

As Redes de Baixa Capacidade e com Perdas (do inglês Low Power and Lossy Networks - LLNs) são comumente utilizadas nas aplicações de Internet das Coisas. O protocolo de roteamento IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) foi desenvolvido para LLNs estáticas. Este trabalho propõe o Mobility Aware RPL (MARPL), um mecanismo para a inclusão do suporte à mobilidade no protocolo RPL canônico. No MARPL, propõe-se a métrica, Variabilidade de Vizinhança. O MARPL adiciona um mecanismo de detecção de mobilidade baseada no monitoramento da variabilidade da vizinhança de um nó. Os resultados obtidos por experimentos computacionais mostram que o MARPL superou os trabalhos relacionados de Fotouhi et al. (2015), Kharrufa et al. (2017) e RPL (RFC 6550, 2012), em termos da taxa de sucesso de transmissão sem impactar negativamente na sobrecarga e consumo de energia.

Orientador: Dra. Janine Knies

JOINVILLE, 2019

ANO 2019 VINÍCIUS DE FIGUEIREDO MARQUES

ABORDAGEM PARA SUPORTE À MOBILIDADE NO PROTOCOLO RPL
BASEADO NA VARIABILIDADE DE VIZINHANÇA



UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
APLICADA – PPGCA

DISSERTAÇÃO DE MESTRADO

**ABORDAGEM PARA SUPORTE À MOBILIDADE NO
PROTOCOLO RPL BASEADO NA VARIABILIDADE
DE VIZINHANÇA**

VINÍCIUS DE FIGUEIREDO MARQUES

JOINVILLE, 2019

**UNIVERSIDADE DO ESTADO DE SANTA CATARINA - UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT
MESTRADO EM COMPUTAÇÃO APLICADA**

VINÍCIUS DE FIGUEIREDO MARQUES

**ABORDAGEM PARA SUPORTE À MOBILIDADE NO PROTOCOLO
RPL BASEADO NA VARIABILIDADE DE VIZINHANÇA**

JOINVILLE

2019

VINÍCIUS DE FIGUEIREDO MARQUES

**ABORDAGEM PARA SUPORTE À MOBILIDADE NO PROTOCOLO
RPL BASEADO NA VARIABILIDADE DE VIZINHANÇA**

Dissertação submetida ao Programa de Pós-Graduação em Computação Aplicada do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, para a obtenção do grau de Mestre em Computação Aplicada.

Orientador: Dra. Janine Kniess

JOINVILLE

2019

**Ficha catalográfica elaborada pelo programa de geração automática da
Biblioteca Setorial do CCT/UDESC,
com os dados fornecidos pelo(a) autor(a)**

Marques, Vinícius de Figueiredo
Abordagem para Suporte à Mobilidade no Protocolo RPL
Baseado na Variabilidade de Vizinhança / Vinícius de
Figueiredo Marques. -- 2019.
107 p.

Orientadora: Janine Kniess
Dissertação (mestrado) -- Universidade do Estado de
Santa Catarina, Centro de Ciências Tecnológicas, Programa
de Pós-Graduação em Computação Aplicada, Joinville, 2019.

1. Redes de Baixa Capacidade e Perdas. 2. Protocolo de
Roteamento. 3. RPL. 4. Mobilidade. I. Kniess, Janine. II.
Universidade do Estado de Santa Catarina, Centro de
Ciências Tecnológicas, Programa de Pós-Graduação em
Computação Aplicada. III. Título.

Abordagem para Suporte à Mobilidade no Protocolo RPL Baseado na Variabilidade de Vizinhança

por

Vinícius de Figueiredo Marques

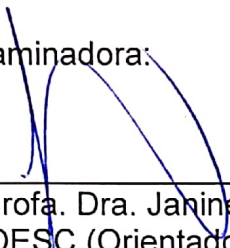
Esta dissertação foi julgada adequada para obtenção do título de

Mestre em Computação Aplicada

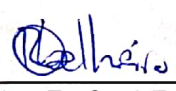
Área de concentração em “Ciência da Computação”,
e aprovada em sua forma final pelo

CURSO DE MESTRADO ACADÊMICO EM COMPUTAÇÃO APLICADA
DO CENTRO DE CIÊNCIAS TECNOLÓGICAS DA
UNIVERSIDADE DO ESTADO DE SANTA CATARINA.

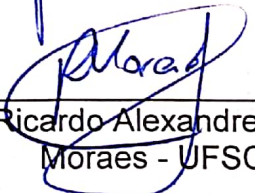
Banca Examinadora:



Prof. Dra. Janine Kniess
CCT/UDESC (Orientadora/Presidente)



Prof. Dr. Rafael Rodrigues Obelheiro
CCT/UDESC



Prof. Dr. Ricardo Alexandre Reinaldo de
Moraes - UFSC

Joinville, SC, 22 de março de 2019.

Dedico este trabalho aos meus pais, familiares, amigos e professores que sempre me apoiaram e incentivaram ao longo da minha vida acadêmica.

AGRADECIMENTOS

Gostaria de agradecer a minha mãe Maria Conceição de Figueiredo e meu pai Gastão Marques Filho por sempre me motivar a estudar e realizar os meus sonhos. Devo agradecimentos especiais aos meus familiares pelo incentivo e apoio.

Agradeço também a todos os professores que eu tive desde o ensino básico ao superior. Sou muito grato por tê-los como educadores. Agradeço a minha orientadora de iniciação científica durante os meus anos de graduação em Ciência da Computação na Universidade Federal de Mato Grosso, Karen da Silva Figueiredo. Dedico a ela minha iniciação na pesquisa acadêmica. Também devo agradecimentos ao professor Cristiano Maciel por ter me incentivado a escolher a UDESC para realizar o mestrado. Agradeço também a minha professora orientadora no mestrado, Janine Kniess, por me orientar durante a pesquisa sempre com muita paciência e dedicação.

"Internet of Things, sometimes referred to as the internet of Objects, will change everything – including ourselves"

Dave Evans

"If these out-of date beliefs are to be called myths, then myths can be produced by the same sorts of methods and held for the same sorts of reasons that now lead to scientific knowledge"

Thomas Samuel Kuhn

RESUMO

Internet das Coisas (do inglês *Internet of Things* - IoT) propõe a ideia de conectar dispositivos de baixa capacidade de processamento, memória e consumo energético à rede mundial de computadores. Tais dispositivos geralmente apresentam capacidade de sensoriamento e comunicação sem fio. As Redes de Baixa Capacidade e com Perdas (do inglês *Low Power and Lossy Networks* - LLNs) são comumente utilizadas nas aplicações de IoT. O protocolo de roteamento *IPv6 Routing Protocol for Low-Power and Lossy Networks* (RPL) foi desenvolvido para LLNs estáticas baseadas em IPv6. Este trabalho propõe o *Mobility Aware RPL* (MARPL), um mecanismo para a inclusão do suporte à mobilidade no protocolo RPL canônico. No MARPL, propõe-se a métrica, Variabilidade de Vizinhança. Esta métrica é utilizada pelo MARPL para determinar a mobilidade de um nó. O MARPL adiciona um mecanismo de detecção de mobilidade baseada no monitoramento da variabilidade da vizinhança de um nó. Este monitoramento é realizado levando-se em consideração o valor de *Receive Signal Strength Indicator* (RSSI) medido durante o recebimento de pacotes de dados e controle. Experimentos computacionais foram realizados no simulador Cooja (OSTERLIND et al., 2006), que disponibiliza dispositivos com o Sistema Operacional Contiki (DUNKELS et al., 2011). Os resultados obtidos com o simulador Cooja mostram que o MARPL superou os trabalhos relacionados de Fotouhi et al. (2015), Kharrufa et al. (2017) e RPL (RFC 6550, 2012), em termos da taxa de sucesso de transmissão sem impactar negativamente na sobrecarga e consumo de energia.

Palavras-chaves: Redes de Baixa Capacidade e Perdas, Protocolo de Roteamento RPL, Mobilidade.

ABSTRACT

Internet of Things (IoT) proposes the use of devices with low energy capacity, processing and memory power for a range of applications such as environmental monitoring and smart health. Such devices usually have sensing and wireless communication capabilities. Low Power and Lossy Networks (LLNs) with mobile devices is a common type of wireless network in IoT applications. The routing protocol IPv6 Routing Protocol for Low-Power and Lossy Network (RPL) was developed for LLNs composed by static nodes. This work proposes the Mobility Aware RPL (MARPL), a mechanism for RPL mobility support. MARPL proposes a metric and a mobility monitoring mechanism based on node neighbor variability. The MARPL mobility monitoring mechanism works by measuring the Received Signal Strength Indication (RSSI) upon the incoming of data or control packets. This work presents the results of computational experiments performed by the Cooja (OSTERLIND et al., 2006) simulator. Cooja provides virtual devices running the Contiki Operational System (DUNKELS et al., 2011). The experiment results demonstrate that MARPL outperforms Fotouhi et al. (2015), Kharrufa et al. (2017) and RPL (RFC 6550, 2012), in terms of packet delivery rate without overhead and energy consumption increasing.

Key-words: Low Power and Lossy Network, Routing Protocol RPL, Mobility.

LISTA DE ILUSTRAÇÕES

Figura 1.1 – Exemplo de Aplicação de LLN.	18
Figura 1.2 – Cenário com Mobilidade dos Nós em uma Topologia do RPL Canônico.	21
Figura 1.3 – Esquema de Trocas de Mensagens com Desconexão Causada pela Mobilidade de um Nó no RPL Canônico.	21
Figura 1.4 – Esquema de Trocas de Mensagens do Reingresso do Nó Móvel na Topologia DODAG.	22
Figura 2.5 – Exemplo de uma Topologia RPL.	34
Figura 3.6 – Classificação das Abordagens para o RPL com Topologias Móveis.	51
Figura 4.7 – Topologia LLN com Movimentação do Nó A.	72
Figura 5.8 – Exemplos de Topologias de LLN Utilizadas nas Simulações com 2 Sorvedouros.	79
Figura 5.9 – Análise da Taxa de Sucesso de Entrega (2 Sorvedouros).	81
Figura 5.10 – Análise do Atraso de Entrega de Pacotes (2 Sorvedouros).	82
Figura 5.11 – Análise do Atraso de Reconexão de Nó Pai (2 Sorvedouros).	83
Figura 5.12 – Análise da Sobrecarga de Mensagens de Controle (2 Sorvedouros).	84
Figura 5.13 – Análise da Energia Residual (2 Sorvedouros).	85
Figura 5.14 – Análise da Taxa de Sucesso de Transmissão Variando-se o Número de Sorvedouros (50 Nós).	86
Figura 5.15 – Análise da Sobrecarga de Mensagens de Controle Variando-se o Número de Sorvedouros (50 Nós).	87
Figura 5.16 – Análise do Atraso de Entrega de Pacotes Variando-se o Número de Sorvedouros (50 Nós).	88
Figura 5.17 – Análise do Atraso de Reconexão de Nó Pai Variando-se o Número de Sorvedouros (50 Nós).	89
Figura 5.18 – Análise da Perda de Pacotes Variando-se o Número de Sorvedouros (50 Nós).	90
Figura 5.19 – Análise da Energia Residual Variando-se o Número de Sorvedouros (50 Nós).	91
Figura 5.20 – Análise da Taxa de Sucesso de Transmissão Variando-se a Velocidade dos Nós (50 Nós).	92
Figura 5.21 – Análise do Atraso de Entrega de Pacotes Variando-se a Velocidade dos Nós (50 Nós).	93
Figura 5.22 – Análise do Atraso de Reconexão de Nó Pai Variando-se a Velocidade dos Nós (50 Nós).	94
Figura 5.23 – Análise da Sobrecarga de Mensagens de Controle Variando-se a Velocidade dos Nós (50 Nós).	95

Figura 5.24–Análise da Energia Residual Variando-se a Velocidade dos Nós (50 Nós).	96
Figura 5.25–Análise da Taxa de Sucesso de Transmissão no Cenário de 1km ² (100 Nós e Velocidade de 5,0 m/s).	96
Figura 5.26–Análise do Atraso de Reconexão de Nó Pai no Cenário de 1km ² (100 Nós e Velocidade de 5,0 m/s).	97
Figura 5.27–Análise da Sobrecarga de Mensagens de Controle no Cenário de 1km ² (100 Nós e Velocidade de 5,0 m/s).	97
Figura 5.28–Análise da Energia Residual no Cenário de 1km ² (100 Nós e Velocidade de 5,0 m/s).	98
Figura 5.29–Análise da Taxa de Sucesso de Transmissão do MARPL Variando-se o Valor de θ (50 Nós e Velocidade de 5,0 m/s).	99
Figura 5.30–Análise das Mensagens de Controle do MARPL Variando-se o Valor de θ (50 Nós e Velocidade de 5,0 m/s).	100
Figura 5.31–Análise do Atraso de Entrega do MARPL Variando-se o Valor de θ (50 Nós e Velocidade de 5,0 m/s).	100
Figura 5.32–Análise do Atraso de Reconexão de Nó Pai do MARPL Variando-se o Valor de θ (50 Nós e Velocidade de 5,0 m/s).	101
Figura 5.33–Análise da Energia Residual do MARPL Variando-se o Valor de θ (50 Nós e Velocidade de 5,0 m/s).	101

LISTA DE TABELAS

Tabela 3.1 – Comparação dos Trabalhos Relacionados: Mecanismos para o Envio de Mensagens de Controle DIO e DIS.	60
Tabela 3.2 – Comparação dos Trabalhos Relacionados: Mecanismos de Gerenciamento da Mobilidade, Métricas para Gerenciamento da Mobilidade ou Otimizadas na Função Objetivo.	61
Tabela 4.3 – Parâmetros do MARPL.	65
Tabela 5.4 – Parâmetros de Simulação	78

LISTA DE SIGLAS E ABREVIATURAS

AODV *Ad Hoc Distance Vector*

CTP *Collection Tree Protocol*

DAG *Directed Acyclic Graph*

DODAG *Destination Oriented Directed Acyclic Graph*

DPP *Data Packet Period*

DSR *Dynamic Source Routing Protocol*

ETX *Expected Transmission Count*

ICMPv6 *Internet Control Message Protocol version 6*

IEEE *Institute of Electrical and Electronics Engineers*

IETF *Internet Engineering Task Force*

IoT *Internet of Things*

LEACH *Low Energy Adaptive Clustering Hierarchy*

LLN *Low Power and Lossy Network*

LOP *Listen Only Period*

MRHOF *Minimum Rank with Hysteresis Objective Function*

ND *Neighbor Discovery*

OF0 *Objective Function Zero*

OLSR *Optimized Link State Routing*

PEGASIS *Power- Efficient Gathering in Sensor Information System*

RPL *IPv6 Routing Protocol for Low Power and Lossy Networks*

RSSF *Redes de Sensores sem Fio*

RSSI *Received Signal Strength Indicator*

SO *Sistema Operacional*

TEEN *Threshold Sensitive Energy Efficient Sensor Network*

UDGM *Unti Disk Graph Medium*

SUMÁRIO

1	INTRODUÇÃO	17
1.1	PROTOCOLOS DE ROTEAMENTO PARA IOT E SEUS REQUISITOS	19
1.2	OBJETIVOS	23
1.3	ESTRUTURA DO TRABALHO	24
2	FUNDAMENTAÇÃO TEÓRICA	25
2.1	INTERNET DAS COISAS	25
2.2	ROTEAMENTO EM REDES AD HOC MÓVEIS	27
2.3	ROTEAMENTO EM REDES DE SENSORES SEM FIO	28
2.4	ROTEAMENTO EM REDES DE BAIXA CAPACIDADE E PERDAS	30
2.5	CONSIDERAÇÕES SOBRE OS PROTOCOLOS DE ROTEAMENTO	32
2.6	PROTOCOLO <i>IPV6 ROUTING PROTOCOL FOR LOW-POWER AND LOSSY NETWORKS</i> (RPL)	33
2.6.1	Fundamentos do Protocolo RPL	33
2.6.1.1	<i>Métricas de Roteamento do RPL</i>	36
2.6.1.2	<i>Mensagens de Controle e Construção de Topologias no RPL</i>	37
2.6.1.3	<i>Funções Objetivos no Protocolo RPL</i>	39
2.6.2	Algoritmo <i>Trickle</i>	40
2.6.3	Mobilidade no RPL e seus Desafios	43
2.6.3.1	<i>Análise das Limitações na Detecção de Mobilidade e na Função Objetivo</i>	44
2.6.3.2	<i>Análise das Limitações do Mecanismo de Envio de Mensagens DIO</i>	45
2.6.3.3	<i>Análises das Limitações do Mecanismo de Envio de Mensagens DIS</i>	45
2.7	CONCLUSÕES DO CAPÍTULO	46
3	TRABALHOS RELACIONADOS	49
3.1	CLASSIFICAÇÃO DAS ABORDAGENS COM MOBILIDADE PARA O RPL	49
3.2	ABORDAGENS PARA GERENCIAMENTO DA MOBILIDADE NO RPL	52
3.2.1	Detecção Reativa de Mobilidade no RPL	53
3.2.2	Detecção Proativa de Mobilidade no RPL	56
3.3	CONSIDERAÇÕES SOBRE AS MÉTRICAS UTILIZADAS NA LITERATURA	58
3.4	CONCLUSÕES DO CAPÍTULO	59

4	ABORDAGEM PARA O SUPORTE A MOBILIDADE NO RPL	63
4.1	MECANISMO DE GERENCIAMENTO DE MOBILIDADE PARA O RPL	64
4.1.1	Monitoramento de Vizinhança no MARPL	66
4.2	FUNÇÃO OBJETIVO DO MARPL E ESCOLHA DO NÓ PAI	73
4.3	CONCLUSÕES DO CAPÍTULO	74
5	EXPERIMENTOS E RESULTADOS	77
5.1	ANÁLISE DO MARPL EM FUNÇÃO DO NÚMERO DE NÓS	80
5.2	ANÁLISE DO MARPL EM FUNÇÃO DO NÚMERO DE SORVEDOU- ROS	86
5.3	ANÁLISE DO MARPL EM FUNÇÃO DA VELOCIDADE DOS NÓS . .	91
5.4	ANÁLISE DA ESCALABILIDADE DO MARPL	94
5.5	ANÁLISE DO AJUSTE DO PARÂMETRO θ UTILIZADO PELO MARPL	98
5.6	CONCLUSÕES DO CAPÍTULO	102
6	CONCLUSÃO	103
6.1	TRABALHOS FUTUROS	104
6.2	OUTRAS CONTRIBUIÇÕES DA DISSERTAÇÃO	104
	REFERÊNCIAS	107

1 INTRODUÇÃO

Internet das Coisas propõe que os dispositivos presentes no dia-a-dia das pessoas estejam conectados na Internet. IoT possibilita que diversas aplicações, como o monitoramento ambiental, segurança pública, transporte e saúde inteligente possam ser amplamente desenvolvidas (GUBBI et al., 2013). Um dos principais objetivos de aplicações de IoT é a análise de uma vasta quantidade de dados coletados pelos dispositivos conectados na Internet. A análise de dados tem como objetivo a extração de informações para a tomada de decisão (OPPITZ; TOMSU, 2018). Em 2025, segundo Oppitz e Tomsu (2018), cerca de 40% das aplicações de IoT serão no setor de negócios e manufatura, e cerca de 30% serão na setor de saúde.

Os dispositivos utilizados em aplicações de IoT comumente possuem capacidade de sensoriamento e de comunicação sem fio, podendo formar uma Rede de Sensores sem Fio (RSSF), geralmente de baixo alcance (OPPITZ; TOMSU, 2018). Estes dispositivos podem apresentar restrições em termos de autonomia energética, baixa capacidade de processamento e memória (GUBBI et al., 2013).

A limitação no alcance do rádio de comunicação adotado pelos dispositivos em IoT demanda um modelo salto-a-salto de comunicação. Este modelo requer a utilização de um protocolo de roteamento que apresente eficiência principalmente em termos de taxa de entrega de pacotes e atraso fim-a-fim.

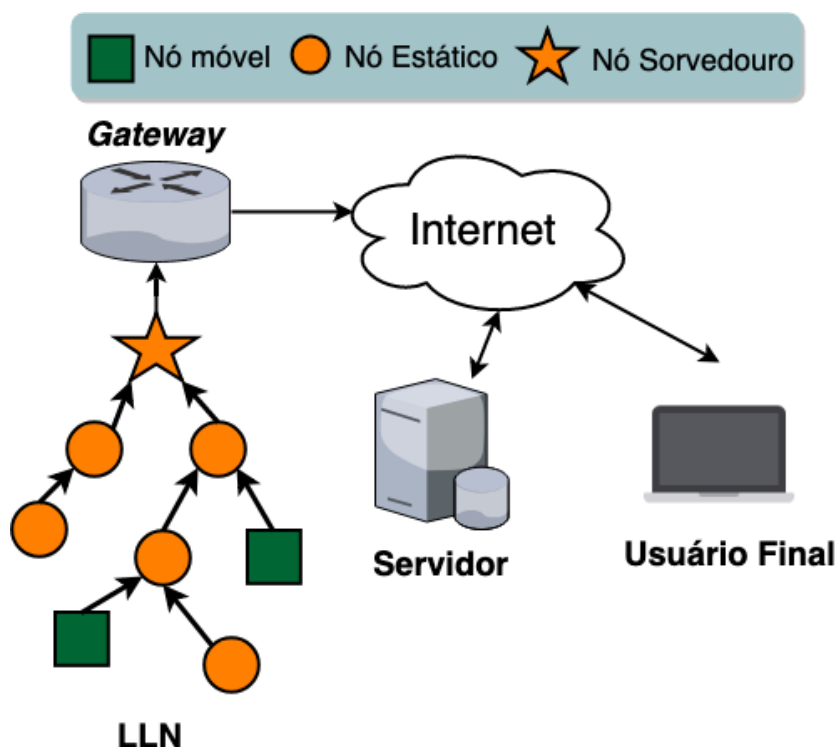
O suporte à mobilidade é um dos principais requisitos em diversas aplicações de IoT, como por exemplo, nos ambientes industriais, saúde inteligente e monitoramento em áreas afetadas por desastres naturais (IOVA et al., 2016). Contudo, o gerenciamento da rede com mobilidade é mais complexo se comparado com as redes que possuem apenas nós estáticos. A comunicação salto-a-salto e a mobilidade na rede adicionam a necessidade da rede ser auto-configurável e resiliente às mudanças na topologia.

Uma classe de redes de sensores largamente utilizada em IoT é a *Low Power and Lossy Network* (LLN) (PAUL; SARASWATHI, 2017). As LLNs são caracterizadas por serem compostas de dispositivos com baixa capacidade de armazenamento, processamento e energia. Os enlaces nas LLNs operam com baixa taxa de transferência e o padrão de comunicação geralmente é do tipo um ponto para muitos pontos, do inglês *Point-to-Multipoints* (P2MP), ou de muitos pontos para um ponto, do inglês *Multipoint-to-Points* (MP2P) (RFC 6550, 2012). Nas redes do tipo LLNs, a topologia pode ser composta de nós estáticos, móveis ou ambos. As LLNs utilizam tecnologias de comunicação a exemplo do padrão IEEE 802.15.4 (IEEE, 2016b). Este padrão foi

projetado pelo grupo *Institute of Electrical and Electronics Engineers* (IEEE), especificamente desenvolvido para dispositivos de baixa capacidade e com rádios de baixo alcance e taxa de transferência (IEEE, 2016b).

Por exemplo, na Figura 1.1 apresenta-se uma possível topologia LLN composta de nós sensores estáticos e móveis distribuídos em um ambiente de monitoramento. Uma LLN é geralmente composta de um ou mais nós sorvedouros, responsáveis pela transmissão dos dados coletados pelos nós sensores (móveis e estáticos) para um servidor através da Internet. A adição de um *gateway* permite a comunicação da LLN com a Internet. Um exemplo de utilização dos dados coletados pela LLN é a análise dos dados para que os usuários finais possam utilizar as informações processadas para a tomada de decisão.

Figura 1.1 – Exemplo de Aplicação de LLN.



Fonte: O autor.

Robustez diante das mudanças na topologia, alta taxa de entrega, baixo atraso de transmissão e mobilidade são requisitos de diversas categorias de aplicação de IoT. Portanto, os protocolos de roteamento para IoT devem ser robustos o bastante para manter sua eficiência em termos de rápida recuperação ou prevenção de inconsistências na topologia, como ciclos e desconexões causadas principalmente pela mobilidade dos nós.

A mobilidade dos nós aumenta a complexidade do roteamento, pois as mudanças na topologia são mais frequentes, quando comparada com as topologias estáticas (ZHAO et al., 2017). As mudanças na topologia causam um aumento no número de desconexões entre os nós da rede. Conseqüentemente, um protocolo de roteamento para LLNs necessita de mecanismos eficientes para a rápida detecção da mobilidade dos nós. A partir da detecção da mobilidade, um protocolo de roteamento pode tomar medidas a fim de minimizar os impactos das desconexões como perda de pacotes e atraso.

1.1 PROTOCOLOS DE ROTEAMENTO PARA IOT E SEUS REQUISITOS

Diversas aplicações de Internet das Coisas formam topologias de rede compostas de dispositivos estáticos e móveis (GARA et al., 2016). Além do suporte à mobilidade, estas aplicações requerem que a rede apresente bom desempenho em termos de atraso fim-a-fim e robustez (RAULT; BOUABDALLAH; CHALLAL, 2014).

O protocolo de roteamento *IPv6 Routing Protocol for Low-Power and Lossy Networks* (RPL) proposto pelo grupo *Internet Engineering Task Force* (IETF) (RFC 6550, 2012) foi inicialmente desenvolvido para redes estáticas de baixa potência e com perdas. O protocolo RPL vem ganhando destaque como protocolo de roteamento para as LLNs, pois permite a utilização do protocolo de rede *Internet Protocol version 6* (IPv6) nos dispositivos de IoT. A utilização do IPv6 viabiliza que uma vasta quantidade de dispositivos sejam endereçáveis na Internet.

O protocolo RPL utiliza o conceito de grafo direcionado, também conhecido como *Destination-Oriented Directed Acyclic Graph* (DODAG) para estruturar a topologia formada pelos nós em uma LLN (RFC 6550, 2012). Há dois tipos de nós em uma LLN executando o RPL: (i) o primeiro tipo é o nó sorvedouro, responsável por iniciar a construção da topologia através da divulgação de sua DODAG com mensagens de controle *DODAG Information Object* (DIO). Os nós sorvedouros, geralmente são definidos previamente, antes do funcionamento da LLN; (ii) o segundo tipo são os nós responsáveis pela coleta e transmissão de dados (chamados de nós coletores), salto-a-salto, em direção ao nó sorvedouro. Os nós coletores também transmitem mensagens DIO para a manutenção na topologia.

Na Figura 1.1, apresenta-se um exemplo de uma DODAG constituída de um nó sorvedouro e diversos nós sensores estáticos e móveis. Os nós sensores devem coletar os dados e transmiti-los para o nó sorvedouro. Os enlaces entre os nós da DODAG são representados por setas. O nó de onde a seta parte é chamado nó filho. O nó pai é o que recebe a seta (onde a seta chega).

Ao receber uma mensagem de controle DIO, um nó pode ingressar na DODAG

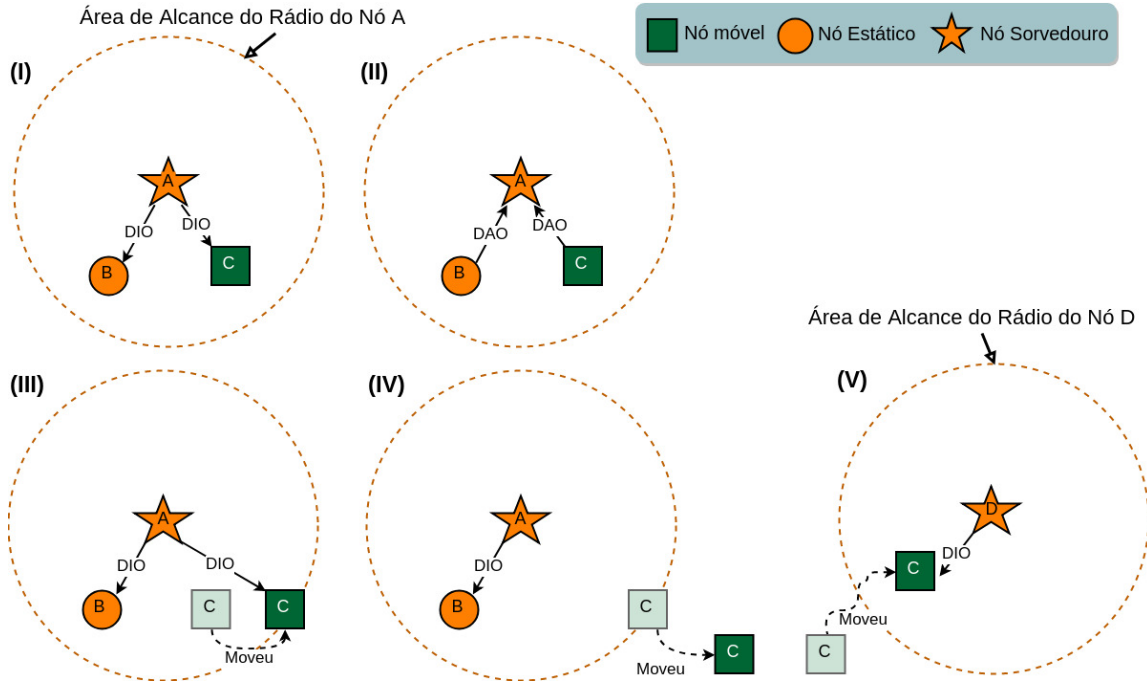
divulgada pela mensagem. O ingresso de um nó em uma DODAG se dá através de um valor que determina o quão distante o nó originador da mensagem DIO está do nó sorvedouro. Este valor é chamado de ranqueamento (*rank*) do nó originador da mensagem DIO. A seleção da melhor rota no RPL visa minimizar este valor. Ou seja, a rota com menor *rank* é a preferida. Para que a seleção de um novo nó pai (ou seja, uma nova rota para o nó sorvedouro da DODAG) possa acontecer, o nó em questão deve receber mensagens DIO (de outros nós) com melhor *rank*, quando comparado ao *rank* do nó pai atual. Para confirmar o reingresso na topologia, o nó deve transmitir uma mensagem do tipo *Destination Advertisement Object* (DAO) de volta ao nó pai. O nó que é utilizado como rota para o nó sorvedouro é chamado de nó pai pelo nó que ingressa na DODAG. Este último é chamado de nó filho (RFC 6550, 2012).

Não obstante, há diversas questões importantes a serem resolvidas no protocolo RPL em relação à mobilidade dos nós. Nas redes móveis, o RPL apresenta baixa taxa de sucesso de entrega de mensagens, alto atraso fim-a-fim e alto atraso de reingresso dos nós móveis na topologia (FOTOUHI; MOREIRA; ALVES, 2015). Para que o RPL tenha bom desempenho em cenários com nós móveis, faz-se necessário que o mesmo utilize algum mecanismo para o monitoramento da mobilidade dos nós. Através da detecção da mobilidade, os nós podem tomar as medidas necessárias para que as consequências oriundas das desconexões causadas pela mobilidade possam ser minimizados.

Para elucidar as limitações do protocolo de roteamento RPL no gerenciamento de topologias com nós móveis, considere o exemplo representado pela Figura 1.2. Os nós **A**, **B** e **D** são estáticos e o nó **C** é móvel. Na Figura 1.2.I o nó **A** transmite mensagens DIO e ambos os nós **B** e **C** a recebem, pois estão dentro da área de alcance do nó **A** (a área máxima de transmissão do nó **A** é representada pelo círculo com bordas tracejadas). Os nós **B** e **C** ingressam na topologia divulgada por **A** e respondem com uma mensagem DAO como pode ser visto na Figura 1.2.II. Na Figura 1.2.III, o nó **C** se movimenta, porém ainda continua dentro da área de cobertura de **A**, podendo assim receber a segunda mensagem DIO transmitida por **A**. Na Figura 1.2.IV, o nó **C** se movimenta novamente e sai da área de cobertura de **A**, e por não haver um mecanismo de gerenciamento de mobilidade no RPL canônico, o nó **C** não tem conhecimento de sua movimentação e que saiu da área de cobertura de seu nó pai **A**. Portanto, todas as mensagens transmitidas por **C** serão perdidas enquanto o mesmo não receber uma mensagem DIO de outro nó cujo *rank* seja melhor do que seu antigo nó pai **A**, como por exemplo, do nó **D** (veja a Figura 1.2.V). A atualização do nó pai ocorrerá no RPL canônico, somente quando o nó **C** receber uma mensagem DIO com um *rank* melhor do que seu antigo nó pai.

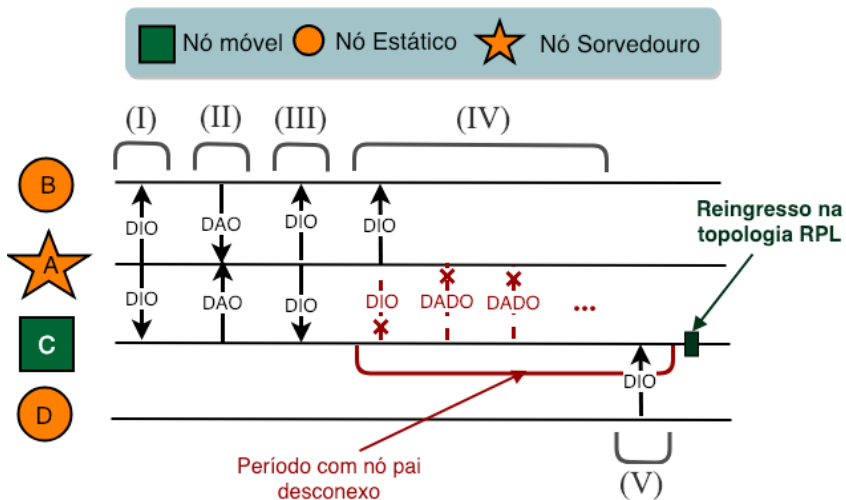
Na Figura 1.3, apresenta-se um esquema de troca de mensagens entre os

Figura 1.2 – Cenário com Mobilidade dos Nós em uma Topologia do RPL Canônico.



Fonte: O autor.

Figura 1.3 – Esquema de Trocas de Mensagens com Desconexão Causada pela Mobilidade de um Nó no RPL Canônico.



