



Universidade Federal do Amazonas
Faculdade de Tecnologia
Programa de Pós-Graduação em Engenharia Elétrica

Roteamento baseado em Entropia de Mobilidade para Sensores sem Fio conectados a Internet das Coisas

Vitor Palmeira Tavares

Manaus – Amazonas
Fevereiro de 2016

Vitor Palmeira Tavares

Roteamento baseado em Entropia de Mobilidade para Sensores sem Fio conectados a Internet das Coisas

Projeto de Qualificação apresentado ao Programa de Pós-Graduação em Engenharia Elétrica, como requisito parcial para obtenção do Título de Mestre em Engenharia Elétrica. Área de concentração: Automação e Controle.

Orientador: Celso Barbosa Carvalho

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

T231r Tavares, Vitor Palmeira
Roteamento baseado em Entropia de Mobilidade para Sensores
sem Fio conectados a Internet das Coisas / Vitor Palmeira Tavares.
2016
63 f.: il.; 31 cm.

Orientador: Celso Barbosa Carvalho
Dissertação (Mestrado em Engenharia Elétrica) - Universidade
Federal do Amazonas.

1. redes de sensores sem fio. 2. roteamento. 3. mobilidade. 4.
protocolo . I. Carvalho, Celso Barbosa II. Universidade Federal do
Amazonas III. Título

Vitor Palmeira Tavares

**Roteamento baseado em Entropia de Mobilidade para
Sensores sem Fio conectados a Internet das Coisas**

Banca Examinadora

Prof. Dr. Celso Barbosa Carvalho
Universidade Federal do Amazonas UFAM

Prof. Dr. André Luiz Duarte Cavalcante
Universidade Federal do Amazonas UFAM

Prof. Dr. José Ferreira de Rezende
Universidade Federal do Rio de Janeiro - UFRJ

Manaus – Amazonas

Fevereiro de 2016

À família.

Agradecimentos

Agradeço à Deus pela oportunidade de cursar o mestrado.

À minha família, pela paciência, compreensão e incentivo durante o período de estudos do mestrado.

À minha noiva, pela compreensão dos momentos ausentes e incentivo.

Ao meu orientador e amigo, Celso Carvalho, por todo o auxílio e incentivo durante o processo de pesquisa.

À todos os professores do programa de pós-graduação com quem tive a oportunidade de cursar matérias que me evoluíram em conhecimento.

Ao programa de pós-graduação da Universidade Federal da UFAM pela oportunidade.

Aos amigos Cláudio Henrique, Pedro Donádio, Walfredo Lucena, Franciso Januário e todos os outros que, de alguma forma, me ajudaram na realização deste trabalho.

“A persistência é o menor caminho do êxito”.

Chaplin

Resumo

Dispositivos eletrônicos conhecidos como dispositivos inteligentes são utilizados em redes de sensores sem fio de forma interativa e autônoma com o mundo físico. Estes dispositivos quando conectados a internet através destas redes tornam a internet mais ubíqua. As redes de sensores sem fio possuem protocolos específicos, como o protocolo de roteamento para redes de baixa potência e perdas (do inglês, *Routing Protocol for Low-Power and Lossy Networks - RPL*). O RPL é um protocolo de roteamento consolidado e foi desenvolvido especificamente para ambientes sem mobilidade. Várias propostas tem sido desenvolvidas para adequar o RPL à mobilidade para atender requisitos ou questões específicas. Este trabalho tem como objetivo propor adequações para que o RPL corresponda as necessidades da mobilidade em sensores sem fio de forma abrangente. A metodologia consiste na proposta de uma arquitetura baseada em entropia de mobilidade para ser integrada ao RPL que é posteriormente simulada extensivamente em um emulador para internet das coisas e redes de sensores sem fio. Esta arquitetura baseada em entropia de mobilidade considera o deslocamento dos nodos como fator de decisão para definir as rotas pelas quais os nodos transmitem e recebem pacotes. Os resultados experimentais mostram que a metodologia proposta pode ser eficaz e eficiente para reduzir perda de pacotes e latência em redes RPL com mobilidade.

Palavras-chave: redes de sensores sem fio, roteamento, mobilidade.

Abstract

Electronic devices known as smart devices are used in wireless sensor networks in an interactive and autonomous way with the physical world. These devices when connected to the internet through these networks become the internet ubiquitous. Wireless Sensor Networks have specific protocols, such as Routing Protocol for Low-Power and Lossy - RPL. The RPL is a consolidated routing protocol and was developed specifically for environments without mobility. Several proposals have been proposed to adapt RPL in order to achieve mobility requirements and specific issues. This work aims to propose adjustments to the RPL in order to fill the mobility requirements. A proposal of framework based on mobility entropy to be integrated into the RPL and then simulated extensively in an emulator for Internet of things and wireless sensor networks. The mobility entropy-based architecture considers the displacement of nodes as a deciding factor to define the routes by which the nodes transmit and receive packets. The experimental results show that the proposed method can be effectively and efficiently to reduce packet loss and latency in networks with RPL mobility.

Keywords: wireless sensor network, routing, mobility.

Índice

Índice de Figuras	xi
Índice de Tabelas	xiii
Abreviações	xiv
1 Introdução	1
1.1 Descrição do Problema	2
1.2 Motivação	3
1.3 Objetivos	4
1.4 Metodologia de Trabalho	4
1.5 Organização da Dissertação	4
2 Fundamentação Teórica	6
2.1 Internet das Coisas	6
2.2 Redes de Sensores Sem Fio	8
2.3 Protocolo de Roteamento para Redes de Baixa Potência e com Perdas	10
2.3.1 Função Objetiva	13
2.3.2 Rank	14
2.3.3 Objeto de Anúncio de Destino	15
2.3.4 Objeto de Informação de DODAG	16
2.3.5 Solicitação de Informação de DODAG	17
2.3.6 Construção do DODAG	17
2.3.7 Manutenção das Tabelas de Roteamento	20
2.3.8 Temporizador <i>Trickle</i>	21

2.3.9	Mecanismo de reparo	23
2.4	Métricas de Roteamento para Redes de Baixa Potência e com Perdas	24
2.4.1	A métrica LQL	24
2.4.2	A métrica ETX	24
2.5	Entropia de Mobilidade	25
3	Trabalhos Relacionados	27
3.1	Arquitetura do CoRPL	33
3.1.1	Controle de Mensagens	34
3.1.2	DIS modificado	34
3.1.3	Tabela de vizinhos	34
3.1.4	O mecanismo Corona	35
3.1.5	DAG raiz e RPL móvel	35
3.1.6	Mecanismo de recuperação de percurso	35
4	Mecanismo proposto	37
4.1	Entropia aplicada ao CoRPL	38
5	Avaliação de Desempenho	40
5.1	Simulador	41
5.2	Cenários de simulação	41
5.3	Resultados	42
5.3.1	Variação em taxas de transmissão	43
5.3.2	Variação de nodos raiz	45
5.3.3	Variação de velocidade dos nodos	48
5.3.4	Variação no tamanho da tabela de vizinhos	48
5.4	Resumo	50
6	Conclusão	54
6.1	Trabalhos Futuros	55
	Referências Bibliográficas	56

Índice de Figuras

2.1	Resumo de possíveis aplicações da IoT em redes de sensores sem fio	9
2.2	Instancia RPL com múltiplos DODAGs	12
2.3	Exemplo de construção de um DODAG em uma instancia RPL	13
2.4	Exemplo de DODAG com dois OF em uma instância RPL	14
2.5	Exemplo de formação de um DODAG - 1º Passo	19
2.6	Exemplo de formação de um DODAG - 2º Passo	19
2.7	Exemplo de formação de um DODAG - 3º Passo	19
2.8	Exemplo de formação de um DODAG - 4º Passo	20
2.9	Exemplo de mensagem DAO no RPL	21
3.1	Fluxograma da escolha de nodo <i>Parent</i> do ME-RPL	29
3.2	Fluxograma do mecanismo <i>Sink Discovery</i>	31
3.3	Exemplo de uma arquitetura da rede CoRPL com quatro DODAGS	33
5.1	Simulação de transferencia de dados em rede RPL com o emulador COOJA	41
5.2	Comparação de perda de pacotes entre o CoRPL e o CoRPL com entropia de mobilidade com variação da taxa de transmissão no modelo randômico de mobilidade	43
5.3	Comparação do atraso médio entre o CoRPL e o CoRPL com entropia de mobilidade com variação da taxa de transmissão de pacotes no modelo randômico de mobilidade	44
5.4	Comparação de perda de pacotes entre o CoRPL e o CoRPL com entropia de mobilidade com variação da taxa de transmissão no modelo Manhattan de mobilidade	44
5.5	Comparação do atraso médio entre o CoRPL e o CoRPL com entropia de mobilidade com variação da taxa de transmissão de pacotes no modelo Manhattan de mobilidade	45

5.6	Comparação de perda de pacotes entre o CoRPL e o CoRPL com entropia de mobilidade com variação de quantidade de nodo raiz no modelo randômico de mobilidade	46
5.7	Comparação do atraso médio entre o CoRPL e o CoRPL com entropia de mobilidade com variação de quantidade de nodo raiz no modelo randômico de mobilidade	46
5.8	Comparação de perda de pacotes entre o CoRPL e o CoRPL com entropia de mobilidade com variação de quantidade de nodo raiz no modelo Manhattan de mobilidade	47
5.9	Comparação do atraso médio entre o CoRPL e o CoRPL com entropia de mobilidade com variação de quantidade de nodo raiz no modelo Manhattan de mobilidade	47
5.10	Comparação de perda de pacotes entre o CoRPL e o CoRPL com entropia de mobilidade com variação de velocidade no modelo randômico de mobilidade	48
5.11	Comparação do atraso médio entre o CoRPL e o CoRPL com entropia de mobilidade com variação de velocidade no modelo randômico de mobilidade	49
5.12	Comparação de perda de pacotes entre o CoRPL e o CoRPL com entropia de mobilidade com variação de velocidade no modelo Manhattan de mobilidade	49
5.13	Comparação do atraso médio entre o CoRPL e o CoRPL com entropia de mobilidade com variação de velocidade no modelo Manhattan de mobilidade	50
5.14	Comparação de perda de pacotes entre o CoRPL e o CoRPL com entropia de mobilidade com variação do tamanho de tabela de vizinhança no modelo randômico de mobilidade	51
5.15	Comparação do atraso médio entre o CoRPL e o CoRPL com entropia de mobilidade com variação do tamanho de tabela de vizinhança no modelo randômico de mobilidade	51
5.16	Comparação de perda de pacotes entre o CoRPL e o CoRPL com entropia de mobilidade com variação do tamanho de tabela de vizinhança no modelo Manhattan de mobilidade	52
5.17	Comparação do atraso médio entre o CoRPL e o CoRPL com entropia de mobilidade com variação do tamanho de tabela de vizinhança no modelo Manhattan de mobilidade	52

Índice de Tabelas

3.1	Tabela de nodos vizinhos CoRPL. [9]	34
3.2	Algoritmo para calcular o valor de Corona ID. [9]	35
4.1	Tabela de vizinhança de nodo RPL.	38
4.2	Algoritmo de escolha de nodos pai com entropia	39
5.1	Tabela de parâmetros de configuração das simulações.	42

Abreviações

DAO - *Destination Advertisement Object*

DIS - *DODAG Information Solicitation*

DIO - *DODAG Information Object*

DMR - *DAG-based Multipath Routing for Mobile Sensor Networks*

DODAG - *Destination Oriented Directed Acyclic graphs*

IETF - *Internet Engineering Task Force*

IoT - *Internet of Things*

LLN - *Low Power and Lossy Networks*

LQI - *Link Quality Level Reliability Metric*

OF - *Object Function*

ROLL - *Routing Over Low-power and Lossy networks*

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

VANET - *Vehicular Adhod Network*

Capítulo 1

Introdução

A internet das coisas (do inglês, *Internet of Things* - IoT) é uma tecnologia emergente capaz de proporcionar a interconectividade de uma variedade de objetos que através de um esquema de endereçamento único, dispositivos como sensores, atuadores, telefones móveis, veículos, RFID¹ e outros poderão interagir entre si [1].

A Internet irá tornar-se uma rede muito mais ubíqua devido a integração com objetos através de sistemas embarcados, resultando em uma rede altamente distribuída [2]. Em razão da grande quantidade de tecnologia que a internet das coisas abrange oportunidades estão surgindo em novas aplicações com a proposta de melhorar a qualidade de vida dos seres humanos. Tecnologias emergentes como sensores sem fio já ocupam lugares nas residências, escritórios e fábricas levando novas fontes de informação, controle e conveniência a vida pessoal e profissional das pessoas. De fato os sensores sem fio são um dos elementos mais importantes na internet das coisas. Conectando as redes de sensores sem fio (do inglês, *Wireless Sensor Network* - WSN) à internet das coisas sistemas inteiros podem prover serviços comuns de forma colaborativa [3].

Uma rede de sensores sem fio é uma coleção de nodos operando de forma colaborativa com propósito de capturar dados de eventos como temperatura, umidade, entre outros. Neste tipo de rede, dados são capturados em dispositivos sem fio que são roteados com destino a um nodo principal que normalmente se conecta com outra rede, podendo inclusive se conectar à Internet ou então a outros dispositivos usando um protocolo diferente [4].

¹do inglês *Radio Frequency Identification*, é uma tecnologia que possibilita a identificação e rastreamento de produtos com o uso de transpônder, leitores com antena e um computador

As limitações dos dispositivos que compõem as redes de sensores sem fio tem um impacto direto no protocolo de roteamento utilizado pela rede, por isso protocolos específicos para este tipo de rede são necessários. O protocolo de roteamento para redes de baixa potência e com perdas (do inglês, *Routing Protocol for Low-Power and Lossy Networks* - RPL) foi desenvolvido especificamente para rotear dados em redes de baixa potência e com perdas (do inglês, *Low Power and Lossy Networks* - LLN). As LLNs são caracterizadas pelo baixo consumo de energia dos dispositivos que geralmente são alimentados por fontes de energia limitadas, apresentam baixas taxas de transmissão de dados e perda de dados elevadas durante a transmissão. Redes de sensores sem fio é um tipo de rede que se enquadra em tais características [1].

O RPL foi proposto pela força tarefa de engenharia para a internet (do inglês, *Internet Engineering Task Force* - IETF), uma comunidade internacional composta de pesquisadores, agências, fabricantes e fornecedores. O propósito inicial da IETF era estabelecer um conjunto de requisitos para redes LLNs e identificar protocolos de roteamento existentes que atendessem tais requisitos, mas logo ficou claro a necessidade de um protocolo específico para este tipo de rede. O IETF então formou um novo grupo de trabalho denominado roteamento sobre redes de baixa potência e com perdas (do inglês, *Routing Over Low-power and Lossy networks* - ROLL), com o propósito de projetar o que se tornaria o RPL.

As redes de sensores sem fio do tipo LLN podem ser estáticas, onde os nodos que compõem a rede são todos estáticos e sem movimentação, ou podem ser dinâmicas, de forma que apenas uma parte dos nodos, ou sua totalidade sejam móveis. Vários são os cenários que exigem suporte à mobilidade. Em [5] o RPL é utilizado em conjunto com redes veiculares (do inglês *Vehicular networking* - VANET) visando o acesso a internet em veículos públicos e particulares. Em [6] é proposto o uso de sensores sem fio para coletar dados sensoriais em robôs móveis. Porém a mobilidade ainda é um dos pontos falhos do protocolo RPL, cuja concepção não considerou as mudanças de topologias decorrentes de movimentação dos nodos durante o roteamento de dados [7].

1.1 Descrição do Problema

O RPL proposto pelo IETF atende as especificações e requisitos de redes de baixa potência e com perdas somente quando estas são estáticas. Em redes com mobilidade dos nodos o RPL apresenta dificuldades para gerenciar rotas devido o tempo de reação necessário para que

o protocolo identifique e intervenha nas alterações de topologia causadas pela mobilidade [8], [9], [7]. Alterações contínuas na topologia de redes de sensores sem fio que utilizam o protocolo RPL, causam degradações de enlaces constantes, excessiva perda de dados e *loops* de roteamento ².

1.2 Motivação

O RPL foi projetado para ser adaptável e flexível afim de suportar diferentes tipos de métricas e restrições como requisitos para construções de rotas. Além disso, trata-se de um protocolo passivo, uma característica fundamental para tecnologias que trabalham com recursos limitados. As redes de sensores sem fio devem fornecer serviços confiáveis, de qualidade, com pouco atraso e energeticamente eficientes. O RPL atende tais requisitos quando aplicado em redes estáticas, porém ainda são necessárias melhorias em sua arquitetura para que seu desempenho em redes com mobilidades seja robusto [10].

Novas abordagens estão sendo propostas para reduzir as limitações da mobilidade em aplicações específicas no RPL, como otimizar consumo de energia da rede, diminuir a quantidade de perda de pacotes, reduzir o tempo de processamento dos dados dentro da rede, entre outros. Em [9] é apresentado o mecanismo Corona, ou CoRPL. Este mecanismo é capaz de adaptar o RPL a mobilidade de forma mais abrangente, permitindo seu uso em aplicações mais amplas. Os autores propõem a divisão da área total de alcance da rede RPL em áreas menores que em conjunto com outros recursos apresentados aprimoram o gerenciamento dos nodos com mobilidade. O processo de definição de rotas é realizado utilizando o CoRPL associado a métricas de qualidade, entretanto informações relevantes ao posicionamento dos nodos não são consideradas para escolha das rotas.

Em [11] é definido que características variáveis correspondentes a mobilidade de uma rede com nodos móveis podem ser utilizada para calcular a entropia que representa o grau de incerteza e a medida de desordem do sistema. Rotas onde múltiplos percursos estão disponíveis podem ser definidas conforme o grau de entropia.

Assim, a relevância deste estudo seria o de propor uma solução para a problemática da mobilidade no RPL utilizando o mecanismo do CoRPL em conjunto com a entropia de mobilidade para definir as rotas que o RPL estabelece para comunicação entre os nodos.

²Condição em que roteadores transmitem pacotes entre si evitando que o destinatário receba estes pacotes

1.3 Objetivos

O objetivo deste trabalho é atenuar a perda de pacotes e latência que ocorre em transmissões de pacotes em redes de baixa potência, com perdas e com mobilidade. Para isso, é proposto utilizar o CoRPL com um mecanismo que utiliza o grau de entropia de mobilidade para estabelecer as rotas utilizadas para roteamento de pacotes no RPL.

Os objetivos específicos do trabalho são:

- a) Desenvolver os algoritmos de entropia de mobilidade utilizando como base a o emulador de redes de sensores sem fio COOJA.
- b) Avaliar o mecanismo proposto por meio da comparação com o mecanismo CoRPL.

1.4 Metodologia de Trabalho

Como metodologia a ser adotada neste trabalho, serão realizadas as seguintes atividades:

- a) Pesquisa bibliográfica sobre redes de sensores sem fio, protocolo RPL e propostas desenvolvidas para integrar a mobilidade ao RPL.
- b) Pesquisa bibliográfica para a realização de estudos dos fundamentos e arquiteturas necessários para o desenvolvimento com o simulador COOJA.
- c) Definição e desenvolvimento da arquitetura de entropia de mobilidade a ser integrada com o RPL e CoRPL
- d) Realizar simulações com a arquitetura proposta e com a arquitetura do CoRPL e apresentar uma análise dos resultados obtidos.

1.5 Organização da Dissertação

Este capítulo apresenta uma breve introdução sobre o trabalho, os problemas que motivaram a pesquisa e os objetivos a serem alcançados

No Capítulo 2, *Fundamentação Teórica*, é apresentado uma introdução sobre a internet das coisas. Em seguida é apresentado um estudo teórico sobre o protocolo de roteamento de

redes de baixa potência e elevadas perdas, o RPL e as métricas desenvolvidas especificamente para este protocolo. Um estudo sobre entropia de mobilidade também é realizado neste capítulo.

