

**PROTOCOLOS DE COLETA DE DADOS PARA
REDES DE SENSORES SEM FIO**

BRUNO PEREIRA DOS SANTOS

**PROCOLOS DE COLETA DE DADOS PARA
REDES DE SENSORES SEM FIO**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais – Departamento de Ciência da Computação como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: LUIZ FILIPE MENEZES VIEIRA
COORIENTADOR: MARCOS AUGUSTO MENEZES VIEIRA

Belo Horizonte
Janeiro de 2015

© 2015, Bruno Pereira dos Santos.
Todos os direitos reservados.

Santos, Bruno Pereira dos

S237p Protocolos de Coleta de Dados para Redes de
Sensores Sem Fio / Bruno Pereira dos Santos. — Belo
Horizonte, 2015
xxv, 103 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de
Minas Gerais – Departamento de Ciência da
Computação

Orientador: Luiz Filipe Menezes Vieira

Coorientador: Marcos Augusto Menezes Vieira

1. Computação – Teses. 2. Redes de sensores sem fio.
3. Roteamento (Administração de redes de
computadores). 4. Sistemas de comunicação sem fio.
I. Orientador. II Coorientador, III Título.

CDU 519.6*22(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO


Protocolos de coleta de dados para redes de sensores sem fio

BRUNO PEREIRA DOS SANTOS

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:


PROF. LUIZ FILIPE MENEZES VIEIRA - Orientador
Departamento de Ciência da Computação - UFMG


PROF. MARCOS AUGUSTO MENEZES VIEIRA - Coorientador
Departamento de Ciência da Computação - UFMG


PROF. ANTONIO ALFREDO FERREIRA LOUREIRO
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 06 de março de 2015.

Dedico este trabalho para minha família, em especial, a minha avó Helenita Silva.

“May the force be with you.”
(Yoda.)

Resumo

Este trabalho aborda o problema de roteamento para coleta de dados em Redes de Redes de Sensores Sem Fio (RSSF). Dois protocolos são apresentados. O primeiro é chamado Centrality-based Routing Aware for L2Ns (CRAL), o qual apresenta roteamento baseado na centralidade dos nós. Os nós centrais atuam como “roteadores” e escolhem as melhores rotas estimadas por métricas de Link Quality Estimation (LQE). Estas rotas também facilitam a fusão de dados na rede. O segundo protocolo é o eXtend Collection Tree Protocol (XCTP) que melhora o Collection Tree Protocol (CTP), utilizado para coleta de dados em RSSF, criando caminhos bidirecionais entre as estações base e os nós sensores da rede, tornando possível a troca de mensagens de retroalimentação (*feedback*) e controle. Os caminhos viabilizam a implementação de protocolos de transporte de dados confiáveis para as RSSF. Os protocolos CRAL e XCTP são alternativas para o roteamento em RSSF devido a seus resultados promissores de confiabilidade, eficiência e consumo de energia.

Palavras-chave: Redes de Sensores Sem Fio, Centralidade, Roteamento, Comunicação Sem Fio.

Abstract

This work presents two protocols to mitigate the data collect in Wireless Sensor Network (WSN). The first is called Centrality-based Routing Aware for L2Ns (CRAL) that routing packets based in centrality of the nodes. The central nodes act as “router” and choose best routes through of metrics for the Link Quality Estimation (LQE). These routes also improve data fusion in the network. We also present the eXtend Collection Tree Protocol (XCTP) that improve the Collection Tree Protocol (CTP), the state-of-the-art to data collection in CTP, creating bidirectional unicast routes between the base stations and sensor nodes, thus enabling two way exchange of controls packets (acknowledgments and feedback) or data packets. Bidirectional paths enables implementation of reliable data transport protocols for WSN. The protocols CRAL and XCTP are alternatives for routing in WSN due its promising results of reliability, efficiency, and energy consumption.

Keywords: Wireless Sensor Network, Centrality, Routing, Wireless Communication.

Lista de Figuras

2.1	Exemplo da operação dos esquemas de roteamento centrados no destinatário centrado nos dados.	12
2.2	Processo para estimar a qualidade do enlace Baccour et al. [2012].	17
4.1	<i>Sink Betweenness Centrality</i> para cada vértice do grafo.	41
4.2	Arquitetura do protocolo CRAL.	43
4.3	Mensagens de controle do protocolo CRAL.	44
4.4	Operação dos algoritmos CRALs.	46
4.5	Operação do algoritmo Trickle Levis et al. [2003b].	50
4.6	Operação dos algoritmos CRALs.	52
4.7	Alcance do rádio sobre o padrão 802.11b.	59
4.8	Expected Transmission Count (ETX) médio das rotas variando número de nós sensores.	62
4.9	Taxa de entrega das mensagens de dados.	63
4.10	Consumo de energia em Joules variando o número de nós sensores.	63
4.11	Efeito <i>overhearing</i> variando o número de nós sensores.	64
4.12	Compromisso entre o uso da técnica <i>Policy Aware</i> e o consumo de energia.	65
4.13	Latência média por saltos variando o número de nós sensores.	66
4.14	Número de transmissões por dados reportados variando o porte da rede.	66
4.15	Porcentagem de mensagens coletadas no tempo.	67
4.16	Steiner Nodes variando o número de nós sensores.	67
4.17	<i>Freshness</i> para o algoritmo CRAL-Low Drop (CRAL-LD).	68
4.18	<i>Freshness</i> para o algoritmo CRAL-Fast Delivery (CRAL-FD).	69
4.19	Compromisso entre número de transmissões e <i>freshness</i>	69
4.20	Avaliação comportamento dos protocolos sobre o número de transmissões por dados gerados em face às funções de fusão de dados.	71
5.1	Arquitetura do protocolo XCTP.	77

5.2	Formato dos pacotes para o protocolo XCTP.	79
5.3	Reação do plano de controle sobre o plano de dados, quando ocorre detecção de <i>loop</i> e atualização de caminhos.	81
5.4	Operação do protocolo Transport Automatic Piggyback Protocol.	84
5.5	Transferência de um arquivo de 512KB de um nó a 5 saltos de distância da Estação Base (EB).	86
5.6	Reação dos protocolos XCTP e Ad hoc On Demand Distance Vector (AODV) na ocorrência de falhas na rede.	88
5.7	Reação do protocolo XCTP na presença de falhas na rede. 50 fluxos estão ativos.	89
5.8	Consumo de memória da tabela de roteamento pelo número de fluxos para os protocolos XCTP, Routing Protocol for low-power and lossy networks (RPL) e AODV.	89
5.9	Tabela de Fluxos Reversos do protocolo variando o número de fluxos ativos e nós sensores na rede.	90
5.10	XCTP requer menor quantidade de mensagens de controle que RPL.	90

Lista de Tabelas

2.1	Comparação entre os tipos de centralidade	22
2.2	Comparativo entre estimadores de enlace	23
3.1	Comparação dos algoritmos de roteamento	31
3.2	Comparação dos paradigmas de comunicação sobre os protocolos relacionados ao XCTP.	33
4.1	Parâmetros para cálculo da centralidade <i>Sink Betweenness</i>	41
4.2	Custo para estabelecimento das rotas para os protocolos relacionados	57
4.3	Resumo dos modelos utilizados para simulações com o Sinalgo	58
4.4	Parâmetros da modelagem 802.11b para Redes de Sensores Sem Fio (RSSF) [Awerbuch et al., 2004; Orinoco wireless networks, 2014]	59
4.5	Cenário padrão de simulação	60
5.1	Parâmetros padrão de simulação	86
5.2	Consumo de memória RAM e ROM em Bytes.	91

Lista de Siglas

AC	Address-Centric
AODV	Ad hoc On Demand Distance Vector
API	Application Programming Interface
ARQ	Automatic Repeat Request
ARR	Acknowledgment Reception Ratio
AS	Sistemas Autônomos
ASL	link Asymmetry Level
ASNR	Average-Signal-to-Noise Ratio
CNS	Center at Nearest Source
CRAL	Centrality-based Routing Aware for L2Ns
CRAL-FD	CRAL-Fast Delivery
CRAL-LD	CRAL-Low Drop
CT	Centrality Tree
CTP	Collection Tree Protocol
DAARP	Data-Aggregation Aware Routing Protocol
DAG	Directed Acyclic Graph
DAO	Destination Advertisement Object
DC	Data-Centric
DIO	DAG Information Object

DIS	DAG Information Solicitation
DODAG	Destination Oriented DAG
DSR	Dynamic Source Routing
DYMO	Dynamic MANET On-demand Routing Protocol
EB	Estação Base
ETT	Expected Transmission Time
ETX	Expected Transmission Count
F-LQE	Fuzzy Link Quality Estimator
FBC	Flow Betweenness-Centrality
InFRA	Information Fusion-based Role Assignment
KLE	Kalman-filter-based Link quality Estimator
L2Ns	low-power and lossy networks
LI	Link Inefficiency
LQE	Link Quality Estimation
LQI	Link Quality Indication
MANET	Mobile Ad-hoc Networks
MD	Mensagem de Descoberta
MG	Mensagem de Gerência
MNR	Mensagem de Nova Rota
MR	Mensagem de Resposta
PerHop	Periodic per hop simple
PerHopAd	Periodic per hop adjusted
PRR	Packet Reception Ratio
PSP	Packet Sucess Probability

QUDG	Quasi Unit Disk Graph
RBC	Routing Betweenness Centrality
RNP	Required Number of Packet retransmissions
RPL	Routing Protocol for low-power and lossy networks
RREP	Route Reply packet
RREQ	Route Request
RSS	Received Signal Strength
RSSF	Redes de Sensores Sem Fio
RSSI	Received Signal Strength Indication
RWBC	Random Walk Betweenness-Centrality
SBC	Sink Betweenness Centrality
SF	link Stability Factor
SINR	Signal-to-Interference-plus-Noise Ratio
SNR	Signal-to-Noise Ratio
SPBC	Shortest Path Betweenness-Centrality
SPRR	smoothed Packet Reception Ratio
SPT	Shortest Paths Tree
TAP2	Transport Automatic Piggyback Protocol
THL	Time Has Live
TLC	Traffic Load Centrality
TTL	Time To Live
UDG	Unit Disk Graph
WMEWMA	Window Mean with Exponentially Weighted Moving Average
WRE	Weighted Regression Estimator

WSN	Wireless Sensor Network
XCTP	eXtend Collection Tree Protocol

Sumário

Resumo	xi
Abstract	xiii
Lista de Figuras	xv
Lista de Tabelas	xvii
Lista de Siglas	xix
1 Introdução	1
1.1 Objetivos Gerais	3
1.1.1 Objetivos Específicos	4
1.2 Motivação	4
1.3 Contribuições	5
1.4 Organização do Trabalho	6
2 Fundamentação Teórica	7
2.1 Roteamento em RSSF	7
2.2 Paradigmas de Roteamento	8
2.2.1 Características das RSSF	9
2.2.2 Análise de Dados	10
2.2.3 Roteamento Address-Centric e Data-Centric	10
2.2.4 Fusão de Dados	11
2.3 Principais Problemas de Roteamento em RSSF	14
2.3.1 Modelagem do Ambiente Sem Fio	14
2.3.2 Estimadores da Qualidade do Enlace	16
2.4 Centralidade e Estimadores de Qualidade do Enlace	18
2.4.1 Centralidade	18

2.4.2	Estimadores de Qualidade do Enlace	22
2.5	Sumário da Fundamentação Teórica	28
3	Trabalhos Relacionados	31
3.1	Trabalhos Relacionados – CRAL	31
3.2	Trabalhos Relacionados – XCTP	33
3.3	Sumário dos Trabalhos Relacionados	37
4	Protocolo de Roteamento CRAL (<i>Centrality-based Routing Aware for L2Ns</i>)	39
4.1	Escopo do Problema	40
4.1.1	Centralidade	40
4.1.2	Estimador de Enlace	42
4.2	O Protocolo <i>Centrality-based Routing Aware for L2Ns</i>	43
4.2.1	Arquitetura	43
4.2.2	Algoritmo	45
4.2.3	Funções de Fusão de Dados	49
4.2.4	Discussão sobre o protocolo CRAL	50
4.2.5	Mecanismo de balanceamento de Energia <i>Policy Aware</i>	51
4.3	Perda de dados	55
4.4	Análise do Protocolo	56
4.5	Avaliação	57
4.5.1	Metodologia	57
4.5.2	Resultado de Simulação	62
4.6	Conclusões do Capítulo	72
5	Protocolo de Roteamento XCTP (<i>eXtend Collection Tree Protocol</i>)	75
5.1	Escopo do Problema	76
5.2	O Protocolo <i>eXtend Collection Tree Protocol</i>	77
5.2.1	Arquitetura	77
5.2.2	Modificação no Pacote de Dados	78
5.2.3	Fluxo Reverso	79
5.2.4	Reação a Mudanças na Topologia	80
5.2.5	Interfaces do XCTP	82
5.2.6	O Protocolo <i>Transport Automatic Piggyback Protocol</i>	83
5.3	Avaliação	84
5.3.1	Metodologia	85
5.3.2	Resultados de Simulação	87

5.4	Conclusões do Capítulo	91
6	Considerações Finais	93
6.1	Trabalhos Futuros	94
	Referências Bibliográficas	95
A	Publicações	103

Capítulo 1

Introdução

Eventos em sistemas computacionais são entidades de programação que representam um acontecimento do mundo real ou não. Como exemplo de sistema de processamento de eventos destacam-se as Redes de Sensores Sem Fios (RSSFs), as quais têm por finalidade coletar, disseminar e analisar adequadamente os eventos. As tarefas de coleta e análise dos dados apresentam grandes desafios, sendo portanto abordadas aqui. Na fase de coleta de dados, algumas questões pertinentes são: como criar uma estrutura de roteamento para entregar os dados ao destinatário? Como confirmar o correto recebimento das mensagens enviadas? Quais são os modelos mais adequados para as características das RSSFs? Já na fase de análise, quais benefícios são possíveis de obter das diferentes técnicas de fusão de dados, mineração de dados, aprendizado de máquinas, dentre outras para que as RSSF sejam eficientes e poupem recursos?

As RSSF geralmente são dirigidas a eventos, além disso, é comum que toda a rede coopere para que os eventos detectados pelos nós sensores trafegam pela rede até seu destino. A tarefa de cooperação entre as entidades da rede é desafiadora em face das severas limitações computacionais, de comunicação, e, em especial, do consumo de energia dos dispositivos empregados nessas redes. Devido à relevância tanto prática quanto teórica dessas questões e a emergência das RSSF se faz necessário propor algoritmos que regulam de modo adequado as seguintes características:

- *Dispositivos*: Diz respeito às características de hardware dos dispositivos que podem apresentar ou não heterogeneidade entre si, ou seja, os nós podem portar diferentes rádios, quantidade de memória e processadores. Sendo assim, é recomendável que um algoritmo para RSSF não assuma quaisquer características de hardware dos nós;
- *Ambiente*: No que tange às características do ambiente em que os dispositivos

serão dispostos é dado destaque a presença ou não de obstáculos. Estas barreiras influem diretamente na qualidade dos enlaces entre os dispositivos que irão se comunicar. Deste modo, para que as informações das variáveis monitoradas pelos nós sensores sejam entregues de modo eficiente se faz necessário que o algoritmo calibre as rotas baseando-se na qualidade dos enlaces, assim deve-se preferir enlaces de alta qualidade em detrimento dos de baixa qualidade;

- *Aplicações:* No quesito das aplicações deve-se entender qual paradigma é suficiente para suprir os requisitos de comunicação da aplicação. A troca de informações em uma rede de sensores, em geral, ocorre de três modos: 1) a partir dos nós sensores em direção às Estações Bases (EBs); 2) iniciando nas EBs em direção aos nós sensores; ou 3) através de comunicação ponto-a-ponto em é permitido que quaisquer duas entidades troquem informações. Estes paradigmas apresentam diversos compromissos, portanto escolher de modo adequado qual paradigma utilizar pode poupar recursos dos nós sensores e reduzir a complexidade dos algoritmos utilizados.

Portanto, calibrar de modo adequado os parâmetros dos dispositivos, ambiente, e aplicação conduzem a eficiência de uso das RSSF.

Esta dissertação trata do problema de roteamento de mensagens em Redes de Sensores Sem Fio. A camada de roteamento nas redes de sensores é de relevante importância, pois expressam grande influência no consumo de recursos dos nós, além de ser fundamental para a escolha dos protocolos tanto acima quanto em baixo na pilha de protocolos. Neste sentido, são mostradas técnicas para ajustar as características acima citadas para que as RSSF sejam usadas de modo eficiente. Assim são propostos dois protocolos de roteamento para RSSF, sendo eles:

1. **Centrality-based Routing Aware for L2Ns (CRAL)** que constrói a estrutura de roteamento para coleta de dados. CRAL se baseia em centralidade para escolher os nós intermediários na rede. Além disso, faz uso de Link Quality Estimation (LQE) para estimar as rotas de melhor qualidade no ambiente sem fio. Sendo assim, serão mostradas três diferentes abordagens do algoritmo CRAL:
 - a) **CRAL-Low Drop (CRAL-LD):** o qual utiliza Expected Transmission Count (ETX) como LQE para encontrar caminhos que necessitem de baixa quantidades de transmissões e retransmissões para entregar as mensagens de dados, deste modo CRAL-LD encontra rotas com baixa probabilidade de perda de pacotes;

- b) **CRAL-Fast Delivery (CRAL-FD)**: que emprega como LQE o estimador Expected Transmission Time (ETT), esta métrica permite encontrar rotas de alta vazão, ao passo que mantém baixo o número de retransmissões para entregar as mensagens de dados;
 - c) **CRAL + *Policy Aware***: na qual utiliza-se o protocolo CRAL em conjunto com políticas conscientes para balancear o consumo de energia dos nós. Esse balanceamento consiste em alterar as rotas fazendo com que o fluxo de tráfego seja compartilhado entre os nós mais e menos centrais, aumentando o tempo de vida desses nós centrais e consequentemente mantendo a rede conectada por mais tempo;
2. **eXtend Collection Tree Protocol (XCTP)** um protocolo de roteamento que é uma extensão do Collection Tree Protocol (CTP). CTP cria uma árvore de roteamento para transferir dados de um ou mais nós sensores para as Estações Base. Porém, CTP não cria os caminhos reversos entre as Estação Bases (EBs) e os nós sensores. O caminho reverso é importante, por exemplo, para que comandos de retroalimentação ou pacotes de confirmação (*acknowledgment*) sejam enviados. Assim, XCTP habilita a comunicação em ambos os caminhos: das EBs para os nós sensores e dos nós sensores para as EBs. XCTP constrói essas rotas de modo consciente, isto é, usando pouca memória e apresentando baixo custo adicional nos pacotes de dados.

1.1 Objetivos Gerais

O objetivo principal deste trabalho é propor dois algoritmos de roteamento de baixo custo computacional sendo eles Centrality-based Routing Aware for L2Ns e o eXtend Collection Tree Protocol. Tais algoritmos são distribuídos, empregam estimadores de qualidade de enlace para calcular as rotas e visam melhorar o uso dos recursos dos dispositivos da rede ao apresentar baixo custo de armazenamento e de mensagens de controle.

Os algoritmos desenvolvidos estão aplicados no contexto das RSSF, pois essa classe de redes apresenta diversos desafios, o que torna essas redes um objeto de pesquisa e de grande interesse tanto na comunidade acadêmica quanto industrial. Acredita-se que os algoritmos aqui propostos possam ser utilizados tanto como base para futuras pesquisas, quanto para produções de aplicações finais em RSSF, pois os algoritmos não assumem quaisquer características específicas dos dispositivos e apresentam arquitetura adaptável, o que facilita os seus ajustes para aplicações específicas.

1.1.1 Objetivos Específicos

Este trabalho tem como objetivos específicos efetuar um conjunto de experimentos sobre os protocolos propostos. As análises realizadas sobre os algoritmos têm como finalidade validá-los e mostrar o comparativo das técnicas com o estado da arte. Abaixo são listados os objetivos específicos de cada algoritmo proposto:

- **CRAL:** para analisar o algoritmo foi criada uma série de testes para mostrar a confiabilidade, eficiência, robustez, flexibilidade e qualidade da árvore de coleta de dados. As duas versões do algoritmo (CRAL-LD e CRAL-FD) são comparadas entre si e com Centrality Tree (CT) estado da arte em roteamento baseado em centralidade. Também será avaliado o compromisso do uso de regras para mitigar o consumo de energia dos nós mais centrais usando *CRAL + Policy Aware*;
- **XCTP:** o protocolo XCTP passou por uma sequência de experimentos por simulação. Destes experimentos avaliou-se o favorecimento à construção de protocolos de transporte de dados confiáveis, robustez na presença de falhas e diferentes cenários, escalabilidade, custo para manter rotas e consumo de memória. XCTP foi comparado com os mais atuais protocolos de roteamento disponíveis na literatura sendo eles: Ad hoc On Demand Distance Vector (AODV), CTP e Routing Protocol for low-power and lossy networks (RPL). Estes experimentos visam validar e mostrar que XCTP é competitivo e pode ser considerado como uma alternativa para aplicações que necessitem de comunicação nos sentidos nós sensores para EBs e vice-versa.

1.2 Motivação

Coleta de dados é uma funcionalidade fundamental para diversas aplicações que operam sobre as RSSF [Krishnamachari et al., 2008; Boukerche et al., 2014]. Estas aplicações podem ser de variados tipos tais como: imagem, vídeo, áudio e/ou simplesmente dados. Para essas aplicações a coleta de dado eficiente é primordial, visto que não é desejável, por exemplo, que os dados não sejam entregues ao coletor. Em outras situações se deseja que um nível de garantia de entrega e/ou velocidade na entrega desses dados. Esses requisitos impõe a necessidade de protocolos de coleta de dados para RSSF que alcance esses objetivos.

Devido às limitações computacionais e energia da maioria das plataformas de RSSFs, fica evidente que o encargo do uso adequado destes recursos também deve ser tratado ao nível de software. Além disso, tem-se observado que a maioria dos

algoritmos de roteamento usam técnicas tradicionais (oriundas das redes com cabo) para rotear mensagens, estas técnicas geralmente não são adequadas para o ambiente sem fio, o que deprecia a eficiência da rede e que frequentemente necessita de grandes quantidades de memória, o que geralmente as tornam inviáveis em redes de sensores de grande escala. Portanto, soluções que pondere características do ambiente sem fio e das limitações dos dispositivos devem ser propostas.

Neste sentido, CRAL apresenta uma abordagem flexível para contornar as questões acima citadas. Além disso, CRAL facilita o uso de técnicas de fusão de dados, pois seu roteamento é baseado na centralidade dos nós e suas rotas são escolhidas através de métricas específicas para RSSF.

Outra importante motivação deste trabalho é a falta de protocolos que permitam comunicação bidirecional entre os nós sensores e as EBs. XCTP resolve este problema de comunicação e amplia o escopo das possíveis aplicações para RSSFs. O protocolo XCTP permite que mensagens sejam trocadas em ambas as direções entre o nó sensor e a estação base que estão se comunicando. Sendo assim, é possível utilizar o serviço de roteamento para construir protocolos de transporte de dados confiáveis fim-a-fim.

1.3 Contribuições

O estudo dos problemas de roteamento em RSSFs, o desenvolvimento de soluções para os problemas abordados e a avaliação dos protocolos propostos trazem as seguintes contribuições:

1. Proposta do CRAL-*Low Drop e Fast Delivery* como algoritmos para roteamento baseado em centralidade e qualidade do enlace para RSSF;
 - Uma avaliação da confiabilidade, eficiência, robustez e flexibilidade dos algoritmos CRAL-LD e CRAL-FD em comparação com Shortest Paths Tree (SPT) e CT (o estado da arte em roteamento baseado em centralidade);
 - Avaliação de diferentes funções de fusão de dados em conjunto com CRAL-LD e CRAL-FD;
2. Proposta do algoritmo Policy Aware para mitigar o problema do maior consumo de energia dos nós mais centrais da rede;
3. Proposta do eXtend Collection Tree Protocol, o qual permite roteamento de mensagens na direção reversa ao sentido padrão do protocolo CTP. XCTP faz isso usando pouca memória adicional para armazenar as rotas reversas;

- Análise de desempenho do protocolo XCTP, AODV, CTP e RPL. Nos experimentos, XCTP mostrou-se mais confiável, eficiente, ágil e robusto;
4. Proposta do protocolo Transport Automatic Piggyback Protocol (TAP2), um protocolo de transporte de dados confiável sobre o protocolo XCTP.