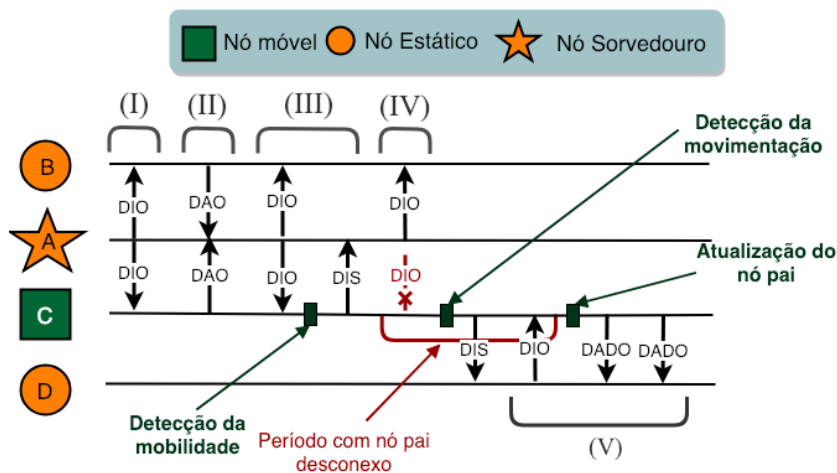
Fonte: O autor.

nós apresentados na Figura 1.2. Como demonstrado na Figura 1.2.IV, a partir deste momento o nó móvel **C** sai da área de alcance do nó sorvedouro **A**. Contudo, o nó **C** continua considerando que a conexão com o nó **A** ainda existe e quaisquer mensa-

gens de dados enviados serão perdidas, como é demonstrado na Figura 1.3.IV. Este comportamento, deve-se ao fato de que o nó **C** não detectou sua movimentação e a desconexão com o nó **A**. Quanto maior for o período de desconexão do nó móvel com seu nó pai, maior será a perda de pacotes. Como pode ser visto na Figura 1.3, o reingresso do nó móvel **C** à topologia RPL acontece somente em 1.3.V, quando o nó **D** divulga sua DODAG através da mensagem de controle DIO. Deste modo, no RPL canônico o período de desconexão do nó móvel **C** será influenciado pela periodicidade da transmissão de mensagens de controle DIO de outros nós em sua vizinhança.

Na Figura 1.4, apresenta-se a troca de mensagens entre os nós da topologia apresentada na Figura 1.2, utilizando-se de um mecanismo de gerenciamento de mobilidade. O mecanismo de detecção da inalcançabilidade do nó pai pode ser composto de duas fases. A primeira é a detecção da mobilidade e a segunda é a rápida atualização de nó pai. A segunda fase é dependente da ocorrência da primeira. A detecção da mobilidade do nó móvel **C** ocorre no momento descrito na Figura 1.2.III. O próximo passo é iniciar rapidamente o procedimento de manutenção local da topologia. Através do gerenciamento de mobilidade, o nó móvel envia mensagens de controle *DODAG Information Solicitation* (DIS) para todos os nós em sua vizinhança, a fim de reconhecer os nós candidatos em sua vizinhança. Desde modo, o nó móvel pode atualizar o nó pai rapidamente. Por exemplo, o nó **C** poderá escolher o nó **D** como novo nó pai mais rapidamente, como pode ser visto na Figura 1.4.

Figura 1.4 – Esquema de Trocas de Mensagens do Reingresso do Nó Móvel na Topologia DODAG.



Fonte: O autor.

O RPL canônico não possui um mecanismo de detecção ou previsão de mobilidade nativamente. Como pode ser notado na Figura 1.2.IV, quando o nó móvel **C** sai da área de cobertura do nó **A**, o nó **C** permanece um longo período desconectado

da topologia e durante este período pode haver perda de pacotes, veja a Figura 1.3.IV. O nó **C** somente reingressa na topologia após ter recebido uma mensagem DIO do nó **D**. No RPL canônico, a periodicidade do envio de mensagens DIO diminui exponencialmente com o passar do tempo. Logo, o nó **C** poderá esperar por um período consideravelmente longo por uma mensagem DIO de outro nó. Quanto maior este tempo de espera, maior pode ser o impacto no desempenho da rede em termos de congestionamento e perda de pacotes.

Segundo a especificação canônica do RPL, mensagens DIS podem ser utilizadas pelos nós para encontrar novas rotas para o nó sorvedouro.

1.2 OBJETIVOS

O objetivo geral deste trabalho é adicionar um mecanismo para gerenciamento da mobilidade no protocolo de roteamento RPL. O mecanismo proposto, denominado de *Mobility Aware RPL* (MARPL), pretende identificar rapidamente a movimentação dos nós e permitir uma rápida atualização da topologia RPL a fim de minimizar o impacto das desconexões. Pretende-se que tal medida beneficie o desempenho do protocolo em topologias móveis. O MARPL consiste de três partes: (i) detecção da mobilidade dos nós através da utilização de uma nova métrica chamada Variabilidade de Vizinhança; (ii) mecanismo de detecção da inalcançabilidade do nó pai e (iii) ajuste no temporizador *Trickle* utilizado pelo RPL canônico. O MARPL baseia-se na premissa de que todos os nós da LLN podem ser móveis, permanecendo estáticos por alguns períodos de tempo.

Os objetivos específicos são:

- Apresentar uma fundamentação teórica dos protocolos de roteamento para Internet das Coisas;
- Analisar o comportamento do protocolo de roteamento RPL em face da mobilidade, tendo em vista que o mesmo foi inicialmente modelado para redes estáticas;
- Desenvolver, implementar e analisar o desempenho de uma estratégia para que a mobilidade seja identificada rapidamente pelo RPL;
- Desenvolver, implementar e analisar um mecanismo para regular a periodicidade da transmissão de mensagens de controle, de modo a permitir a diminuição do impacto das desconexões entre os nós.

1.3 ESTRUTURA DO TRABALHO

A organização do trabalho é dividida em mais cinco capítulos. No Capítulo 2, apresenta-se a fundamentação teórica na qual são descritos os protocolos de roteamento propostos para Internet das Coisas. No Capítulo 3 são apresentados os trabalhos relacionados ao RPL em relação à mobilidade. A abordagem proposta neste trabalho, MARPL, é apresentada no Capítulo 4; sendo composta por: mecanismo de gerenciamento de mobilidade, mecanismo para ajustar o temporizador *Trickle*, e a otimização da troca do nó pai. As metodologias utilizadas para a experimentação da abordagem proposta, as comparações com o RPL canônico (RFC 6550, 2012), com a proposta de Fotouhi et al. (2015) e com Kharrufa et al. (2017) são apresentadas no Capítulo 5. Por fim, no Capítulo 6, apresenta-se as conclusões obtidas e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, apresenta-se uma fundamentação teórica sobre Internet das Coisas, do inglês *Internet of Things* (IoT), os protocolos de roteamento para redes *ad hoc* móveis e redes de sensores sem fio comuns em aplicações de IoT e suas limitações ao serem utilizados em *Low Power and Lossy Networks* (LLNs). Também apresentam-se as características de funcionamento do protocolo *IPv6 Routing Protocol for Low-Power and Lossy Networks* (RPL) e os desafios para sua aplicação em redes móveis.

2.1 INTERNET DAS COISAS

Internet das Coisas tem como principal objetivo integrar diversos dispositivos com capacidades de sensoriamento e comunicação à internet. Dentre os avanços tecnológicos e econômicos que permitiram a expansão das aplicações de IoT, cita-se (OPPITZ; TOMSU, 2018): (i) a redução dos custos dos componentes utilizados pelos dispositivos de IoT, como *microchips*, sensores de *Global Positioning System* (GPS) e acelerômetros; (ii) o aumento do espaço de endereçamento da internet através da utilização do *Internet Protocol version 6* (IPv6) em detrimento da utilização do *Internet Protocol version 4* IPv4. Uma maior capacidade de endereçamento faz-se necessário devido à larga quantidade de dispositivos que terão conexão com a internet em aplicações de IoT; e (iii) o aumento na disponibilidade e redução dos custos de serviços em nuvem como armazenamento de dados e hospedagem de sistemas web.

Um dispositivo de IoT é comumente composto por dois elementos: (i) sensoriamento ou atuação; (ii) conectividade e comunicação sem fio. O componente de sensoriamento é responsável pela coleta de dados, enquanto que o componente de atuação pode ser utilizado pelo dispositivo para interagir com o mundo físico (SETHI; SARANGI, 2017). O componente de comunicação utilizado por dispositivos de IoT pode variar de acordo com a tecnologia de comunicação utilizada.

Em termos da comunicação, os dispositivos de IoT podem formar diferentes topologias dependendo da aplicação e da tecnologia utilizada para a comunicação (YAQOOB et al., 2017). Um projeto de aplicação de IoT deve levar em consideração alguns atributos da aplicação para determinar qual tipo de tecnologia de comunicação sem fio é a mais adequada: latência, taxa de transmissão, tolerância à falha, escalabilidade, números de saltos e alcance do rádio (YAQOOB et al., 2017). Devido às limitações nos recursos dos dispositivos, a comunicação apresenta alguns desafios: endereçamento e identificação, baixo consumo energético pelo componente de comu-

nicação, protocolos de roteamento eficientes e mobilidade dos nós (SETHI; SARANGI, 2017).

Nos últimos anos, padrões de rádio de baixo consumo energético vem sendo projetados para os dispositivos de IoT. O grupo IEEE 802.15 pertencente ao comitê de padrões IEEE 802 do *Institute of Electrical and Electronics Engineers* (IEEE) trabalha em projetos de padrões para *Wireless Personal Area Network* (WPAN). Estes padrões compreendem protocolos para a camada física e de acesso ao meio. O padrão de interesse para as LLNs é o IEEE 802.15.4, pois este padrão de rádio foi desenvolvido visando aplicações com restrições energéticas e baixa taxa de transmissão. Os principais objetivos desse padrão são a facilidade de instalação, transferência confiável de dados, operação de curto alcance, baixo custo e consumo energético (IEEE, 2016b).

O grupo IEEE 802.15 possui outros projetos de padrões para WPANs, como por exemplo: o 802.15.6 que é destinado às *Body Area Networks* (BAN) (IEEE, 2012). As redes BAN são de curto alcance e permitem a comunicação sem fio por dispositivos nas proximidades ou dentro do corpo humano. Cita-se como outro exemplo, o padrão 802.15.3, desenvolvido para as WPANs que apresentam alta taxa de transmissão (IEEE, 2016a).

O endereçamento e a identificação única dos dispositivos são questões importantes em LLNs. O protocolo de rede IPv6 permite o endereçamento de um número muito grande de dispositivos (RFC 4944, 2007), sendo atualmente o melhor esquema de endereçamento no contexto de IoT (OPPITZ; TOMSU, 2018). Porém, as LLNs operam em tecnologias de comunicação de baixa capacidade, por exemplo, o IEEE 802.15.4. Devido ao fato dos pacotes IPv6 ocuparem um espaço maior do que um quadro do 802.15.4 consegue suportar, mecanismos de adaptação são necessários para que os pacotes IPv6 possam ser transmitidos eficientemente nas LLNs. Motivados por esta questão, o grupo IETF propôs um protocolo de adaptação chamado *IPv6 over Low-Power Wireless Personal Area Networks* (6LoWPAN). O principal objetivo do 6LoWPAN é permitir que dispositivos empregados em LLN tenham conexão IPv6 de maneira eficiente (RFC 4944, 2007).

Nas próximas seções, discutem-se alguns protocolos clássicos de roteamento, no contexto de *Mobile Ad Hoc Network* (MANET) e Redes de Sensores sem Fio (RSSF). As limitações para o uso desses protocolos em LLNs também são discutidas. As MANETs e as RSSFs assemelham-se as LLNs devido ao fato de transmitirem as informações salto-a-salto sem a necessidade de uma configuração infraestruturada que exige, por exemplo, a presença de um ponto de acesso. Porém, os protocolos de roteamento para MANETs apresentam limitações no seu uso em RSSFs, como alta sobrecarga e consumo energético.

2.2 ROTEAMENTO EM REDES AD HOC MÓVEIS

Nesta seção, apresentam-se protocolos para *Mobile Ad Hoc Networks* (MANETS), e discute-se os desafios de suas operações em redes de baixa capacidade e com perdas como as LLNs.

O protocolo *Ad hoc On-Demand Distance Vector* (AODV) foi proposto para redes *ad hoc* móveis (RFC 3561, 2003). Este protocolo segue uma abordagem reativa para a descoberta de rotas. A descoberta de rota ocorre através do uso de três pacotes de controle: *Route Requests* (RREQs), *Route Replies* (RREPs) e *Route Errors* (RERRs). Um nó inicia o processo de descoberta de rota assim que necessitar transmitir um pacote e quando não existir uma rota disponível em sua tabela de roteamento. Para tal, o nó propaga uma mensagem de requisição de rota (RREQ) em *multicast*. O nó destinatário, ao receber um pacote RREQ, transmite uma mensagem de resposta (RREP) em *unicast* para o nó originador da requisição. Quando o nó originador receber o pacote RREP, o procedimento de descoberta de rotas é terminado. Um pacote do tipo RERR é utilizado quando uma rota encontra-se indisponível para notificar outros nós que uma desconexão aconteceu (RFC 3561, 2003). No protocolo AODV existe a necessidade de haver constantes inundações na rede por mensagem de controle para que rotas possam ser descobertas e atualizadas, contribuindo para um aumento na sobrecarga da rede, fato este que limita seu uso em LLNs. Ressalta-se que este protocolo não foi projetado para suportar o endereçamento fornecido pelo *IPv6 over Low-Power Wireless Personal Area Networks* (6LoWPAN), que é uma versão comprimida do IPV6 para atuar em redes 802.15.4.

Dynamic Source Routing Protocol (DSR) é um protocolo composto de dois mecanismos: descoberta e manutenção de rota (RFC 4728, 2007). Assim como o protocolo AODV, o DSR é um protocolo com um procedimento reativo de descoberta de rotas, portanto todas as operações do protocolo operam sob demanda. O DSR foi proposto para redes com alta mobilidade (RFC 4728, 2007). O processo de descoberta de rotas do DSR é similar ao do AODV pela utilização de pacotes de requisição e resposta de rotas. A diferença está no fato de que o pacote de requisição armazena o endereço de todos os nós pelos quais passou até chegar ao nó destino. Um nó executa o procedimento de manutenção de uma determinada rota nas seguintes situações: enquanto a rota estiver sendo utilizada, quando houver mudanças na topologia ou se a rota não estiver mais disponível. A manutenção de rotas é realizada com as rotas alternativas (p.ex. o nó possui outros vizinhos com rotas para o nó destino) ou pelo início de outro processo de descoberta de rota (RFC 4728, 2007).

O protocolo DSR não é adequado para as LLNs, pois os pacotes de descoberta de rota são inundados na rede. Outro aspecto que limita o seu uso em LLNs é a

ausência de um mecanismo que possibilite a manutenção local de desconexão entre nós. Outra limitação presente tanto no protocolo DSR quanto no protocolo AODV para a utilização destes protocolos em LLNs é a baixa responsividade diante das constantes mudanças na topologia causadas pela mobilidade ou pela degradação da qualidade do enlace (FOTOUHI; MOREIRA; ALVES, 2015).

O protocolo *Optimized Link State Routing* (OLSR) (RFC 3626, 2003), diferentemente dos protocolos AODV e DSR, é um protocolo proativo de roteamento. Ou seja, as rotas são construídas antes do envio de mensagens de dados. A descoberta de rotas no OLSR é feita através da divulgação de mensagens contendo informações sobre o estado dos enlaces (RFC 3626, 2003). Uma técnica chamada *MultiPoint Relay* (MPR) é utilizada pelo OLSR para reduzir a sobrecarga gerada pelas mensagens de controle. A técnica MPR consiste em limitar o número de nós na topologia que podem transmitir mensagens de controle para manter a topologia atualizada. Mensagens de controle chamadas de HELLO servem para gerenciar os nós que podem retransmitir mensagens de controle na rede. As informações sobre os enlaces são transmitidas através dos pacotes de controle *Topology Control* (TC). Através dos pacotes TCs, os nós podem calcular os custos das rotas (RFC 3626, 2003).

Segundo Barcelo et al. (2016), protocolos de roteamento como o AODV, DSR e OLSR por terem sido projetado para MANETs e não para redes de sensores, apresentam problemas como alta sobrecarga e ineficiência energética quando utilizados em RSSFs (BARCELO et al., 2016). Isto acontece porque estes protocolos foram projetados para um padrão de comunicação no qual muitos nós transmitem pacotes para diversos outros nós. Contudo, as redes de sensores sem fio geralmente possuem um padrão de comunicação no qual os nós sensores transmitem mensagens para alguns nós específicos da rede. Por exemplo, os nós sorvedouros. Desta maneira, os protocolos utilizados em redes de sensores possuem algumas características que os fazem ser mais econômicos em termos de sobrecarga (BARCELO et al., 2016). Na Seção 2.3, apresentam-se alguns protocolos próprios para RSSFs.

2.3 ROTEAMENTO EM REDES DE SENSORES SEM FIO

O protocolo *Low Energy Adaptive Clustering Hierarchy* (LEACH) é um protocolo de roteamento para RSSF que se utiliza de uma estrutura hierárquica entre os nós. Tal estrutura se baseia na formação de *clusters* entre os nós da topologia (HEINZELMAN; CHANDRAKASAN; BALAKRISHNAN, 2000). Cada *cluster* possui um *Cluster-Head* (CH) responsável por agregar e retransmitir os dados coletados pelos nós sensores para os pontos de acesso da RSSF. Através da utilização da clusterização dos nós, o LEACH busca permitir aos nós utilizarem seus recursos de maneira mais eficiente, porque os nós sensores não necessitam transmitir os dados coletados

diretamente para os pontos de acesso. O LEACH utiliza uma abordagem de rotação entre os CHs (HEINZELMAN; CHANDRAKASAN; BALAKRISHNAN, 2000), ou seja, os CHs são substituídos aleatoriamente, permitindo que o consumo de energia seja distribuído entre os nós.

Lindsey et al. (2002) propuseram um protocolo para RSSFs chamado *Power-Efficient Gathering in Sensor Information System* (PEGASIS). O PEGASIS gera aleatoriamente longas cadeias de nós sensores, direcionadas aos pontos de acesso da RSSF. Cada cadeia de nós sensores possui um nó líder. Geralmente o nó líder é um nó próximo ao ponto de acesso (LINDSEY; RAGHAVENDRA, 2002). A agregação dos dados também pode empregada com a finalidade de reduzir os custos das transmissões. Cada nó da cadeia de nós sensores transmite seus pacotes de dados para o próximo nó mais próximo do ponto de acesso. Este nó pode unir os dados do pacote recebido com os seus próprios dados e retransmitir em direção ao ponto de acesso (LINDSEY; RAGHAVENDRA, 2002). Eventualmente, os pacotes de dados irão chegar ao nó líder da cadeia de nós sensores. Este então, realiza a agregação dos dados e transmite os dados agregados em direção ao ponto de acesso. Com estes procedimentos, o PEGASIS busca reduzir o consumo de energia dos nós ao determinar que somente os nós líderes de cada cadeia de nós sensores poderão realizar agregação de dados e transmiti-los aos pontos de acesso (LINDSEY; RAGHAVENDRA, 2002).

O protocolo *Threshold Sensitive Energy Efficient Sensor Network* (TEEN) foi projetado para redes reativas, ou seja, redes que realizam alguma ação em resposta às mudanças nos dados coletados (MANJESHWAR; AGRAWAL, 2001). Similarmente ao protocolo LEACH, o TEEN utiliza o conceito de *Cluster Head* (CH). O protocolo TEEN utiliza dois limiares para determinar se um dado coletado necessita ser transmitido para o CH. O *Hard Threshold* (HT) avalia o valor do dado coletado, se este valor ultrapassa o limiar, este é transmitido para o CH. O *Soft Threshold* é um limiar que determina o valor mínimo de variação que o dado coletado pode alcançar para ser enviado para o CH. O protocolo TEEN foi planejado para redes com aplicações sensíveis, nas quais determinadas condições do ambiente necessitam ser constantemente monitoradas (MANJESHWAR; AGRAWAL, 2001).

Collection Tree Protocol (CTP) (GNAWALI et al., 2009) é um protocolo de roteamento para redes de sensores sem fio baseado em dois mecanismos que oferecem alta resiliência diante das mudanças na topologia: (i) validação de rotas através do tráfego de dados para rapidamente detectar inconsistências na topologia da rede; e (ii) o envio de pacotes de controle com periodicidade adaptativa visando aplicar mecanismos de reparo da rota (GNAWALI et al., 2009). A topologia formada pelo CTP é uma estrutura em árvore cujo foco principal são as aplicações de coleta de dados. O CTP é um protocolo projetado para agir rapidamente na detecção de ciclos na topo-

logia e para recuperar-se das inconsistências causadas pelo alto dinamismo na rede (GNAWALI et al., 2009).

O mecanismo de validação de rotas do CTP, utiliza-se de pacotes de dados para validar a topologia da rede e detectar ciclos. A detecção de ciclos ocorre através do uso dos pacotes de dados que contém uma métrica para estimar a distância entre o nó remetente e o nó destinatário medida em número de saltos. Um ciclo é detectado quando um nó recebe um pacote cuja distância do remetente ao nó destinatário é menor ou igual à distância do nó em questão ao nó destino. Um mecanismo de reparo da topologia é ativado para que este ciclo seja corrigido (GNAWALI et al., 2009).

Gnawali et al. (2019) apontam que o uso de pacotes de controle, geralmente utilizado pelos protocolos de roteamento podem impactar negativamente na sobrecarga da rede e conseqüentemente aumentar o atraso de transmissão e consumo de energia. No protocolo CTP, o envio periódico de pacotes de controle é realizado por uma extensão do algoritmo *Trickle* (RFC 6206, 2011). O algoritmo reage rapidamente às mudanças na topologia com poucos pacotes de controle sendo transmitidos (GNAWALI et al., 2009). A principal motivação para a especificação do CTP foi a necessidade de um protocolo de roteamento para RSSF com o foco na coleta de dados. Portanto, o CTP foi projetado para aplicações que contenham diversos nós coletores de dados. Os nós coletores transmitem estes dados a um ou mais nós sorvedouros (GNAWALI et al., 2009).

Na Seção 2.4, serão apresentados alguns protocolos de roteamento desenvolvidos para LLNs, e que são compatíveis com o 6LoWPAN.

2.4 ROTEAMENTO EM REDES DE BAIXA CAPACIDADE E PERDAS

As LLNs possuem certas características como baixo, alcance do rádio, autonomia energética e poder de processamento. Entende-se que é desejável que os dispositivos de uma LLN possam ser unicamente identificadas na Internet (OLSSON, 2014). Portanto, os protocolos para este contexto, precisam ser econômicos em relação ao consumo de energia e serem compatíveis com tecnologias como o protocolo *IPv6 over Low-Power Wireless Personal Area Networks* (6LoWPAN). O 6LoWPAN (RFC 4944, 2007) foi desenvolvido pelo *Institute of Electrical and Electronics Engineers* (IEEE) para permitir que dispositivos de baixa capacidade tivessem conexão IPv6 e compatibilidade com o padrão de rádio 802.15.4. O 6LoWPAN possui três principais características: compressão de cabeçalhos IPv6 e *User Datagram Protocol* UDP; fragmentação e reconstrução dos pacotes IPv6. Para que os pacotes IPv6 possam ser transmitidos por enlaces padrão IEEE 802.15.4, os pacotes IPv6 precisam ser fragmentados devido à capacidade de geração própria dos endereços IPv6 (OLSSON,

2014).

Kim et al. (2007) propuseram o protocolo de roteamento *6LoWPAN Ad hoc Routing Protocolo* (LOAD) baseado nos protocolos AODV e 6LoWPAN. No LOAD, a descoberta de rotas é feita sob demanda, ou seja, à medida que novos pacotes de dados necessitam ser transmitidos. As rotas são descobertas através do *multicast* de mensagens de controle de descoberta de rotas (RREQ), assim como ocorre no protocolo AODV. Este procedimento aumenta a sobrecarga na rede, o que pode ser uma limitação para a sua aplicação em LLN (CHANG et al., 2010). Motivados por esta limitação do protocolo LOAD, Chang et al. (2010) apresentaram uma modificação no protocolo LOAD para que o mesmo suporte o conceito de múltiplos caminhos para reduzir a sobrecarga causada pelo mecanismo de descoberta de rotas do AODV. A proposta dos autores é chamada de MLOAD (CHANG et al., 2010). Desta maneira, se uma rota não estiver mais disponível, o nó possui rotas reservas que podem ser utilizadas sem a necessidade de um novo processo de descoberta de rotas (CHANG et al., 2010). Contudo, o protocolo MLOAD de Chang et al. (2010), apresentam baixa escalabilidade e elevado consumo de memória do dispositivo (KUMAR; TIWARI, 2012).

Haggerty et al. (2010) argumentam que protocolos de roteamento como, o AODV e o DSR são destinados para dispositivos com quantidade considerável de recursos (DAWSON-HAGGERTY; TAVAKOLI; CULLER, 2010). Assim, Haggerty et al. (2010) propuseram um protocolo de roteamento para LLN, Hydro. O protocolo Hydro utiliza o conceito de *Directed Acyclic Graph* (DAG) para a construção da topologia que é composta por dois tipos de nós: nó sensor e nó sorvedouro. O gerenciamento da topologia é uma combinação de mecanismos distribuídos e centralizado (DAWSON-HAGGERTY; TAVAKOLI; CULLER, 2010). Os nós sensores constroem e mantêm uma DAG que oferece rotas dos nós sensores para o nó sorvedouro. Os nós sorvedouros mantêm uma visão global da rede, chamado de *Link State Database*, utilizando informações da topologia enviadas por cada nó sensor. O envio das informações é feito de modo oportunístico (DAWSON-HAGGERTY; TAVAKOLI; CULLER, 2010).

Baseado na análise da proposta de Haggerty et al. (2010), entende-se que o protocolo de roteamento Hydro foi projetado sob algumas premissas: a primeira, é a existência de dispositivos de maior capacidade de memória e processamento para atuar como nó sorvedouro. Pois, o nó sorvedouro otimiza as rotas ponto a ponto para todos os nós da topologia. Portanto, o nó sorvedouro precisa ser capaz de armazenar os endereços dos nós sensores e executar algoritmos de otimização de rotas a medida que novas informações sobre a topologia são enviadas pelos nós sensores. Pelo fato do protocolo Hydro utilizar um conceito de otimização global das rotas, sua utilização em LLNs pode ser limitada, pois depende da capacidade de processamento dos dispositivos utilizados.

Na Seção 2.5, discute-se acerca dos protocolos apresentados e suas limitações na utilização em LLNs.

2.5 CONSIDERAÇÕES SOBRE OS PROTOCOLOS DE ROTEAMENTO

A carência de um protocolo de roteamento adequado para as LLNs motivou o grupo *Internet Engineering Task Force* (IETF) Roll a especificar um novo protocolo denominado por *IPv6 Routing Protocol for Low-Power and Lossy Network* (RPL) (RFC 6550, 2012). O RPL, assim como o Hydro e LOAD, é compatível com o protocolo 6LoWPAN, uma versão comprimida do protocolo IPv6 para redes de baixa capacidade.

O RPL, Hydro e CTP possuem abordagens semelhantes no tocante a construção da topologia: utiliza-se uma estrutura baseada em árvore para a organização da topologia. O RPL apresenta melhorias em relação ao CTP nos quesitos: escalabilidade, consumo de energia e taxa de sucesso de entrega (LONG et al., 2012). Segundo Long et al. (2012), o melhor desempenho em termos de consumo de energia e taxa de sucesso de entrega do RPL, em comparação com o CTP, está diretamente relacionado ao tamanho da rede e ao padrão de tráfego de pacotes. O RPL apresenta um mecanismo mais complexo e eficiente para detecção e recuperação de ciclos na rede (LONG et al., 2012). Outra vantagem do RPL sobre o CTP é a possibilidade de realizar processos locais para recuperação de inconsistências na topologia. Com isso, a manutenção da topologia exige uma menor quantidade de nós (LONG et al., 2012).

Em comparação com o Hydro, o protocolo RPL apresenta mecanismos mais eficientes para o gerenciamento da topologia em diferentes padrões de tráfego: MP2M, P2P e P2MP. Outra vantagem do RPL em comparação com os outros protocolos de roteamento com foco em aplicações de coleta de dados, como o Hydro e CTP, é a grande flexibilidade do RPL no atendimento de requisitos de diferentes tipos de aplicação. Por exemplo, determinadas aplicações podem demandar baixo atraso de entrega, outras, uma alta taxa de sucesso de entrega. O RPL propõe a utilização de funções objetivas para a otimização de determinadas métricas de acordo com as necessidades da aplicação (GADDOUR; KOUBÂA, 2012). No RPL, o envio periódico de mensagens de controle e a construção proativa da topologia, dispensando inundações de pacotes de controle para descoberta de rotas, torna o RPL um protocolo com baixa sobrecarga e mais resiliente às mudanças na topologia (GADDOUR; KOUBÂA, 2012). Por ser objeto de pesquisa deste trabalho, os detalhes de funcionamento do RPL serão descritos na Seção 2.6.

O protocolo CTP foi desenvolvido para as RSSFs, mas apresenta diversas estratégias pertinentes para LLNs. Por exemplo, a manutenção local de rotas e a detecção de ciclos. Motivados pela necessidade de um protocolo de roteamento que

atenda os requisitos das LLNs, e que seja interoperável com protocolos já existentes para IoT como o 6LoWPAN, o grupo IETF projetou o protocolo de roteamento RPL. O protocolo RPL será descrito em detalhes na próxima seção.

2.6 PROTOCOLO *IPV6 ROUTING PROTOCOL FOR LOW-POWER AND LOSSY NETWORKS* (RPL)

Serão apresentadas nesta seção as características do protocolo *IPv6 Routing Protocol for Low-Power and Lossy Networks* (RPL) (RFC 6550, 2012) com o objetivo de elucidar o funcionamento deste protocolo. Na Subseção 2.6.2, serão discutidos os mecanismos de descoberta de vizinhos e o funcionamento do temporizador *Trickle* (RFC 6206, 2011) utilizado no gerenciamento da transmissão de mensagens de controle *DODAG Information Object* (DIO).

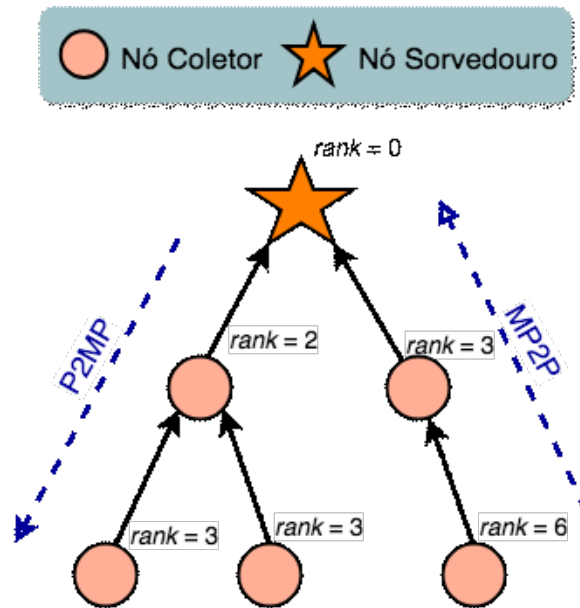
2.6.1 Fundamentos do Protocolo RPL

O protocolo de roteamento RPL é do tipo vetor de distância e realiza roteamento por endereçamento através do protocolo IPv6. Protocolos do tipo vetor de distância calculam a melhor rota baseada nas distâncias. No caso do RPL, a distância é determinada por um ranqueamento (*rank*) que cada nó possui em relação ao nó sorvedouro. O *rank* de um nó é calculado por uma função objetivo que pode levar em consideração métricas como a distância por saltos (RFC 6550, 2012) e qualidade do enlace. Como exemplificado na Figura 2.5, o *rank* do nó deve respeitar a propriedade: aumentar seu valor a medida em que o nó está mais distante do nó sorvedouro (RFC 6550, 2012). Os valores dos *ranks* da Figura 2.5 são apenas para exemplificação.

O RPL foi projetado visando aplicações de sensoriamento e de coleta de dados. No RPL há basicamente dois papéis que cada nó da rede pode desempenhar: nó sorvedouro (também chamado de *sink*) e nó coletor (GADDOUR; KOUBÂA, 2012) como pode ser observado na Figura 2.5. O nó sorvedouro é responsável por receber os dados coletados por outros nós e transmiti-los para fora da rede de sensores. O nó coletor é responsável por coletar os dados de interesse da aplicação. O modelo de comunicação normalmente utilizado nesse tipo de topologia é de Múltiplos Pontos para um Ponto (MP2P), cuja exemplificação é representada pela seta azul pontilhada à direita na Figura 2.5. Em outras palavras, vários nós coletores transmitem dados salto-a-salto até o nó sorvedouro (GADDOUR; KOUBÂA, 2012). O protocolo RPL também suporta o fluxo de transmissão de um Ponto para Muitos Pontos (P2MP), exemplificado pela seta azul pontilhada à esquerda na Figura 2.5.

No contexto da descoberta de rota e construção da topologia, uma abordagem reativa é caracterizada pelo fato de um nó descobrir a rota até o destino no momento

Figura 2.5 – Exemplo de uma Topologia RPL.



Fonte: O autor.

em que for transmiti-la. A topologia é construída à medida em que os nós da rede necessitam transmitir suas mensagens. Na literatura, são encontrados vários protocolos de roteamento reativos, como o AODV (RFC 3561, 2003), DSR (RFC 4728, 2007) e LOAD (CHANG et al., 2010). Algoritmos reativos de descoberta de rota constroem a topologia sob demanda à medida que novas mensagens necessitam ser transmitidas pelos nós. Como resultado desse comportamento, tais algoritmos não respondem rapidamente às mudanças da topologia (FOTOUHI; MOREIRA; ALVES, 2015).

Na abordagem proativa, as rotas são descobertas e a topologia é construída antes que apareça a necessidade de um nó transmitir mensagens de dados. Portanto, a rota para o destino já será de conhecimento do nó no momento do envio dos pacotes de dados. Exemplos de protocolos que seguem essa abordagem na fase de descoberta de rotas são o OLSR (RFC 3626, 2003), CTP (GNAWALI et al., 2009), o RPL (RFC 6550, 2012) e o Hydro (DAWSON-HAGGERTY; TAVAKOLI; CULLER, 2010).

A detecção da inalcançabilidade de nós vizinhos é essencial em topologias com nós móveis, pois a mobilidade aumenta a probabilidade de desconexão na rede (COBARZAN; MONTAVONT; NOEL, 2014). O RPL utiliza a abordagem reativa nativamente, isto é, não há a troca de mensagem do tipo de sinalização para que a inalcançabilidade do vizinho seja detectada. Uma elevada troca de mensagens de controle aumenta a sobrecarga nos enlaces e degradaria rapidamente os recursos dos dispositivos (RFC 6550, 2012). Fotouhi et al. (2015) propuseram um mecanismo proativo

de detecção de inalcançabilidade que emprega trocas constantes de mensagens de controle para detectar mudanças na vizinhança para o RPL. Os autores defendem que a abordagem proativa oferece maior responsividade a mudanças na topologia, resultando, no aumento da taxa de entrega de pacotes quando comparada com a especificação canônica do protocolo RPL (FOTOUHI; MOREIRA; ALVES, 2015).

Segundo Bouaziz et al. (2106), protocolos proativos para a detecção de inalcançabilidade de nós vizinhos geralmente dependem de mecanismos de predição da mobilidade dos nós. A utilização de abordagens proativas de suporte à mobilidade em LLNs apresenta limitações como: o alto consumo de memória, processamento e energia para que as predições de movimentação sejam mais assertivas (BOUAZIZ; RACHEDI, 2016). A adoção de uma estratégia proativa pode aumentar significativamente a sobrecarga na rede devido ao constante monitoramento da validade das conexões.

