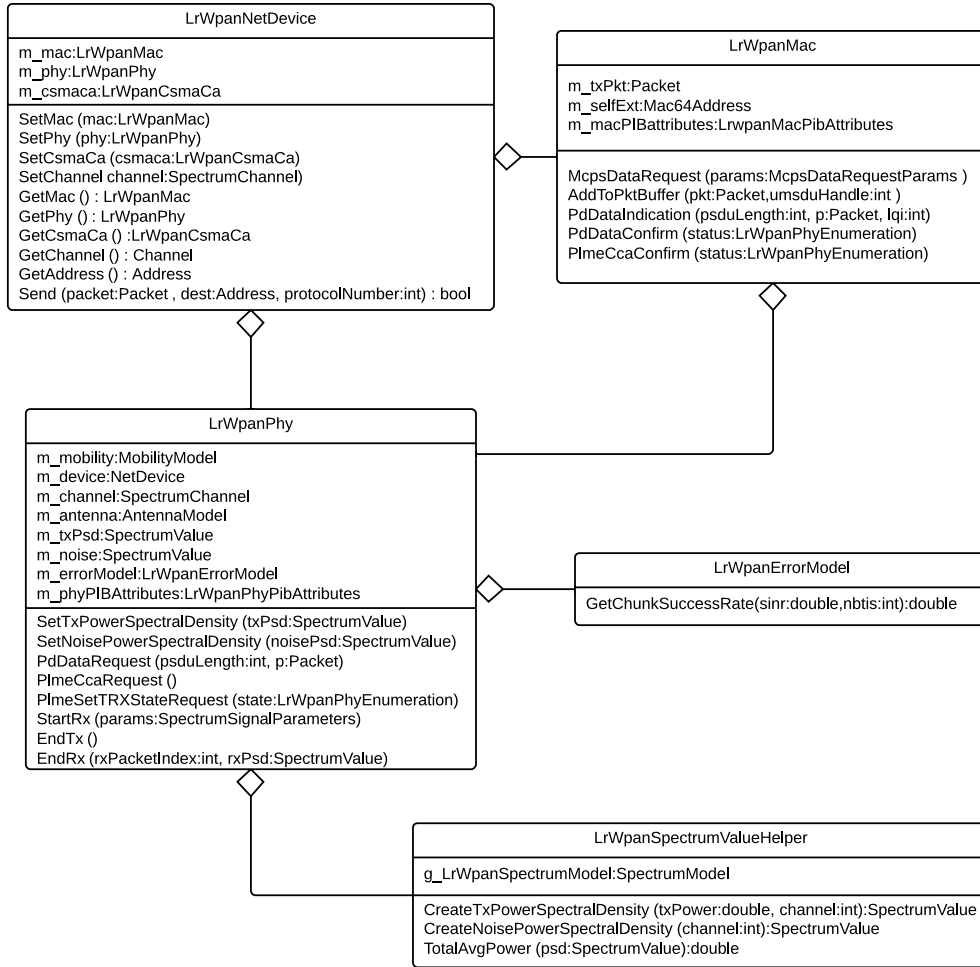


Figura 4.5: Diagrama simplificado da implementação do padrão IEEE 802.15.4 no ns-3

máscara de espectro alcança até 60 MHz , causando interferência em canais adjacentes.

4.3 Implementação do IEEE 802.15.4 no ns-3

A Figura 4.5 mostra um diagrama simplificado do modelo atual do IEEE 802.15.4 do ns-3, para uma melhor visualização apenas as principais classes, métodos e atributos são expostos. A implementação segue de perto a arquitetura do padrão, ele implementa os serviços definidos diretamente como métodos, mas com as adaptações necessárias para garantir os padrões de codificação do ns-3.

A implementação é dividida em várias classes, cada uma para um recurso específico fornecido pelo padrão. A classe *LrWpanNetDevice* liga as camadas de enlace, camadas físicas e classes de CSMA/CA para fornecer um *NetDevice* simples. Embora esta classe supostamente seja fornecida pela camada de rede, uma implementação simples foi concebida para permitir o teste da implementação. Ela fornece o método *Send*, que é invocado pela camada superior para solicitar uma transmissão de pacotes.

O *LrWpanMac* implementa vários recursos da camada de enlace, o método *McpsDataRequest* solicita a transmissão de um pacote, ele verifica o atual modo de funcionamento e estado do dispositivo, cria o cabeçalho e rodapé do pacote, e adiciona-o ao *buffer* do dispositivo. Depois que o pacote é adicionado ao *buffer*, se ativado, o procedimento de CCA começa imediatamente, caso não esteja ativado a antena é alterada para o modo de transmissão. Depois de realizado o procedimento de CCA, a camada física notifica a camada de enlace, emitindo o método *PlmeCCAConfirm*, que continua o processo de transmissão ou o processo de CSMA/CA, no caso de um canal ocupado.

Quando a camada física termina a transmissão o método *PdDataConfirm* é invocado para notificar a camada de enlace, que espera por uma confirmação ou inicia o esquema CSMA/CA para enviar um novo pacote, uma vez que apenas o modo não-segmentado é suportado.

O método *PdDataIndication* é invocado pela camada física quando um pacote é recebido. O método *MacFrameFiltering* é invocado para validar o pacote recebido, em seguida, o cabeçalho do pacote e estado atual do dispositivo são levados em conta para decidir como proceder: agendar a transmissão de um pacote ACK (essa funcionalidade está em desenvolvimento e não apresenta comportamento adequado); terminar o procedimento e remover o último pacote do *buffer*, no caso de uma confirmação; ou continuar o processo de associação. As outras comunicações de gerenciamento de rede não são suportadas.

As classes *LrWpanMacHeader* e *LrWpanTrailer* herdam classes de cabeçalho e rodapé do ns-3 para implementar os cabeçalhos e rodapés definidos pelo padrão.

A classe *LrWpanHelper* facilita a construção de simulações, fornecendo vários métodos:

- *ForceAssociateNode* estaticamente associa todos os Nodes à uma determinada rede;
- *Install* cria *LrWpanNetDevices* para um dado conjunto de Nodes;
- *EnableAsciiAll* permite o rastreamento da simulação em um arquivo de texto;
- *EnablePcapAll* permite o rastreamento da simulação em arquivos Pcap;
- O construtor da classe *LrWpanHelper* cria um canal do tipo *SingleModelSpectrumChannel* e usa um modelo de perda de propagação *Log Distance*.

A classe *LrWpanPhy* implementa a camada física do padrão. O método *PdDataRequest* é invocado pela camada de enlace sempre que uma nova transmissão deve ocorrer, ele cria o PSD e calcula duração da transmissão e, em seguida, invoca o método *StartTx*, que pertence a classe *SpectrumChannel* e notifica todos os outros dispositivos que compartilham o meio em que a transmissão tenha sido feita. Quando a transmissão é finalizada, o método *EndTx* avalia o sucesso ou falha da transmissão e notifica a camada de enlace invocando o método *PdDataConfirm*.

Quando uma nova transmissão é feita, a classe *SpectrumChannel* notifica todos os dispositivos através do método *StartRx*, que identifica se o pacote é LR-WPAN compatível e inicia o processo de recebimento, caso contrário apenas descarta o pacote, sem levar em conta o seu impacto (interferência) em recepções futuras. O método *EndRx* é chamado no final da recepção do pacote, ele calcula o SINR, embora tendo apenas o ruído térmico da antena em conta, e aplica o modelo de erro definido pelo padrão através da classe *LrWpanErrorModel*. O sucesso ou o fracasso da recepção é notificado para a camada de enlace.

Os métodos *SetTxPowerSpectralDensity* e *SetNoisePowerSpectralDensity* utilizam o objeto *LrWpanSpectrumValueHelper* para criar uma máscara de espectro em relação a uma transmissão IEEE 802.15.4, e ruído térmico da antena, respectivamente. Embora inicialmente tal máscara de espectro não reflita uma transmissão real, ela foi atualizada e utilizada neste trabalho. A Figura 4.6 mostra a distribuição do PSD através do espectro. Um método para calcular o total da potência média é também fornecido, ele é usado para calcular SINR.

4.4 Implementação do modo de operação TSCH no ns-3

O TSCH é um novo modo operacional da camada de enlace introduzido pela emenda IEEE 802.15.4e, ele estende a gama de aplicações do IEEE 802.15.4, habilitando o padrão para redes de natureza determinística. Apesar de vários grupos trabalharem atualmente na implementação do IEEE 802.15.4 para ns-3 ([11] alega uma implementação completa do protocolo), e o repositório oficial estar em constante atividade aprimorando a implementação do padrão, no melhor do nosso conhecimento não há simuladores que atualmente suportam TSCH ou qualquer funcionalidade da emenda IEEE 802.15.4e.

A implementação é baseada no repositório oficial, embora este trabalho tente integrar o modo TSCH sem quebrar a compatibilidade com o modo não-segmentado, isso não pode ser assegurado devido ao estágio inicial de desenvolvimento em que a implementação oficial está neste momento.

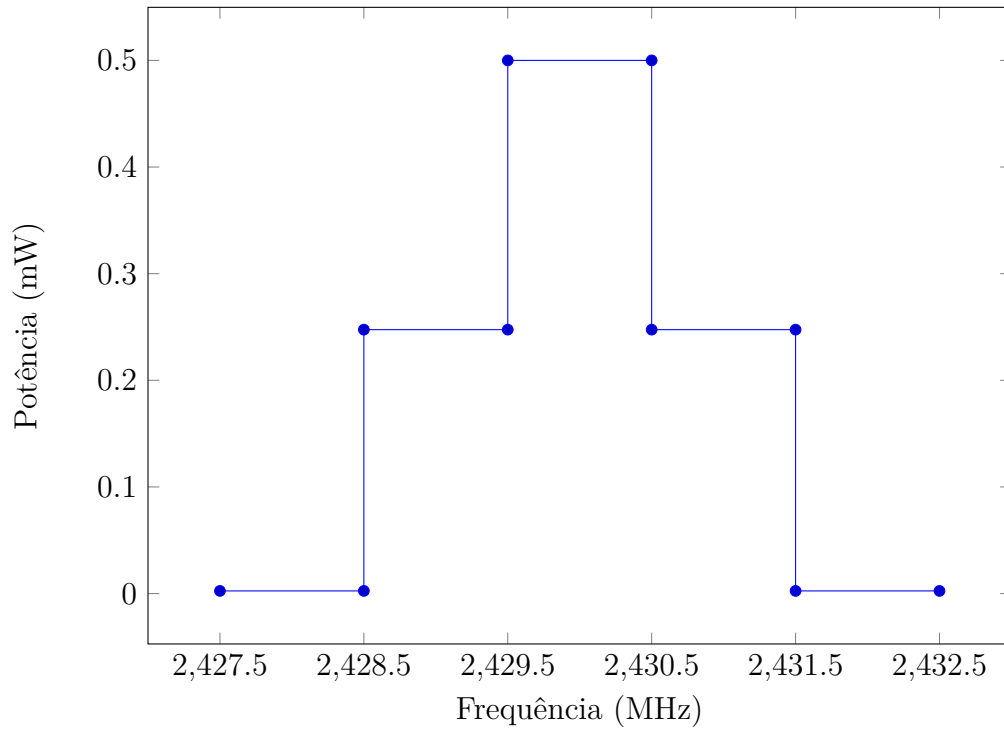


Figura 4.6: Máscara do espectro do padrão IEEE 802.15.4 para o canal 16 (frequência central = 2430MHz) com uma potência de 0dBm

LrWpanMac
m_TschRetries:int m_currentChannel:int def_MacChannelHopping:LrWpanMacChannelHopping currentLink:TschCurrentLink m_waitingLink:bool def_MacTimeslotTemplate:LrWpanMacTimeslotTemplate m_macSlotframeTable:list m_macLinkTable:list
MlmeTschModeRequest (params:MlmeTschModeRequestParams) MlmeSetSlotframeRequest (params:MlmeSetSlotframeRequestParams) MlmeSetLinkRequest (params:MlmeSetLinkRequestParams) SetDefaultHoppingSequence(uint16_t) SetHoppingSequence(sequence:vector, id:int) ScheduleTimeslot(handle:int, size:int) IncAsn() TschWaitAck() TschRxWait() TschAckDone() TschDataDone() TschSendAck(receivedMacHeader:LrWpanMacHeader) TschEndWait(params:MlmeRXEnableRequestParams) TschStartSend() TschStartCCA()

Figura 4.7: Adições realizadas na classe *LrWpanMac* para habilitar o modo TSCH

Name	Implemented
MLME-SET-SLOTFRAME	Yes
MLME-SET-LINK	Yes
MLME-TSCH-MODE	Yes
MLME-KEEP-ALIVE	No

Tabela 4.1: Serviços TSCH

É importante lembrar que a implementação atual é baseada no padrão de 2006, enquanto a emenda baseia-se no padrão de 2011. Uma vez que as principais alterações introduzidas pela versão 2011 são a adição de diferentes camadas físicas, não há grande impacto neste trabalho.