1.4 Organização do Trabalho

A dissertação está organizada da seguinte forma. No Capítulo 2 é apresentada a fundamentação teórica do trabalho, destacando os principais paradigmas de roteamento e problemas das RSSF. Em seguida, no Capítulo 3 os trabalhos relacionados sobre centralidade, LQE e protocolos relacionados ao CRAL e XCTP são discutidos e comparados a respeito de suas características. O protocolo CRAL e seus detalhes como arquitetura, análise assintótica e avaliação de desempenho são apresentados no Capítulo 4. Logo após, no Capítulo 5, o protocolo XCTP é apresentado e avaliado em comparação com os protocolos mais atuais da literatura. Finalmente, o Capítulo 6 conclui a dissertação e descreve os trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Nesta seção serão discutidos os fundamentos básicos para maior clareza a respeito dos trabalhos aqui desenvolvidos e analisados. Inicialmente são declarados os princípios básicos para quaisquer protocolos de roteamento no contexto das RSSF. Estes princípios são metas fundamentais dos protocolos propostos neste trabalho. Em seguida, serão definidos os fundamentais paradigmas de roteamento para RSSFs existentes, bem como os principais problemas do ambiente sem fio e soluções aproximadas já consolidadas para os problemas. Logo após, centralidade e LQE são os objetos de discussão em detalhes. Finalmente, apresentamos as principais inspirações adquiridas da fundamentação teórica e que são utilizadas como base para atingir os objetivos gerais e específicos deste trabalho.

2.1 Roteamento em RSSF

Em termos gerais, o roteamento é o processo de selecionar os melhores caminhos em uma rede tornando possível o tráfego das mensagens. Em RSSF, o processo de roteamento envolve quatro requisitos, os quais são tidos como fundamentais. Estes requisitos introduzem uma série de compromissos entre si e internos ao requisito. Portanto, todo protocolo para esse tipo de rede deve ter como objetivo a manutenção do equilíbrio entre estes requisitos. Abaixo são detalhadas as definições de cada um dos requisitos para o roteamento em redes de sensores:

1. **Confiabilidade:** o protocolo deve entregar a maior quantidade de mensagens de dados, quando existe uma rota entre os participantes da comunicação;
2. **Robustez:** o protocolo deve operar em diferentes topologias, cargas, e quantidade de nós sensores e na presença de falhas na rede;

3. **Eficiência:** o protocolo deve entregar as mensagens de dados com a menor quantidade de transmissões e tempo possível, economizando energia e mantendo a menor quantidade de estados possível;
4. **Flexibilidade:** o protocolo deve permitir técnicas adicionais sejam anexadas com facilidade tais como fusão de dados, mineração de dados, aprendizado de máquinas.

Os compromissos inerentes aos requisitos e inter-requisitos é evidente. Para exemplificar, o requisito de confiabilidade deve ser ajustado para balancear de modo interno a entrega de mensagens, o consumo de energia e manutenção das rotas. De modo inter-requisitos, pode-se destacar o compromisso entre confiabilidade e eficiência, por exemplo, caso as rotas apresentem alta probabilidade de perda de mensagens, ou seja, baixa confiabilidade, isto implica em uma menor eficiência do algoritmo, pois se torna preciso que mais transmissões/retransmissões sejam realizadas para entregar as informações. Compromissos semelhantes surgem com os demais requisitos, o que conduz a demanda de protocolos construídos especialmente para lidar com esses problemas.

2.2 Paradigmas de Roteamento

Para melhor assimilação, aqui o roteamento em RSSF será comparado com o roteamento em Mobile Ad-hoc Networkss (MANETs). O roteamento em RSSF e as conhecidas MANETs apresentam diversas similaridades. Ambos os processos de seleção de rotas envolvem comunicação através de múltiplos saltos e não utilizam cabos como meio físico de transmissão. Contudo, a natureza das aplicações e requisitos de roteamento entre os dois diferem de modo significativo em vários aspectos.

Em primeiro lugar, as aplicações em RSSF usualmente necessitam de um dos três principais paradigmas de comunicação:

1. *Muitos-para-Um (Many-to-One)*: em que múltiplos nós sensores monitoram variáveis do ambiente e reportam estas informações, as quais são coletadas por EBs. Este paradigma é conhecido como coleta de dados e é o mais comum paradigma de comunicação, pois atende as necessidades de comunicação da maioria das aplicações em RSSF. Para atingir os objetivos da coleta de dados cria-se uma árvore de roteamento com raiz nas EBs e folhas nos nós sensores da rede;
2. *Um-para-Muitos (One-to-Many)*: é o paradigma conhecido como disseminação de dados, o qual tem característica inversa ao do paradigma de coleta de dados

anteriormente definido. Na disseminação de dados, comumente as EBs enviam comandos para um ou vários nós sensores da rede a fim de re-configurar ou modificar algum parâmetro dos dispositivos na rede. Para realizar a disseminação de dados pode-se, por exemplo, efetuar inundações na rede;

3. *Qualquer-para-Qualquer (Any-to-Any)*: também conhecido como Point-to-Point, esse paradigma é o mais geral possível. Ele permite que quaisquer dois nós da rede se comuniquem. Por ser mais abrangente, esta abordagem é a mais complexa e geralmente exige maior quantidade de recursos, principalmente de armazenamento, pois se faz necessário manter rotas para todos os dispositivos alcançáveis na rede.

2.2.1 Características das RSSF

Além desses paradigmas é comum que os nós sensores necessitam hora atuar em seu papel original de monitorar e reportar informações sobre variáveis do ambiente, hora atuar como roteadores cooperando no encaminhamento de dados oriundos dos demais nós sensores da rede. Portanto, as aplicações devem considerar quais as necessidades de comunicação e utilizar um algoritmo de roteamento adequado para tal paradigma.

Em segundo lugar, em RSSF quando os dados a serem coletados ou disseminados pela rede de sensores são baseados em um fenômeno ou propagação comum é frequente alguma redundância de informação trafegando na rede. Sendo assim, remover essas redundâncias trabalhando o modelo de comunicação pode ser ideal para a redução de tráfego na rede, número de transmissões e conseqüentemente do consumo de energia.

Em terceiro lugar, as RSSF apresentam pouca mobilidade quando comparada com as MANETs. Ainda assim, características do ambiente sem fio introduzem dinâmica na rede, por exemplo, segundo os autores Srinivasan et al. [2008] enlaces podem apresentar taxas intermediária de entrega e podem deixar de existir em poucos segundos. Assim, moldar adequadamente o algoritmo de roteamento é fundamental para que enlaces de alta qualidade sejam escolhidos para as rotas em detrimento de enlaces falhos.

Uma quarta questão é que a maior restrição das redes de sensores é o fator energético. Diferente das MANETs, a maioria das aplicações em RSSF, necessitam que os dispositivos estejam em um local e os analistas em outro, isto implica que os nós estão inacessíveis por longos períodos de tempo e devem se manter em operação durante o máximo de tempo possível. Além disso, é comum que a rede tenha grande escala. Sendo assim, o algoritmo de roteamento necessita ser consciente destas condições e devem lidar de modo apropriado.

Por essas razões, protocolos já construídos para MANETs não são adequados para RSSF. Deste modo, se faz necessário que protocolos de roteamento sejam construídos especialmente para RSSF a fim de extrair as características peculiares das aplicações para ajustar corretamente as rotas da rede sem fio.

2.2.2 Análise de Dados

Na seção anterior foi comentado que é frequente que dados monitorados pelos sensores apresentem alguma redundância. Quando uma aplicação para redes de sensores apresenta tal característica é possível que a análise de dados seja realizada em dois principais momentos: 1) quando todos os dados chegam ao destinatário final (geralmente a EB) e são processados de uma vez; 2) durante o roteamento dos dados, quando os nós intermediários nos sucessivos saltos da rota interceptam a mensagem, efetuam algum processamento e reencaminham a mensagem até o destinatário final.

Durante os processos acima citado, diferentes técnicas de fusão de dados, mineração de dados ou até aprendizado de máquinas podem ser empregadas para realizar a análise de dados. Neste trabalho, o foco dado é na análise de dados durante a transmissão dos dados e utilizando a fusão de dados. Grande atenção é canalizada para esta técnica pois vários benefícios podem ser obtidos dentre outros destacam-se: redução do consumo de energia, ocupação do canal de comunicação, colisões Chen et al. [2009].

A ideia da fusão de dados é combinar os dados, em rota, de diferentes nós sensores – eliminando redundância dos dados, minimizando o número de transmissões e consequentemente poupando energia. O roteamento para fusão de dados é classificado em dois grupos principais segundo Krishnamachari et al. [2008]: i) roteamento tradicional também chamado de *Address-Centric (AC)* que visa encontrar o menor caminho entre a origem e o destinatário e rotar as mensagens através deles; ii) abordagem centrada nos dados, do termo em inglês *Data-Centric (DC)*, que tem por objetivo encontrar rotas partindo dos múltiplos nós sensores para nós intermediários comuns, os quais permitem fusão de dados.

2.2.3 Roteamento Address-Centric e Data-Centric

Para demonstrar as diferenças de operação do esquema de roteamento *Data-Centric*, o qual favorece o uso de técnicas de fusão de dados e o esquema *Address-Centric* o qual não visa o uso de fusão de dados. Será assumido que a rede está operando sobre o paradigma de coleta de dados com vários nós sensores e uma estação base. Inicialmente a EB faz uma consulta/requisição de dados aos nós da rede e quando os nós sensores

possuem a informação requisitada eles respondem com os dados. A principal diferença entre os esquemas de roteamento DC e AC se dá principalmente nos caminhos que os dados seguem rota até o sorvedouro:

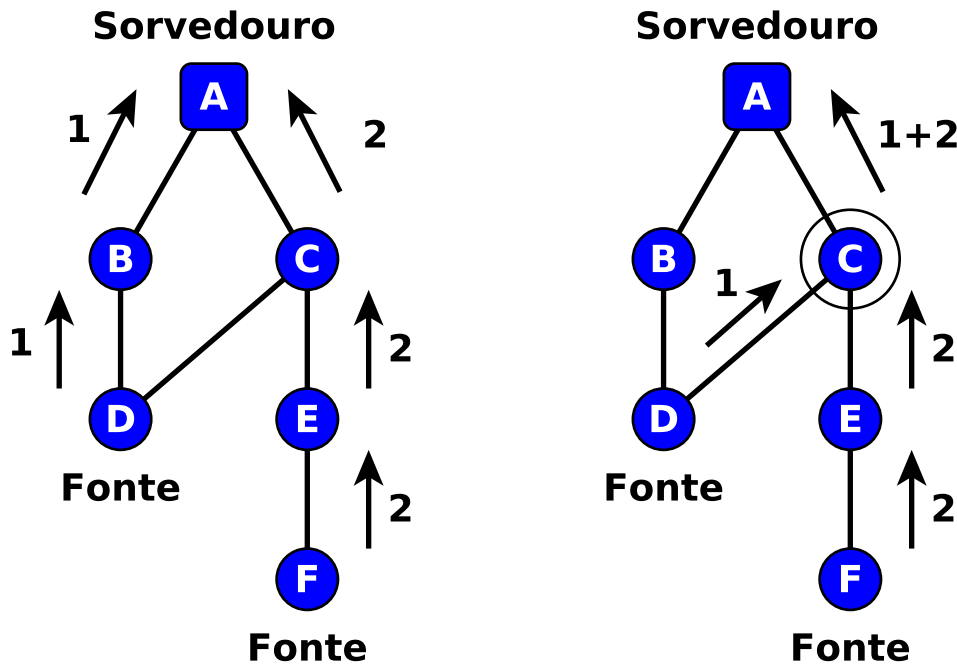
- **Roteamento Address-Centric:** cada nó sensor envia, de modo independente, dados ao longo dos menores caminhos, baseando-se na consulta/requisições, até o destinatário da mensagem;
- **Roteamento Data-Centric:** os nós sensores enviam os dados para o EB, porém os nós intermediários que interceptam as mensagens, em rota, olham o conteúdo da mensagem e então desempenham algum tipo de análise (fusão/filtragem/regressão) dos dados originários dos diferentes nós sensores.

A Figura 2.1 exemplifica a operação dos esquemas de roteamento centrado no destinatário e nos dados. Em ambas as figuras os nós D e F possuem os dados 1 e 2 respectivamente para enviar à EB (nó A). O esquema AC é mostrado na Figura 2.1a, cada informação é enviada de modo independente pelos menores caminhos até o destino, isto é, os dados oriundos em D são encaminhados por B até o destino A , de modo similar os dados de F usam a rota ECA . Na Figura 2.1b, o esquema DC busca escolher rotas dos nós fontes até o destinatário que se sobrepõem. Estas rotas favorecem que nós intermediários efetuem processamentos sobre os dados. As operações podem ser algum tipo de fusão ou mineração de dados, ou até mesmo uma técnica de aprendizado de máquina. No exemplo da Figura 2.1b, mostra a fusão de dados, com uso desta técnica os dados 1 e 2 são somados e somente uma mensagem é encaminhada para o sorvedouro. O maior benefício ao agregar mensagens para a RSSF é a redução no número de transmissões necessárias para entregar os dados e, como consequência o tempo de vida da rede é prolongado.

2.2.4 Fusão de Dados

Uma das metas deste trabalho é empregar fusão de dados em conjuntos com o protocolo de roteamento em RSSF. Entender o contexto em que a fusão de dados é útil é de fundamental importância para o desenvolvimento de protocolos de roteamento do tipo DC, deste modo é válido considerar os seguintes cenários em termos do tipo de dinâmica dos dados que são enviados aos sorvedouros:

1. Sem redundância: em que todos os nós sensores enviam dados com informações completamente diferentes;



(a) Roteamento *Address-Centric*, em que os dados são encaminhados independentemente.

(b) Roteamento *Data-Centric*, em que os nós intermediários realizam fusão de dados.

Figura 2.1: Exemplo da operação dos esquemas de roteamento centrados no destinatário centrado nos dados.

2. Completamente redundante: os nós sensores enviam informações idênticas para a estação base;
3. Com alguma redundância: as informações enviadas pelos nós sensores apresentam algum nível de redundância de modo não determinístico.

No primeiro caso, nenhuma operação de fusão de dados pode ser efetuada sobre os dados – ambos os esquemas AC e DC de roteamento obtém o mesmo número de transmissões para entregar todos os dados. No segundo caso, o esquema AC pode ser modificado para se igualar ou superar o esquema DC, dado que o sorvedouro pode analisar os dados duplicados e solicitar não a todos, mas somente um ou alguns dos nós da rede. No terceiro caso, entretanto, ao usar o esquema AC não é possível modificá-lo para ser melhor do que o esquema DC, quando os dados apresentam redundância não determinística, implicando que o sorvedouro não pode solicitar dados de específicos nós e desligar os demais. Uma vez que todos os nós sensores da rede transmitem dados com alguma redundância, a melhor opção é utilizar técnicas de fusão de dados sobre as rotas que o esquema DC constrói.

A fusão de dados é a combinação de dados de diferentes fontes, no caso das RSSF, a combinação de dados de diferentes nós sensores. O ato de combinar pode ser implementado de diferentes maneiras [Zhang et al., 2013; Chen et al., 2014; Qin & Gu, 2011; Macias et al., 2013]. Um dos modos mais simples é supressão de dados duplicados – no exemplo da Figura 2.1b se os nós fontes D e F enviam os mesmos dados, o nó C pode encaminhar somente das cópias. Outro modo de fusão de dados são as funções de max e min em que um nó intermediário pode receber um conjunto de dados como entrada e enviar somente um resultado.

2.2.4.1 Fusão de Dados Ótima

Diversos esforços foram realizados no intuito de encontrar a técnica ótima de fusão de dados para RSSF [Krishnamachari et al., 2008; Winter, 1987; Takahashi & Matsuyama, 1980; Dolev et al., 2010]. A modelagem da RSSF como um grafo é amplamente aceita e utilizada para realizar estudos sobre os diversos aspectos da rede. Com o uso de técnicas da teoria foi possível formalizar os principais limites da fusão de dados ótima no âmbito das RSSF.

Seja $G = (V, E)$ um grafo que represente a rede de sensores sem fio, onde V é o conjunto de n nós sensores, $s \in V$ é um nó especial denominado sorvedouro e E consiste das arestas entre os nós que podem se comunicar entre si diretamente. Pressupondo que o número de transmissões entre dois nós diretamente ligados na árvore de roteamento é um, uma árvore ótima para fusão de dados pode ser pensada como o inverso da árvore *multicast*, isto é, em vez de uma única fonte de envio para vários receptores, tem-se todos os nós fontes enviando informações para o mesmo receptor. É bem conhecido que a árvore *multicast* com o menor número de arestas é o problema *minimum Steiner Tree* em um grafo que represente a rede Krishnamachari et al. [2008]. Sendo assim, o seguinte resultado pode ser facilmente obtido:

Resultado 1: O número ótimo de transmissões requeridas para um esquema de roteamento DC é igual ao número de arestas na *minimum Steiner Tree* da rede, o qual contém o conjunto de nós $(v_1, v_2, \dots, v_n, s)$ Hwang et al. [1992].

Corolário: Assumindo que os nós estão dispostos de forma aleatória e um grafo G que representa a rede. A tarefa de construir um esquema de roteamento DC ótimo é NP-Difícil.

2.3 Principais Problemas de Roteamento em RSSF

O roteamento tradicional realizado em redes com cabos não é apropriado para redes de sensores sem fio. É comum no roteamento tradicional que grande quantidade de rotas para os dispositivos da rede sejam armazenadas e, que os enlaces entre os nós adjacentes apresentam baixa perda de dados. O oposto ocorre em redes sem fio e em especial nas RSSF. Atualmente, as RSSF podem contar com centenas ou milhares de nós, cada um com algum poder computacional, capacidade de sensoriamento e comunicação sem fio. Geralmente os recursos dos nós sensores são restritos, pois para manter o baixo custo dos dispositivos são empregados componentes de hardware de menor capacidade. Estas características dos elementos de rede colocam a tarefa de rotear mensagens como uma das mais desafiadoras no âmbito das RSSF. Estas restrições provocam diversos compromissos na tarefa de rotear mensagens tais como:

- Administrar o número de mensagens de controle ao passo que mantém baixo o consumo de energia do nó;
- Reagir de modo ágil às mudanças topológicas providenciando rotas sempre que os dispositivos são alcançáveis;
- Manter os caminhos para rotear as mensagens nos nós sensores utilizando a mínima quantidade de estados possíveis;
- Prover mecanismo de roteamento ao passo que não impõe requisitos exagerados de hardware dos dispositivos.

Estes e outros compromissos surgem com frequência no processo de seleção das rotas em RSSF. Sendo assim, os algoritmos de roteamento devem equilibrar esses compromissos a fim de não consumir todos os recursos do dispositivo.

Além desses compromissos oriundos das limitações de hardware, existem complicações adicionais devido ao ambiente sem fio em que essas redes operam. Para que se construam boas rotas em uma rede sem fio é necessário que se levem em conta alguns fatores que impactam diretamente no desempenho de toda a rede: atenuação, distorção, ruídos, efeito *multipath*, absorção atmosférica [Moscibroda et al., 2006; Jain, 2007; Camp et al., 2002; Jain et al., 2005].

2.3.1 Modelagem do Ambiente Sem Fio

Diante dos diversos desafios colocados pelas RSSF vários modelos foram propostos para estudar e viabilizar a construção de protocolos para essas redes. Tal como descrito

anteriormente, as RSSF são modeladas como grafos e usando a teoria acerca dos grafos estudos são realizados para compreender o comportamento da rede. A modelagem das RSSF é constituída de diversas partes, as quais permitem estudos refinados. Estas partes viabilizam a construção de simuladores e/ou módulos para os nós sensores. A seguir é descrita a modelagem do ambiente sem fio para este trabalho:

- *Modelo de Conectividade:* modelo que decide quais são os vizinhos um nó possui e insere/remove arestas se existem ou não conexões entre eles. Os modelos mais conhecidos são Unit Disk Graph (UDG) Clark et al. [1991] e Quasi Unit Disk Graph (QUDG) Kuhn et al. [2004];
- *Modelo de Interferência:* é o modelo que decide quais mensagens podem ser interpretadas, isto é, se a mensagem foi recebida com sucesso ou não devido a interferências causadas por outros pacotes, ruídos do ambiente, distorções, atenuações ou efeito *multipath*. O modelo mais representativo desta categoria é um modelo Signal-to-Interference-plus-Noise Ratio (SINR);
- *Modelo de Confiabilidade:* estes modelos descrevem o nível de confiabilidade dos enlaces, geralmente utilizando fundamentos probabilísticos. Este nível de confiança dos enlaces infere a respeito das perdas de mensagens. Assim, representado a decodificação ou não das mensagens que podem não serem decodificadas por diversas razões tais como: falha de transmissão, ruídos alteram bits, perda de conectividade dentre outros;
- *Modelo de Transmissão:* modelo que determina o tempo que uma mensagem necessita até alcançar o nó destinatário;
- *Modelo de Mobilidade:* descreve como os nós sensores se movem na região a ser monitorada. Como destaques tradicionais desse modelo tem-se *Random Way Point Model* e *Random Direction* Camp et al. [2002].

Uma das principais dificuldades no desenho dos protocolos de roteamento para RSSF é escolher rotas sem fio que apresentem enlaces de boa qualidade. Para tanto se fez necessária à caracterização dos enlaces e a construção de modelos que permitam inferir a qualidade dos enlaces. Assim, deu-se início a pesquisas sobre a qualidade dos enlaces e a criação de estimadores para qualificá-los, os quais são chamados de Link Quality Estimation (LQE). Estes LQE são utilizados pelos protocolos de roteamento para deduzir informações sobre o estado dos enlaces e inferir melhores rotas. A próxima seção apresenta mais detalhes sobre tais estimadores.

2.3.2 Estimadores da Qualidade do Enlace

Nos últimos anos, pesquisas sobre LQE tem se tornado um grande atrativo Baccour et al. [2012]. Isto porque o impacto desses estimadores é grande sobre o desempenho dos protocolos de roteamento em RSSF. Estes protocolos com informações dos LQEs agora podem reagir adequadamente perante a dinâmica das redes sem fio. A causa da dinâmica da rede é explicada por diversos fatores como atenuação, reflexão, difração, ruídos e outros que contribuem para a degradação da qualidade do sinal sem fio que é propagado. O efeito desses fatores é ainda mais significante sobre a propagação de sinais sem fio quando os rádios utilizados são de baixo custo (*low-power radios*¹), comumente empregados em L2Ns. Como consequência, os enlaces sem fio das L2Ns são imprevisíveis e a sua qualidade varia sobre o tempo Srinivasan et al. [2010a] e espaço [Zhou et al., 2004; Reijers et al., 2004], além disso é frequentemente que a conectividade apresenta assimetria [Cerpa et al., 2005a; Zhou et al., 2004; Awerbuch et al., 2004], ou seja, os enlaces apresentam características distintas entre as conexões A – B e B – A.

Estimadores de enlace em RSSF é um item fundamental para um desenho eficiente de sofisticados protocolos de roteamento. LQEs geralmente estão localizados entre as camadas de enlace e redes, estes estimadores visam qualificar os enlaces e manter o protocolo de roteamento informado sobre as mudanças de estado dos enlaces.

Um LQE basicamente consiste em uma métrica, isto é, uma expressão matemática, a qual é utilizada dentro de uma janela de tempo w (por exemplo, a cada w segundos ou baseado em w recepção/envio de mensagens). O processo de estimar a qualidade de um enlace envolve três etapas: i) monitoramento do enlace; ii) medição do enlace; iii) avaliação da métrica. As três etapas são detalhadas a seguir e ilustradas na Figura 2.2:

1. Monitoramento do enlace: define uma estratégia sobre o tráfego no enlace permitindo medições sobre o mesmo. Esta etapa divide-se em três tipos:
 - a) Monitoramento ativo: os nós monitoram os enlaces para seus vizinhos enviando pacotes de sonda (*beacons*). Estes pacotes podem ser enviados ou por broadcast ou unicast. O monitoramento ativo necessita que pacotes de sonda sejam trocados entre os nós vizinhos e que mensagem de confirmação (ACK – *acknowledgments*) seja enviada pelo receptor. A periodicidade é uma característica dessa técnica de monitoramento, isto é, os beacons são enviados em certa frequência. Esta característica implica em um compromisso entre consumo de energia (baixa taxa) e precisão sobre a qualidade

¹Dáí o termo *low-power and lossy networks* (L2Ns).