No RPL, a construção da topologia e descoberta de rotas é realizada de forma proativa através do nó raiz. Porém, a especificação canônica do RPL não determina um mecanismo para a identificação da inalcançabilidade do nó pai. Desta maneira, tal mecanismo é um tema de pesquisa em aberto. A atualização da topologia no RPL fica a cargo somente do temporizador *Trickle* (na Seção 2.6.2, apresenta-se o mecanismo de funcionamento do *Trickle* e aspectos da sua implementação).

A construção da topologia no RPL é inicializada pelo nó sorvedouro e é baseada no conceito de Grafos Direcionados e Acíclicos (DAG). A topologia é construída como uma árvore, cuja posição dos nós é relacionada com seus *ranks*, de maneira que os nós mais próximos do nó sorvedouro possuam *ranks* menores em relação aos nós mais distantes. A árvore é chamada de *Destination Oriented Directed Acyclic Graph* (DODAG), no qual o nó sorvedouro é chamado de nó raiz (RFC 6550, 2012).

A escolha do melhor nó pai, por parte de um nó que deseja ingressar na topologia RPL é guiada pelo valor de *rank*. Este valor permite a identificação de ciclos e a diferenciação entre os nós pais (com *ranks* menores) e nós irmãos (com *ranks* iguais). No RPL, o valor de *rank* é calculado por uma função objetivo que pode otimizar uma ou mais métricas. As funções objetivos disponibilizadas no RPL canônico (RFC 6719, 2012) (RFC 6552, 2012), e a escolha de determinadas métricas para a construção da topologia influenciam o desempenho do protocolo de roteamento em relação ao consumo de energia dos nós, tempo de vida da rede e taxa de sucesso de entrega de pacotes.

A especificação do RPL define duas funções objetivos, *The Minimum Rank with Hysteresis Objective Function* (MRHOF) (RFC 6719, 2012) e *Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks* (OF0) (RFC 6552, 2012). A função objetivo OF0 otimiza a métrica de distância em número de saltos. A

escolha desta métrica permite que rotas mais curtas sejam selecionadas. A função objetivo MRHOF também otimiza a distância em número de saltos, mas inclui a métrica *Expected Transmission Count* (ETX) no cálculo do *rank*. Assim, a função objetivo MRHOF não só permite a escolha de rotas mais curtas, como também a escolha de rotas com maior probabilidade de sucesso da entrega do pacote ao nó destino. A especificação canônica do RPL recomenda que diferentes aplicações de IoT experimentem utilizar outras métricas que atendam melhor suas necessidades e restrições (RFC 6550, 2012). O RPL pode utilizar tanto métricas para otimização, quanto para restrição na escolha da melhor rota. Na Subseção 2.6.1.1, apresenta-se uma discussão sobre o uso de diferentes métricas no processo de roteamento do RPL

Gara et al. (2016) e Bouaziz et al. (2017) propõem a utilização de métricas baseadas no *Received Signal Strength Indicator* (RSSI) na função objetivo MRHOF, além das métricas de distância em número de saltos e ETX já adotadas na especificação canônica do protocolo RPL. Segundo a análise da literatura, os trabalhos com foco em mobilidade frequentemente utilizam métricas relacionadas ao RSSI, ETX e velocidade. As métricas derivadas do RSSI são normalmente utilizadas nas propostas proativas de mecanismos para o gerenciamento de mobilidade no RPL (GARA et al., 2016) (KHARRUFA et al., 2017).

O valor de RSSI indica a potência do sinal medido durante o recebimento de pacotes de um determinado nó vizinho no RPL e seu valor é medido em decibéis. A partir do monitoramento do valor de RSSI, pode-se estimar a distância relativa entre dois nós (PARAMESWARAN et al., 2009). Apesar de não ser considerado a métrica mais indicada para se estimar com exatidão a distância entre dois nós (PARAMESWARAN et al., 2009), o RSSI é aplicado para obter uma distância aproximada em algumas das propostas da literatura, (GARA et al., 2016) (KHARRUFA et al., 2017) e (FOTOUHI; MOREIRA; ALVES, 2015). Na abordagem apresentada no Capítulo 4, variações nos valores de RSSI, utilizados na métrica de Variabilidade de Vizinhança utilizados para identificar a movimentação do nó.

2.6.1.1 Métricas de Roteamento do RPL

As métricas otimizadas nas funções objetivos disponibilizadas pelo protocolo de roteamento RPL podem ser classificadas de acordo com as seguintes características (RFC 6551, 2012): (i) podem ser de enlace ou de nó; (ii) dinâmica ou estática; e (iii) para a otimização ou a restrição de rotas.

As métricas de nó possuem informações relacionadas ao nó em questão (p.ex. energia residual ou posição geográfica). As métricas de enlace são relacionadas com a qualidade do meio compartilhado entre dois ou mais nós (p.ex. potência de sinal). Métricas dinâmicas têm seu valor atualizado ao longo do funcionamento da rede. Em

contrapartida, os valores das métricas estáticas permanecem inalterados. As métricas de otimização de rotas são praticadas na escolha das melhores rotas pelas funções objetivos. As métricas de restrições possibilitam o estabelecimento de limiares para eliminar candidatos a nó pai com rotas sub-ótimas. Uma mesma métrica pode ser utilizada tanto para otimização na função objetivo, quanto como uma restrição. Por exemplo, a métrica distância em número de saltos pode ser aplicada na escolha do nó pai com o menor caminho ao nó sorvedouro, e para suprimir rotas muito longas (p.ex. rotas acima de um determinado número de saltos) (RFC 6551, 2012).

Os valores de métricas dinâmicas como, de nó (p.ex. energia residual do nó) ou de enlace (p.ex. ETX e RSSI), podem variar rapidamente devido a alta dinamicidade do enlace. Para que métricas dinâmicas sejam eficazes, faz-se necessário que mecanismos para suavizar variações abruptas dos valores dessas métricas sejam empregados, evitando flutuações no roteamento através de trocas desnecessárias do nó pai (RFC 6551, 2012). O uso de métricas dinâmicas é recomendável para se obter um roteamento mais resiliente (RFC 6551, 2012). Os mecanismos para suavizar variações das métricas precisam levar em consideração a periodicidade com que os valores são atualizados. Em Vasseur et al. (2012) indica-se o uso de múltiplos limiares para que haja a atualização da métrica. Os autores mencionam que a presença de múltiplos limiares evitam que aconteçam mudanças desnecessárias em seus valores (RFC 6551, 2012).

Adicionalmente, as métricas também podem ser classificadas como primárias (p.ex. ETX) ou compostas (p.ex. energia residual). Métricas primárias representam um efeito específico, como a qualidade do enlace, ou para atingir um objetivo único, como reduzir a quantidade de retransmissões. Métricas compostas visam combinar duas ou mais métricas para que diferentes necessidades e restrições da rede sejam atingidas (KARKAZIS et al., 2013).

2.6.1.2 Mensagens de Controle e Construção de Topologias no RPL

O protocolo RPL utiliza quatro mensagens de controle do tipo *Internet Control Message Protocol for IPv6* (ICMPv6) para a construção e manutenção da topologia (RFC 6550, 2012):

- *DODAG Information Object* (DIO): mensagem inicialmente transmitida pelo nó sorvedouro da DODAG. A mensagem do tipo DIO é utilizada na construção da topologia no RPL. Nas mensagens DIO, são carregadas informações relevantes para que os nós da rede possam escolher a melhor rota para o nó sorvedouro. O nó sorvedouro, através das mensagens DIO, divulga métricas e restrições para

o cálculo dos custos das rotas, e para eliminar rotas por nós vizinhos que não satisfaçam restrições específicas da aplicação (RFC 6551, 2012).

- *DODAG Information Solicitation* (DIS): mensagem utilizada por nós que ainda não pertencem a uma determinada DODAG. Estes nós enviam mensagens do tipo DIS, buscando obter mensagens de resposta do tipo DIO. A partir do recebimento de mensagens DIO, o nó pode escolher um nó pai e ingressar em uma DODAG, analisar sua vizinhança e realizar uma escolha mais assertiva da melhor rota a sua disposição em um determinado momento.
- *Destination Advertisement Object* (DAO): mensagem transmitida por todos os nós da rede, exceto pelo nó sorvedouro. O objetivo da transmissão de mensagens DAO é a propagação de informações de roteamento dos nós filhos para seus respectivos nós pais ou para o nó sorvedouro da DODAG.
- *Destination Advertisement Object Acknowledgement* (DAO-ACK): mensagem enviada como resposta a uma mensagem do tipo DAO e seu uso é opcional.

O protocolo RPL considera a existência de dois tipos de rotas: rotas em direção ao nó sorvedouro (rotas para cima), e em direção aos nós folhas (rotas para baixo). A construção de rotas para baixo é declarada como opcional pela especificação canônica do protocolo RPL (RFC 6550, 2012). Em aplicações de sensoriamento comuns em IoT, o fluxo de transmissão da topologia normalmente parte dos nós sensores em direção ao nó sorvedouro (RFC 5867, 2010), (RFC 5826, 2010) e (RFC 5673, 2009).

A construção da topologia é iniciada pelo nó sorvedouro ao transmitir mensagens DIO em *multicast* para seus vizinhos (RFC 6550, 2012), sendo fortemente influenciada pela função objetivo utilizada pela DODAG construída através do nó sorvedouro. A função objetivo adotada pelo nó sorvedouro é determinada no cabeçalho da mensagem DIO. Esta função determina como uma ou mais métricas (p.ex. ETX e distância em saltos), em conjunto com uma ou mais restrições (p.ex. número máximo de saltos que um pacote pode percorrer na topologia), devem ser consideradas pelos nós para o cálculo do *rank* dos candidatos a nó pai (RFC 6550, 2012). O valor de *rank* determina o quão distante o nó em questão está do nó sorvedouro. Para a escolha do nó pai no RPL canônico, o *rank* do pai deve ser menor que o *rank* do nó em questão (RFC 6550, 2012). O propósito da filosofia adotada para o cálculo do *rank* é possibilitar a identificação de ciclos na rede e a escolha de rotas mais eficientes entre os nós coletores e o nó sorvedouro.

O processo de construção de rotas para cima se dá em termos de três elementos (RFC 6550, 2012): (i) um conjunto de candidatos a nó pai é um subconjunto

dos nós vizinhos que um determinado nó consegue alcançar via troca local de mensagens de controle; (ii) os nós que compõem este subconjunto são selecionados de acordo com a função objetivo utilizada, OF0 ou MRHOF, por exemplo; e (iii) o nó pai é selecionado dentre os nós candidatos do conjunto quando o seu valor de *rank* for o menor do conjunto (RFC 6550, 2012).

Com base nas pesquisas realizadas em relação ao RPL canônico acerca das transmissões de mensagens de controle, observou-se que a influência direta da periodicidade das transmissões tem impacto significativo na rápida detecção da mobilidade dos nós e inalcançabilidade do nó pai.

2.6.1.3 Funções Objetivos no Protocolo RPL

O valor do *rank* no protocolo RPL é calculado através de uma função objetivo. O *rank* representa a distância de um nó em relação ao nó sorvedouro em uma DO-DAG. Pelo fato do valor do *rank* ser utilizado para a detecção e mitigação de ciclos na topologia, ele deve apresentar as seguintes propriedades (RFC 6550, 2012):

- O valor do *rank* deve diminuir à medida que o nó está mais próximo do nó sorvedouro. Deste modo, este valor pode ser utilizado para determinar a localização do nó na DODAG, e como resultado, mitigar potenciais ciclos na topologia. Na especificação canônica do protocolo RPL, afirma-se que o valor de *rank* não deve ser necessariamente considerado como o custo da rota (RFC 6550, 2012), pois o custo da rota pode levar em consideração outras métricas relacionadas à qualidade do enlace.
- A função objetivo considerada no cálculo do *rank* e da seleção do nó pai escolhe o próximo nó pai a qualquer momento em que uma informação do nó vizinho é atualizada. Através da função objetivo escolhida, o RPL identifica se a rota até o nó sorvedouro pelo nó pai atual está obsoleta. A seleção de um novo nó pai no RPL canônico pode somente ocorrer através do recebimento de mensagens DIO com valores de *rank* menores quando comparado com o nó pai atual.

Conforme descrito anteriormente, o RPL canônico possui duas funções objetivos. *Objective Function Zero* (OF0) (RFC 6552, 2012) e *Minimum Rank with Hysteresis Objective Function* (MRHOF) (RFC 6719, 2012). O cálculo do *rank* pela função objetivo OF0 é realizado através da variável *step_of_rank*, que mostra o quanto o valor do *rank* deve variar de um nó para o outro. O uso da função objetivo OF0 pode resultar na escolha de rotas mais curtas, porém de menor qualidade (RFC 6552, 2012). Segundo Thubert (2012), é preferível utilizar a MRHOF quando o objetivo for melhorar a estabilidade na rede.

A função objetivo MRHOF foi proposta para selecionar a melhor rota de maneira a reduzir trocas constantes do nó pai. Dois mecanismos no RPL são necessários para que a melhor rota seja selecionada pela função MRHOF: (i) o primeiro trata da descoberta da rota com o nó pai de menor *rank* e; (ii) o segundo mecanismo restringe a troca de nó pai somente se o *rank* do candidato a nó pai for menor do que o *rank* do nó pai atual dado o limiar *PARENT_SWITCH_THRESHOLD* (RFC 6719, 2012). Em outras palavras, a troca de nó pai ocorre somente se $rank_{nó_candidato} + PARENT_SWITCH_THRESHOLD < rank_{nó_pai}$.

2.6.2 Algoritmo *Trickle*

O algoritmo *Trickle* (RFC 6206, 2011) fornece uma série de primitivas e mecanismos de comunicação para gerenciar a frequência com que as mensagens de controle são transmitidas em uma rede (RFC 6206, 2011). Este algoritmo é utilizado pelo protocolo RPL como temporizador para controlar a frequência de envio das mensagens DIO. Originalmente, o algoritmo *Trickle* foi concebido visando reduzir a periodicidade da transmissão de pacotes de controle a medida em que a topologia vai se tornando estável (p.ex. em redes com apenas nós estáticos). Entretanto, através do temporizador *Trickle* é possível aumentar rapidamente essa periodicidade para que a topologia possa ser atualizada o mais rápido possível, quando alguma inconsistência for identificada na topologia (p.ex. um ciclo ou rota desatualizada).

A utilização do temporizador *Trickle* pelo RPL, baseia-se na premissa de que os nós devem transmitir mensagens de controle, com exceção das situações em que sua mensagem já não é relevante para ser transmitida (RFC 6206, 2011). Em outras palavras, um nó não deve encaminhar uma mensagem DIO divulgando uma DODAG, caso outros nós em sua vizinhança já o tenham feito em uma quantidade suficiente. Com o uso do *Trickle*, é possível reduzir a sobrecarga da transmissão de mensagens de controle.

O *Trickle* utiliza os conceitos de consistência e inconsistência de transmissões, cuja definição fica a cargo dos protocolos que o utilizarem. No RPL, o conceito de inconsistência indica a possibilidade de um ciclo na topologia. Por exemplo: se um nó recebe um pacote em direção ao nó sorvedouro da DODAG, e se o nó que recebeu o pacote para ser roteado identifica que o seu *rank* é maior do que o *rank* do nó remetente, há um indício de que o pacote não está seguindo em direção ao nó sorvedouro. Ressalta-se, que o valor de *rank* dos nós diminuem a medida que o nó está mais próximo do nó sorvedouro. No algoritmo *Trickle* utilizado no RPL (RFC 6550, 2012), ao identificar uma inconsistência, o envio de mensagens DIO aumenta exponencialmente para que a topologia RPL possa ser rapidamente atualizada.

O temporizador *Trickle* determina a periodicidade com que mensagens de con-

trole DIO devem ser propagadas na rede através do ajuste de três parâmetros de configuração (RFC 6206, 2011):

- I_{min} : o tamanho mínimo do intervalo definido em unidades de tempo (p.ex. segundos ou milissegundos). Faz-se necessário que seu valor seja menor do que I_{max} .
- I_{max} : define o tamanho máximo que um intervalo pode assumir.
- k : uma constante de redundância que deve ser maior que 0. Tal constante é definida pelo protocolo que for utilizar este algoritmo e serve para limitar a quantidade de mensagens de controle transmitidas quando a topologia RPL não necessita de ajustes.

Além dos parâmetros I_{min} , I_{max} e k , o *Trickle* possui três variáveis que servem para o controle dos intervalos de tempo em que o nó deve restringir a transmissão dos pacotes de controle (RFC 6206, 2011):

- I : o tamanho atual do intervalo.
- t : valor no intervalo entre $\frac{I}{2}$ e I . Somente no instante t , o nó tem a possibilidade de transmitir mensagens de controle DIO para seus vizinhos.
- c : um contador que registra a quantidade de mensagens consistentes que foram recebidas. A transmissão de mensagens de controle DIO no instante t é permitida somente se o valor de c for menor do que a constante k .

O funcionamento do *Trickle* é descrito através do Algoritmo 2.1. O primeiro passo do algoritmo é escolher um valor para a variável I entre o conjunto fechado $[I_{min}, I_{max}]$ (ver linha 2), e então o primeiro intervalo *Trickle* é iniciado na linha 3. O procedimento *START_INTERVAL* é executado a cada intervalo de tempo definido pelo temporizador *Trickle*. O primeiro passo após o início do intervalo é iniciar o valor do contador c em 0 (linha 5). Escolhe-se, um valor aleatório no conjunto $[\frac{I}{2}, I)$ para o limiar de supressão das transmissões de mensagens de controle DIO, definido pela variável t (linha 6) (RFC 6206, 2011). Do início do intervalo até o tempo t , o nó fica impossibilitado de transmitir mensagens DIO. Este período de supressão é chamado de *Listen Only Period* (LOP). Segundo Ghaleb et al. (2015), o tamanho do LOP impacta diretamente no tempo necessário para a rede convergir e para que nós móveis possam reingressar na topologia RPL quando ocorrerem as desconexões causadas pela mobilidade.

Enquanto o temporizador *Trickle* estiver ativo, a cada recebimento de uma mensagem considerada consistente (p.ex. no RPL, uma mensagem de controle DIO de uma DODAG sem ciclos) o contador c é incrementado (linha 10). Caso o nó receba uma mensagem inconsistente (i.e. indicando que a DODAG precisa ser atualizada), o valor de I é atualizado (ver linha 12) e o temporizador é reiniciado. O envio de mensagens de controle DIO só é permitido quando o intervalo de tempo atingir o valor de t e se a quantidade de mensagens consistentes recebidas for menor do que o limiar k (ver linha 15). Após o término de cada período de tempo definido pelo temporizador *Trickle*, o valor da variável I é dobrado e o temporizador é reiniciado (RFC 6206, 2011).

Algoritmo 2.1 Algoritmo *Trickle*

```

1: procedure TRICKLE
2:    $I \leftarrow \text{random value in } [I_{min}, I_{max}]$ 
3:   START_INTERVAL()
4: procedure START_INTERVAL
5:    $c \leftarrow 0$ 
6:    $t \leftarrow \text{random value in } [\frac{I}{2}, I)$ 
7:   while trickle timer not expired do
8:     if new message then
9:       if consistent_message then
10:         $c \leftarrow c + 1$ 
11:       else
12:         $I \leftarrow I_{min}$ 
13:        reset_trickle_timer()
14:        START_INTERVAL()
15:       if trickle timer = t and  $c < k$  then
16:        transmit_dio_message()
17:        update_trickle_timer()
18:    $I \leftarrow I * 2$ 
19:   START_INTERVAL()

```

O algoritmo *Trickle* possui dois mecanismos de controle para a transmissão de mensagens de controle DIO na rede. O primeiro mecanismo é o de supressão da transmissão de mensagens do tipo DIO consideradas irrelevantes (i.e. quando um número c de mensagens consistentes já foram transmitidas na vizinhança). O segundo é a resolução de inconsistências através do aumento exponencial na periodicidade da transmissão de mensagens DIO (GHALEB; AL-DUBAI; EKONOMOU, 2015). Exemplos de melhorias para estes mecanismos são apresentados em Ghaleb et al. (2015) e Bouaziz et al. (2017).

A constante k e o intervalo de valores que I e t podem assumir, servem para suprimir a transmissão de mensagens de controle DIO por um determinado período de tempo. Esse período de supressão de transmissão cresce exponencialmente ao longo

do tempo. Esta estratégia foi proposta no RPL para topologias estáticas. Quando o período LOP é muito longo, a rápida detecção da inalcançabilidade do nó pai pode ficar comprometida devido à redução na transmissão de mensagens de controle (GARA et al., 2016).

Durante o período LOP, os nós somente recebem transmissões das mensagens de controle DIO dos nós vizinhos. No período LOP, os nós não tem permissão para transmitir mensagens de controle DIO aos nós vizinhos. Essa regra se mantém enquanto o período LOP não terminar e se o valor do contador c for menor do que a constante k (veja a linha 15 no Algoritmo 2.1). Em outras palavras, o nó transmite se o mesmo não ouviu um número k de mensagens DIO consistentes de seus vizinhos. Nas situações em que o período LOP fica ativo por muito tempo (YASSEIN et al., 2016), a convergência de uma rede com mobilidade será mais lenta. Nas redes com mobilidade, nas quais o tempo de resposta às mudanças na topologia causada pela mobilidade deve ser o menor possível, o uso da estratégia adotada pelo LOP pode comprometer consideravelmente o tempo de convergência da rede (GHALEB; AL-DUBAI; EKONOMOU, 2015).

A constante k influencia diretamente o tamanho do período LOP. Configurar o valor de k para um número grande o bastante que represente o infinito neste contexto eliminaria esse período (RFC 6550, 2012). Porém, tal atitude é desencorajada pela especificação do algoritmo *Trickle*. Desabilitar a supressão significa que todos os nós podem transmitir mensagens DIO em todos os intervalos, aumentando o congestionamento em topologias muito densas (RFC 6206, 2011). Uma proposta para tornar a constante k em um valor ajustável dinamicamente foi apresentada em Ghaleb et al. (2015).

Com base na análise da literatura, conclui-se que o ajuste dos parâmetros do *Trickle* deve ser realizado com cuidado. Adicionar um valor pequeno para o LOP pode aumentar a sobrecarga na rede. Por outro lado, um LOP muito longo reduz drasticamente a responsividade da rede diante das mudanças na topologia.

2.6.3 Mobilidade no RPL e seus Desafios

Diversas aplicações no contexto de Internet das Coisas necessitam que haja uma coexistência entre dispositivos estáticos e móveis conectados em uma mesma rede. Portanto, a cooperação entre nós móveis e fixos vem sendo amplamente considerada na literatura (LAMAAZI et al., 2015).

No escopo da mobilidade, pode-se dizer que o padrão de movimentação e o padrão do tráfego de pacotes são elementos que impactam diretamente no desempenho de um algoritmo de roteamento (BAI; HELMY, 2004). Exemplos de métricas

diretamente ligadas ao padrão de movimentação dos nós da rede são: taxa de sucesso de entrega e taxa de transferência (LAMAASI et al., 2016).

A presença de nós móveis na rede traz uma série de desafios ao RPL em termos de taxa de sucesso de transmissão, perda de pacotes, sobrecarga, desconexões e consumo de energia. Porém, apesar de haver sido planejado para redes estáticas, este protocolo pode ser adaptado para uma rede composta por nós móveis (GARA et al., 2016).

2.6.3.1 *Análise das Limitações na Detecção de Mobilidade e na Função Objetivo*

O RPL carece de um mecanismo para a detecção da mobilidade dos nós. Através de um mecanismo para a detecção da mobilidade, o RPL poderia utilizar as mensagens de controle, DIO e DIS para atualizar a topologia o mais rápido possível. Desta forma, o RPL seria mais resiliente as constantes mudanças na topologia causadas pela movimentação dos nós. Ainda, a falta de um mecanismo eficiente para a troca do nó pai pelo nó móvel prejudica sua aplicação direta em topologias com nós móveis. Na especificação canônica do RPL, somente o recebimento de mensagens DIO com um valor de *rank* menor, em relação ao nó pai atual pode iniciar o procedimento de substituição do nó pai. O fato em si não é um problema e deve ser mantido, pois permite que ciclos na topologia sejam reduzidos ou evitados em alguns casos. Porém, faz-se necessário que os nós móveis possam iniciar um procedimento de atualização da topologia quando detectarem sua movimentação.

Quando um nó sai da área de cobertura de seu nó pai, este nó pode continuar levando em consideração uma conexão não mais existente entre eles devido a falta de um mecanismo eficiente de detecção de desconexões. Esta limitação está diretamente associada à inexistência de um mecanismo de identificação da inalcançabilidade do nó pai (ponto crítico em topologias móveis), e da falta de políticas sobre como os nós devem proceder para atualizar a topologia em caso de mobilidade em sua vizinhança.

Alguns trabalhos relacionados, como Fotouhi et al. (2015) e Bouaziz et al. (2017), defendem estratégias de previsão de mobilidade. As propostas que empregam a ideia de previsão geralmente utilizam informações estimadas por valores de RSSI (SAVVIDES; HAN; STRIVASTAVA, 2001). No entanto, as medições de RSSI são altamente influenciáveis por interferências e pelo efeito de somreamento de sinal. Desse modo, existe a necessidade de utilizar o RSSI de maneira mais eficiente para tornar a detecção ou a previsão de mobilidade mais assertiva.

O reingresso na topologia acontece através da escolha de um novo nó pai. Essa escolha se dá de acordo com o *rank* de cada nó. O valor do *rank* é determinado por uma função objetivo. A função objetivo utilizada MHROF (RFC 6719, 2012) para o

cálculo do *rank* também faz uso da métrica, Variabilidade de Vizinhança do nó candidato, em conjunto com as métricas padrões do RPL, ETX e distância em número de saltos.

2.6.3.2 *Análise das Limitações do Mecanismo de Envio de Mensagens DIO*

Na especificação canônica do protocolo RPL, é utilizado o temporizador *Trickle*, responsável por regular a troca de mensagens de controle DIO. Como foi descrito anteriormente, durante o período *Listen Only Period* (LOP) o nó fica impossibilitado de transmitir pacotes de controle DIO. A supressão da transmissão se faz necessária para que um enlace não seja sobrecarregado e cada nó tenha a possibilidade de transmitir seus pacotes (RFC 6206, 2011). Esse mecanismo também possibilita uma redução na utilização da largura de banda, pois permite ao nó verificar quando a transmissão da mensagem DIO é desnecessária. A transmissão da mensagem DIO é desnecessária quando a DODAG se encontra atualizada. Portanto, o problema não é a existência do período LOP, mas sim a maneira como ele é ajustado.

Na especificação canônica do RPL, este período é regulado, assumindo-se que a topologia se estabiliza ao longo do tempo, levando em consideração que o RPL foi inicialmente proposto para redes estáticas. O gerenciamento da mobilidade é um tema em aberto na especificação do RPL. Porém, em ambientes onde coexistam dispositivos estáticos e móveis, a mobilidade dos nós pode aumentar consideravelmente a frequência dessas desconexões. Além disso, o aumento nas desconexões causadas pela mobilidade dos nós faz com que o temporizador do *Trickle* seja reiniciado com mais frequência. Assim sendo, a periodicidade na transmissão de mensagens DIO será geralmente curta (GARA et al., 2016), gerando uma maior sobrecarga na rede.

O aumento dos períodos em que um nó móvel pode ficar desconexo da DODAG, causado por um longo LOP é um problema cumulativo na rede, a medida que cresce o número de saltos. O atraso é acumulado de acordo com o número de saltos em uma determinada rota. Adicionalmente, no RPL, há um impacto gerado pelo tempo para a rede convergir e, conseqüentemente, um acréscimo no tempo necessário para que inconsistências sejam resolvidas (GHALEB; AL-DUBAI; EKONOMOU, 2015).

2.6.3.3 *Análises das Limitações do Mecanismo de Envio de Mensagens DIS*

Além da ausência de um mecanismo de gerenciamento para o envio de mensagens de controle DIO em topologias com mobilidade, outro problema em aberto na versão canônica do RPL é a falta de uma especificação de como as mensagens DIS devem ser transmitidas (COBARZAN; MONTAVONT; NOEL, 2014). Contudo, a especificação do RPL sugere que a transmissão de mensagens DIS pode ser aplicada

para averiguar possíveis métricas ou configurações da DODAG (RFC 6550, 2012). Destaca-se que as mensagens de controle DIS servem para requisitar mensagens DIO de seus nós vizinhos, possibilitando ao nó identificar qual deles deve ser selecionado como nó pai.

O envio de mensagens DIS pode ser utilizado por um nó móvel para atualizar seu nó pai assim que uma desconexão for identificada. Porém, enquanto não receber mensagens DIO de seus vizinhos, seu antigo nó pai continua sendo considerado ativo (COBARZAN; MONTAVONT; NOEL, 2014). Essa característica do protocolo pode acarretar em um aumento na perda de pacotes, devido à utilização de um rota não mais existente.

Conclui-se que um mecanismo para o envio de mensagens de controle DIS em redes móveis faz-se necessária. A principal motivação para o envio de mensagens DIS no contexto de LLN com mobilidade é o fato de que o nó móvel pode utilizar essas mensagens para monitorar sua vizinhança a fim de atualizar as possíveis rotas.

2.7 CONCLUSÕES DO CAPÍTULO

Neste capítulo, apresentou-se uma fundamentação teórica sobre os protocolos de roteamento para as MANETs, as RSSFs, e para as LLNs. Discutiu-se as limitações da aplicação dos protocolos de roteamento desenvolvidos para as RSSFs e MANETs em LLNs. Dentre as principais limitações identificadas, citam-se a sobrecarga de pacotes de controle para a construção e manutenção das topologias, e a falta de mecanismos para a manutenção local de rotas. A utilização de protocolos como o AODV, DSR e OLSR em aplicações de LLN não é adequada. As LLNs são redes projetadas para aplicações de coleta de dados, cujo padrão de comunicação é de vários nós transmitindo seus dados para um ou mais nós sorvedouros.

Os protocolos de roteamento para RSSF apresentados neste capítulo apresentam características específicas para determinados tipos de aplicação. Por exemplo, o protocolo TEEN foi projetado para aplicações de monitoramento de dados sensíveis. Contudo, para a integração de diferentes redes de sensores na Internet, faz-se necessária a utilização de tecnologias de endereçamento com baixo custo computacional como o 6LoWPAN (RFC 4944, 2007). O 6LoWPAN é uma tecnologia de adaptação de rede que permite a compressão dos cabeçalhos IPv6, de modo que estes possam ser utilizados pelos dispositivos nas LLNs.

A utilização do protocolo 6LoWPAN permite que os dispositivos de baixa capacidade, como os utilizados em LLNs, sejam unicamente identificados na Internet. Alguns protocolos apresentados neste capítulo como o Hydro, LOAD e MLOAD são compatíveis com o protocolo 6LoWPAN. Porém estes protocolos não foram projeta-

dos de maneira otimizada para as LLNs. O grupo *Internet Engineering Task Force* (IETF) projetou o protocolo *IPv6 Routing Protocol for Low-Power and Lossy Network* (RPL) (RFC 6550, 2012) para ser utilizado em LLNs. Os detalhes do protocolo RPL foram apresentados neste capítulo. Discutiu-se as limitações encontradas na especificação canônica do RPL quando o protocolo é utilizado em redes com mobilidade, e apresentou-se uma breve descrição de como a proposta desse trabalho visa atenuar essas limitações.

As limitações encontradas no RPL canônico são divididas em três categorias: (i) uso do mecanismo de envio de mensagens DIO, diretamente relacionada ao ajuste do temporizador *Trickle*; (ii) ausência de um mecanismo para o envio de mensagens DIS para que haja atualizações na topologia DODAG quando necessário; (iii) ausência de mecanismos para a identificação de mobilidade e utilização de uma métrica relacionada à mobilidade durante o procedimento de seleção de rotas.

Foram apresentados os detalhes de funcionamento do algoritmo *Trickle* utilizado pelo RPL para o gerenciamento da periodicidade de transmissão de mensagens de controle DIO. Discutiu-se, as limitações da aplicação do *Trickle* em topologias com mobilidade e os ajustes necessários.

No próximo capítulo são apresentados os trabalhos relacionados no contexto do gerenciamento da mobilidade no RPL canônico.

3 TRABALHOS RELACIONADOS

Neste capítulo, os trabalhos relevantes que lidam com o tema da mobilidade no protocolo RPL são apresentados. Na Seção 3.1, apresenta-se uma classificação dos trabalhos relacionados que tratam da mobilidade no protocolo RPL. Os trabalhos relacionados foram classificados de acordo com critérios elencados no escopo deste trabalho e também estão listados na Seção 3.1. As abordagens para o gerenciamento de mobilidade são apresentadas na Seção 3.2. Na Seção 3.3, apresentam-se as considerações relacionadas às métricas utilizadas pelas abordagens.

3.1 CLASSIFICAÇÃO DAS ABORDAGENS COM MOBILIDADE PARA O RPL

Nesta seção, apresenta-se uma classificação dos trabalhos relacionados da literatura que lidam com a mobilidade no protocolo RPL. A classificação foi realizada através da análise dos desafios que a mobilidade causa no protocolo RPL. O primeiro está relacionado com a falta de um mecanismo eficiente para a detecção da mobilidade. As propostas encontradas na literaturam foram divididas em detecção (abordagens reativas) e predição (abordagens proativas). Este tópico é influenciado pela periodicidade do envio de mensagens de controle responsáveis pela construção e manutenção da topologia RPL. O segundo está relacionado à falta de mecanismo eficiente de atualização de nó pai, pois este é um dos aspectos mais críticos do uso do RPL em LLNs com nós móveis. Na especificação canônica do RPL, um mecanismo de detecção de mobilidade e a atualização de nó pai é uma questão em aberto quando os nós são móveis. Nas LLNs compostas por nós móveis, um mecanismo de atualização do nó pai depende de um mecanismo eficiente para a detecção da mobilidade.

As duas questões em aberto mencionadas anteriormente são dependentes de dois pontos: (i) a utilização de métricas para a detecção de mobilidade e atualização de nó pai e; (ii) mecanismo para o envio de mensagens de controle DIO.

Através da revisão de literatura, foram identificados os seguintes critérios para a classificação das abordagens que contemplam estratégias para a mobilidade no RPL.