Em *Trabalhos Relacionados*, Capítulo 3, é apresentado um resumo de trabalhos que utilizam técnicas para adaptar a mobilidade ao RPL.

Em *Mecanismo Proposto*, Capítulo 4, é apresentada a arquitetura que combina o mecanismo do CoRPL com a entropia de mobilidade para seleção de nodos pai com propósito de determinar as rotas mais estáveis para transferência de pacotes.

No Capítulo 5, *Avaliação de Desempenho*, são apresentados os cenários e os parâmetros necessários para realizar a avaliação da proposta, os resultados são expostos e averiguados.

No Capítulo 6, *Conclusão*, são expostas as contribuições do trabalho e as sugestões para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

2.1 Internet das Coisas

A Internet das Coisas, possui diversas definições descritas por pesquisadores e comunidades de pesquisas. Em [1], a ideia básica que conceitua a Internet das Coisas é a presença pervasiva ao nosso redor de uma variedade de objetos ou coisas que através de um esquema de endereçamento único sejam capazes de interagir entre si e através da Internet com um propósito ou objetivo em comum. Em [12], a Internet das coisas é definida como uma revolução tecnológica que representa o futuro da computação e da comunicação. Esta revolução viabilizará a comunicação de dispositivos inteligentes com a Internet.

Um objeto inteligente é um equipamento eletrônico composto de um sensor ou atuador, um microprocessador, um dispositivo de comunicação e uma fonte de energia. O sensor ou atuador provê ao objeto inteligente a habilidade de interagir com o mundo físico. O microprocessador possibilita transformar a informação capturada pelos sensores, ainda que com velocidade e complexidades limitados. O dispositivo de comunicação permite envio das leituras de dados dos sensores para o mundo afora e o recebimento de dados de outros objetos inteligente. O tamanho de um objeto inteligente geralmente não excede alguns centímetros cúbicos para que sejam incorporados em outros dispositivos de uso cotidiano. Estes objetos interagem com o mundo físico obtendo informação através de seus sensores e respondendo ao mundo físico através de seus atuadores. Seus sensores captam propriedades físicas que podem variar de propriedades mais simples como faixa de luz, temperatura, umidade do ar, ou propriedades mais complexas como detectar a poluição do ar, a presença de um carro ou determinado nível de

desgaste em uma máquina industrial [4]. Os atuadores podem interagir com o mundo físico de formas simples ao realizar o desligamento de uma lâmpada LED, ou de formas mais elaboradas, como por exemplo, ao alterar a temperatura de um determinado ambiente dentro de uma residência ou edifício. Porém o diferencial dos objetos inteligentes é a comunicação entre dispositivos, desta forma, objetos são capazes de tomar decisões com base em ações realizadas por outros objetos inteligentes.

A tendência é que estes dispositivos diminuam cada vez mais de tamanho e seus custos sejam reduzidos com o tempo. Desta forma, a inserção desta tecnologia na indústria, medicina, nas residências e em outras áreas tende a crescer, afetando diretamente o estilo e a qualidade de vida humana. Segundo o Conselho Nacional de Inteligência da América¹ (do inglês *US National Intelligence Council*), no ano de 2025 nodos² conectados à Internet estarão presentes em todas as coisas, pacotes de comidas, móveis, documentos, roupas, entre outros. Estas novas possibilidades criam novas expectativas sobre desafios que ainda precisam ser vencidos para que a total operabilidade dos dispositivos inteligentes seja alcançada. Para que os dispositivos interconectados a Internet funcione com um alto grau de inteligência capacitando sua adaptação e comportamento autônomo algumas características da Internet das Coisas, como recursos escassos em termos computacionais e eficiência energética, segurança da informação, escalabilidade ainda precisam evoluir [1]

Avanços tecnológicos recentes possibilitaram o surgimento da Internet das Coisas, tais como redes de sensores sem fio, comunicação móvel e computação ubíqua. Porém ainda existem desafios a serem superados para alavancar a disseminação dessa tecnologia, principalmente com relação ao desenvolvimento de aplicações e à alta heterogeneidade decorrente da diversidade de tecnologias de hardware e software desse ambiente que ainda não está totalmente padronizada com protocolos de comunicações que permitam a interação entre as diversas tecnologias que envolvem a Internet das Coisas e seus fabricantes.

¹Conselho de Inteligência Americano cujo objetivo é elaborar planejamentos estratégicos para médio e longo prazo

²Elemento computacional com capacidade de processamento, memória e interface de comunicação sem fio

2.2 Redes de Sensores Sem Fio

Uma estrutura computacional importante para a composição da internet das coisas são as redes sensores sem fio (do inglês *Wireless Sensor Networks*). Os benefícios da integração de redes de sensores sem fio com a internet das coisas vão além do simples acesso remoto a este tipo de rede. Como em um sistema de informação heterogêneo, será possível colaboração entre os dois sistemas em prol de um serviço comum. Uma rede de sensores sem fio pode ser totalmente autônoma enquanto executa tarefas colaborativas definidas para a rede. O uso de algum dispositivo que seja integrado com a internet das coisas somente é necessário quando surge a necessidade de interação com o mundo externo a rede [3]. Este tipo de integração de fato é matéria de pesquisa de companhias como a IBM, que desenvolve o *Smart Planet*, uma iniciativa que considera sensores sem fio como o pilar fundamental para gerenciamento de sistemas de fornecimento de água para cidades inteligentes [13]. Ou o Sistema Nervoso Central da Terra da HP Labs (do inglês *HP's Central Nervous System for the Earth - (CeNSE)*), que pretende desenvolver uma rede de bilhões de sensores sem fio ao redor do mundo com capacidade de captar cheiros, imagens, áudios entre outras coisas e enviar estes dados para poderosos servidores para processamento e análise [14]. Porém as tecnologias necessárias para que ocorra esta total integração ainda estão em desenvolvimento, como é o caso do 6LopwPan, o protocolo RPL, entre outros, definidos pelo IETF.

Conforme [15, 16], as redes de sensores sem fio podem ser empregadas nas seguintes áreas:

Ambiente

Rastreamento e monitoramento de animais e insetos, monitoramento da atmosfera e solo terrestre, detecção de produtos químicos, biológicos, incêndios florestais e enchentes

Saúde

Rastreamento e monitoramento de pacientes e médicos, auxílio no diagnóstico de doenças e administração de drogas, monitoramento fisiológico como frequência cardíaca, índice de colesterol e glicose, entre outros.

Residência

Monitorando dados do cotidiano e através de atuadores automatizando atividades corriqueiras realizadas até então manualmente. Sensores e atuadores acoplados em aparelhos domésticos possibilitam que o usuário residencial controle remotamente seus equipamentos.

Indústria

Monitoramento de produção, estrutura, máquinas, sensores de pressão, temperatura e de outros parâmetros para identificação de vazamentos e aquecimentos. Rastreamento e monitoramento de inventários.

Militar

Detecção de movimentação inimiga, ataques biológicos, químicos e nucleares. Monitoramento e rastreamento de tropas e veículos equipados com sensores. Apoio tático em terrenos desconhecidos

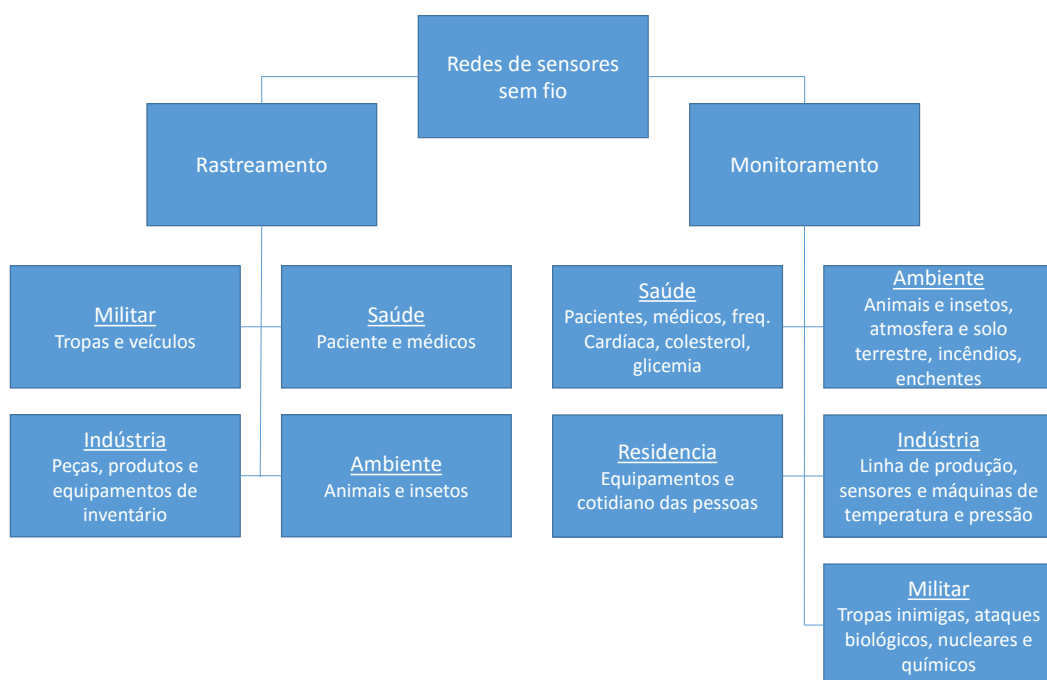


Figura 2.1: Resumo de possíveis aplicações da IoT em redes de sensores sem fio.

Fonte: YICK, et.al. (2008, p. 2296) – com adaptações.

A imagem 2.1 ilustra um resumo das possíveis áreas em que a internet das coisas pode ser utilizada juntamente com redes de sensores sem fio conforme apresentado em [15, 16].

2.3 Protocolo de Roteamento para Redes de Baixa Potência e com Perdas

O protocolo de roteamento para redes de baixa potência e com perdas, ou RPL está descrito no documento RFC 6550 e foi desenvolvido com a finalidade de atender aos requisitos de um tipo específico de redes de sensores sem fio, as redes de de baixa potência e com perdas, ou LLN. Os dispositivos conectados a este tipo de rede possuem limitações de tamanho, largura de banda, e geralmente são alimentados por baterias. Por estes motivos os protocolos desenvolvidos para redes LLNs devem otimizar ao máximo a utilização dos recursos destes dispositivos. Em resumo, as LLNs devem atender aos seguintes critérios descritos em [17, 7]:

- As LLNs devem priorizar o baixo consumo de energia dos dispositivos que integram a rede. Para isso são desenvolvidos protocolos de comunicação reativos onde os nodos realizam um processo de descoberta de outros nodos para determinar possíveis rotas somente quando necessário. Estas informações são armazenadas em tabelas internas e atualizadas de tempos em tempos.
- Geralmente as LLNs são implementadas sobre camadas de enlaces com restrição de quadros, logo as LLNs devem ser adaptadas para estes tipos de camadas.
- Os protocolos de roteamento de LLNs devem oferecer suporte a três tipos de padrões de dados, são eles tráfego ponto a ponto, multi-ponto a multi-ponto e ponto a multi-ponto. O objetivo deste suporte é proporcionar um roteamento eficiente para padrões de tráfego em árvores em cenários de elevadas perdas de dados e baixo consumo de energia.

Redes baseadas em IPs, de forma geral possuem um padrão de ocorrência de erro relativamente baixo, o perfil destas redes apresenta uma distribuição uniforme e os protocolos de roteamento projetados para atender este tipo de rede reagem rapidamente ao detectar falhas no enlace, evitando grandes oscilações. Este padrão de comportamento conduziu a criação de mecanismos que rapidamente recriavam rotas com o auxílio de percursos de *backup* logo que uma falha no enlace é identificada, com o propósito de minimizar a interrupção de tráfego de dados. Em redes LLNs o comportamento é diferente [4], neste tipo de rede as características principais são o baixo consumo de energia e elevadas perdas de pacotes. As LLNs são com-

posta por grandes quantidades de nodos com limitações e os enlaces são instáveis e voláteis e normalmente suportam baixas taxas de transmissão de dados.

Na prática, assim que um nodo RPL obtém um endereço IPv6 global ele une-se a um Grafo Acíclico Dirigido e Orientado a Destino (do inglês *Destination Oriented Directed Acyclic graphs* - DODAG). Uma instância RPL é composta de um ou vários DODAGs. Na Figura 3 que demonstra uma instancia RPL é possível visualizar três grafos, cada grafo é um DODAG e uma mesma instância RPL pode conter mais de um DODAG. A imagem demonstra a comunicação entre os pontos que é feita subindo a estrutura do grafo ou descendo a mesma até alcançar os nodos localizados nas extremidades conhecidos como nodos folhas [18].

Devido à escassez de recursos em LLNs, o controle de tráfego da rede deve ser minimizado ao máximo com propósito de consumir a menor quantidade de banda possível. O uso de um mecanismo de checagem ágil torna-se necessário e o controle de tráfego deve ser reduzido a medida que a topologia da rede se estabiliza. Devido à natureza de recursos escassos dos nodos, o estado em que é mantido um nodo deve ser o mais breve possível [19].

O RPL constrói um DODAG, onde percursos são construídos para cada nodo da rede com destino ao nodo de sincronismo ou raiz dentro do grafo e o nodo raiz é responsável por conectar a rede RPL à uma rede externa, como a internet por exemplo. Os DODAGs oferecem um percurso de rota alternativo, sempre que possível, e este é um recurso necessário a qualquer tipo de LLN. No RPL pode existir um ou mais nodo raiz que são configurados pelo administrador da rede. O mecanismo de descoberta de nodos dentro do RPL é baseado na troca de mensagens ICMPv6³.

O RPL define dois novos tipos de mensagens ICMPv6, o Objeto de informação de DODAG (do inglês *DODAG Information Object* - DIO) e o Objeto de Anúncio de Destino (do inglês *Destination Advertisement Object* - DAO). As mensagens DIO são enviadas pelos nodos para passar informações referentes ao gráfico DODAG a outros nodos. Um DODAG contém a informação necessária para a construção da árvore de comunicação dos nodos pertencentes ao RPL, inclusive dados sobre o nodo no instante em que ele passa a agregar o DODAG. Quando um nodo recebe uma mensagem DIO de mais de um nodo vizinho várias regras definidas no RPL são consideradas pelo nodo para decidir qual DODAG irá agregar, ou se irá agregar. Uma vez que o nodo passa a fazer parte do DODAG, este já tem conhecimento de uma rota que leve

³Protocolo de ICMP baseado no IPv6 utilizado para reportar erros no processamentos de pacotes de dados [20]

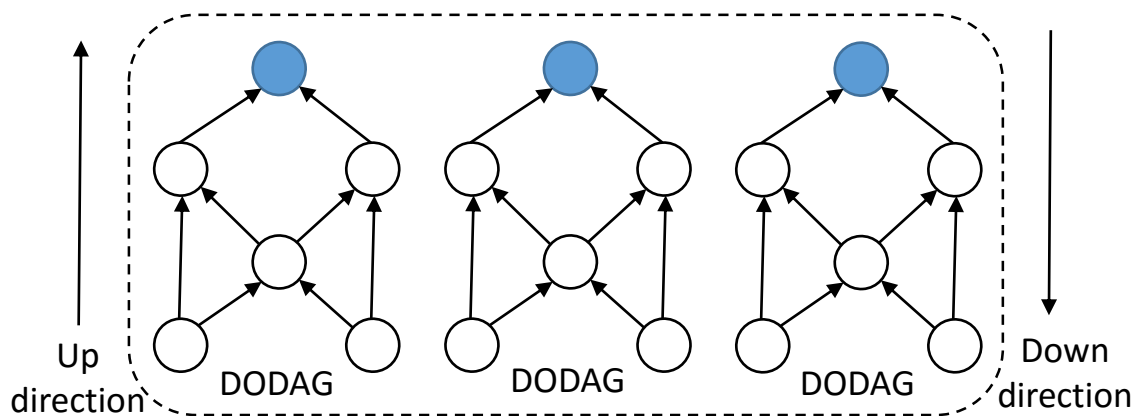


Figura 2.2: Instancia RPL com múltiplos DODAGs
 Fonte: VASSEUR, et.al. 2010 – com adaptações.

ao nodo raiz [19].

O RPL utiliza as terminologias *Up Direction* e *Down Direction* para descrever a direção em que o tráfego de dados flui no RPL. *Up Direction* significa que os dados partem de um nodo em direção ao raiz e *Down Direction* o sentido contrário, os dados fluem do raiz à outro nodo interno da rede. A figura 2.2 ilustra as direções que fluem os pacotes de dados dentro de uma instância RPL com múltiplos DODAGs. Em [19], os nodos também são classificados da seguinte maneira:

- Nodos pais (do inglês, *Parent*), são os nodos sucessores imediatos no sentido *Up Direction* de um nodo de referência, ou seja, um nível acima ao nodo em questão.
- Nodos filhos (do inglês, *Children*), são os nodos sucessores imediatos no sentido *Down Direction* de um nodo de referência, ou seja, um nível abaixo ao nodo em questão.
- Nodos irmãos (do inglês, *Sibling*), são os nodos que se encontram no mesmo nível que outro nodo de referência. Esta comparação é feita utilizando uma informação propagada pelo DODAG, o *rank* dos nodos, que neste caso, são iguais.

Na Figura 3.1 é possível visualizar um exemplo de um DODAG em construção. No exemplo, o nodo 13 é pai do nodo 24. Os nodos 22 e 23 são irmãos pois possuem o mesmo *rank* e existe um enlace estabelecido entre os dois. Os nodos 43 e 44 são filhos do nodo 33.

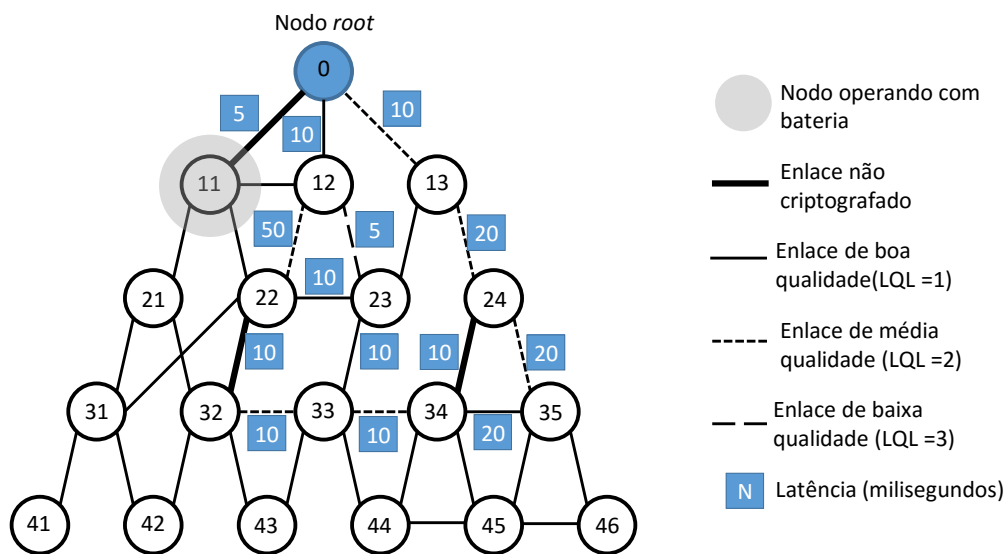


Figura 2.3: Exemplo de construção de um DODAG em uma instancia RPL
 Fonte: VASSEUR, et.al. (2010, p. 260) – com adaptações.