As características que não são necessárias para o cenário de simulação proposto não são implementadas neste trabalho: a formação da rede não é necessária uma vez que os dispositivos foram configurados anteriormente pela camada superior, e mesmo que erros de sincronização possam ocorrer em um ambiente real, neste estudo o sistema de programação de ns-3 administra a temporização da simulação e todos os dispositivos estão perfeitamente sincronizados.

A fim de suportar os vários novos recursos introduzidos pela emenda IEEE 802.15.4e, a estrutura do cabeçalho da camada de enlace foi atualizada. Embora novos campos tenham sido introduzidos, a compatibilidade com o padrão anterior foi mantida, isto foi alcançado através de espaços previamente reservados. Este trabalho estende a classe *LrWpanMacHeader* para suportar a nova estrutura de cabeçalho, acrescentando Elementos de Informação e supressão de números sequenciais.

A Figura 4.7 mostra as adições realizadas por este trabalho na classe *LrWpanMac*, a fim de ativar o suporte TSCH, esta é uma adição ao modelo apresentado na Figura 4.5, a interface de métodos e classes é mantida a mesma, só o seu comportamento é atualizado.

A Tabela 4.1 lista os quatro serviços introduzidos pela emenda em relação ao modo TSCH. O único serviço não implementado neste trabalho é o *MLME-KEEP-ALIVE*, que é usado para manter os dispositivos sincronizados. Os serviços *MLME-SET-SLOTFRAME* (*MlmeSetSlotframeRequest*) e *MLME-SET-LINK* (*MlmeSetLinkRequest*) foram implementadas completamente, eles são usados para adicionar, modificar e apagar *slotframes* e links, respectivamente. O *MLME-TSCH-MODE* (*MlmeTschModeRequest*) é usado para iniciar e parar o modo de TSCH.

A Figura 4.8 mostra o fluxograma de um dispositivo em modo TSCH, quando o serviço *MLME-TSCH-MODE* é invocado por uma camada superior, o método *ScheduleTimeslot* é agendado a cada intervalo de um *timeslot* (10 ms na configuração padrão), este método verifica se o *timeslot* atual está atribuído para o dispositivo, procedendo para uma

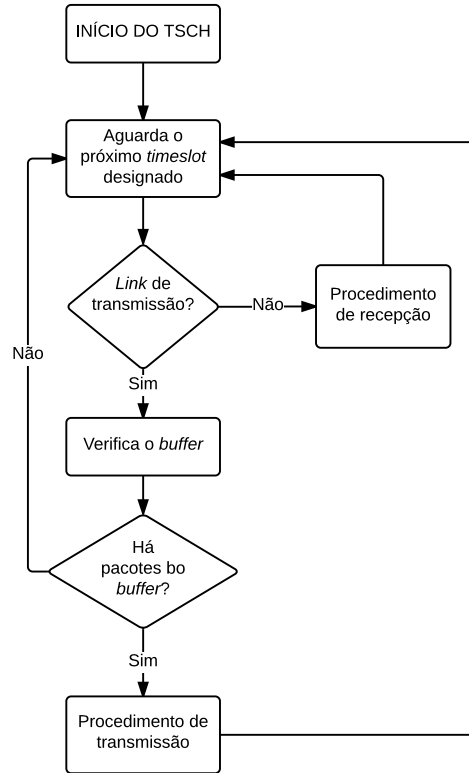


Figura 4.8: Fluxograma de uma rede em modo TSCH

transmissão, no caso de haver um pacote no *buffer*, ou uma recepção.

A Figura 4.9 mostra os fluxogramas dos procedimentos de recepção e transmissão implementados. Quando um *timeslot* de transmissão é atingido o dispositivo primeiramente altera o seu canal de comunicação de acordo com a lista de sequência de salto de canal, então o procedimento CCA é agendado para o momento *macTsCCAOOffset*. Depois que o procedimento CCA é executado (*TschStartCCA*) o serviço PLME-CCA.confirm é invocado e, no caso de um canal ocupado, termina as atividades do *timeslot* atual, ou altera o rádio para o modo de transmissão, no caso de um canal ocioso. Depois que o rádio está configurado para o modo de transmissão o serviço da camada física PD-DATA.request é invocado para transmitir o pacote atual. Quando a camada física termina a transmissão do pacote o serviço PD-DATA.confirm é invocado, o qual é implementado na camada de enlace, para agendar o método *TschWaitAck* para *macTsRxAckDelay*. Quando o momento *macTsRxAckDelay* for atingido o método *TschWaitAck* altera o rádio para o modo de recepção e em *macTsAckWait* o método *TschAckDone* verifica se um pacote ACK está sendo recebido. Se não for detectada a recepção, o número de tentativas atuais é incre-

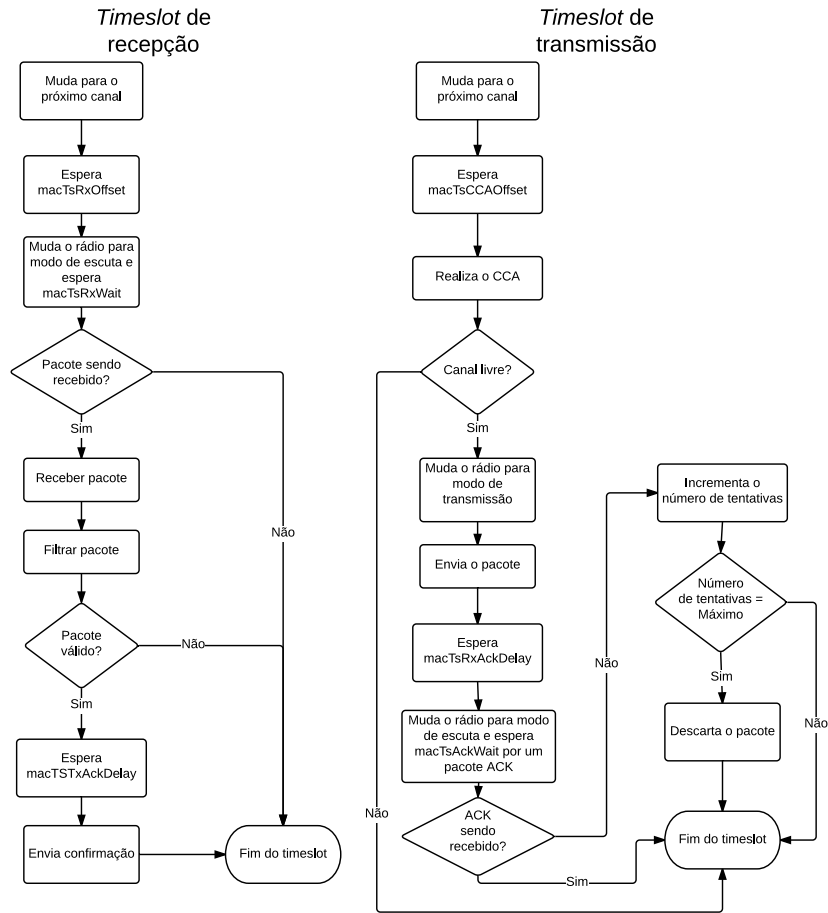


Figura 4.9: *Timeslots* de transmissão e recepção

mentado, e se o máximo é atingido (5 por padrão) o pacote é descartado. O intervalo de tempo chega ao fim depois de receber o pacote de confirmação.

Em um *timeslot* de recepção a primeira atividade é mudar o canal atual de acordo com a lista da sequência de salto de canal, em seguida, em `macTsRxOffset` o método *TschEndWait* altera o rádio para o modo de recepção e espera por um pacote durante `macTsRxWait`. Caso nenhum pacote seja detectado até o fim de `macTsRxWait`, o *timeslot* chega ao fim, de outro modo o pacote recebido é filtrado para determinar se é um pacote válido. Depois de receber um pacote válido o método *TschSendAck* é agendado para `macTsTxAckDelay` para criar e enviar o pacote de confirmação e então o *timeslot* chega ao fim.

Vários serviços definidos pelo padrão IEEE 802.15.4 foram atualizados para suportar o modo TSCH:

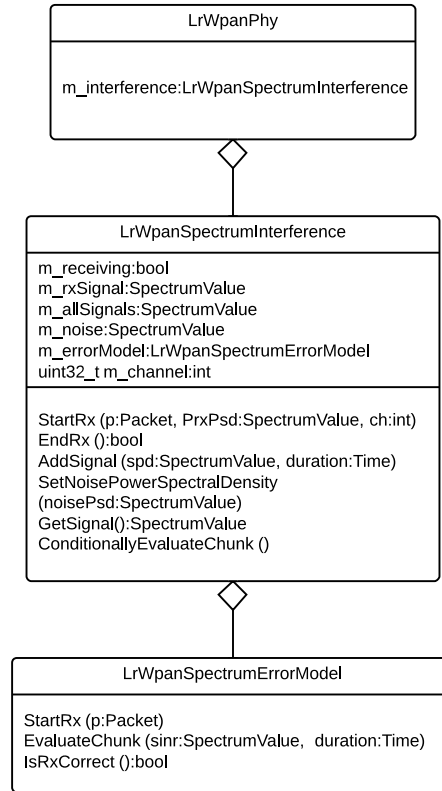


Figura 4.10: Diagrama simplificado das classes *LrWpanSpectrumInterference* e *LrWpanSpectrumErrorModel*

- PLME-CCA.confirm: atualizado para iniciar diretamente a transmissão se o canal está ocioso;
- PD-DATA.confirm: atualizado para esperar o próximo *timeslot* atribuído após uma transmissão;
- MCPS-DATA.request: quando uma nova transmissão é solicitada pela camada superior, se o dispositivo estiver no modo TSCH o pacote é simplesmente adicionado ao *buffer*, quando um novo intervalo de tempo de transmissão atribuído é atingido o primeiro pacote do *buffer* é enviado;
- PD-DATA.indication: atualizado para utilizar o novo procedimento de confirmação quando um pacote é recebido;

Para fornecer suporte completo à interferência a classe *LrWpanSpectrumInterference* foi criada e integrada à classe *LrWpanPhy*, baseada na classe *SpectrumInterference* for-

necida pelo ns-3, é implementado um canal AWGN, em que todos os outros sinais são interpretados como interferência. A classe *LrWpanErrorModel* foi atualizada para ser compatível com a classe de interferência, o método *EvaluateChunk* é invocado sempre que as condições do canal são alteradas e o dispositivo está recebendo um pacote. A Figura 4.10 mostra o diagrama das classes *LrWpanSpectrumInterference* e *LrWpanSpectrumErrorModel*.

A fim de melhor representar a realidade, o método *TotalAvgPower* da classe *LrWpanSpectrumValueHelper* foi atualizado para considerar apenas o canal atual ao realizar a integração através do dado PSD para obter a potência média total, que é usada pela classe *LrWpanSpectrumInterference* para analisar o SINR da transmissão atual.

Vários exemplos foram desenvolvidos a fim de validar este trabalho, *lr-wpan-tsch-one* implementa um *slotframe* em que vários dispositivos têm um *timeslot* de transmissão para enviar dados ao coordenador, a Figura 4.11 mostra a taxa de transferência obtida ao executar esse exemplo com pacotes de tamanhos diversos, a natureza determinística do modo TSCH permite a sua validação por análise matemática. O exemplo *lr-wpan-tsch-simple-interference* implementa duas redes TSCH em execução ao mesmo tempo, causando interferência uma na outra e *lr-wpan-spectrum-error-model-plot* adapta a implementação do antigo modelo de erro para apoiar a nova arquitetura de interferência, a Figura 4.12 mostra o BER gerada pelo *LrWpanSpectrumErrorModel* através de vários valores de SNR, o resultado pode ser validado contra [4, FigureE.2].

Modelagem do consumo de energia

Em redes de sensores sem fio o consumo de energia é um fator fundamental, devido à sua implantação, muitas vezes dispositivos não têm uma fonte de energia contínua, mas contam com uma bateria para fornecer energia. Também pode ser o caso de que não seja possível trocar tal bateria devido à localização do dispositivo, por exemplo, se a rede de sensores é implementada num ambiente perigoso ou de difícil acesso.