- Detecção de mobilidade: comumente empregado nas propostas reativas de mecanismos para a atualização de nós pais. O mecanismo utilizado para detecção deve ser o mais eficiente e rápido possível, visando reduzir a perda de pacotes causada pela desconexão dos nós.
- Predição de mobilidade: necessário para que os nós móveis possam prever

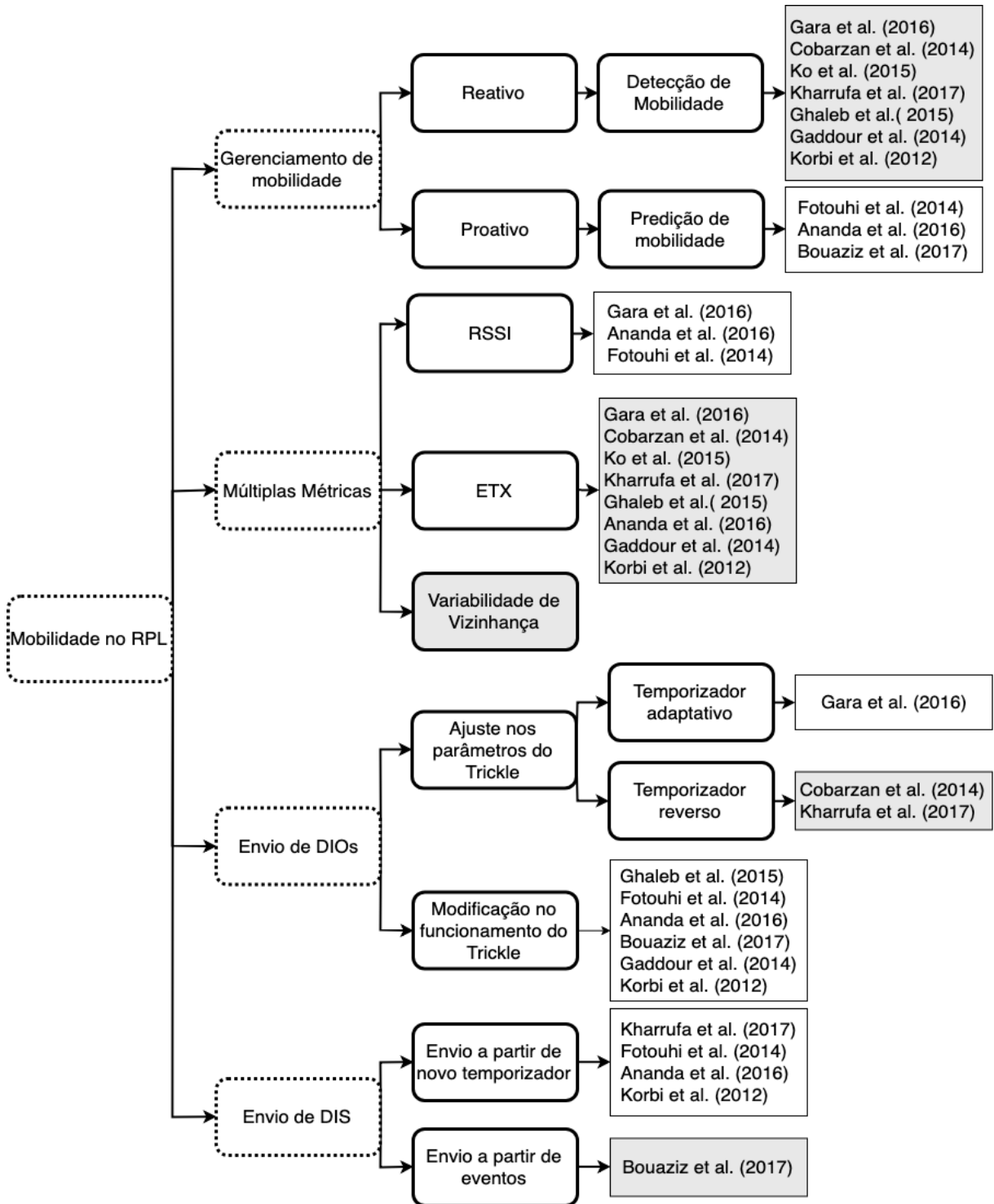
quando sairão do raio de alcance do nó pai. Para tal, faz-se necessário ter alguma informação de localização dos nós e velocidade de deslocamento do nó. Predições de mobilidade são geralmente encontradas em abordagens proativas, enquanto que a detecção de mobilidade é usada em propostas reativas. Ambas abordagens servem ao propósito de suprir a falta de uma especificação na versão canônica do RPL sobre como os nós pais devem ser atualizados, no contexto de topologias com nós móveis.

- Temporizador *Trickle* adaptativo: consiste em ajustar o temporizador com parâmetros dinâmicos. O ajuste dinâmico do *Trickle* tem como finalidade aumentar a transmissão de mensagens de controle DIO quando a topologia estiver instável e diminuir caso contrário.
- Temporizador *Trickle* reverso: reverter o temporizador de modo que os intervalos de transmissão de mensagens de controle diminuam ao invés de aumentarem como ocorre no *Trickle* canônico. Essa abordagem é utilizada quando se deseja que o número de mensagens de controle DIO seja gradualmente aumentada.
- Modificação do funcionamento do temporizador *Trickle*: alguns passos do algoritmo do temporizador *Trickle* são alterados e não somente os parâmetros.
- Envio de mensagens DIS a partir de um novo temporizador: novos temporizadores podem ser utilizados pelo nó móvel para o envio periódico de requisições sobre as medidas atuais da qualidade do enlace com seu nó pai. No presente trabalho de mestrado, mensagens DIS são utilizados pelos nós móveis quando identificam uma desconexão com seu nó pai. Os nós móveis transmitem mensagens DIO para os nós em sua vizinhança para que a DODAG possa ser atualizada.
- Envio de mensagens DIS a partir de eventos: eventos como a saída da cobertura do nó pai podem indicar ao nó móvel a necessidade de requisitar mensagens DIO, através do envio de mensagens DIS. Pode-se utilizar mensagens DIS para fins de monitoramento da degradação da qualidade do enlace entre dois nós, ou até mesmo para informar o nó pai de uma potencial desconectividade.

Na Figura 3.6, ilustram-se os trabalhos da literatura classificados de acordo com critérios listados anteriormente. Os retângulos com linhas pontilhadas representam os critérios de classificação e os retângulos em linhas sólidas, as abordagens propostas. Os autores das propostas também são explicitados na figura. As estratégias utilizadas na proposta deste trabalho de mestrado (descritos no Capítulo 4) são indicadas com a coloração cinza na Figura 3.6, por exemplo, a abordagem proposta é

reativa e oferece detecção da mobilidade. Na Seção 3.2, discutem-se os detalhes das propostas listadas na Figura 3.6.

Figura 3.6 – Classificação das Abordagens para o RPL com Topologias Móveis.



Fonte: O autor.

Além da classificação apontada na Figura 3.6, analisou-se as estratégias consideradas pelas propostas da literatura para que o RPL funcione adequadamente em redes móveis:

- Tratamento diferenciado para os nós móveis: no trabalho de Gara et al. (2016), os autores propuseram que nós móveis sejam tratados de maneira diferente dos nós estáticos. Neste caso, os nós móveis não podem rotear pacotes. Os nós móveis somente podem atuar como geradores de dados e realizar a transmissão desses dados para os nós estáticos da rede. Não permitir que nós móveis roteiem pacotes pode prejudicar a cobertura da rede. Deste modo, essas abordagens só funcionam em cenários com a cobertura assegurada por nós estáticos. No escopo do presente trabalho (descrito no Capítulo 4) não existe uma diferenciação entre nós móveis e estáticos.
- Conhecimento *a priori* da mobilidade: assume-se que os nós móveis e nós estáticos sabem *a priori* sua condição de móvel ou estático. Esta estratégia é principalmente adotada pelas abordagens proativas que tentam prever a mobilidade dos nós (GARA et al., 2016). Ou seja, os nós móveis são considerados móveis e os nós estáticos são considerados estáticos por todo o funcionamento da rede.
- Alteração de pacotes de controle do RPL: modifica-se a estrutura do cabeçalho dos pacotes de controle do RPL. Essa modificação é utilizada principalmente para que os nós da rede saibam quais nós são estáticos e quais são móveis através da adição de *flags* informativas no cabeçalho. Alterar a estrutura original do cabeçalho dos pacotes de controle de um protocolo de roteamento pode tornar a proposta incompatível com o protocolo original. No trabalho de Gara et al. (2016), adota-se esta estratégia. No escopo deste trabalho (descrito no Capítulo 4), a estrutura padrão dos cabeçalhos do RPL não é alterada, visando manter a compatibilidade com o protocolo na sua versão canônica.

3.2 ABORDAGENS PARA GERENCIAMENTO DA MOBILIDADE NO RPL

As abordagens que propõem extensões para o RPL no quesito mobilidade são classificadas como reativas ou proativas. Nesta seção, apresentam-se os trabalhos relacionados com a mobilidade no RPL, incluindo os reativos e proativos. As melhorias propostas para o *Trickle* e novos mecanismos para o envio de mensagens DIS também são discutidas nesta seção.

3.2.1 Detecção Reativa de Mobilidade no RPL

Gara et al. (2016) apresentam um temporizador adaptativo para que o RPL lide com a mobilidade dos nós. O temporizador considera a movimentação aleatória, tempo de pausa e velocidade dos nós para o gerenciamento da mobilidade. Além do temporizador adaptativo, os autores apresentam uma alteração no procedimento de seleção do nó pai com a utilização da métrica de RSSI em conjunto com ETX na função objetivo MHROF (GARA et al., 2016).

Para modificar o temporizador, os autores incluíram a variável *Time to Leave* (TL). A TL representa o tempo previsto para que o nó móvel saia da área de cobertura do seu nó pai. O valor de TL é calculado a partir da distância do nó móvel em relação ao seu nó pai, e da sua direção. Os autores apresentaram simulações com 10 nós estáticos e um número de 20 a 100 nós móveis. Nos experimentos, mediu-se a sobrecarga, atraso, taxa de sucesso de entrega e consumo de energia. Segundo os autores, a proposta obteve bons resultados em todas as métricas avaliadas.

O trabalho de Gara et al. (2016) apresenta dois pontos negativos. O primeiro é a necessidade de alteração da estrutura dos pacotes de controle, o que torna a modificação incompatível com implementações baseadas na especificação canônica do RPL. O segundo é a utilização da distância euclidiana, calculada com base nas posições dos nós. Outrossim, os autores não especificam a maneira que essas posições seriam descobertas.

Na abordagem de Cobarzan et al. (2014), os autores argumentam que um nó móvel pode esperar muito tempo por uma mensagem DIO devido aos valores padronizados do temporizador *Trickle* (p.ex. um cenário em que um nó móvel está aguardando o recebimento de mensagens DIO para ingressar em uma DODAG, enquanto os nós vizinhos estão no estado de LOP). Segundo os autores, mesmo que o RPL já disponha de mecanismos que permitam um nó reingressar na topologia após uma desconexão, ainda faltam mecanismos mais eficientes para que a detecção dessa desconexão seja identificada rapidamente (COBARZAN; MONTAVONT; NOEL, 2014).

Neste contexto, Cobarzan et al. (2014) propõem um mecanismo para detecção de inalcançabilidade de um nó e rápido reingresso na topologia, assegurando que somente o nó móvel e o nó pai estejam envolvidos nesse processo. O objetivo é reduzir a sobrecarga e o atraso. Esse mecanismo opera através da utilização de um temporizador reverso para o *Trickle* quando o nó em questão possui um nó filho móvel. Revertendo o funcionamento do temporizador *Trickle*, a periodicidade de transmissão de mensagens de controle DIO começa reduzida durante o processo de atualização da topologia RPL e aumenta gradativamente com o passar do tempo. Cobarzan et al. (2014) afirmam que a proposta se baseia na ideia de que a probabilidade do nó mover-

se novamente, e desconectar-se do seu nó pai, aumenta com o passar do tempo.

Assim como Gara et al. (2016), Corbazan et al. (2014) altera o cabeçalho dos pacotes de controle do RPL. Os autores assumem que os nós móveis se locomovem com maior frequência no decorrer do tempo. Ou seja, que o tempo de pausa dos nós móveis diminui ao longo do tempo. O temporizador reverso de Corbazan et al. (2014) tem inicialmente um intervalo curto, e cresce exponencialmente ao longo da execução do algoritmo. Essa premissa pode não ser verdade em cenários nos quais os nós móveis possuem tempo de pausa aleatória (GARA et al., 2016). A proposta de Corbazan et al. (2014) determina que os nós móveis não devem realizar roteamento de pacotes. A justificativa de tal escolha é devido ao fato de que nós móveis ao rotear os pacotes, podem provocar ciclos na topologia (COBARZAN; MONTAVONT; NOEL, 2014). Contudo, o RPL possui nativamente mecanismos para lidar com ciclos (RFC 6550, 2012). Ademais, a utilização de nós móveis no processo de roteamento pode acrescentar algumas vantagens como uma maior diversidade de rotas.

Ko et al. (2015) propuseram uma camada de rede com suporte à mobilidade para o rápido restabelecimento de rotas assim que uma desconexão, causada pela mobilidade, é detectada. Este mecanismo de detecção, chamado de MoRoRo, é baseada na perda de pacotes e utiliza Lógica *Fuzzy* para estimar a qualidade do enlace mais indicado para a nova rota.

Os autores mencionam que as abordagens da literatura impedem os nós móveis de rotear pacotes, ou seja, podem apenas agir como nós folhas na topologia. Apesar de tais abordagens apresentarem um bom desempenho em termos de confiabilidade e estabilidade na rede, requerem um alto planejamento antes da rede entrar em execução (KO; CHANG, 2015), pois os nós estáticos precisam ser alocados em posições estratégicas para que haja cobertura suficiente. A proposta deste trabalho de mestrado leva este fato em consideração e permite que os nós móveis possam rotear pacotes.

Algumas propostas proativas de gerenciamento de mobilidade encontradas na literatura utilizam mecanismos de *hand-off*, como Fotouhi et al. (2014) e Anand et al. (2016). Porém, Ko et al. (2015) apontaram possíveis problemas para o uso de tais mecanismos. Estas abordagens permitem o nó móvel trocar de nó pai antes que uma desconexão por inalcançabilidade entre os nós aconteça. Em contrapartida, podem produzir alta troca de mensagens de controle na tentativa de prever a mobilidade dos nós. Segundo Ko e Chang (2015), as propostas baseadas em *hand-off*, não são adequadas para os dispositivos de baixa capacidade como os geralmente empregados em LLNs, pois aumenta a sobrecarga na rede e o consumo de energia.

Outro aspecto negativo da proposta de Ko et al. (2015) é o fato do estima-

dor de mobilidade basear-se na perda de pacotes. A perda de pacotes é detectada pela camada de rede e informada ao MoRoRo para que este possa verificar a inalcançabilidade do nó pai (KO; CHANG, 2015). Contudo, a perda de pacotes pode ser influenciada por flutuações na qualidade do enlace até mesmo em topologias estáticas, e trocas de nós pais podem ser realizadas desnecessariamente com a proposta de Ko et al. (2015).

Kharrufa et al. (2017) apresentam uma melhora para o RPL no escopo do algoritmo *Trickle* chamado D-RPL. Ao receber um pacote de um determinado nó vizinho, o nó em questão compara o valor de RSSI atual do pacote com o valor de RSSI medido em transmissões anteriores deste mesmo vizinho. Se o valor de RSSI atual somado com uma constante K_{rssi} for menor do que a medição anterior, o algoritmo *Trickle* é executado com o seu temporizador reverso, assim como na proposta de Gara et al. (2016). Ao mesmo tempo que o temporizador reverso é executado, o nó envia mensagens DIS para todos os seus vizinhos para avaliar as opções de nó pai disponíveis (KHARRUFA et al., 2017). Os autores também defendem a especificação de um novo mecanismo para o envio de mensagens DIS para analisar a vizinhança do nó móvel. As mensagens DIS serão transmitidas em *multicast* para todos os vizinhos. A recepção de DIS transmitidas em *multicast*, seguindo a especificação canônica do RPL, podem causar o reinício do temporizador *Trickle*.

Ghaleb et al. (2015) propõem uma modificação ao algoritmo *Trickle* utilizado pelo RPL, substituindo seu mecanismo canônico de supressão de transmissão. No algoritmo de Ghaleb et al. (2015), estima-se o período em que o nó ficaria proibido de enviar mensagens DIO. As modificações propostas pelos autores são divididas em três etapas. A primeira, consiste em trocar o intervalo fechado que a variável t , representando o período de supressão, pode assumir. O intervalo deixa de ser $[\frac{I}{2}, I]$ e passa a ser $[0, I]$. Na segunda, muda-se o modo com que o valor da variável c , representando o limiar utilizado na supressão, é alterada. No lugar de reiniciar o seu valor para 0 no começo de cada intervalo, o valor é reiniciado apenas no primeiro intervalo em I_{min} e durante o tempo t é escolhido aleatoriamente. A terceira modificação consiste em tornar dinâmico o valor da constante k de acordo com o tamanho do intervalo vigente. Essa última modificação pretende reduzir as chances de transmissão dos nós com menores intervalos de tempo. Isto porque o valor de c seria reiniciado para zero no tempo aleatório t , provocando intervalos desiguais entre os nós (GHALEB; AL-DUBAI; EKONOMOU, 2015). Os resultados apresentados pelos autores indicam que houve redução no tempo de convergência sem aumentar o consumo de energia e taxa de sucesso de transmissão. Contudo, não foram realizados experimentos com nós móveis.

Co-RPL é uma extensão para o RPL, proposto por Gaddour et al. (2014). Co-

RPL divide a rede em regiões circulares chamadas de coronas. Cada corona possui uma identificação única e cada nó na rede pode pertencer a apenas uma corona por vez (GADDOUR et al., 2014). No centro de cada corona, localiza-se um nó sorvedouro. As mensagens de controle DIS e DIO são modificadas no Co-RPL. No Co-RPL, o cabeçalho da mensagem de controle DIS contém um bit para diferenciar as mensagens DIS do Co-RPL com as mensagens DIS do RPL. O cabeçalho da mensagem de controle DIO também contém o número de identificação da corona e o bit para diferenciar mensagens DIO do RPL. No Co-RPL, os nós não podem selecionar um nó pertencente a mesma corona como nó pai. A motivação é mitigar o surgimento de ciclos. Os autores modificam o funcionamento do mecanismo de envio das mensagens de controle DIO, de modo que os nós enviam mensagens DIO imediatamente em duas situações: assim que um nó ingressa na DODAG e no recebimento de mensagens DIS do Co-RPL.

O protocolo *Mobility Enhanced RPL* (ME-RPL) foi proposto por Korbi et al. (2012). No ME-RPL, os nós móveis e estáticos diferenciam-se através de uma identificação no cabeçalho do pacote de controle (KORBI et al., 2012), assim como proposto por Gaddour et al. (2014). Os autores não propuseram alterações no temporizador *Trickle*, como Corbazan et al. (2014) e Ghaleb et al. (2015). No ME-RPL, utiliza-se um temporizador dinâmico para o envio de mensagens DIS. O envio de mensagens DIS é realizado para que a topologia possa ser atualizada assim que identificar sua movimentação. A movimentação dos nós é detectada através do monitoramento da frequência de atualizações do nó pai (KORBI et al., 2012). O ME-RPL se baseia na premissa de que informações históricas do nó em relação à frequência na troca de nó pai, é um indicativo de movimentação dos nós. Ou seja, no ME-RPL, assume-se que um nó com alta troca de nó pai tende a manter este comportamento no futuro.

3.2.2 Detecção Proativa de Mobilidade no RPL

Fotouhi et al. (2015) propuseram um protocolo de roteamento denominado *Mobile Compliant RPL* (mRPL) utilizando um mecanismo de *hand-off*, *SmartHop*, (FOTOUHI et al., 2012) que estende o funcionamento padrão do RPL. Segundo os autores, este mecanismo permite aos dispositivos que estão executando o protocolo RPL realizar a troca de nós pais com baixo custo em termos de sobrecarga e consumo de energia. Os autores defendem a aplicação de um mecanismo proativo de detecção de mudanças na topologia (FOTOUHI; MOREIRA; ALVES, 2015).

Os autores adotam a premissa de que os nós móveis sabem de sua mobilidade *a priori*. Neste modelo, os nós móveis são tratados diferentemente dos nós estáticos, característica comum em propostas proativas de gerenciamento de mobilidade.

O algoritmo *SmartHop* utiliza três parâmetros que determinam o seu desempenho. O primeiro parâmetro é o número de pacotes requeridos para que o RSSI médio seja calculado durante um período de tempo. O segundo parâmetro é um limiar que define o valor mínimo para o RSSI que inicia um *hand-off* e um valor máximo para terminá-lo. É importante frisar que valores muito baixos para o valor mínimo podem provocar *hand-offs* desnecessários. O terceiro parâmetro é o número extra de mensagens de controle que os nós móveis transmitem no processo de troca do nó pai. Esta troca de mensagens serve ao propósito de garantir que o nó móvel tenha escolhido um pai que tenha permanecido estável por um determinado tempo. O terceiro parâmetro está diretamente relacionado com o segundo (FOTOUHI; MOREIRA; ALVES, 2015).

Além dos três novos parâmetros, os autores propõem a utilização de mais quatro temporizadores em conjunto com o *Trickle*. Estes novos temporizadores visam permitir que os nós identifiquem rapidamente a degradação do enlace em um curto período de tempo (FOTOUHI; MOREIRA; ALVES, 2015).

Um aspecto negativo da proposta do Fotouhi et al. (2014) diz respeito ao uso do valor de RSSI como a única informação para seleção do nó pai. Neste contexto, os autores não utilizaram a função objetivo, como definido no RPL canônico (ANAND; TAHILIANI, 2016).

No trabalho de Anand et al. (2016), os autores demonstram que não empregar a função objetivo pode conduzir o nó móvel a selecionar caminhos sub-ótimos. A não utilização da função objetivo para a seleção do nó pai na proposta de Fotouhi et al. (2015) pode provocar ciclos que o mecanismo canônico do RPL não poderia identificar e corrigir. No RPL canônico, os ciclos são detectados e corrigidos com base nos valores de *rank* que são calculados pela função objetivo. Fotouhi et al. (2015) propuseram uma nova regra para mitigar os ciclos: em topologias com nós móveis, os nós filhos de um nó móvel não podem se tornar o nó pai desse mesmo nó logo após uma desconexão (FOTOUHI; MOREIRA; ALVES, 2015).

Anand et al. (2016) apresentam uma modificação para o protocolo mRPL. Os nós móveis selecionam um nó pai baseado somente no valor de RSSI e não por uma função objetivo (ANAND; TAHILIANI, 2016). Para corrigir tal problema, os autores consideram a média dos *ranks* dos nós pais candidatos em conjunto com o valor RSSI. A proposta foi avaliada através de simulações no Cooja e com a métrica ETX na função objetivo. Essas simulações seguiram um modelo parecido com o de Fotouhi et al. (2014) em relação às métricas avaliadas e quantidades de nós.

Como apontado pelos autores Anand et al. (2016), há a necessidade de se avaliar essa proposta em cenários com diferentes topologias e modelos de mobilidade, e também de se analisar os melhores ajustes nos parâmetros utilizados pelo mRPL

(ANAND; TAHILIANI, 2016).

Outros trabalhos encontrados na literatura visam melhorar o protocolo mRPL através da inclusão de modelos de predição de mobilidade. Bouaziz et al. (2017) defendem a utilização de um mecanismo para a predição da movimentação do nó baseado em RSSI, sendo os nós estáticos próximos ao nó móvel responsáveis por realizarem a predição (BOUAZIZ; RACHEDI; BELGHITH, 2017). O mecanismo de predição visa solucionar dois problemas. O primeiro é relacionado ao alto custo das abordagens proativas em termos de consumo de energia. O segundo problema refere-se ao fato de que os valores de RSSI podem facilmente sofrer interferência.

Duas premissas são consideradas em Bouaziz et al. (2017) para que a proposta funcione. A primeira é que as posições de todos os nós estáticos da rede são de conhecimento do nó móvel desde o começo do funcionamento da rede. Somente a posição do nó móvel é predita pelos estáticos. A direção do nó móvel é calculada através das distâncias entre o nó móvel e os nós estáticos, em conjunto com os valores de RSSI medidos pelos nós estáticos. A necessidade do conhecimento *a priori* da posição inicial dos nós estáticos limita a aplicabilidade da solução e está relacionada com a segunda premissa, pois nesse caso, os nós móveis devem ser sempre móveis e os nós estáticos sempre estáticos.

3.3 CONSIDERAÇÕES SOBRE AS MÉTRICAS UTILIZADAS NA LITERATURA

Na Seção 3.2.1 e Seção 3.2.2 foram apresentadas propostas para o gerenciamento da mobilidade. O gerenciamento é realizado através da análise de uma ou mais métricas da qualidade de enlace. A métrica RSSI é amplamente utilizada para este fim. Os valores dessa métrica podem servir para identificar se o nó móvel está saindo da área de cobertura do nó pai.

Dong e Dargie (2012) afirmam que a utilização do valor de RSSI pode não ser a mais indicada para ambientes fechados devido a alguns elementos que podem influenciar o valor medido, como a distância física, polarização da antena utilizada pelo dispositivo e fatores como a reflexão, refração e difração das ondas eletromagnéticas (DONG; DARGIE, 2012). Desse modo, faz-se necessária uma melhor utilização do RSSI através da combinação com outras métricas para evitar trocas desnecessárias de nós pais.

A partir da análise dos trabalhos relacionados, conclui-se que as métricas mais utilizadas em conjunto com os valores de RSSI para a detecção ou predição de mobilidade são: distâncias entre o nó móvel e os nós estáticos, e velocidade. Essas métricas são geralmente dependentes de informações como posições geográficas ou de modelos de movimentação dos nós. As informações de posições geográficas são co-

mumente obtidas através de tecnologias como o *Global Positioning System* (GPS). A utilização de tal tecnologia pode oferecer vantagens para o protocolo de roteamento como, seleção de rotas mais curtas. Porém, a principal desvantagem é o considerável custo energético, tendo em vista que os dispositivos utilizados em LLNs operam com capacidade restrita de energia residual. A utilização de métricas baseadas em modelos de movimentação dos nós, utilizadas como forma de predição da direção da movimentação do nó, demandam poder computacional pelos dispositivos por serem baseados em modelos estatísticos de predição. Logo, sua aplicação pode ser restrita em determinados dispositivos de baixo poder de processamento como os utilizados em LLNs.

Este trabalho de mestrado propõe a utilização de uma métrica chamada Variabilidade de Vizinhança para o mecanismo de gerenciamento de mobilidade e atualização de nó pai. A Variabilidade de Vizinhança está relacionada com a movimentação dos nós de sua vizinhança. Maiores detalhes da definição dessa métrica são apresentados no Capítulo 4.

3.4 CONCLUSÕES DO CAPÍTULO

Neste capítulo, foi apresentada uma classificação dos trabalhos relacionados da literatura em termos do oferecimento de mecanismos para o gerenciamento da mobilidade, a utilização de métricas para o monitoramento da mobilidade e atualização mais eficiente de nó pai.

As propostas de gerenciamento de mobilidade são categorizados como reativas ou proativas. As reativas executam mecanismos de manutenção da topologia após a detecção de uma desconexão entre o nó móvel e seu nó pai. As proativas realizam a troca preventiva de nós pais ao prever que o nó móvel sairá do alcance de seu nó pai em um espaço curto de tempo. Abordagens que tentam prever posições futuras do nó móvel podem não ser muito eficientes em aplicações com nós que possuam movimentação aleatória. Outro ponto importante acerca das propostas proativas de gerenciamento de mobilidade é a utilização de modelos de predição, cujo cálculo pode ser custoso computacionalmente para os dispositivos utilizados em LLNs.

As abordagens que propõem melhorias para o algoritmo *Trickle*, responsável por regular a periodicidade em que as mensagens de controle DIO devem ser transmitidas pelos nós, foram divididas em três categorias. As duas primeiras categorias são relacionadas ao ajuste de parâmetros do temporizador *Trickle*: temporizador adaptativo e temporizador reverso. Um exemplo de temporizador adaptativo pode ser encontrado em Gara et al. (2016). Cobarzan et al. (2014) e Kharrufa et al. (2015) são exemplos de propostas que utilizam um temporizador reverso. A terceira categoria

está relacionada a modificações no funcionamento do temporizador *Trickle* conforme apresentado no trabalho de Bouaziz et al. (2017).

As mensagens de controle DIS podem ser utilizadas pelos nós móveis para rapidamente detectar as desconexões causadas pela mobilidade, ou para iniciar um procedimento de manutenção da DODAG. Porém, a especificação canônica do RPL não descreve como e quando o envio de mensagens DIS devem acontecer. Sua utilização em topologias com nós móveis é descrita em alguns trabalhos como uma forma dos nós avaliarem sua vizinhança. O envio dessas mensagens pode ser a partir de novos temporizadores inseridos no protocolo ou através de respostas a eventos como, a mobilidade ou a predição de que o nó sairá da área de cobertura de seu nó pai. Exemplos da adição de novos temporizadores ao algoritmo *Trickle* foram propostos em Fotouhi et al. (2015).

Na Tabela 3.1, apresenta-se uma comparação dos trabalhos relacionados em termos dos mecanismos para o envio das mensagens de controle DIO e DIS. Nota-se que uma boa parte dos trabalhos reativos não consideram o uso das mensagens DIS por parte do nó móvel, como por exemplo, Gara et al. (2016), Cobarzan et al. (2014) e Ko et al. (2015). Verificou-se que os trabalhos reativos em sua maioria modificam o mecanismo de ajuste dos parâmetros do temporizador *Trickle* sem alterar seu funcionamento interno. Por outro lado, a maioria dos trabalhos proativos propõe modificações no funcionamento do algoritmo *Trickle*, e utilizam as mensagens DIS por parte do nó móvel para avaliar sua vizinhança. Por exemplo, Fotouhi et al. (2015).

Tabela 3.1 – Comparação dos Trabalhos Relacionados: Mecanismos para o Envio de Mensagens de Controle DIO e DIS.

Autores	Envio de DIO	Envio de DIS
Gara et al. (2016)	Temporizador adaptativo	-
Corbazan et al. (2014)	Temporizador reverso	-
Ko et al. (2015)	-	-
Kharrufa et al. (2017)	Temporizador reverso	Novo temporizador
Ghaleb et al. (2015)	Modificação do <i>Trickle</i>	-
Fotouhi et al. (2014)	Modificação do <i>Trickle</i>	Novo temporizador
Ananda et al. (2016)	Modificação do <i>Trickle</i>	Novo temporizador
Bouaziz et al. (2017)	Modificação do <i>Trickle</i>	Envio a partir de eventos
Korbi et al. (2012)	Modificação do <i>Trickle</i>	Novo temporizador
Gaddour et al. (2014)	Modificação do <i>Trickle</i>	-
Marques (2019)	Temporizador reverso	Envio a partir de eventos

Na Tabela 3.2, apresenta-se uma comparação dos trabalhos relacionados em termos das métricas utilizadas para o gerenciamento da mobilidade ou otimizadas através da função objetivo para a seleção das melhores rotas. Observa-se que a maioria das propostas proativas de detecção de mobilidade empregam métricas baseadas nos valores de RSSI na função objetivo. A utilização dessa métrica é justificável, pois

ajuda na seleção do novo nó pai, de modo que o nó vizinho mais próximo é favorecido. No escopo deste trabalho, a métrica Variabilidade de Vizinhança é utilizada. Esta métrica é baseada em valores de RSSI medidos durante o recebimento de mensagens de dados e controle. Acredita-se que a Variabilidade de Vizinhança é uma métrica robusta para determinar a mobilidade dos nós, uma vez que não utiliza os valores de RSSI diretamente, como proposto nos trabalhos relacionados discutidos anteriormente.

Tabela 3.2 – Comparação dos Trabalhos Relacionados: Mecanismos de Gerenciamento da Mobilidade, Métricas para Gerenciamento da Mobilidade ou Otimizadas na Função Objetivo.

Autores	Gerenciamento de mobilidade	Métricas utilizadas
Gara et al. (2016)	Detecção Reativa	RSSI e ETX
Corbazan et al. (2014)	Detecção Reativa	ETX
Ko et al. (2015)	Detecção Reativa	ETX
Kharrufa et al. (2017)	Detecção Reativa	ETX
Ghaleb et al. (2015)	Detecção Reativa	ETX
Fotouhi et al. (2014)	Detecção Proativa	RSSI
Ananda et al. (2016)	Detecção Proativa	RSSI e ETX
Bouaziz et al. (2017)	Detecção Proativa	ETX
Korbi et al. (2012)	Detecção Reativa	ETX
Gaddour et al. (2012)	Detecção Reativa	ETX
Marques (2019)	Detecção Reativa	ETX e Variabilidade de Vizinhança

Métricas baseadas em RSSI veem sendo adotadas em trabalhos que tratam da mobilidade, pois permitem ao nó móvel escolher uma rota cuja conexão tem maior probabilidade de durar por mais tempo (p.ex. identificar qual é, ou será, o nó vizinho mais próximo). Porém, sua utilização exclusiva no processo de seleção do nó pai, pode levar o nó móvel a escolher rotas sub-ótimas (ANAND; TAHILIANI, 2016). Esta métrica utilizada isoladamente pode reduzir a eficiência do mecanismo de detecção de ciclos do RPL, necessitando-se de novos mecanismos como, os expostos em Fotouhi et al. (2015). Conclui-se, que as métricas baseadas em RSSI devem ser utilizadas em conjunto com a ETX ou outras métricas que possam indicar a qualidade da rota.

A abordagem proposta (descrita no Capítulo 4) permite que nós móveis possam rotear pacotes. O impacto que as desconexões podem causar no desempenho da rede é reduzido através de ajustes no temporizador *Trickle* e pelo mecanismo utilizado para identificar as desconexões.

Um potencial problema em permitir que nós móveis roteiem pacotes é aumentar a taxa de perda de pacotes e desconexão na rede, dependendo do padrão de mobilidade do nó. A presente proposta aborda essa problemática com uma nova métrica que permite mensurar a probabilidade de um nó estar móvel ou estático. Esta métrica, chamada de Variabilidade de Vizinhança, é calculada através das variação das potências de sinal medidas durante o recebimento de pacotes de seus vizinhos.

O procedimento de seleção de nó pai, na abordagem proposta, utiliza a métrica Variabilidade de Vizinhaça para escolher o nó pai com menor mobilidade. Utiliza-se também os *ranks* dos nós, calculados pela função objetivo. Dessa maneira, o nó móvel poderá escolher um novo nó pai com alta probabilidade de estar estático e mais próximo do nó sorvedouro.

No Capítulo 4, apresentam-se maiores detalhes dos mecanismos propostos para a rápida detecção de mobilidade, para o ajuste no temporizador *Trickle* e para a atualização do nó pai.

4 ABORDAGEM PARA O SUPORTE A MOBILIDADE NO RPL

Neste capítulo apresenta-se uma abordagem para suporte a mobilidade no protocolo RPL. A abordagem proposta, denominada de *Mobility Aware RPL* (MARPL), apresenta um mecanismo para a detecção da mobilidade e utiliza modelo reativo para identificação de desconexões.