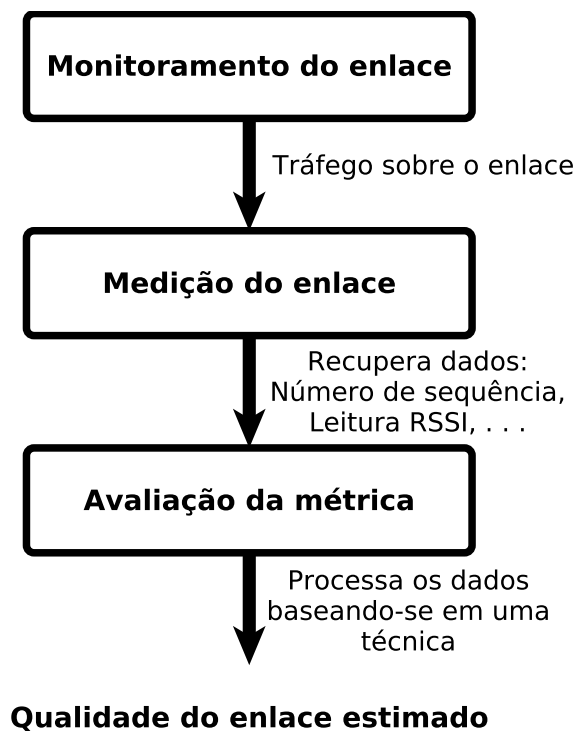


Figura 2.2: Processo para estimar a qualidade do enlace Baccour et al. [2012].

dos enlaces (alta taxa). *Adaptive beaconing rate* Levis et al. [2003b] é uma alternativa para balancear este compromisso;

- b) Monitoramento passivo: explora o tráfego existente sem custo adicional de envio de sondas, o que o torna diferente do monitoramento ativo;
 - c) Monitoramento híbrido: utiliza um mecanismo híbrido combinando ambos tipos monitoramentos o ativo e o passivo. Por ser uma mescla das técnicas anteriormente apresentadas, com o monitoramento híbrido é possível realizar um melhor balanceamento entre qualidade dos enlaces atualizados e economia de energia. Por exemplo, o LQE *Four bit* Fonseca et al. [2007], também utilizado no protocolo CTP, emprega monitoramento híbrido, isto é, quando não existe tráfego na rede beacons são utilizados, ao passo que na existência de tráfego os próprios pacotes de dados são oportunisticamente utilizados para inferir informações sobre os enlaces;
2. Medição do enlace: estas medidas são realizadas ao recuperar informações úteis de: i) pacotes de dados ou de confirmação recebidos; ii) mensagens enviadas.
- Dados recuperados de pacotes de dados ou de confirmação recebidos tais como número de sequência, timestamp, Received Signal Strength Indication (RSSI)

são usados para computar a qualidade do enlace no lado do receptor. Por outro lado, dados recuperados das mensagens enviadas tais como número de sequência, timestamp, número de retransmissões do pacote são utilizados para computar o estimador de qualidade de enlace do lado do emissor;

3. Avaliação da métrica: baseando-se nas medidas dos enlaces, uma métrica é utilizada para produzir um estimador de qualidade do enlace. Em geral, a métrica é projetada de acordo com uma técnica de aproximação que pode ser uma simples média ou uma técnica mais sofisticada como filtragem, regressão ou lógica fuzzy.

2.4 Centralidade e Estimadores de Qualidade do Enlace

Inicialmente será realizada uma breve apresentação sobre os trabalhos relacionados sobre centralidade e estimadores de qualidade do enlace. Estes trabalhos servem apoio para a construção das estruturas de roteamentos dos protocolos CRAL e XCTP. A princípio serão abordados trabalhos sobre centralidade e posteriormente os trabalhos sobre LQEs. Destas referências, será dado enfoque a centralidade *Sink Betweenness* e estimadores ETX e ETT utilizados pelo algoritmo CRAL, além do LQE *Four bit* utilizado no XCTP.

2.4.1 Centralidade

Em teoria dos grafos e em análises de rede, o termo *centralidade* refere-se a um indicador, o qual é utilizado para identificar quais vértices são os mais importantes dentro da rede. As técnicas de centralidade possuem ampla aplicabilidade incluindo identificação de pessoas influentes em uma rede social Lee & Kim [2010], no escopo da Internet, Jackson et al. [2007] sugere que monitores de tráfego sejam colocados em enlaces no nível dos Sistemas Autônomos (AS) da Internet, escolhendo nós com alto grau como critério de escolha. A característica de centralidade dos nós também é importante no campo de espalhamento de doenças ou epidemiologia computacional, onde se tem mostrado que imunizando nós centrais o impacto da epidemia pode ser significativamente reduzido Pastor-Satorras & Vespignani [2002].

O índice de centralidade pode ser utilizado para responder questões como "O que caracteriza um vértice importante?". A resposta para tal questão pode ser dada através do valor de uma função sobre os vértices do grafo. Espera-se que o resultado

produzido pela função forneça uma lista ordenada de vértices (*ranking*) que identifica os nós mais importantes.

A palavra "importância" apresenta um amplo número de significados, o que implica em diferentes definições para centralidade. Por exemplo, um possível critério de importância pode ser o tipo de fluxos que passa pelos nós da rede, deste modo classificam-se que tipo de fluxos são mais importantes. Outro exemplo de importância ser concebida como o envolvimento dos nós com os demais numa rede, desta forma a centralidade pode ser classificada com base em uma medida de coesão Borgatti & Everett [2006]. Já neste trabalho, a importância de um nó é definida como *a quantidade de participações do nó nos caminhos dos demais nós da rede*.

Diversas medidas de centralidade tais como *Degree*, *Closeness* e *Betweenness* foram apresentadas para analisar redes e entender a dinâmica global da rede e as regras que regem os nós individualmente [Borgatti & Everett, 2006; Freeman, 1977; Sabidussi, 1966; Freeman, 1979]. Para as medidas de centralidade descritas a seguir, subentende-se que a centralidade é aplicada sobre um grafo $G = (V, E)$, onde V são os vértices do grafo e E são as arestas, em certos casos definições adicionais são descritas.

- *Centralidade de Grau (Degree Centrality)*: em teoria dos grafos, o grau de um vértice de um grafo é o número de arestas incidentes no vértice, no caso de grafos direcionados usualmente definem-se dois graus separados. O primeiro chamado de *grau de entrada (indegree)* para arcos de possuem cabeça no nó a ser verificado e *grau de saída (outdegree)* para arcos que possuem cauda no nó.

Formalmente, a centralidade de grau $CD(v)$ de um vértice v , para um dado grafo é definida como:

$$CD(v) = \frac{deg(v)}{N - 1} \quad (2.1)$$

onde $deg(v)$ é o grau do vértice e N é o número de vértices no grafo;

- *Centralidade de Proximidade (Closeness Centrality)*: esta medida representa a noção de proximidade, isto é, a distância natural entre todos os pares de nós na rede, definida como o comprimento dos menores caminhos. Existem várias definições desta medida sendo que as duas principais são:

1. A distância geodésica média: o Closeness de um vértice $v \in V$ aos demais vértices é dado pela Equação 2.2 que descreve formalmente a centralidade

para esta definição:

$$\tilde{C}_{CL}(v) = \frac{\sum_{u \neq v} d_{v,u}}{N-1} \quad (2.2)$$

onde $d_{v,u}$ é a distância geodésica entre os vértices v e u , N é o número de vértices no grafo. No caso da rede ser desconexa $\tilde{C}_{CL}(v) = \infty, \forall v \in V$;

2. A Recíproca: em que se aplica a soma total das distâncias geodésicas de um vértice v aos outros vértices:

$$C_{CL}(v) = \frac{1}{\sum_{u \neq v} d_{v,u}} \quad (2.3)$$

no caso da rede ser desconexa $C_{CL}(v) = \frac{1}{\infty} = 0, \forall v \in V$;

- *Excentricidade*: esta medida informa a distância máxima entre um vértice v e qualquer outro vértice u do grafo. Formalmente, o E_{cc} para um vértice v a calculado seguindo a equação:

$$E_{cc}(v) = \mathbf{max}_{u \in V}(d_{v,u}) \quad (2.4)$$

os vértices com menor E_{cc} serão os mais centrais. Além disso, o valor máximo e mínimo do E_{cc} serão respectivamente o diâmetro e o raio da rede;

- *Centralidade Rank de Páginas (Page RankTM)*: o Page RankTM foi criado baseado no comportamento de usuários que navegam pelas páginas da Internet. O modelo considera que um usuário fica entediado e vai para uma página aleatória com probabilidade $1 - d$:

$$PR(v) = \frac{1-d}{N} + d \sum_{u \in B_v} \frac{PR(u)}{L(u)} \quad (2.5)$$

onde B_v é o conjunto de páginas que possuem enlaces apontando para v , $L(u)$ é o número de *hyperlinks* que apontam para u e por estudos realizados geralmente se assume o valor de $d = 0.85$;

- *Centralidade Stress (Stress Centrality)*: esta medida enumera os caminhos mínimos (geodésicos) para efetuar o *ranking* dos nós da rede. A Equação 2.6 mostra como a Centralidade Stress C_s é calculada para um nó $v \in V$ do grafo $G = (V, E)$:

$$C_s(v) = \sum_{s \neq v} \sum_{t \neq v} \sigma_{s,t}(v) \quad (2.6)$$

onde $\sigma_{s,t}(v)$ é a quantidade caminhos mínimos que começam no nó s e vão até o destino t passando pelo nó v ;

- *Centralidade Betweenness (Betweenness Centrality)*: esta medida quantifica o número de vezes que um nó atua como ponte ao longo dos menores caminhos entre quaisquer dois outros nós. Originalmente apresentado por Freeman [1977] com o intuito de quantificar o controle de uma pessoa sobre a comunicação entre outras pessoas em uma rede social. Formalmente, o *Betweenness* de um vértice v é computado da seguinte forma:

$$C_B(v) = \sum_{s \neq v} \sum_{t \neq v} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}} \quad (2.7)$$

onde $\sigma_{s,t}$ é o número total de menores caminhos de s até t e $\sigma_{s,t}(v)$ é a quantidade caminhos mínimos que começam no nó s e vão até o destino t passando pelo nó v . O *Betweenness* apresenta diferentes versões tais como a versão normalizada que divide o *Betweenness* pelo número de pares de vértices que não incluem v , versões para grafos direcionais e a mais interessante para este trabalho é a versão destacada a seguir:

- *Sink Betweenness*: versão criada especialmente para cenários das RSSF, em que a comunicação se dá partindo dos nós sensores para a estação base e vice-versa. Para capturar este comportamento a medida *Betweenness* foi alterada para considerar um sorvedouro como nó terminal. Formalmente, o Sink Betweenness Centrality de um vértice v é definido como:

$$SBC(v) = \sum_{i \in \psi_v} \frac{\sigma_{v,s}}{\sigma_{i,s}} \quad (2.8)$$

onde s é o sorvedouro, $\sigma_{v,s}$ é o número de menores caminhos de v para o sorvedouro, $\sigma_{i,s}$ é o número de menores caminhos de i para o sorvedouro, sempre que $\psi_v = \{i \in V | v \in SP_{i \rightarrow s}\}$ seja satisfeito. $SP_{i \rightarrow s}$ é o conjunto de todos os menores caminhos de i até o s , ou seja, ψ_v é o conjunto de todos os nós que contenham v em, pelo menos, um de seus menores caminhos até o sorvedouro. No próximo capítulo a discussão sobre a centralidade *Sink Betweenness* será retomada através de exemplos de cálculo e caso de uso.

Na Tabela 2.1 as centralidades acima descritas são comparadas em termo do requisito de importância exigido pelo algoritmo CRAL, ou seja, a centralidade deve

capturar o comportamento da coleta de dados em RSSF, efetuando a contagem das participações de um determinado nó nos caminhos dos demais nós da rede. Diante deste requisito, das medidas de centralidade apresentadas na Tabela 2.1, o grupo de centralidades *Stress*, *Betweenness* e *Sink Betweenness* se enquadram no critério de importância demandado.

Do grupo que apresenta a propriedade demandada, a centralidade *Stress* não pondera os caminhos pelos caminhos mínimos existentes entre os nós s e t , o que pode acarretar em supervalorização de alguns nós. Por outro lado, a centralidade *Betweenness* obtém todas as características da centralidade *Stress* e vai além ao ponderar todos os caminhos mínimos por $\sigma_{s,t}$ (vide Equação 2.7), sendo ideal para a construção de protocolos de roteamento que visam identificar nós que são frequentemente usados como roteadores. No que diz respeito à centralidade *Sink Betweenness*, esta medida se molda aos comuns cenários de comunicação das redes de sensores, em que se utilizam prioritariamente o paradigma Muitos-para-Um. Este trabalho usa a centralidade *Sink Betweenness*, pois suas características a colocam como medida apropriada para a construção de rotas em RSSF.

Tabela 2.1: Comparação entre os tipos de centralidade

Tipos de Centralidade	Característica desejada: Contar o número de participações do nó nos menores caminhos dos demais nós da rede
Degree	–
Closeness	–
Excentricidade	–
Page Rank	–
Stress	✓
Betweenness	✓
Sink Betweenness	✓

2.4.2 Estimadores de Qualidade do Enlace

Estimadores de enlace para RSSF podem ser classificados em duas grandes categorias: i) LQEs baseados em software; ii) LQEs baseados em hardware. Na Tabela 2.2² são apresentados os principais estimadores de enlace, além da diferenciação entre as grandes categorias, os LQEs também são qualificados segundo as propriedades definidas no Capítulo 2 na Subseção 2.3.2, sendo elas: i) Técnica para avaliação da métrica; ii) Suporte a assimetria dos enlaces; iii) Tipo de monitoramento.

²Tabela inspirada nos trabalhos [Xu & Lee, 2006a; Baccour et al., 2009, 2012; Javaid et al., 2009].

Tabela 2.2: Comparativo entre estimadores de enlace

			Técnica	Suporte a assimetria	Monitoramento
Baseado em Software	Baseado em PRR	PRR	Média	–	A ou P
		WMEWMA	Filtragem	–	P
		KLE	Filtragem	–	–
	Baseados em RNP	RNP	Média	–	P
		LI	Probabilidade	–	P
		<i>ETX</i>	Média	✓	A e P
		<i>ETT</i>	Média	✓	A e P
	Baseados em Score	<i>Four bit</i>	Filtragem	✓	A ou P
		WRE	Regressão	–	P
		MetricMap	Treino e classificação	–	P
		F-LQE	Lógica Fuzzy	✓	P
Baseado em Hardware	RSSI, LQI e SNR	Média das leituras do rádio	–	A ou P	

*A para Ativo

*P para Passivo

2.4.2.1 Estimadores Baseados em Hardware

Três estimadores são pertencentes ao grupo dos LQE baseados em hardware: RSSI, Link Quality Indication (LQI) e Signal-to-Noise Ratio (SNR). Como técnica para estimar a qualidade do enlace esses LQEs efetuam leituras diretas sobre o rádio (exemplo CC2420 Instruments [2006], Atmel AT86RF230 Power [2010]). A principal vantagem no uso dos estimadores baseados em hardware é que não se faz necessária qualquer computação adicional para qualificar os enlaces em termos de Packet Reception Ratio (PRR)³. Abaixo são relacionados e discutidos cada um dos estimadores:

1. *RSSI*: este estimador é capaz de fornecer rápida e precisa avaliações sobre o enlace, caso ele apresente ótima qualidade Srinivasan et al. [2006]. RSSI é bastante estável, somente de uma leitura do rádio (durante a recepção de um pacote) é suficiente para determinar se o enlace apresenta boa qualidade;
2. *LQI*: este indicador pode determinar se o enlace é de ótima qualidade ou não, entretanto LQI não avalia bem enlaces de qualidade intermediária. Proposto durante a padronização IEEE 802.15.4 [2003], o LQI usa os primeiros oito símbolos de um pacote recebido para gerar uma pontuação para o enlace que varia de 50 a 110 (maiores valores são melhores). É argumentado em Srinivasan et al. [2010b], que altos valores de LQI implicam que o enlace apresenta boa qualidade, além disso quando LQI apresenta baixa variância uma leitura é suficiente para decidir se o enlace é de qualidade boa ou ruim. Porém, quando a variância é alta, isto é,

³PRR é a razão entre o número de pacotes recebidos com sucesso e o número de pacotes transmitidos.

os enlaces apresentam qualidade intermediária implica que são necessárias várias leituras para que o enlace seja qualificado precisamente;

Dos argumentos e conclusões apresentadas pelos autores [Srinivasan et al., 2006, 2010b; Tang et al., 2007; Srinivasan & Philip, 2006] o estimador LQI é um melhor indicador do PRR do que RSSI, pois em média LQI apresenta maior correlação com PRR, quando comparado com o RSSI. Por outro lado, Srinivasan et al. [2010b] alegam que RSSI apresenta a vantagem de ser mais estável do que LQI;

3. *SNR*: se mostra um bom indicador e preditor para PRR, porém não é preciso, especialmente quando os enlaces são de qualidade intermediária. Quando os enlaces são de boa qualidade SNR apresenta bom indicador para PRR, entretanto estudos [Senel et al., 2007; Ma, 2005] mostram que o relacionamento teórico entre SNR e PRR é difícil, além de ser altamente dependente de hardware e condições do ambiente como temperatura;

SNR apresenta melhor estimador de qualidade quando comparado com RSSI. O RSSI analisa somente o sinal recebido e o ruído no receptor do pacote. Por outro lado, SNR descreve o quão forte e puro é o sinal recebido, além de considerar o ruído dos diferentes nós, deste modo SNR pode ser uma métrica melhor que RSSI Srinivasan et al. [2010b]. Os autores Senel et al. [2007] mostraram que SNR não pode ser utilizado sozinho como estimador para PRR, porém pode ser útil para em conjunto com outro LQE encontrar enlaces com ótimo PRR de modo mais preciso.

Os estimadores baseados em hardware também apresentam limitações, sendo elas: i) estas medidas somente consideram pacotes recebidos com sucesso, deste modo quando existem sucessivas perdas de pacotes, o LQE pode superestimar os enlaces; ii) apesar da abordagem em hardware fornecer rápidas e precisas avaliações sobre os enlaces de modo a classificá-los como bons ou ruins, eles não são capazes de fornecer estimadores para enlaces de qualidade intermediária de modo preciso [Fonseca et al., 2007; Senel et al., 2007; Srinivasan et al., 2010b, 2006]. Por estas razões, LQEs baseados em software foram propostos. Estes LQE implementados por software permitem uma maior flexibilidade levando em consideração assimetria dos enlaces, mais sofisticados métodos de avaliação dos enlaces e captura das características inerentes do roteamento em ambientes sem fio. Apesar das limitações dos LQEs baseados em hardware isto não significa que estes estimadores não são úteis. A combinação de LQEs baseados em software e hardware se mostra bastante eficiente como é o caso do estimador *Four*

bit (discutido na Seção 2.4.2.2) que mescla informações de diferentes camadas para qualificar os enlaces.

2.4.2.2 Estimadores Baseados em Software

LQEs baseados em software podem ainda ser subdivididos em três categorias: i) Baseados em PRR, que realizam contagem ou aproximação do Packet Reception Ratio; ii) Baseados em RNP, que ou contam ou aproxima o Required Number of Packet retransmissions (RNP); iii) Baseados em *Score*, os quais fornecem uma pontuação identificando a qualidade dos enlaces.

Baseados em PRR. *PRR* é um estimador que fica localizado no lado do receptor dos pacotes. PRR foi amplamente utilizado em protocolos de roteamento [Jiang et al., 2006; Couto et al., 2005; Philip Buonadonna, 2007]. Além disso, PRR também é bastante utilizado para avaliar a precisão de LQEs baseados em hardware, onde estimadores altamente correlatos com PRR são considerados bons estimadores. A eficiência do PRR é alta quando se consegue ajustar o tempo de janela de análise da métrica.

Window Mean with Exponentially Weighted Moving Average (WMEWMA) foi apresentado pelos autores Woo & Culler [2003]. Tal como PRR, WMEWMA também é caracterizado por sua localização no lado do receptor dos pacotes, além disso apresenta monitoramento passivo (não usa pacotes de sonda para estimar o enlace). Os autores argumentam que WMEWMA aproxima-se do PRR, sendo altamente reativo a mudanças de topologia, preciso em avaliar o enlace, estável quando os enlaces apresentam baixa variação e eficiente pois utiliza baixa quantidade de memória.

Kalman-filter-based Link quality Estimator (KLE) Senel et al. [2007] foi proposto para atacar o problema de baixa reatividade dos LQEs baseados em software e que utilizam média como técnica de avaliação de qualidade do enlace. A ideia por trás do KLE é avaliar a qualidade do enlace com somente uma recepção de pacote, ao contrário de esperar a recepção de uma determinada quantidade de pacotes. Ao receber um pacote, o Received Signal Strength (RSS) é extraído e utiliza-se o filtro de Kalman para produzir uma estimação do RSS. Através de experimentos com dois nós (um receptor e um emissor), os autores provaram que a técnica é capaz de detectar mudanças da qualidade do enlace de modo mais rápido que PRR, contudo a precisão não foi avaliada.

Baseados em RNP. O *Required Number of Packet retransmissions (RNP)* apresentado por Cerpa et al. [2005b] é um estimador localizado no lado do emissor que conta a média do número de pacotes transmitidos/retransmitidos requeridos antes de uma

recepção de pacote com sucesso. RNP pode ser calculado como o número de transmissões e retransmissões de pacotes durante uma janela de tempo, dividido pelo número de pacotes recebidos com sucesso menos 1 (exclui-se a transmissão do primeiro pacote). O estimador RNP assume que é usado um protocolo Automatic Repeat Request (ARQ) no nível de enlace, que faz com que o nó emissor retransmita o pacote que não foi recebido corretamente. Uma métrica similar ao RNP é o Acknowledgment Reception Ratio (ARR), que é computada como a razão entre o número de ACK recebidos pelo número de pacotes transmitidos durante uma janela de tempo.

Link Inefficiency (LI) proposto pelos autores LaI et al. [2003] como uma aproximação para o RNP. Como principais características do LI destacam-se monitoramento do tipo passivo e utiliza técnica de probabilidade para avaliar a qualidade dos enlaces. Os autores definiram Packet Success Probability (PSP) como uma aproximação para o PRR. Entretanto, trabalhos posteriores [Senel et al., 2007; Aguayo et al., 2004] mostraram que usar PSP ao invés de PRR pode não ser um LQE adequado e preciso.

O *Expected Transmission Count (ETX)* Couto et al. [2005] é um estimador localizado do lado do receptor que usa monitoramento ativo. ETX é definido como $ETX = \frac{1}{d_f \times d_r}$, onde d_f é a probabilidade do pacote chegar com sucesso ao destinatário, já d_r é a probabilidade de um pacote ACK ser recebido com sucesso. ETX apresenta monitoramento ativo ou passivo e leva em conta assimetria nos enlaces. Em Couto et al. [2005] os autores mostram que protocolos de roteamento baseados na métrica ETX fornecem rotas de alta vazão em redes sem fio com múltiplos saltos, uma vez que ETX minimiza o número total esperado de transmissões requeridas para entregar um pacote ao destino. Couto ainda destaca que ETX pode ser utilizado para fazer decisões precisas entre rotas, pois penaliza rotas com mais saltos, visto que essas rotas possuem menor vazão devido a interferências nos sucessivos saltos.

Expected Transmission Time (ETT) Draves et al. [2004] é uma variação do ETX. ETT mantém todas as características do ETX e vai além possibilitando o ajuste da taxa de transmissão e do tamanho das mensagens de dados. ETT pode ser calculado de acordo com uma das Equações (2.9, 2.10 e 2.11) exibidas a seguir:

$$ETT_l = ETX_l \times T \quad (2.9)$$

$$ETT_l = ETX_l \times \frac{S_F}{B_l} \quad (2.10)$$

$$ETT_l = ETX_l \times \frac{S_F}{S_L/(T_S - T_L)} \quad (2.11)$$

onde S_F o tamanho fixo do pacote de dados, B_l é a largura de banda do enlace l , S_L é a largura do pacote de dados, $T_S - T_L$ é o intervalo de tempo entre a chegada de dois pacotes consecutivos e T é o tempo de envio de um pacote sobre o enlace l .

Four bit foi projetado para ser utilizado por protocolos de roteamento e fornecer 4-bits de informações Fonseca et al. [2007]. Estas informações são coletadas e compiladas de diferentes camadas: *white bit* vindo da camada de hardware que permite rapidamente identificar bons enlaces, ACK bit obtido da camada de enlace indicando se a confirmação foi recebida para um pacote enviado. *Pin* e *Compare* bits são da camada de rede e são utilizados no gerenciamento da tabela de vizinhança do nó. Fonseca avaliou o desempenho do *Four bit* em conjunto com o protocolo CTP obtendo melhores resultados do que o até então estado da arte MultiHopLQI Philip Buonadonna [2007].