2.3.1 Função Objetiva

A Função Objetiva (do inglês *Object Function* - OF) define como os nodos RPL selecionam e otimizam as rotas dentro da Instancia RPL. Isso é feito através de métricas e restrições que são adicionadas à mensagem DIO e repassadas aos nodos durante a construção do DODAG. O OF é identificado por um código, o OCP (do inglês *Objective Code Point*), dentro das opções configuráveis da mensagem DIO. Cada DODAG é composto de uma função objetiva, logo podem existir vários OFs dentro de uma mesma LLN. O OF também é responsável por passar a informação de como o nodo deve calcular o *rank* que é uma informação gerada a partir do nodo raiz e passada dentro do OF na construção do DODAG para evitar *loop* de dados seleção de nodos e para seleção de nodos pais e irmãos [19].

Supondo que duas funções objetivas (OF1 e OF2) sejam criadas dentro de uma única instância RPL no exemplo em 2.2 e o nodo 34 do exemplo tenha que definir um percurso para cada OF recebido. Para o OF1 são definidos as seguintes métricas e restrições:

- Utilizar o LQL como métrica global, ou seja, dar preferencia a menor soma acumulada entre os saltos.
- Evitar sempre que possível enlaces com baixa e média qualidade.
- Evitar enlaces não criptografados.

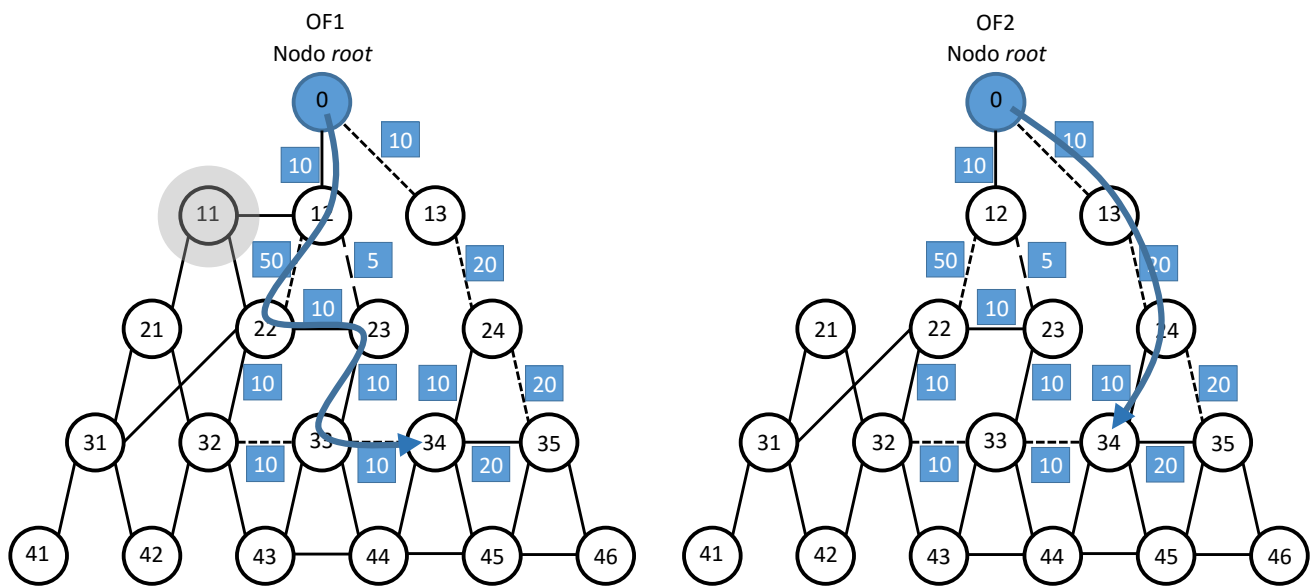


Figura 2.4: Exemplo de DODAG com dois OF em uma instância RPL
 Fonte: VASSEUR, et.al. – com adaptações.

O percurso resultante destas métricas e restrições é a seta azul do OF1 na figura 2.4. Durante a criação da rota, dois possíveis percursos parecem satisfazer melhor esta OF. São elas 34-35-24-13-0 e 34-33-23-22-12-1. Como o OF prioriza os enlaces de boa qualidade o percurso 34-33-23-22-12-1 é escolhido devido ao menor número de enlaces de baixa e média qualidade.

Para o OF2 são definidas as seguintes métricas e restrições:

- Dar preferência a rotas com menor latência acumulada.
- Evitar enlaces com baixa qualidade.
- Evitar nodos que operam com baterias.

O percurso resultante para o nodo 34 utilizando o OF2 é exemplificado em 2.4 - OF2. Neste caso algumas rotas nem foram consideradas por terem um salto no nodo 11 que opera com bateria. Outro salto desconsiderado é entre os nodos 12 e 23 pois o enlace possui baixa qualidade. O percurso que melhor se encaixa neste OF é 34-24-13-0.

2.3.2 Rank

O *rank* é um número escalar que compõe o objeto de função para construção do DODAG, e seu valor inicial é definido pelo nodo raiz. O *rank* é propagado pelo OF para todos os nodos, e

a medida que os nodos passam a agregar o DODAG, o valor do *rank* é incrementado. O *rank* é usado no momento da escolha do nodo antecessor para evitar *loops* de dados na rede e também para evitar enlaces de grandes distâncias entre os nodos [4].

2.3.3 Objeto de Anúncio de Destino

A mensagem DAO é um objeto composto de um conjunto de variáveis descritas em [19]. Estas variáveis são utilizadas pelo RPL através do DODAG sempre no *Up Direction* com propósito de popular as tabelas de nodos antecessores, com isso permitindo o tráfego P2P⁴. As principais variáveis que compõem o corpo da mensagem DAO são:

- RPLInstanceID: Sequência de 8 bits que indica a instancia atual do DODAG, conforme informado pela mensagem DIO ao nodo.
- Bit K: A variável K é uma flag utilizada para informar que houve o recebimento do DAO por outro nodo.
- Bit D: A variável D indica que o ID de identificação do DODAG está presente. Este bit somente é configurado quando uma instancia do RPLInstanceID é usado.
- Flags: Variável de 6 bits utilizados para configuração do nodo.
- Reserved: variável não utilizada que deve ser inicializada como zero.
- DAOSequence: Variável incrementada quando a mensagem DAO é enviada e um aviso de recebimento (DAO-ACK) é retornado.
- DODAGID: Variável do tipo inteiro de 128-bits que armazena um valor criado no raiz que identifica unicamente o DODAG. O uso desta variável está associado ao bit D, somente quando o bit D é configurado o valor desta variável pode ser preenchida. Todavia, quando um RPLInstance global estiver em uso, esta variável não é utilizada pelo nodo.

As mensagens DAO são inicializadas sempre que ocorre alguma atualização no nodo pai ou então quando o nodo recebe uma mensagem DAO de seus nodos sucessores. Porém a mensagem DAO não é propagada imediatamente após a ocorrência destes eventos, em [19] é

⁴Protocolo descentralizador onde cada computador conectado realiza funções de servidor e cliente simultaneamente

definido a constante *DelayDAO* com valor fixado em um segundo. Qualquer um dos eventos que inicializa a mensagem DAO também inicia uma contagem regressiva, e somente após um segundo a mensagem DAO é enviada. Este recurso é usado para obrigar o nodo a aguardar por mais tempo mensagens de nodos sucessores antes de enviar a mensagem DAO ao seu nodo pai.

2.3.4 Objeto de Informação de DODAG

As mensagens DIO são propagadas pelos nodos para informar o DODAG e suas características a outros nodos, mais especificamente as mensagens DIO são usadas para formação e manutenção do DODAG. Segundo a definição em [19], o corpo da mensagem DIO é composto das seguintes variáveis:

- **Grounded:** Variável que sinaliza se o nodo raiz está aterrado, ou seja, conectado a alguma outra rede externa à LLN (conexão com a Internet, rede privada, entre outros).
- **Destination Advertisement Trigger:** Bit T, quando este bit é utilizado, sua configuração é feita pelo nodo raiz e uma atualização completa das rotas é feita no DODAG no sentido *Down Direction* (reparo global).
- **Destination Advertisement Stored(S):** O bit S é usado para indicar que um nodo antecessor está armazenando em sua tabela de roteamento localizações de outros nodos que foram adquiridas através de mensagens DAO anteriormente propagadas.
- **Destination Advertisement Supported (Flag A):** Esta flag é configurada quando o nodo raiz oferece suporte a uma coleção de prefixos para anúncios e habilita estes anúncios para o DODAG.
- **DODAGPreference (Prf):** A variável Prf é um campo de 3 bits configurável pelo nodo raiz para reportar preferência por algum DODAG, no caso de existirem múltiplos DODAGs. Este campo pode ser utilizado pelo DODAG com propósito de se tornar mais atrativo aos nós.
- **DODAGSequenceNumber:** Esta variável é um número sequencial que representa a iteração do DODAG e é exclusivamente controlado pelo nodo raiz. A partir desta variável os nodos podem distinguir em qual versão do DODAG se encontram conectados.

- **RPLInstanceID:** Variável utilizada para identificar a instancia do DODAG disponibilizada a partir do nodo raiz. Uma LLN pode conter múltiplas DODAGs simultaneamente e esta variável identifica ao nodo qual DODAG ele pertence. Cada instancia DODAG possui um **DODAGSequenceNumber**.
- **Destination Advertisement Trigger Sequence Number(DTSN):** Variável do tipo inteiro de 8bits configurada pelo nodo que envia a mensagem DIO que é usada por um processo que ocorre nos nodos para manter as rotas do sentido *Down Direction*.
- **DODAGID:** Variável do tipo inteiro de 128 bits que identifica a unicidade do DODAG.
- **DODAG Rank:** A variável determina a posição relativa do nodo dentro do DODAG e é usado para evitar *loops* dentro da rede. O rank é computado conforme a função objetiva é configurada.

2.3.5 Solicitação de Informação de DODAG

Solicitação de Informação de DODAG (do inglês *DODAG Information Solicitation - DIS*) é um tipo de mensagem que pode ser enviada pelos nodos para solicitar informações sobre o DODAG a outro nodo da rede RPL. Este recurso é utilizado quando o nodo pretende obter informações de nodos adjacentes ou então quando o nodo está em modo de inatividade (do inglês - *idle*) e não pertence a rede RPL, neste caso o nodo envia a mensagem DIS por *broadcast* temporariamente a procura de nodos com os quais possa estabelecer comunicação.

2.3.6 Construção do DODAG

A construção de um DODAG é realizado considerando uma ou mais métricas, a função objetiva, a política de configuração de nodos e a detecção e exclusão de *loop* na rede. O processo é iniciado sempre no nodo raiz que a propagação da mensagem DIO.

Assim que outro nodo recebe a mensagem DIO, este deve decidir se irá processar a mensagem ou descarta-la e aguardar por uma nova. Caso a mensagem tenha sido recebida corretamente pelo nodo, então o DIO será processado e um novo *rank* é computado de forma incremental ($\text{rank}_{\text{antecessor}+1}$) para o nodo. Se existir alguma métrica ou restrição na função objetiva o nodo verifica se o enlace atende aos requisitos mínimos destas restrições e/ou métricas, somente após atender os requisitos mínimos um nodo antecessor é definido como pai e com ele

é estabelecido um enlace para tráfego de dados. Após este processo o nodo passa a agregar o DODAG e envia mensagens DIO para outros nodos adjacentes que repetem o processo, assim a formação do DODAG continua até que todos os nodos ao alcance da rede sejam comunicados [19].

Após a construção do DODAG, a rede está pronta para operação. Todos os nodos que a compõem pertencem a alguma das seguintes classificações [4]:

- O nodo principal ou nodo raiz (do inglês, *root node*) conecta a rede RPL a outro tipo de rede fazendo o sincronismo de dados dos nodos internos com a rede externa. Este também é responsável por iniciar o processo de construção da rede a partir de configurações pré-definidas.
- Os nodos de roteamento (do inglês, *route node*) são os nodos da rede que fazem roteamento de dados, capturam dados através de sensores e enviam a um destinatário em direção ao nodo raiz.
- Os Nodos folhas (do inglês, *leaf node*) são os nodos responsáveis por informar ao raiz que a construção do DODAG finalizou. Estes nodos nunca são utilizados para roteamento no RPL convencional e o valor armazenado em seus *rank* sempre é o último número incrementado. Também possuem sensores os quais capturam informações que são enviadas ao destinatário pelo nodo raiz.

O processo de construção de um DODAG é ilustrado nas figuras 2.5, 2.5, 2.6, 2.7, 2.8. Nestes exemplos uma topologia RPL é construída utilizando um DODAG e um objeto de função cujo único critério de escolha de nodo pai é utilizar o menor custo de percurso, ou seja, percursos que minimizem o valor da métrica ETX acumulado ao longo das rotas.

- No passo 1 o nodo raiz inicia o processo de construção do DODAG enviando uma mensagem DIO por *multicast*.
- No passo 2 os nodos 11, 12 e 13 recebem a mensagem DIO, identificam que a mensagem foi enviada pelo raiz (*rank=0*) e o elegem como nodo pai, calculam os valores de ETX e o novo *rank* que é inserido no objeto de função para ser propagado aos nodos adjacentes.

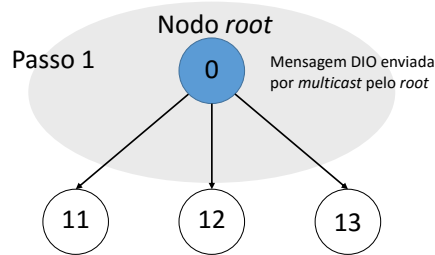


Figura 2.5: Exemplo de formação de um DODAG - 1º Passo

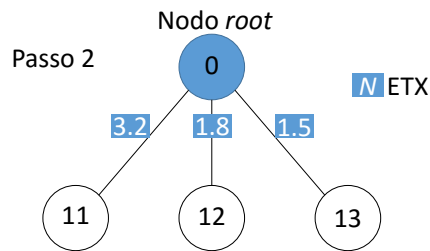


Figura 2.6: Exemplo de formação de um DODAG - 2º Passo

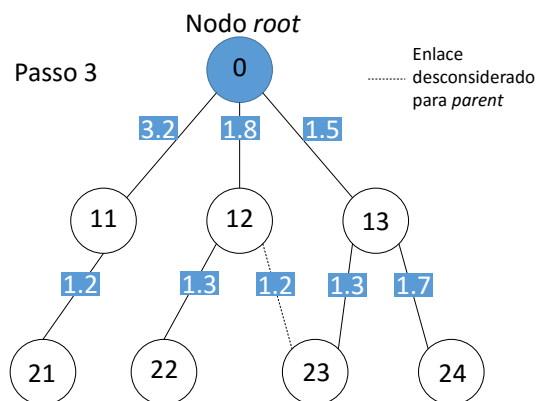


Figura 2.7: Exemplo de formação de um DODAG - 3º Passo

- O passo 3 mostra o DODAG resultante da primeira iteração. Os nós 23 e 24 selecionaram o nó 13 como *parent*. O nó 23 poderia selecionar o 12 ($ETX_{total} = 3.0$), porém o custo de percurso do nó 13 é menor ($ETX_{total} = 2.8$)
- O passo 4 mostra o DODAG finalizado. Novamente, o nó 34 poderia escolher o 24 ($ETX_{total} = 4.7$) como *parent*, porém o nó eleito foi o 23 pois o custo de percurso é menor ($ETX_{total} = 4.6$).

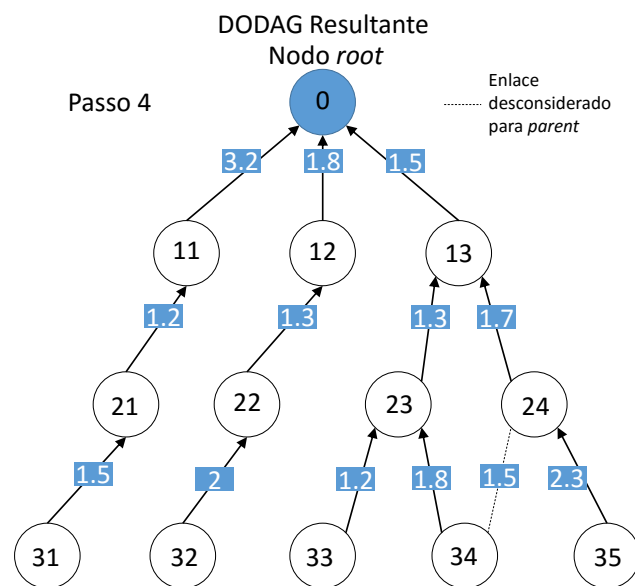


Figura 2.8: Exemplo de formação de um DODAG - 4º Passo

2.3.7 Manutenção das Tabelas de Roteamento

Após a construção do DODAG os nodos já definiram seus nodos pais, mas essa informação precisa ser divulgada ao próprio nodo pai para caso precise fazer o roteamento de dados. Então o próximo passo é popular as tabelas de roteamento ao longo do DODAG para que os nodos sejam capazes de enviar informações no *Down Direction* [19]. Para popular as tabelas de roteamento os nodos enviam mensagens DAO *unicast* ao nodo pai, a mensagem DAO carrega consigo o número de prefixo do nodo emissor. Este prefixo é armazenado na tabela de manutenção de roteamento. Supondo que no exemplo da figura 2.9 o nodo D envia uma mensagem DAO para o nodo B. A mensagem DAO irá conter, dentre outros dados, a seguinte informação:

Target A::D/128

O nodo C envia mensagens DAO ao nodo B com a seguinte informação:

Target A::C/128

E o nodo B envia ao nodo A mensagens DAO com as seguintes informações:

Target A::B/128

Target A::C/128

Target A::D/128

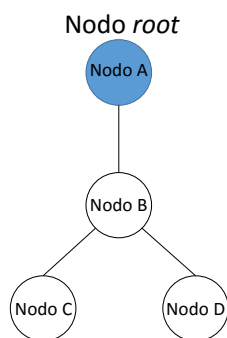


Figura 2.9: Exemplo de mensagem DAO no RPL

Desta forma quando os nodos precisam se comunicar com outros nodos fazendo uma comunicação P2P ou então quando é necessário que o nodo raiz envie uma mensagem específica para algum outro nodo o percurso de roteamento é conhecido através do uso do prefixo propagado pela mensagem DAO.

2.3.8 Temporizador *Trickle*

As LLNs são compostas por dispositivos cujos recursos são escassos, por isso é imperativo que o tráfego de informações de controle seja limitado. O RPL utiliza um mecanismo temporizador adaptativo para este fim chamado temporizador *trickle* (do inglês *trickle timer*). Este temporizador possui um algoritmo com propósito de gerenciar a troca de informações em redes LLNs de maneira robusta, com eficiência energética, simples e escalar [21, 22].

Quando são identificados eventos inconsistentes que degradam a qualidade dos enlaces na rede, o *trickle* é reiniciado e mensagens DIO são propagadas na rede com maior frequência com o objetivo de estabilizar a rede o mais breve possível. Um evento inconsistente pode ser a identificação de um *loop* de dados ou falha no enlace de comunicação de algum nodo pertencente a rede. Este mecanismo associa a frequência das mensagens DIO com a estabilidade da rede e toda vez que alguma inconsistência é detectada em nodos vizinhos sua frequência aumenta, porém quando a rede está estável a frequência de mensagens DIO cai drasticamente [23].

O *trickle* não foi desenvolvido especificamente para o RPL, mas sim para ser usado em qualquer tipo de rede LLN e seu comportamento é controlado pelos seguintes parâmetros [21], [24]:

- Intervalo de tempo corrente I .
- O cronômetro T cujo valor é definido dentro do intervalo $[I - I/2]$.
- O contador C que contabiliza a quantidade de transmissões consistentes recebidas.
- Constante de redundância K que é um número inteiro maior que zero que o nodo recebe na mensagem DIO.
- I_{\min} que é o menor valor possível para I . Este parâmetro o nodo recebe através da mensagem DIO
- I_{\max} que é o maior valor possível para I definido conforme a equação (2.1).

$$I_{\max} = I_{\min} \times 2^{I_{\text{doubling}}} \quad (2.1)$$

Sendo que $I_{\text{doubling}} \in \{1,2,4,8,16\}$. O valor inicial de I_{doubling} é recebido através da mensagem DIO.