No MARPL, os nós móveis e estáticos são tratados indistintamente, ou seja, os nós móveis podem ser utilizados para o roteamento de pacotes diferentemente do proposto em Corbazan et al. (2014). Os autores apresentaram uma proposta de gerenciamento de mobilidade que impede os nós móveis de rotearem pacotes. Através do tratamento sem distinção entre nós estáticos e móveis uma boa cobertura da rede pode ser obtida. Ressalta-se que tratar nós estáticos e móveis indistintamente oferece mais dois benefícios:

- Diversificação das rotas: auxilia na resolução do problema conhecido por *hot spot*. Tal problema, relaciona-se com o fato de que os nós mais próximos do nó sorvedouro ficam sobrecarregados devido à alta taxa de pacotes sendo transmitidos por meio destes nós. Essa sobrecarga é perceptível principalmente em topologias densas resultando em congestionamento na rede, e por consequência, perdas de pacotes e atraso de transmissão.
- Otimização do desempenho: contribui para a redução do consumo de energia dos nós estáticos (YOUNIS et al., 2014). A redução do consumo de energia dos nós estáticos pode prolongar o tempo de vida da rede devido a uma melhor distribuição do consumo de energia entre os nós.

Assume-se neste trabalho as seguintes premissas: os nós são móveis, porém possuem algum momento de estabilidade. Em outras palavras, os nós ficam estáticos por alguns períodos de tempo. Outra premissa considerada, é a de que os nós móveis e estáticos não executam versões diferentes do protocolo. Ou seja, ambos os nós estáticos e móveis executam o MARPL (compatível com o RPL Canônico). A identificação do estado do nó (móvel ou estático) fica a cargo do mecanismo de detecção de mobilidade proposto no MARPL.

Na Seção 4.1, será apresentado o mecanismo proposto para o gerenciamento de mobilidade no MARPL. O gerenciamento dá-se através do monitoramento da vizinhança do nó e identificação das diferenças de potência do sinal, RSSI, das mensagens de dados e controle recebidas. O mecanismo utiliza a métrica proposta, Variabilidade da Vizinhança (γ), cujo objetivo no escopo deste trabalho é representar

a movimentação de um nó. A utilização da métrica Variabilidade de Vizinhança pelo processo de seleção do nó pai é apresentada na Seção 4.2.

4.1 MECANISMO DE GERENCIAMENTO DE MOBILIDADE PARA O RPL

O mecanismo de gerenciamento de mobilidade, conforme mencionado previamente, é reativo. Portanto, a identificação da desconexão ocorre somente quando um nó sai da área de cobertura de seu nó pai. No escopo deste trabalho, assume-se que não haverá tentativa de prever a desconexão com base em dados históricos do nó móvel como na proposta de Bouaziz et al. (2017). Desta maneira, busca-se não impactar negativamente o desempenho geral da rede com a troca desnecessária de mensagens de controle quando um nó estiver na área de cobertura de seu nó pai. O mecanismo de gerenciamento de mobilidade ajusta a troca de mensagens de controle DIO e DIS somente nos momentos em que a mobilidade do nó for detectada. À vista disso, os impactos negativos decorrentes das desconexões na rede são minimizados.

Durante o funcionamento da rede, no MARPL, todos os nós monitoram a variabilidade de suas respectivas vizinhanças. Os nós mantêm atualizada uma tabela de vizinhança, a exemplo do RPL canônico. Na tabela os nós armazenam o endereço IP dos nós vizinhos e as duas potências de sinal, RSSI, mais recentes medidas durante o recebimento de mensagens de dados ou controle (DIO, DIS e DAO). Um nó é considerado um nó vizinho quando houver um recebimento bem sucedido de um pacote pela camada de Acesso ao Meio. Através das potências de sinal armazenadas na tabela de vizinhança, o nó calcula a métrica Variabilidade de Vizinhança (γ). No recebimento de uma mensagem, seja de controle (p.ex. DIO, DIS ou DAO) ou de dados, caso o nó vizinho não esteja na tabela de vizinhança do nó em questão, o nó vizinho é então registrado.

Por meio do monitoramento da vizinhança, os nós executando o MARPL podem identificar uma desconexão com o nó pai. A desconexão é identificada, se após um determinado período de tempo, o nó não receber de seu nó pai um pacote de dado ou controle. Neste caso, considera-se que há uma probabilidade do nó pai não encontrar-se mais ao alcance do nó em questão (detalhes estão na Subseção 4.1.1).

A especificação canônica do RPL utiliza o temporizador *Trickle* para gerenciar a transmissão das mensagens de controle DIO de maneira a suprimir transmissões dessas mensagens por um período de tempo para reduzir a sobrecarga. Este temporizador é regulado mediante dois parâmetros de configuração I_{min} e I_{max} . I_{min} representa o período mínimo do intervalo de supressão do *Trickle* e I_{max} representa o período máximo, correspondendo ao número de vezes no qual o valor de I_{min} deve ser dobrado (RFC 6206, 2011). No MARPL, propõe-se que os nós pais possam ajustar o

Tabela 4.3 – Parâmetros do MARPL.

Parâmetro	Descrição
I_{min}	Período mínimo para o envio de mensagens DIO.
I_{max}	Período máximo para o envio de mensagens DIO.
$T_{monitoring}$	Período em que o nó deve monitorar sua vizinhança. Seu valor é determinado pela frequência das transmissões de dados coletados pelos sensores.
$T_{reachable}$	Período máximo em que um nó deve aguardar o recebimento de alguma mensagem de seu nó pai, a fim de determinar sua alcançabilidade. A mensagem pode ser de dados ou controle. O valor de $T_{reachable}$ também é determinado pela frequência na transmissão de dados, assim como $T_{monitoring}$.
θ	Parâmetro de utilizado para ajustar o temporizador $T_{reachable}$.
γ	Métrica Variabilidade de Vizinhança. Utilizada para determinar a mobilidade de um nó.
y	O número de vizinhos com duas medidas consecutivas de RSSI.
K_{γ}	Limiar de normalização da métrica γ . Seu valor é determinado por cada nó através da maior variabilidade de vizinhança já identificada pelo nó.

temporizador *Trickle* de acordo com a Variabilidade de Vizinhança de seus nós filhos. Este ajuste é feito de maneira a aumentar a transmissão de mensagens DIO. Esta medida aumenta a troca de mensagens de controle entre os nós pai e filho somente quando o nó filho estiver em movimentação. O aumento temporário no envio de mensagens DIO por parte de um nó pai pode auxiliar na prevenção de desconexões e na manutenção da topologia.

O mecanismo de gerenciamento de mobilidade proposto neste trabalho de dissertação adota uma abordagem entre camadas. A camada de Rede utiliza medições da potência de sinal vindas da camada de Acesso ao Meio. A potência do sinal medida é utilizada no cálculo da métrica Variabilidade de Vizinhança (γ) de um nó. A proposta para o monitoramento de vizinhança será descrito na Subseção 4.1.1.

Na Tabela 4.3, apresentam-se os parâmetros utilizados ou ajustados pelo MARPL. Como pode ser visto na tabela, é proposto no MARPL a utilização de um novo temporizador, aplicado no cálculo da métrica γ . Na Equação 4.1, apresenta-se o cálculo do ajuste do temporizador $T_{monitoring}$. Este temporizador é adotado no monitoramento da vizinhança e atualização da métrica γ . Seu valor é ajustado de acordo com a periodicidade no envio de pacotes de dados, do inglês *Data Packet Period* (DPP). A periodicidade é o intervalo de tempo entre duas amostras de dados coletados na rede. O valor de DPP pode variar em domínios diferentes de aplicação de IoT. Assume-se que os nós sensores conhecem o valor de DPP. A utilização do temporizador $T_{monitoring}$ no

MARPL é melhor descrita na Subseção 4.1.1.

$$T_{monitoring} = DPP \quad (4.1)$$

O MARPL possui um mecanismo para detectar a desconexão de um nó com seu nó pai. Este mecanismo funciona através do temporizador $T_{reachable}$. Este temporizador é ajustado através da Equação 4.2. A motivação para o desenvolvimento de $T_{reachable}$ é que supostamente dentro de um período determinado pelo DPP da aplicação, um nó já teria coletado o dado e transmitido este dado para a rede. Logo, se o nó pai de um nó filho não transmitiu ao menos uma mensagem de dados dentro do período definido pelo DPP, assume-se a possibilidade de que este nó pai esteja fora da área de alcance do nó filho e a DODAG necessita ser atualizada. No MARPL, a atualização é iniciada pelo nó que identifica a inalcançabilidade de seu nó pai, através do envio de uma mensagem DIS para todos os seus vizinhos. Este ato pode causar o reinício do temporizador *Trickle* em alguns vizinhos caso seja identificado uma inconsistência na topologia, segundo a especificação canônica do RPL (RFC 6550, 2012), possibilitando a atualização na DODAG.

Contudo, $T_{reachable}$ não depende somente do valor de DPP. Um parâmetro de tolerância θ é adicionado. θ é utilizado para regular um tempo adicional no mecanismo para que o nó possa iniciar o procedimento de atualização da topologia. Este tempo adicional de espera definido por θ é necessário pois um nó pode não ter recebido uma mensagem de dado ou controle de seu nó pai por outros motivos além da desconexão dos nós (p.ex. o enlace pode estar congestionado no momento da transmissão de um pacote). Define-se que o valor de θ seja maior ou igual a 1. Ou seja, $T_{reachable}$ deve ter no mínimo o valor do DPP.

$$T_{reachable} = \theta \times DPP \mid \theta \geq 1 \quad (4.2)$$

4.1.1 Monitoramento de Vizinhança no MARPL

No MARPL os nós monitoram constantemente a movimentação de sua vizinhança através do recebimento de mensagens de controle do RPL (p.ex. DIO, DIS e DAO) ou mensagens de dados. Cada nó na rede tem uma tabela de vizinhança. A tabela de vizinhança armazena para cada vizinho o endereço 6LoWPAN, os últimos dois valores de RSSI medidos no recebimento de mensagens e o momento da última atualização das informações de cada vizinho. A ausência de nós vizinhos impossibilita um monitoramento baseado na variabilidade de vizinhança. Portanto, o mecanismo de gerenciamento de mobilidade parte da premissa de que sempre haja ao menos

um nó vizinho. Vale ressaltar, que um nó realiza o monitoramento da sua vizinhança enquanto o mesmo estiver com o rádio de comunicação ativo.

Para que a tabela de vizinhança permaneça sempre atualizada, os registros dos vizinhos que não se encontram mais na vizinhança do nó devem ser removidos da tabela. Posto que, a manutenção da tabela de vizinhança é realizada a cada recebimento de mensagens de dados ou controle. O registro de um vizinho é invalidado na tabela de vizinhança do MARPL quando o seu registro na tabela de vizinhança permanece desatualizada por um período de tempo maior do que $T_{reachable}$. Contudo, o registro do nó pai não é removido por este procedimento de atualização da tabela. Tendo em vista que, o MARPL não realiza diretamente a atualização do nó pai, pois o procedimento canônico do RPL é utilizado para este propósito.

O funcionamento do MARPL é descrito no Algoritmo 4.2. Após seu ingresso em uma DODAG, um nó monitora constantemente o recebimento de mensagens de controle ou de dados advindos dos nós que estão na sua vizinhança, com a finalidade de detectar sua mobilidade. O monitoramento é iniciado através do temporizador $T_{monitoring}$. O temporizador $T_{monitoring}$ opera baseado na taxa de amostragem de dado do sensor, ou seja, a frequência de medições realizadas pelo sensor. A taxa de amostragem pode variar de acordo com as características da aplicação. Quando uma nova mensagem é recebida com sucesso pela camada de Acesso ao Meio (linha 3), o nó mede a potência de sinal (RSSI) em decibéis e atualiza o registro do vizinho na tabela de vizinhança do MARPL. O nó também registra o IP do originador da mensagem e o momento em que esta mensagem foi recebida (linha 4). A cada recebimento de mensagens, faz-se a remoção dos vizinhos não mais presentes na vizinhança do nó através do método `INVALIDATE_UNREACHABLE_NEIGHBORS` (linha 5). Este método analisa a última atualização de cada entrada na tabela de vizinhança do MARPL (linha 23). Caso um vizinho não tenha seu registro atualizado dentro do período definido por $T_{reachable}$ (linha 24), o mesmo é invalidado da tabela de vizinhança pois encontra-se inalcançável (linha 25). A invalidação de registros de vizinhos inalcançáveis é necessária para que os RSSIs medidos durante o recebimento de mensagens desses vizinhos não sejam utilizados em novos cálculos da métrica γ .

No recebimento de novos pacotes de controle ou de dados, o MARPL faz uma série de verificações. A primeira é analisar se o pacote é de controle, mais especificamente se é uma mensagem DIS ou DAO (linha 6). No MARPL, as mensagens DIS são utilizadas pelos nós móveis que identificaram uma desconexão com o seu nó pai com o objetivo de atualizar a topologia. Segundo a especificação canônica do RPL, o recebimento de uma mensagem DIS pode provocar o reinício do temporizador *Trickle*, aumentando o envio de mensagens DIO. Conseqüentemente, as correções necessárias na topologia são realizadas. As correções são feitas através das potenciais

Algoritmo 4.2 MARPL

```

1: procedure MOBILITY_MONITORING
2:   while  $T_{monitoring}$  do
3:     if received a packet then ▷ pacote de controle ou dado
4:       update sender IP,  $\gamma$ , RSSI and packet reception time in the neighbor table
5:       INVALIDATE_UNREACHABLE_NEIGHBORS
6:       if packet is a DIS or DAO then ▷ se é pacote de controle
7:         TRICKLE_ADJUSTMENT
8:       if packet is DIO then
9:          $vizinho_{rank\_novo} \leftarrow vizinho_{rank\_original} + \gamma$ 
10:      if packet is from preferred parent then
11:        restart  $T_{reachable}$  timer
12:      if  $T_{monitoring}$  timer expires then
13:        if  $var\{\Delta p_i\}_{i=1}^y > K_\gamma$  then ▷ se a variância é maior do que  $K_\gamma$ 
14:           $K_\gamma = var\{\Delta p_i\}_{i=1}^y$  ▷ atualizar  $K_\gamma$  com a maior variância  $var$ 
15:           $\gamma \leftarrow var\{\Delta p_i\}_{i=1}^y / K_\gamma$ 
16:          restart  $T_{monitoring}$  timer
17:      if  $T_{reachable}$  timer expires then
18:        if received no packet from the preferred parent then ▷ pacote de controle ou dado
19:          if  $\gamma > 0$  then
20:            send multicast DIS to neighbors
21:            restart  $T_{reachable}$  timer
22: procedure INVALIDATE_UNREACHABLE_NEIGHBORS
23:   for each neighbor entry in neighbor table do
24:     if neighbor last update time +  $T_{reachable} \leq$  current time then
25:       invalidate neighbor entry

```

atualizações de nós pai decorrentes do aumento no envio de mensagens DIO.

As mensagens DAO são utilizadas pelos nós após o reingresso na topologia, seguindo a especificação canônica do RPL (RFC 6550, 2012). No MARPL, os pacotes DIS e DAO carregam a métrica γ do nó originador do pacote. Dessa maneira, o nó que receber tais mensagens pode realizar o ajuste do *Trickle* caso necessário. O ajuste do *Trickle* no MARPL segue uma abordagem inspirada por Kharrufa et al. (2015) e Corbazan et al. (2014). O ajuste do *Trickle* é descrito pelo Algoritmo 4.3. O primeiro passo para o ajuste do *Trickle* é atribuir o valor de γ do pacote recebido para a variável γ_{packet} (linha 2). Se $\gamma_{packet} > \gamma$ (ou seja, se o nó filho estiver com movimentação maior que o do nó pai), então divide-se a variável I do *Trickle* pela metade (linha 4). Vale lembrar que a variável I no *Trickle* representa o tamanho do atual intervalo de supressão de mensagens DIO. Para assegurar que o ajuste no *Trickle* esteja de acordo com as regras determinadas na proposta original do *Trickle*, o MARPL verifica se o valor de I não é menor do que I_{min} (linha 5). Sendo I_{min} o período mínimo em que a supressão deve acontecer.

O objetivo deste ajuste no temporizador *Trickle* é permitir que a transmissão

Algoritmo 4.3 Trickle Adjustment

```

1: procedure TRICKLE_ADJUSTMENT
2:    $\gamma_{packet} \leftarrow \gamma$  from control packet ▷ DIO, DIS or DAO
3:   if  $\gamma_{packet} > \gamma$  then
4:      $I \leftarrow I/2$ 
5:     if  $I < I_{min}$  then
6:        $I \leftarrow I_{min}$ 

```

de mensagens DIO seja temporariamente maior para que o *Listen Only Period* (LOP) seja reduzido. Esta medida aumenta o desempenho do MARPL em termos de taxa de entrega de pacotes através da prevenção de desconexões e de manutenções na topologia.

No recebimento de pacotes, o MARPL também verifica se o pacote é de controle do tipo DIO. Caso positivo, o *rank* do vizinho originador do pacote é atualizado (linha 9). A atualização do valor de *rank* é feita de modo a incluir a métrica Variabilidade de Vizinhaça (γ) no processo de atualização do nó pai.

Através da Equação 4.3, define-se que quanto maior for a métrica γ , maior será o *rank* do nó candidato. Nós candidatos com o *rank* mais alto têm menor chance de serem selecionados como nó pai por outros nós. Através da métrica γ o MARPL favorece os nós que se movimentam menos para serem selecionados como nó pai. O MARPL também analisa se o pacote recebido veio do nó pai. Em caso positivo (linha 10), o temporizador $T_{reachable}$ é reiniciado (linha 11). O temporizador $T_{reachable}$ é responsável pela detecção de desconexão entre um nó e seu nó pai.

$$vizinho_{rank_novo} = vizinho_{rank_original} + \gamma \quad (4.3)$$

No MARPL, o temporizador $T_{monitoring}$, utilizado para periodicamente recalculiar a métrica γ , executa por tempo suficiente para que todos os nós em sua vizinhaça transmitam ao menos um pacote. Após $T_{monitoring}$ expirar (linha 12), MARPL calcula a variância das diferenças dos RSSIs (Δ_p) para todos os nós vizinhos que tenham duas medidas consecutivas de RSSI registradas na tabela de vizinhaça. Na Equação 4.4, Δ_p é calculada pelo módulo da diferença das duas últimas medidas (t e $t - 1$) de RSSI de cada vizinho (i).

No Algoritmo 4.2, na linha 13, o valor do limiar K_γ pode ser atualizado se a variância das diferenças de RSSI (Δ_{pi}) para os vizinhos com duas medidas de RSSI consecutivas (representados pelo conjunto y) é a maior já calculada pelo nó. K_γ serve para normalizar a variância dos Δ_p de cada vizinho. K_γ inicialmente possui o valor 0 e é atualizado a cada cálculo da métrica γ , se $var\{\Delta_{p_i}\}_{i=1}^y > K_\gamma$. Conforme a Equação 4.5.

A métrica γ é atualizada através da Equação 4.6, linha 15 no Algoritmo 4.2. Divide-se a variância das diferenças de RSSI (Δp_i) pelo limiar K_γ . O cálculo de γ depende de três condições: (i) o número de vizinhos com duas medidas consecutivas de RSSI deve ser maior do que 0, ou seja $y > 0$. Caso $y = 0$, o valor de γ também será 0, pois não há diferenças de potência de sinal para analisar; (ii) as diferenças de RSSI, Δp_i , para cada vizinho i , tal que $i \in y$, devem ser diferentes de 0. Isto indica que o nó vizinho pode estar se movimentando; e (iii) K_γ deve ser maior do que 0.

$$\Delta p_i = | p_{i,t-1} - p_{i,t} | \quad (4.4)$$

$$K_\gamma = \max\{\text{var}\{\Delta p_i\}_{i=1}^y\} \quad (4.5)$$

$$\gamma = \frac{\text{var}\{\Delta p_i\}_{i=1}^y}{K_\gamma} \mid y > 0, \Delta p_i \neq 0, K_\gamma > 0 \quad (4.6)$$

O mecanismo de detecção da inalcançabilidade do nó pai no MARPL funciona através do temporizador $T_{reachable}$. Conforme o Algoritmo 4.2 na linha 17, quando $T_{reachable}$ expira, o MARPL analisa se foi recebido algum pacote de controle ou dado do nó pai (linha 18) e se γ do nó é maior do que 0 (linha 19). Em caso positivo, o MARPL considera que houve uma desconexão entre os nós e a topologia precisa ser localmente atualizada. Para tal, o nó envia uma mensagem DIS em *multicast* para todos os vizinhos. Segundo a especificação canônica do RPL, o recebimento de uma mensagem DIS em *multicast* pode causar o reinício do temporizador *Trickle* (RFC 6550, 2012), aumentando o envio de mensagens DIO a fim de atualizar a topologia.

A métrica γ de um nó é transmitida a seus vizinhos através das mensagens de controle DIO para que a mesma possa ser utilizada no processo de escolha de um nó pai no reingresso do nó na topologia. Na Equação 4.6 é possível que γ seja um valor fracionário dentro do conjunto fechado $[0, 1]$. A transmissão de valores fracionários é custosa computacionalmente. Para resolver este problema no MARPL, multiplica-se o valor γ por 100, e realiza-se uma conversão do valor real para um valor natural para obter um valor entre 0 e 100 representando a porcentagem de variabilidade na vizinhança do nó. A finalidade dessa conversão, e estabelecimento do limite é para que γ possa ser representada por 1 *Byte* de informação. Visa-se reduzir o custo na transmissão da métrica entre os nós em comparação com o valor representado em ponto flutuante. Também, busca-se manter a integridade com o RPL canônico, na qual o cálculo do *rank* dos nós por um função objetivo é feito com valores naturais.

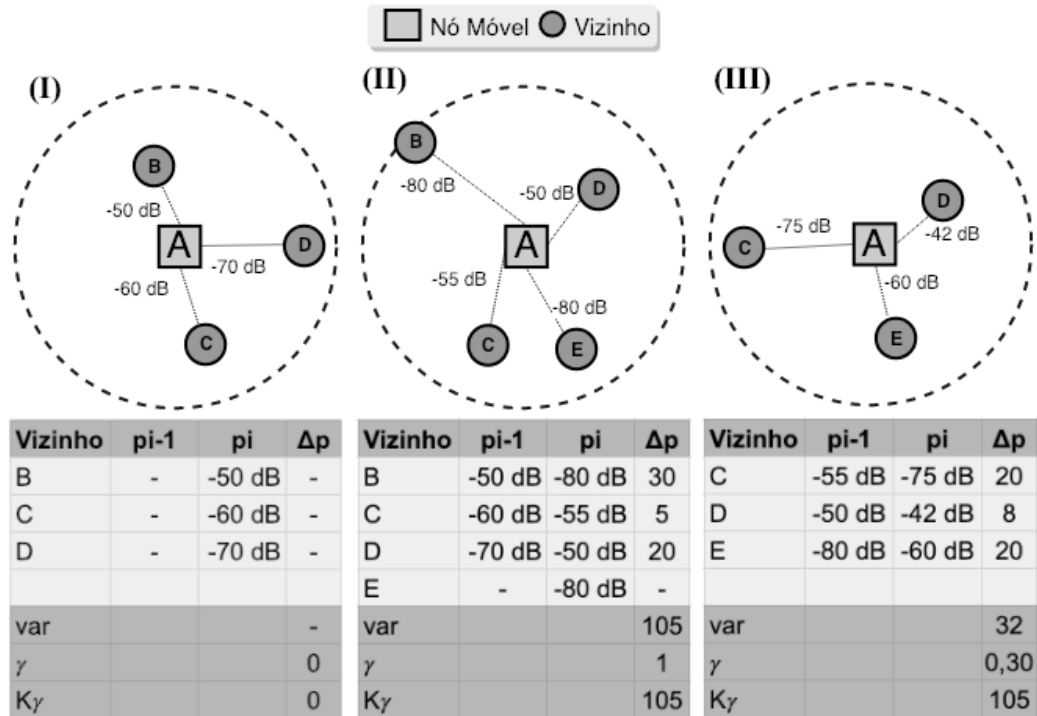
No MARPL, transmite-se a métrica γ através das mensagens de controle do RPL DIS e DAO. Segundo a especificação canônica do RPL, uma mensagem DAO é

sempre transmitida quando um nó ingressa em uma DODAG para informar seu nó pai. Como dito anteriormente, mensagens DIS são utilizadas pelo MARPL no mecanismo de detecção da desconexão com o nó pai. A motivação para a inclusão de γ nas mensagens de controle DIS e DAO é possibilitar ao nó pai ajustar o temporizador *Trickle* assim que receber uma mensagem DIS ou DAO de um nó filho em movimentação, porém que esteja na sua área de cobertura.

Em suma, o envio da métrica γ no MARPL é motivado pelas seguintes razões:

- O pacote DIO é utilizado pelo nó para divulgar sua DODAG para que seus vizinhos possam escolhê-lo como nó pai. Para a inclusão de γ no pacote DIO, utiliza-se o contêiner de métricas da DAG. Este contêiner pode ser transmitido pelos pacotes DIO ou DAO no RPL canônico para divulgar métricas que estão dentro da DODAG (RFC 6550, 2012). O MARPL também utiliza este contêiner nas mensagens DIS. Assim, todas as mensagens de controle do MARPL contém este contêiner, que pode incluir as métricas de distância em saltos e ETX, assim como a Variabilidade de Vizinhança (γ) do nó. A inclusão da métrica γ no contêiner de métricas da DODAG transmitido pelos pacotes DIO, permite aos nós vizinhos conhecer a Variabilidade de Vizinhança do nó candidato durante o processo de escolha do próximo nó pai. Por meio deste mecanismo, os candidatos com menor γ (ou seja, menor mobilidade) têm maior probabilidade de serem selecionados como nó pai.
- O pacote DIS na especificação canônica do RPL é transmitido por um nó que está fora de uma DODAG, com a finalidade de receber mensagens DIO de seus vizinhos e poder eleger um novo nó pai (RFC 6550, 2012). A especificação canônica também propõe que as mensagens DIS sejam utilizadas por nós que necessitem obter informações atualizadas sobre a DODAG na qual estejam conectados. Propõe-se no MARPL, o uso das mensagens DIS para que os nós móveis possam iniciar um processo de atualização da topologia quando identificada uma desconexão com o nó pai. Outra motivação para a inclusão de γ nos pacotes DIS, é que através da métrica γ , contida no contêiner de métricas da DODAG nas mensagens DIS, um nó pai pode ajustar o temporizador *Trickle*. O objetivo desta estratégia é reduzir e auxiliar na rápida resolução das desconexões entre os nós. Conseqüentemente, aumentar a taxa de sucesso de entrega de pacotes.
- No MARPL, o pacote DAO também contém o valor γ do nó. Adotou-se este procedimento para que um nó pai possa ajustar o temporizador *Trickle* assim que identificar a mobilidade do nó filho.

Figura 4.7 – Topologia LLN com Movimentação do Nó A.



Fonte: O autor.

Na Figura 4.7, apresenta-se um exemplo de cenário de uma topologia LLN com o nó **A** em movimento. São exemplificadas na figura três execuções do temporizador $T_{monitoring}$ pelo gerenciamento de mobilidade do nó móvel **A**. A potência de sinal (RSSI), medida em decibéis, entre o nó **A** e seus vizinhos é representada pelas linhas tracejadas. A tabela de vizinhança do nó móvel **A** é apresentada para cada período de tempo da Figura 4.7. O nó móvel é identificado como um quadrado e os nós vizinhos são identificados como círculos.

Durante o período representado na Figura 4.7.I o nó **A** identifica a presença de três novos vizinhos em sua vizinhança com os respectivos valores das potências de sinais medidos em decibéis, apresentados na representação da tabela de vizinhança do nó **A** abaixo da topologia na Figura 4.7.I. Os valores de RSSI da tabela são para exemplificação, porém, baseados em valores aproximados dos valores medidos em simulação, conforme experimentos apresentados no Capítulo 5. Como o período de monitoramento da vizinhança da Figura 4.7.I é o primeiro a ser executado pelo nó **A**, o valor de γ é 0, pois ainda não existem nós com duas medições consecutivas de RSSI em sua tabela de vizinhança (ou seja, $y = 0$).

No segundo período exemplificado na Figura 4.7.II, o nó **A** identifica a sua movimentação pela variância das diferenças dos RSSIs medidos no recebimento de

mensagens dos três vizinhos conhecidos, como pode ser visto na tabela de vizinhança do nó **A** naquele período. Neste período, o valor de γ , com base na Equação 4.6, é 1 pois o valor de K_γ passa a ser o valor da variância (*var*), que neste exemplo é 105. A métrica γ será convertida para 100 quando for transmitido pelos pacotes de controle, representando que a variabilidade da vizinhança do nó é de 100%.

No período descrito pela Figura 4.7.III, o nó móvel **A** identifica diferenças de RSSI nos três vizinhos (**C**, **D** e **E**). Portanto, todas as diferenças de potência de sinal serão utilizadas no cálculo de γ . A variância neste período é de 32. Como $32 < 105$, o valor de K_γ não é atualizado e γ recebe o valor de 0,30, segundo a Equação 4.6. O valor de γ indica que a vizinhança do nó **A** começa a se estabilizar no período da Figura 4.7.III. É considerado no MARPL que o nó **A** está mais apto para ser escolhido como nó pai no período da Figura 4.7.III, se comparado com o período da Figura 4.7.II. A justificativa dessa conclusão é dada pelo fato do valor de γ na Figura 4.7.III ser menor do que na Figura 4.7.II.

4.2 FUNÇÃO OBJETIVO DO MARPL E ESCOLHA DO NÓ PAI

No RPL, as métricas utilizadas pela função objetivo determinam quais critérios o nó candidato deve atender para ser selecionado como nó pai. A métrica Variabilidade de Vizinhança (γ) é utilizada na função objetivo do MARPL com o propósito de estipular que nós candidatos com menor mobilidade tenham maior probabilidade de serem escolhidos como nó pai. Dessa maneira, a escolha leva em conta a mobilidade do nó. Em outras palavras, os nós com menores valores de γ terão maior probabilidade de serem escolhidos como nó pai.

A escolha de nós pais no MARPL é realizada pelo *rank* do candidato, calculado pela Equação 4.3. No processo de seleção do nó pai no MARPL, assim como no RPL canônico, utiliza-se a métrica *Expected Transmission Counter* (ETX) e a métrica γ indicando a movimentação do nó candidato. A métrica ETX, utilizada no cálculo do $vizinho_{rank_original}$ na Equação 4.3, permite a escolha de rotas com maiores chances de sucesso em relação à taxa de sucesso de entrega nos pacotes. Por outro lado, a métrica γ possibilita a escolha de rotas com menor probabilidade de desconexão. No MARPL, a Equação 4.3 é utilizada pela função objetivo MRHOF (RFC 6719, 2012). A combinação das métricas ETX, utilizada para o cálculo do *rank* no RPL canônico, e γ permite que os nós possam escolher nós pais com rotas mais eficientes (característica da métrica ETX) e com menor probabilidade de desconexão (determinada pela métrica γ), dado que o nó escolhido apresenta uma mobilidade menor.

4.3 CONCLUSÕES DO CAPÍTULO

Foi proposta neste capítulo uma abordagem para o gerenciamento da mobilidade no RPL chamada MARPL. A abordagem é composta de três mecanismos: (i) monitoramento da vizinhança do nó e cálculo da métrica Variabilidade de Vizinhança (γ) que é utilizada no processo de seleção de rotas; (ii) identificação de inalcançabilidade do nó pai quando o nó não receber mensagens do nó pai por um determinado período de tempo; (iii) ajuste dos procedimentos de envio de mensagens de controle responsáveis pela construção e manutenção da topologia RPL.

O mecanismo de gerenciamento de mobilidade propõe dois novos temporizadores chamados de $T_{monitoring}$ e $T_{reachable}$. $T_{monitoring}$ é utilizado para o monitoramento da vizinhança e cálculo da métrica Variabilidade de Vizinhança (γ). O MARPL utiliza o $T_{reachable}$ para a detecção da inalcançabilidade do nó pai.

Durante o funcionamento do temporizador $T_{monitoring}$, um nó monitora o recebimento de mensagens de outros nós em sua vizinhança para que o mesmo possa calcular o valor da métrica γ e verificar sua mobilidade. γ é uma métrica derivada das diferenças de potência de sinal medidas pelo nó no recebimento de mensagens dos nós vizinhos. O temporizador $T_{monitoring}$ é regulado de acordo com o *Data Packet Period* (DPP) dos nós, ou seja, a taxa de produção de dados dos nós sensores. O valor do DPP é dependente da aplicação. O temporizador $T_{reachable}$ é iniciado quando o nó ingressa na DODAG e funciona por um determinado tempo, porém suficiente para que o nó pai tenha transmitido ao menos uma mensagem de dado. O temporizador $T_{reachable}$ determina o período de tempo em que um nó deve aguardar por uma mensagem de dado ou controle vindo do nó pai. Na ocorrência de desconexão do nó com o nó pai, o nó pode iniciar o procedimento de reingresso na DODAG e manutenção da topologia RPL através do envio de mensagens DIS em *multicast* seguindo a especificação canônica do RPL.

Propõe-se no MARPL que a métrica Variabilidade de Vizinhança (γ) de um nó seja transmitida ao seu nó pai através das mensagens DIS e DAO. Através dessa métrica, o nó pai pode ajustar a periodicidade na transmissão de mensagens DIO regulada pelo temporizador *Trickle*. Caso o valor de γ de um nó filho seja maior do que o γ do nó pai, considera-se que há uma possibilidade do nó filho estar se movimentando para fora da área de cobertura do nó pai. Desta maneira, o nó pai reduz o período de supressão de transmissão de mensagens DIO pela metade para que o *Listen Only Period* (LOP) seja reduzido. Com base nos resultados apresentados no Capítulo 5.1, comprova-se que esta medida possibilitou o aumento na transmissão de mensagens, sem comprometer negativamente na sobrecarga da rede.

No Capítulo 5, apresenta-se uma análise de desempenho do MARPL em ter-

mos da taxa de sucesso de entrega, taxa de desconexões, atraso de reingresso na topologia RPL, atraso de entrega, sobrecarga e consumo energético. A análise de desempenho foi realizada com o uso do simulador Cooja (OSTERLIND et al., 2006), um simulador largamente utilizado para analisar protocolos para LLNs. Comparou-se os resultados obtidos com a implementação canônica do RPL (RFC 6550, 2012) e com os trabalhos relacionados de Kharrufa et al. (2017) e Fotouhi et al. (2015).

5 EXPERIMENTOS E RESULTADOS

São apresentados neste capítulo os experimentos computacionais realizados com o simulador Cooja (OSTERLIND et al., 2006), e as análises dos resultados comparativos obtidos através de simulações de LLNs executando a abordagem proposta nesta dissertação de mestrado, *Mobility Aware RPL* (MARPL). Comparou-se o MARPL com o RPL canônico, mRPL de Fotouhi et al. (2015) e D-RPL de Kharrufa et al. (2017). O D-RPL foi escolhido por ser reativo e por utilizar-se de um Temporizador Reverso para o ajuste do *Trickle*, assim como o MARPL. O mRPL foi escolhido como objeto de comparação, porque segue uma abordagem proativa para o gerenciamento de mobilidade.

O simulador Cooja (OSTERLIND et al., 2006) foi utilizado nos experimentos, pois é uma ferramenta de simulação pertencente ao sistema operacional Contiki (DUNKELS et al., 2011). O sistema operacional Contiki foi desenvolvido para simular microcontroladores de baixa capacidade como os utilizados em LLNs. O Contiki OS oferece suporte nativo ao protocolo de roteamento RPL.