Baseados em Score. Ao contrário dos LQEs anteriormente discutidos, estimadores baseados em *score* fornecem uma pontuação definida dentro de uma janela para qualificar os enlaces.

Weighted Regression Estimator (WRE) proposto por Xu & Lee [2006b] mostram como explorar a correlação entre a força do sinal recebido (RSS) e a distância entre os participantes da comunicação. WRE efetua uma regressão baseada em um vetor de entrada que contém as localizações junto com a qualidade dos enlaces. A função refina o vetor de entrada continuamente. Um problema desta abordagem é que mudanças bruscas na posição dos nós não são capturadas facilmente. Como característica favorável destaca-se o monitoramento passivo.

MetricMap foi proposto por Wang et al. [2007] como alternativa LQE para o protocolo de roteamento MintRoute Woo et al. [2003a]. MetricMap usa um algoritmo de classificação para rotular enlaces em grupos (por exemplo, enlaces no grupo bom ou no grupo ruim). O algoritmo possui uma fase de treinamento, na qual utiliza uma base de dados contendo amostras de RSSI, carga no canal, e profundidade dos nós. Como o WRE, MetricMap também usa monitoramento passivo.

Fuzzy Link Quality Estimator (F-LQE) proposto por Baccour et al. [2010] é um LQE localizado no lado do receptor. F-LQE estima a qualidade do enlace por meio de quatro propriedades do enlace sendo elas: smoothed Packet Reception Ratio (SPRR), link Stability Factor (SF), link Asymmetry Level (ASL) e Average-Signal-to-Noise Ratio (ASNR). F-LQE apresenta monitoramento passivo e, ao contrário dos demais LQEs baseado em *score*, F-LQE tem suporte a enlaces assimétricos.

Avaliar o desempenho de LQEs é um problema desafiador. Uma das principais razões é fornecer quantificadores de precisão dos LQEs. Não é escopo deste trabalho efetuar tais análises, entretanto foram utilizados 3 diferentes estimadores, os quais estão em destaque na Tabela 2.2 sendo eles: ETX, ETT e *Four bit*. A escolha desses estimadores se deve, principalmente às suas características que se enquadram nos requisitos de eficiência, precisão, baixo custo de implementação, robustez [Fonseca et al., 2007; Couto et al., 2005; Draves et al., 2004; Javaid et al., 2009] requeridos pelas implementações dos algoritmos CRAL-LD, CRAL-FD e XCTP.

2.5 Sumário da Fundamentação Teórica

Esta seção apresenta, de forma resumida, o conteúdo deste capítulo e as principais inspirações adquiridas através da fundamentação teórica. Foram apresentados os conceitos e definições básicas das RSSF e do roteamento para estas redes, bem como os principais problemas no âmbito do desenvolvimento de protocolos em geral e especialmente dos protocolos de roteamento.

No que tange os requisitos básicos de roteamento em Redes de Sensores Sem Fio, isto é, confiabilidade, robustez, eficiência e flexibilidade serão utilizados como pontos de análises, de modo que através de métricas específicas (apresentadas nos Capítulos 4 e 5) para que seja possível quantificar cada um dos itens, o que viabiliza a comparação do comportamento entre os algoritmos aqui apresentados e os já conhecidos na literatura⁴.

No diz respeito aos paradigmas de roteamento este trabalho aborda em primeira instância construir rotas para coleta de dados (*Muitos-para-Um* vide Seção 2.2) e, em caso específico, viabilizar o roteamento ponto-a-ponto. As principais inspirações adquiridas dos conceitos e definições acerca dos paradigmas de roteamento e análise de dados são: i) favorecer a análise de dados durante o processo de roteamento, pois evita o uso desnecessário de banda e consumo de energia da rede; ii) utilizar uma abordagem centrada nos dados (DC) para construir as rotas e dar suporte ao item que será referenciado a seguir; iii) prover uma solução de roteamento que aproximada para o problema de Steiner (Seção 2.2.4.1) que é NP-Difícil. Deste modo, o roteamento apoia o uso de técnicas de fusão de dados para reduzir o número de transmissões e assim economizar energia.

Na Seção 2.4 foram discutidas as principais técnicas para medir a importância dos nós em uma rede e estimadores de qualidade que capturam as características dos

⁴Os principais algoritmos relacionados serão apresentados no Capítulo 3.

enlaces sem fio. Destas técnicas ressaltaram-se as centralidades *Betweenness* e *Sink Betweenness* e aos estimadores ETX, ETT e *Four bit*.

As soluções aqui apresentadas são baseadas nos mais modernos e consolidados modelos que parametrizam o ambiente das RSSF a respeito da conectividade sem fio, interferência, confiabilidade, transmissão entre outros que serão especificados em momento oportuno. Além disso, todas as soluções deste trabalho utilizam algum LQE para auxiliar o algoritmo de roteamento na escolha das rotas.

Capítulo 3

Trabalhos Relacionados

Nos capítulos anteriores foram apresentados os conceitos, definições e principais problemas das RSSF. Essa revisão da teoria provocou a proposição de dois protocolos de roteamento que são as principais contribuições deste trabalho, sendo eles: Centrality-based Routing Aware for L2Ns e eXtend Collection Tree Protocol.

Esta seção apresenta os mais relevantes trabalhos relacionados, os quais estão divididos em duas partes. Na primeira, serão apresentados algoritmos de roteamento clássicos do tipo *Address-Centric*, bem como abordagens mais sofisticadas do tipo *Data-Centric*. Estes protocolos são classificados e comparados de modo tabular e escrito com o algoritmo CRAL proposto. Na segunda parte desta seção, protocolos de roteamento já utilizados em produção final são discutidos, classificados e comparados com o protocolo XCTP, o qual é uma produção deste trabalho.

3.1 Trabalhos Relacionados – CRAL

Tabela 3.1: Comparação dos algoritmos de roteamento

	SPBC	TLC	FBC	RBC	CT	CRAL-LD	CRAL-FD	CTP	SPT	CNS	ImFRA	DAARP
Centralidade	BC	BC	BC	BC	SBC	SBC	SBC	-	-	-	-	-
Cod. distribuído	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓
LQE	-	-	-	-	-	✓	✓	✓	-	-	-	-

BC – Betweenness Centrality
SBC – Sink Betweenness Centrality

Na Tabela 3.1 são classificados os algoritmos de roteamento de acordo com as características a seguir: i) aplicação de centralidade na construção da árvore de roteamento; ii) uso de LQE para escolher enlaces de melhor qualidade. iii) implementação distribuída ou centralizada. Neste sentido, este trabalho apresenta as bases de como

construir os algoritmos *CRAL-Low Drop* e *Fast Delivery*, os quais são distribuídos e usam combinações de técnicas de centralidade e LQEs.

Shortest Path Betweenness-Centrality (SPBC) foi apresentado inicialmente no cenário das redes sociais para medir o potencial de influência de nós possuem individualmente sobre o fluxo de informações na rede Freeman [1977]. SPBC assume que somente os menores caminhos são usados para transportar os dados e que cada nó envia um pacote para cada outro nó. Sendo assim, o SPBC de um nó n é o número esperado de pacotes que passam por n .

Traffic Load Centrality (TLC) Brandes [2008] juntamente com SPBC fazem parte do grupo de técnicas tradicionais no uso da centralidade *Betweenness* para roteamento. TLC assume que o fluxo de tráfego segue as rotas de menor comprimento, porém utiliza um mecanismo de roteamento diferente do SPBC. A estratégia de roteamento do TLC é mais realística, onde cada nó escolhe de modo aleatório um vizinho, o qual está mais próximo do destino para encaminhar os pacotes.

A principal limitação das estratégias de roteamento SPBC e TLC é que ambas utilizam os menores caminhos para definir suas rotas, enquanto na prática o tráfego da rede pode ser desviado dos menores caminhos para aumentar o desempenho da rede Couto et al. [2005]. Além disso, SPBC e TLC são inflexíveis no que tange o balanceamento de cargas, tolerância a falhas e acordo de níveis de serviço Dolev et al. [2010].

Flow Betweenness-Centrality (FBC) Freeman et al. [1991], ao contrário de SPBC e TLC, assume que as rotas são simples (não possuem ciclos) e considera igualmente caminhos de todos os comprimentos. De modo preciso, FBC de um nó é a soma das frações do fluxo máximo entre cada par de nós. FBC é razoável ao assumir rotas simples, porém FBC falha ao considerar todas as rotas igualmente, pois em redes são preferíveis os caminhos mais curtos em detrimento dos mais longos.

Routing Betweenness Centrality (RBC) foi apresentado em Dolev et al. [2010] como uma generalização das estratégias SPBC, TLC, e FBC. Contudo, estes algoritmos não foram projetados de modo distribuído como CRAL-LD e CRAL-FD. Além disso e não consideram as restrições impostas pelas RSSF como, por exemplo, taxa de perda nos enlaces da rede ou energia residual dos nós.

Dos algoritmos não baseados em centralidade e que não usam LQE apresentados na Tabela 3.1, tem-se *SPT* e *Center at Nearest Source (CNS)* como representantes tradicionais. Para SPT, os menores caminhos são utilizados para encaminhar todo o fluxo da rede. Já para CNS, um nó mais próximo do sorvedouro recebe mensagens dos descendentes e encaminha os dados para o sorvedouro.

Information Fusion-based Role Assignment (InFRA) e Data-Aggregation Aware

Routing Protocol (DAARP) são discutidos em Villas et al. [2010], estas abordagens são aproximações da árvore de Steiner Fasolo et al. [2007]. InFRA e DAARP são otimizados para eventos que ocorrem de forma aglomerada (*clustering*) na área monitorada, isto é, os eventos ocorrem em poucos pontos da região monitorada e somente os nós a uma distância S do local do evento reportam informações. Ao contrário dessa estratégia que restringe as aplicações das RSSF, os algoritmos CRALs são otimizados para a estratégia que generaliza a anterior. Desta forma, permite-se que os eventos ocorram em locais aleatórios da região monitorada. Além disso, CRAL utiliza centralidade para escolher os nós intermediários da árvore de roteamento, enquanto InFRA e DAARP fazem o roteamento para cada *cluster* de evento detectado.

Em ambientes sem fio, a qualidade do enlace é um quesito que gera grande impacto no desempenho da RSSF Baccour et al. [2009]. O consumo energético, tempo de entrega dos pacotes e efeito *overhearing* sofrem quando a qualidade dos enlaces é baixa. CTP usa o estimador de qualidade de enlace denominado ETX, específico para redes sem fio. Algoritmos como RBC e CT Oliveira et al. [2010] usam centralidade para construir suas rotas, porém não consideram a qualidade dos enlaces. CRAL-LD, ao contrário dos anteriores, além de utilizar centralidade *Betweenness* para escolher os melhores nós repetidores no meio da rede, emprega ETX para encontrar caminhos de qualidade, minimizar o consumo de energia e efeito *overhearing*. CRAL-FD utiliza ETT como estimador de enlace. ETT é uma variação do ETX que ajusta a taxa de transmissão, deste modo a configuração das rotas escolhidas conduzem a alta vazão e baixa perda de pacotes na rede.

3.2 Trabalhos Relacionados – XCTP

Tabela 3.2: Comparação dos paradigmas de comunicação sobre os protocolos relacionados ao XCTP.

Um-para-Muitos (Disseminação de dados)	Muitos-para-Um (Coleta de dados)	Qualquer-para-Qualquer (Ponto a Ponto)
<i>Não confiável</i>	<i>Não confiável</i> <i>Confiável</i>	<i>Confiável</i>
Directed Diffusion DIP, DRIP e DHV Deluge	CTP MintRoute XCTP MultiHopLQI	AODV, DYMO e DSR Hydro RPL

Os principais protocolos relacionados ao XCTP são apresentados na Tabela 3.2. Estes protocolos estão classificados de acordo com o paradigma de comunicação (*Muitos-para-Um*, *Um-para-Muitos* e *Ponto-a-Ponto*) e se permite comunicação confiável. Neste trabalho, comunicação confiável diz respeito à possibilidade de troca

bidirecional de informações entre os participantes da comunicação. Na Tabela 3.2 é destacado o protocolo XCTP que é, no melhor dos nossos conhecimentos, o único que permite para coleta de dados que é confiável. Em outras palavras, XCTP é um protocolo de coleta de dados que também permite rotas *unicast* no sentido EB-para-nó sensor. Além disso, XCTP oferece uma interface que facilita o desenvolvimento de protocolos de transporte fim-a-fim.

Directed Diffusion Protocol Intanagonwiwat et al. [2003] é um dos pioneiros protocolos de disseminação de dados na rede. Este protocolo usa sucessivas inundações para disseminar pequenas quantidades de informação pela rede. Em geral, as informações disseminadas são para configuração de algum parâmetro dos nós sensores podendo ser desde variáveis a serem monitoradas ou até frequência de amostragem.

Os protocolos *DIP*, *DRIP*, *DHV* Levis et al. [2005] do mesmo modo que *Directed Diffusion* também são usados para disseminação de pequenas quantidades de informação pela rede. *DIP*, *DRIP*, *DHV* oferecem eventual consistência de rotas e usam temporizadores baseados no algoritmo *Trickle* apresentado em Levis et al. [2003b] como mecanismos auxiliares a disseminação. *DRIP* trata cada informação como uma entidade separada, isto permite um maior gerenciamento e controle de quando e quão rápido os dados serão disseminados. Já os protocolos *DIP* e *DHV* tratam os dados como um grupo sem distinção entre eles, isto implica que o controle e os parâmetros de disseminação são aplicados de modo igual para todos os dados.

Deluge é um confiável mecanismo de disseminação de dados para propagação de grandes quantidades de dados¹ de um ou mais nós fonte para todos os demais nós da rede, *Deluge* faz isso através da comunicação com múltiplos saltos das RSSF. Para ser robusto em face das perdas de comunicação e falhas dos nós, *Deluge* adota uma abordagem epidêmica, em que os dados são de tamanho fixo (páginas) e permitem ser transmitidas de modo incremental. Os autores mostram que *Deluge* é confiável na distribuição de dados ao passo que mantém constante a quantidade de estados locais armazenados. Além disso, é demonstrado que *Deluge* é econômico no consumo de energia.

MultiHopLQI é um protocolo para coleta de dados em RSSF com múltiplos saltos Philip Buonadonna [2007]. Baseando-se no indicador fornecido pelo LQI o protocolo *MultiHopLQI* escolhe as rotas para encaminhar os pacotes. O ponto fraco deste protocolo é que somente o estimador baseado em hardware é utilizado, isto implica que a qualificação de enlaces intermediários é prejudicada acarretando em super ou sub valorização dos enlaces.

¹Mais do que se cabe na memória RAM dos notes.

MintRoute foi proposto por Woo et al. [2003a] para estabelecer rotas entre os nós sensores e uma EB. O protocolo tenta escolher vizinhos que possuem melhor qualidade de enlace. *MintRoute* exige que uma estimativa da taxa de tráfego de dados seja codificada junto com o protocolo e ambos estimadores de enlace e controle de tráfego dependem deste parâmetro. Pelo fato da maior integração entre a camada de roteamento e as mais baixas (enlace e física) *MultiHopLQI* leva vantagem, pois faz uso de estimadores fornecidos pelo rádio utilizado.

Collection Tree Protocol introduzido em Gnawali et al. [2009] é um dos protocolos de roteamento para coleta de dados mais atuais e renomados da literatura. Os autores do CTP mostraram a eficiência do protocolo, além de deixar o código aberto para a comunidade. CTP identifica e age rapidamente às mudanças na dinâmica da topologia da rede ao passo que mantém baixo o consumo de energia. Os autores argumentam que CTP alcança os requisitos de agilidade e eficiência necessários para RSSF, enquanto oferecem alta confiabilidade na entrega dos pacotes de dados, além de ser consciente ao reduzir o envio de mensagens de sonda para economizar energia. Os experimentos práticos com CTP comprovaram a entrega de dados foi superior a 90%. CTP quando comparado com *MultiHopLQI* apresenta 73% a menos de mensagens de sonda do que e reduz a latência em até 99.8%.

CTP e *Deluge* quando comparados apresentam finalidades opostas. CTP é até o momento o estado da arte em coleta de dados, ao passo que *Deluge* também o é para disseminação de grandes quantidades de dados em RSSF. *XCTP* por sua vez apresenta características intermediárias entre esses dois extremos, pois *XCTP* cria a árvore de roteamento para coleta de dados tal como CTP, indo além e construindo também rotas *unicast* entre a estação base e os nós sensores permitindo, por exemplo, a reprogramação de nós sensores funcionalidade semelhante à do protocolo *Deluge*, porém com o adicional de permitir que mensagens de confirmações sejam trocadas de modo fim-a-fim.

Hydro é um protocolo sobre o paradigma ponto-a-ponto apresentado por Dawson-Haggerty et al. [2010]. Este protocolo utiliza um sofisticado mecanismo baseado em tabelas para construir rotas que possibilitam a comunicação entre quaisquer pares de nós. Estas rotas forma um *Directed Acyclic Graph* (DAG) que possuem, como raiz, um ou mais nós especiais que mantém a visão da topologia global da rede. *Hydro* difere de *XCTP* em dois aspectos. Primeiro eles são construídos sobre diferentes paradigmas, isto é, *hydro* cria rotas ponto-a-ponto, enquanto *XCTP* constrói rotas bidirecionais entre as EBs e os nós da rede. Em segundo lugar, *XCTP* não armazena rotas ao não ser aquelas entre a estação base e os nós sensores e vice-versa, ou seja, *Hydro* consome mais memória pois necessita armazenar rotas para fluxos entre quaisquer nós da rede.

Memória e requisitos de comunicação são os principais fatores que impactam nas abordagens dos protocolos XCTP e Hydro. Em caso de comunicação no sentido de coleta de dados e reconfiguração de nós sensores, o protocolo XCTP pode ser uma alternativa razoável, visto que geralmente RSSF apresentam limitações de quantidade de memória. Já se a aplicação necessita de comunicação fim-a-fim entre os nós sensores, o protocolo Hydro pode representar uma alternativa melhor. Vale ressaltar que o protocolo XCTP também pode ser utilizado para comunicação ponto-a-ponto (detalhes no Capítulo 5 Seção 5.2), porém esta não é a principal finalidade do protocolo.

Routing Protocol for low-power and lossy networks ou simplesmente RPL foi apresentado por Winter [2012]. O protocolo RPL constrói DAGs e Destination Oriented DAGs (DODAGs) como estrutura de roteamento. Cada nó envia mensagens de controle do tipo DAG Information Object (DIO) que contém informações que permite o nó descobrir uma instância RPL, aprender a configuração e os parâmetros e selecionar os DODAG corretamente. Além disso existem as mensagens DAG Information Solicitation (DIS) e Destination Advertisement Object (DAO). DIS solicita informações sobre DODAG a um nó já pertencente a estrutura de roteamento. Já as mensagens do tipo DAO são usadas para anunciar rotas com destino a cada nó existente na rede.

A principal diferença entre RPL e XCTP é que o primeiro utiliza mensagens de controle para criar rotas reversas entre as raízes dos DAGs e os demais nós, enquanto XCTP não necessita deste custo adicional, pois XCTP usa os pacotes de dados para construir tais rotas. XCTP também permite que rotas não utilizadas sejam removidas através de políticas baseadas em Time To Live (TTL), deste modo evita-se consumo adicional de memória para armazenar rotas pouco utilizadas.

Ad hoc On Demand Distance Vector (AODV) é um protocolo sob demanda² implementado para comunicação Ponto-a-Ponto por Perkins & Royer [1999]. AODV fornece *unicast* e *multicast* comunicação e aplica mecanismos de TTL para remoção de rotas não utilizadas. Quando um nó, operando com AODV, necessita enviar uma mensagem para um outro nó qualquer, inicia-se um processo de descoberta de rotas que efetua uma inundação na rede com pacotes Route Request (RREQ). Os nós que recebem o pacote RREQ armazenam internamente o caminho reverso e quando o destinatário recebe o RREQ, o processo de resposta se inicia. No processo de resposta, o destinatário sabe que existe um caminho reverso até o emissor, assim uma mensagem Route Reply packet (RREP) é encaminhada até o emissor do RREQ. Deste modo, AODV consegue construir rotas ponto-a-ponto. *Dynamic MANET On-demand Routing Protocol (DYMO)* é o sucessor do AODV e compartilha muitas de suas características,

²Cria rotas somente quando as entidades necessitam estabelecer comunicação. Protocolos sob demanda também são chamados de protocolos reativos.

contudo é mais fácil de implementar e foi projetado para permitir melhorias futuras.

Dynamic Source Routing (DSR) apresentado em Johnson et al. [2007] tal como AODV também é um protocolo sob demanda e ponto-a-ponto. DSR também utiliza um mecanismo de descoberta de rotas similar ao utilizado pelo AODV, porém as rotas são armazenadas no cabeçalho dos pacotes RREQ. Para responder uma solicitação de estabelecimento de rotas, o destinatário da envia uma mensagem RREP que também contém toda a rota no cabeçalho. Para a manutenção das rotas cada nó intermediário que intercepta um pacote de RREQ ou RREP mantém em cache rotas parciais e com elas oportunisticamente aprende novas rotas.

AODV e DSR apresentam características bastante similares, apresentando diferenças no estilo de armazenamento das rotas. Já XCTP quando comparado com AODV e DSR revelam-se maiores contrastes. Por exemplo, ao contrário de DSR, o protocolo XCTP não armazena quaisquer informações de rotas no cabeçalho dos pacotes. Já AODV apresenta similaridades com XCTP no estilo de armazenamento de rotas. No entanto, XCTP ao contrário de AODV, não salva rotas que não são as reversas entre os nós sensores e as EBs, além disso AODV é um protocolo reativo ao passo que XCTP é proativo, isto é, XCTP constrói e mantém as rotas³ antes que sejam demandadas sendo, portanto, mais ágil quando as entidades necessitam trocar dados.

Nenhum dos protocolos aqui relacionados permitem enviar mensagens *unicast* em ambos os sentidos EB-para-nós e vice-versa como XCTP, exceto pelos protocolos que permitem comunicação ponto-a-ponto, os quais geralmente ou requerem grande quantidade de memória para armazenar estados ou embutem custo adicional aos pacotes armazenando rotas em seus cabeçalhos ou ainda realizam inundações para cada entidade que solicite comunicação. XCTP não compartilha dessas características colocando-o como uma alternativa para o roteamento em coleta de dados para RSSF.

3.3 Sumário dos Trabalhos Relacionados

Este capítulo apresentou os principais trabalhos relacionados aos temas aqui propostos, com foco aprimorado e comparativo aos protocolos de roteamento.

Os protocolos relacionados ao CRAL foram classificados e confrontados de modo descritivo ao longo da Seção 3.1. Destes protocolos destacam-se os que possuem abordagem centrada nos dados tais como: CRAL-LD, CRAL-FD, CT, InFRA e DAARP. Para efeito comparativo utilizou-se as características inerentes de cada abordagem como versão distribuída do algoritmo, uso de técnicas de centralidade, uso de estimadores,

³XCTP cria as rotas para coleta de dados proativamente, as rotas reversas são construídas de modo oportunista quando mensagens de dados são enviadas ou quando solicitada pela EB.

características de operação e tipo de construção de rotas se AC ou DC. Com base nesses comparativos, buscou-se mostra os pontos fortes e fracos de cada algoritmo e inserir o protocolo CRAL como uma alternativa de roteamento para coleta de dados.

Na sequência do capítulo, foram apresentados os protocolos relacionados ao XCTP, classificando-os de acordo com seu paradigma de roteamento e construção de rotas que permitam comunicação bidirecional. Com base nessas características XCTP é o único protocolo considerado de Coleta de Dados Confiável. Como representantes significativos de cada uma das classes estabelecidas na Tabela 3.2 têm-se: Deluge, CTP, XCTP, Hydro e RPL. Deluge e CTP são extremos opostos quando diz respeito aos paradigmas utilizados, XCTP atenua estes extremos por deter características tanto de coleta quanto de disseminação. Hydro e RPL são protocolos mais complexos que permitem comunicação ponto-a-ponto, por isso utilizam mais recursos, ao passo que XCTP mitiga o uso de recursos, principalmente memória que é uma das maiores limitações das RSSF.