Quando o *trickle* é iniciado o parâmetros C , I e T recebem respectivamente os valores 0, I_{\min} e um valor randômico dentro do intervalo $[I - I/2]$.

Toda vez que o nodo recebe uma transmissão consistente de mensagem DIO de um nodo pai o *trickle* incrementa o contador C . Quando o cronômetro T zera os parâmetros C e K são comparados, se C for menor que K então uma mensagem DIO é propagada pelo nodo.

Quando o intervalo de tempo I expira o *trickle* dobra o valor de I , o contador C é reiniciado e um novo valor para T é definido. Este processo se repete até que I alcance seu valor máximo I_{\max} .

Toda vez que uma mensagem DIO inconsistente é recebida o *trickle* é reiniciado com o objetivo de enviar mensagem DIO com maior frequência.

Entende-se por mensagem consistente e inconsistente as transmissões recebidas pelo nodo que atendem ao critérios definidos nas mensagens DIO e as mensagens que não atendem tais critérios respectivamente.

2.3.9 Mecanismo de reparo

O mecanismo de reparo é uma ferramenta do protocolo RPL com a finalidade de reparar a topologia da rede quando ocorre falha em alguma rota ou então quando os requisitos mínimos de métricas ou restrições já não são mais atendidos por um enlace. O RPL possui dois mecanismos de reparos de rotas, o mecanismo global e o local. Quando é detectado falha em um enlace de comunicação ou em um nodo vizinho e não existe uma rota alternativa na tabela de rotas do nodo, o mecanismo de reparo local é disparado para localizar e estabelecer um novo enlace com um nodo alternativo. Todavia, quando este mecanismo é utilizado várias vezes dentro de um DODAG, divergências na topologia da rede começam a surgir, como a formação de *loops*, prejudicando os percursos de nodos sucessores aos que efetuam reparo de rota. Para corrigir este problema, o RPL disponibiliza um segundo mecanismo, o reparo global que aciona o nodo raiz para realizar a total reconstrução do DODAG. Este segundo mecanismo tem um custo, a rede fica totalmente instável até que a reconstrução do DODAG seja finalizada, além disso, ocorre um intenso tráfego de dados de controle na rede [19].

2.4 Métricas de Roteamento para Redes de Baixa Potência e com Perdas

As redes de baixa potência e com perdas possuem características específicas quando comparadas com redes conectadas por cabos ou redes do tipo *ad hoc*. Por este motivo em [25] são definidas métricas e restrições para serem utilizadas especificamente em redes RPL. As métricas e restrições definidas em [25] são opcionais do ponto de vista de implementação do RPL sendo inclusive possível utilizá-las de maneira parcial, como é feito em [5], [9], [7] e [26].

Duas métricas são descritas em [25], são elas a métrica de confiabilidade de nível de qualidade de enlace, (do inglês *Link Quality Level Reliability Metric* - LQL) e a métrica que contabiliza as transmissões esperadas (do inglês *expected transmission count* - ETX), sendo esta segunda definida em [27] como a métrica que melhor se adapta às características de redes RPL.

2.4.1 A métrica LQL

O LQL é usado para quantificar a confiabilidade de um enlace em valores discretos com variação de 0 à 7. 0 indica que a confiabilidade do enlace é desconhecida, 1 e 7 indicam enlaces com alta confiabilidade e baixa confiabilidade respectivamente [25].

2.4.2 A métrica ETX

O ETX mede a probabilidade de que uma transmissão seja realizada com sucesso [28], e seu cálculo é feito conforme a equação (2.2) definida em [25], sendo D_f e D_r , respectivamente, a probabilidade de entrega de uma quadro no enlace nos sentidos direto e reverso.

$$ETX = \frac{1}{D_f \times D_r} \quad (2.2)$$

Para calcular o ETX cada nodo envia mensagens em *broadcast* periodicamente para os nodos vizinhos com período τ igual a um segundo. Para evitar problemas de sincronização cada *broadcast* enviado é ajustado em até $\pm 0,1\tau$ [28].

2.5 Entropia de Mobilidade

Em [11] a entropia é definida como a incerteza e o grau de desordem em um sistema. Os autores utilizam os conceitos da entropia para modelar um *framework* capaz de selecionar rotas estáveis entre determinada origem e destino em um ambiente onde múltiplos percursos estejam disponíveis com propósito de obter estabilidade e conectividade em redes móveis do tipo *ad hoc*.

O cálculo de entropia é realizado localmente nos nodos, e cada nodo deve conhecer sua situação em relação a rede. O *framework* define que em um rede *ad hoc* sem fio móvel, o número de nodos vizinhos de um determinado nodo móvel m é descrito por M_m . O conjunto de variáveis associadas ao nodo é representado por $a_{m,n}$, onde n é um nodo vizinho de m . Dois nodos podem ser considerados vizinhos se for possível uma comunicação direta entre ambos, o conjunto de variáveis $a_{m,n}$ representa a medida da velocidade relativa entre os dois nodos. Qualquer alteração no sistema pode ser descrita como uma mudança nos valores das variáveis $a_{m,n}$ no curso do tempo t , tal que $a_{m,n}(t) \rightarrow a_{m,n}(t + \Delta t)$. O vetor velocidade de um nodo m é descrito como $v(m,t)$. E o vetor velocidade do nodo n , como $v(n,t)$. A velocidade relativa $v(m,n,t)$ entre os dois nós é definida conforme equação (2.3).

$$v(m,n,t) = v(m,t) - v(n,t) \quad (2.3)$$

A mobilidade relativa entre os dois nodos pares (m,n) durante um intervalo de tempo Δt é definido com a média da velocidade relativa absoluta durante determinado tempo Δt , conforme a equação (2.4) de mobilidade relativa entre os nodos (m,n) .

$$a_{m,n} = \frac{1}{N} \sum_{i=1}^N |v(m,n,t_i)| \quad (2.4)$$

Na equação (2.4), a variável N representa a quantidade total de tempo discreto t_i em que a velocidade pode ser calculada e informada a outros nodos vizinhos dentro do intervalo Δt . Baseado nisto, é possível definir a entropia $H_m(t, \Delta t)$ no nodo móvel m durante o intervalo de tempo Δt . A entropia pode ser definida para todo o conjunto de nodos vizinhos ao nodo m ou

então para apenas um subconjunto de interesse (F_m). O valor de entropia é obtido a partir da equação (2.5).

$$H_m(t, \Delta t) = \frac{\sum_{K \in F_m} P_k(t, \Delta t) \log P_k(t, \Delta t)}{\log C(F_m)} \quad (2.5)$$

Na equação (2.5) $C(F_m)$ corresponde ao grau de cardinalidade do conjunto F_m e $P_k(t, \Delta t)$ é a velocidade relativa de um nodo m definido na equação (2.6).

$$P_k(t, \Delta t) = \left(\frac{a_{m,n}}{\sum_{i \in F_m} a_{m,i}} \right) \quad (2.6)$$

A Equação (2.5) resulta no valor de entropia normalizado ($0 \leq H_m(t, \Delta t) \leq 1$). Rotas mais estáveis apresentam entropia $H_m(t, \Delta t)$ mais próximas de 1. Rotas instáveis apresentam a entropia $H_m(t, \Delta t)$ mais próximas de 0.

Capítulo 3

Trabalhos Relacionados

Neste capítulo serão descritos trabalhos que criam propostas de modificação do RPL para cenários de mobilidade aleatória e e para redes veiculares (do inglês, *Vehicular Adhod Network* - VANET).

A proposta em [10] é adaptar o RPL para redes móveis veiculares Ad Hoc, já que o RPL foi originalmente desenvolvido para redes estáticas e a dinâmica deste tipo de rede não permite respostas à mudanças com velocidade que atenda as necessidades das VANETS, aumentando as chances de enlaces de baixa qualidade serem criados e *loops* serem formados dentro da rede. Para auxiliar nesta problemática, os autores propõem três mudanças no RPL, são elas:

- a) Uma mensagem de *ping request* é enviada imediatamente após a identificação de um nodo vizinho com o objetivo de verificar a qualidade da comunicação entre os dois nodos utilizando a métrica ETX. Caso o valor do ETX seja menor que o existente entre o nodo e seu pai então é realizada a troca de enlace e um novo pai é definido. No RPL este processo é mais demorado pois quando um nodo vizinho novo é identificado o cálculo de ETX é iniciado somente após um período τ .
- b) A mensagem DIO contém as informações necessárias para que um nodo se estabeleça em um DODAG e todas as mensagens DIO são propagadas conforme o temporizador *trickle* expira. Nesta proposta a mensagem DIO é enviada imediatamente após a eleição de um nodo pai, ignorando o temporizador *trickle*.
- c) A mensagem DAO contém as informações necessárias para popular as tabelas de roteamento de nodos antecessores e são propagadas após a eleição do nodo pai respeitando o

DelayDAO. Nesta proposta a mensagem DAO, assim como a mensagem DIO é enviada imediatamente após a eleição de um nodo pai, ignorando o *DelayDAO*.

As simulações realizadas e apresentadas no trabalho demonstram que houve uma melhora no desempenho da rede pois notou-se redução de perda de pacotes durante as transmissões quando comparado ao RPL puro, todavia os resultados das simulações de taxa de vazão de pacotes e de atraso nos pacotes mantiveram-se aproximadas nos resultados do RPL e do RPL modificado.

Apesar da significativa melhora apresentada no trabalho, os resultados não contemplam a sobrecarga de tráfego de controle na rede, consequência do aumento na frequência de envio de mensagens DIO e DAO. A sobrecarga de tráfego de controle no RPL pode ser um problema crônico em redes densas pois existe a possibilidade de inviabilizar o tráfego de dados, já que informações de controle transitam na rede com prioridade.

Em [7] novamente a problemática da mobilidade aplicada ao RPL é abordada. Os autores propõem uma melhoria no RPL, o *Mobility Enhanced RPL*, ou ME-RPL. No RPL tradicional existem três tipos de nodos dentro de uma rede, os nodos de sincronia, roteamento e folhas. No ME-RPL uma nova classificação é adicionada, o nodo móvel. No momento que algum nodo é classificado como móvel, esta informação é propagada na rede através de mensagens DIO para que este nodo seja evitado ao máximo ser escolhido como pai. O objetivo desta estratégia é atenuar as alterações na topologia da rede. Sempre que forem identificados dois ou mais nodos como possíveis pais e um deles for estático, este terá vantagem na escolha mesmo que seu valor de *rank* seja superior ao do nodo em movimentação.

O fluxo para escolha do nodo pai ilustrada na figura 3.1 funciona da seguinte forma:

- Se dois nodos tem o mesmo *rank* (quadro 2), então:
 - Se os dois nodos são estáticos ou então móveis, então o nodo até então escolhido como melhor pai conforme as regras do objeto de função é eleito (quadro 6).
 - Se um dos nodos for estático e o outro móvel, o nodo estático sempre será selecionado como pai (quadro 5).
- Se os dois nodos tiverem *ranks* diferentes (quadro 2), então :
 - Se os dois nodos forem estáticos ou móveis, então o nodo com menor *rank* é eleito como pai (quadro 8).

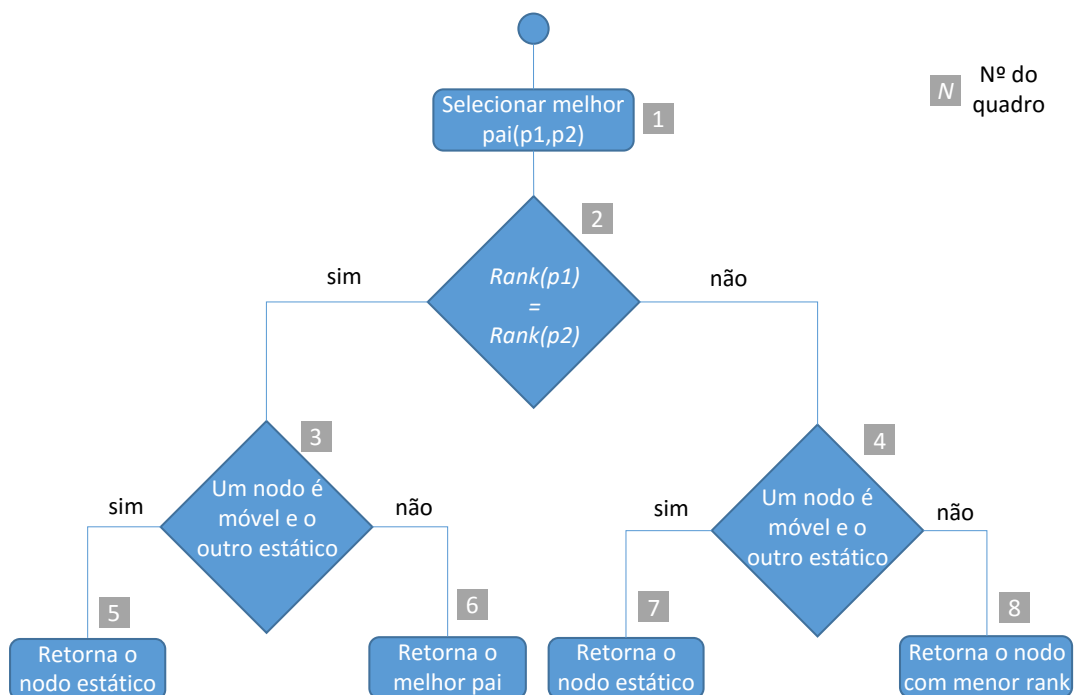


Figura 3.1: Fluxograma da escolha de nodo *Parent* do ME-RPL
 Fonte: KORBI, et.al. – com adaptações.

- Se um dos nodos for estático e o outro móvel, então o fixo sempre é selecionado como pai (quadro 7).

Outra abordagem utilizada neste trabalho é o envio dinâmico de mensagens DIS na rede. Quando a mensagem DIS é propagada via *broadcast* por algum nodo, todos os outros que recebem esta mensagem respondem enviando uma mensagem DIO. O ME-RPL aumenta a frequência de transmissão de mensagens DIS, de maneira que os nodos da rede conheçam, mais rapidamente, a existência de outros nodos adjacentes.

Os resultados de simulações que comparam o RPL e o ME-RPL demonstram que nodos móveis dificilmente são escolhidos para roteamento no ME-RPL. Simulações comparativas de estabilidade de rotas e de perda de pacotes de dados demonstram uma melhora significativa no ME-RPL, porém as simulações apresentadas foram realizadas com uma quantidade pequena de nodos móveis na rede em relação a quantidade total.

Em vários tipos de redes para sensores sem fio, como VANETs, a maior parte dos nodos estão em movimento, este padrão pode prejudicar o modelo ME-RPL considerando que a arquitetura propõe limitar ao máximo o uso de nodos móveis para roteamento.

Em [8] os autores enfatizam a problemática que envolve os nodos localizados próximos aos raiz. Estes geralmente gastam muito mais energia que os outros mais distantes do raiz devido a quantidade de retransmissão de mensagens, com isso tendem a ser os primeiros a descarregar. Quando isso ocorre, as chances do DODAG ser refeito são grandes pois nodos próximos ao raiz desconectados geram grande instabilidade na rede. Este problema é ainda mais acentuado em redes com alta densidade de nodos e mobilidade.

Para reverter este problema, em [8] é proposto um sistema de roteamento híbrido composto de nodos reativos e pró-ativos. Nodos raiz com mobilidade são posicionados na rede e se locomovem após determinado intervalo de tempo. O tamanho máximo de um DODAG é definido conforme a mobilidade dos nodos, quanto mais rápido o nodo raiz se movimenta, menor é a quantidade de nodos conectados a ele. Uma área de abrangência é definida como zona de nodos membros e os nodos que se encontram dentro desta área se comunicam e enviam dados para nodo raiz da forma convencional do RPL (nodos pró-ativos).

Os nodos localizados fora da zona de membros utilizam a metodologia *Sink Discovery* proposta pelos autores que trabalha de forma reativa, ou seja, o nodo somente estabelece a rota até o nodo raiz no momento em que precisa transmitir dados.

Conforme demonstra o fluxograma da figura 3.2, O nodo fonte, se não pertencer a zona de membros dispara o mecanismo de *sink discovery*. Um σ é definido para representar a quantidade máxima de saltos possíveis para que o nodo fonte possa estabelecer enlace. Então uma mensagem de solicitação (do inglês - *request*) é propagada e o nodo fonte aguarda por uma resposta durante um intervalo de tempo definido em t_{\square} . Caso algum nodo responda a solicitação e alguma rota até o nodo raiz seja estabelecida então o nodo fonte recebe uma mensagem de confirmação. Caso contrário, o σ é incrementado e uma nova solicitação é feita, este processo é repetido até que o alcance do nodo fonte chegue ao máximo possível. E se mesmo com este último recurso, o nodo fonte não conseguir estabelecer uma rota em direção ao nodo raiz então este passa a propagar *broadcasts* por toda rede até que seja estabelecida a rota.

O trabalho considera a mobilidade somente parcialmente, pois a proposta define apenas os nodos raiz com mobilidade. Além disso, existe uma grande preocupação dos autores em melhorar apenas o consumo de energia ignorando outras questões como atraso das informações e perda de dados durante o tráfego na rede. A proposta apresentada não foi validada através de simulações ou casos de teste.

Em [29] é proposto um modelo que estende o RPL, o roteamento em multipercorso

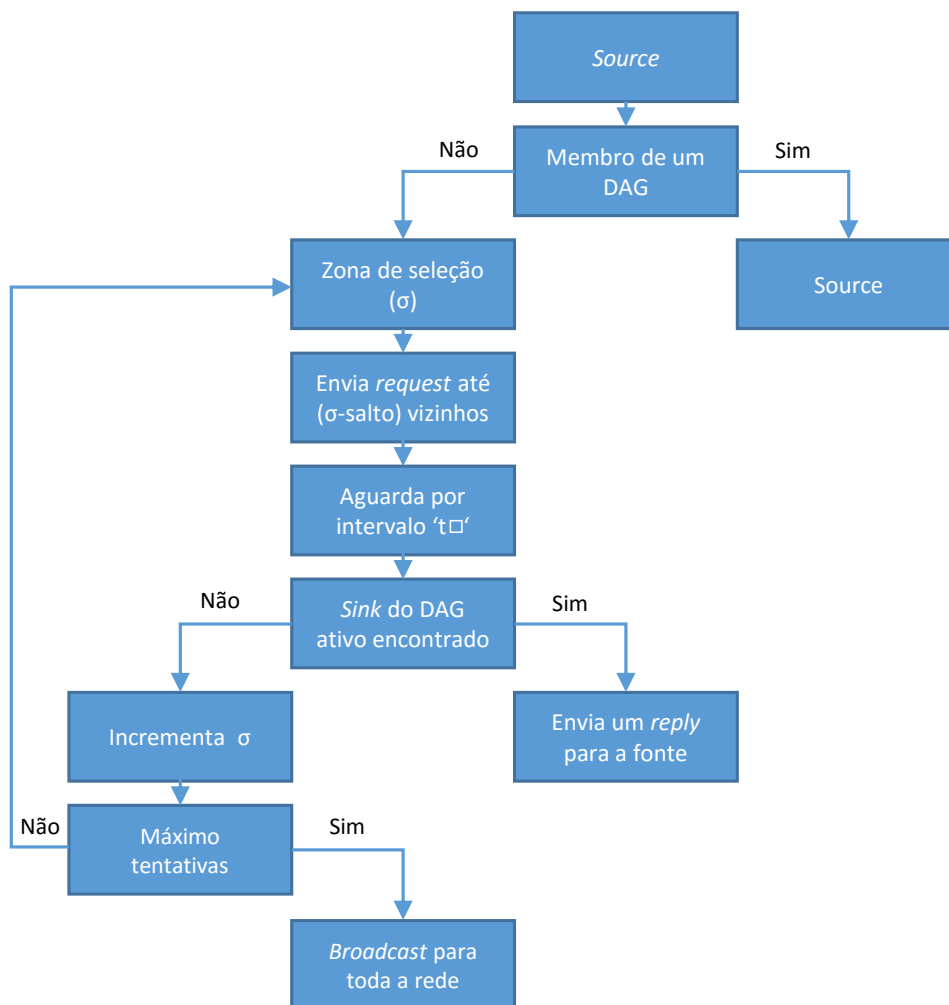


Figura 3.2: Fluxograma do mecanismo *Sink Discovery*
 Fonte: SAFDAR, et.al – com adaptações.

baseado em DAG para redes de sensores móveis (do inglês *DAG-based Multipath Routing for Mobile Sensor Networks - DMR*). Neste modelo qualquer nodo pode ser móvel, O DMR funciona bem similar ao RPL exceto nas duas seguintes situações.