Na implementação canônica do RPL no sistema *Contiki*, os períodos mínimo e máximo de supressão na transmissão das mensagens de controle DIO, gerenciados pelo temporizador *Trickle*, são determinados por dois parâmetros I_{min} e I_{max} respectivamente. Os períodos de tempo mínimo e máximo de supressão do *Trickle* são calculados a partir destes parâmetros como expoentes de base 2 (DUNKELS et al., 2011).

No *Trickle* implementado pelo sistema Contiki, utiliza-se canonicamente o valor 12 para o parâmetro I_{min} e 8 para o I_{max} . Deste modo, na implementação canônica do RPL no Contiki OS, o período mínimo de tempo de espera para o envio de mensagens DIO é de 2^{12} , ou seja 4.096 milissegundos. O valor máximo é igual a 2^{12+8} , ou seja 1.048.576 milissegundos. Vale lembrar que o período de supressão definido pelo *Trickle* é regulado pela variável I , cujo valor está contido entre I_{min} e I_{max} . É proposto neste trabalho, que o valor de I seja reduzido pela metade quando um nó pai identificar um nó filho em movimentação na sua vizinhança. Este ajuste foi categorizado neste trabalho como Temporizador Reverso. Outras propostas que se encaixam nessa categoria podem ser encontradas em Kharrufa et al. (2017) e Cobarzan et al. (2014). A partir deste ajuste, objetiva-se reduzir o *Listen Only Period* (LOP) através do aumento na transmissão de mensagens de controle.

Na Tabela 5.4, apresenta-se os valores utilizados nos parâmetros das simulações. Os parâmetros foram parcialmente baseados no trabalho de Kharrufa et al.

(2017), porém, com algumas modificações com a finalidade de tornar os cenários mais parecidos com os ambientes de aplicações reais. Assim como em Kharrufa et al. (2017), o alcance da transmissão é de 50 metros. A velocidade máxima dos nós variam de 1,0 m/s, 2,5 m/s e 5,0 m/s. Os valores de I_{min} e I_{max} foram ajustados conforme o padrão da implementação do RPL no Contiki OS, 12 e 8 respectivamente.

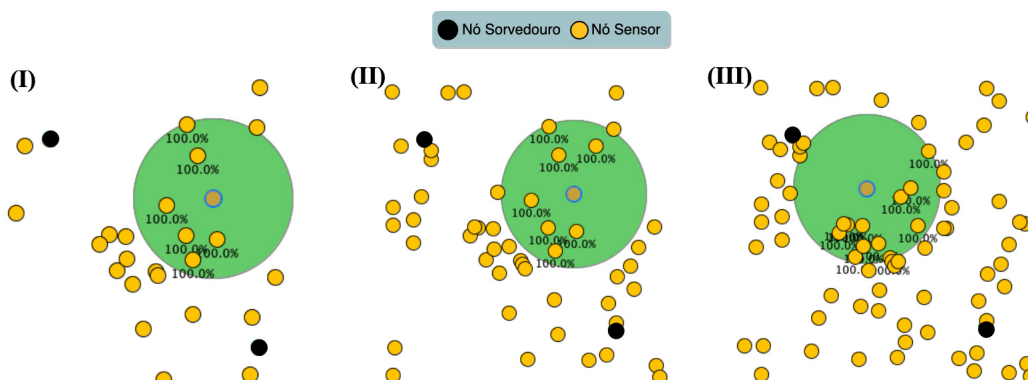
Tabela 5.4 – Parâmetros de Simulação

Parâmetros	Valor
Número de nós sensores	25, 50 e 75 e 100
Número de nós sorvedouros	2, 4, 6, 8 e 20
Tipo do dispositivo simulado	Zolertia Z1
Dispositivo de rádio simulado	CC2420 (INSTRUMENTS, 2006)
Tempo de simulação	10 minutos
Distribuição dos nós	Aleatória
Modelo de mobilidade	Steady-State Random Waypoint
Velocidade máxima dos nós	1,0 m/s, 2,5 m/s e 5,0 m/s
Pausa máxima	40 segundos
Meio de transmissão	<i>Unit Disk Graph Medium (UDGM)</i>
Alcance da transmissão do rádio	50m
Área da simulação	200m x 200m e 1000m x 1000m
DPP	3 segundos
θ	1, 2 e 3
Intervalo de Confiança	95%

A taxa de geração de dados, do inglês *Data Packet Period* (DPP), foi ajustada para 3 segundos. O DPP varia de acordo com as características da aplicação. No escopo deste trabalho de mestrado, utilizou-se das características de um monitoramento urbano (GUBBI et al., 2013). Os experimentos de Fotouhi et al. (2015) foram realizados numa área de 8m x 10m, e em cenários com apenas 1 nó móvel. Por outro lado, Kharrufa et al. (2017) realizaram experimentos numa área de 150m x 150m com todos os nós sensores podendo se movimentar. Visando avaliar os protocolos MARPL, RPL (RFC 6550, 2012), D-RPL (KHARRUFA et al., 2017) e mRPL (FOTOUHI; MOREIRA; ALVES, 2015), um número de 25, 50 e 75 nós sensores foram distribuídos nas simulações com uma área de 200m x 200m. Os protocolos também foram avaliados em cenários, onde variou-se o número de nós sorvedouros e a velocidade de deslocamento dos nós. Com o objetivo de avaliar a escalabilidade do MARPL, um número de 100 nós foram distribuídos em uma área de 1000m x 1000m. Na Figura 5.8, apresentam-se exemplos das três topologias configuradas nas simulações realizadas no Cooja. Na Figura 5.8.I, 25 nós são móveis e nas Figuras 5.8.II e 5.8.III, 50 e 75 nós são móveis respectivamente.

Nativamente, o simulador Cooja não oferece suporte à mobilidade. Para corrigir esta limitação do simulador, utilizou-se o *plugin Mobility COOJA plugin* (OSTER-

Figura 5.8 – Exemplos de Topologias de LLN Utilizadas nas Simulações com 2 Sorvedouros.



Fonte: O autor.

LIND, 2010) que permite a adição de modelos de mobilidade no Cooja. Modelos de mobilidade descrevem a movimentação de cada nó em uma topologia.

Nos experimentos, utilizou-se o modelo de mobilidade *Steady-State Random Waypoint*. Este modelo é derivado do modelo *Random Waypoint* utilizado por Kharrufa et al. (2017). O modelo de mobilidade *Random Waypoint* é caracterizado pela movimentação aleatória dos nós e seus parâmetros são: a velocidade média dos nós, a área da simulação e o tempo médio de pausa. O *Steady-State Random Waypoint* utiliza o *Random Waypoint* para a criação dos padrões de movimentação dos nós. Além disso, o *Steady-State Random Waypoint* permite a inclusão de um período de tempo no qual um nó em movimentação pode permanecer estático (ASCHEBRUCK et al., 2010). Ressalta-se que a capacidade dos nós permanecerem estáticos por algum período de tempo é uma premissa para o funcionamento do MARPL. A pausa máxima para os nós foi ajustada em 40 segundos, pois dessa maneira a topologia se modifica constantemente. Os arquivos com os padrões de movimentação foram gerados pela ferramenta Bonnmotion (ASCHEBRUCK et al., 2010).

Os cenários configurados, com áreas de 200m x 200m e 1000m x 1000m, são compostos de dois tipos de nós: os nós sorvedouros, ou seja os nós raízes da DODAG, que possuem a responsabilidade de receber os dados coletados pelos nós sensores, e os nós sensores que transmitem as informações salto-a-salto a um nó sorvedouro. Este tipo de cenário é comum em redes IoT de sensoriamento.

Todos os nós sensores coletam o mesmo número de dados durante a execução de cada simulação. Dessa maneira, busca-se que todos os protocolos sejam testados nas mesmas condições de rede. Um conjunto de 20 simulações foram executadas com os protocolos RPL canônico, MARPL, D-RPL proposto por Kharrufa et

al. (2017) e mRPL proposto por Fotouhi et al. (2015).

A análise de desempenho dos protocolos foi realizada com base nos arquivos de *log* gerados nas simulações e em termos das seguintes métricas:

- Taxa de sucesso de entrega, do inglês *Packet Delivery Rate* (PDR): a taxa de pacotes de dados entregues com sucesso para os sorvedouros;
- Atraso de entrega: tempo necessário para os nós sorvedouros receberem os pacotes de dados;
- Atraso de reconexão: período de desconexão dos nós;
- Sobrecarga: a taxa de geração de pacotes de controle;
- Energia residual: a energia restante nas baterias dos dispositivos após a execução da rede.

Testes de significância estatística foram executados para assegurar que os resultados obtidos pelas análises apresentam uma diferença significativa entre si. Na primeira etapa dos testes, foram realizadas análises individuais nos resultados obtidos para cada protocolo, a fim de determinar se o conjunto de dados apresenta uma distribuição normal. Para tal, aplicou-se o teste *Shapiro-Wilk* (SHAPIRO; WILK, 1965). Nos conjuntos de resultados com uma distribuição normal, aplicou-se o teste de significância *Paired Student's t-Test* (STUDENT, 1908). Nos conjuntos de resultados com uma distribuição não normal, aplicou-se um teste não paramétrico *Wilcoxon Signed-Rank Test* (WILCOXON, 1945). Após a realização dos testes de significância estatística, pode-se concluir se as distribuições dos resultados de cada protocolo são distintas. Logo, se apresentam diferenças estatisticamente significantes.

Os resultados da análise em função do número de nós na rede são apresentados na Subsecção 5.1.

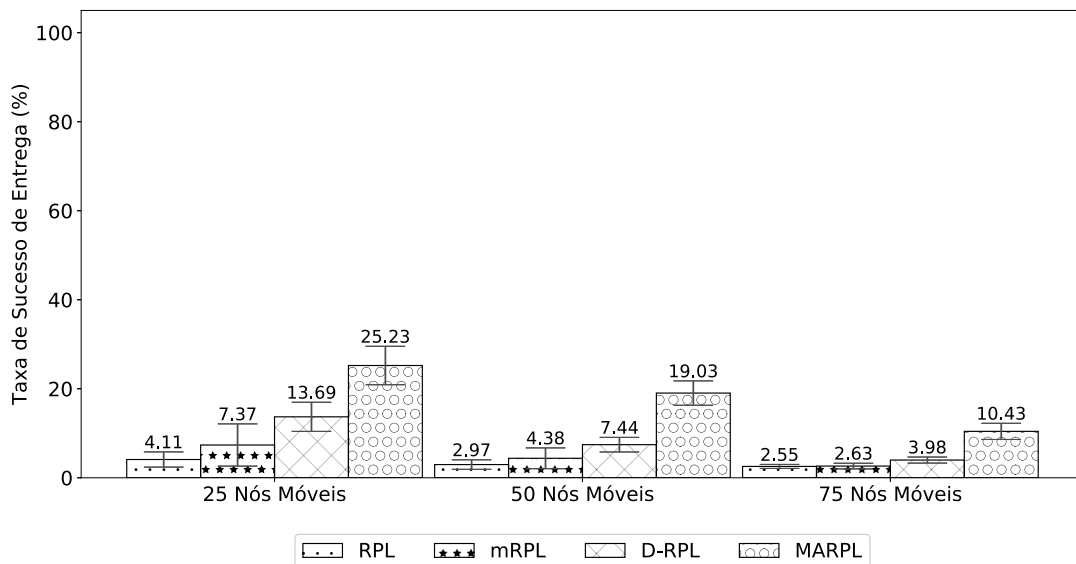
5.1 ANÁLISE DO MARPL EM FUNÇÃO DO NÚMERO DE NÓS

Nesta seção, descrevem-se as simulações para avaliar a taxa de sucesso de entrega (PDR), atraso, sobrecarga e energia residual do MARPL em relação aos protocolos, RPL, D-RPL e mRPL. O tamanho da área é de 200m x 200m, a velocidade é de 5,0 m/s e o número de nós móveis varia entre 25, 50 e 75. Dois sorvedouros foram distribuídos na área simulada e o valor de θ é 3.

Na Figura 5.9, descrevem-se os resultados da taxa de sucesso de entrega (PDR). Como pode-se observar, a medida que o número de nós na topologia aumenta, o PDR diminui para todos os protocolos avaliados. Conclui-se que a densidade da

rede está diretamente relacionado com o PDR resultante. Todavia, o MARPL obteve melhores resultados em todos os cenários avaliados. 25,23% na topologia com 25, 19,03% com 50 e 10,43% com 75 nós móveis. O D-RPL ficou com o segundo maior PDR nos experimentos realizados, 13,69%, 7,44% e 3,98%. O MARPL e o D-RPL utilizam uma abordagem de temporizador reverso para o ajuste no *Trickle*. Porém, o ajuste no MARPL é realizado apenas pelo nó pai quando este identifica um nó filho em movimentação na sua vizinhança. Portanto, o ajuste é realizado apenas quando a topologia necessita ser atualizada. Através de análises estatísticas, averiguou-se a relevância estatística de cada resultado. Os protocolos RPL e mRPL obtiveram os menores resultados de PDR. Segundo estas análises estatísticas, os resultados dos protocolos RPL e mRPL não apresentaram diferenças significativas em relação ao PDR nos cenários com 50 e 75 nós móveis.

Figura 5.9 – Análise da Taxa de Sucesso de Entrega (2 Sorvedouros).

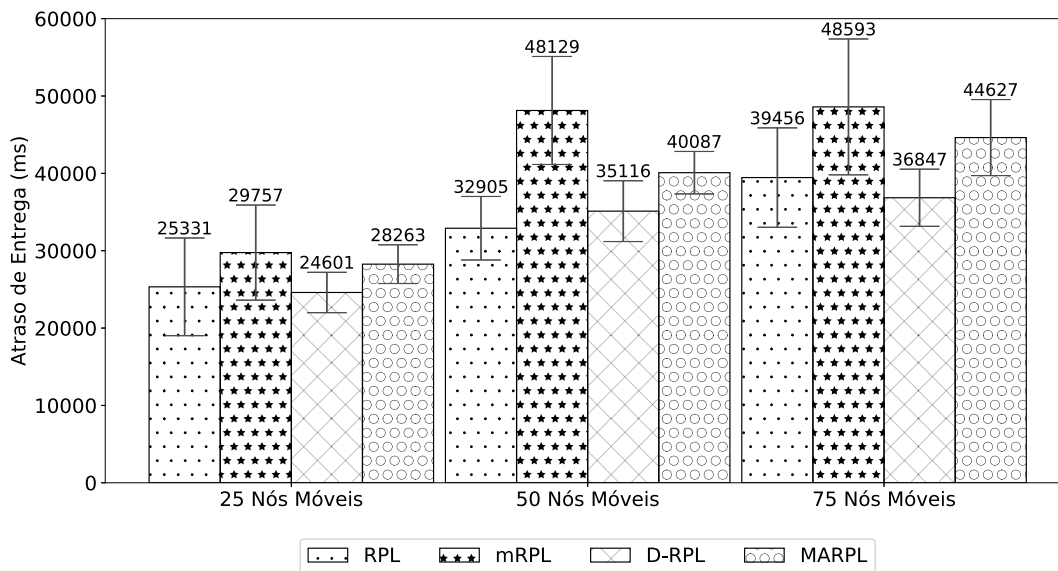


Fonte: O autor.

Os resultados obtidos em termos de atraso de entrega de pacotes, em milissegundos, são apresentados na Figura 5.10. Os protocolos RPL e D-RPL obtiveram, em média, atrasos menores em todos os cenários avaliados. O RPL com 25331, 32905 e 39456 milissegundos nos cenários com 25, 50 e 75 nós móveis respectivamente. O D-RPL com 24601, 35116 e 36847 milissegundos. O mRPL apresentou os maiores atrasos em média. O único resultado que apresentou uma diferença estatística foi entre o MARPL e o mRPL no cenário com 50 nós móveis. O mRPL monitora constantemente uma conexão entre um nó filho e seu nó pai. Contudo, o mRPL não possui um mecanismo de rápida atualização da DODAG, isso justifica tanto o alto atraso de

entrega, quanto a baixa taxa de sucesso apresentada na Figura 5.9. Conclui-se com base nos testes de significância estatística, que os protocolos RPL, mRPL, D-RPL e MARPL apresentaram desempenho equivalente em relação ao atraso.

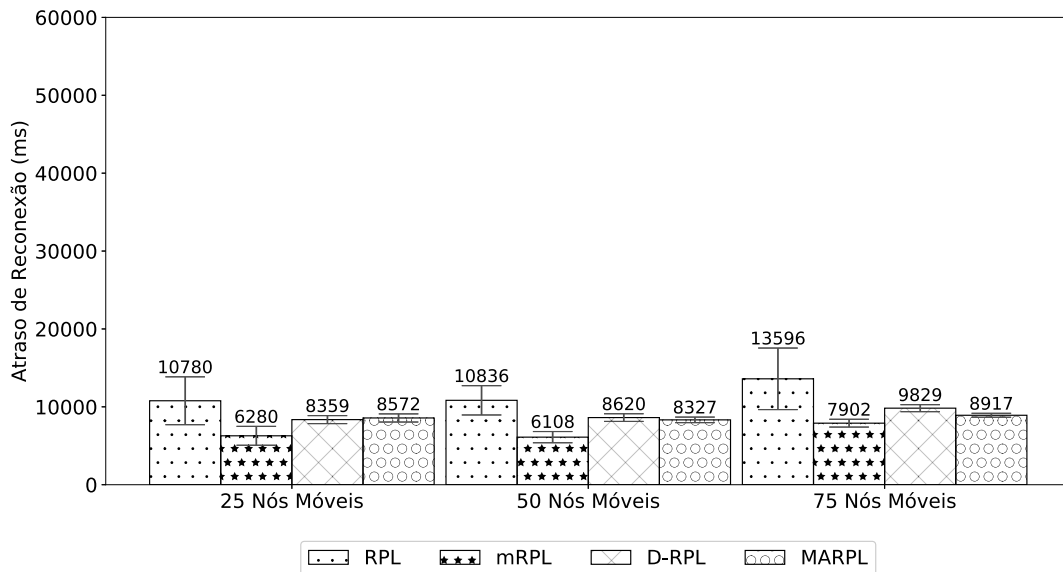
Figura 5.10 – Análise do Atraso de Entrega de Pacotes (2 Sorvedouros).



Fonte: O autor.

Os resultados do atraso de reconexão com o nó pai são apresentados na Figura 5.11. O protocolo RPL apresentou os maiores atrasos de reconexão (10780, 10836 e 13596 milissegundos), explicitando a falta de um mecanismo de rápida identificação de desconexão. O menor atraso de reconexão foi obtido pelo mRPL (6280, 6108 e 7902 milissegundos), tendo em vista que o foco principal deste protocolo é a rápida detecção de desconexão através da inclusão de novos temporizadores. Entretanto, conforme pode ser observado na Figura 5.9, o mRPL alcançou um baixo percentual na taxa de sucesso de entrega (PDR), quando comparado com o MARPL. O MARPL e D-RPL apresentaram resultados muito semelhantes nos cenários com 25 e 50 nós móveis. No cenário com 25 nós móveis, o MARPL obteve 8572 e o D-RPL, 8359 milissegundos em média. No cenário com 50 nós móveis, o MARPL obteve 8327 e o D-RPL, 8620 milissegundos. Somente no cenário com 75 nós móveis, o MARPL (com 8917 milissegundos) apresentou resultados estatisticamente significante, quando comparado com o D-RPL (com 9829 milissegundos). O atraso de reconexão dos protocolos RPL, mRPL, D-RPL e MARPL aumentam a medida que o número dos nós aumenta. Este comportamento é justificado pelo fato que o aumento no número de nós móveis pode ocasionar um aumento no número de saltos que os pacotes de dados necessitam trafegar.

Figura 5.11 – Análise do Atraso de Reconexão de Nó Pai (2 Sorvedouros).



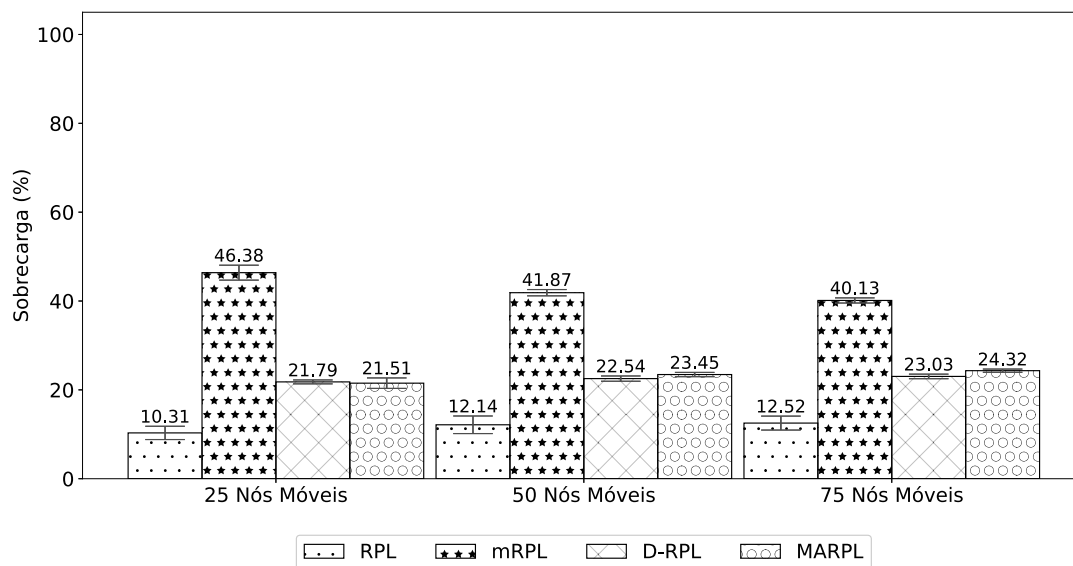
Fonte: O autor.

Na Figura 5.12, apresenta-se a sobrecarga gerada pelos protocolos analisados. O mRPL apresentou a maior sobrecarga entre todos os protocolos avaliados, 46,38%, 41,87% e 40,13% respectivamente. A alta taxa de sobrecarga é justificada pelas constantes trocas de mensagens de controle realizada pelo mRPL para que os nós possam rapidamente identificar sua desconexão. Conclui-se que a abordagem proativa adotada pelo mRPL para rapidamente detectar uma desconexão reduz o atraso de reconexão, porém, degrada o desempenho geral da rede. O RPL apresentou menor sobrecarga pela falta de mecanismos para gerenciar a mobilidade dos nós (10,31%, 12,14% e 12,52%). Tendo em vista que um mecanismo com essa finalidade faz uso de mensagens de controle para a detecção da mobilidade. Os resultados apresentados pelos protocolos MARPL e D-RPL foram similares em todos os cenários avaliados. No cenário com 25 nós móveis, não se obteve diferença estatística entre os dois protocolos. Nos cenários com 50 e 75 nós móveis, o MARPL apresentou a segunda maior sobrecarga em média, quando comparado aos protocolos RPL, D-RPL e mRPL, com 23,45% e 24,32% respectivamente. As sobrecargas obtidas pelo D-RPL nos cenários com 50 e 75 nós móveis foram similares a do MARPL, com 22,54% e 23,03% respectivamente

A energia residual média foi avaliada para os protocolos RPL, D-RPL, mRPL e MARPL. O resultado da energia residual é a média das porcentagens das baterias de cada nó após a execução da rede. Todos os nós possuem inicialmente 700 Joules.

Na Figura 5.13, descrevem-se os resultados referentes a energia residual dos

Figura 5.12 – Análise da Sobrecarga de Mensagens de Controle (2 Sorvedouros).



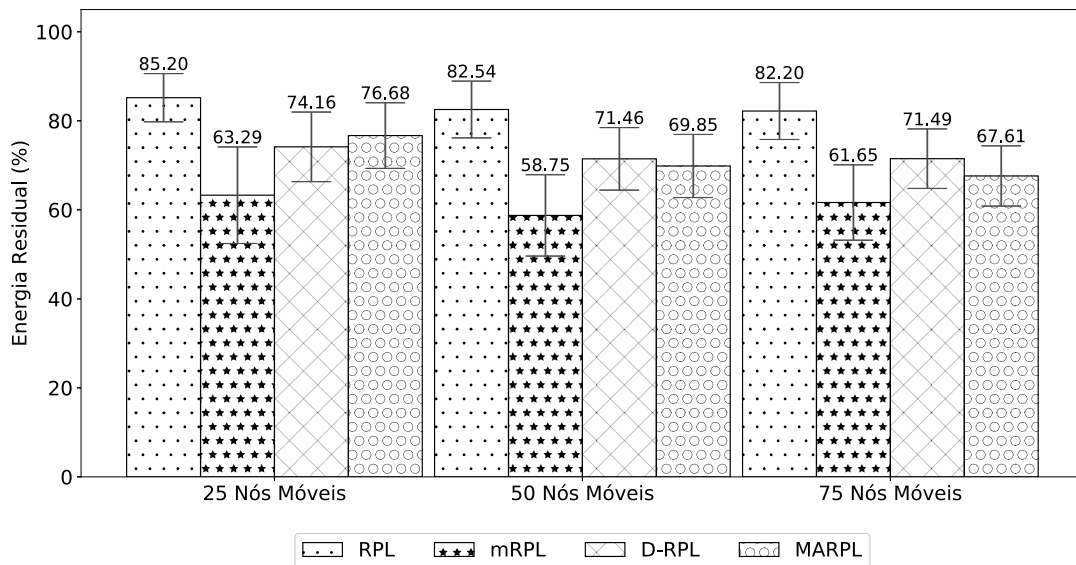
Fonte: O autor.

nós. O RPL foi o protocolo que mais conservou energia. Esse resultado é justificado por dois fatores: o RPL possui a menor taxa de sobrecarga entre os protocolos analisados, devido ao fato de gerar um número menor de mensagens de controle. O RPL apresentou os menores PDRs, indicando que menos pacotes de dados foram encaminhados com sucesso na rede. Portanto, dois fatores impactam a energia residual: a sobrecarga e o PDR. Quanto maior a sobrecarga, maior o número de pacotes de controle sendo trafegado na rede. Quanto maior o PDR, maior o número de pacotes de dados que foram transmitidos salto-a-salto na rede. Nos experimentos, definiu-se que todos os nós de cada simulação coletam a mesma quantidade de dados para serem enviados para os nós sorvedouros. Logo, a geração de dados foi a mesma para todos os protocolos em todas as simulações. A diferença está no número de pacotes de dados que trafegaram com sucesso através dos múltiplos saltos de acordo com as características de cada protocolo.

A taxa de sobrecarga gerada pelo mRPL provocou um aumento do consumo de energia. Observou-se que a energia residual do mRPL ficou abaixo dos outros protocolos, porque o mRPL transmite um número maior de mensagens de controle. Dessa maneira, entende-se que o mRPL apresentou o resultado menos satisfatório em termos de consumo energético. Dentre as propostas analisadas para o gerenciamento de mobilidade neste trabalho, o MARPL apresentou menor consumo energético no cenário com 25 nós móveis, mesmo apresentando uma sobrecarga similar ao do D-RPL e um alto PDR se comparado com os outros protocolos. Contudo, nos cenários

mais densos com 50 e 75 nós móveis, o MARPL apresentou uma energia residual menor do que a obtida pelo D-RPL. O PDR obtido pelo MARPL justifica o maior consumo energético, pois mais pacotes de dados foram trafegados com sucesso na rede. Desta maneira, os nós no MARPL passaram mais tempo com o rádio ativo do que os nós executando o D-RPL.

Figura 5.13 – Análise da Energia Residual (2 Sorvedouros).



Fonte: O autor.

Através dos experimentos apresentados, conclui-se que, em comparação com os outros trabalhos analisados, o MARPL apresenta a maior taxa de sucesso de entrega, com um atraso de entrega similar aos outros protocolos e com baixa sobrecarga. A abordagem para o gerenciamento de mobilidade utilizada pelo MARPL é considerada como o principal fator para estes resultados. Tal abordagem é composta de três mecanismos que tentam minimizar os impactos das desconexões causadas pela mobilidade através de atualizações da topologia sempre que a vizinhança de um nó apresenta altos índices de mobilidade. Em outras palavras, altos valores da métrica Variabilidade de Vizinhança. O MARPL também apresentou um dos menores atrasos de reconexão. Este resultado demonstra que o MARPL consegue atualizar rapidamente a topologia, mantendo um bom desempenho na rede.

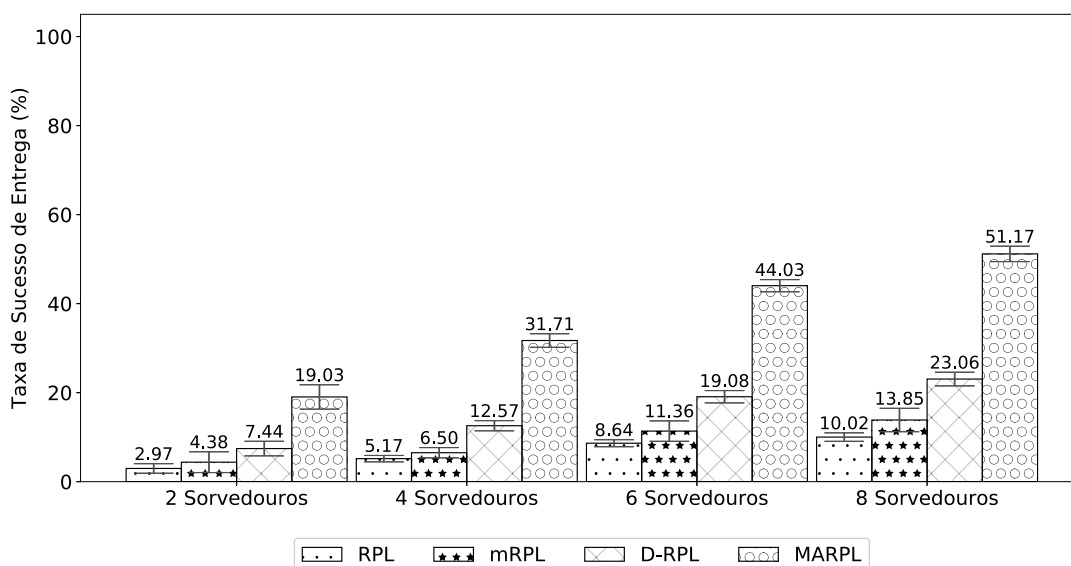
Na próxima subsecção apresenta-se uma análise de desempenho dos protocolos avaliados em relação à variação do número de sorvedouros.

5.2 ANÁLISE DO MARPL EM FUNÇÃO DO NÚMERO DE SORVEDOUROS

Nesta subsecção são apresentadas os resultados das análises no cenário com a área de 200m x 200m, 50 nós móveis, velocidade máxima de 5,0 m/s e com o número de nós sorvedouros variando entre 2, 4, 6 e 8. O valor de θ é 3. Nos experimentos, buscou-se avaliar o desempenho do MARPL diante de um número maior de sorvedouros.

Na Figura 5.14, apresentam-se os resultados da métrica taxa de sucesso de transmissão (PDR). O PDR para todos os protocolos avaliados aumenta à medida que o número de nós sorvedouros também aumenta. Observa-se que o aumento no número de nós sorvedouros implica um aumento no número de DODAGs, para os quais os nós podem transmitir seus pacotes de dados. O protocolo MARPL apresentou os maiores PDRs, 19,03%, 31,71%, 44,03% e 51,17% nas simulações com 2, 4, 6 e 8 sorvedouros respectivamente. Através da análise estatística realizada, verificou-se que os resultados obtidos pelo mRPL e RPL no cenário com 2 sorvedouros não foram estatisticamente diferentes. Porém, nos cenários com 4, 6 e 8 sorvedouros, o mRPL apresentou maior PDR do que o RPL. O protocolo RPL apresentou os menores PDRs. Este resultado é justificado pela ausência de mecanismos para o gerenciamento da mobilidade dos nós.

Figura 5.14 – Análise da Taxa de Sucesso de Transmissão Variando-se o Número de Sorvedouros (50 Nós).

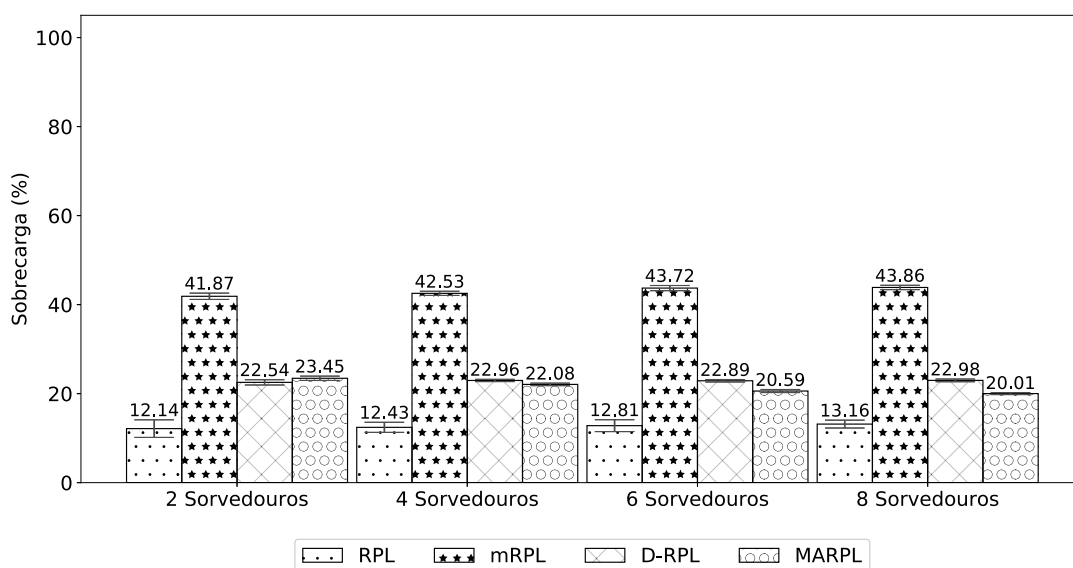


Fonte: O autor.

Na Figura 5.15, apresenta-se a sobrecarga gerada pelas mensagens de controle dos protocolos avaliados. O RPL apresentou a menor sobrecarga nos cenários

avaliados: 12,14%, 12,43%, 12,81% e 13,16% nos cenários com 2, 4, 6 e 8 nós sorvedouros. Este comportamento justifica-se pela ausência de mecanismos para o gerenciamento de mobilidade no RPL. Contudo, a análise estatística apontou que os resultados do protocolo RPL não apresentaram diferença estatística em relação ao aumento de sorvedouros.

Figura 5.15 – Análise da Sobrecarga de Mensagens de Controle Variando-se o Número de Sorvedouros (50 Nós).