Em decorrência das declarações realizadas neste capítulo é possível estabelecer um posicionamento dos objetos de estudo aqui propostos em relação aos da literatura sendo eles: i) identificação da classificação do protocolo CRAL, segundo os critérios de classificação centralidade, algoritmo distribuído, uso de LQE, tipo de abordagem AC/DC. ii) identificação da classificação do protocolo XCTP. Baseando-se nos critérios de paradigmas de roteamento e possibilidade de comunicação bidirecional.

Capítulo 4

Protocolo de Roteamento CRAL (*Centrality-based Routing Aware for L2Ns*)

Neste capítulo será apresentado o algoritmo CRAL para mitigar o problema de roteamento otimizado para coleta de dados em RSSF. CRAL faz isso selecionando nós intermediários da rede usando o critério de centralidade Sink Betweenness Centrality (SBC) e considerando que os enlaces entre os nós sensores possuem taxa intermediária de perda, além de possuírem diferentes taxas de transmissão. Serão apresentadas duas versões do algoritmo CRAL: 1) CRAL-Low Drop utiliza como LQE a métrica ETX que reduz o número de perdas de mensagens ao escolher enlaces que necessitam de menor quantidade de transmissões para entregar os dados; 2) CRAL-Fast Delivery emprega ETT como métrica para estimar os enlaces, desta forma enlaces com menor tempo de entrega são escolhidos.

Este capítulo evidencia as bases da arquitetura do protocolo CRAL que é utilizada nas versões CRAL-LD e CRAL-FD. A arquitetura do protocolo é dividida em duas partes. O plano de controle caracteriza a primeira parte da arquitetura, este plano tem por objetivo construir e modificar a tabela de roteamento baseada em centralidade. Já o plano de dados caracteriza a segunda parte da arquitetura, neste caso o plano de dados tem por objetivo consultar a tabela de rotas para o correto encaminhamento dos pacotes. Além da arquitetura, este capítulo mostra a metodologia quantitativa realizada sobre o protocolo proposto e os principais algoritmos relacionados da literatura, de modo que foi realizado um levantamento sobre os principais modelos e métricas que possam quantificar os experimentos realizados em laboratório.

As análises sobre os experimentos mostraram que as versões CRAL-LD e

CRAL-FD, em comparação com os concorrentes da literatura, são mais inteligentes na escolha das rotas ao somente optar por rotas de alta qualidade e que representam uma melhor aproximação para a árvore de Steiner. CRAL-LD e CRAL-FD apresentam alta taxa de entrega de dados alcançando 99.6% e 99.1% respectivamente, enquanto os protocolos comparados entregam cerca de 60% nos cenários avaliados. Além disso, CRAL-LD reduz drasticamente o número de transmissões por dados reportados ao passo que CRAL-FD reduz consideravelmente o tempo de entrega de cada pacote de dados. O protocolo CRAL em conjunto com o algoritmo para balanceamento do consumo de energia mostrou-se satisfatório ao reduzir o consumo de energia dos nós mais centrais e próximos à EB ao passo que mantém todas as características do protocolo original. No que diz respeito aos protocolos operando junto com as funções de fusão de dados, CRAL entrega dados atualizados em todos os cenários quando comparado com CT. CRAL-LD e CRAL-FD são alternativas para balancear o compromisso entre número de transmissões (consumo de energia) e a rápida entrega dos dados ao utilizar a função de fusão de dados *Periodic per hop adjusted (PerHopAd)*. Também é mostrado um algoritmo para balancear o consumo de energia dos nós centrais e dos nós próximos à EB, bem como os compromissos existentes ao usar essa abordagem.

Este capítulo está organizado da seguinte maneira: inicialmente na Seção 4.1 será delimitado o escopo do problema, em seguida na Seção 4.2 será apresentado o protocolo como uma solução para o referido problema, descrevendo sua arquitetura e algoritmo em detalhes, ainda na seção é apresentado o algoritmo *Policy Aware* utilizado para reduzir o consumo de energia dos nós mais centrais da rede. Na sequência, a perda de pacotes em roteamento com fusão de dados é discutida na Seção 4.3, bem como as soluções existentes da literatura para atenuar o problema. Na Seção 4.4 é apresentada uma análise de complexidade assintótica do algoritmo proposto. Posteriormente, na Seção 4.5 é descrita a metodologia empregada nos experimentos, bem como os resultados de simulação obtidos. Finalmente, na Seção 4.6 são apresentadas as considerações finais do capítulo.

4.1 Escopo do Problema

4.1.1 Centralidade

A árvore de Steiner é conhecida em redes de computadores por suas diferentes aplicações Winter [1987]. Para RSSF, esta árvore apresenta pontos ótimos para a fusão de dados, o que pode minimizar o consumo de energia da rede como um todo Fasolo et al. [2007]. Entretanto, o problema de encontrar a árvore de Steiner é NP-Difícil

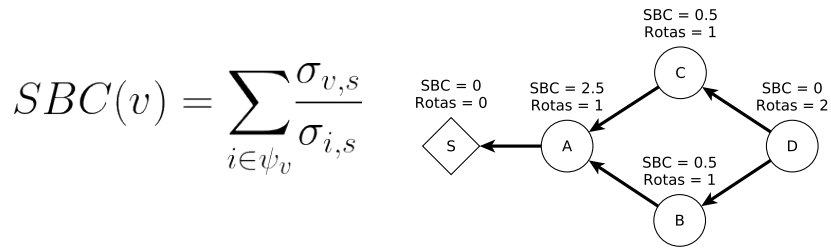


Figura 4.1: *Sink Betweenness Centrality* para cada vértice do grafo.

(vide Seção 2.2.4.1), assim soluções aproximadas devem ser propostas.

O protocolo CRAL é uma solução aproximada para a árvore de Steiner. CRAL realiza essa aproximação por estabelecimento de um critério de importância aplicado sobre os nós da rede. Este critério é determinado pela capacidade de monitoramento e controle sobre os dados que fluem na rede. Um fator decisivo para caracterização deste critério é a localização topológica do nó a qual deve ser identificada e qualificada.

CRAL faz uso da centralidade *Sink Betweenness* para identificar e qualificar os nós que atendem o critério de importância. Neste sentido, a centralidade classifica os nós que são mais influentes na rede, em outras palavras, é realizada a identificação dos nós intermediários que mais participam nos caminhos dos demais nós. A qualificação é realizada de acordo com o nível de centralidade, isto é, o nó é escolhido como nó roteador quanto maior for seu valor de centralidade. Estes nós sensores possuem a responsabilidade de encaminhar as mensagens dos nós em níveis mais baixos da árvore de roteamento. Usando os nós mais importantes, segundo a sua centralidade, a quantidade de nós intermediários é reduzida e pontos de fusão de dados são favorecidos. Isto coloca esta estratégia como uma aproximação do problema da árvore de Steiner.

Tabela 4.1: Parâmetros para cálculo da centralidade *Sink Betweenness*

Nó	Número de caminhos	Número de caminhos (Descendentes)	$\psi_{\text{nó}}$
S	0	$[\emptyset]$	$[\emptyset]$
A	1	$[1,1,2]$	$[B, C, D]$
B	1	$[2]$	$[D]$
C	1	$[2]$	$[D]$
D	2	$[\emptyset]$	$[\emptyset]$

Sink Betweenness Centrality é uma métrica de centralidade apropriada para coleta de dados em RSSF (vide Seção 2.4.1). O cálculo do *SBC* de um nó t depende do número de caminhos possíveis partindo de t até o sorvedouro e do número de suas participações, como nó intermediário, nos menores caminhos dos nós descendentes até o sorvedouro. A Equação 2.8 do Capítulo 3 define matematicamente o *SBC*, esta equação é novamente

exibida na Figura 4.1, além disso também é exibida nesta figura um exemplo de grafo com todos os valores de SBC já calculados.

Por exemplo, para calcular o SBC do nó A da Figura 4.1 é necessário saber que a quantidade de rotas partindo de A até S é 1 (numerador da equação) e o número de participações de A nas menores rotas dos vértices B, C, D que são respectivamente 1, 1, 2 (denominadores da equação). Portanto, $SBC_A = \frac{1}{1} + \frac{1}{1} + \frac{1}{2}$. O raciocínio é o mesmo para encontrar o SBC dos demais vértices. Note que S possui 0 rotas e D não faz parte do menor caminho de nenhum outro vértice, portanto o $SBC_{S,D} = 0$. A Tabela 4.1 mostra, para cada nó da Figura 4.1, o número de caminhos, o conjunto ψ e informações sobre os descendentes de cada nó. Estas informações estão arranjadas na forma de pares ordenados (número de caminhos, quantidade de descendentes).

4.1.2 Estimador de Enlace

O estudo de estimadores de qualidade de enlace está sendo objeto de estudo dos últimos anos [Baccour et al., 2009; Javaid et al., 2009; Baccour et al., 2012]. Todo este esforço é explicado pelo fato de que os estimadores podem ter grande impacto no desempenho da rede, além de afetar o projeto dos protocolos das camadas superiores.

A principal diferença entre as duas versões apresentados do protocolo CRAL encontra-se no LQE utilizado. CRAL-LD utiliza ETX, esta métrica é uma aproximação mais apropriada para estimar a qualidade dos enlaces sem fio da rede. ETX tem como principal característica minimizar o número esperado de transmissões e retransmissões requeridas para entregar os pacotes, estas características dão nome a versão do protocolo *Low Drop*. ETT é o LQE utilizado pela versão CRAL-FD, esta métrica mantém todas as características do ETX e vai além possibilitando o ajuste da taxa de transmissão e do tamanho das mensagens. ETT considera que os enlaces possuem tempos de transmissão diferentes, com isso o algoritmo CRAL-FD escolhe enlaces mais rápidos, esta escolha acelera a entrega de pacotes individuais, por isso o nome *Fast Delivery* foi escolhido para a versão do CRAL baseado no *Expected Transmission Time*. É notório que para o uso eficiente da métrica ETT se faz necessário que o rádio forneça diferentes taxas de transmissões ou que seja aplicado dois ou mais rádios operando em canais diferentes e com taxas de transmissão diferentes Draves et al. [2004]. Na Seção 4.5.1 será apresentada uma modelagem para rádios operando sobre o padrão IEEE 802.11b utilizado neste trabalho.

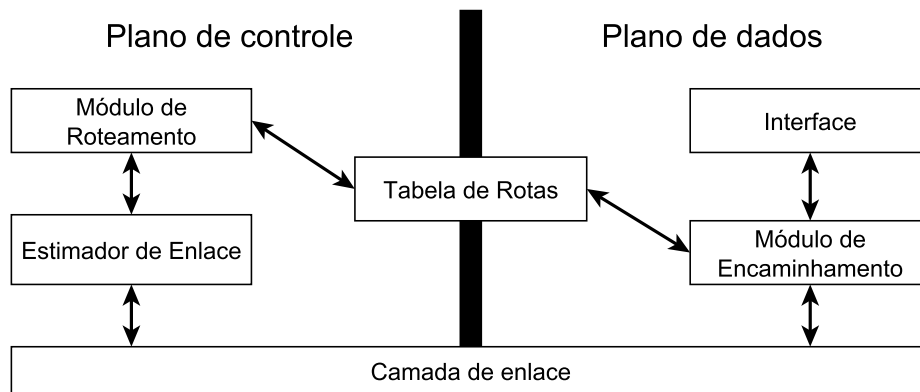


Figura 4.2: Arquitetura do protocolo CRAL.

4.2 O Protocolo *Centrality-based Routing Aware for L2Ns*

4.2.1 Arquitetura

Na Figura 4.2 é mostrada a arquitetura do algoritmo CRAL, bem como os principais módulos e seus relacionamentos. A tabela de rotas é o principal objeto a ser manipulado pelos demais módulos da arquitetura, por esse motivo esta tabela é o elemento central mostrado na Figura 4.2. Os demais módulos estão divididos em duas partes: o Plano de dados e o Plano de Controle. Estes módulos serão discutidos em detalhes a seguir:

1. *Plano de Dados*: apresenta como principal elemento o Módulo de Encaminhamento. Este módulo a parte da arquitetura responsável pela decisão sobre o que será feito com um pacote que acaba de ser interceptado pela interface de rede. A Figura 4.2.1 apresenta um módulo de encaminhamento que tem como tarefa efetuar consultas na tabela de rotas. O módulo de encaminhamento recupera informações sobre o destinatário no pacote de dados e busca, na tabela de rotas, o próximo salto para o destinatário do pacote interceptado (caso o próprio nó não o seja);

Na Figura 4.2, na parte do Plano de Dados também existe um módulo de Interface. Este módulo é responsável por prover serviços para a camada superior na pilha de protocolos. Esta interface oferece mecanismos para utilização das rotas de coleta de dados;

2. *Plano de Controle*: no roteamento, o plano de controle é a parte da arquitetura do CRAL que visa efetuar o mapeamento da rede. Mais precisamente, o módulo de roteamento gerencia (insere, atualiza e remove) entradas da Tabela de Rotas,

isto possibilita que o plano de dados tome decisões sobre os pacotes de dados de modo independente. O objetivo deste módulo é povoar a Tabela de Rotas de modo que permitam comunicação unicast sob o paradigma da coleta de dados. A Tabela de Rotas pode conter uma lista de endereços de próximo salto para rotas em direção a uma EB. O Módulo de Roteamento também pode conter regras para definir ações e prioridades sobre certos tipos de pacotes, por exemplo, *os pacotes de dados devem ser tratados antes de pacotes de sonda*.

As seleções das rotas baseiam-se nas regras para calcular *Sink Betweenness Centrality* presentes no Módulo de Roteamento, bem como nas informações providas pelo módulo Estimador de Enlace. Este módulo possui acesso a informações da camada de enlace sobre o estado de cada enlace para os nós vizinhos. Deste modo, o Plano de Controle efetua decisões otimizadas sobre as melhores rotas para a coleta de dados.

Abaixo na Figura 4.2 é exibida a camada de enlace que prover serviços tais como: acesso a informações dos enlaces, filas de envio e recepção de pacotes, alocação para uso do rádio de comunicação.

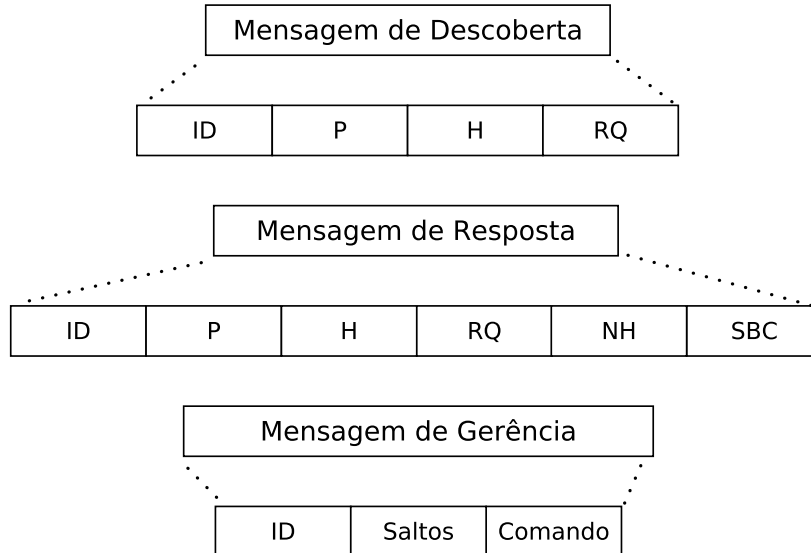


Figura 4.3: Mensagens de controle do protocolo CRAL.

O protocolo CRAL faz uso de três tipos distintos de mensagem de controle para construir a estrutura de roteamento. A Figura 4.3 mostra em detalhes cada uma das mensagens. 1) Mensagem de Descoberta (MD), o qual que contém os campos *ID* do nó, *P* número de caminhos, *H* número de saltos, e *RQ* qualidade da rota; 2) Mensagem de Resposta (MR) apresenta os mesmos campos da MD, porém adiciona os campos

NH e SBC para representarem respectivamente o próximo salto e a centralidade Sink Betweenness; 3) Mensagem de Gerência (MG) necessitam somente de um campo para ID do nó que enviou a mensagem, um campo para informar o número de saltos e um para informar o comando a ser executado (comandos: entrada, saída, alerta). A seguir será detalhada a finalidade de cada um dos tipos de pacote, no que tange o mecanismo de roteamento do algoritmo CRAL.

4.2.2 Algoritmo

Algoritmo 1: Fase de Inicialização

```

[1] Procedimento Inicializa_Nó()
[2]    $EB \leftarrow 1$  // Estação base
[3]    $Caminhos_t \leftarrow 1$ 
[4]    $SBC_t \leftarrow 0$  // Centralidade Sink Betweenness
[5]    $Saltos_t \leftarrow \infty$ 
[6]    $RQ_t \leftarrow \infty$  // Qualidade da Rota
[7]    $PróximoSalto_t \leftarrow \emptyset$ 
[8]    $MAX_{SBC} \leftarrow \emptyset$ 
[9]   se  $Nó_{id} = EB$  então
[10]    // Inicializa Pacote MD
[11]     $MD_H \leftarrow Saltos$ 
[12]     $MD_P \leftarrow Caminhos$ 
[13]    Broadcast MD
[14]   fim
[15]   retorna
[16] Fim

```

Similar à maioria dos algoritmos de coleta de dados, CRAL começa pela requisição de coleta efetuada por uma EB. O pseudo Algoritmo 1 mostra a inicialização das variáveis e o agendamento da requisição para o estabelecimento da árvore de coleta dados, a estrutura é montada usando os estimadores de enlace ETX e ETT. A Figura 4.4a ilustra a situação da rede no momento em que a EB efetua a requisição, a qual que desencadeia a inundação inicial na rede.

O restante do algoritmo CRAL pode ser dividido em duas fases principais: Fases de Descoberta e de Resposta. Abaixo será detalhada cada uma dessas fases:

1. *Fase de Descoberta*: esta fase tem início no momento em que um pacote MD é interceptado. Ao receber um pacote MD, o Módulo de Roteamento é sinalizado para aplicar as regras exibidas no pseudo Algoritmo 2. A linha 2 verifica se uma nova rota com melhor qualidade foi encontrada, o termo $RQ_{t \rightarrow MD_{ID}}$ representa

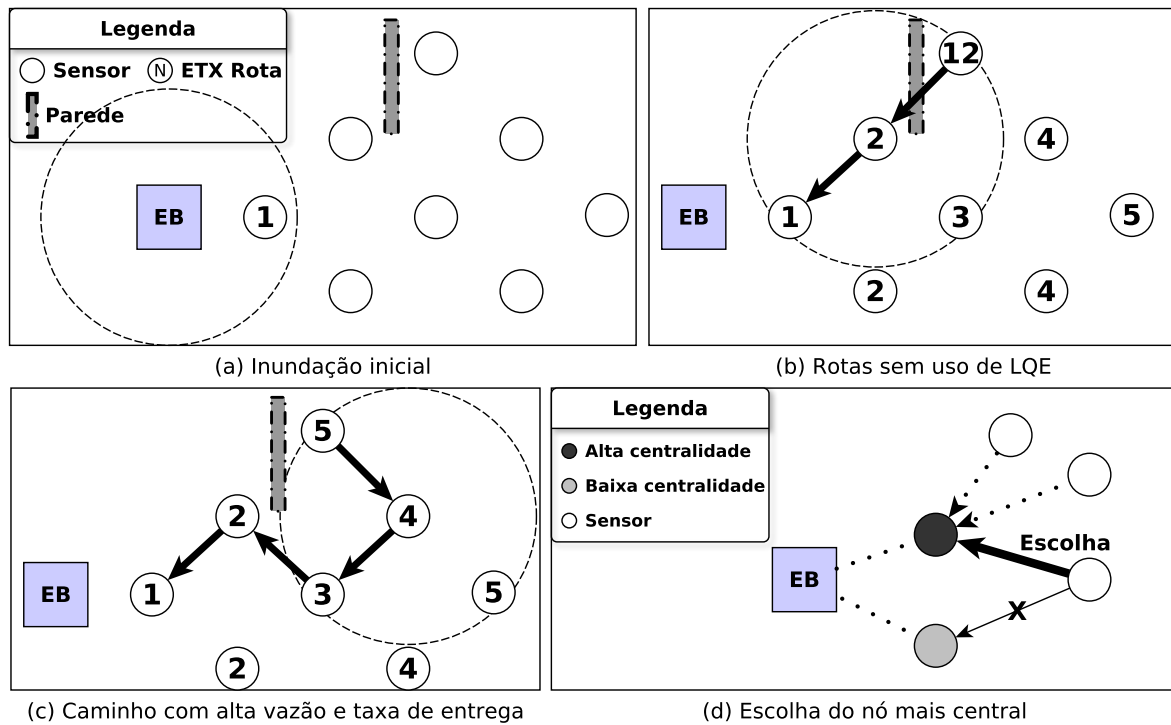


Figura 4.4: Operação dos algoritmos CRALs.

a qualidade do enlace entre o nó t e o nó emissor do pacote MD_{ID} . As linhas 3–9 fazem o preenchimento adequado das variáveis e atualização do pacote. Na existência de duas ou mais rotas entre o nó e a EB, a condição da linha 10 será satisfeita e o contador de caminhos será atualizado. Ao receber a requisição os nós enviam uma resposta de volta na árvore de rotas já construída. A resposta e o encaminhamento da mensagem de descoberta são engatilhados pelos comandos das linhas 16–17, onde os pacotes MD e MR são enfileirados para envio. Deste modo, cada nó pode deduzir quantos filhos possui e seus caminhos, o que permite o cálculo da centralidade *Sink Betweenness*.

A Figura 4.4b e 4.4c exibem rotas sem e com o uso de LQEs respectivamente. Na Figura 4.4b é exibida uma rota de baixa qualidade, em que o nó atrás da parede apresenta alto valor de ETX implicando em sucessivas falhas de envio, estas rotas geralmente são escolhidas quando não são utilizados LQEs. Por outro lado, na Figura 4.4c representa rotas escolhidas pelo protocolo CRAL, neste caso o enlace que atravessa a parede não é escolhido. Desta forma, somente rotas com menor taxa de perda e alta vazão são escolhidas e conseqüentemente aumenta-se a vazão e reduz o consumo de energia da rede;

2. *Fase de Resposta*: esta fase é caracterizada pelo cálculo da centralidade *Sink*

Algoritmo 2: Fase de Descoberta

```

[1] Procedimento Descoberta(Pacote MD)
[2]   se  $MD_{RQ} + RQ_{t \rightarrow MD_{id}} < RQ_t$  então
[3]     // Nova rota foi encontrada
[4]      $Saltos_t \leftarrow MD_H + 1$ 
[5]      $Caminhos_t \leftarrow MD_P$ 
[6]      $ProximoSalto \leftarrow MD_{ID}$ 
[7]      $MD_{RQ} \leftarrow MD_{RQ} + RQ_t$ 
[8]      $MD_H \leftarrow Saltos_t$ 
[9]      $MD_P \leftarrow Caminhos_t$ 
[10]  senão se  $MD_{RQ} + RQ_{t \rightarrow MD_{id}} = RQ_t$  então
[11]    // Outro caminho de mesma qualidade foi encontrado
[12]     $Caminhos_t \leftarrow MD_P$ 
[13]     $MD_P \leftarrow Caminhos_t$ 
[14]  fim
[15]  // preenche o pacote com os dados do nó
[16]  Broadcast MD
[17]  Broadcast MR
[18]  retorna
[19] Fim

```

Betweenness e seleção dos nós mais centrais como nós intermediários. Cada nó preenche uma mensagem MR com suas informações de número de caminhos, centralidade, saltos e qualidade de rotas e envia através dos caminhos encontrados na Fase de Descoberta (veja linha 17 do Algoritmo 2).

Ao receber uma mensagem MR os nós executam as regras mostradas no pseudo Algoritmo 3. Na linha 2, cada nó verifica se recebeu um pacote MR de um nó descendente¹, caso esta condição seja verificada também é testado se MR_{ID} pertence ao conjunto ψ_t para então calcular a centralidade SBC (linha 5). Cada nó t conhece seu conjunto ψ_t , pois seus descendentes mandam mensagens de resposta através das rotas *unicast* criadas na fase *Descoberta*, estas mensagens trafegam através de todos os nós ψ_t até o próprio t . Após o cálculo do SBC, o nó atualiza as informações do pacote MR e efetua o encaminhamento da mensagem MR linhas 6–7. No caso da condição da linha 2 não ser satisfeita, a condição da linha 9 será verificada. Esta condição constata dois requisitos para a seleção do próximo salto até a EB: se o nó é descendente direto do MR_{ID} e se a rota é a de maior qualidade vista.