A natureza determinística do TSCH se adapta muito bem às redes que possuem energia restrita, uma vez que os dispositivos têm um tempo predeterminado para transmitir e receber dados, nenhuma energia é gasta disputando o meio, por exemplo, usando um esquema CSMA/CA. Normalmente, o rádio é o elemento que consome a maior parte da energia em um dispositivo. Em uma rede TSCH, dispositivos poupam energia, desligando seus rádios enquanto não estiverem em um *timeslot* atribuído.

Vilajosana et. al. [23] apresenta um modelo de consumo de energia para as redes TSCH que é baseado no consumo de energia de cada tipo de *timeslot*. A Tabela 4.2 lista os seis tipos de intervalo de tempo diferentes identificados por [23]. Cada tipo de intervalo de tempo foi dividido em várias etapas relacionadas com as atividades do micro

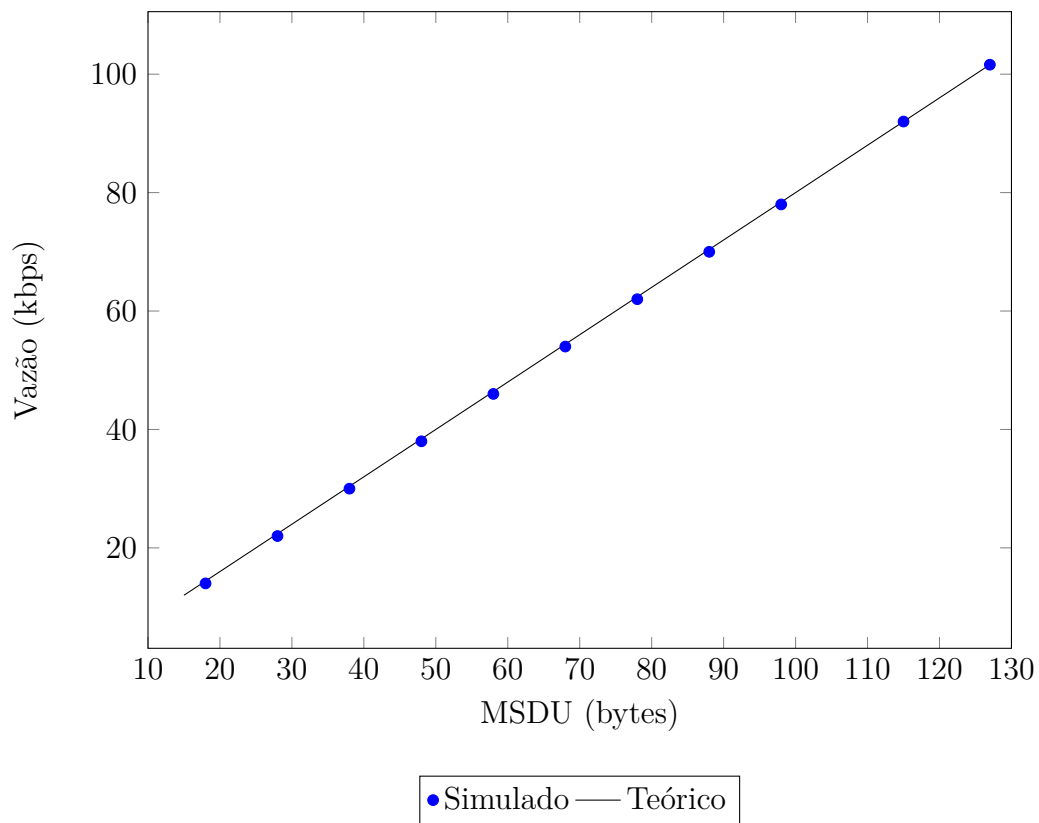


Figura 4.11: Vazão de uma rede TSCH configurada com *timeslots* de 10ms com vários tamanhos de pacotes.

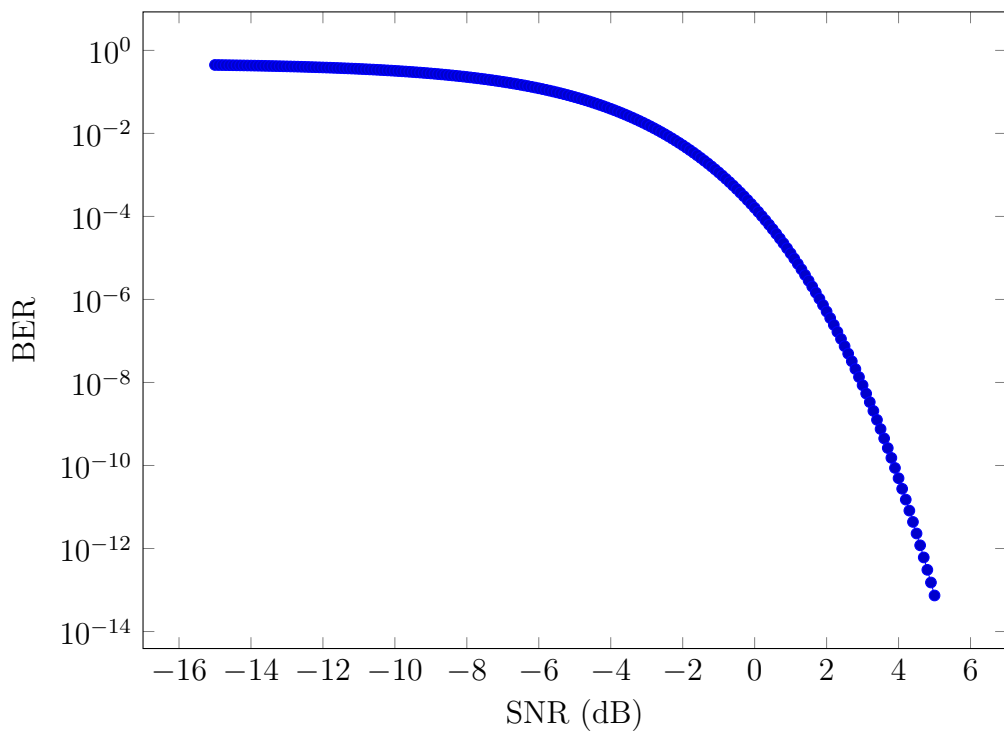


Figura 4.12: Validação do *SpectrumErrorModel*

Tipo do <i>timeslot</i>	Descrição
TxDataRxAck	O dispositivo envia um pacote e recebe a confirmação.
TxData	O dispositivo envia um pacote mas não recebe confirmação.
RxDataTxAck	O dispositivo recebe um pacote e envia a confirmação.
RxData	O dispositivo recebe um pacote mas não envia confirmação.
Idle	O dispositivo escuta por um pacote mas não recebe nada.
Sleep	O <i>timeslot</i> não é atribuído ao dispositivo.

Tabela 4.2: Tipos de *timeslot*

Constante	Duração (μs)
macTimeslotLength	15000
macTsTxOffset	4000
macTsRxOffset	305
macTsRxWait	2600
macTsAckWait	1000
macTsTxAckDelay	4000
macTsRxAckDelay	3500

Tabela 4.3: Configuração to *timeslot* do OpenWSN

controlador e do rádio do dispositivo, a energia consumida em um *timeslot* é a soma da energia consumida por cada fase.

Duas plataformas de hardware são utilizadas para verificar o modelo apresentado: o OpenMoteSTM, que apresenta um micro controlador de 32 bits (STMicroelectronic STM32F103RB), e o GINA, que possui um micro controlador de 16 bits (Texas Instruments MSP430F2618), ambas as plataformas utilizam o mesmo módulo de rádio ATMEL AT86RF231. Os motes são carregados com a implementação OpenWSN TSCH [24], a Tabela 4.3 lista a configuração do *timeslot* do OpenWSN.

Para verificar o modelo criado, o consumo de energia calculado é comparado com medições efetuadas em ambas as plataformas, a Figura 4.4 lista a carga consumida, em μC , de todos os tipos de *timeslot* usando o modelo proposto e as medidas nos dispositivos.

Com os dados fornecidos, é possível adquirir o total de energia consumida por um aparelho simplesmente rastreando todos os intervalos de tempo pelos quais ele passou. Um experimento foi configurado com motes GINA em que três dispositivos enviam um pacote a cada 4 segundos a um propagador, que por sua vez envia os dados recebidos para um nó central. A Tabela 4.5 mostra a energia consumida e calculada de tal experimento, *slotframes* são compostos por 100 *timeslot* (duração total de 1,5 segundos). Como pode ser visto os valores medidos e calculados são bem aproximados, confirmando assim a validade do modelo proposto. Neste trabalho, o modelo de energia é utilizado para prever

Tipo de <i>timeslot</i>	Medido (μC)		Calculado (μC)	
	GINA	OpenMote	GINA	OpenMote
Idle	47.9	101.1	54.1	85.2
Sleep	4.9	37.8	8.2	9.2
TxDataRxAck	92.6	161.9	103.3	151.2
TxData	69.6	119.2	76.7	123.1
RxDataTxAck	96.3	217.0	105.2	175.9
RxData	72.1	154.8	78.0	125.0

Tabela 4.4: Carga consumida por cada tipo de *timeslot*, medida e calculada

Dispositivo	Medido		Calculado	
	Folha	Propagador	Folha	Propagador
Energia consumida (μA)	455.0	581.9	415.4	569.8

Tabela 4.5: Energia total consumida em um *slotframe*

a energia consumida pelos dispositivos no cenário proposto.

4.5 Resumo do capítulo

Neste capítulo a arquitetura do simulador ns-3 é apresentada. A classe *Node* representa um dispositivo genérico em que são adicionadas aplicações para gerar tráfego. As aplicações são ligadas a placas de rede, representadas pela classe *NetDevice*. A classe *SpectrumModel* simula o meio em que a comunicação ocorre, este modelo suporta transmissões em diferentes frequências e lida com modelos de propagação e atraso.

A implementação do modo TSCH, realizada neste trabalho, também é apresentada. Com exceção do serviço de sincronização, todos os serviços referentes ao modo TSCH definidos pela emenda são implementados. O modelo de consumo de energia apresentado por Vilajosana et. al. [23] é implementado para possibilitar o estudo deste fator, através da classificação do consumo de energia de diferentes tipos de *timeslots* é possível alcançar o consumo de energia total de um dispositivo.

Capítulo 5

Resultados

Neste capítulo são apresentados os resultados obtidos por este trabalho. A primeira seção descreve em detalhes o cenário de simulação projetado e sua importância, a segunda seção apresenta os resultados da simulação e os valida por análise analítica, a terceira seção apresenta as melhorias que um esquema de autoaprendizagem trariam para o TSCH.

5.1 Simulação

É importante projetar uma simulação que melhor representa o cenário desejado, simulações podem ser essenciais para evitar o desperdício de recursos, por exemplo, em uma rede sem fio que tem uma vasta cobertura uma boa simulação fornece o dimensionamento correto do número e posição das antenas necessárias, se a simulação não for feita corretamente a implantação da rede não oferecerá a cobertura correta, causando uma necessidade de desperdiçar recursos para corrigir o problema.

Em estudos acadêmicos simulações são usadas para identificar gargalos em padrões e validar novas propostas, neste trabalho um cenário de simulação é criado para validar a implementação do modo operacional IEEE 802.15.4e TSCH para o simulador ns-3 e para identificar como este modo operacional pode se beneficiar de recursos de autoaprendizagem, para alcançar isto o cenário projetado altera as condições ambientais e os requisitos da aplicação. Este trabalho é, no o melhor de nosso conhecimento, o primeiro que foi realizado no ns-3 e implementa o modo operacional TSCH da emenda IEEE 802.15.4e.