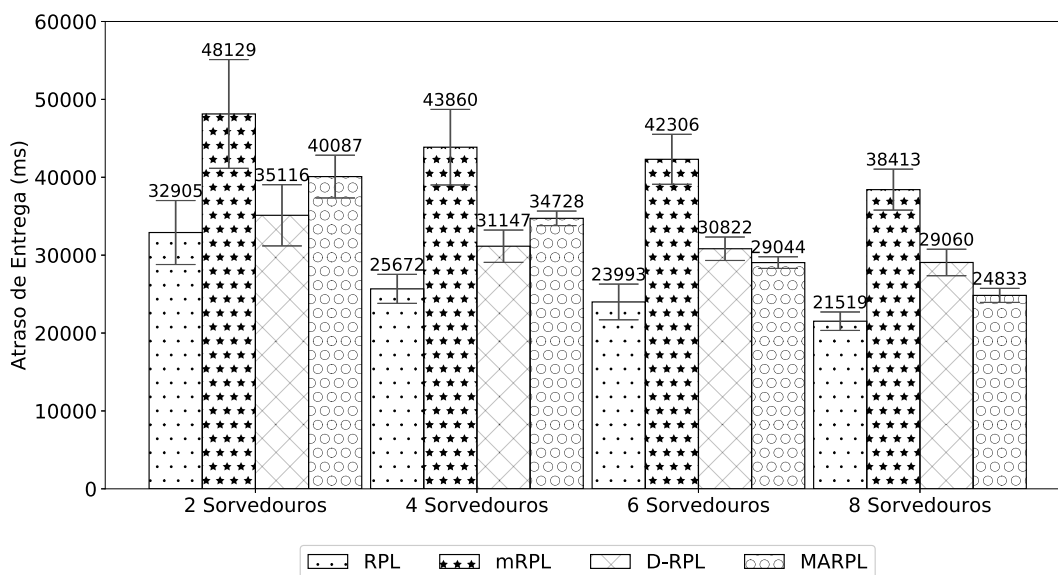


Fonte: O autor.

Avaliando-se somente os protocolos que propõem mecanismos para o gerenciamento de mobilidade, verificou-se que o D-RPL apresentou um comportamento semelhante ao RPL. Tanto no RPL quanto no D-RPL, a sobrecarga resultante não apresentou diferença estatística quando variou-se a quantidade de nós sorvedouros. O mRPL apresentou as maiores sobrecargas: 41,87%, 42,53%, 43,72% e 43,86%. Contudo, os resultados com 6 e 8 sorvedouros, 42,53% e 43,72%, não apresentaram diferença estatística. O resultado obtido pelo mRPL é justificado pela sua natureza proativa em lidar com a mobilidade e as desconexões, através de constantes trocas de mensagens entre os nós. O MARPL foi o único protocolo a apresentar uma menor sobrecarga, à medida em que o número de sorvedouros aumentou, obtendo: 23,45%, 22,08%, 20,59% e 20,01%. No cenário com 4 sorvedouros, o MARPL apresentou a menor sobrecarga, enquanto obteve os maiores PDRs, como visto na Figura 5.14. O resultado obtido pelo MARPL aponta que este protocolo se beneficia do aumento do número de nós sorvedouros, pois, no MARPL, as diversas DODAGs são atualizadas sempre que nós móveis são identificados, reduzindo a necessidade de aumento nas transmissões de mensagens de controle.

Na Figura 5.16, apresentam-se os resultados do atraso de entrega dos protocolos RPL, D-RPL, mRPL e MARPL no cenário com 50 nós móveis, e um número de 2, 4, 6 e 8 sorvedouros. Verifica-se que o aumento no número de nós sorvedouros impacta na redução do atraso para todos os protocolos. O RPL obteve os menores atrasos para todos os cenários avaliados, 32905, 25672, 23993 e 21519 milissegundos para os cenários com 2, 4, 6 e 8 sorvedouros respectivamente. O resultado obtido pelo RPL em termos de atraso de entrega pode ser justificado pelo baixo PDR. Os valores baixos da métrica RPL no RPL indicam que somente os nós mais próximos aos sorvedouros tiveram seus pacotes de dados entregues com sucesso. Através dos testes de significância estatística, pôde-se constatar que o atraso obtido pelo D-RPL se assemelha com o do RPL no cenário com 2 sorvedouros. Os testes de significância estatística também revelaram que os resultados do mRPL nos cenários com 4 e 6 sorvedouros são semelhantes (apesar do fato de que o atraso do mRPL reduziu de 43860 para 42306 milissegundos).

Figura 5.16 – Análise do Atraso de Entrega de Pacotes Variando-se o Número de Sorvedouros (50 Nós).



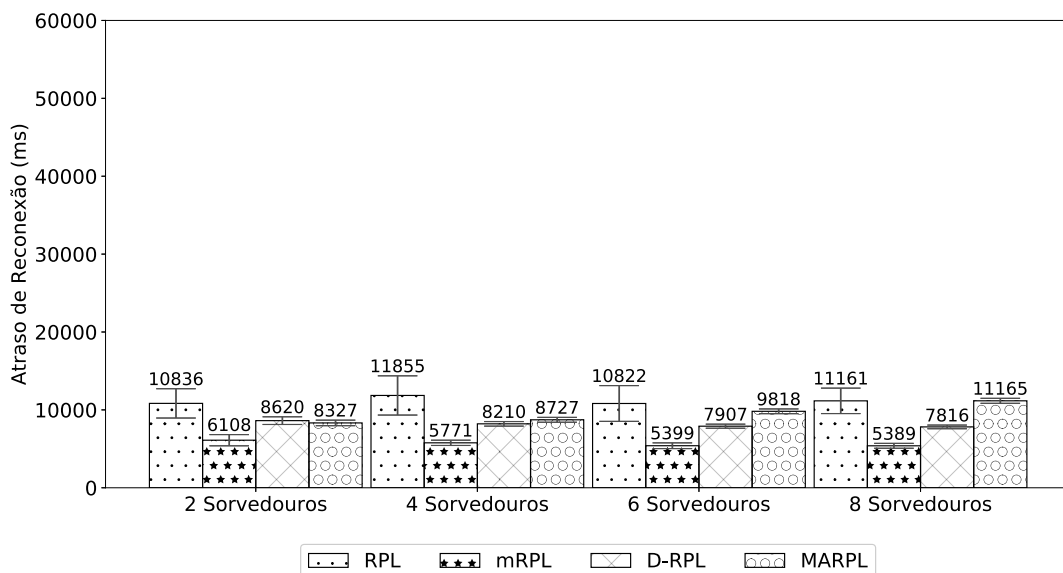
Fonte: O autor.

Observa-se que o atraso de entrega do MARPL reduz se comparado com os outros protocolos avaliados, 40087, 34728, 29044 e 24844 milissegundos para os cenários com 2, 4, 6 e 8 sorvedouros respectivamente. Isto porque, o MARPL busca manter a topologia sempre atualizada. A atualização da topologia é realizada através dos mecanismos de identificação de desconexão e de ajuste no temporizador *Trickle*. Portanto, no MARPL, os nós possuem as rotas atualizadas sempre que mudanças na topologia forem identificadas.

Os resultados de atraso de reconexão são apresentados na Figura 5.17. O protocolo mRPL apresenta os menores atrasos de reconexão, similarmente aos resultados obtidos nos experimentos nos quais variou-se o número de nós móveis, cujo resultados são apresentados na Figura 5.11. A justificativa para estes resultados do mRPL é a constante troca de mensagens que o protocolo realiza para rapidamente detectar as desconexões. Contudo, estas trocas de mensagens aumentam a sobrecarga do mRPL, como apresentado anteriormente na Figura 5.15.

As análises de significância estatística revelaram que os atrasos de reconexão dos protocolos RPL, mRPL e D-RPL não apresentaram diferença estatística a medida que o número de sorvedouros aumentou. Porém, o protocolo MARPL, apresentou uma tendência de aumento no atraso de reconexão à medida que o número de sorvedouros aumenta. Este resultado é consequência da redução da sobrecarga, como é apresentada na Figura 5.15. Como consequência, com menos pacotes de controle na rede, o atraso de reconexão aumenta, pois a taxa de atualização da topologia é reduzida. Todavia, observa-se que no MARPL, o atraso de reconexão não impactou negativamente no resultado do PDR e atraso de entrega. Como pode ser identificado, nas Figuras 5.17, 5.14 e 5.16.

Figura 5.17 – Análise do Atraso de Reconexão de Nó Pai Variando-se o Número de Sorvedouros (50 Nós).

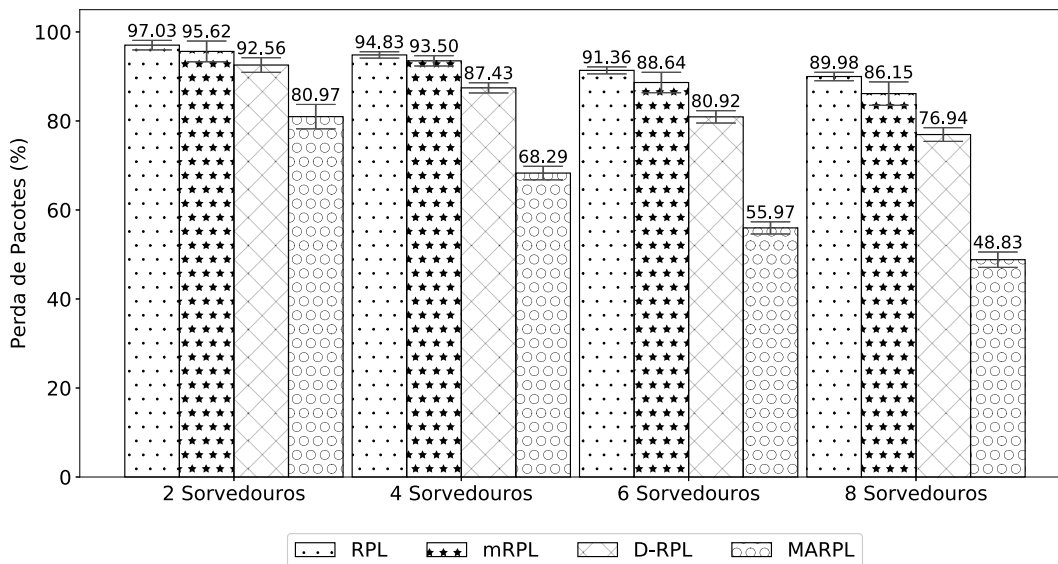


Fonte: O autor.

Na Figura 5.18 apresentam-se os resultados de perda de pacotes obtidos nos experimentos variando-se o número de nós sorvedouros. Verifica-se que o MARPL apresentou o menor percentual de perdas de pacotes entre todos os protocolos ava-

liados: 80,97%, 68,29%, 55,97% e 48,83%. Devido ao baixo percentual de perda de pacotes, o MARPL alcançou melhor desempenho em termos de PDR, atraso de entrega e sobrecarga, a medida que o número de sorvedouros aumenta.

Figura 5.18 – Análise da Perda de Pacotes Variando-se o Número de Sorvedouros (50 Nós).



Fonte: O autor.

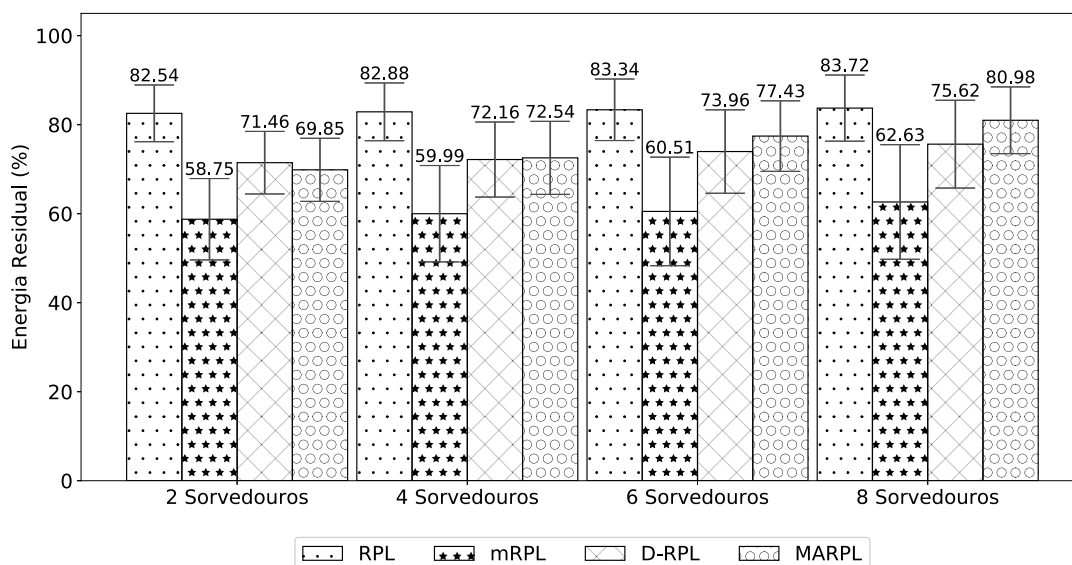
Na Figura 5.19, são descritos os resultados da energia residual do MARPL em comparação aos protocolos RPL, D-RPL e mRPL. O RPL obteve resultados similares para todos os cenários avaliados: 82,54%, 82,88%, 83,34% e 83,72%. Contudo, o RPL foi o protocolo que menos gastou energia durante as simulações. Este resultado é justificado pelo fato de que o RPL apresentou as menores sobrecargas de pacotes de controle, conforme apresentado na Figura 5.15.

Para os protocolos mRPL, D-RPL e MARPL, o aumento no número de sorvedouros causou um aumento na energia residual. Observa-se, dentre as propostas para o gerenciamento de mobilidade que o MARPL foi o mais eficiente quanto ao consumo de energia em relação ao aumento do número de sorvedouros.

Quando comparado o resultado de energia residual do MARPL apresentado na Figura 5.19 contra os resultados de PDR (Figura 5.14), atraso (Figura 5.16) e sobrecarga (Figura 5.15), constata-se que o MARPL alcançou o melhor desempenho em termos de taxa de sucesso de entrega, atraso de entrega, sobrecarga e energia residual ao passo que o número de nós sorvedouros aumenta.

Na Subsecção 5.3, apresenta-se a avaliação de desempenho do MARPL, variando-se a velocidade dos nós da rede.

Figura 5.19 – Análise da Energia Residual Variando-se o Número de Sorvedouros (50 Nós).



Fonte: O autor.

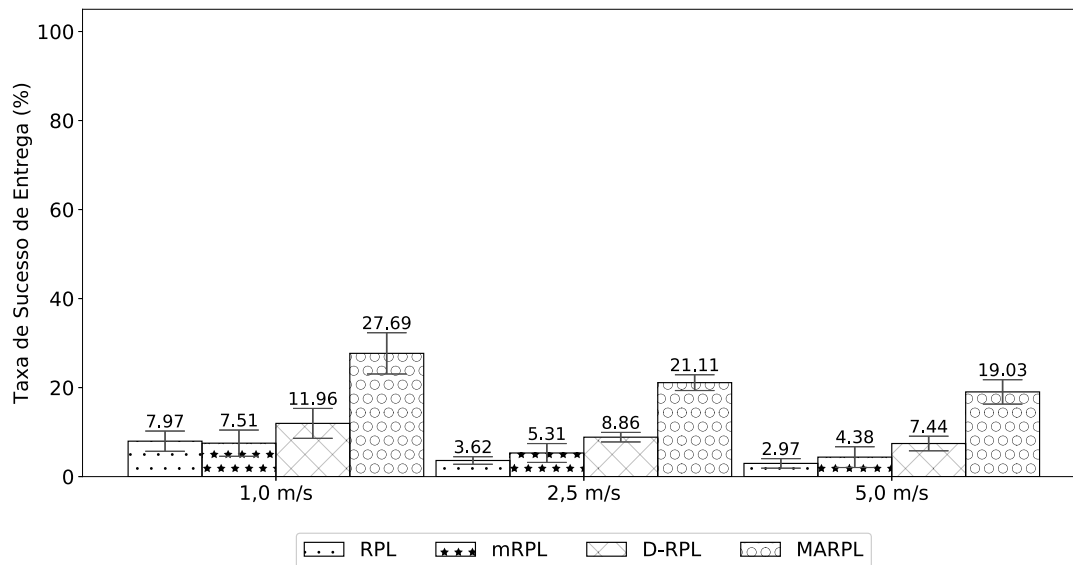
5.3 ANÁLISE DO MARPL EM FUNÇÃO DA VELOCIDADE DOS NÓS

Nesta subsecção são apresentados os resultados das simulações realizadas, visando avaliar o potencial do MARPL em relação à variação da velocidade dos nós. A velocidade dos nós foi ajustada em 1,0 m/s, 2,5 m/s e 5,0 m/s. O tamanho da área é de 200m x 200m, o número de sorvedouros é 2, foram distribuídos 50 nós móveis e o valor de θ é 3.

Na Figura 5.20, são apresentados os resultados de taxa de sucesso de entrega (PDR). Observa-se que o PDR diminui à medida em que a velocidade máxima dos nós aumenta. Os menores PDRs foram dos protocolos RPL e mRPL. Segundo as análises de significância estatística, as diferenças entre estes dois protocolos não foram estatisticamente significantes em nenhum dos cenários avaliados. O MARPL apresentou os maiores PDR em todos os cenários: 27,69%, 21,11% e 19,03%. Para o MARPL, os resultados nos cenários com velocidades máximas de 2,5 m/s e 5,0 m/s não apresentaram diferença estatística. A diminuição do PDR à medida que a velocidade máxima dos nós aumenta é justificada pelo aumento de instabilidade na rede, consequência da maior mobilidade dos nós.

Os resultados do atraso de entrega de pacotes são apresentados na Figura 5.21. Os resultados obtidos pelo RPL e D-RPL (40164 e 43321 milissegundos respectivamente) foram os menores no cenário com a velocidade de 1,0 m/s. Estes resultados são estatisticamente semelhantes conforme os testes estatísticos. Neste mesmo ce-

Figura 5.20 – Análise da Taxa de Sucesso de Transmissão Variando-se a Velocidade dos Nós (50 Nós).

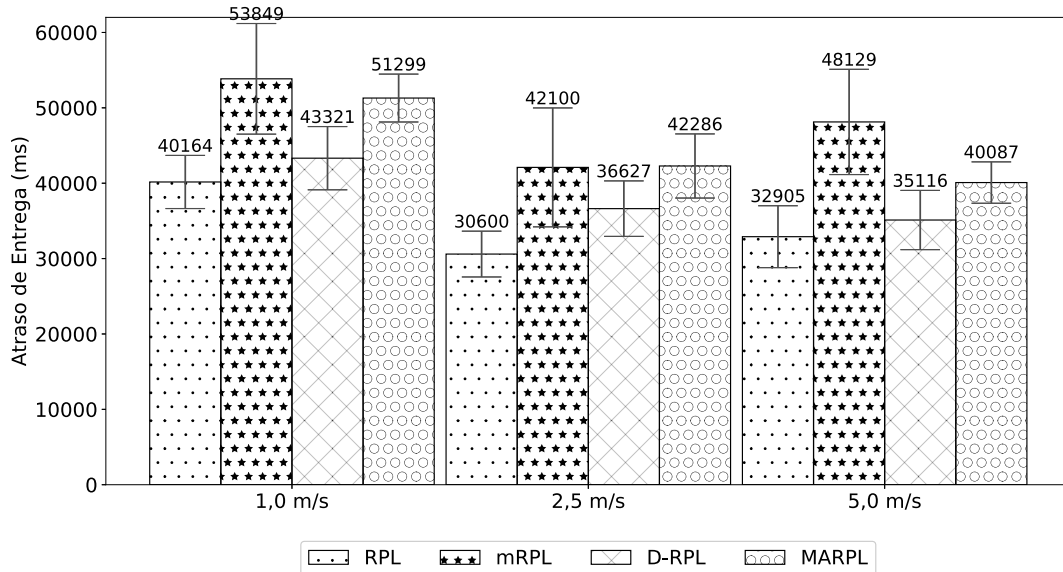


Fonte: O autor.

nário, os atrasos dos protocolos mRPL e MARPL (53849 e 51299 milissegundos respectivamente) foram os maiores e são estatisticamente semelhantes. No cenário com velocidade de 2,5 m/s, o RPL apresentou menor atraso de entrega com 30600 milissegundos. O D-RPL obteve o segundo menor atraso de entrega com 36627 milissegundos com a velocidade de 2,5 m/s. O mRPL e MARPL tiveram resultados estatisticamente semelhantes (42100 e 42286 milissegundos respectivamente). No cenário com velocidade máxima de 5,0 m/s, os resultados do RPL e D-RPL são novamente estatisticamente semelhantes com 32905 e 35116 milissegundos respectivamente. Deste modo, o MARPL apresenta o segundo maior atraso, estatisticamente diferente, com 40087 milissegundos. A conclusão, é de que o aumento na velocidade dos nós causa uma tendência de redução no atraso de entrega no MARPL. Isto indica que o MARPL é um protocolo capaz de manter bom desempenho em termos de atraso de entrega à medida que a instabilidade da topologia aumenta.

O atraso de entrega obtido pelos protocolos RPL e mRPL não apresentou uma tendência de redução à medida que a velocidade dos nós aumentou. O protocolo RPL apresentou 40164 milissegundos de atraso no cenário com velocidade de 1,0 m/s, 30600 e 32905 nos cenários com velocidades de 2,5 m/s e 5,0 m/s respectivamente. Observa-se que o atraso do RPL no cenário com velocidade de 2,5 m/s foi menor do que o obtido no cenário com 5,0 m/s. Isto também foi observado quando analisou-se o protocolo mRPL. Este comportamento é justificado pelo fato de que os protocolos

Figura 5.21 – Análise do Atraso de Entrega de Pacotes Variando-se a Velocidade dos Nós (50 Nós).



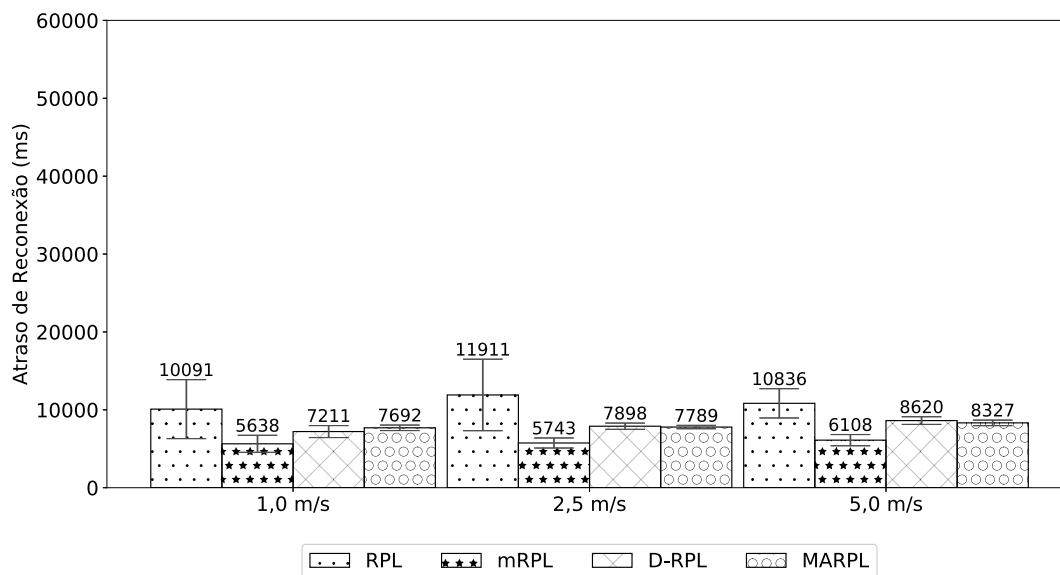
Fonte: O autor.

RPL e mRPL não possuem mecanismos de ajuste no temporizador *Trickle*. Sem estes ajustes, a topologia demora mais para atualizar quando houver movimentação dos nós. Como pode ser observado nos resultados apresentados pelos protocolos D-RPL e MARPL, que realizam ajustes no temporizador *Trickle*, o atraso tende a diminuir à medida que a velocidade dos nós aumenta.

Na Figura 5.22, apresentam-se os resultados do atraso de reconexão para os protocolos avaliados. Constata-se que o atraso de reconexão dos protocolos que possuem mecanismos para o gerenciamento de mobilidade, mRPL, D-RPL e MARPL, reduz conforme a velocidade aumenta. O aumento na velocidade dos nós aumenta a frequência das desconexões. Segundo a análise de significância estatística, os resultados encontrados com os protocolos MARPL e D-RPL não foram estatisticamente diferentes nos cenários com velocidades máximas de 2,5 m/s e 5,0 m/s.

Na Figura 5.23, os resultados de sobrecarga são apresentados. O aumento do deslocamento dos nós implica em um aumento na sobrecarga dos pacotes de controle no protocolo MARPL. Este comportamento é justificado pela mobilidade dos nós, que necessitam buscar novos nós pais à medida que se deslocam. Contudo, segundo a análise de significância estatística, os resultados dos protocolos D-RPL e MARPL para os cenários com 2,5 e 5,0 m/s não apresentaram diferença estatística. Os resultados dos protocolos RPL e mRPL não apresentaram diferenças estatísticas em todos os cenários analisados na Figura 5.23.

Figura 5.22 – Análise do Atraso de Reconexão de Nó Pai Variando-se a Velocidade dos Nós (50 Nós).



Fonte: O autor.

Na Figura 5.24, apresentam-se os resultados de energia residual. Segundo a análise de significância estatística, o MARPL não apresentou diferença estatística nas velocidades de 1,0 m/s, 2,5 m/s e 5,0 m/s, com 70,20%, 69,64% e 69,85% respectivamente. Este mesmo comportamento se observa no D-RPL. Contata-se que há um aumento na energia residual nos protocolos RPL e mRPL à medida que a velocidade aumenta. Isto em razão da redução da taxa de sucesso de entrega que o aumento no deslocamento dos nós causa, como pode ser visto na Figura 5.20. Dessa forma, menos pacotes de dados foram trafegados com sucesso na rede.

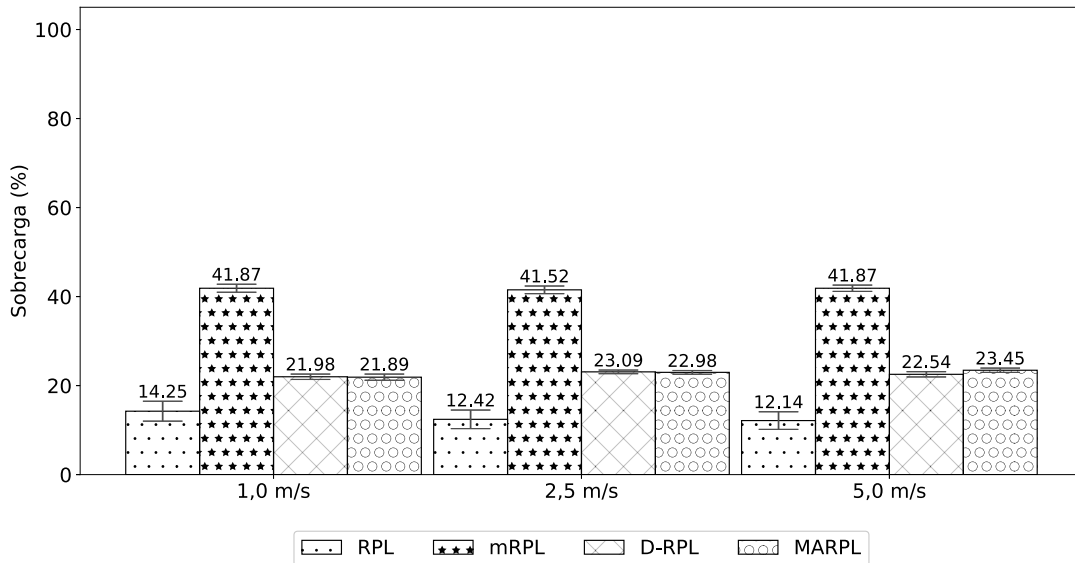
Na Subsecção 5.4, é apresentada a análise da escalabilidade do protocolo MARPL.

5.4 ANÁLISE DA ESCALABILIDADE DO MARPL

Nesta subsecção, avaliou-se a escalabilidade do protocolo MARPL em uma área de 1000m x 1000m. A velocidade máxima é de 5,0 m/s. O número de nós móveis é 100 e 20 sorvedouros foram utilizados. O aumento tanto no número de nós sensores quanto de nós sorvedouros neste cenário, é justificado dado o fato deste ser esparso. O MARPL foi comparado com os protocolos RPL, mRPL e D-RPL.

Na Figura 5.25, expõem-se os resultados do PDR para os protocolos avaliados. Verifica-se que o PDR de todos os protocolos avaliados é baixo no cenário esparso. O RPL com 1,36%, mRPL com 1,72%, D-RPL com 1,46% e MARPL com

Figura 5.23 – Análise da Sobrecarga de Mensagens de Controle Variando-se a Velocidade dos Nós (50 Nós).



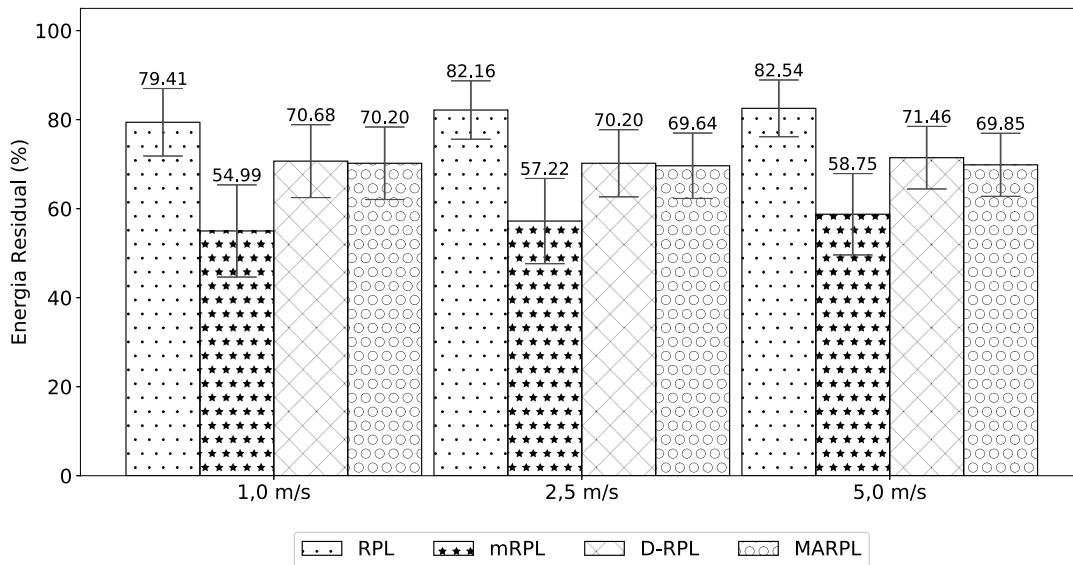
Fonte: O autor.

2,26%. Os testes de significância estatística realizados apontam que os protocolos RPL, mRPL e D-RPL não apresentaram PDRs estatisticamente diferentes. Os baixos valores de PDR são um indício de que protocolos baseados no RPL não apresentam bom desempenho em topologias muito esparsas. Tendo em vista que, a estrutura da topologia RPL, chamada de *Destination Oriented Directed Acyclic Graph* (DODAG), deve ser uma estrutura contínua de conexão entre os nós em direção aos nós sorvedouros. O maior PDR neste cenário foi do MARPL, devido aos mecanismos para tratamento das desconexões que mantém a topologia atualizada com maior frequência.

Na Figura 5.26, ilustram-se os resultados do atraso de reconexão. O RPL e D-RPL apresentam atrasos menores e estatisticamente semelhantes (26975 e 22984 milissegundos respectivamente). O mRPL e MARPL apresentam resultados maiores e estatisticamente semelhantes (47034 e 43348 milissegundos respectivamente). Este desempenho do MARPL é justificado pelo fato de que em uma rede muito esparsa, como a do cenário avaliado, o monitoramento da vizinhança é dificultado. Porém, os nós que estão conectados na DODAG, conseguem transmitir com sucesso mais pacotes para os nós sorvedouros no MARPL, quando comparado com os outros protocolos (como pode ser visto na Figura 5.25).

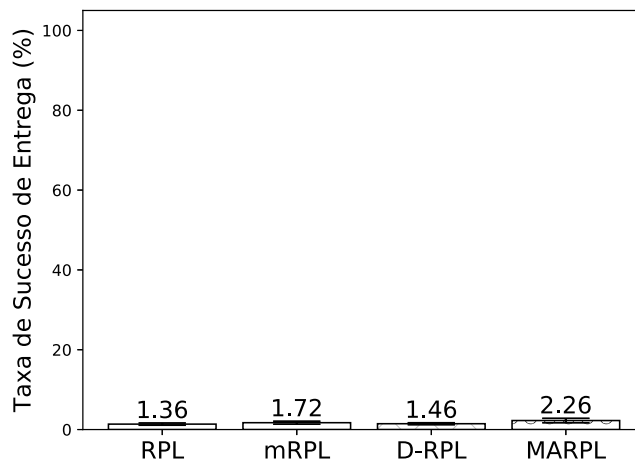
Os resultados de sobrecarga são apresentados na Figura 5.27. O RPL obteve a menor sobrecarga com 10,37% e o mRPL a maior com 56,51%. O MARPL apresen-

Figura 5.24 – Análise da Energia Residual Variando-se a Velocidade dos Nós (50 Nós).



Fonte: O autor.

Figura 5.25 – Análise da Taxa de Sucesso de Transmissão no Cenário de 1km² (100 Nós e Velocidade de 5,0 m/s).

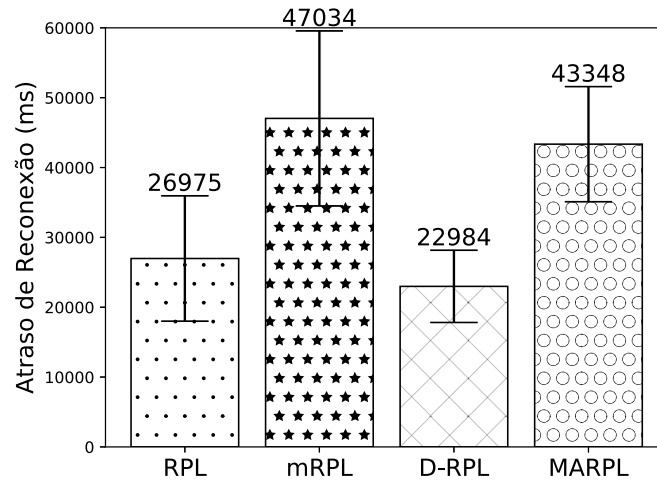


Fonte: O autor.

tou a menor sobrecarga (16,02%) entre os protocolos com suporte à mobilidade.