Para escolher o nó mais central a condição da linha 11 é testada. Nesta condição é

¹São os nós que possuem $RQ_x > RQ_t$, onde RQ é a qualidade da rota entre o nó t até a EB.

verificado se o nó MR_{ID} é o nó com a maior importância, segundo a centralidade Sink Betweenness. Finalmente, nas linhas 12–13 o nó MR_{ID} é identificado com o maior SBC visto (MAX_{SBC}) e como $PróximoSalto_t$. A Figura 4.4d ilustra o processo de seleção dos nós mais centrais do protocolo CRAL.

Algoritmo 3: Fase de Resposta

```

[1] Procedimento Resposta(Pacote MR)
[2]   se  $MR_{RQ} + RQ_{t \rightarrow MR_{ID}} > RQ_t \wedge MR_{ID} \in \psi_t$  então
[3]     // Mensagem recebida de um nó descendente
[4]     // Calcula a centralidade
[5]      $SBC_t \leftarrow SBC + \frac{Caminhos_t}{MR_P}$ 
[6]      $MR_{SBC} \leftarrow SBC_t$ 
[7]     Encaminha(MR)
[8]     retorna
[9]   senão se  $MR_{RQ} + RQ_{t \rightarrow MR_{ID}} \leq RQ_t$  então
[10]    // O nó que recebeu a MR é descendente direto de
[11]     $MR_{ID}$ 
[12]    se  $MR_{SBC} > MAX_{SBC}$  então
[13]       $MAX_{SBC} \leftarrow MR_{SBC}$ 
[14]       $PróximoSalto_t \leftarrow MR_{ID}$ 
[15]    fim
[16]    retorna
[17]  fim
[18] Procedimento Encaminha(Pacote MR)
[19]    $MR_{NH} \leftarrow PróximoSalto_t$ 
[20]    $MR_{QR} \leftarrow RQ_t$ 
[21]   broadcast MR
[22] Fim

```

A entrada e saída dos nós na rede são rotinas simples. Na primeira, o nó t envia para seus vizinhos diretos uma MG com o campo *comando* contendo o valor de entrada. Os nós ao receberem a mensagem MGs, enviam uma MD com suas informações atualizadas, isto irá disparar o ciclo de atualização de rotas abordadas nos Algoritmos 2 e 3, como consequência o recálculo da centralidade *Sink Betweenness* e seleção do nó mais central. Ao se desligar da rede, um nó envia uma MG com o campo *comando* contendo o valor de saída. Os nós filhos diretos ao receberem essa mensagem MG disparam uma nova mensagem MG com o comando configurado para entrada, isto desencadeia o estabelecimento de sua nova rota para os nós descendentes.

4.2.3 Funções de Fusão de Dados

A principal ideia da fusão de dados em RSSF é enviar mensagens agregadas, ao invés de enviar individualmente as mensagens dos nós para o sorvedouro. Isto faz da fusão de dados um poderoso mecanismo para reduzir o número de transmissões e consequentemente o consumo de energia da rede. Este trabalho foca em fusão de dados para RSSF no cenário em que o sorvedouro coleta os dados que são produzidos periodicamente pelos nós sensores. Neste sentido, três tipos de fusão de dados são as mais empregadas na literatura Solis & Obraczka [2006], sendo elas:

1. *Simple*: nesta técnica todos os nós fundem todas as mensagens que chegam durante uma pré-determinada quantidade de tempo t . Após esse tempo, apenas um pacote é enviado. A técnica *Simple* é de fácil implementação, não necessita de sincronização entre os nós, porém mensagens oriundas dos nós folha levam $t \times saltos$ para serem entregues ao sorvedouro;
2. *Periodic per hop simple (PerHop)*: nesta estratégia de fusão dos dados, os nós esperam uma pré-definida quantidade de tempo t para receber pelo menos uma mensagem de cada um dos seus descendentes. Caso uma mensagem de cada descendente seja recebida ou o tempo t seja atingido, uma mensagem com os dados fundidos é enviada para o próximo salto da rota até o sorvedouro. Cada nó espera o mesmo tempo t . A implementação desta técnica depende somente do conhecimento dos nós a respeito dos seus descendentes na árvore de roteamento. A técnica PerHop comparado com *Simple* apresenta maior sofisticação implementação. Com PerHop espera-se produzir dados mais atualizados, no que tange a rapidez com que os dados chegam à estação base. Entretanto, no pior caso, PerHop entrega os dados com tempo $t \times saltos$;
3. *Periodic per hop adjusted (PerHopAd)*: nesta abordagem os nós esperam uma quantidade de tempo t , porém este tempo depende da distância (em tempo) do nó até o sorvedouro. O tempo é calculado como $t = e - (shd * h)$, onde e é o período de tempo para coletar e reportar o valor da variável monitorada, shd é o tempo que um pacote leva para efetuar um salto na rede e h é à distância em saltos do nó até a EB. Além desse mecanismo de tempo, a função de fusão de dados dispara a mensagem agregada assim que uma mensagem de cada nó descendente é recebida.

Espera-se que os dados produzidos sejam mais atualizados ao utilizar PerHopAd quando comparado com a técnica PerHop. O tempo de espera t da função

PerHopAd faz com que aconteça um efeito de cascata, assim é possível reduzir ainda mais o tempo de entrega das mensagens agregadas, quando comparado com *PerHop*, sempre que seja frequente o envio de dados dos nós sensores. Porém PerHopAd requer a informação adicional sobre tempo de propagação sobre os enlaces.

4.2.4 Discussão sobre o protocolo CRAL

Manutenção da escolha dos enlaces é uma questão pertinente do algoritmo CRAL. Os estimadores de enlace ETX e ETT são baseados em software e apresentam monitoramento do tipo ativo (veja o Capítulo 3 Seção 2.4.2), ou seja, esses LQEs necessitam que mensagens de sonda sejam enviadas para qualificar os enlaces. Esta característica implica num compromisso entre estabelecimento/manutenção das rotas e o consumo de energia. Para equilibrar esse compromisso, o protocolo CTP utiliza o algoritmo Trickle Levis et al. [2003b], o qual é adaptável no sentido de enviar poucas sondas quando a rede é estável, e aumentar o número de sondas quando a rede é instável reagindo adequadamente a mudanças topológicas. A Figura 4.5 mostra a operação do *Trickle*, que inicia enviando sondas em curto intervalo de tempo, ao perceber a estabilidade da rede, o algoritmo aumenta o intervalo exponencialmente, ao passo que se uma inconsistência for identificada o intervalo será reinicializado. Este algoritmo também pode operar em conjunto com o algoritmo CRAL, visto que a sua arquitetura permite que sejam anexados diferentes algoritmos para controlar o Estimador de Enlace (vide Figura 4.2).

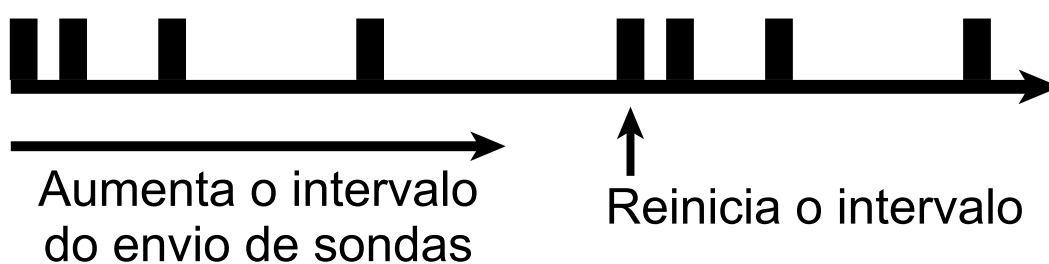


Figura 4.5: Operação do algoritmo Trickle Levis et al. [2003b].

Mitigar a questão dos nós centrais e próximos do sorvedouro tendem a consumir mais energia é outro problema do roteamento baseado em centralidade. Uma maneira de tratar este problema seria efetuar o balanceamento do tráfego entre os nós com maior centralidade e os com baixa centralidade. Outra alternativa seria rotear baseando-se na energia residual dos nós, fazendo com que a energia restante nos nós sensores seja critério de seleção para nós intermediários, além disso quando certo limiar de energia

seja atingido mensagens de alerta sejam enviadas, deste modo ativando um mecanismo de atualização de rotas com maior tempo de vida. Uma terceira abordagem seria usar técnicas de *energy hole*, seguindo modelos o apresentado por Ramos et al. [2014] e inicialmente definidos por Li & Mohapatra [2005] para mitigar o problema de maior consumo de energia dos nós próximo as EB.

Os pontos fortes e fracos do algoritmo CRAL também devem ser considerados ao utilizá-lo como protocolo de roteamento. Pode destacar como pontos fortes a fácil implementação de otimizações, pois sua arquitetura é bem dividida em plano de dados e plano de controle, desta forma permite-se fácil adaptações e manutenções. Outro ponto relevante do algoritmo CRAL é o eficiente modo como é realizado o cálculo da importância dos nós, o qual necessita que apenas duas inundações sejam realizadas implicando em baixo custo adicional para criar a estrutura de roteamento.

CRAL é um protocolo centrado nos nós, segundo a classificação dada por Krishnamachari Krishnamachari et al. [2008] e é otimizado para o *random-sources Model* em que os dados podem surgir em qualquer ponto da rede. Vale ressaltar que as otimizações realizadas para o protocolo CRAL favorecem a coleta de dados periódica em detrimento do monitoramento esporádico, em que os dados são reportados em intervalos esparsos.

Uma marca do protocolo CRAL é o roteamento pelos nós mais centrais, isto pode ser considerado como ponto dual, no sentido de que apresenta características que podem ser classificadas como fortes e como fracas. Os nós mais centrais podem ser pontos interessantes para fusão de dados que por sua vez pode reduzir o consumo de energia sendo, portanto, um ponto forte. Por outro lado, os nós mais centrais tendem a consumir mais energia, o que indica um ponto fraco. Assim, é aconselhável que protocolo CRAL seja empregado em redes que possua fácil acesso aos nós, facilitando a substituição dos módulos de energia, ou em redes em que os nós estejam acoplados a fontes de energia. Além disso, a identificação dos nós mais centrais pode ser útil para a substituição destes nós por outros mais poderosos contendo mais recursos computacionais, visto que o tráfego da rede tende a passar por esses nós.

Vale ressaltar que o protocolo foi construído baseado em primitivas que não se adequam bem a redes altamente dinâmicas, caso assim seja, o protocolo irá realizar diversas inundações para calcular a centralidade de todos os nós da rede. Isto pode ser relevante em caso de redes com severas restrições de energia.

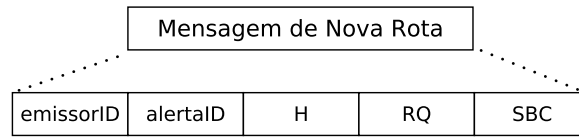


Figura 4.6: Operação dos algoritmos CRALs.

4.2.5 Mecanismo de balanceamento de Energia *Policy Aware*

Nesta seção será apresentado um mecanismo de consciência do consumo de energia, o qual será chamado de *Policy Aware*. Este mecanismo pode ser utilizado pelos nós implementados com o protocolo CRAL para mitigar o consumo exagerado de energia dos nós mais centrais da rede. O modo como o mecanismo foi construído permite uma diversidade de regras para atingir um balanceamento de consumo de energia.

O mecanismo de balanceamento acrescenta um tipo de pacote Mensagem de Nova Rota (MNR) ao protocolo CRAL. A Figura 4.6 mostra os campos da mensagem MNR que são: *emissorID* para indicar o endereço do nó que enviou o pacote, *alertaID* contendo o endereço do nó que enviou o alerta, *H* para indicar o número de saltos, *RQ* informando a qualidade da rota do nó que enviou a mensagem e *SBC* contendo a informação sobre a centralidade.

Algoritmo 4: Mecanismo inicial do balanceamento do consumo de energia.

```

[1] Procedimento Inicialização()
[2]    $MenorMetrica_t \leftarrow +\infty$ 
[3]    $MaiorCentralidade_t \leftarrow -\infty$ 
[4]   se  $Política = Limiar$  então
[5]     // Limiar foi atingido então envia um pacote de alerta
[6]      $MG_{comando} \leftarrow Alerta$ 
[7]     Broadcast MG
[8]     Atualiza Limiar
[9]     retorna
[10]  senão
[11]    Incrementa(Política)
[12]    retorna
[13]  fim
[14] Fim

```

No Algoritmo 4, a fase inicial do mecanismo de balanceamento do consumo de energia é exibida. Nas linhas 2 e 3 variáveis para manter o rastro da melhor rota e da maior centralidade vista são inicializadas. Na linha 4 é verificado se a política utilizada atingiu certo limiar. Esta política pode ser, por exemplo, um nível de bateria, número de mensagens transmitidas ou recebidas, ou ainda algo mais sofisticado como

uma combinação dos supracitados. Caso o limiar seja atingido, é desencadeado o mecanismo de alteração das rotas dos nós descendentes ao nó MG_{ID} com o comando alerta (linha 7), na linha 8 o nó pode atualizar a informação do limiar. Em caso do limiar ainda não ter sido alcançado, uma função $Incrementa(Política)$ na linha 11 é utilizada para atualizar o valor da política, por exemplo, caso a política seja um contador de transmissões a função $Incrementa(Política)$ incrementaria este contador.

O Algoritmo 5 é exibido o processo de atualização dos enlaces devido ao envio de uma mensagem contendo um alerta. No Algoritmo 5 são exibidos dois processos distintos um para tratar a interceptação de mensagens de gerência contendo um Alerta e, um processo para tratar mensagens de Nova Rota. Na linha 1 é exibido o processamento para pacotes de gerência com o comando Alerta, ao receber um alerta é verificado se a mensagem é oriunda de um nó em um mesmo nível ou um mais baixo (próximo da EB) na árvore de roteamento, em caso verdadeiro, significa que o interceptador do alerta pode estar qualificado para ser próximo salto dos descendentes do nó emissor do alerta, portanto o interceptador deve fazer um *broadcast* se elegendo com possível intermediário de rota (linha 3). Já na linha 5 é verificado se o nó que recebeu o alerta é descendente direto do nó emissor do alerta, ou seja, se o nó t tem como próximo salto o nó emissor do alerta. Neste caso, as variáveis para manter o rastro da melhor rota e maior centralidade são reinicializadas.

Na linha 11 do Algoritmo 5 é exibido o processamento do pacote MNR. Inicialmente na linha 12 é checado se a mensagem de nova rota é para os descendentes do nó $MNR_{alertaID}$ e se é a melhor rota vista desde o momento da interceptação da mensagem de Alerta. Caso isso se verifique, a condição da linha 13 também será checada para selecionar o nó mais central, em caso de empate na qualidade das rotas. A condição linha 13 sendo verdadeira ou não desencadeia a atualização apropriadas das rotas (linhas 11 e 21). Finalmente, o nó que acabou de atualizar suas rotas encaminha as suas informações de nova rota para os descendentes do nó $MNV_{alertaID}$ que são alcançáveis, permitindo ao máximo que os descendentes do nó $MNV_{alertaID}$ tenham opções de rotas.

4.2.5.1 Discussão sobre Policy Aware

Como avaliado na Seção 4.5 o uso do mecanismo de *Policy Aware* pode levar a ganho significativo no equilíbrio do consumo de energia na rede, aumentando o tempo de da rede e, em especial, dos nós com maior centralidade e próximos da EB.

Por um lado, a técnica apresenta pode obter os benefícios supracitados. Por outro lado, é evidente que a estratégia adotada para balancear o consumo de energia introduz

Algoritmo 5: Mecanismo para balanceamento do consumo de energia.

```

[1] Procedimento InterceptaMA(Pacote MG)
[2]   se  $MG_{saltos} \geq Saltos_t$  então
[3]     Broadcast MNR
[4]     retorna
[5]   senão se  $MG_{ID} = PróximoSalto_t$  então
[6]      $MenorMetrica_t \leftarrow +\infty$ 
[7]      $MaiorCentralidade_t \leftarrow -\infty$ 
[8]     retorna
[9]   fim
[10] Fim
[11] Procedimento InterceptaMNR(Pacote MNR)
[12]   se  $MNR_{alertaID} = PróximoSalto_t \wedge MNR_{RQ} \leq MenorMetrica_t$ 
[13]     então
[14]       se
[15]          $MNR_{RQ} = MenorMetrica_t \wedge MNR_{SBC} > MaiorCentralidade_t$ 
[16]           então
[17]              $MaiorCentralidade_t \leftarrow MNR_{SBC}$ 
[18]              $PróximoSalto_t \leftarrow MNR_{emissorID}$ 
[19]              $Saltos_t \leftarrow MNR_H$ 
[20]           senão
[21]              $MenorMetrica_t \leftarrow MNR_{RQ}$ 
[22]              $PróximoSalto_t \leftarrow MNR_{emissorID}$ 
[23]              $Saltos_t \leftarrow MNR_H$ 
[24]              $RQ_t \leftarrow MNR_{RQ} + RQ_{t \rightarrow MNR_{emissorID}}$ 
[25]           fim
[26]         // Atualiza todos os campos do MNR antes de
[27]         encaminhá-lo
[28]         Broadcast MNR retorna
[29]       fim
[30]   fim
[31] Fim

```

novos compromissos como o envio de novos pacotes de controle na rede e alteração das rotas de fusão dentre outras. Por exemplo, apesar desse custo adicional o uso da técnica conduz ao aumento do tempo de vida dos nós sensores mais centrais, para o aumento do tempo de vida destes nós é preciso aumentar o número de nós Steiners na rede, isto é, o roteamento por nós que não são tão centrais. Vale ressaltar que na presença de enlaces altamente assimétricos o uso do mecanismo *Policy Aware* pode conduzir à deterioração da vazão da rede. Portanto, ao optar pelo uso desta técnica deve-se ter em mente todos os custos envolvidos.

4.3 Perda de dados

Esta seção aborda o problema da perda de dados em protocolos de coleta em RSSF, dentre outras questões serão abordadas técnicas para tratar essas perdas e a principal inspiração que levou a proposta do segundo objeto de análise deste trabalho, a saber o protocolo XCTP.

Utilizar técnicas de fusão de dados faz com que em algum momento do roteamento dos pacotes de dados contenham informações sobre muitas leituras. Isto introduz um problema para o roteamento que é a garantia de entrega destes pacotes. A perda de um pacote que contém as informações sobre várias leituras conduz a um grande impacto no desempenho da coleta de dados.

Uma das técnicas mais utilizadas para reduzir a perda de dados é enviar proativamente o dado múltiplas vezes. É notório que esta abordagem introduz um compromisso entre consumo de energia (pelo envio redundante dos dados) e probabilidade de entrega dos pacotes. Além disso o envio redundante pode não ser adequado para algumas aplicações, causando que protocolos no nível de aplicações tratem os dados redundantes.

Em Solis & Obraczka [2006] são apresentados 3 diferentes mecanismos de transmissão redundantes: i) *Double-send*, o qual envia cada pacote duas vezes; ii) *Max-send* envia cada pacote que o dado foi agregado ao longo do caminho até a EB, requerendo um campo adicional no pacote para contagem de fusões; iii) *Adaptive-send*, o qual utiliza um algoritmo mais complexos que utiliza uma relação entre o número aceitável de perdas e a probabilidade de perda da rota. Solis argumenta que *adaptive-send* e *Max-send* atingem os maiores índices de entrega de dados, entretanto são as técnicas que introduzem mais pacotes redundantes na rede, causando maior consumo de energia. Vale pontuar que em nenhuma dessas técnicas o valor de 100% de entrega é alcançado.

Uma forma de atingir os 100% de entrega é construir rotas que permitam que os

pacotes de dados sejam confirmados, ou seja, rotas bidirecionais entre os participantes da comunicação. Por exemplo, o uso de protocolos tais como AODV, RPL e Hydro provê tais rotas e permitem que mensagens de controle e confirmação sejam trocadas entre os nós. Por um lado, estes protocolos atendem ao requisito de prover mecanismos para que seja possível garantir a entrega dos dados. Por outro lado, estes protocolos geralmente consomem demasiadamente recursos dos nós sensores, principalmente, memória para armazenamento das rotas ponto-a-ponto. Este problema foi a principal fonte inspiradora para o desenvolvimento do segundo objeto de estudo deste Trabalho.

No Capítulo 5 será apresentada uma abordagem para solucionar o problema da coleta confiável em RSSF e que mitiga o consumo de memória e custo adicional para criar e manter as rotas, principais problemas dos protocolos ponto-a-ponto.

4.4 Análise do Protocolo

Nesta seção será realizada uma análise do custo assintótico do protocolo CRAL em relação ao consumo de memória e quantidade de mensagens enviadas, utilizando a notação "Big-Oh". Para esta análise consideramos uma rede com N nós sensores e que apresenta uma densidade média constante de v vizinhos, isto é, os nós da rede, em média, conseguem manter comunicação direta com v vértices.

O primeiro aspecto a ser analisado é o consumo de memória do protocolo. CRAL como exibido na Seção 4.2.2 mantém na tabela de rotas somente uma entrada, desta maneira dado que a rede apresenta N nós, o consumo de memória do algoritmo é $O(m)$, onde m é o maximal número de arestas na árvore de roteamento. A tabela de rotas do algoritmo CRAL pode ser uma lista, deste modo supondo que a lista tenha em média o tamanho v , o custo de memória não é alterado sendo, portanto $O(v \times m)$, onde v é a densidade da rede. Comparando o custo de espaço de armazenamento do protocolo CRAL com os apresentados na literatura (vide Tabela 3.1), o protocolo RBC possui custo de espaço $O(m)$, os demais protocolos não apresentam tais análises.

No aspecto da quantidade de mensagens enviadas para que seja estabelecida a árvore de roteamento (custo adicional de comunicação). O custo do algoritmo CRAL é de N pacotes com cada nó disseminando a consulta original que engatilha a formação da árvore de rotas. Na fase de resposta, se cada nó filho responde a consulta inicial existe o custo adicional de $N - 1$ pacotes uma vez que cada nó, exceto a raiz, será um nó filho. Então o custo total do estabelecimento da árvore é $2N - 1$ pacotes, ou seja, $O(N)$. A Tabela 4.2 apresenta o custo para estabelecimento das rotas para cada um dos algoritmos relacionados, SPT apresenta *overhead* N pacotes, pois não se faz necessário

Tabela 4.2: Custo para estabelecimento das rotas para os protocolos relacionados

Protocolo	Complexidade
CRAL	$O(N)$
SPT	$O(N)$
CNS	$O(e \times N)$
RBC	$O(N^2 \times m)$
CT	$O(N)$
SPBC	$O(N \times E)$
TLC	$O(N \times E)$
FBC	–
InFRA	–
DAARP	–
CTP	–

E – número de arestas

N – número de nós na rede

e – número de eventos que ocorrem na rede

m – maximal número de aresta na árvore de rotas

que os nós respondam a requisição de rota. O protocolo CNS apresenta *overhead* superior de $O(e \times N)$, pois precisa de uma inundação para cada evento identificado. CT apresenta custo $O(2 \times N)$ pois precisa de duas inundações para estabelecer suas rotas. O algoritmo RBC necessita de $O(N^2 m)$ pacotes de controle para criar as rotas. Já os protocolos InFRA, DAARP e CTP não apresentaram demonstrações sobre o custo dos algoritmos utilizando notação assintótica.

Esta análise permite inferir que dos protocolos apresentados CRAL é um dos que apresenta menor custo adicional para estabelecimento das rotas, portanto um protocolo competitivo e eficiente em face aos existentes na literatura.

4.5 Avaliação

Nesta seção será apresentada a avaliação das versões CRAL-LD e CRAL-FD. Esta seção está dividida em duas partes: Metodologia e Resultados. Na primeira parte será descrita a metodologia utilizada para realizar os experimentos, em que serão detalhados os cenários utilizados, bem como todo o arcabouço ferramental empregado para executar os experimentos e coletar os resultados. Na parte de Resultados, os resultados obtidos sobre os experimentos são apresentados e para cada um são feitas análises de forma gráfica e escrita.

4.5.1 Metodologia

Dos algoritmos apresentados na Tabela 3.1, RBC e seus derivados não foram projetados de modo distribuído e para RSSF, portanto não podem ser comparados com CRAL-LD e CRAL-FD. InFRA e DAARP são de classes distintas de algoritmos de coleta de dados, eles não apresentam resultados satisfatórios quando eventos ocorrem de forma não aglomerada, além disso não apresentam código aberto à comunidade. CTP não utiliza centralidade e não foi projetado para operar com fusão de dados. CNS é um algoritmo clássico e apresenta resultados inferiores aos apresentados pelos algoritmos CT e SPT. Portanto, serão efetuadas comparações entre as versões do algoritmos CRAL, bem como entre os algoritmos CT e SPT. Sendo que SPT² será utilizado como *baseline*.