O cenário proposto simula uma rede de sensores acústicos sem fio usada em um ambiente residencial para ajudar seus moradores de várias maneiras, tais como a economia de energia, desligando as luzes de um quarto vazio. Como mostrado na Figura 5.1 a residência possui dimensões de 8 por 11 metros e tem 5 cômodos. A rede de sensores possui 16 dispositivos, um sendo o servidor central, que recebe e processa todas as informações geradas pelos outros 15 dispositivos, que são sensores acústicos. A distância entre os

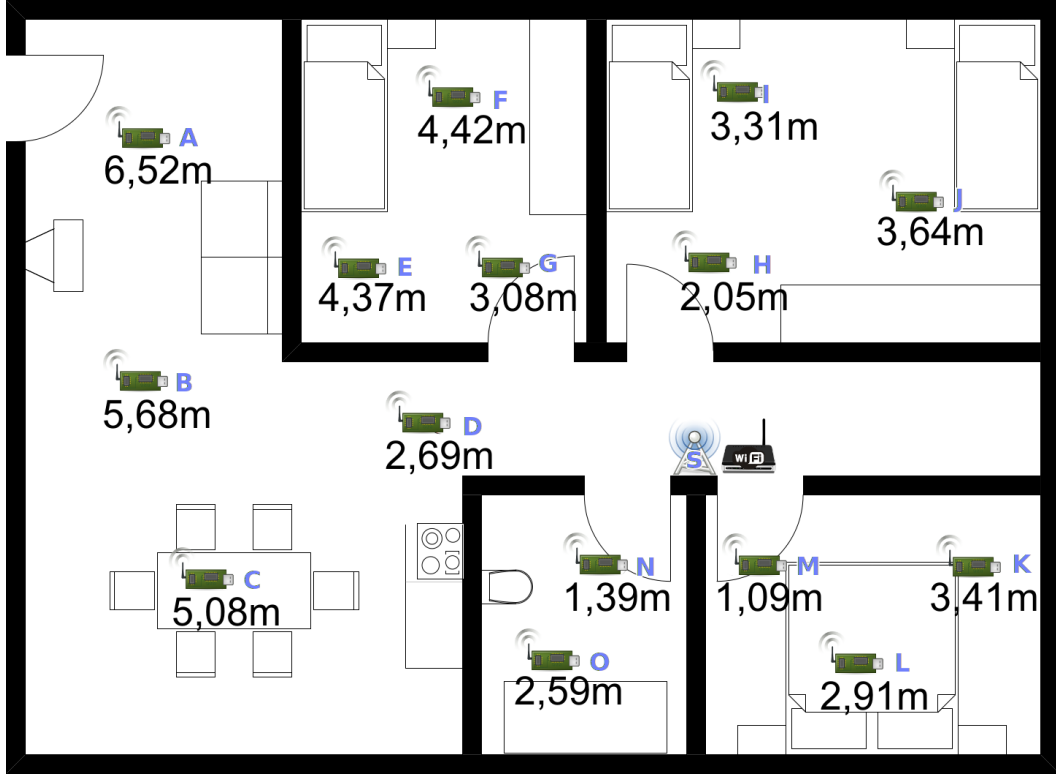


Figura 5.1: Residência utilizada como cenário de simulação

A → S	B → S	C → S	D → S	E → S	F → S	G → S	H → S	I → S	J → S	K → S	L → S	M → S	N → S	O → S
TS 0	TS 1	TS 2	TS 3	TS 4	TS 5	TS 6	TS 7	TS 8	TS 9	TS 10	TS 11	TS 12	TS 13	TS 14

Figura 5.2: Configuração do *slotframe* para o cenário proposto

sensores e o servidor central é mostrada na Figura 5.1, que varia de 1,09 metros a 6,52 metros.

A rede funciona no modo TSCH, como mostrado na Figura 5.2, a rede foi configurada com um *slotframe* de 15 *timeslots*, um para cada sensor se comunicar com o servidor central. Cada intervalo de tempo tem um comprimento de 10ms. Os sensores produzem 114 *bytes* de dados (máximo permitido considerando um cabeçalho de 13 *bytes*) a cada 10ms, a frequência ou o tamanho dos dados gerados não se altera a menos que o sensor fique inativo (o comodo está vazio), neste caso o sensor não envia dados. Todos os parâmetros de configuração são definidos para o valor padrão definido pelo protocolo.

Para simular um ambiente dinâmico um dispositivo compatível com IEEE 802.11-2007[2] (Wi-Fi) é colocado ao lado do nó central. Dispositivos Wi-Fi são muitas vezes utilizados em residências para fornecer acesso à Internet aos dispositivos móveis, como *smartphones* e *laptops*. A classe *WifiSpectrumValue5MhzFactory*, fornecida pelo modelo *Spectrum*, é utilizada configurada com uma potência de transmissão de 0.1 W e canal

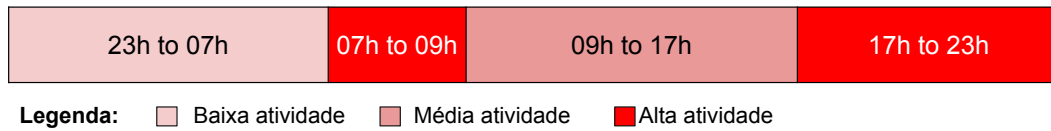


Figura 5.3: Atividade da rede de sensors durante o dia

de comunicação 6 para gerar um PSD de acordo com o padrão acima mencionado, a classe *WaveGenerator* é usada para propagar o sinal Wi-Fi através do meio. A classe *MultiModelSpectrumChannel* é utilizada uma vez que o modelo Wi-Fi usa uma resolução de 5 MHz e a rede LR-WPAN usa uma resolução de 1 MHz.

A classe *FriisSpectrumPropagationLossModel* é utilizada para representar a perda de energia do sinal durante a transmissão, de acordo com este modelo, no cenário de simulação proposto a perda máxima observada (pelo dispositivo mais distante do nó central) é de -56,33 dB, enquanto a perda mínima observada é de -40,80 dB.

O cenário proposto tem uma duração de 24 horas, tentando representar um dia normal de uma família comum, o número de pessoas na casa muda conforme o dia passa, bem como a atividade da rede de sensores e do dispositivo Wi-Fi. A Figura 5.3 representa a atividade assumida em um dia comum, ela contém quatro regiões:

- 23h a 07h: Período em que a família está dormindo, portanto, há muito pouca atividade de rede;
- 07h a 09h: Período em que a família está se preparando para o trabalho e escola, todo mundo está usando *smartphones*;
- 09h a 17h: Período em que a maioria dos membros da família estão trabalhando ou na escola, há atividade moderada na casa, equivalente a atividade de 2 pessoas;
- 17h a 23h: Período em que todo mundo está em casa novamente, mesma atividade que o segundo período.

É importante notar que em um ambiente real, os períodos de cada região não são tão rigorosamente definidos, a transição entre as regiões ocorre com os moradores saindo e entrando na casa. Esta simulação é, no entanto, simplificada por uma mudança brusca, a fim de melhor analisar os limites superior e inferior de cada um dos períodos em relação a vários aspectos de desempenho.

Como mostrado na Tabela 5.1, existem 3 níveis de atividade diferentes, em um nível de atividade baixo há apenas 3 nós da rede ativos, eles representam 20% do uso da rede, neste nível não há nenhuma atividade do dispositivo Wi-Fi. Em um nível de atividade médio há 6 nós ativos, eles representam 40% de uso da rede, o dispositivo Wi-Fi tem um

Nível de atividade	Descrição	Nós ativos	Atividade Wi-Fi
Baixo	Todos dormindo	20%	Nenhuma
Médio	1 ou 2 pessoas em casa	40%	50%
Alto	Casa cheia	100%	100%

Tabela 5.1: Níveis de atividade

ciclo de trabalho de 50% em um período de 1 segundo (0,5 segundos ligado e 0,5 segundos desligado). Em um nível alto de atividade 100% dos nós estão ativos e o dispositivo Wi-Fi também está ativo o tempo todo. Os nós ativados em cada nível são sempre os últimos, por exemplo, em um período de baixa atividade os nós ativos são os O, M e N.

As antenas dos nós foram configuradas com os valores padrão de potência de transmissão e sensibilidade do receptor, 0 *dBm* e -85 *dBm*, respectivamente. A lista de salto de canal é composta por todos os canais disponíveis na faixa de 2.4 *GHz*.

A fim de capturar as condições do meio no servidor central um *SpectrumAnalyzer* também é colocado ao seu lado, ele é configurado para calcular a média do PSD a cada segundo. Para melhor reproduzir um dispositivo real ele produz um ruído de $1.0 \times 10^{-15} W/Hz$.

5.2 Resultados da simulação

Os resultados aqui apresentados foram obtidos através da execução do cenário de simulação proposto no simulador ns-3, para diminuir a duração da simulação uma hora foi considerada como um minuto, devido a essência determinística do TSCH esta abordagem não tem impacto sobre os resultados.

A presença de um dispositivo Wi-Fi no cenário simulado provoca uma forte influência sobre o comportamento da rede de sensores. Esta influência não é igual a todos os dispositivos, que varia de acordo com a distância entre o dispositivo Wi-Fi e o nó. Essa variação acontece devido à atenuação que o sinal Wi-Fi sofre ao longo do caminho. Como foi dito na seção anterior esta atenuação é representada pelo modelo de perda de propagação Friis.

Para compreender o impacto do sinal Wi-Fi é importante analisar a sua máscara de espectro. A Figura 5.4 mostra os canais da rede de sensores que sofrem influência do transmissor Wi-Fi, o sinal é originado do canal central 6, sua máscara de espectro pode compreender até 60 *MHz*.

A Figura 5.5 mostra o BER observado pelo nó central de uma transmissão realizada por cada dispositivo da rede de sensores, como pode ser visto, o dispositivo Wi-Fi influencia todos os dispositivos da rede, mesmo o sinal do dispositivo mais próximo é muito baixo comparado com o sinal Wi-Fi. Embora não esteja presente na Figura 5.4 o canal 11

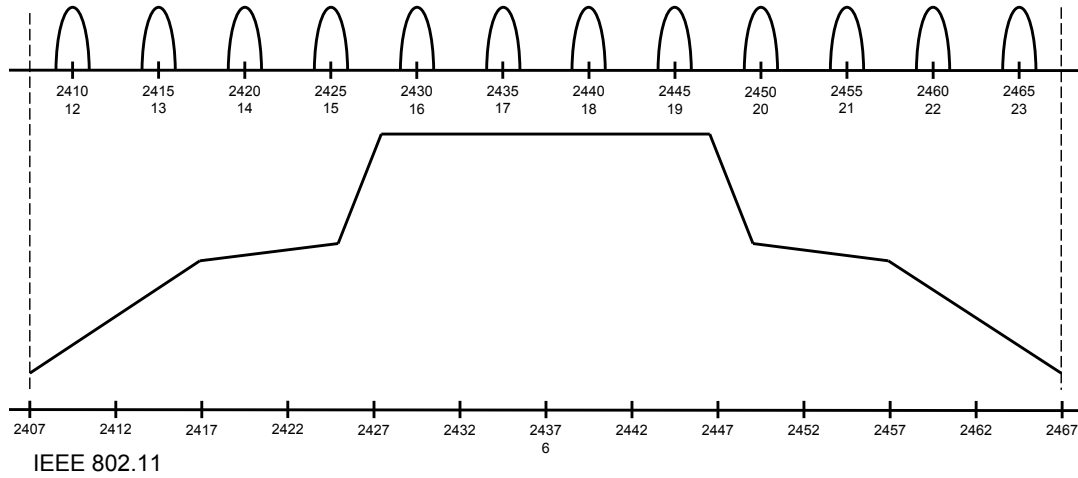


Figura 5.4: Comparação entre as mascaras de espectro do Wi-Fi e IEEE 802.15.4

também é afetado pelo sinal Wi-Fi, apenas 0,5 MHz da máscara da transmissão Wi-Fi está presente em tal canal, mas devido à posição do dispositivo (ao lado do nó central) já é suficiente para perturbar as condições deste canal.

Um BER de 10^{-9} é muitas vezes considerado o limiar de uma condição aceitável, a Figura 5.5 mostra que isso só é conseguido nos canais 24, 25 e 26, em que não há interferência Wi-Fi, considerando que todas as transmissões nos canais afetados seriam um fracasso ou não passariam pelo processo de CCA, apenas 18,75% dessas transmissões seriam bem sucedidas.

A interferência causada pelo dispositivo Wi-Fi afeta vários aspectos da rede de sensores, neste estudo três aspectos são analisados: vazão, atraso e consumo de energia. A fim de validar os dados obtidos uma análise matemática é realizada e comparada com os resultados da simulação, isso é possível devido à natureza determinística do modo operacional TSCH.

Vazão

A fim de analisar o impacto da interferência causada pelo dispositivo Wi-Fi é sobre a vazão da rede é importante qual é a vazão necessária para que a rede funcione em perfeitas condições, em outras palavras, qual é o desempenho sem a presença de interferência. A fim de obter esses dados o dispositivo Wi-Fi foi removido da simulação e todos os outros parâmetros foram mantidos os mesmos. Tal cenário é também importante para a validação da implementação do TSCH proposto pelo presente trabalho.