- a) Ao enviar informações DIO através de *broadcast* para construção do DODAG, a métrica utilizada no DIO é o *rank*. O autor justifica o uso do *rank* como métrica alegando que o conhecimento de saltos necessários para que os dados cheguem ao nodo fonte é suficiente para um bom funcionamento em redes de baixo consumo e elevadas perdas com mobilidade. Somente quando um nodo recebe vários DIOS de nodos adjacentes com o mesmo *rank*, a métrica LQI é utilizada para estabelecer qual dos nodos adjacentes com mesmo *rank* possui melhor qualidade de enlace.

- b) O modelo cria rotas alternativas sempre que possível usando os nodos irmãos. Com isso é possível restabelecer novas rotas com maior facilidade em casos de perda de enlace. Caso não existam nodos irmãos para auxiliar no restabelecimento de rota, então o DMR inicia o processo de criação de um novo DODAG para reparar as rotas globais.

O autor apresenta simulações de desempenho do modelo proposto, porém sua proposta é comparado com redes que empregam as métricas de roteamento AODV (*Ad hoc On-demand Distance Vector*, [30]) e AOMDV (*On demand multipath distance vector routing in ad hoc networks*, [31]). Estas duas não foram projetadas para roteamento em redes com limitações de tráfego de dados e de energia, como as redes de baixa potência e com perdas.

Em [9] é proposto uma extensão do RPL que oferece maior suporte à mobilidade em redes sensores sem fio móveis, o Corona-RPL, ou CoRPL. Esse mecanismo permite a localização de nodos em movimento dividindo a rede em áreas circulares ao redor de vários nodos raiz, estas áreas são classificadas como coroas. Nesta proposta os nodos raiz são considerados como estáticos e sem mobilidade, já que estes são os nodos que se conectam a Internet e outros tipos de redes, e todos outros nodos possuem mobilidade. As coroas possuem raio iguais ao alcance máximo de transmissão do nodo raiz.

O CoRPL insere uma informação a mais na mensagem DIO que é propagada no DODAG, o Corona ID, ou C_ID que identifica as coroas. O C_ID é utilizado como coordenada relativa para que os nodos saibam sua distância em relação a quantidade de saltos até o nodo raiz. Após selecionar todos os nodos com o menor C_ID, o nodo define seu nodo pai através da métrica de qualidade ETX.

O trabalho também identifica uma problemática que surge com esta abordagem, a ocorrências de sombras, ou áreas em que o enlace é degradado devido a movimentação dos nodos. Quando nodo identifica este problema e perde o enlace com seu nodo pai os dados passam a ser enviados a qualquer outro nodo que esteja na proximidade até que um enlace com um novo nodo pai seja restabelecido.

Nos resultados das simulações demonstradas, o CoRPL apresenta um melhor desempenho no consumo de energia, atraso e perda de dados quando comparado com o RPL padrão. Porém o CoRPL não considera na sua arquitetura o posicionamento relativo entre os nodos que se movimentam, esta consideração poderia melhorar ainda mais o desempenho do CoRPL comparado ao RPL padrão.

3.1 Arquitetura do CoRPL

O CoRPL é definido em [32] como uma arquitetura que divide a área de uma rede em áreas menores circulares centradas em um nodo raiz chamadas Coronas com o objetivo de balancear o consumo de energia e o tráfego da rede. Em [9] o raio de uma Corona é definido como o alcance máximo do raio de transmissão de um sensor raiz. O exemplo 3.3 apresenta a arquitetura baseada em Coronas. Neste exemplo, todos os nodos com exceção dos raiz tem mobilidade. Existem quatro nodos raiz centralizados em um DAG e cada Corona é associada a um DAG. Os nodos com mobilidade podem pertencer somente a um DAG de cada vez e podem trocar de Corona.

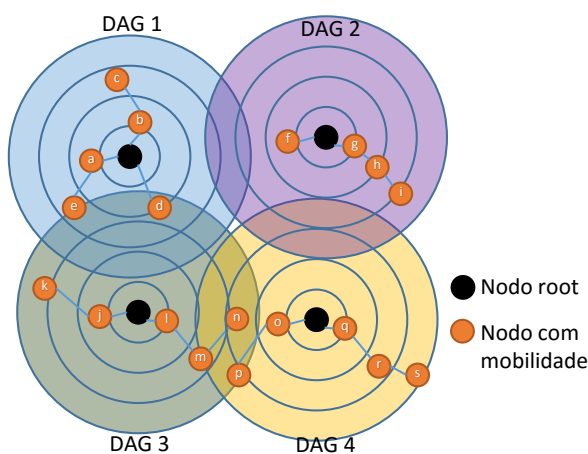


Figura 3.3: Exemplo de uma arquitetura da rede CoRPL com quatro DODAGS
Fonte: Gaddour, et.al (2014).

A especificação do RPL determina o envio por *broadcast* de mensagens DIO em intervalos de tempo determinado pelo algoritmo *trickle* para troca de informações de conectividade entre os nodos. Este intervalo de tempo é dinâmico e aumenta conforme a estabilidade da rede [19], porém a mobilidade dos nodos compromete a estabilidade e como consequência o intervalo de tempo do *trickle* é reiniciado com maior frequência ocasionando grande fluxo de dados de controle na rede.

Outro problema ocasionado pela mobilidade dos nodos ocorre durante o envio de mensagens DIO. Estas mensagens são propagadas conforme o temporizador *trickle* e este foi desenvolvido para uso em redes com nodos estáticos. Seu uso em redes com nodos móveis compromete a identificação de enlaces com boa qualidade. Por estes motivos em [17] é proposto que o nodo raiz envie mensagens DIOs periodicamente para que as informações do posicionamento do nodo seja atualizada com frequência. Além disso, os autores propõem que mediante a detecção

de inconsistências em um nodo, seus vizinhos transmitam mensagens DIO imediatamente sem a necessidade de aguardar o período do temporizador expirar.

3.1.1 Controle de Mensagens

Em [17] o legado do RPL é mantido para controle de mensagens, dessa forma é possível manter a compatibilidade entre o RPL e o CoRPL. No cabeçalho das mensagens DIO e DIS foi adicionado um bit de sinalização com o propósito de manter esta compatibilidade.

3.1.2 DIS modificado

A mensagem DIS é enviada pelo nodo quando este não estabelece um pai dentro da rede, logo este nodo não se conecta a nenhuma DAG. Com o envio do DIS o nodo inicia uma busca por um pai mais próximo. O bit F foi adicionado ao cabeçalho da mensagem DIS com o propósito de distinguir um DIS modificado do padrão. Se o valor do bit F for igual a 0, ou então inexistente a mensagem é interpretada como DIS padrão. Caso contrário ($F = 1$), a mensagem DIS é interpretada como enviada de um nodo executando o CoRPL.

3.1.3 Tabela de vizinhos

Todos os nodos, inclusive o nodo raiz podem localizar outros nodos vizinhos através de uma tabela de manutenção de vizinhos que é preenchida durante o processo de construção do DAG. Para cada nodo vizinho que compõe a tabela são armazenadas as informações de ID do nodo, ID do DAG ao qual o nodo pertence, o Corona ID e uma métrica de qualidade de enlace. A tabela 3.1 exemplifica a composição da tabela de vizinhos do nodo p da figura 3.3. Esta tabela é atualizada sempre que o nodo recebe uma nova mensagem DIO.

Nodo ID	DAG ID	Corona ID	Qualidade
m	3	3	45
n	3	4	60
o	4	2	75

Tabela 3.1: Tabela de nodos vizinhos CoRPL. [9]

3.1.4 O mecanismo Corona

O mecanismo Corona possibilita que os nodos dentro de uma rede de sensores sem fio móveis saibam sua distância em relação a quantidade de rotas existentes até chegar ao nodo raiz, e um nodo pode pertencer somente a uma coroa. A tabela 3.2 demonstra o algoritmo que representa a lógica que mantém cada nodo associado sempre a uma Corona. O nodo de sincronismo é o único que possui o valor de C_ID igual a 0 (linha 1). Cada nodo verifica sua tabela de vizinhos e computa seu próprio C_ID selecionando o vizinho com menor de C_ID incrementado de 1 (linha 3). Após a definição desta variável o nodo realiza um *broadcast* da mensagem DIO para que outros nodos possam realizar o mesmo procedimento (linha 5).

Algoritmo do Mecanismo Corona

```

1:  $C\_ID$  (DAGroot)  $\leftarrow$  0
2: DAG raiz broadcasts DIOs
3: para todos nodos móveis que recebem DIO do nodo raiz fazer
4:      $C\_ID$  (nodo móvel)  $\leftarrow \min C\_ID(\text{vizinho}) + 1$ 
5:     nodo móvel broadcast DIO com novo  $C\_ID$ 

```

Tabela 3.2: Algoritmo para calcular o valor de Corona ID. [9]

3.1.5 DAG raiz e RPL móvel

No CoRPL, mesmo após a construção do DAG o nodo raiz continua enviando mensagens DIO periodicamente, desta forma, nodos de outras DAGs próximas podem avaliar o menor Corona_ID de outras DAGs e assim decidir se continuam ou migram para outra DAG. Todos os nodos dentro da rede RPL continuam escutando DIOs de outros nodos vizinhos. Quando um nodo recebe o primeiro DIO, este computa o *rank* e o C_ID conforme o algoritmo descrito na tabela ??.

3.1.6 Mecanismo de recuperação de percurso

Segundo [9] um problema grave que ocorre em redes RPL com mobilidade é a existência de sombras na rede, ou seja, lugares onde os nodos móveis se encontram e não conseguem alcançar um nodo raiz pois perdem a conectividade com qualquer outro nodo. Este problema causa uma degradação significativa na rede pois se este nodo estiver exercendo a função de

roteamento, então todos os dados que deveriam ser enviados ao raiz nodo e de outros nodos conectados a ele são perdidos e a conexão só é restabelecida quando este nodo conseguir localizar e estabelecer conexão com qualquer outro nodo pai.

Ainda em [9] é proposto que na percepção de perda de conectividade de um nodo de roteamento, todos os outros nodos conectados a ele estabeleçam novas conexões com qualquer outro nodo que tenha valor de Corona ID menor que o seu imediatamente.

Capítulo 4

Mecanismo proposto

Este trabalho propõe um mecanismo que objetiva melhorar o desempenho do protocolo RPL em cenários onde os nodos em redes fio são móveis. Para isto foi realizado um estudo bibliográfico sobre entropia de mobilidade para determinar o modelo e parâmetros necessários a adequação do RPL e CoRPL ao modelo de entropia de mobilidade.

No RPL convencional os nodos fazem uso das mensagens DIO para aprender seus parâmetros de configurações, descobrir a instância do RPL, selecionar um nodo pai e manter o DODAG. As mensagens DIO são propagadas de acordo com o temporizador *trickle*. Conforme a equação definida em (2.1), caso os parâmetros I_{\min} e I sejam configurados com os valores 100 milissegundos e 16 respectivamente, o intervalo máximo de tempo de envio de mensagens DIO é exemplificado em (4.1)

$$I_{\max} = 100^{-3} \times 2^{16} = 6553,6s \quad (4.1)$$

Em uma rede RPL com mobilidade as mudanças na topologia física ocasionam desconectividade entre os nodos com maior frequência do que em redes estáticas. Por este motivo o intervalo de tempo resultante em (4.1) pode comprometer a comunicação entre nodos com mobilidade [9], [10].

Para evitar este problema, tal como em [9], em nossa proposta ajustaremos transmissão entre as mensagens DIO, caso ocorra a detecção de alguma inconsistência na rede, a mensagem DIO será propagada imediatamente sem aguardar o *trickle timer* expirar.

Todos os nodos possuem uma tabela de vizinhança, identificando o ID dos nodos que podem ser selecionados para roteamento. Em redes que utilizam o RPL, os principais critérios utilizados para seleção dos nodos vizinhos são qualidade de enlace, que normalmente utiliza a métrica ETX e o valor de *rank*, que indica a proximidade dos nodos e evita a formação de possíveis *loops* de dados. A Tabela 4.1 demonstra um exemplo de uma tabela de vizinhança do nodo 34 que compõe o DODAG na figura 2.8.

Tabela de vizinhança do nodo 34		
Vizinho	ETX	rank
23	1.8	2
24	1.5	2

Tabela 4.1: Tabela de vizinhança de nodo RPL.

Nesta proposta será adicionado um novo critério para seleção de nodo pai, o grau de entropia de mobilidade. Conforme em [11], um sistema auto organizável que possui partículas que se movimentam apresentam parâmetros variáveis que podem ser utilizados para determinar as características deste sistema. Uma rede sem fio composta de nodos com mobilidade pode ser comparada a um sistema auto organizável de forma que o parâmetro vetor velocidade dos nodos pode ser utilizado para cálculo de estabilidade de rotas. Desta forma é possível utilizar o modelo baseado em entropia de mobilidade como parâmetro determinante na escolha de rotas mais estáveis dos nodos.

O valor de entropia é obtido através da Equação 2.5 e o resultado, já normalizado estará contido entre 0 e 1. Quanto mais próximo de 1 for o grau de entropia, mais estável a rota pode ser considerada. Adaptando o *trickle* do RPL e adicionando um critério que considere a movimentação relativa entre os nodos móveis espera-se obter uma melhora significativa na performance das redes sensores com mobilidade.

4.1 Entropia aplicada ao CoRPL

Quando o nodo raiz é inicializado, este começa o processo de construção do DODAG enviando mensagens DIO em *broadcast* a todos os nodos vizinhos, que dão continuidade ao processo reenviando mensagens a outros nodos adjacentes. Neste caso, quando o nodo recebe

a mensagem DIO e processa a informação, é iniciado o processo de eleição do nodo pai e o Algoritmo 4.2 é iniciado (linha 03).

Caso o nodo seja inicializado e não receba nenhuma mensagem DIO em cinco segundos de espera, este entra em modo de inatividade (do inglês, *idle mode*). E assim permanece enviando mensagens DIS continuamente até receber uma mensagem DIO como resposta de outro nodo (linha 01).

Em 4.2 o parâmetro C_ID do nodo é utilizado para fazer uso das coroas implementadas no CoRPL, assim os nodos selecionam outros nodos adjacentes mais próximos do nodo raiz que estão na tabela de nodos vizinhos (linha 5). O cálculo de entropia é realizado somente para os nodos com menor C_ID (linha 10), e entre estes nodos, o que tiver menor grau de entropia de mobilidade é eleito como nodo pai. Para o caso em que dois ou mais nodos tenham o mesmo grau de entropia a métrica de qualidade é utilizada como fator de decisão na escolha. Caso somente um nodo seja selecionado com menor C_ID então este é eleito como o nodo pai.

Algoritmo de escolha de nodos pai com entropia

```

01: repetir
02:     broadcast DIS
03: até receber DIOs
04: para todos DIOS recebidos faça
05:     se C_ID (nodo móvel) ≤ C_ID (vizinhos) então
06:         armazenar ID (nodo móvel)
07:     senão
08:         nodo vizinho descarta DIO
09: para todos ID (nodo móvel) faça
10:     calcular entropia do nodo
11:     Melhor pai ← vizinho com melhor entropia
12: se C_ID(vizinho) mudar então
13:     broadcast DIOs
14: senão
15:     aguardar até o trickle timer expirar

```

Tabela 4.2: Algoritmo de escolha de nodos pai com entropia

Capítulo 5

Avaliação de Desempenho

Neste capítulo é apresentada a avaliação de desempenho do mecanismo proposto de escolha do nodo pai utilizando o grau de entropia de mobilidade. Esta avaliação é feita comparando os resultados obtidos de taxas de perda de pacotes e de atraso médio em simulações com variações de taxas de transmissão de dados, quantidade de nodos raiz, velocidade dos nodos.

Com o objetivo de se avaliar o impacto da quantidade de nodos armazenados para cálculo da entropia, optou-se também em realizar simulações com variações de tamanho da tabela de nodos vizinhos.

Dois modelos de mobilidade foram utilizados nas simulações, o modelo randômico de mobilidade (do inglês - *Random Waypoint Mobility Model*) [33], [9], [34] e o modelo de grade Manhattan (do inglês - *Manhattan Grid Model*) [35], [34], [36].

O modelo randômico é amplamente utilizado para validar protocolos de roteamento devido sua simplicidade [37]. Seu funcionamento consiste na escolha de uma posição aleatória a qual o nodo se desloca com velocidade definida também aleatoriamente entre o intervalo $[0 - V_{\max}]$, sendo V_{\max} a velocidade máxima permitida para os nodos.

O modelo randômico é sem memória, ou seja, a informação sobre o estado ou posição anterior não tem influência em decisões futuras e a velocidade atual é independente da velocidade no instante anterior. Todavia, em muitas aplicações reais este modelo não representa a mobilidade dos nodos em redes de sensores sem fio [38]. Por este motivo, o modelo de mobilidade Manhattan também foi utilizado nas simulações. Este modelo apresenta correlação temporal que é a dependência do tempo t com o tempo t_{anterior} , esta característica é descrita em [39] como dependência temporal. O modelo Manhattan simula a movimentação dos nodos

em ruas e pode ser usado para representar a movimentação em área urbana [39]. Os nodos se movimentam em direções horizontais ou verticais e sempre que ocorre a intercessão das rotas pré-definidas do modelo um cálculo probabilístico é feito com 0.5, 0.25 e 0.25 de probabilidade do nodo seguir na mesma direção, dobrar a direita e dobrar a esquerda respectivamente [34].

Para a utilização de entropia de mobilidade é necessário conhecer o posicionamento do nodo em relação ao outro nodo que se deseja calcular a estabilidade da rota. Nas simulações o posicionamento é obtido pelo emulador através das coordenadas X e Y do exato local em que se encontra o nodo no momento da seleção do nodo pai, simulando a localização precisa baseada em GPS, como em [40].

5.1 Simulador

Para a realização dos experimentos foi utilizado o COOJA, um emulador de redes de sensores sem fio utilizado em [9], [26], [41], [42] que é parte integrante de um sistema operacional para internet das coisas, o Contiki [43]. A figura 5.1 demonstra um exemplo de simulação de uma rede RPL realizado no simulador COOJA.