Tabela 4.3: Resumo dos modelos utilizados para simulações com o Sinalgo

Modelo	Valor
Mobilidade	Sem Mobilidade
Transmissão	802.11b Tabela 4.4
Conectividade	UDG Raio máximo 796
Confiabilidade	Arestas ponderadas]0, 1]

Todos os protocolos analisados foram implementados para o simulador Sinalgo *v.0.75.3* Sinalgo [2014], bem como a modelagem para rádios com múltiplas taxas de transmissões. A Tabela 4.3 apresenta, em resumo, os modelos utilizados para o simulador Sinalgo em todos os experimentos aqui realizados. Dos modelos providos pelo Sinalgo foram utilizados: i) UDG como modelo de conectividade; ii) SINR como modelo de interferência; iii) *Random Distribution* para dispor os nós sensores na região de interesse.

Adicionalmente foram implementados os modelos de confiabilidade e transmissão para o Sinalgo. No modelo de confiabilidade, as arestas do grafo que representa os enlaces da rede foram ponderadas seguindo uma distribuição normal de valores entre]0, 1], em que valores próximos a 0 representam enlaces de baixa qualidade e valores próximos a 1 retratam enlaces de boa qualidade. Desta forma, para cada pacote enviado por uma aresta, um número aleatório entre [0, 1] é gerado para verificar se o pacote foi recebido com sucesso ou não. Em caso deste número apresentar valor inferior ao peso da aresta a mensagem é decodificada corretamente pelo receptor, em caso contrário o pacote não foi recebido corretamente deve ser re-transmitido.

²SPT por ser um protocolo do tipo AC não é exibido em alguns resultados, pois ou não condiz com a avaliação ou não apresenta valores não competitivos com as abordagens otimizadas para roteamento DC apresentadas.

O modelo com múltiplas taxas de transmissão utilizado neste trabalho é baseado no padrão IEEE 802.11b IEEE 802.11b-1999 [1999]. Os parâmetros de alcance são baseados nas especificações do rádio Lucent ORiNOCO PC Card Orinoco wireless networks [2014]. A Figura 4.7 e a Tabela 4.4 mostram os parâmetros para modelagem com múltiplas taxas de transmissão, em que a distância entre os nós influi na taxa de transmissão. A Tabela 4.4 também mostra o tempo de propagação para cada uma das taxas do padrão 802.11b. O protocolo que não considera múltiplas taxas será assumido o uso da taxa mínima, ou seja, 1 Mbps com alcance máximo 796.

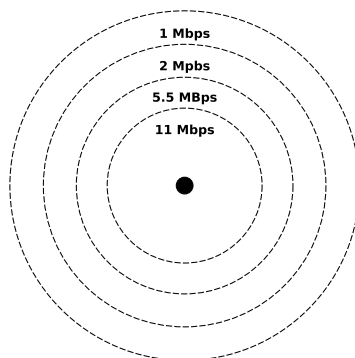


Figura 4.7: Alcance do rádio sobre o padrão 802.11b.

Tabela 4.4: Parâmetros da modelagem 802.11b para RSSF [Awerbuch et al., 2004; Orinoco wireless networks, 2014]

Taxa (Mbps)	Alcance Máximo (Metros)	Tempo de Propagação Pacotes de 1500 Bytes (μsec)
11.0	399	2542
5.5	531	3673
2.0	669	7634
1.0	796	13858

Diferentes configurações do cenário foram experimentadas para analisar o comportamento dos algoritmos a seguir destacamos parâmetros que foram variados:

- i) *Posição da EB*: a posição do nó raiz da árvore de roteamento impacta diretamente na profundidade dos nós, portanto este parâmetro foi variado podendo assumir os seguintes valores: borda, centro ou aleatório. Em geral, a estação base posicionada na borda da rede conduz a árvores de maior profundidade ao passo que o posicionamento da EB no centro gera árvores com menor altitude. O posicionamento aleatório apresenta valores de altitude da árvore intermediários.

- ii) *Número de nós sensores*: para averiguar a escalabilidade dos protocolos apresentados variou-se o número de nós sensores dispostos na região de interesse. Os portes das redes experimentadas foram classificados em três grupos: pequeno porte (128 nós), médio porte (256 e 512 nós sensores) e grande porte (1024 nós). As dimensões da área monitorada foram ajustadas para manter a densidade de $20\text{nós}/m^2$.
- iii) *Tráfego*: no que tange os parâmetros de tráfego, a cada simulação cada nó envia 20 pacotes de dados. O período para reportar dados é de 1s durante 20s desde o estabelecimento da árvore de roteamento. Geralmente a construção da árvore de rotas é finalizada nos 3s iniciais após a requisição da EB. Por esse motivo, os dados analisados foram coletados após 10s de simulação deixando, portanto, uma margem para a construção e estabilização da árvore de rotas.
- iv) *Função de Fusão de Dados*: as técnicas de fusão de dados utilizadas nos experimentos são as descritas na Seção 4.2.3 sendo delas: Simple, PerHop e PerHopAd.
- v) *Miscelâneos*: um número máximo de retransmissões foi estabelecido pelo fato da rede apresentar perda de pacotes. O número máximo de retransmissões devido à perda de pacotes é 10. Os enlaces apresentam taxa de perda que seguem uma distribuição normal com valores $]0, 1]$.

O cenário padrão utilizado nos experimentos é resumido na Tabela 4.5. Será comentado quando algum dos parâmetros é variado para realizar o experimento.

Tabela 4.5: Cenário padrão de simulação

Parâmetro	Valor
Estação Base	1 borda
Número de nós sensores	1024
Rádio Padrão	802.11b
Taxa de perda no enlace (valor/aresta)	$0 < p \leq 1$
Densidade ($\text{nós}/m^2$)	20
Mensagens de dados (mensagens/nó)	20
Duração do envio das mensagens de dados (s)	20
Período de envio das mensagens (s)	1
Número padrão de retransmissões	10
Função para fusão de dados	Nenhuma
Tempo de espera para fusão de dados (s)	1

As análises aqui realizadas têm como objetivos principais validar os algoritmos propostos e avaliar o desempenho dos algoritmos CRAL-LD e CRAL-FD em relação ao estado da arte. Foram utilizadas as seguintes métricas para comparação:

- 1) *Consumo de energia*: será analisado o consumo médio em Joules dos nós sensores na rede. Os autores Balasubramanian et al. [2010] apresentam um estudo empírico sobre o consumo de energia dos rádios habilitados para Wi-Fi. Deste estudo os autores derivam um modelo linear exibido na Equação 4.1, onde x é o tamanho do pacote de dados em bytes. Este modelo foi empregado em nossas simulações para estimar o consumo de energia dos nós sensores.

$$E_{WiFi}(x) = 0.007(x) + 5.9J \quad (4.1)$$

Neste quesito também será avaliado o compromisso entre o consumo de energia e o nível de centralidade do nó sensor.

- 2) *Número de Steiner nodes*: o número de nós Steiner informa o quão bom é o algoritmo na aproximação da árvore de Steiner. Neste trabalho, esta métrica é a quantidade de nós intermediários na árvore de roteamento. Desta forma, quanto menor o número desses nós melhor será a árvore de roteamento, no sentido de fusão de dados;
- 3) *Eficiência*: a eficiência dos protocolos neste trabalho é definida como o número de transmissões e retransmissões necessárias para entregar os pacotes de dados à EB pelo número de pacotes enviados.
- 4) *Confiabilidade*: a confiabilidade do protocolo é definida como o número de pacotes entregues pelo número de pacotes enviados à EB. Esta métrica informa o quanto as rotas construídas pelos protocolos são confiáveis na entrega dos dados.
- 5) *Efeito Overhearing*: este efeito diz às mensagens interceptadas pelos nós sensores. O efeito *overhearing* é definido como o número de mensagens interceptadas *não* destinadas ao nó Basu & Redi [2004]. Valores altos de *overhearing* implicam em maior consumo desnecessário de energia pelos nós sensores.
- 6) *Latência*: a definição de latência utilizada neste trabalho é o tempo médio pelo número de saltos que um pacote de dados possui;
- 7) *Qualidade da rota*: esta métrica é definida pelo valor do ETX da rota, ou seja, o ETX de cada enlace pertencente à árvore de roteamento. Esta métrica informa a qualidade da rota no que tange o número necessário de transmissões e retransmissões para entregar os pacotes de dados;

8) *Freshness*: ao utilizar técnicas para fusão de dados, a velocidade em que os dados são coletados pelas EBs são de extrema relevância. O *Freshness* dos dados indica o quão rápido os dados estão chegando ao sorvedouro. Altos valores de freshness indicam que os dados apresentam idade avançada, isto é, os dados foram gerados a um longo tempo atrás. Formalmente o *Freshness* de um pacote é dado por: $\Delta(pkt) = T_c - T_g$, onde T_c é o tempo em que o pacote alcança a EB e T_g é o tempo que o dado foi gerado.

Para realizar os experimentos foram criadas 33 topologias diferentes para cada um dos portes de rede, em cada uma das topologias realizou-se 33 simulações, totalizando 4356 simulações. Para cada gráfico apresentado na Seção 4.5.2 a curva representa os valores médios das simulações e as barras, quando existem, representam o intervalo de confiança de 95%.

4.5.2 Resultado de Simulação

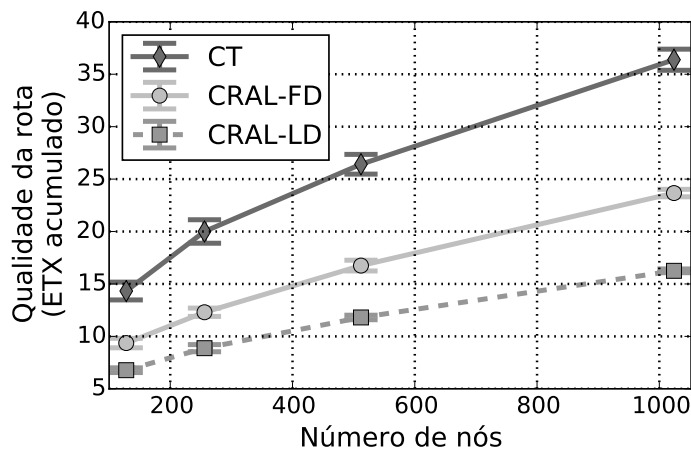


Figura 4.8: ETX médio das rotas variando número de nós sensores.

Confiabilidade: inicialmente para mostrar a melhor qualidade das rotas encontradas pelos algoritmos CRAL-LD e CRAL-FD foram calculados ETX acumulado médio das rotas. A Figura 4.8 mostra a qualidade das rotas encontradas pelos algoritmos CRAL-LD, CRAL-FD e CT variando o porte da rede. Fica evidente que CRAL-LD encontra rotas com menor número de retransmissões para entregar os dados, pois é otimizado para encontrar rotas com baixo ETX. Já a versão CRAL-FD apresenta maior ETX acumulado que CRAL-LD, pois prioriza rotas com alta vazão além da qualidade, isto geralmente conduz a maior quantidade de saltos e conseqüentemente maior ETX

de rota. CT em todos os cenários apresenta rotas de baixa qualidade, isto se deve ao uso do *hop count* como métrica.

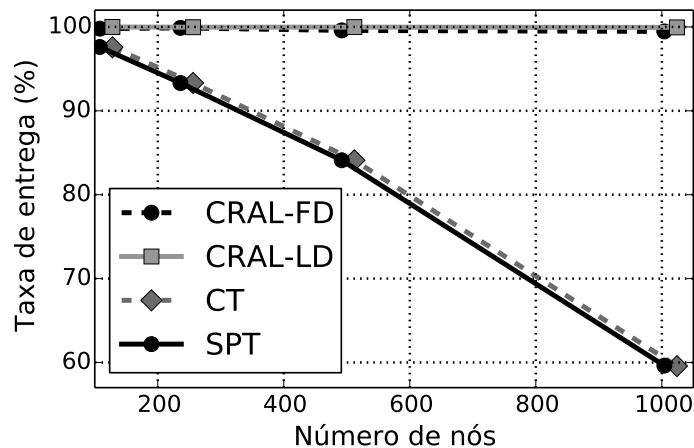


Figura 4.9: Taxa de entrega das mensagens de dados.

Para mostrar que as estratégias aqui apresentadas são confiáveis, neste experimento utilizou-se o número máximo de 10 retransmissões por salto para os pacotes de dados e então foi verificada a taxa de entrega dos pacotes variando o porte da rede. Assim, é possível verificar a confiabilidade das rotas calculadas pelos diferentes algoritmos.

A Figura 4.9 exibe os resultados para cada um dos protocolos avaliados. Percebe-se que CRAL-LD e CRAL-FD apresentam alta taxa média de entrega dos pacotes de dados, alcançando 99.6% e 99.1% respectivamente, enquanto que CT e SPT apresentam queda de confiabilidade quase linear, quando o crescimento da rede é exponencial. CT e SPT alcançam valores similares, conseguindo entregar aproximadamente 60% de entrega dos dados gerados em redes de grande porte, isto pode ser explicado pelos enlaces de baixa qualidade escolhidos e descarte de pacotes. Portanto, é de grande importância a consideração dos enlaces com taxa de perda intermediária na escolha das rotas.

As demais análises sobre os protocolos foram realizadas utilizando reenvio ilimitado dos pacotes, ou seja, o número de transmissões é infinito. Desta forma, as mensagens são retransmitidas até que sejam entregues até a EB.

Eficiência e Robustez: foram feitas análises sobre o consumo de energia das abordagens CRAL-LD, CRAL-FD, CT e SPT em termos de bateria e efeito *overhearing*. Na Figura 4.10 é apresentado o consumo em Joules para cada algoritmo, percebe-se que CRAL-LD e CRAL-FD apresentam menor consumo de energia, porém CRAL-FD

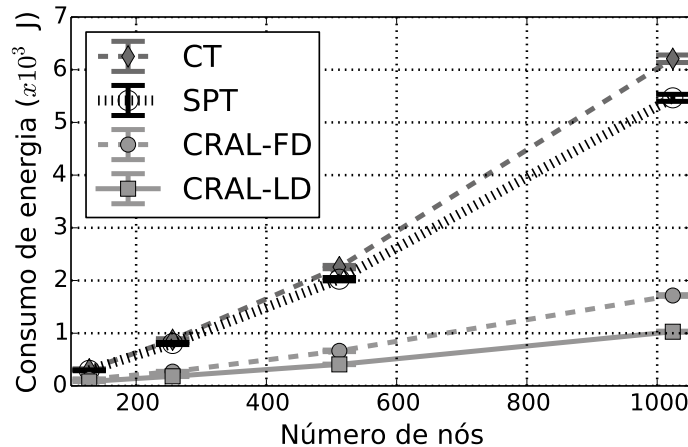


Figura 4.10: Consumo de energia em Joules variando o número de nós sensores.

supera o consumo de CRAL-LD, isto se deve ao fato de CRAL-FD apresentar rotas com ETX acumulado e número de saltos ligeiramente superiores ao CRAL-LD. Ambos CRAL-LD e CRAL-FD apresentam drástica redução de consumo de energia quando comparados com SPT e CT, quando em redes de grande porte CRAL-LD e CRAL-FD economizam até 6 vezes mais energia do que os concorrentes. SPT e CT são similares para redes de tamanho pequeno e intermediário, porém CT apresenta maior consumo energético em redes de grande porte.

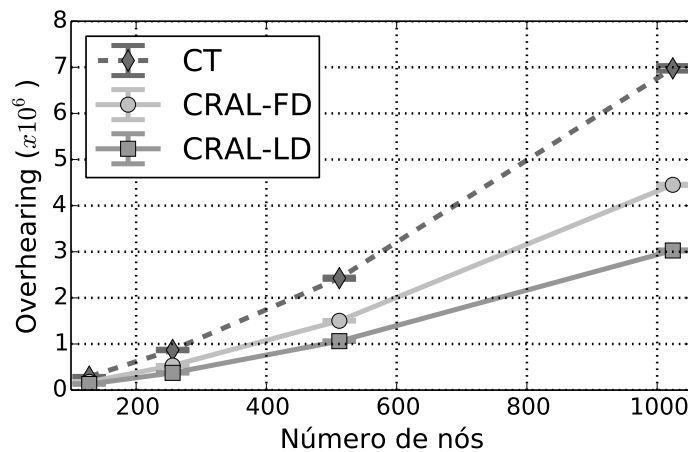


Figura 4.11: Efeito *overhearding* variando o número de nós sensores.

O efeito *overhearding* é um resultado complementar que explica o consumo de energia dos nós sensores. A Figura 4.11 no eixo X o tamanho da rede e no eixo Y o número de mensagens interceptadas não destinadas ao nó. Quando a rede é de pequeno porte a diferença entre os protocolos não é evidente, entretanto quando o tamanho da rede cresce a desigualdade se torna clara. CRAL-LD é o algoritmo que apresenta

menores valores, seguido por CRAL-FD e, por fim, CT. Com redes de grande porte, CRAL apresentam valores que chegam a ser 3 vezes menores quando comparado com CT e até 10 vezes menores do que os apresentados por SPT.

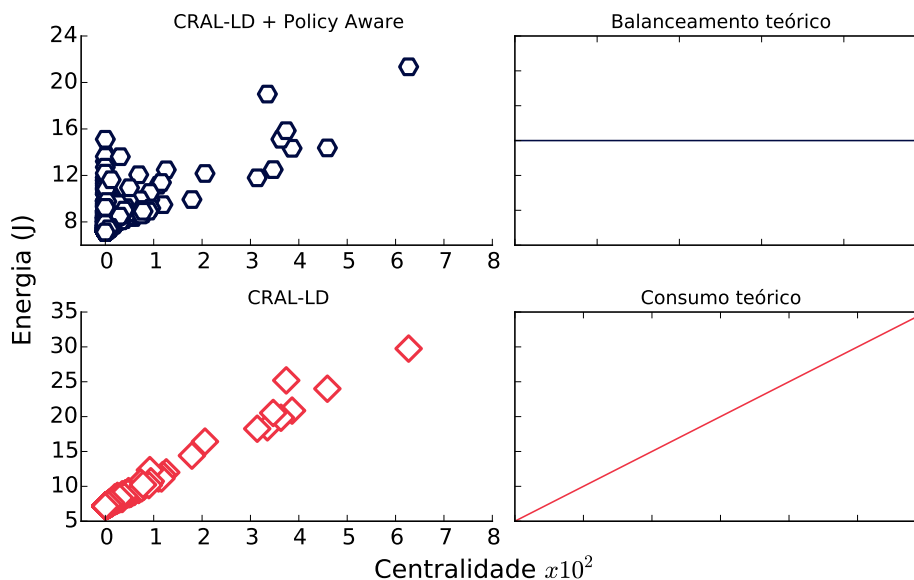


Figura 4.12: Compromisso entre o uso da técnica *Policy Aware* e o consumo de energia.

O compromisso entre o uso da técnica *Policy Aware* para balancear o consumo de energia entre os nós menos e mais centrais é apresentado na Figura 4.12. A figura destaca *Policy Aware* e o protocolo CRAL-LD, a política utilizada foi o número de encaminhamentos de pacotes realizados pelo nó e o limiar estabelecido foi aproximadamente 1000 encaminhamentos (o que dá aproximadamente 5% do tráfego da rede descrito na Tabela 4.5), ou seja, ao encaminhar 5% das mensagens o nó dispara um alerta para atualização das rotas dos descendentes. O balanceamento teórico, consumo do protocolo CRAL-LD sem *Policy Aware* e *Consumo teórico* também são exibidos na Figura 4.12. É possível notar que a distribuição do consumo de energia é melhor no gráfico *CRAL-LD + Policy Aware* do que no gráfico *CRAL-LD*. Ressaltando o nó mais central, percebe-se a diminuição do seu consumo de energia que cai de aproximadamente $31J$ para $21J$, outra observação é que possuem baixa centralidade passam a assumir o controle do roteamento e assim distribuindo o consumo de energia como é possível notar no gráfico *CRAL-LD + Policy Aware*. É possível concluir que *Policy Aware* realiza um balanceamento melhor do consumo de energia.

A Figura 4.13 apresenta a latência em milissegundos para diferentes portes da rede. Neste experimento, a latência é o tempo médio por saltos que as mensagens levam para alcançar a EB. Percebemos que a estratégia que utiliza estimador de enlace que considera múltiplas taxas de transmissão (CRAL-FD) obtém melhores resultados.

CT não é exibido, visto que seus resultados apresentam valores superiores a 30% aos aferidos por CRAL-LD ou CRAL-FD. Um resultado complementar é o número de transmissões por dados reportados. Este deve ser entendido como eficiência, isto é, a entrega dos dados com menor quantidade de transmissões possível.

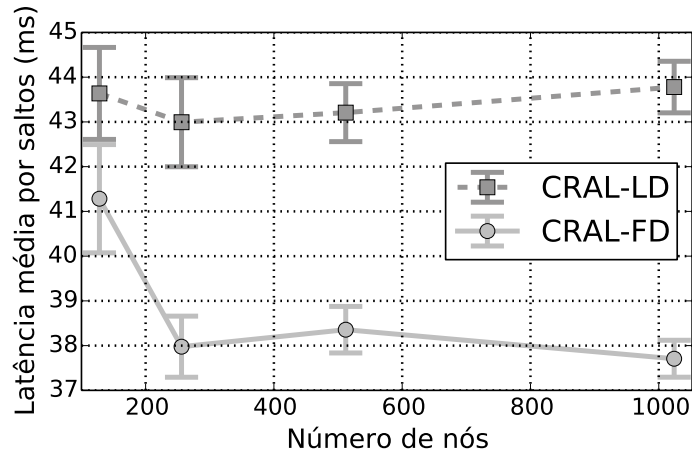


Figura 4.13: Latência média por saltos variando o número de nós sensores.

A análise dos experimentos sobre eficiência dos protocolos permite afirmar que CRAL-LD e CRAL-FD são mais eficientes quando comparados com CT e SPT. A Figura 4.14 exhibe no eixo X o porte das redes analisadas e no eixo Y o número de transmissões por dados reportados. CRAL-LD e CRAL-FD apresentam resultados que não superam 8 transmissões por mensagem de dados, ao passo que SPT e CT superam 25 transmissões por mensagem de dados. Alto número de transmissões para entregar os dados acarreta em desperdício de energia, aumento do efeito *overhearing* e alta latência na rede.

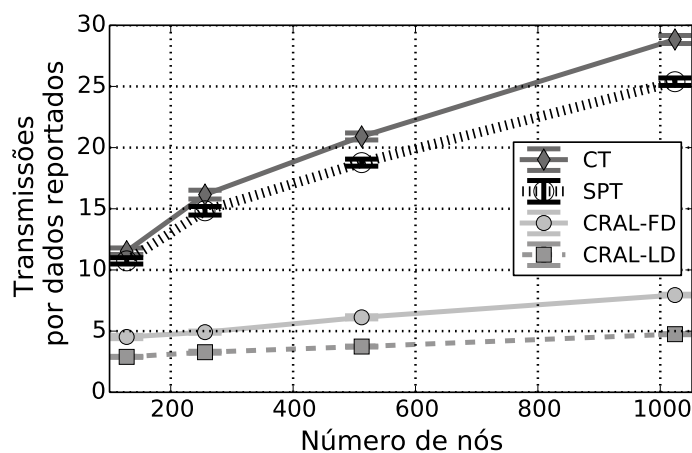


Figura 4.14: Número de transmissões por dados reportados variando o porte da rede.

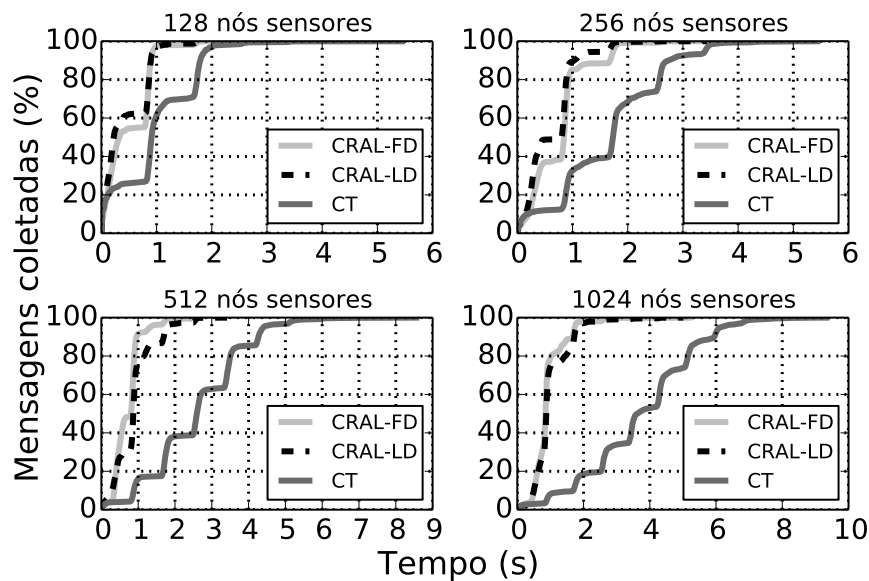


Figura 4.15: Porcentagem de mensagens coletadas no tempo.