A validação dos resultados é realizada por análise analítica. Cada nível de atividade é analisado quanto ao seu rendimento e, em seguida, comparados com os resultados simu-

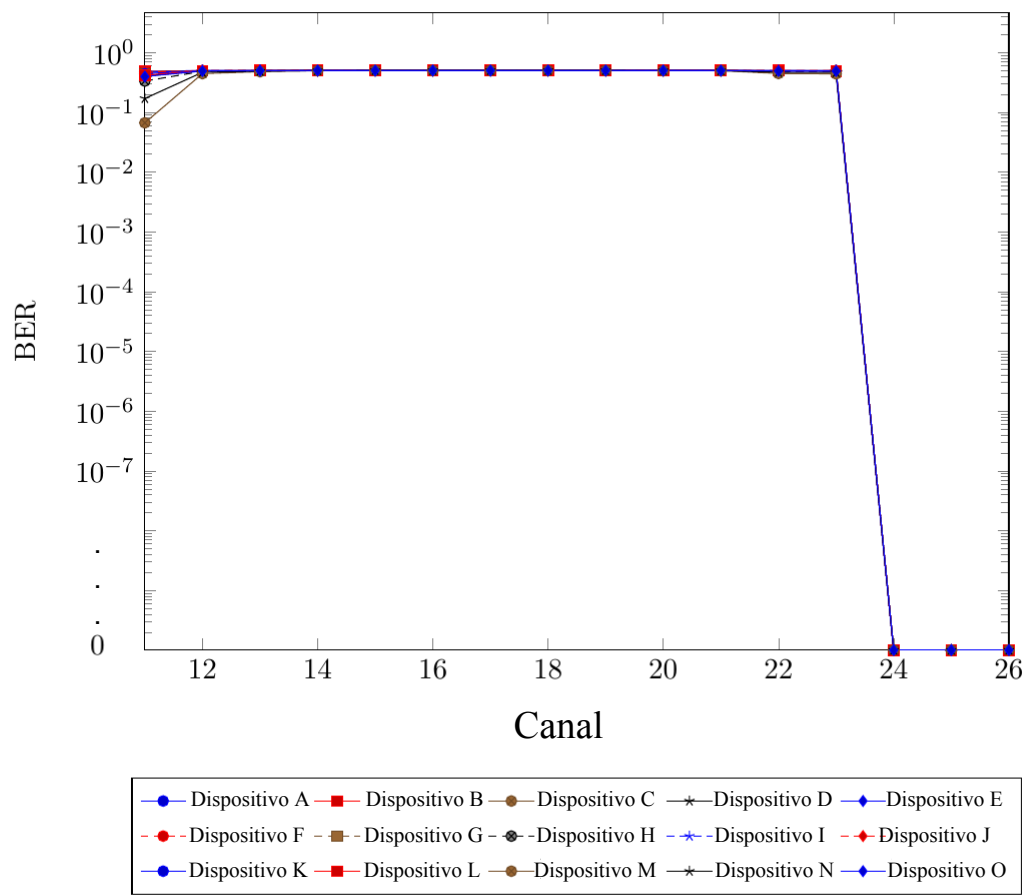


Figura 5.5: BER de cada canal de todos os dispositivos

Nível de atividade	Nós ativos	Vazão máxima
Baixo	3	20.32 kbps
Médio	6	40.64 kbps
Alto	15	101.60 kbps

Tabela 5.2: Vazão máxima para cada nível de atividade

lados. Uma vez que a configuração do *slotframe* permanece o mesmo através de todos os níveis de atividade (com 15 *timeslots*), há um rendimento ótimo diferente para cada um. A Tabela 5.2 mostra a capacidade máxima de cada nível de atividade, a qual é calculada da seguinte forma:

$$VazaoMaxima_i = \frac{TamanhoPacote * 8 * i}{TamanhoSlotframe * DuracaoTimeslot} \quad (5.1)$$

Em que $VazaoMaxima_i$, em *kbps* significa a vazão máxima com i dispositivos ativos, *TamanhoPacote* está em *bytes* e deve incluir o tamanho do cabeçalho, *TamanhoSlotframe* é o número de *timeslots* no *slotframe* e *DuracaoTimeslot*, em segundos, é a duração de um único *timeslot*.

A Figura 5.6 mostra os resultados simulados com e sem interferência. Os resultados sem interferência, como esperado, correspondem aos valores obtidos através da análise matemática. Esse resultado valida a implementação em um cenário ideal em que não há perda de pacotes devido a interferência. Em tal cenário, um recurso de autoaprendizagem não seria útil uma vez que as condições da rede não mudam. A primeira região é relativa a um nível de baixa atividade, que não sofre interferência e, como esperado, a vazão de 20,32 kbps é atingida.

A segunda região, relativa ao nível mais alto de atividade, atinge 18,9% (19,202 *kbps*) da vazão requerida pela aplicação, este resultado corresponde à taxa de sucesso de 18,75% obtido previamente, com transmissões de dados apenas sendo bem sucedidas em 3 dos 16 canais.

A terceira região, relativa ao nível de atividade mediano, chega a 60% (24.384 *kbps*) da vazão exigida pela aplicação, tendo em conta que o dispositivo Wi-Fi está ativo durante 50% do tempo, em metade da duração dessa região a vazão deve ser tal como exigido pela aplicação, na outra metade deve ser apenas 18,75% do valor desejado, este regime corresponde a 59,375% da vazão necessária, validando o valor obtido pela simulação.

Atraso

O atraso em relação a cada um dos pacotes também é afetado pela interferência do dispositivo Wi-Fi. Numa rede TSCH sem interferência, quando existe um pacote para

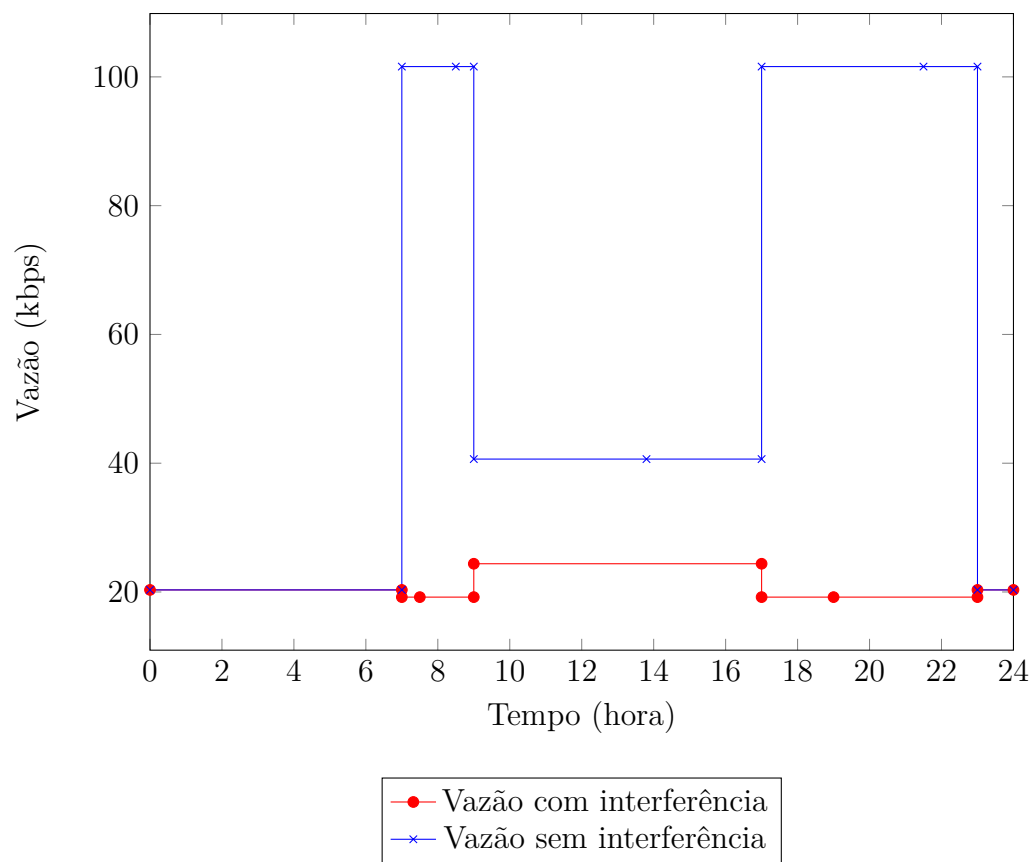


Figura 5.6: Vazão simulada, com e sem interferência

cada *timeslot* o atraso máximo pode ser no máximo a duração do *slotframe* (o pacote chega ao buffer logo após o último pacote foi enviado), na simulação deste trabalho esse valor é de 0.15s. Esta característica é muito importante para o bom funcionamento de uma rede, se um pacote leva mais do que é esperado para chegar ao seu destino a rede pode não funcionar corretamente.

É importante notar que os *buffers* dos dispositivos foram esvaziados entre cada zona de atividade, isto é feito para observar os resultados de uma região sem o impacto da anterior. Caso os *buffers* não estejam vazios, após uma região com interferência existiram pacotes na fila, afetando os resultados da próxima região.

A Figura 5.7 mostra o atraso médio do dispositivo N, que está ativo em todas as regiões através do cenário simulado. Como esperado, o atraso aumenta consideravelmente em presença de interferência, em tais casos, a rede não funciona corretamente. A primeira região não apresenta qualquer aumento no atraso, pois não possui interferência.

O atraso observado em regiões com interferência aumenta com o tempo, isto se dá devido ao fato de que a aplicação gera pacotes a cada 150 ms. O atraso nas regiões com interferência apresenta um aumento contínuo, alcançando rapidamente valores impraticáveis. A fim de manter a proporção correta na Figura 5.7, as regiões com interferência foram cortadas. A inclinação da linha se relaciona com a taxa de sucesso anteriormente calculada, no nível de atividade mais alto a inclinação é de cerca de 77.6° e no nível de atividade mediano a inclinação é de 34.9°. De acordo com os resultados obtidos fica claro a necessidade do uso de autoaprendizagem na presença de interferência.

Consumo de energia

O consumo de energia é um fator crucial em uma rede de sensores sem fio, neste trabalho a energia consumida pelos dispositivos é simulada através da utilização do modelo energético desenvolvido por [23] e discutido no capítulo 4.

Os dispositivos utilizados em [23] são o OpenMoteSTM, um dispositivo de alta capacidade com um micro controlador de 32 bit e um alto consumo de energia, e GINA, um dispositivo de baixa capacidade com um micro controlador de 16 bit e baixo energia consumo. Este estudo considera apenas a plataforma GINA, que opera a 3.7V, uma vez que a tarefa dos dispositivos é apenas a de captação do sensor acústico, não necessitando de um alto poder de processamento.

Em [23], CCAs e transmissões falhas não foram contempladas. Neste trabalho, um *timeslot* em que ocorre falha no processo de CCA consome a mesma quantidade de energia que um *timeslot* não atribuído. Embora precisão a energia consumida no *timeslot* não seja representada com precisão, é o tipo de *timeslot* que mais se assemelha à situação, uma vez que a única atividade em tal *timeslot* é o procedimento CCA. Quando ocorre

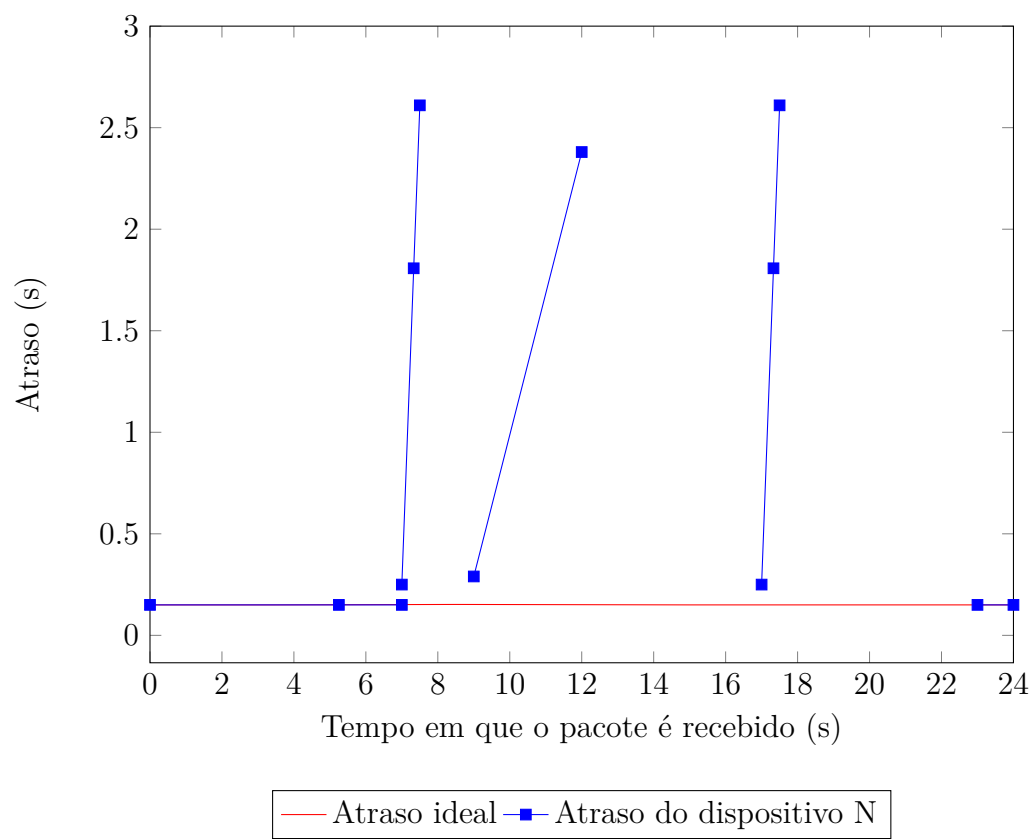


Figura 5.7: Atraso do dispositivo N

Tipo de <i>timeslot</i>	OpenWSN	IEEE 802.15.4e
Idle	47.9	31.9
Sleep	4.9	3.3
TxDataRxAck	92.6	61.7
TxData	69.6	46.4
RxDataTxAck	96.3	96.3
RxData	72.1	64.2

Tabela 5.3: Corrente consumida, em μC , para cada tipo de *timeslot*

uma falha na transmissão e nenhum ACK é recebido, neste trabalho o consumo de energia é dado com o mesmo que o tipo de *timeslot* TxData. Embora haja consumo de energia ao escutar pelo ACK que foi recebido, este é o tipo de *timeslot* que mais se assemelha à situação.