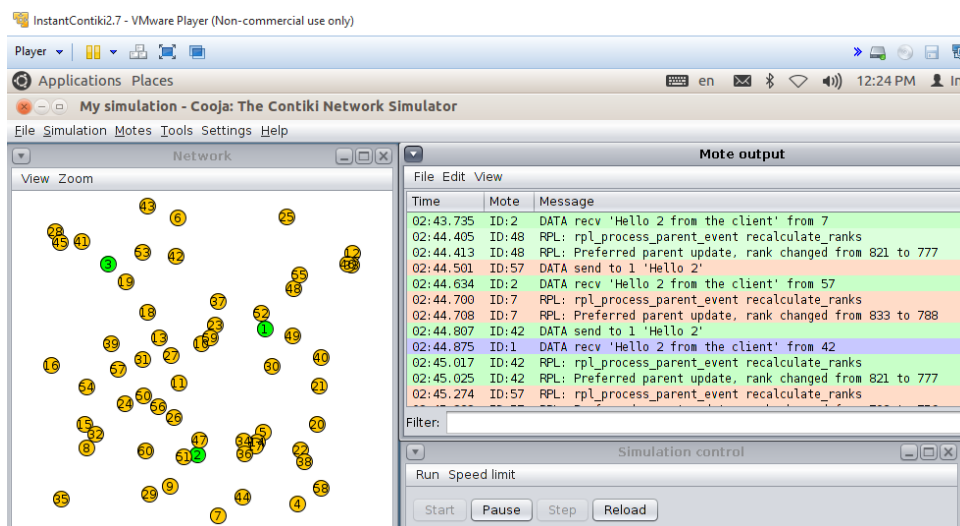


Figura 5.1: Simulação de transferencia de dados em rede RPL com o emulador COOJA

5.2 Cenários de simulação

O cenário de simulação do mecanismo proposto é composto de nodos raízes e nodos de roteamento móveis distribuídos aleatoriamente em uma área de $500 * 500m^2$. A quantidade

total de nodos nas simulações variam entre 10 e 100. Todas as simulações tiveram duração de 60 minutos. A velocidade de deslocamento dos nodos variam entre 1 m/s à 4 m/s. A quantidade de pacotes enviados por cada nodo em direção ao nodo raiz (*Up direction*) variam entre 10 à 60 pacotes por minuto e a quantidade de nodos raiz por varia entre 1 a 3 nodos. A tabela 5.1 apresenta os parâmetros utilizados nas simulações.

Parâmetros de configurações globais		valores
área de distribuição do nodos		500 * 500m ²
nodo emulado		Tmote sky
potência de transmissão dos nodos		0dBm
raio de alcance de transmissão		50m
tempo de simulação		60 min
modelos de mobilidade		modelo randômico, modelo Manhattan
Parâmetros de medição variáveis		
parâmetro de medição	quantidade de nodos	velocidade
variação de taxa de transmissão	[10-20-40-60-80-100]	3 m/s
variação de tamanho da tabela de vizinhos	[10-20-40-60-80-100]	3 m/s
variação de nodo raiz	[10-20-40-60-80-100]	3 m/s
variação de velocidade	[10-20-40-60-80-100]	[1m/s, 2m/s, 3m/s, 4m/s]
parâmetro de medição	quantidade de nodos raiz	taxa de transmissão
variação de taxa de transmissão	3	10pkt/min, 30pkts/min, 60pkts/min
variação de tamanho da tabela de vizinhos	3	30pkts/min
variação de nodo raiz	[1-2-3]	30pkts/min
variação de velocidade	3	30pkts/min

Tabela 5.1: Tabela de parâmetros de configuração das simulações.

5.3 Resultados

Para avaliar a performance da entropia de mobilidade aplicada aos nodos foram realizadas extensivas simulações com variações dos parâmetros a seguir para os modelos de mobilidade randômico e Manhattan.

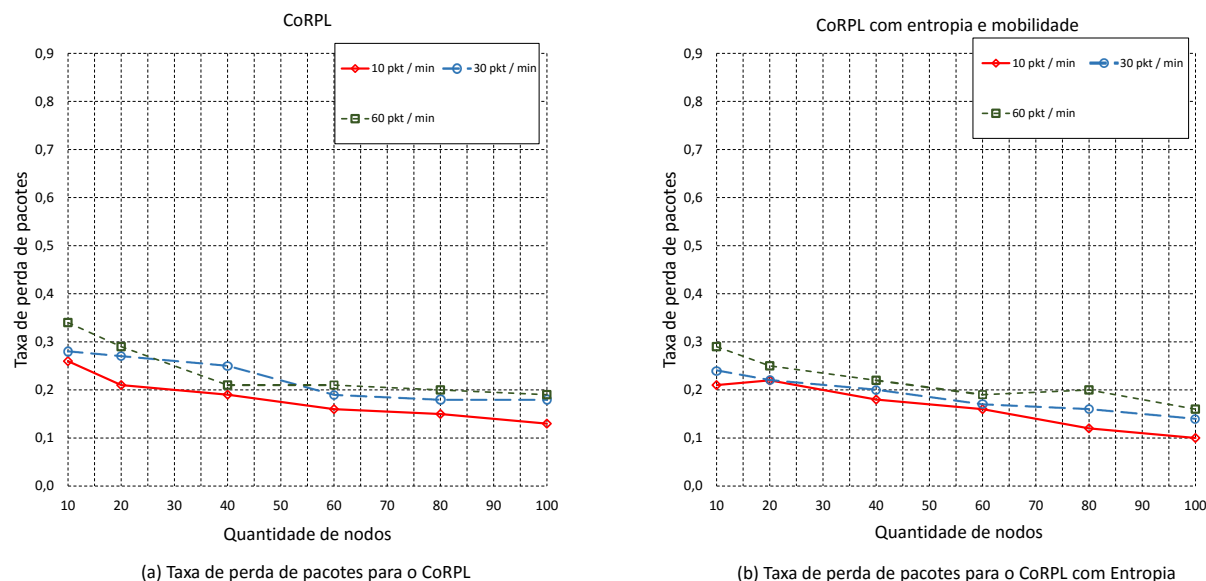
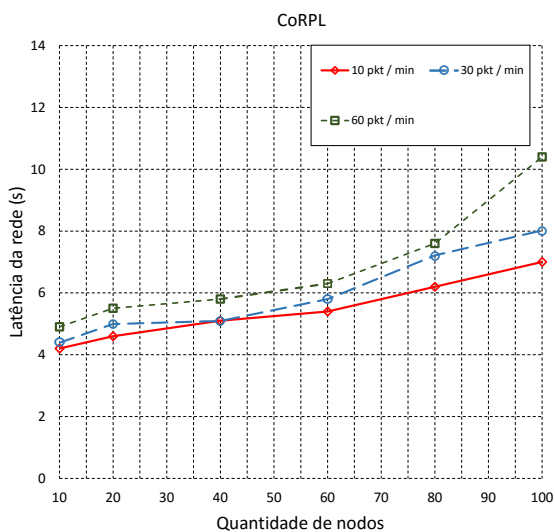


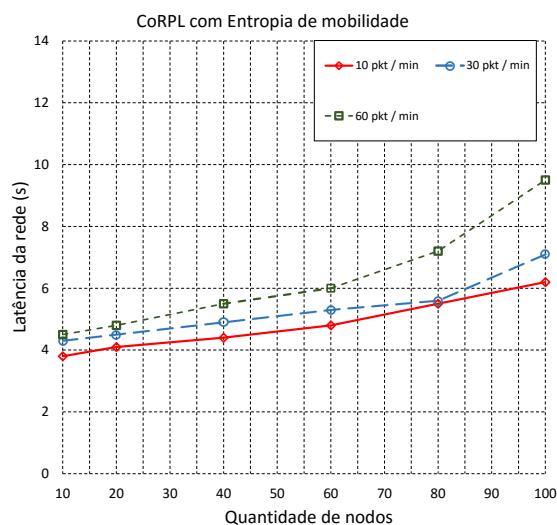
Figura 5.2: Comparação de perda de pacotes entre o CoRPL e o CoRPL com entropia de mobilidade com variação da taxa de transmissão no modelo randômico de mobilidade

5.3.1 Variação em taxas de transmissão

Os resultados para o modelo randômico em 5.2 demonstram diminuição na taxa de perda de pacotes conforme a quantidade de nodos aumenta. Este comportamento é esperado pois quanto maior a quantidade de nodos em uma mesma área, menor é a distância entre eles o que resulta em enlaces com melhor qualidade. Além disso, os nodos tem mais opções de escolha para roteamento dos dados. Os resultados em 5.2 (b) demonstram uma redução da taxa de perda de pacotes pequena quando comparados com 5.2 (a). Os resultados para o modelo Manhattan em 5.4 apresentam reduções na taxa de perda de pacotes superiores aos resultados do modelo randômico. No CoRPL as taxas de perdas chegam a 35% enquanto que no CoRPL com entropia de mobilidade as taxas de perdas nunca ultrapassam 20%. Os resultados de simulações de atraso médio em 5.3 e 5.5 demonstram que a latência é reduzida conforme a quantidade de nodos aumenta. Além disso, conforme a taxa de perda de pacotes reduz, a latência também é atenuada, o que resulta em menor latência para o modelo com entropia de mobilidade. No modelo randômico a latência chega a ser reduzida em até 1 segundo, no modelo Manhattan a redução de latência chega a 2 segundos. Este fenômeno ocorre pois no modelo de entropia os nodos tendem a manter as rotas por tempos maiores antes que mecanismos de reparos sejam utilizados devido a menor variação de velocidade entre os nodos, o que reflete em maior estabilidade.

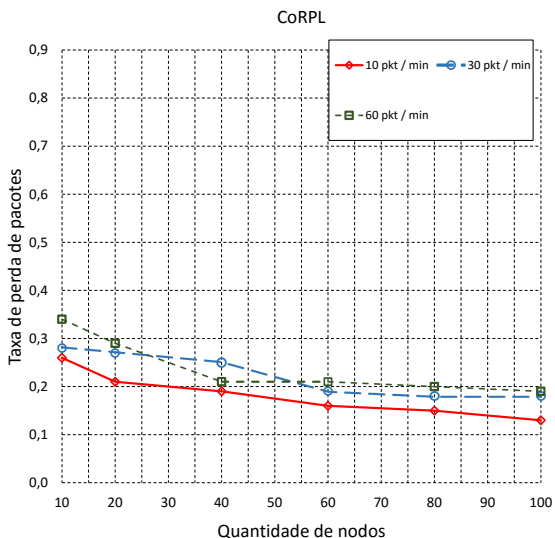


(a) Latência para o CoRPL

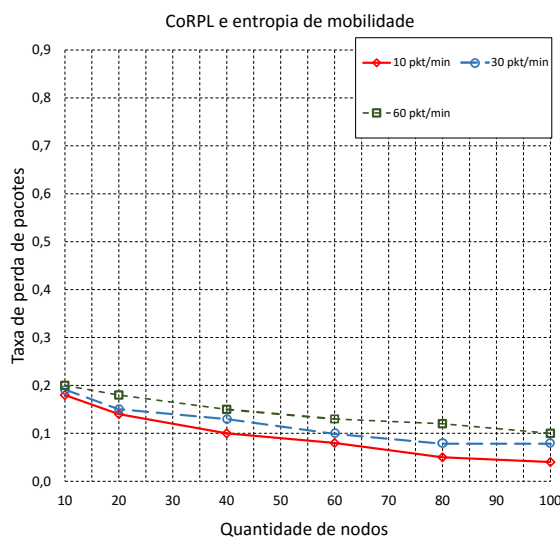


(b) Latência para o CoRPL com Entropia

Figura 5.3: Comparação do atraso médio entre o CoRPL e o CoRPL com entropia de mobilidade com variação da taxa de transmissão de pacotes no modelo randômico de mobilidade



(a) Taxa de perda de pacotes para o CoRPL



(b) Taxa de perda de pacotes para o CoRPL com Entropia

Figura 5.4: Comparação de perda de pacotes entre o CoRPL e o CoRPL com entropia de mobilidade com variação da taxa de transmissão no modelo Manhattan de mobilidade

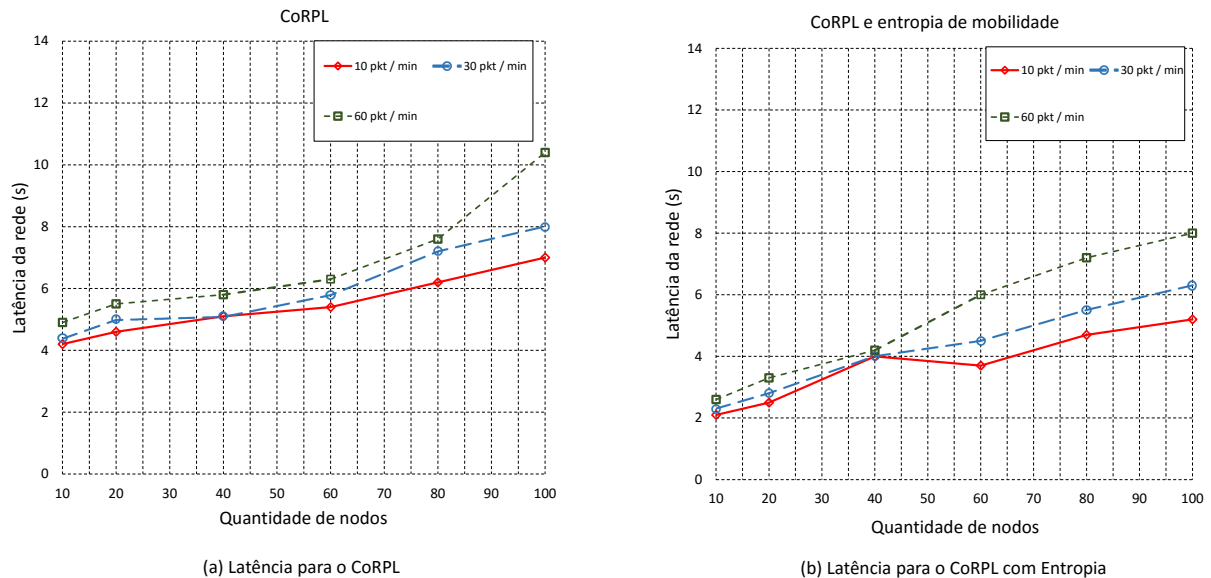
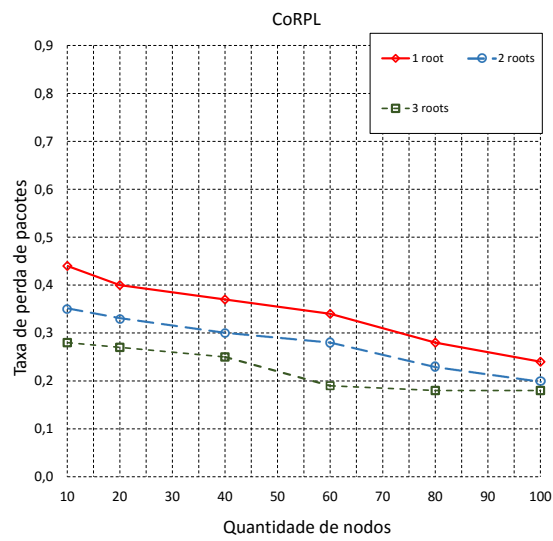


Figura 5.5: Comparação do atraso médio entre o CoRPL e o CoRPL com entropia de mobilidade com variação da taxa de transmissão de pacotes no modelo Manhattan de mobilidade

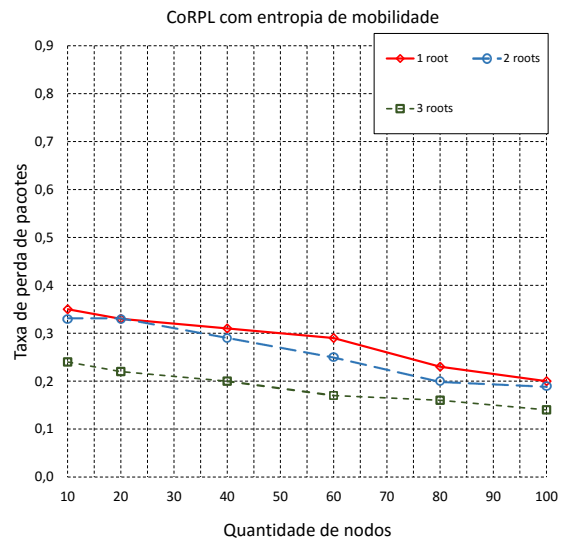
5.3.2 Variação de nodos raiz

No CoRPL a quantidade de nodos raiz define a quantidade de coroas na rede RPL. Nos resultados em 5.6, 5.7, 5.8 e 5.9 as simulações foram realizadas com 1, 2 e 3 nodos. Para o modelo de mobilidade randômico, as simulações com o CoRPL e CoRPL com entropia de mobilidade apresentaram resultados aproximados na redução da taxa de perda de pacotes com taxas variando entre 25% à 45% para 10 nodos e 15% à 25% para 100 nodos em ambos os casos. No modelo Manhattan de mobilidade em 5.8 a redução é mais expressiva. No CoRPL as taxas de perdas de pacotes variam entre 19% a 44%, enquanto que no modelo com entropia de mobilidade a variação das taxas ocorre entre 18% à 25%. Em 5.8 (b) além da redução na taxa de perda de pacotes, as variações da taxa de perda para 1, 2 e 3 nodos raiz mantiveram-se bem aproximadas nas simulações com 10 à 100 nodos, novamente demonstrando um reflexo de maior estabilidade nas rotas estabelecidas.

Nos resultados de atraso médio em 5.7 e 5.9 o impacto na latência é intensificado em redes com 40 nodos ou mais. No modelo Manhattan a redução de latência chega a ser superior a 2 segundos de diferença em simulações compostas de 100 nodos e somente 1 nodo raiz.