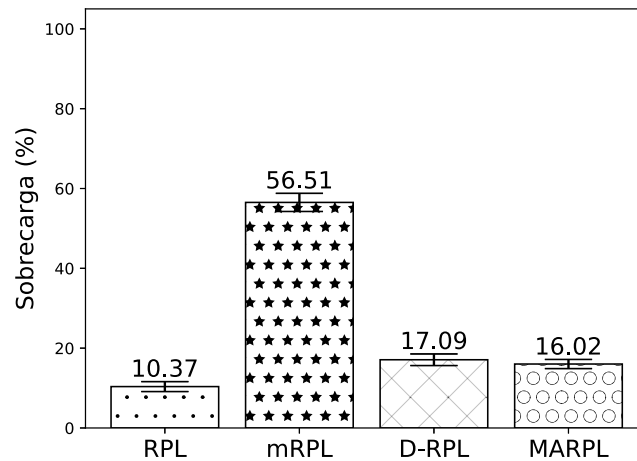
Na Figura 5.28, analisa-se a energia residual no mesmo cenário de configuração da Figura 5.25. O MARPL apresentou a maior economia de energia, mesmo quando comparado com o RPL canônico. A energia residual, o PDR (na Figura 5.25)

Figura 5.26 – Análise do Atraso de Reconexão de Nó Pai no Cenário de 1km² (100 Nós e Velocidade de 5,0 m/s).



Fonte: O autor.

Figura 5.27 – Análise da Sobrecarga de Mensagens de Controle no Cenário de 1km² (100 Nós e Velocidade de 5,0 m/s).

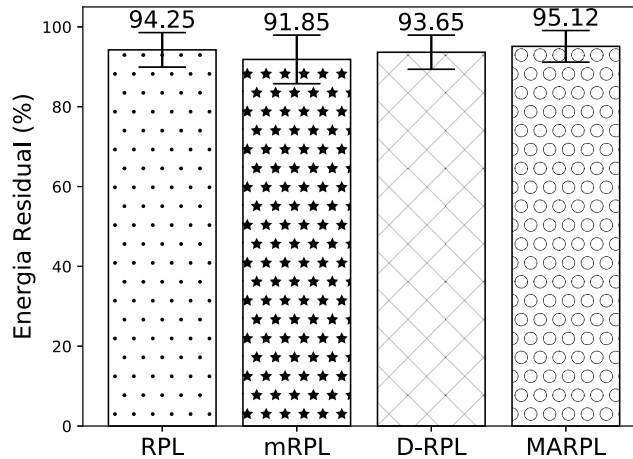


Fonte: O autor.

e a sobrecarga (demonstrado na Figura 5.27) do MARPL apontam que este protocolo obteve o melhor custo benefício em termos de PDR e gasto energético.

O MARPL apresentou um dos maiores atrasos de reconexão no cenário avaliado, como pode ser visto na Figura 5.26. Contudo, o MARPL superou os outros pro-

Figura 5.28 – Análise da Energia Residual no Cenário de 1km² (100 Nós e Velocidade de 5,0 m/s).



Fonte: O autor.

protocolos avaliados quanto ao PDR, sobrecarga e energia residual. A partir deste resultado, conclui-se que mesmo em um cenário esparsos e com alta instabilidade da topologia (devido à movimentação dos nós), o MARPL apresentou melhor performance em relação às métricas mencionadas.

Na Subsecção 5.5, analisa-se o impacto do parâmetro θ , utilizado pelo MARPL para ajustar o temporizador $T_{reachable}$. Este temporizador é utilizado para detectar a desconexão entre os nós.

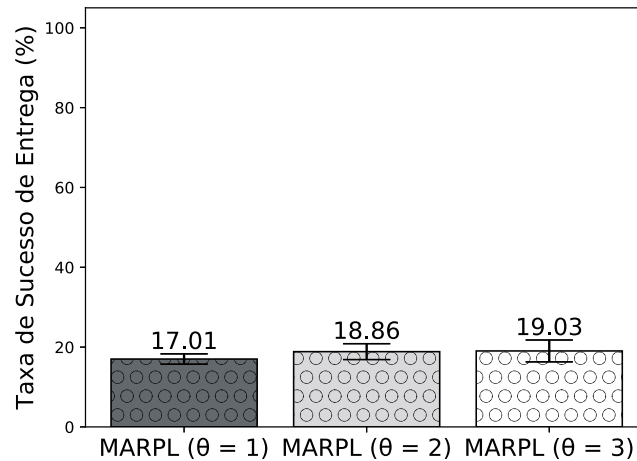
5.5 ANÁLISE DO AJUSTE DO PARÂMETRO θ UTILIZADO PELO MARPL

Nesta subsecção, são apresentados os resultados dos ajustes feitos no parâmetro θ . O valor de θ influencia o período de tempo que um nó deve aguardar por recebimento de pacotes de dados ou controle do nó pai para determinar se houve ou não uma desconexão entre os nós. Os cenários avaliados seguem a mesma configuração dos experimentos descritos nas Seções 5.1, 5.2 e 5.3, área de 200m x 200m, velocidade máxima de 5,0 m/s, 50 nós móveis e 2 nós sorvedouros.

Na Figura 5.29, são demonstrados os PDRs do MARPL variando os valores de θ entre 1, 2 e 3. Observa-se que o maior PDR, 19,03%, foi obtido quando $\theta = 3$. Porém, não houve diferença estatística entre $\theta = 2$ e $\theta = 3$. Dado os PDRs obtidos, considera-se que o aumento do valor de θ influencia o PDR obtido pelo MARPL.

Na Figura 5.30a, analisam-se os resultados de sobrecarga dos pacotes de controle no MARPL, diante da variação do parâmetro θ . Verifica-se que é possível re-

Figura 5.29 – Análise da Taxa de Sucesso de Transmissão do MARPL Variando-se o Valor de θ (50 Nós e Velocidade de 5,0 m/s).



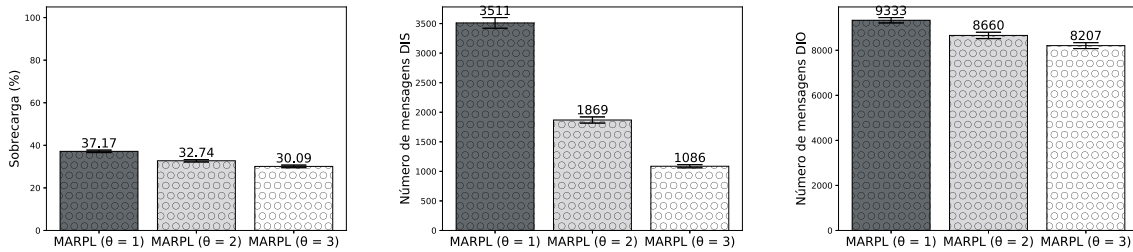
Fonte: O autor.

duzir a sobrecarga do MARPL aumentando o valor de θ . Com $\theta = 3$, um nó aguarda por mais tempo pelo recebimento de pacotes de controle ou dados de seu nó pai. Conseqüentemente, os nós enviam menos mensagens DIS, como pode ser visto na Figura 5.30b. A redução no envio de mensagens DIS, reduz também as chances dos nós vizinhos reiniciarem o temporizador *Trickle* e causa uma redução na transmissão de mensagens DIO, veja a Figura 5.30c. Conforme argumentado anteriormente, no MARPL, um nó que identifica sua desconexão com o nó pai transmite uma mensagem DIS para todos os nós em sua vizinhança. Desta maneira, a topologia pode ser atualizada através do aumento na transmissão de mensagens DIO causado pelo reinício do temporizador *Trickle*, conforme a especificação canônica do RPL (RFC 6550, 2012).

Na Figura 5.31, exibe-se o atraso de entrega do MARPL. Verifica-se que o menor atraso (40087 milissegundos) foi obtido quando $\theta = 3$. Isso se justifica, porque $\theta = 3$ implica em menor sobrecarga, ou seja, menos colisões na transmissão de pacotes na rede. Isso permite que os pacotes de dados possam ser transmitidos mais rapidamente.

O valor de θ impacta o atraso de reconexão. Conforme é demonstrado na Figura 5.32, pois quando $\theta = 1$, o atraso de reconexão foi maior, 9196 milissegundos. $\theta = 2$ e $\theta = 3$ apresentaram atrasos de reconexão semelhantes estatisticamente (8307 e 8327 respectivamente). θ é utilizado como um fator de ajuste para o temporizador $T_{reachable}$, de modo que θ multiplica o valor da taxa de geração de dados da aplicação da rede. Nas simulações apresentadas na Seções 5.1, 5.2 e 5.3, esta taxa é de 3

Figura 5.30 – Análise das Mensagens de Controle do MARPL Variando-se o Valor de θ (50 Nós e Velocidade de 5,0 m/s).



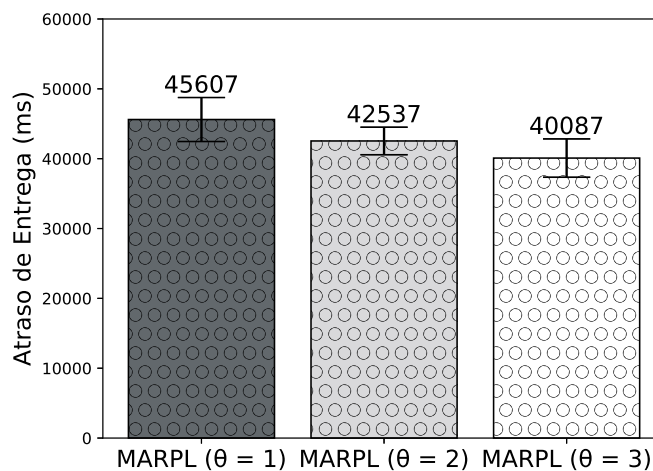
(a) Sobrecarga.

(b) Mensagens DIS.

(c) Mensagens DIO.

Fonte: O autor.

Figura 5.31 – Análise do Atraso de Entrega do MARPL Variando-se o Valor de θ (50 Nós e Velocidade de 5,0 m/s).

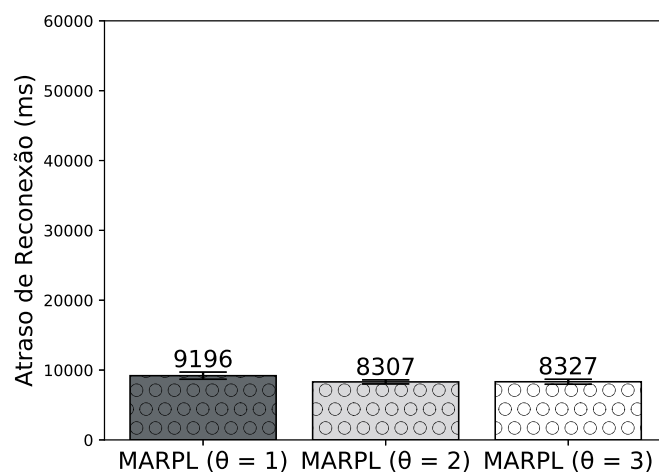


Fonte: O autor.

segundos. Conclui-se com base na análise do atraso de reconexão que $T_{reachable}$ deve preferencialmente ter um valor acima da taxa de geração de dados. Por exemplo, se $\theta = 1$, $T_{reachable}$ terá o mesmo valor que a taxa de geração de dados dos sensores, aumentando o atraso de reconexão.

Na Figura 5.33, apresenta-se a energia residual nos cenários com θ igual a 1, 2 e 3: 70,15%, 69,66% e 69,85% respectivamente. Segundo a análise estatística realizada, não houve diferenças expressivas no consumo de energia, variando-se o

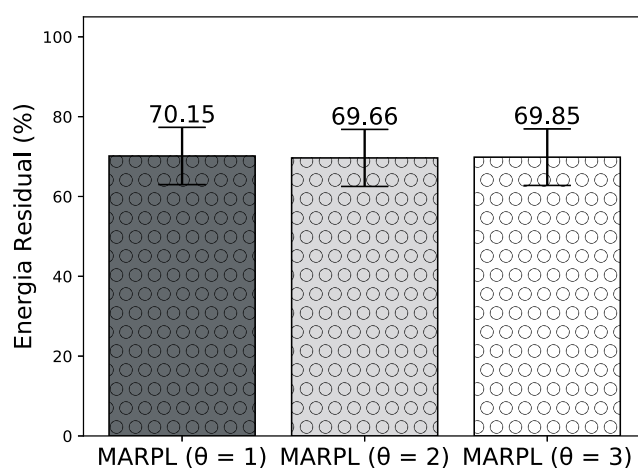
Figura 5.32 – Análise do Atraso de Reconexão de Nó Pai do MARPL Variando-se o Valor de θ (50 Nós e Velocidade de 5,0 m/s).



Fonte: O autor.

valor de θ .

Figura 5.33 – Análise da Energia Residual do MARPL Variando-se o Valor de θ (50 Nós e Velocidade de 5,0 m/s).



Fonte: O autor.

5.6 CONCLUSÕES DO CAPÍTULO

Neste capítulo foi apresentada a análise de desempenho do MARPL em relação à taxa de sucesso de entrega, atraso de entrega, atraso de reconexão, sobrecarga e energia residual. Com o objetivo de avaliar os protocolos RPL, D-RPL, mRPL e MARPL em diferentes cenários, variou-se o número de nós móveis, do número de nós sorvedouros, da velocidade dos nós, da escalabilidade e do ajuste do parâmetro θ do MARPL.

As análises em termos do número de nós móveis na topologia apontaram que o aumento no número de nós sensores móveis causa redução no PDR pelo aumento na densidade da rede, e consequentes colisões. O aumento no número de nós em movimentação também impacta na instabilidade da rede. Este fato justifica a diminuição do PDR nestes cenários. O aumento no número de nós sensores também impacta negativamente no atraso de entrega, por motivos similares à redução do PDR e pelo aumento na sobrecarga de pacotes de controle. Uma vez que o aumento no número de nós sensores causa um aumento na sobrecarga do MARPL.

O aumento no número de nós sorvedouros possibilitou que o MARPL obtivesse alto PDR, atraso de entrega e energia residual. Portanto, conclui-se que um aumento no número de nós sorvedouros pode ser uma estratégia para aumentar o desempenho do MARPL.

Com o aumento na velocidade dos nós, verificou-se uma redução no PDR no MARPL. Esta redução é compreensível, pois há um aumento na instabilidade da rede. Contudo, o atraso de entrega é reduzido. Pois, o aumento no deslocamento dos nós, e consequente aumento na instabilidade da topologia, requer a transmissão de mais mensagens de controle para que a topologia seja atualizada com maior frequência. Um aumento na frequência de atualização da topologia permite o descobrimento de rotas mais eficientes ao longo do tempo.

A análise de escalabilidade revelou que os protocolos baseados no RPL apresentam limitações quando utilizados em topologias esparsas. Um indício disto é o baixo PDR obtido por todos os protocolos avaliados. Contudo, mesmo nestes cenários, o MARPL apresentou o maior PDR, baixa sobrecarga e maior energia residual. Porém, o MARPL apresentou alto atraso de reconexão nestes cenários. A partir da análise do ajuste do parâmetro θ utilizado pelo MARPL, conclui-se que um aumento no valor de θ apresentou melhores desempenhos para os cenários avaliados.

Na Seção 6, apresentam-se as conclusões deste trabalho de mestrado, e as perspectivas de trabalhos futuros.

6 CONCLUSÃO

Foi proposto neste trabalho de dissertação de mestrado, um mecanismo de gerenciamento de mobilidade para o protocolo de roteamento *IPv6 Routing Protocol for Low-Power and Lossy Networks* (RPL), chamado *Mobility Aware RPL* (MARPL). O protocolo RPL foi desenvolvido com foco em redes de baixa capacidade *Low Power and Lossy Network* (LLN), comum em aplicações de sensoriamento de Internet das Coisas (o inglês *Internet of Things* - IoT). O protocolo RPL é considerado pela literatura como o protocolo de roteamento mais apto a ser utilizado em aplicações com LLN. Apesar do protocolo RPL em sua versão canônica haver sido desenvolvido para redes constituídas apenas por nós estáticos, diversas aplicações de IoT compostas por dispositivos móveis têm adotado o RPL como protocolo de roteamento. O MARPL foi desenvolvido, visando mitigar as limitações que o protocolo RPL apresenta quando utilizado com topologias que possuem nós móveis.

O MARPL consiste de uma abordagem reativa e realiza o monitoramento da vizinhança de um nó. Através deste monitoramento, um nó identifica possíveis variações das potências de sinal medidas entre si e seus vizinhos, para que seja aferida a possibilidade do nó estar se movimentando. Por meio de uma abordagem reativa, aliada com a métrica RSSI, é possível analisar a movimentação da vizinhança, mantendo baixo o número de mensagens de controle na rede.

O MARPL é composto de três mecanismos: (i) um mecanismo para o cálculo da métrica Variabilidade de Vizinhança. Este mecanismo apresenta uma abordagem entre camadas. A camada de Rede utiliza medições da potência de sinal vindas da camada de Acesso ao Meio para o cálculo da métrica; (ii) a métrica Variabilidade de Vizinhança é utilizada pelo mecanismo de ajuste do temporizador *Trickle*, responsável pelo envio periódico de mensagens de controle DIO; (iii) um mecanismo para a identificação da inalcançabilidade do nó pai e reingresso na DODAG. No MARPL, para a atualização do nó pai, utiliza-se a métrica Variabilidade de Vizinhança. Desta maneira, as rotas escolhidas são aquelas compostas por nós que tenham maior probabilidade de estarem estáticos. Com base nos resultados apresentados no Capítulo 5, verificou-se que o MARPL reduz o impacto das frequentes desconexões causadas pela mobilidade, principalmente em termos de taxa de sucesso de entrega nos cenários em que variou-se o número de nós, número de sorvedouros e velocidade dos nós.

O funcionamento dos mecanismos propostos acima dependem de dois temporizadores para o gerenciamento da mobilidade no MARPL: $T_{monitoring}$, o período em que um nó deve monitorar sua vizinhança. No MARPL, $T_{monitoring}$ determina a periodi-

cidade de atualização da métrica Variabilidade de Vizinhaça, e seu valor é calculado da frequência da geração de dados pelos sensores; $T_{reachable}$, o período máximo em que um nó deve aguardar o recebimento de uma mensagem de seu nó pai, a fim de determinar sua alcançabilidade. $T_{reachable}$, assim como $T_{monitoring}$, tem seu valor calculado a partir da frequência da geração de dados da aplicação.

O MAPRL foi implementado através da extensão da implementação do RPL contida no Sistema Operacional Contiki. O Contiki foi proposto para dispositivos de baixa capacidade, como os utilizados em aplicações de IoT. A implementação foi analisada através de experimentos com simulações e posterior análise dos *logs* gerados. Implementou-se também as propostas mRPL de Fotouhi et al. (2015) e D-RPL de Kharrufa et al. (2017). Portanto, a análise apresentada neste trabalho foi realizada comparando-se o MARPL com o RPL canônico e mais essas duas propostas. As implementações poderão servir de base para que novas pesquisas possam ser realizadas sobre roteamento em LLNs

Através das análises realizadas no Capítulo 5, verificou-se que a abordagem proposta, MAPRL, apresentou bom desempenho na taxa de sucesso de transmissão, redução no atraso de entrega, sobrecarga e consumo energético. Contudo, o MARPL possui limitações em termos do atraso de reconexão, como foi apresentado na Seção 5. Análises do ajuste do parâmetro θ , utilizado pelo MARPL para ajustar o temporizador $T_{reachable}$, também foram apresentadas. Concluiu-se que o valor de θ impacta no PDR, atraso de entrega, atraso de reconexão e sobrecarga.

6.1 TRABALHOS FUTUROS

A partir da análise de escalabilidade e em função do número de nós servidores, verificou-se que o MARPL obteve alto atraso de reconexão em alguns dos cenários avaliados. Como trabalhos futuros, sugere-se o estudo de métodos para mitigar essa limitação do MARPL, preservando o bom desempenho em termos de taxa de sucesso de entrega, atraso, sobrecarga e consumo energético.

Propõe-se a execução de experimentos com outras propostas de mecanismos para o gerenciamento da mobilidade no RPL encontradas na literatura. Ademais, recomenda-se realização de testes com dispositivos reais.

6.2 OUTRAS CONTRIBUIÇÕES DA DISSERTAÇÃO

Os resultados obtidos durante a pesquisa sobre roteamento em LLNs e com o MARPL foram publicadas nas seguintes conferências:

- V. Marques e J. Kniess, “Mobility Aware RPL (MARPL): Mobility to RPL on Neigh-

bor Variability”, 14th International Conference on Green, Pervasive and Cloud Computing (GPC 2019), (MARQUES, 2019).

- V. Marques, J. Kniess e R. Parpinelli, “An Ant Colony-based Mesh Routing Protocol for Maximizing Low Power and Lossy Networks Lifetime”, 16th ACM International Symposium on Mobility Management and Wireless Access (MobiWac 2018), (MARQUES, 2018a).
- V. Marques, J. Kniess e R. Parpinelli, “An Energy Efficient Mesh LLN Routing Protocol Based on Ant Colony Optimization”, 16th International Conference on Industrial Informatics (INDIN 2018), (MARQUES, 2018b).

REFERÊNCIAS

- ANAND, M. C. R.; TAHILIANI, M. P. mRPL++: Smarter-HOP for optimizing mobility in RPL. In: IEEE. **2016 IEEE Region 10 Symposium (TENSYP)**. Bali, Indonésia: IEEE, 2016. p. 36–41.
- ASCHENBRUCK, N. et al. Bonnmotion: a mobility scenario generation and analysis tool. In: ICST (INSTITUTE FOR COMPUTER SCIENCES, SOCIAL-INFORMATICS AND TELECOMMUNICATIONS ENGINEERING). **Proceedings of the 3rd international ICST conference on simulation tools and techniques**. Torremolinos, Spain: ACM, 2010. p. 51.
- BAI, F.; HELMY, A. A survey of mobility models. **Wireless Adhoc Networks**, University of Southern California, v. 206, p. 147, 2004.
- BARCELO, M. et al. Addressing mobility in rpl with position assisted metrics. **IEEE Sensors Journal**, IEEE, v. 16, n. 7, p. 2151–2161, 2016.
- BOUAZIZ, M.; RACHEDI, A. A survey on mobility management protocols in wireless sensor networks based on 6lowpan technology. **Computer Communications**, Elsevier, v. 74, p. 3–15, 2016.
- BOUAZIZ, M.; RACHEDI, A.; BELGHITH, A. EKF-MRPL advanced mobility support routing protocol for internet of mobile things: Movement prediction approach. **Future Generation Computer Systems**, Elsevier, p. 19–24, 2017.
- CHANG, J.-M. et al. The 6lowpan ad-hoc on demand distance vector routing with multi-path scheme. In: IET. **IET International Conference on Frontier Computing**. Taichung, Taiwan, 2010. p. 1–6.
- COBARZAN, C.; MONTAVONT, J.; NOEL, T. Analysis and performance evaluation of RPL under mobility. In: IEEE. **2014 IEEE Symposium on Computers and Communications (ISCC)**. Funchal, Portugal: IEEE, 2014. p. 1–6.
- DAWSON-HAGGERTY, S.; TAVAKOLI, A.; CULLER, D. Hydro: A hybrid routing protocol for low-power and lossy networks. In: IEEE. **2010 First IEEE International Conference on Smart Grid Communications**. Gaithersburg, Estados Unidos da América, 2010. p. 268–273.
- DONG, Q.; DARGIE, W. Evaluation of the reliability of RSSI for indoor localization. In: IEEE. **2012 International Conference on Wireless Communications in Underground and Confined Areas**. Clermont Ferrand, França: IEEE, 2012. p. 1–6.
- DUNKELS, A. et al. **The Contiki OS: The operating system for the internet of things**. 2011. <http://www.contikios.org>. Online; Acessado em: 01/03/2019.
- FOTOUHI, H.; MOREIRA, D.; ALVES, M. mRPL: Boosting mobility in the internet of things. **Ad Hoc Networks**, Elsevier, p. 17–35, 2015.
- FOTOUHI, H. et al. Smart-Hop: A reliable handoff mechanism for mobile wireless sensor networks. **Wireless Sensor Networks**, Springer, p. 131–146, 2012.

GADDOUR, O.; KOUBÂA, A. Rpl in a nutshell: A survey. **Computer Networks**, Elsevier, v. 56, n. 14, p. 3163–3178, 2012.

GADDOUR, O. et al. Co-RPL: RPL routing for mobile low power wireless sensor networks using Corona mechanism. In: **9th IEEE International Symposium on Industrial Embedded Systems (SIES 2014)**. Pisa, Itália: IEEE, 2014. p. 200–209.

GARA, F. et al. An adaptive timer for RPL to handle mobility in wireless sensor networks. In: **2016 International Wireless Communications and Mobile Computing Conference (IWCMC)**. Paphos, Chipre: IEEE, 2016. p. 678–683.

GHALEB, B.; AL-DUBAI, A.; EKONOMOU, E. E-Trickle: Enhanced trickle algorithm for low-power and lossy networks. In: IEEE. **2015 IEEE International Conference on Computer and Information Technology**. Liverpool, Reino Unido, 2015. p. 1123–1129.

GNAWALI, O. et al. Collection Tree Protocol. In: ACM. **7th ACM Conference on Embedded Networked Sensor Systems**. New York, Estados Unidos da América: ACM, 2009. p. 1–14.

GUBBI, J. et al. Internet of Things (IoT): A vision, architectural elements, and future directions. **Future generation computer systems**, Elsevier, v. 29, n. 7, p. 1645–1660, 2013.

HEINZELMAN, W. R.; CHANDRAKASAN, A.; BALAKRISHNAN, H. Energy-efficient communication protocol for wireless microsensor networks. In: IEEE. **Proceedings of the 33rd annual Hawaii international conference on system sciences**. Hawaii, Estados Unidos da América, 2000. p. 1–10.

IEEE. Ieee standard for local and metropolitan area networks part 15.6: Wireless body area networks. **IE EE Std 802.15. 6**, IEEE, p. 1–271, 2012.

IEEE. Ieee standard for high data rate wireless multi-media networks. **IE EE Std 802.15. 3**, IEEE, p. 1–510, 2016. Online; Acessado em: 01/03/2019.

IEEE. **IEEE Standard for Low-Rate Wireless Networks**. [S.I.]: IEEE, 2016. 1–708 p. <https://ieeexplore.ieee.org/servlet/opac?punumber=7460873>. Online; Acessado em: 01/03/2019.

INSTRUMENTS, T. **CC2420: 2.4 GHz IEEE 802.15. 4/ZigBee-ready RF Transceiver**. [S.I.]: Texas Instruments, 2006. 1–93 p. <https://www.ti.com/lit/ds/symlink/cc2420.pdf>. Online; Acessado em: 01/03/2019.

IOVA, O. et al. Rpl: The routing standard for the internet of things... or is it? **IEEE Communications Magazine**, IEEE, v. 54, n. 12, p. 16–22, 2016.

KARKAZIS, P. et al. Evaluating routing metric composition approaches for QoS differentiation in low power and lossy networks. **Wireless networks**, Springer, v. 19, n. 6, p. 1269–1284, 2013.

KHARRUFA, H. et al. Dynamic RPL for multi-hop routing in IoT applications. In: **2017 13th Annual Conference on Wireless On-demand Network Systems and Services (WONS)**. Jackson, Estados Unidos da América: IEEE, 2017. p. 100–103.

KO, J.; CHANG, M. MoMoRo Providing Mobility Support for Low-Power Wireless Applications. **IEEE Systems Journal**, v. 9, n. 2, p. 585–594, jun. 2015. ISSN 1932-8184.

KORBI, I. E. et al. Mobility enhanced rpl for wireless sensor networks. In: **2012 third international conference on the network of the future (NOF)**. Gammarth, Tunísia: IEEE, 2012. p. 1–8.

KUMAR, V.; TIWARI, S. Routing in ipv6 over low-power wireless personal area networks (6lowpan): A survey. **Journal of Computer Networks and Communications**, Hindawi, v. 2012, p. 1–11, 2012.

LAMAAZI, H. et al. Performance assessment of the routing protocol for low power and lossy networks. In: **Wireless Networks and Mobile Communications (WINCOM), 2015 International Conference on**. Marrakech, Marrocos: IEEE, 2015. p. 1–8.

LAMAAZI, H. et al. Mobility support for the routing protocol in low power and lossy networks. In: **Advanced Information Networking and Applications Workshops (WAINA), 2016 30th International Conference on**. Crans-Montana, Suíça: IEEE, 2016. p. 809–814.

LINDSEY, S.; RAGHAVENDRA, C. S. Pegasus: Power-efficient gathering in sensor information systems. In: IEEE. **Proceedings, IEEE Aerospace Conference**. Big Sky, Estados Unidos da América, 2002. v. 3, p. 3–3.

LONG, N. T. et al. Comparative performance study of RPL in wireless sensor networks. In: **2012 19th IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT)**. Eindhoven, Holanda: IEEE, 2012. p. 1–6.

MANJESHWAR, A.; AGRAWAL, D. P. Teen: a routing protocol for enhanced efficiency in wireless sensor networks. In: IEEE. San Francisco, Estados Unidos da América, 2001. p. 1–7.

MARQUES, J. K. V. An Ant Colony-based Mesh Routing Protocol for Maximizing Low Power and Lossy Networks Lifetime. In: **16th ACM International Symposium on Mobility Management and Wireless Access**. Montreal, Canadá: ACM, 2018. p. 1–7.

MARQUES, J. K. V. An Energy Efficient Mesh LLN Routing Protocol Based on Ant Colony Optimization. In: **16th International Conference on Industrial Informatics**. Porto, Portugal: IEEE, 2018. p. 1–6.

MARQUES, J. K. V. Mobility Aware RPL (MARPL): Mobility to RPL on Neighbor Variability. In: **14th International Conference on Green, Pervasive and Cloud Computing**. Uberlândia, Brasil: Em Publicação, 2019. p. 1–15.

OLSSON, J. 6LoWPAN Demystified. **Texas Instruments**, v. 13, p. 1–13, 2014.

OPPITZ, M.; TOMSU, P. Internet of things. **Inventing the Cloud Century**, Springer, p. 435–469, 2018.

OSTERLIND, F. **Mobility COOJA plugin**. 2010. <https://sourceforge.net/p/contikiproject/s/code/HEAD/tree/sics.se/mobility/>. Online; Acessado em: 01/03/2019.

OSTERLIND, F. et al. Cross-level sensor network simulation with cooja. In: **31st IEEE Conference on Local Computer Networks**. Tampa, Estados Unidos da América: IEEE, 2006. p. 641–648.

PARAMESWARAN, A. T. et al. Is RSSI a reliable parameter in sensor localization algorithms: An experimental study. In: IEEE. **Field failure data analysis workshop (F2DA09)**. Birmingham, Reino Unido, 2009. v. 5.

PAUL, P. V.; SARASWATHI, R. The Internet of Things — A comprehensive survey. In: **2017 International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC)**. Melmaruvathur, Índia: IEEE, 2017. p. 421–426.

RAULT, T.; BOUABDALLAH, A.; CHALLAL, Y. Energy efficiency in wireless sensor networks: A top-down survey. **Computer Networks**, Elsevier, v. 67, p. 104–122, 2014.

RFC 3561. **Ad hoc on-demand distance vector (AODV) routing**. [S.I.]: Internet Engineering Task Force, 2003. 1–37 p. <https://tools.ietf.org/html/rfc3561>. Online; Acessado em: 01/03/2019.

RFC 3626. **Optimized Link State Routing Protocol (OLSR)**. [S.I.]: Internet Engineering Task Force, 2003. 1–37 p. <https://tools.ietf.org/html/rfc3626>. Online; Acessado em: 01/03/2019.

RFC 4728. **The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4**. [S.I.]: Internet Engineering Task Force, 2007. 1–107 p. <https://tools.ietf.org/html/rfc4728>. Online; Acessado em: 01/03/2019.

RFC 4944. **Transmission of IPv6 packets over IEEE 802.15.4 networks**. [S.I.]: Internet Engineering Task Force, 2007. 1–30 p. <https://tools.ietf.org/html/rfc4944>. Online; Acessado em: 01/03/2019.

RFC 5673. **Industrial Routing Requirements in Low-Power and Lossy Networks**. [S.I.]: Internet Engineering Task Force, 2009. 1–27 p. <https://tools.ietf.org/html/rfc5673>. Online; Acessado em: 01/03/2019.

RFC 5826. **Home Automation Routing Requirements in Low-Power and Lossy Networks**. [S.I.]: Internet Engineering Task Force, 2010. 1–17 p. <https://tools.ietf.org/html/rfc5826>. Online; Acessado em: 01/03/2019.

RFC 5867. **Building automation routing requirements in low-power and lossy networks**. [S.I.]: Internet Engineering Task Force, 2010. 1–26 p. <https://tools.ietf.org/html/rfc5867>. Online; Acessado em: 01/03/2019.

RFC 6206. **The Trickle Algorithm**. [S.I.]: Internet Engineering Task Force, 2011. 1–13 p. <https://tools.ietf.org/html/rfc6206>. Online; Acessado em: 01/03/2019.

RFC 6550. **RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks**. [S.I.]: Internet Engineering Task Force, 2012. 1–157 p. <https://tools.ietf.org/html/rfc6550>. Online; Acessado em: 01/03/2019.

RFC 6551. **Routing metrics used for path calculation in low-power and lossy networks**. [S.I.]: Internet Engineering Task Force, 2012. 1–30 p. <https://tools.ietf.org/html/rfc6551>. Online; Acessado em: 01/03/2019.

RFC 6552. **Objective function zero for the routing protocol for low-power and lossy networks (RPL)**. [S.l.]: Internet Engineering Task Force, 2012. 1–14 p. <https://tools.ietf.org/html/rfc6552>. Online; Acessado em: 01/03/2019.

RFC 6719. **The Minimum Rank with Hysteresis Objective Function**. [S.l.]: Internet Engineering Task Force, 2012. 1–13 p. <https://tools.ietf.org/html/rfc6719>. Online; Acessado em: 01/03/2019.

SAVVIDES, A.; HAN, C.-C.; STRIVASTAVA, M. B. Dynamic fine-grained localization in ad-hoc networks of sensors. In: . Roma, Itália: ACM, 2001. p. 166–179.

SETHI, P.; SARANGI, S. R. Internet of things: architectures, protocols, and applications. **Journal of Electrical and Computer Engineering**, Hindawi, v. 2017, p. 1–26, 2017.

SHAPIRO, S.; WILK, M. An analysis of variance test for normality. **Biometrika**, v. 52, n. 3, p. 591–611, 1965.

STUDENT. The probable error of a mean. **Biometrika**, JSTOR, p. 1–25, 1908.

WILCOXON, F. Individual comparisons by ranking methods. **Biometrics bulletin**, JSTOR, v. 1, n. 6, p. 80–83, 1945.

YAQOOB, I. et al. Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges. **IEEE wireless communications**, IEEE, v. 24, n. 3, p. 10–16, 2017.

YASSEIN, M. B. et al. A new dynamic trickle algorithm for low power and lossy networks. **Engineering MIS (ICEMIS)**, IEEE, Agadir, Marrocos, p. 1–6, 2016.

YOUNIS, M. et al. Topology management techniques for tolerating node failures in wireless sensor networks: A survey. **Computer Networks**, Elsevier, v. 58, p. 254–283, 2014.

ZHAO, M. et al. A comprehensive study of RPL and P2P-RPL routing protocols: Implementation, challenges and opportunities. **Peer-to-Peer Networking and Applications**, Springer, v. 10, n. 5, p. 1232–1256, 2017.