O modelo de energia criado é baseado em uma implementação OpenWSN do TSCH. A configuração de *timeslot* predefinida de tal dispositivo é diferente do utilizado no presente estudo, a fim de utilizar os valores de consumo de energia uma conversão foi realizada. A Tabela 5.3 exibe o consumo de energia pela plataforma GINA utilizando a configuração de *timeslot* de OpenWSN e da configuração padrão do protocolo, como pode ser visto o consumo de energia de cada tipo de *timeslot* é menor na configuração do protocolo, essa discrepância é devida à diferença do tamanho dos *timeslots* entre as duas configurações. 15ms na configuração do OpenWSN e 10ms na configuração padrão do protocolo. Embora com menor consumo de energia em um único *timeslot*, a configuração padrão consome mais energia ao longo do tempo, uma vez que possibilita a ocorrência de mais *timeslots* durante o mesmo período.

A conversão é obtida pela multiplicação da carga de cada tipo de *timeslot* pela relação entre a duração dos *timeslots*. Embora este método forneça uma conversão simples e aproximada, um método melhor seria considerar a diferente duração de cada fase que compõe um tipo de *timeslot*, infelizmente a discriminação de todos os tipos de *timeslot* não foi disponibilizada pelo estudo apresentado, assim apenas o primeiro método de conversão é aplicável. O valor medido da carga consumida em cada tipo de *timeslot* é utilizado para fornecer uma melhor aproximação.

A Figura 5.8 mostra o consumo de energia nos dispositivos O, L e I em cada região da linha do tempo do cenário simulado (considerando uma hora sendo 1 minuto) e ao longo de toda a simulação. O dispositivo O representa os nós ativos em toda a simulação, o dispositivo L representa nós ativos apenas nas regiões 2, 3 e 4, e o dispositivo I representa nós ativos apenas nas regiões 2 e 4. O dispositivo O consome mais energia porque está ativo em todas as regiões, isto é de fato como na realidade, onde geralmente a sala é o

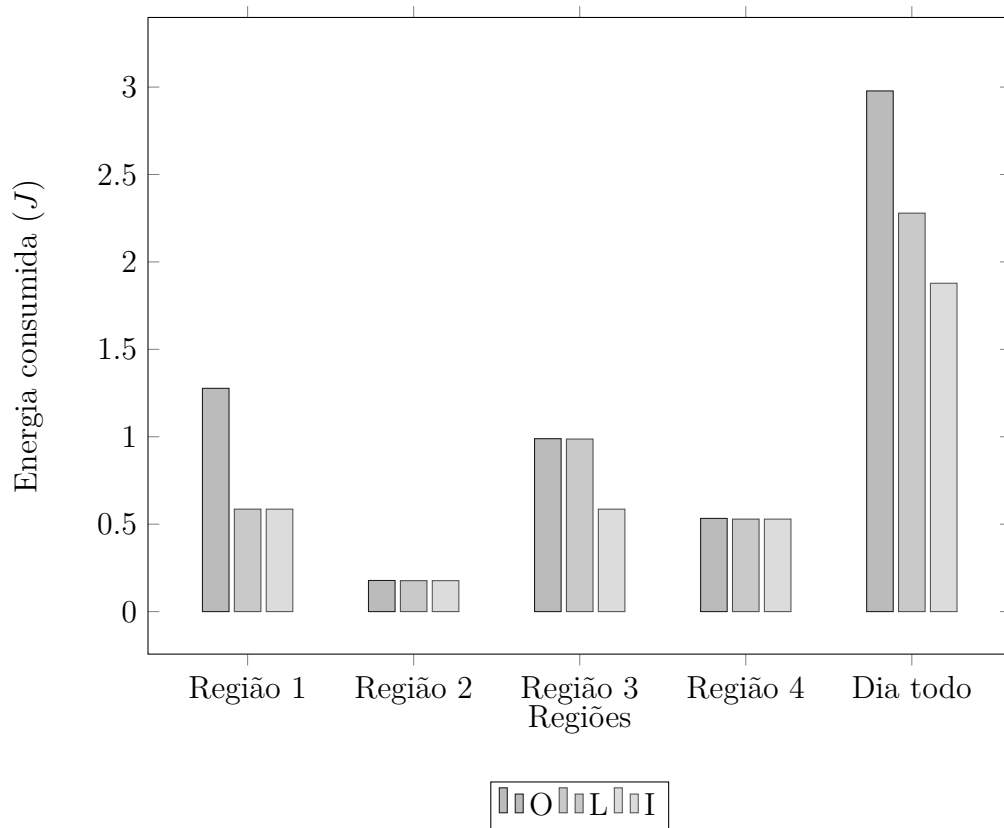


Figura 5.8: Consumo de energia, em J , dos dispositivos O, L e I em cada região e durante toda a simulação

espaço que apresenta atividade na maior parte do tempo. Uma vez que as regiões 1 e 3 têm a mesma duração, o impacto do dispositivo Wi-Fi (com um ciclo de trabalho de 50%) no consumo de energia pode ser observado, na região 1 o dispositivo O consome $1,277J$ enquanto na região 3 consome $0,989J$. Embora o rendimento alcançado na região 3 é de 60% do valor ideal, a energia consumida é de 77%, essa diferença deve-se aos pacotes que são enviados, mas não alcançam a estação central com a qualidade exigida, gastando assim a energia em uma transmissão sem êxito.

A Tabela 5.4 mostra a corrente média consumida por cada dispositivo em toda a simulação. Os dispositivos ativos em mais de um nível de atividade, como esperado, consomem mais corrente do que os outros. Devido à sua posição estratégica esses dispositivos estarão sempre ativos, dessa forma devem possuir uma bateria com maior capacidade de energia, a fim de evitar a falta de carga antes do restante da rede. Neste caso, a capacidade da bateria deve ser cerca de 60% maior para os dispositivos O, M e N, e 22% maior para os dispositivos G, K e J.

A Tabela 5.5 exhibe a duração média, em dias, de cada dispositivo considerando uma bateria com capacidade de $2600mAh$. A estação central (S) não está incluída pois, devido ao seu alto nível de atividade, seja necessário que esteja conectada a uma fonte de energia

Dispositivo	Energia	Dispositivo	Energia
O	0.559 <i>mA</i>	G	0.353 <i>mA</i>
N	0.561 <i>mA</i>	F	0.352 <i>mA</i>
M	0.561 <i>mA</i>	E	0.353 <i>mA</i>
L	0.428 <i>mA</i>	D	0.353 <i>mA</i>
K	0.428 <i>mA</i>	C	0.352 <i>mA</i>
J	0.428 <i>mA</i>	B	0.352 <i>mA</i>
I	0.353 <i>mA</i>	A	0.352 <i>mA</i>
H	0.353 <i>mA</i>	S	4.797 <i>mA</i>

Tabela 5.4: Média da corrente consumida por cada dispositivo durante a simulação

Dispositivo	Tempo de vida (dias)	Dispositivo	Tempo de vida (dias)
O	194	G	307
N	193	F	307
M	193	E	307
L	253	D	306
K	253	C	308
J	253	B	308
I	307	A	308
H	306		

Tabela 5.5: Tempo de vida de cada dispositivo considerando uma bateria de 2600*mAh*

externa. O tempo de vida é inferior a um ano para todos os dispositivos, variando entre 193 e 308 dias no melhor dos casos. É importante notar que os valores apresentados apenas representam o consumo das camadas de enlace e física do dispositivo, as camadas superiores não são levadas em conta, mas também consomem energia.

5.2.1 Autoaprendizagem

O modo operacional TSCH não fornece qualquer tipo de recurso de autoaprendizagem. Como mostrado pelos resultados da última seção, um dispositivo Wi-Fi causa grande impacto em uma rede de sensores sem fio, diminuindo drasticamente a vazão e aumentando o atraso, evidentemente o TSCH poderia se beneficiar de um método de autoaprendizagem.

O cenário proposto apresenta duas alterações na rede: o número de nós ativos e a presença intermitente de um dispositivo Wi-Fi perto do ponto central. A natureza determinística do TSCH não permite que ele se adapte de acordo com o número de nós ativos por ele próprio, é preciso que uma camada superior detecte e gerencia tal mudança. Uma adaptação às condições do meio é possível e pode ser feita pelo TSCH,

mudando os canais utilizados durante a comunicação para apenas aquelas que possuem uma quantidade aceitável de interferência.

De acordo com o padrão IEEE 802.15.4e, após entrar para uma rede as configurações de *slotframe* podem ser mudadas por uma camada superior. Na simulação proposta, a modificação feita por uma camada superior seria no tamanho do *slotframe*, modificando de 15 para 3 em uma região de baixa atividade, ou 6 em uma região de atividade média. Em uma região de alta atividade nenhuma mudança ocorreria uma vez que todos os dispositivos estão ativos.

Para mudanças no ambiente a rede pode se adaptar usando o LQI para avaliar a condição de cada canal. Como mostrado na Figura 5.5 os únicos canais não impactados pela interferência causada pelo dispositivo Wi-Fi são os 24, 25 e 26, apenas usando tais canais o desempenho da rede aumentaria consideravelmente.

A fim de alterar os canais disponíveis, a lista de salto de canais de cada dispositivo deve ser atualizada. É importante notar que para o correto funcionamento da rede todos os dispositivos devem ter a mesma lista de salto de canal. No caso do transmissor possuir uma lista de salto de canal diferente do receptor, cada um poderá estar utilizando um canal diferente e a comunicação não terá sucesso. Como já foi mencionado, isto é uma responsabilidade da camada superior e, neste estudo, considera-se que a alteração é feita com sucesso e todos os dispositivos possuem sempre a mesma lista.

Ambas adaptações são realizadas antes que o dispositivo Wi-Fi seja ligado e que o número de dispositivos ativos se altere. Em um ambiente real a rede necessita de algum tempo para detectar a interferência e assegurar que todos os dispositivos possuam a lista de salto de canal atualizada, neste caso, a variação entre regiões não ocorreria abruptamente, e sim de maneira suave através de algum período de tempo. O mesmo é válido para a adaptação do *slotframe*, os moradores da casa não iriam sair ou entrar de uma só vez.

A Figura 5.9 exhibe o impacto na vazão da rede quando aplicadas as adaptações da lista de salto de canal e do *slotframe*, é apresentado das técnicas isoladas e em conjunto:

- Adaptação do *slotframe*: A única região afetada é a terceira, relacionada ao nível de atividade médio. Como pode ser visto a vazão máxima é atingida, o aumento de *timeslots* disponíveis para cada dispositivo ativo, que vai de 1 em 0.15s para 1 em 0.06s, possibilitam que os pacotes que não foram enviados em períodos com interferência são enviados no período sem interferência.
- Adaptação da lista de salto de canal: a vazão da rede vai para o seu máximo em todas as regiões ao adaptar a lista de salto de canal para conter somente canais sem interferência, neste caso, todas as transmissões são bem-sucedidas. É evidente a necessidade de empregar esta técnica em uma rede de sensores sem fio.