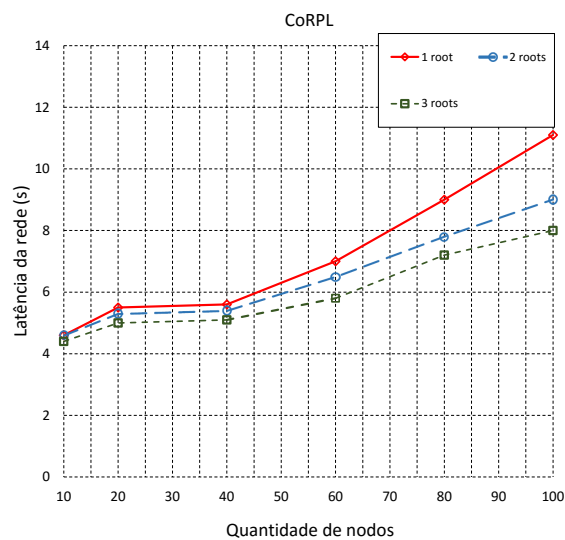


(a) Taxa de perda de pacotes para o CoRPL

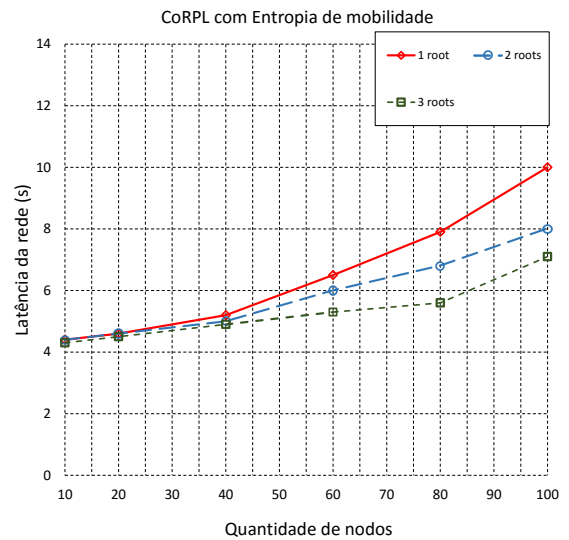


(b) Taxa de perda de pacotes para o CoRPL com Entropia

Figura 5.6: Comparação de perda de pacotes entre o CoRPL e o CoRPL com entropia de mobilidade com variação de quantidade de nodo raiz no modelo randômico de mobilidade



(a) Latência para o CoRPL



(b) Latência para o CoRPL com Entropia

Figura 5.7: Comparação do atraso médio entre o CoRPL e o CoRPL com entropia de mobilidade com variação de quantidade de nodo raiz no modelo randômico de mobilidade

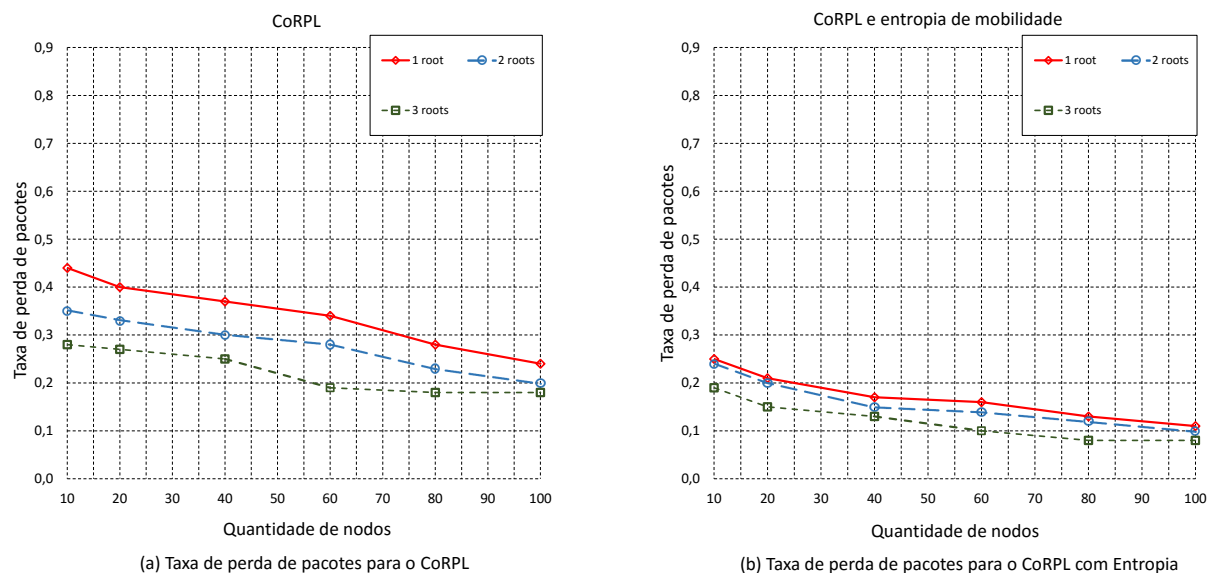


Figura 5.8: Comparação de perda de pacotes entre o CoRPL e o CoRPL com entropia de mobilidade com variação de quantidade de nodo raiz no modelo Manhattan de mobilidade

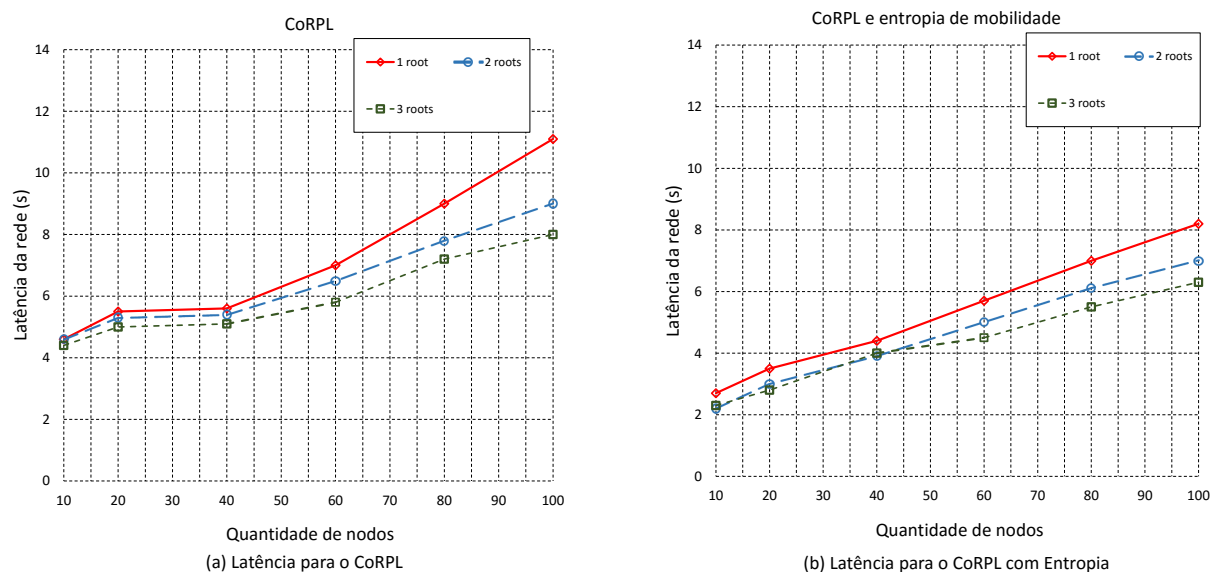


Figura 5.9: Comparação do atraso médio entre o CoRPL e o CoRPL com entropia de mobilidade com variação de quantidade de nodo raiz no modelo Manhattan de mobilidade

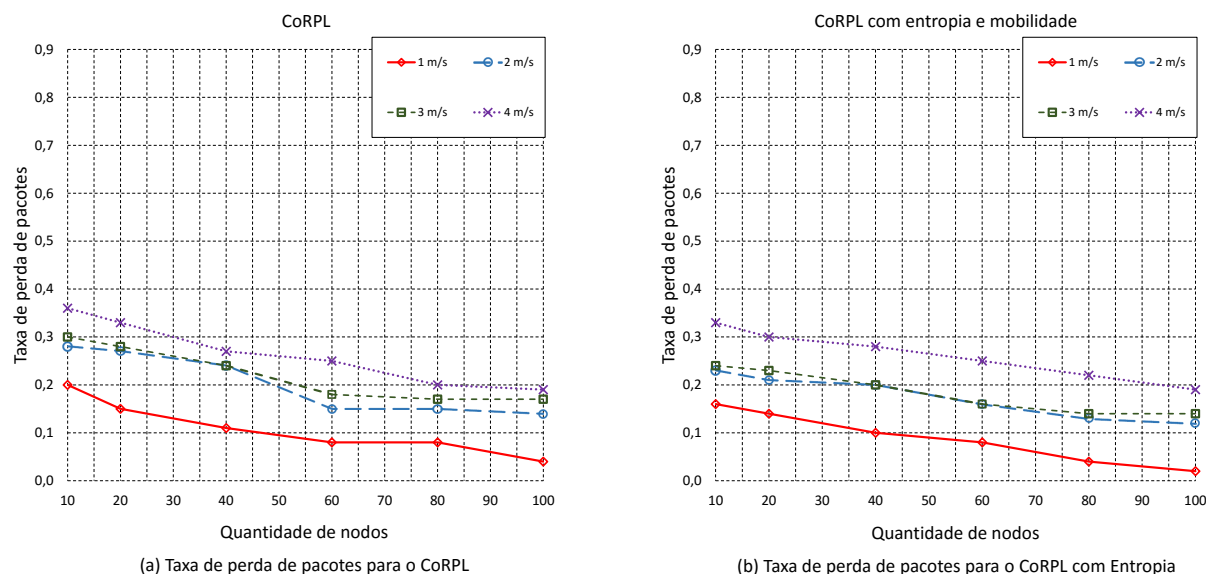


Figura 5.10: Comparação de perda de pacotes entre o CoRPL e o CoRPL com entropia de mobilidade com variação de velocidade no modelo randômico de mobilidade

5.3.3 Variação de velocidade dos nodos

Os resultados com variações de velocidade em 5.10 no modelo randômico demonstram novamente que a influência do CoRPL e o CoRPL com entropia são aproximadas. Os valores de taxa de perda de pacotes com 10 e para 100 nodos variam entre 15% à 35% e 3% à 20% respectivamente. Os resultados de atraso médio na rede em 5.11 também demonstram valores aproximados, porém em 5.11 (a) o CoRPL sem entropia de mobilidade apresenta variações de atraso médio acima de 60 nodos mais elevadas. Para as simulações com 100 nodos, o atraso médio chega a 12 segundos em 5.11 (a), enquanto que em 5.11 (b) o atraso médio não ultrapassa os 9 segundos. Os resultados em 5.12 (b) demonstram que a taxa de perda de pacotes nunca excede 20% enquanto que em 5.12 (a) as variações chegam 36%.

Os resultados de atraso médio para o modelo com entropia de mobilidade também apresentam reduções. Em 5.13 (a) os atrasos chegam a 5 segundos e 12 segundos para as simulações com 10 e 100 nodos respectivamente, enquanto que em 5.13 (b) os resultados apresentam atraso médio de 3 segundos e 9 segundos para as simulações com 10 e 100 nodos respectivamente.

5.3.4 Variação no tamanho da tabela de vizinhos

As simulações de tamanho máximo da tabela de nodos vizinhos foram realizadas com o máximo de 8, 4, 3 e 2 vizinhos, com velocidade dos nodos à 3m/s e taxa de transmissão

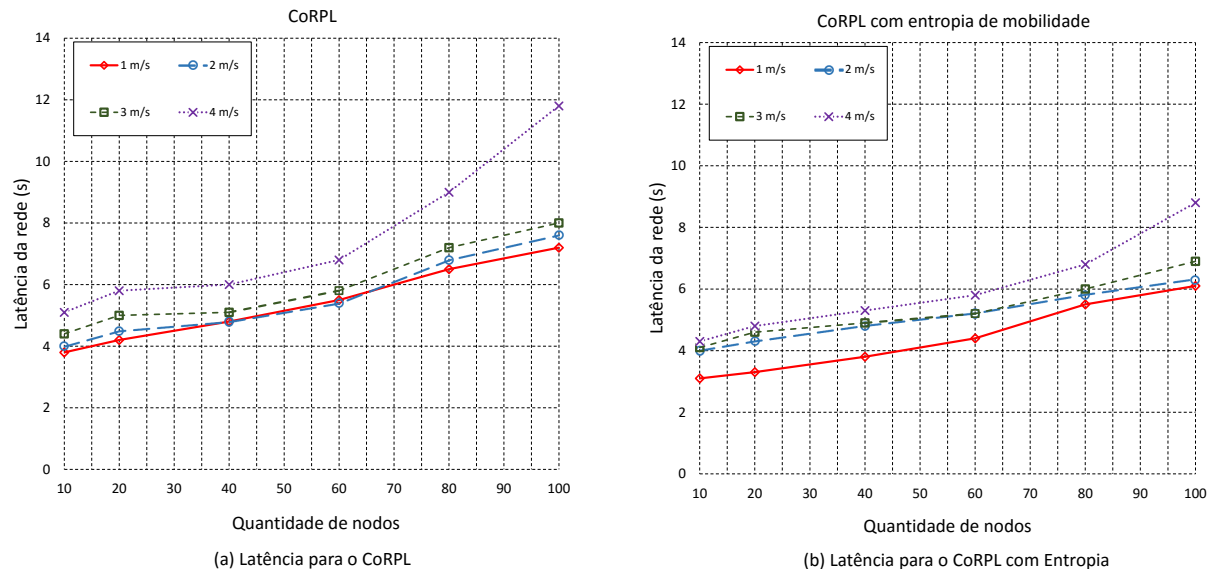


Figura 5.11: Comparação do atraso médio entre o CoRPL e o CoRPL com entropia de mobilidade com variação de velocidade no modelo randômico de mobilidade

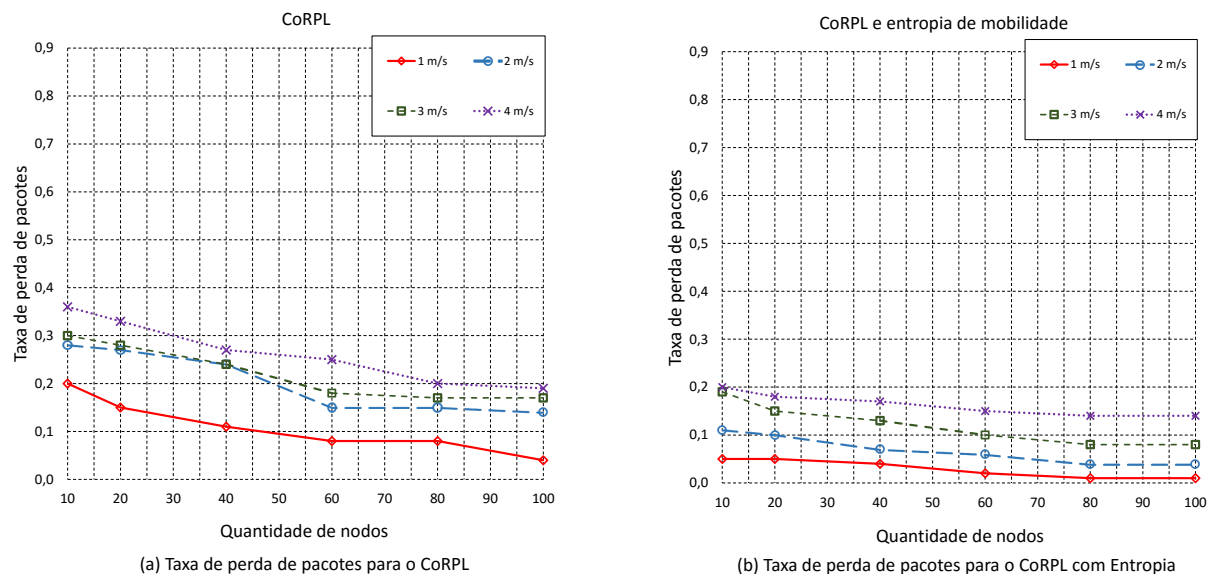


Figura 5.12: Comparação de perda de pacotes entre o CoRPL e o CoRPL com entropia de mobilidade com variação de velocidade no modelo Manhattan de mobilidade

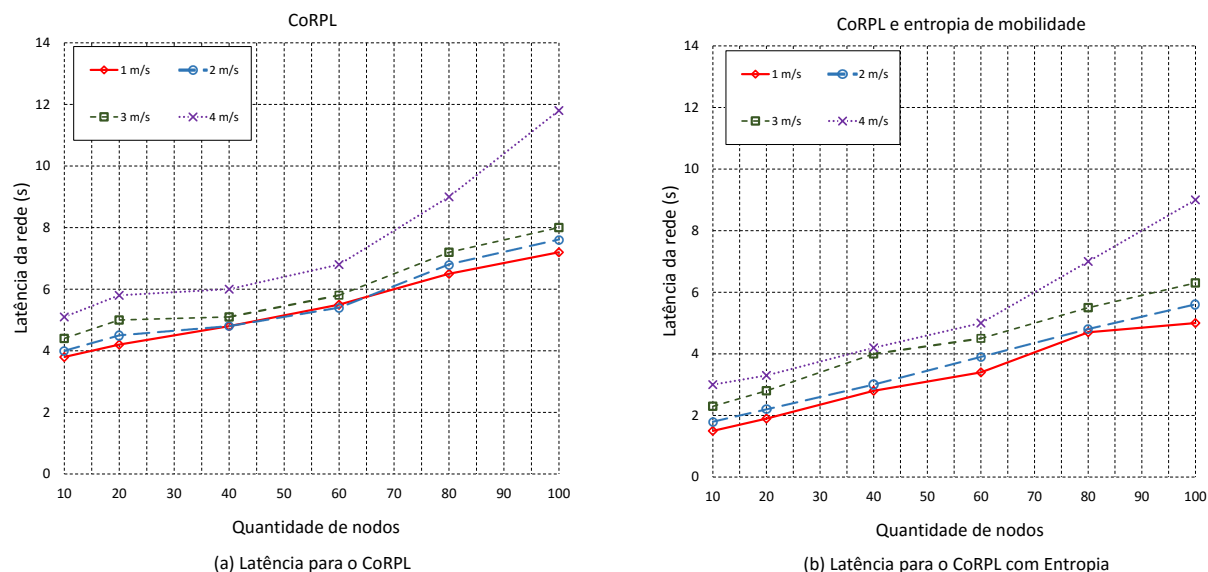


Figura 5.13: Comparação do atraso médio entre o CoRPL e o CoRPL com entropia de mobilidade com variação de velocidade no modelo Manhattan de mobilidade

de pacotes igual a 30 pacotes por minuto. A tabela de nodos vizinhos por padrão no RPL armazena até 8 nodos vizinhos. Os resultados de perda de pacotes para os modelos randômico e Manhattan em 5.14 demonstram que a redução do tamanho máximo de nodos de 8 para 4 associados a entropia de mobilidade apresentam ganhos. No CoRPL as variações de taxa perda de pacotes com tamanho máximo igual a 8 ocorrem entre 20% à 30%. No modelo de entropia de mobilidade os resultados variam entre 5% à 25%. O mesmo padrão ocorre em 5.16 para o modelo Manhattan. As taxas de perda de pacotes para uma tabela de vizinhos com tamanho máximo igual a 4 variam entre 5% à 12%.

Nos gráficos de atraso médio pode ser visto um considerável aumento na latência. Em 5.15 conforme o tamanho da tabela de vizinhos diminui, o atraso médio aumenta. Nas simulações para 100 nodos a diferença de atraso é de 9 segundos para o modelo com entropia de mobilidade. No modelo Manhattan em 5.17 (b) esse acréscimo é próximo de 7 segundos. Este aumento no atraso médio ocorre devido o uso mais frequente dos mecanismos de reparo globais do RPL.