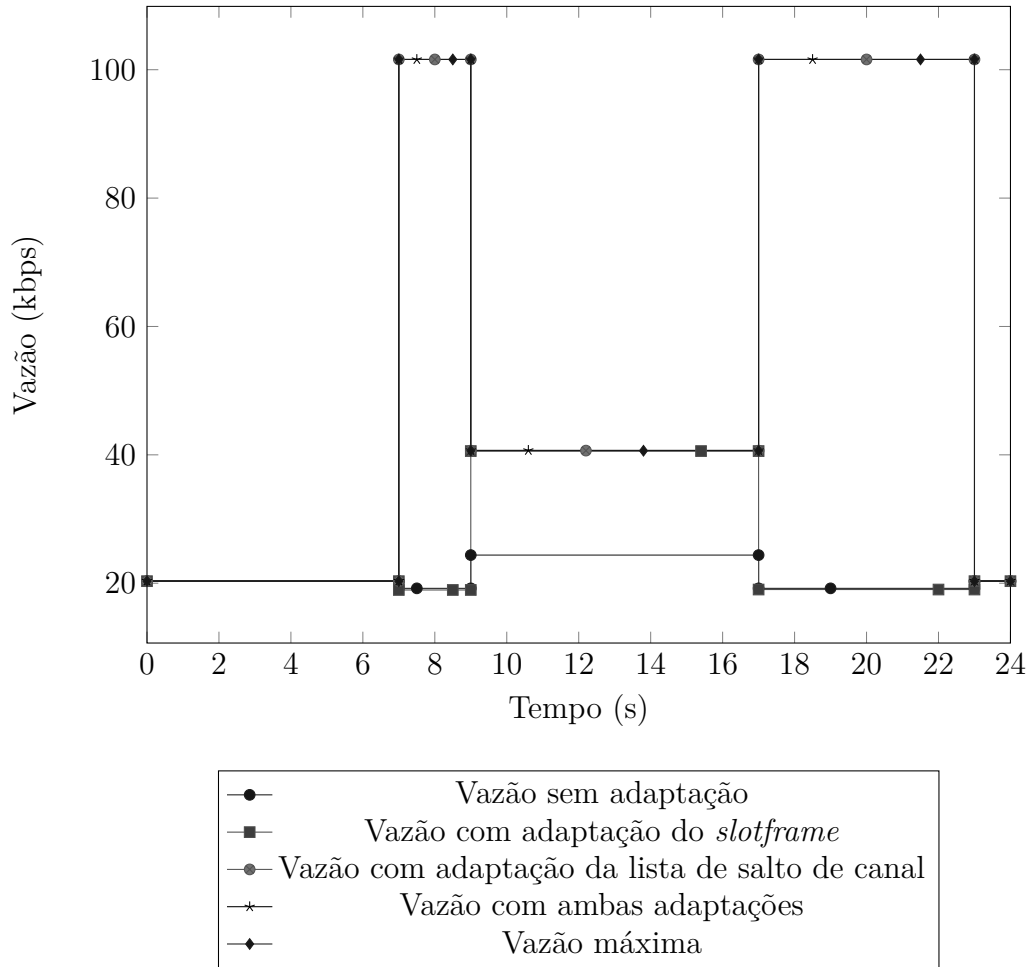


Figura 5.9: Vazão com e sem técnicas de autoaprendizagem

- Adaptações combinadas: em relação à vazão da rede a adaptação do *slotframe* não traz qualquer vantagem uma vez que a adaptação da lista de salto de canal faz com que a rede atinja o seu máximo em todas as regiões.

A Figura 5.10 exibe o impacto no atraso da rede quando aplicadas as adaptações da lista de salto de canal e do *slotframe*, é apresentado o impacto de cada técnica isolada e ambas as técnicas combinadas. Como mencionado anteriormente os *buffers* dos dispositivos são esvaziados depois de cada região para garantir que o desempenho da região anterior não afete a atual.

- Adaptação do *slotframe*: o atraso observado na primeira região, relativa ao nível baixo de atividade, diminui significativamente para 0.02638s, isto se dá devido a cada dispositivo ativo possuir cinco vezes mais *timeslots* disponíveis para fazer a transmissão. Na segunda região, em relação a um alto nível de atividade, não há alteração na configuração do *slotframe* e assim o atraso permanece o mesmo. Na terceira região, em relação a um nível de atividade médio, o atraso médio é de 0.1667

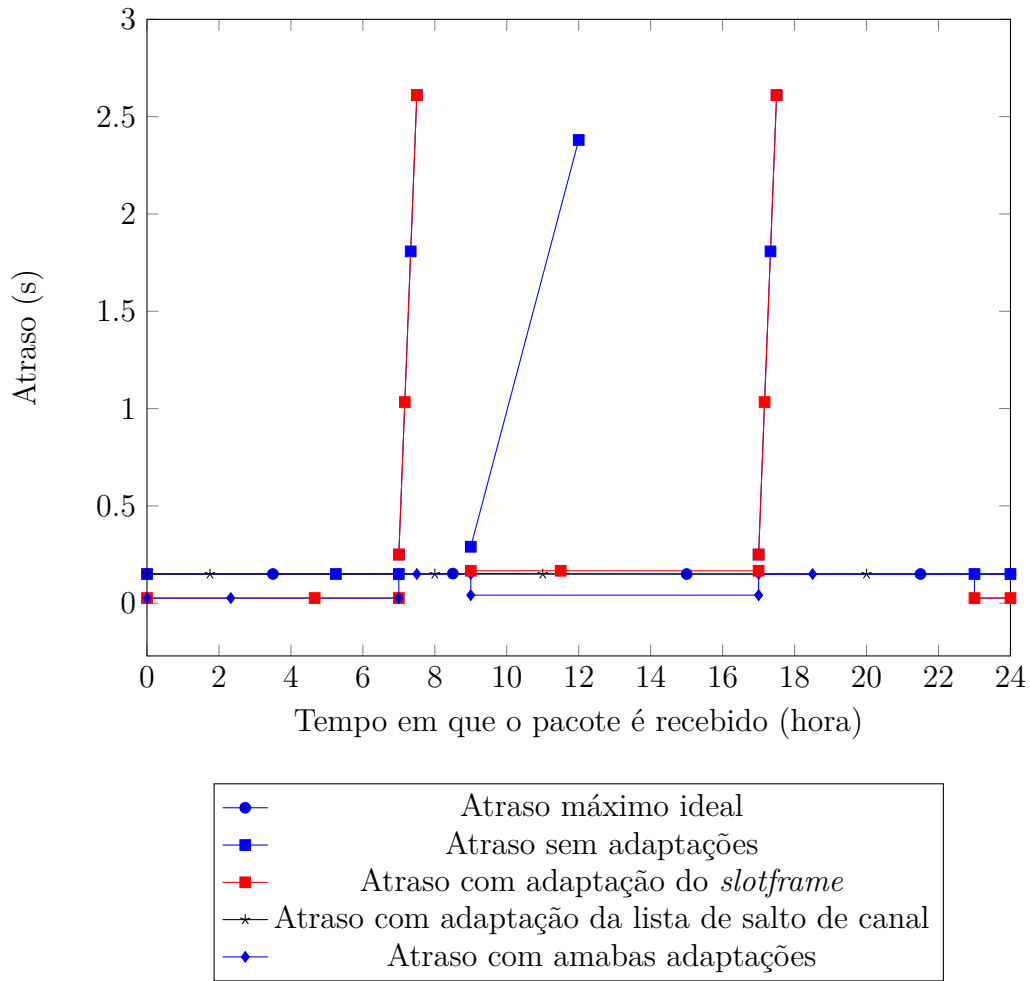


Figura 5.10: Atraso do dispositivo O com e sem técnicas de autoaprendizagem

$\pm 0.1243s$, embora não aumentando com o tempo ainda apresenta uma variação elevada, essa variação se deve à interferência intermitente no canal que bloqueia a maioria dos pacotes quando presente.

- Adaptação da lista de salto de canal: o atraso atinge $0.15s$ em todas as regiões ao alterar a lista de canais para apenas os canais que não sofram interferência, neste caso, todos os pacotes são enviados. Mais uma vez fica evidente a necessidade da aplicação dessa técnica na presença de interferência.
- Adaptações combinadas: ao empregar ambas as técnicas de autoaprendizagem o atraso é o mínimo observado, nas regiões 1 e 5 o atraso é tão baixo como observado com adaptação do *slotframe*, nas regiões 2, 3 e 4 o atraso é tão baixo quanto observado com adaptação do canal. Diferente da vazão, a adaptação do *slotframe* fornece melhorias únicas no desempenho do atraso.

É importante perceber que apesar de todos os gráficos mostrarem uma mudança

abrupta entre as regiões de atividade, num ambiente real tais alterações ocorrerão de forma suave durante um certo período de tempo, a adaptação também ocorreria de forma suave uma vez que a camada superior levaria um tempo para detectar as alterações e reconfigurar todos os dispositivos.

Os resultados apresentados nesta seção avaliam e confirmam a necessidade de autoaprendizagem em redes de sensores sem fio TSCH, ambas as técnicas de adaptação aumentam o desempenho da rede em diferentes aspectos e devem ser utilizadas em união para fornecer o melhor desempenho possível.

5.3 Resumo do capítulo

Neste capítulo o cenário de simulação projetado para observar o comportamento do modo TSCH em circunstâncias dinâmicas é apresentado, seguido pela análise dos resultados e por fim as melhorias que técnicas de autoaprendizagem possibilitam ao desempenho da rede. O cenário de simulação é composto por uma residência doméstica de 5 cômodos. Quinze sensores e uma central são dispostos por toda a casa. Um dispositivo Wi-Fi é inserido ao lado do nó central para causar interferência na rede de sensores e assim simular um ambiente dinâmico. A fim de representar mudanças nos requisitos da aplicação a carga gerada pelos sensores varia ao longo da simulação, representando a movimentação de um dia típico de uma família.

Os resultados obtidos mostram claramente que o impacto do sinal Wi-Fi degrada de forma significativa o desempenho da rede, que é analisado em termos de vazão, atraso e consumo de energia. O desempenho da rede chega a cair 81% em termos de vazão nos momentos de maior atividade. O atraso médio da rede aumenta com o tempo, chegando rapidamente a valores impraticáveis, pois os pacotes não conseguem ser enviados e os *buffers* dos sensores ficam saturados. O tempo de vida dos sensores que possuem maior atividade (da sala de estar, onde a atividade dos residentes é concentrada) é muito menor do que dos sensores de ambientes como quartos ou cozinha, evidenciando a necessidade de uma bateria de maior capacidade nesses dispositivos.

A lista de canais utilizados pelo *Channel Hopping* é alterada para utilizar apenas canais que não sofrem interferência da rede Wi-Fi, dessa forma o desempenho da rede alcança o nível necessário para o seu correto funcionamento. Alterando o tamanho do *slotframe* para comportar apenas dispositivos ativos é observada a queda do atraso médio da rede. Os resultados obtidos evidenciam a necessidade da implementação de técnicas de autoaprendizagem para que redes de sensores sem fio possam atuar de forma satisfatória em ambientes e aplicações dinâmicas.

Capítulo 6

Conclusão

O padrão IEEE 802.15.4 é o protocolo de fato para as camadas de enlace e física em redes de sensores sem fio. Sua última alteração foi lançada em 2012 e ainda carece de avaliação. Este trabalho implementou o modo operacional TSCH do 802.15.4e no conhecido simulador de rede de código aberto ns-3, no melhor de nosso conhecimento, esta foi a primeira implementação do 802.15.4e em um simulador conhecido. Utilizando o modelo desenvolvido, um estudo do desempenho do TSCH em cenários dinâmicos é realizado.

É fornecida uma visão geral das principais características definidas pelo padrão, incluindo as camadas de enlace e física. A emenda 802.15.4e também foi incluída, primeiramente apresentando a nova estrutura de elementos de informação e de salto de canal, em seguida, foram detalhados os três novos modos operacionais da camada de enlace:

- LLDN, um modo operacional que utiliza *beacons*, é recomendado para redes de baixa latência. O *superframe* é definido pelo coordenador, agendando *timeslots* para os nós se comunicarem com o coordenador;
- DSME, uma versão atualizada do modo *slotted* apresentado no 802.15.4, principalmente elimina a limitação de apenas 7 *timeslots* garantidos e inclui a possibilidade de múltiplos saltos no período GTS;
- TSCH, o objeto de estudo deste trabalho, um modo operacional em que um *superframe* repete-se ao longo do tempo, contendo os *timeslots* em que os dispositivos se comunicam de forma determinística.