5.4 Resumo

Neste capítulo foi apresentado uma arquitetura para integrar a entropia de mobilidade ao mecanismo de coroas proposto pelo CoRPL, ambos integrados ao RPL. Esta arquitetura foi

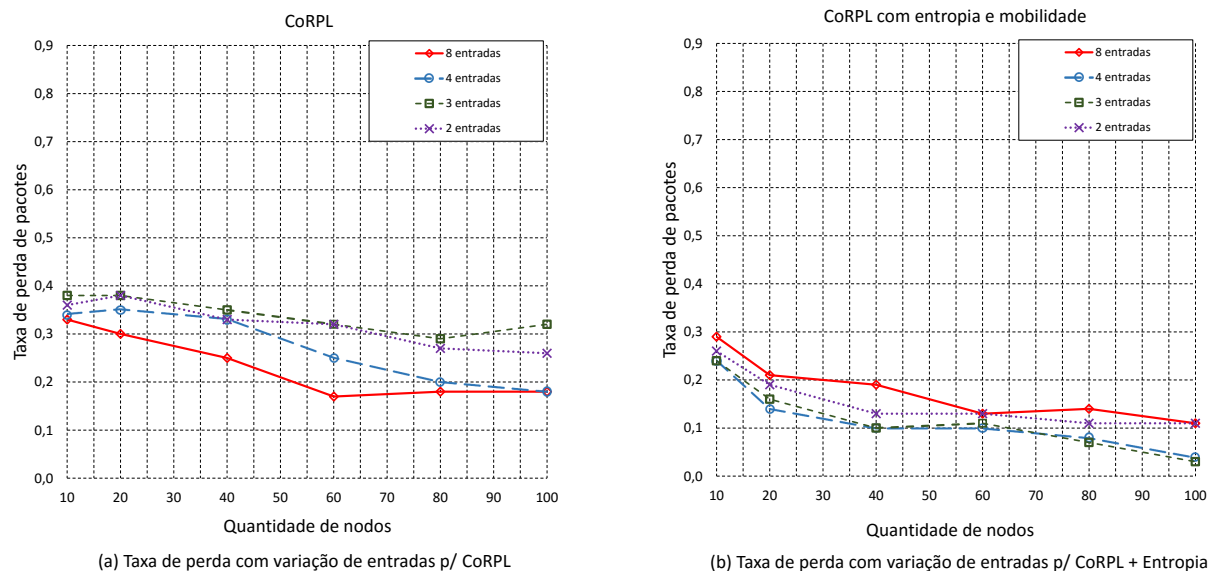


Figura 5.14: Comparação de perda de pacotes entre o CoRPL e o CoRPL com entropia de mobilidade com variação do tamanho de tabela de vizinhança no modelo randômico de mobilidade

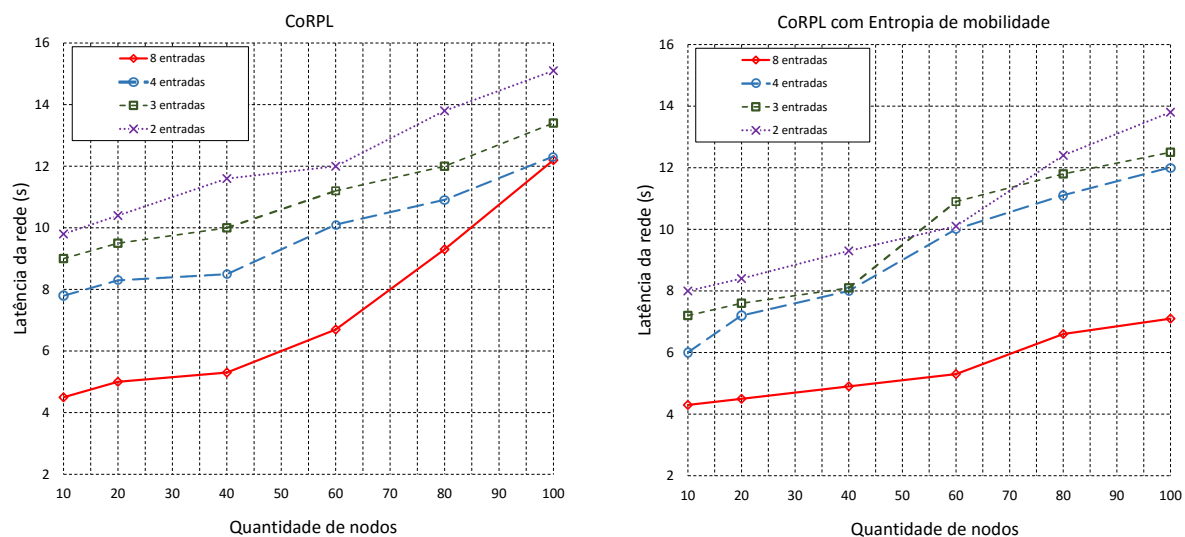


Figura 5.15: Comparação do atraso médio entre o CoRPL e o CoRPL com entropia de mobilidade com variação do tamanho de tabela de vizinhança no modelo randômico de mobilidade

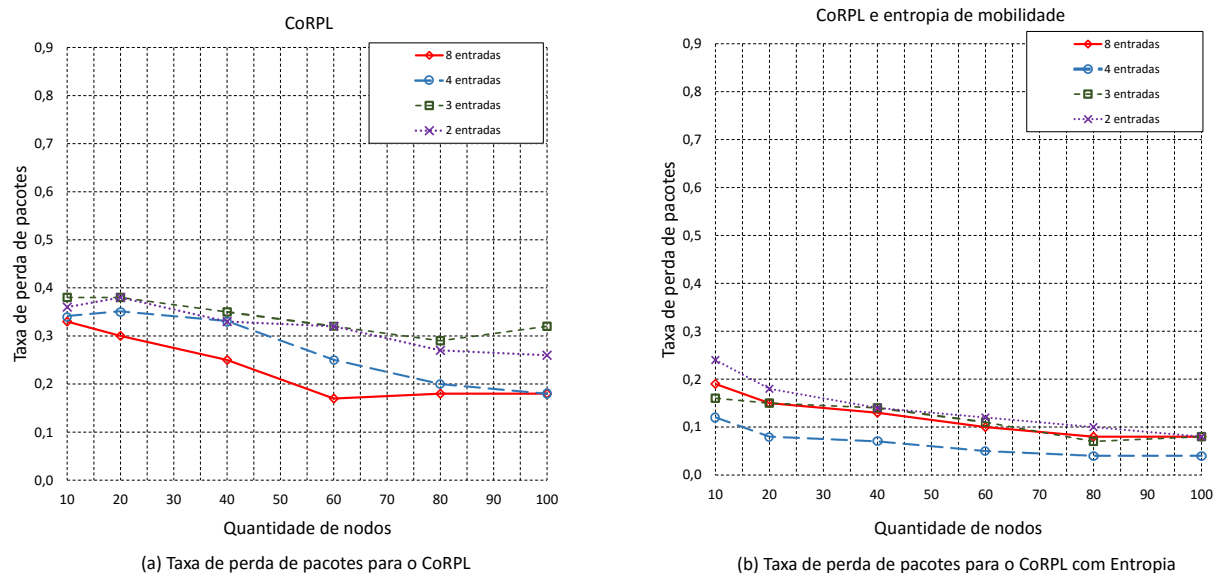


Figura 5.16: Comparação de perda de pacotes entre o CoRPL e o CoRPL com entropia de mobilidade com variação do tamanho de tabela de vizinhança no modelo Manhattan de mobilidade

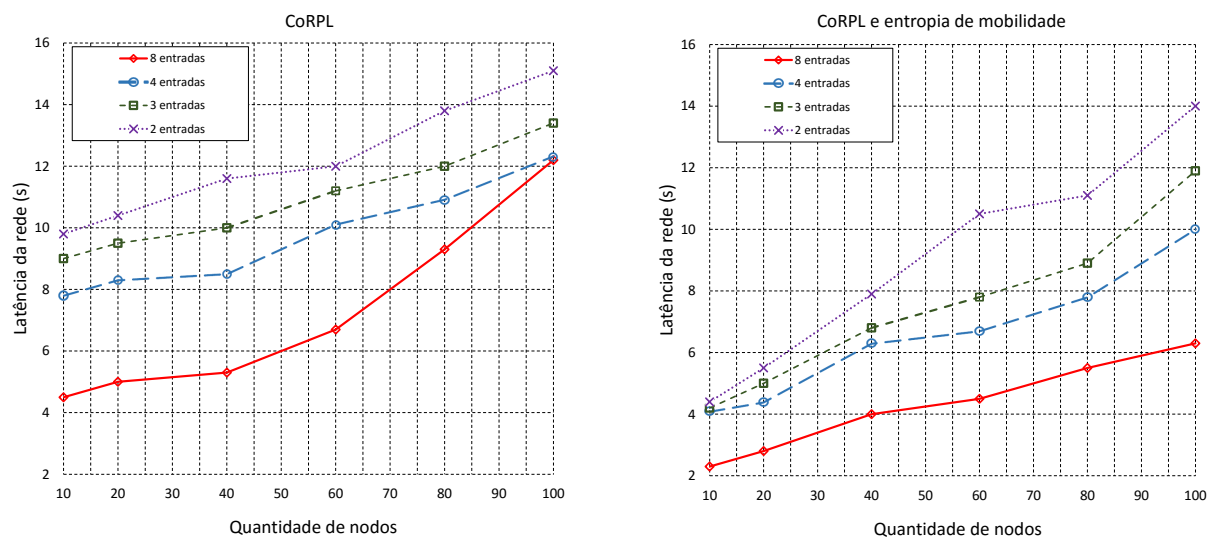


Figura 5.17: Comparação do atraso médio entre o CoRPL e o CoRPL com entropia de mobilidade com variação do tamanho de tabela de vizinhança no modelo Manhattan de mobilidade

desenvolvida para que nodos consigam definir rotas que possam ser mantidas por mais tempo e com qualidade.

Mostrou-se que o mecanismo de entropia de mobilidade pode ser mais eficiente do que o uso de métricas de qualidade em redes com mobilidade pois os resultados de latência na rede e taxas de perda de pacotes apresentaram reduções no modelo proposto.

Os resultados de taxas de perda de pacotes demonstram que os modelos de mobilidade com correlação temporal são mais beneficiados com a entropia de mobilidade do que os modelos sem correlação temporal pois em todos os resultados o modelo de mobilidade Manhattan apresentam reduções mais significativas quando comparada ao modelo randômico.

O modelo proposto utiliza o parâmetro C_ID para selecionar os nodos mais próximos do nodo raiz e posteriormente calcular a estabilidade das rotas para definir o nodo pai. Apesar da métrica de qualidade ser utilizada somente em casos de valores de entropia iguais, em [29] é definido que o conhecimento dos saltos necessários para que os dados cheguem ao nodo raiz é suficiente para uma boa performance em redes de baixo consumo e elevadas perdas com mobilidade. Os resultados de variações de tamanho da tabela de vizinhos evidencia este impacto positivo, pois nos resultados de simulações com redução do tamanho máximo da tabela de vizinho para 4 nodos apresentam diminuição nas taxas de perda de pacotes, logo, uma tabela de vizinhos maior começa a armazenar dados de nodos com cuja qualidade no enlace estabelecido é inferior. Além disso, a redução da tabela de vizinhos contribuiu para que o uso maior do mecanismo de reparo local mantendo as tabelas de vizinhos sempre atualizadas, o que também contribui para diminuição na taxa de perda de pacotes, porém o uso deste recurso tem um impacto negativo. Os resultados mostraram que conforme o tamanho da tabela de vizinhos diminui, a latência da rede aumenta proporcionalmente.

Capítulo 6

Conclusão

O trabalho realizado até o momento alcançou o objetivo esperado, que consistia no desenvolvimento de uma proposta capaz de reduzir perdas de pacotes e atraso médio durante o tráfego de dados em uma rede LLN com mobilidade utilizando o o protocolo RPL e o mecanismo CoRPL. Os experimentos realizados com a entropia de mobilidade comprovam este desempenho. Resultados de taxas de perda de pacotes, que chegam a variar nos piores casos entre 28% à 45% no CoRPL são reduzidos para uma variação entre 19% à 25% com a entropia de mobilidade no modelo de mobilidade Manhattan. Reduções no atraso da rede também ocorreram, em alguns casos o atraso médio para o CoRPL chega a variar entre 7 e 12 segundos. Essa variação é atenuada no modelo de entropia com mobilidade para 5 a 9 segundos com modelo de mobilidade Manhattan. Os resultados demonstraram que a eficácia da entropia de mobilidade é mais acentuada quando usado em conjunto com um modelo de mobilidade com correlação temporal. A redução do tamanho da tabela de vizinhos para no máximo 4 nodos vizinhos juntamente com a entropia de mobilidade demonstra variações de perda de pacotes entre 11% à 24% , uma redução ainda maior. Porém esta técnica somente seria viável para aplicações em que a latência não seja fator importante. Pois os resultados de atraso médio aumentam a medida que o tamanho da tabela de vizinhos diminui. Este comportamento ocorre devido os mecanismos de reparo local e global serem usados com mais frequência gerando impacto direto na latência da rede.

6.1 Trabalhos Futuros

Nesta seção serão apresentadas algumas sugestões de trabalhos futuros, que poderão dar continuidade aos experimentos realizados para esta dissertação.

A próxima etapa de desenvolvimento pretende analisar o comportamento da entropia de mobilidade quanto a vazão de dados e sua eficiência energética para uma análise de performance quanto a qualidade de serviço.

Pretende-se também validar o experimento de entropia de mobilidade e testar sua performance em redes reais com o uso do sistema operacional Contiki e com sensores compatíveis, como por exemplo o sensor Tmote Sky.

Para trabalhos futuros, a entropia de mobilidade pode ser associada ao RPL para ser empregada em redes com velocidades de locomoção mais elevada, como por exemplo as redes veiculares.

Referências Bibliográficas

- [1] GADDOUR, O. et al. Co-rpl: Rpl routing for mobile low power wireless sensor networks using corona mechanism. In: IEEE. *Industrial Embedded Systems (SIES), 2014 9th IEEE International Symposium on*. [S.l.], 2014. p. 200–209.
- [2] ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. *Computer networks*, Elsevier, v. 54, n. 15, p. 2787–2805, 2010.
- [3] XIA, F. et al. Internet of things. *International Journal of Communication Systems*, v. 25, n. 9, p. 1101, 2012.
- [4] ALCARAZ, C. et al. Wireless sensor networks and the internet of things: Do we need a complete integration? In: *1st International Workshop on the Security of the Internet of Things (SecIoT10)*. [S.l.: s.n.], 2010.
- [5] VASSEUR, J.-P.; DUNKELS, A. *Interconnecting smart objects with ip: The next internet*. [S.l.]: Morgan Kaufmann, 2010.
- [6] TIAN, B. et al. Application of modified rpl under vanet-wsn communication architecture. In: IEEE. *Computational and Information Sciences (ICCIS), 2013 Fifth International Conference on*. [S.l.], 2013. p. 1467–1470.
- [7] CHEN, T.-C.; CHEN, T.-S.; WU, P.-W. On data collection using mobile robot in wireless sensor networks. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, IEEE, v. 41, n. 6, p. 1213–1224, 2011.
- [8] KORBI, I. et al. Mobility enhanced rpl for wireless sensor networks. In: IEEE. *Network of the Future (NOF), 2012 Third International Conference on the*. [S.l.], 2012. p. 1–8.

- [9] SAFDAR, V. et al. A hybrid routing protocol for wireless sensor networks with mobile sinks. In: IEEE. *Wireless and Pervasive Computing (ISWPC), 2012 7th International Symposium on*. [S.l.], 2012. p. 1–5.
- [10] LEE, e. a. K. C. A comprehensive evaluation of rpl under mobility. *International Journal of Vehicular Technology*, p. 10+, 2012.
- [11] AN, B.; PAPAVALASSILIOU, S. An entropy-based model for supporting and evaluating route stability in mobile ad hoc wireless networks. *Communications Letters, IEEE, IEEE*, v. 6, n. 8, p. 328–330, 2002.
- [12] TAN, L.; WANG, N. Future internet: The internet of things. In: IEEE. *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*. [S.l.], 2010. v. 5, p. V5–376.
- [13] IBM. *Smart Planet*. 2012. Disponível em: <<http://www.ibm.com/smarterplanet/br/pt/>>. Acesso em: 20 de Agosto de 2015.
- [14] Hewlett-Packard Development Company, L.P. *HP's Central Nervous System for the Earth*. 2015. Disponível em: <<http://www8.hp.com/us/en/hp-information/environment/cense.html.VdXqTvlVhAo>>. Acesso em: 20 de Agosto de 2015.
- [15] AKYILDIZ, I. F. et al. Wireless sensor networks: a survey. *Computer networks*, Elsevier, v. 38, n. 4, p. 393–422, 2002.
- [16] YICK, J.; MUKHERJEE, B.; GHOSAL, D. Wireless sensor network survey. *Computer networks*, Elsevier, v. 52, n. 12, p. 2292–2330, 2008.
- [17] HERBERG, U.; CLAUSEN, T. A comparative performance study of the routing protocols load and rpl with bi-directional traffic in low-power and lossy networks (lln). In: ACM. *Proceedings of the 8th ACM Symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*. [S.l.], 2011. p. 73–80.
- [18] PISTER, K. et al. *Industrial routing requirements in low-power and lossy networks*. [S.l.], 2009.

- [19] TOOLS.IETF.ORG. *RFC 6550 - RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*. 2012. Disponível em: <<https://tools.ietf.org/html/rfc6550>>. Acesso em: 11 de agosto de 2015.
- [20] TOOLS.IETF.ORG. *RFC 4443 - Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*. 2006. Disponível em: <<https://tools.ietf.org/html/rfc4443>>. Acesso em: 11 de agosto de 2015.
- [21] TOOLS.IETF.ORG. *RFC 6206 - The Trickle Algorithm*. 2011. Disponível em: <<https://tools.ietf.org/html/rfc6206>>. Acesso em: 11 de agosto de 2015.
- [22] LEVIS, P. et al. The emergence of a networking primitive in wireless sensor networks. *Communications of the ACM*, ACM, v. 51, n. 7, p. 99–106, 2008.
- [23] VASSEUR, J. et al. Rpl: The ip routing protocol designed for low power and lossy networks. *Internet Protocol for Smart Objects (IPSO) Alliance*, v. 36, 2011.
- [24] ALI, H. A performance evaluation of rpl in contiki. 2012.
- [25] TOOLS.IETF.ORG. *RFC 6551 - Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks*. 2012. Disponível em: <<https://tools.ietf.org/html/rfc6551>>. Acesso em: 20 de dezembro de 2015.
- [26] DAWANS, S.; DUQUENNOY, S.; BONAVENTURE, O. On link estimation in dense rpl deployments. In: IEEE. *Local Computer Networks Workshops (LCN Workshops), 2012 IEEE 37th Conference on*. [S.l.], 2012. p. 952–955.
- [27] ZAIDI, Z. R.; SHASTRY, S. What is wrong with broadcast probing based etx estimation for wireless links? In: ACM. *Proceedings of the 10th ACM international symposium on Mobility management and wireless access*. [S.l.], 2012. p. 143–146.
- [28] COUTO, D. S. D. et al. A high-throughput path metric for multi-hop wireless routing. *Wireless Networks*, Springer-Verlag New York, Inc., v. 11, n. 4, p. 419–434, 2005.
- [29] HONG, K.-S.; CHOI, L. Dag-based multipath routing for mobile sensor networks. In: IEEE. *ICT Convergence (ICTC), 2011 International Conference on*. [S.l.], 2011. p. 261–266.

- [30] PERKINS, C.; BELDING-ROYER, E.; DAS, S. *Ad hoc on-demand distance vector (AODV) routing*. [S.l.], 2003.
- [31] MARINA, M. K.; DAS, S. R. On-demand multipath distance vector routing in ad hoc networks. In: IEEE. *Network Protocols, 2001. Ninth International Conference on*. [S.l.], 2001. p. 14–23.
- [32] ZHANG, H.; SHEN, H. Balancing energy consumption to maximize network lifetime in data-gathering sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, IEEE, v. 20, n. 10, p. 1526–1539, 2009.
- [33] BETTSTETTER, C.; WAGNER, C. et al. The spatial node distribution of the random waypoint mobility model. Citeseer.
- [34] A mobility scenario generation and analysis tool. Disponível em: <<http://sys.cs.uos.de/bonnmotion/doc/README.pdf>>. Acesso em: 05 de janeiro de 2016.
- [35] JAYAKUMAR, G.; GOPINATH, G. Performance comparison of manet protocols based on manhattan grid mobility model. *Journal of Mobile communication*, v. 2, n. 1, p. 18–26, 2008.
- [36] GERHARZ, M. et al. Link stability in mobile wireless ad hoc networks. In: IEEE. *Local Computer Networks, 2002. Proceedings. LCN 2002. 27th Annual IEEE Conference on*. [S.l.], 2002. p. 30–39.
- [37] BROCH, J. et al. A performance comparison of multi-hop wireless ad hoc network routing protocols. In: ACM. *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*. [S.l.], 1998. p. 85–97.
- [38] CAMP, T.; BOLENG, J.; DAVIES, V. A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, Wiley Online Library, v. 2, n. 5, p. 483–502, 2002.
- [39] DIVECHA, B. et al. Impact of node mobility on manet routing protocols models. *Journal of Digital Information Management*, Digital Information Research Foundation, v. 5, n. 1, p. 19, 2007.

-
- [40] CHENG, B. et al. An accurate gps-based localization in wireless sensor networks: a gm-wls method. In: IEEE. *Parallel Processing Workshops (ICPPW), 2011 40th International Conference on*. [S.l.], 2011. p. 33–41.
- [41] OSTERLIND, F. et al. Cross-level sensor network simulation with cooja. In: IEEE. *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*. [S.l.], 2006. p. 641–648.
- [42] ACCETTURA, N. et al. Performance analysis of the rpl routing protocol. In: IEEE. *Mechatronics (ICM), 2011 IEEE International Conference on*. [S.l.], 2011. p. 767–772.
- [43] OS, C. *Contiki Operational System*. 2015. Disponível em: <<http://www.contiki-os.org>>. Acesso em: 10 de outubro de 2015.