O estado da arte avançado por este trabalho é descrito no capítulo 3, uma breve análise sobre ferramentas de simulação e seu valor na área de pesquisa em redes de computadores é apresentado, em seguida, a fim de escolher um simulador adequado para realizar este trabalho, o atual cenário de ferramentas de simulação que suportam o padrão IEEE 802.15.4 é apresentado.

No melhor de nosso conhecimento, atualmente não há nenhum simulador que ofereça suporte para a emenda 802.15.4e. A fim de avaliar os simuladores selecionados, eles foram classificados em código-livre e proprietário. Com base nos benefícios para a pesquisa acadêmica o simulador *Network Simulator*[17] foi escolhido. O *Network Simulator* possui duas versões diferentes em operação, ns-2 e ns-3. O ns-2 foi lançado pela primeira vez em 1996 e possui uma ampla gama de modelos suportados. Devido a estrutura ultrapassada do ns-2 o ns-3 foi desenvolvido, assegurando uma nova arquitetura mais adequada para a computação moderna, não há compatibilidade com os modelos do ns-2. Embora a atual implementação do padrão 802.15.4 seja melhor no ns-2, a versão ns-3 foi escolhida devido a sua estrutura moderna e crescente adoção por parte da comunidade acadêmica.

O modelo TSCH para ns-3 desenvolvido por este trabalho é baseado em um modelo incompleto do 802.15.4, a sua implementação atual está em desenvolvimento e suporta apenas algumas funcionalidades do padrão. A camada de enlace suporta apenas o modo *unslotted*, sem apoio para a formação de PAN e do modo *slotted*, já a camada física suporta apenas a faixa de frequência de $2,4GHz$, e não há suporte à interferência. A implementação carece de testes para verificar seu correto funcionamento, durante este trabalho alguns problemas foram encontrados em funcionalidades já implementadas, tais como o sistema de confirmação de recebimentos de pacotes.

A abstração chave do ns-3 é de um nó que contém vários aplicativos e dispositivos de rede. O modelo de espectro do ns-3 é usado para fornecer suporte a interferência, ele apresenta uma representação do espectro em função da frequência, suportando múltiplos modelos de perda e atraso de propagação. O modelo de espectro também oferece um dispositivo Wi-Fi que propaga um PSD compatível com o IEEE 802.11 [2] pelo do canal, ele é usado por este trabalho para simular a interferência causada por tal dispositivo. O modelo TSCH desenvolvido por este trabalho envolve a adaptação da implementação 802.15.4 atual para suportar as novas funcionalidades introduzidas pela emenda, tais como salto de canal e elementos de informação, a máquina de estado TSCH é implementada principalmente através da adição de métodos na classe *LrWpanMac*, que implementa todas as funções relacionadas à camada de enlace, a implementação segue rigorosamente a estrutura de serviços apresentada no protocolo. O suporte à interferência é adicionado à camada física usando componentes do modelo de espectro do ns-3. Foram feitas várias modificações no processo. A fim de facilitar o desenvolvimento de simulações a classe *LrWpanHelper* também é atualizada com vários novos métodos para criar e gerenciar *slotframes* e *timeslots*.

Neste trabalho é utilizado um modelo de energia para o modo TSCH. Utilizando os dados adquiridos durante a simulação foi possível calcular a energia consumida por cada dispositivo ao longo do cenário. O modelo define todos os tipos possíveis de *timeslots* e

a atividade do rádio e do micro controlador em cada tipo. A soma da energia consumida pelo rádio e micro controlador através das etapas de um *timeslot* representa a energia total durante esse tipo de *timeslot*. A energia total pode ser adquirida pela soma de cada *timeslot* pelo qual o dispositivo passa durante a simulação.

Os benefícios trazidos pela utilização de mecanismos de autoaprendizagem no modo TSCH são apresentados no capítulo 5, em que o cenário de simulação proposto é apresentado e os seus resultados são discutidos e validados. O cenário de simulação é representa uma rede doméstica de sensores acústicos sem fio que automatiza várias tarefas de uma casa, como desligar as luzes de cômodos vazios e notificar os moradores de possíveis acidentes. Quinze dispositivos estão distribuídos pelos cômodos da casa, com uma distância que varia entre 1,09 metros e 6,52 metros da estação central. Um dispositivo Wi-Fi foi colocado ao lado da estação central. Para simular mudanças no ambiente e nos requisitos da aplicação, uma linha do tempo é apresentada, tentando representar um dia comum em uma residência, as cargas da rede de sensores e do dispositivo Wi-Fi são variadas de acordo com a hora do dia.

Na apresentação dos resultados o impacto da interferência causada pelo dispositivo Wi-Fi é evidente. A fim de entender tal impacto, uma análise do BER da transmissão de cada dispositivo é realizada. Os resultados obtidos mostram que o dispositivo Wi-Fi causa grande impacto no desempenho da rede de sensores, bloqueando a maioria dos canais disponíveis para a comunicação. Isto se dá devido à máscara de espectro de uma transmissão Wi-Fi e a proximidade do dispositivo com a estação central.

Os resultados obtidos através da execução do cenário de simulação proposto são analisados em três aspectos: vazão, atraso e consumo de energia. A fim de medir o impacto da interferência causada pelo dispositivo Wi-Fi, uma análise sem interferência é fornecida para cada aspecto analisado, seguido por análises matemáticas do impacto da interferência a fim de validar os resultados apresentados.

Os resultados obtidos mostram que a rede de sensores sem fio não pode funcionar adequadamente quando estiver na presença de interferência Wi-Fi. O rendimento observado é muito influenciado pelo sinal Wi-Fi, diminuindo para 18,9% quando o sinal Wi-Fi estava ativo 100% do tempo e 60% quando ativo apenas 50% do tempo. O atraso também é afetado pela interferência. O atraso máximo sem interferência é 0,15s, em relação ao comprimento do *slotframe*, mas na presença do sinal Wi-Fi rapidamente aumenta para valores impraticáveis. A energia por bit consumida por cada dispositivo aumenta na presença de interferências, o que pode ser explicado pelos *timeslots* em que o dispositivo gastou energia ao falhar em transmitir um pacote, devido ao CCA detectando o meio ocupado ou o pacote não ser corretamente recebido pela estação central. O tempo de vida, em dias, de cada dispositivo foi estimado considerando uma bateria de 2600mAh,

neste caso, dispositivos ativos durante todo o dia apresentam uma vida útil de 193 dias, enquanto que os dispositivos menos ativos podem chegar a 308 dias antes de drenar a bateria.

Os benefícios adquiridos ao se utilizar técnicas de autoaprendizagem são apresentados na última seção do capítulo 5. No cenário apresentado a rede deve adaptar-se à interferência, mudando os canais utilizados pelo esquema de salto de canal e o tamanho do *slotframe* para suportar apenas os dispositivos ativos. A taxa de transferência atinge seu valor máximo quando adapta a configuração dos canais. Uma vez que apenas os canais que não sofrem interferência são utilizados, todas as transmissões são bem-sucedidas. O atraso atinge o valor desejado de 0,15s ao se aplicar a adaptação dos canais, mas quando se aplica a adaptação de *slotframe* ele diminui ainda mais, chegando a 0,027s, quando apenas três dispositivos estão ativos. Na presença de interferência a adaptação *slotframe* por si só não é suficiente para que a rede apresente um comportamento satisfatório.

Este trabalho mostra que para uma rede TSCH apresentar o comportamento esperado em um ambiente com interferência, é indispensável a implantação de técnicas de autoaprendizagem, principalmente restringindo os canais utilizados pelo esquema de salto de canal para apenas aqueles com uma qualidade mínima. Apresenta-se também que a mudança do tamanho do *slotframe* para incluir apenas dispositivos ativos pode trazer uma grande melhoria no atraso da rede. A análise deste trabalho foi possibilitada pelo desenvolvimento do modelo TSCH no simulador de redes ns-3, o qual no melhor do nosso conhecimento é o primeiro simulador a implementar o padrão IEEE 802.15.4e. Como resultado da implementação várias adições e correções também foram realizadas no modelo IEEE 802.15.4, com a adição principal sendo o suporte interferência.

Como trabalho futuro, sugere-se a implementação dos poucos recursos do modo TSCH ainda não desenvolvidos, tais como *timeslots* compartilhados e suporte à sincronização. A implementação completa do modelo IEEE 802.15.4e, incluindo os modos DSME e LLDN proporcionaria uma ferramenta para auxiliar a avaliação de todo o padrão e de novas propostas apresentadas pela comunidade acadêmica. Como mencionado no capítulo 5, a camada superior é a responsável por realizar as adaptações realizadas neste trabalho, como trabalho futuro deve-se estudar como identificar as mudanças no ambiente e aplicação a fim de realizar uma adaptação da rede.

Referências

- [1] Transmission control protocol. RFC 793, 1981. 22
- [2] IEEE Std. 802.11-2007. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Standard for Information Technology, 2007. vi, vii, x, 3, 29, 43, 60
- [3] IEEE Std. 802.15.4-2003. *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*. IEEE Standard for Information Technology, 2003. 2, 5
- [4] IEEE Std. 802.15.4-2006. *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*. IEEE Standard for Information Technology, 2006. 2, 5, 38
- [5] IEEE Std. 802.15.4-2011. *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*. IEEE Standard for Information Technology, 2011. vi, vii, x, 5, 6, 7, 8, 10, 12, 13, 14, 15
- [6] IEEE Std. 802.15.4e 2012. *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) Amendment 1: MAC sublayer*. IEEE Standard for Information Technology, 2012. vi, vii, x, 2, 5, 11
- [7] Nicola Baldo e Marco Miozzo. Spectrum-aware channel and phy layer modeling for ns3. In *Spectrum-aware channel and PHY layer modeling for ns3*, 2009. 26
- [8] David Crecraft e David Gorham. *Eletronics*. CRC Press, 2003. ISBN: 0748770364. 26
- [9] Peng Du e Dr. George Roussos. Adaptive time slotted channel hopping for wireless sensor networks. In *Computer Science and Electronic Engineering Conference (CEEC)*, 2012. 3, 23
- [10] Richard M. Fujimoto, Kalyan S. Perumalla, e George F. Riley. *Network Simulation*. Morgan & Claypool, 2005. ISBN: 1598291106, 9781598291100. 17
- [11] Khalid EL GHOLAMI, Najib ELKAMOUN, Kun Mean HOU, Yibo CHEN, Jean-Pierre CHANET, e Jianjin LI. A new wpan model for ns-3 simulator. In *New Information Communication Science and Technology for Sustainable Development: France-China International Workshop*, 2103. 32

- [12] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, 2005. ISBN: 0521837162, 9780521837163. 9, 26
- [13] IEEE. Ieee 802.15 wpanTMtask group 4 (tg4). URL: <http://www.ieee802.org/15/pub/TG4.html>, last checked on 2014-04-09. 2, 5
- [14] Red Hat Inc. Cygwin. <http://cygwin.com/>. 24
- [15] Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model, 1994. 2, 5, 22
- [16] J. D. Kraus. *Antennas*. McGraw-Hill, 1950. 29
- [17] ns 3 project. ns-2 network simulator. <http://www.isi.edu/nsnam/ns/>. 18, 60
- [18] ns 3 project. ns-3 network simulator. <http://www.nsnam.org/>. vi, 3, 18
- [19] Inc SCALABLE Network Technologies. Qualnet. <http://scalable-networks.com/>. 18
- [20] Mo Sha, Rahav Dor, Gregory Hackmann, Chenyang Lu, Tae-Suk Kim, e Taerim Park. Self-adapting mac layer for wireless sensor networks. In *Real-Time Systems Symposium (RTSS)*, 2013. 23
- [21] Tcl Core Team. Tool command language. <http://tcl.tk/>. 18
- [22] TECTOS. Netsim. <http://tetcos.com/>. 18
- [23] Xavier Vilajosana, Qin Wang, Fabien Chraim, Thomas Watteyne, Tengfei Chang, e Kristofer S. J. Pister. A realistic energy consumption model for tsch networks. In *IEEE SENSORS JOURNAL*, 2014. 38, 41, 50
- [24] T. Watteyne, X. Vilajosana, B. Kerkez, F. Chraim, K. Weekly, Q. Wang, e et al. Openwsn: A standards-based low-power wireless development environment. In *Trans. Emerging Telecommun. Technol*, volume 23, pages 480–493, 2012. 40