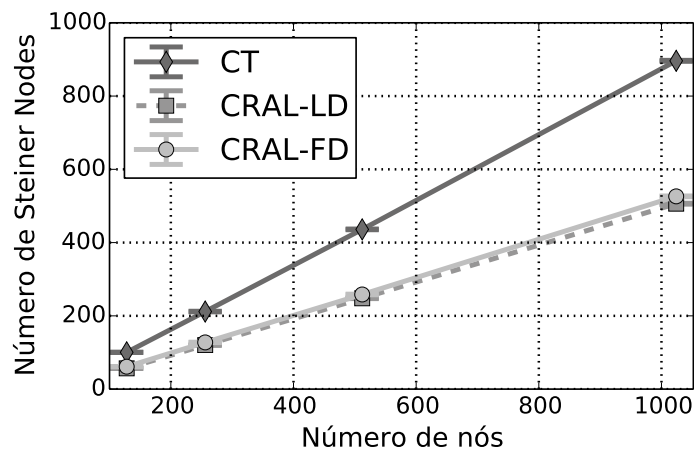


Figura 4.16: Steiner Nodes variando o número de nós sensores.

Flexibilidade e favorecimento à fusão de dados: o baixo número de nós Steiner representa árvore de roteamento de melhor qualidade para fusão de dados e, como consequência, maior economia de energia. A Figura 4.16 mostra o número de nós Steiner para cada um dos algoritmos de roteamento. CT apresenta maior número de Steiner *Nodes* quando comparado com CRAL-LD e CRAL-FD para todos os cenários considerados. A maior diferença é atingida com redes de grande porte, em que CRAL-LD e CRAL-FD apresentam aproximadamente metade dos Steiner *Nodes* de CT.

Para avaliar a flexibilidade das abordagens aqui apresentadas as diferentes técnicas de fusão de dados, discutidas na Seção 4.2.3, foram incorporadas aos algoritmos. O desempenho dos algoritmos sobre o *freshness* dos dados coletados é exibido nas

Figuras 4.15, 4.17 e 4.18.

Na Figura 4.15 são mostrados, para cada protocolo, a porcentagem de coletas e o tempo de entrega. O cenário utilizado neste experimento foi utilizar *PerHopAd* como função de fusão de dados, posicionando a EB no centro da rede, com esta configuração todos os protocolos apresentam seus melhores resultados. É fácil perceber que os algoritmos CRAL entregam rapidamente as mensagens. Em redes de grande porte, CRAL-LD e CRAL-FD entrega mais que 90% das mensagens com apenas 2s, enquanto CT entrega somente 20% com o mesmo tempo. Esta característica é similar em todos os cenários avaliados. Quando os algoritmos CRAL terminam a entrega de todos os dados, CT leva em média o dobro desse tempo para finalizar a transmissão dos dados.

Nas Figuras 4.17 e 4.18 são apresentados os resultados do *freshness* variando a posição da EB, porte da rede e funções de fusão dos dados para os algoritmos CRALs. No eixo X das figuras mostram as várias funções de fusão de dados, a barra *None* representa o algoritmo sem função de fusão de dados. No eixo Y é apresentado o número de mensagens enviadas, sendo que cada nó envia 20 mensagens totalizando $20 \times < \text{Porte da rede} >$. O *Freshness* neste experimento foi discretizado como 1s, 2s, ... (conforme mostrado na legenda das figuras), deste modo tem-se para cada função as mensagens recebidas com até 1s, 2s, ... de atraso, o qual chamaremos de idade da mensagem.

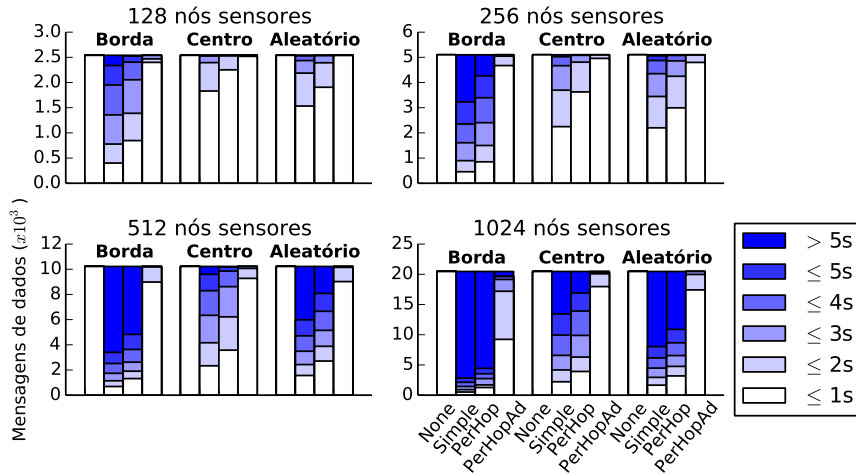


Figura 4.17: *Freshness* para o algoritmo CRAL-LD.

Ao comparar as duas abordagens percebe-se pouca mudança. Assim, pode-se escolher CRAL-FD se o tempo de entrega de cada mensagem é um quesito mais importante ou CRAL-LD caso a taxa de entrega seja o fator mais relevante. A melhor configuração para este experimento é o uso do protocolo CRAL em conjunto com a

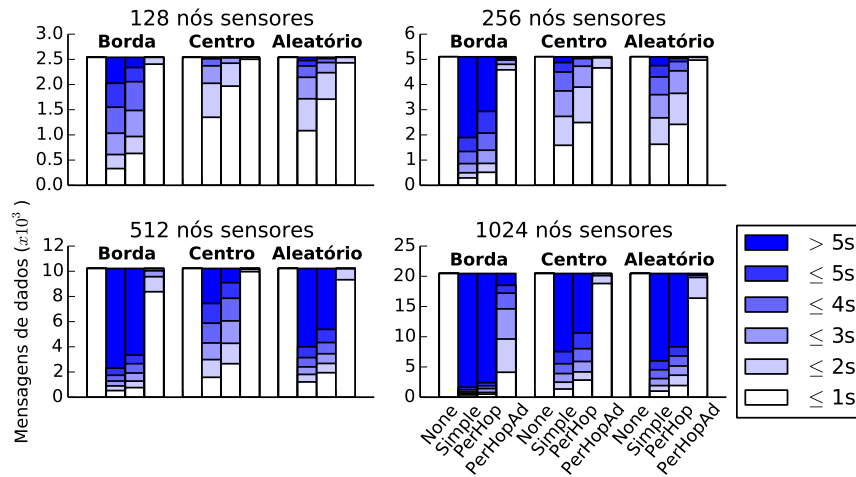


Figura 4.18: *Freshness* para o algoritmo CRAL-FD.

função PerHopAd, a qual mais e se aproxima dos tempos de entrega sem o uso das funções de fusão de dados (barra *None*).

Em relação à idade dos dados com e sem uso de funções de fusão de dados, percebe-se que as funções *Simple* e *PerHop* não são tão ágeis quanto *PerHopAd*, isto se deve ao efeito cascata da função *PerHopAd*, o qual favorece a fusão de dados ao passo que mantém o tempo baixo de entrega dos dados fundidos. Assim, a combinação do *PerHopAd* com CRAL torna a fusão de dados ágil em redes de pequeno, intermediário e grande porte. O emprego de técnicas de fusão de dados com o algoritmo CT diminui consideravelmente a velocidade da entrega dos dados, sendo pouco ágil e escalável como foi comprovado pelos resultados exibidos na Figura 4.15.

Um dos maiores benefícios no uso de funções de fusão de dados é a redução do número de transmissões na rede. A redução do número de transmissões acarreta, por exemplo, em menor consumo energético, menor probabilidade de colisões na rede e atenuação do efeito overhearding. Nas Figuras 4.19, 4.20a, 4.20b e 4.20c mostram o compromisso e benefícios do uso das funções de fusão de dados.

O compromisso entre o número de transmissões e o *freshness* dos dados é analisado na Figura 4.19. Os algoritmos CRAL com o uso de fusão de dados reduzem o número de transmissões em detrimento do tempo de entrega das mensagens. As duas versões do protocolo apresentam sempre menor tempo de entrega dos dados quando não usam funções de fusão de dados. Quando se utiliza técnicas de fusão, é possível perceber que CRAL-FD em algumas simulações balanceia este compromisso citado, obtendo baixo tempo de entrega e número de transmissões. Já CRAL-LD, reduz ao máximo o número de transmissões e mantém um tempo de entrega moderado entre $4s$ e $7s$. CT apresenta número de transmissões e tempo superiores aos dos CRAL-LD e CRAL-FD em todos os

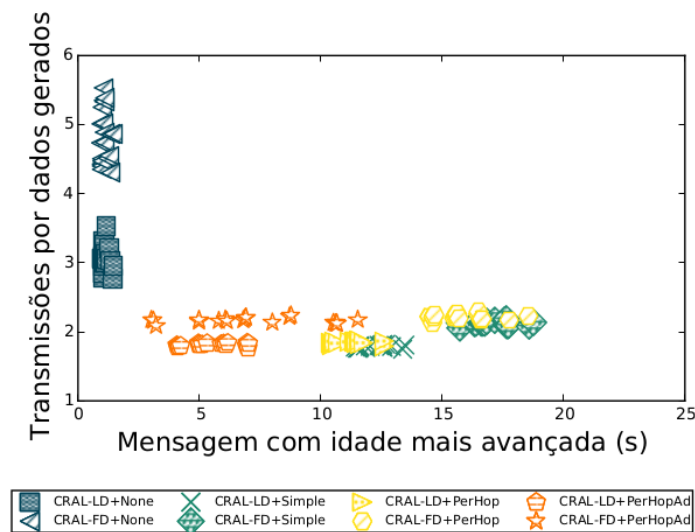


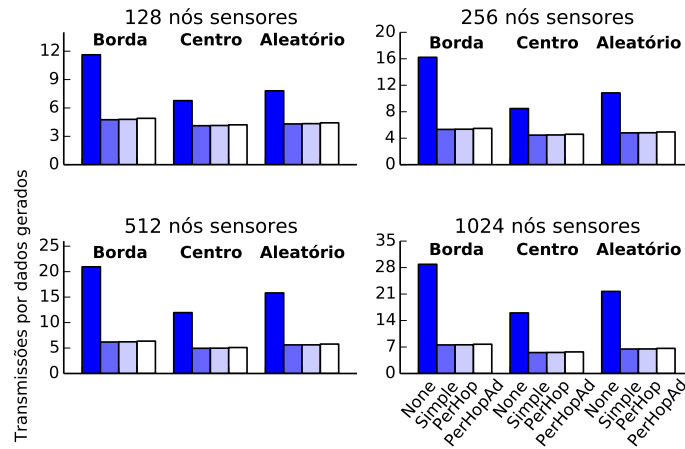
Figura 4.19: Compromisso entre número de transmissões e *freshness*.

cenários avaliados, por isso não são apresentados neste gráfico. A Figura 4.19 também evidencia a diferenças de *freshness* entre as funções de fusão de dados, por exemplo, *Simple* e *PerHop* apresentam valores, em geral, superiores aos apresentados pela técnica *PerHopAd*.

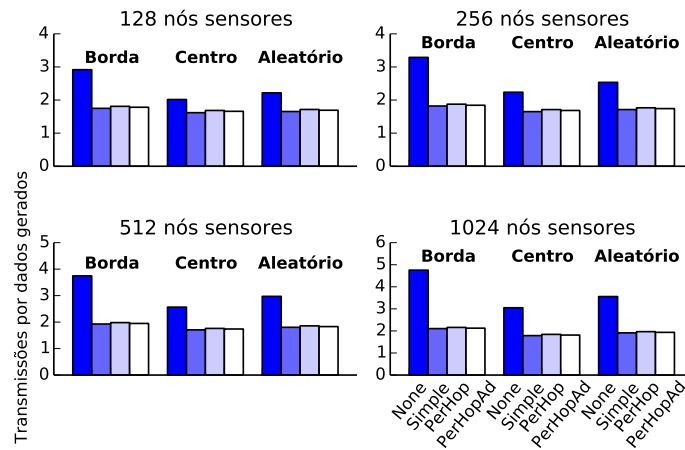
O comportamento do protocolo dos protocolos para diferentes funções de fusão de dados é exibido na Figura 4.20. Para este experimento, variou-se o número de transmissões por dados reportados para cada uma das técnicas de fusão bem como posicionamento da EB. CT em face às funções de fusão de dados apresentou os resultados exibidos na Figura 4.20a. É notório que o uso da fusão de dados reduz de modo drástico o número de transmissões para entregar os dados, é possível perceber que são poupadas mais que metade das transmissões em todos os cenários. Uma implicação direta desse resultado é a redução no consumo de energia.

As Figuras 4.20b e 4.20c exibem as mesmas informações da Figura 4.20a, porém para os algoritmos CRAL-LD e CRAL-FD respectivamente. Os algoritmos CRALs se mostram mais econômicos do que CT em termos de transmissões por dados gerados. É possível perceber que o número de transmissões dessas abordagens sem o uso de funções de fusão de dados são aproximadamente o número médio de saltos da rede. Isto mostra que as abordagens são eficientes mesmo sem o uso de fusão de dados quando comparado com CT, o qual apresenta valores médios entre 12 e 28 transmissões por dados gerados para os diferentes portes de rede, enquanto CRAL-LD e CRAL-FD não ultrapassam 8 transmissões por pacote de dados em qualquer cenário.

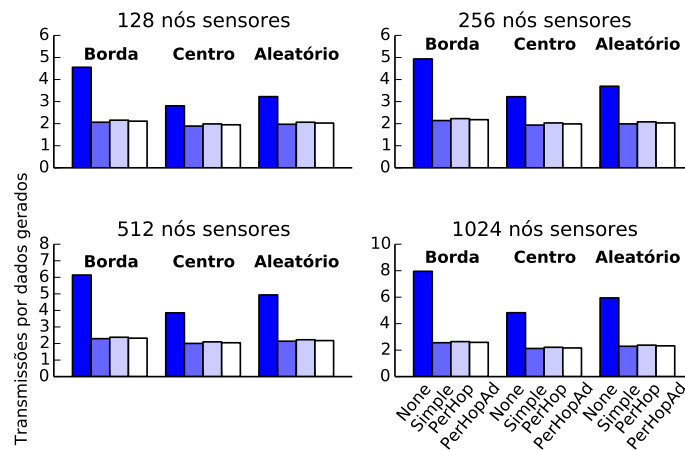
Quando são empregadas as funções de agregação em conjunto com o protocolo CRAL, o número de transmissões cai pela metade na maioria dos casos, chegando a



(a) Algoritmo CT.



(b) Algoritmo CRAL-LD.



(c) Algoritmo CRAL-FD.

Figura 4.20: Avaliação comportamento dos protocolos sobre o número de transmissões por dados gerados em face às funções de fusão de dados.

ser aproximadamente 3 vezes menor quando se utiliza a versão CRAL-FD em redes de grande porte e posicionado o sorvedouro na borda da rede. Os algoritmos CRAL, quando comparado com CT, mostram uma substancial redução de transmissões e consequentemente menor consumo de energia, tanto com fusão de dados, quanto sem o uso dessa técnica.

4.6 Conclusões do Capítulo

Este capítulo apresentou uma avaliação por simulação do protocolo proposto CRAL, um algoritmo que roteia mensagens baseado na importância topológica dos nós intermediários em RSSF. As avaliações realizadas serviram e validar o protocolo CRAL acerca do seu correto funcionamento, bem como mostrar que CRAL é de fato uma aproximação para a árvore de Steiner, que é a solução ótima para o problema considerado neste trabalho.

Foram apresentadas as premissas e requisitos necessários para a construção do protocolo CRAL, bem como discutidos seus principais pontos fortes e fracos. Além disso, apresentou-se as mais avançadas técnicas de fusão de dados em redes de sensores sem fio. Ao operar com diferentes funções de fusão de dados, mostra que a arquitetura do protocolo é de fato flexível, facilitando por exemplo manutenção e adaptação de recurso.

A análise de complexidade do algoritmo CRAL a respeito do custo de memória e mensagens de controle para estabelecimento da árvore de roteamento foram derivadas. Esta análise assintótica foi comparada com a complexidade dos algoritmos relacionados.

Diversos experimentos por simulação foram realizados e o desempenho do protocolo CRAL foi confrontado com os dos protocolos CT e SPT. Como resultados destaca-se que CRAL consome menor quantidade de energia sendo até 6 vezes mais econômico do que os concorrentes. Em relação à eficiência o protocolo proposto reduz drasticamente o número necessário de transmissões por dados reportados à EB. No que tange a comparação da confiabilidade dos protocolos, as abordagens CRAL-LD e CRAL-FD apresentaram respectivamente 99.6% e 99.1% ao passo que CT e SPT queda de confiabilidade quando o tamanho da rede cresce, alcançando somente 60% de entrega dos dados gerados em redes de grande porte.

CRAL-FD se mostrou mais rápido ao entregar os pacotes de dados à EB, seguido por CRAL-LD e CT. Isto se deve ao uso de LQE otimizados para encontrar rotas de alta vazão e baixa perda de pacotes.

O protocolo proposto em conjunto com técnicas de fusão de dados se mostrou

mais rápido na entrega de dados. Para avaliar este quesito foi utilizado o conceito de *freshness*. O CRAL e a função de fusão de dados PerHopAd se mostraram mais eficazes entregando dados atualizados ao passo que mantém baixo o número de transmissões, isto implicou em menor redução do consumo de energia. Além disso, o protocolo CRAL-LD entrega 90% das mensagens com menos de 2s, enquanto os protocolos concorrentes levam o triplo do tempo nas mesmas condições de cenário.

Baseado nos resultados apresentados, é possível concluir que CRAL é uma alternativa para coleta de dados em RSSF. Tal inferência é baseada nas características do protocolo sendo elas: confiabilidade, eficiência, robustez e flexibilidade, além de apresentar baixo consumo de energia e ser consciente das características inerentes do ambiente sem fio.

Capítulo 5

Protocolo de Roteamento XCTP (*eXtend Collection Tree Protocol*)

Este capítulo apresenta o protocolo eXtend Collection Tree Protocol, um protocolo de roteamento que é uma extensão do CTP. CTP é de fato um dos principais protocolos de roteamento para coleta de dados em RSSF. CTP cria uma árvore de roteamento para transferir dados de um ou mais nós sensores para a raiz da árvore (Estação Base). Porém, CTP não cria caminhos reversos entre a EB e os nós sensores da rede. Estes caminhos são importantes, por exemplo, para envio de pacotes de confirmação (*acknowledgment*) ou comandos de retroalimentação (*freedback*) fim-a-fim. O objetivo principal deste capítulo é o algoritmo XCTP que cria estes caminhos reversos, deste modo habilitando a comunicação em ambos os sentidos: EB para os nós e nós para EB.

XCTP cria estas rotas reversas explorando o plano de controle da arquitetura do CTP, necessitando de baixa quantidade de memória para armazenar estados e custo mínimo adicional aos pacotes de dados. Com esses caminhos reversos é possível implementar um protocolo de transporte confiável¹ para RSSF. Assim, também será apresentado o Transport Automatic Piggyback Protocol, um protocolo de transporte com ARQ sobre o protocolo XCTP. Adicionalmente serão apresentados os detalhes sobre a arquitetura e operação do protocolo XCTP para criar e manter as rotas reversas, bem como os detalhes da implementação do TAP2.

O protocolo XCTP foi implementado para o sistema operacional TinyOS. Uma Avaliação experimental do protocolo foi realizada sobre o TOSSIM e os resultados foram comparados com os mais avançados protocolos sendo eles: CTP, RPL e AODV.

¹Vale ressaltar que confiável para este trabalho é a viabilização de rotas bidirecionais que permitem, por exemplo, troca de mensagens de confirmação.

Experimentos de escalabilidade e stress foram realizados aplicando diferentes quantidades de nós sensores e cargas. A avaliação dos resultados mostrou que XCTP é mais confiável do que CTP, pois entrega 100% dos pacotes de dados. XCTP necessita enviar menos pacotes de controle que RPL. XCTP é mais rápido ao recuperar-se de falhas na rede, além de manter menor quantidade de estado que AODV.

O restante deste capítulo está organizado da seguinte forma: na próxima seção será definido o escopo do problema que o protocolo XCTP visa resolver. Em seguida na Seção 5.2 serão descritos os detalhes de arquitetura, modificações realizadas no protocolo CTP, bem como uma especificação da operação do protocolo XCTP, descrevendo seus casos de uso, seu funcionamento e interfaces de programação. Ainda da Seção 5.2 será apresentado o protocolo Transport Automatic Piggyback Protocol. Logo após, na Seção 5.3 é especificada a metodologia utilizada para avaliação dos protocolos XCTP, CTP, AODV e RPL, além de detalhes e comentários sobre os resultados obtidos dos experimentos computacionais. Finalmente, na Seção 5.4 serão apresentadas as considerações finais.

5.1 Escopo do Problema

RSSF são compostas por um grande número de nós com capacidades de monitoramento, computação e comunicação sem fio. Estas redes possuem restrições computacionais e de energia. Diversas aplicações em RSSF necessitam transportar grandes quantidades de dados (imagem ou vídeo por exemplo). Geralmente essas aplicações não são tolerantes a perda de dados, assim é importante prover mecanismos para uma coleta de dados confiável.

A comunicação em RSSF geralmente segue um dos paradigmas de comunicação: Muitos-para-Um, Um-para-Muitos e Ponto-a-Ponto (vide Capítulo 2 Seção 2.2). Os dois primeiros paradigmas permitem respectivamente a coleta e a disseminação de dados. Contudo, com o roteamento em somente uma direção torna-se impossível construir mecanismos que assegurem a entrega de dados fim-a-fim. O paradigma ponto-a-ponto permite a comunicação entre quaisquer dois pares de nós na rede, porém acrescenta complexidade ao protocolo e requer grandes quantidades de memória para armazenar todas as possíveis rotas, tornando esta abordagem inviável para redes de sensores com quantidades limitadas de memória.

A entrega de dados é um dos mais desafiadores problemas em RSSF, isto se deve a frequente instabilidade dos enlaces que em frações de segundo, muitas vezes menor que 1s, altera a topologia da rede Srinivasan et al. [2008]. Portanto, este trabalho se

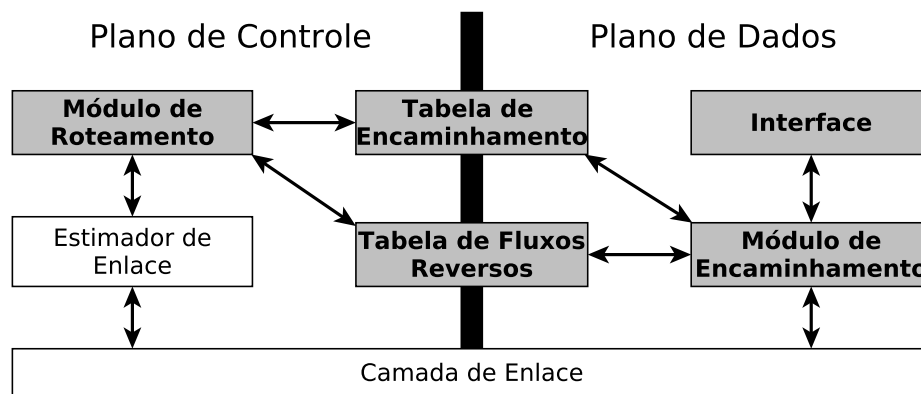


Figura 5.1: Arquitetura do protocolo XCTP.

propõe a enfrentar os seguintes problemas no roteamento em RSSF: i) confiabilidade, ii) robustez, iii) eficiência. Requisitos já definidos neste trabalho (Capítulo 2 Seção 2.1).

XCTP é uma alternativa para coletar dados de modo confiável, robusto e de modo eficiente. Este protocolo balanceia os compromissos impostos pelas as três metas fundamentais das RSSF. XCTP ajusta o modelo de comunicação de coleta de dados para fornecer rotas que permitam que comandos de retroalimentação, pacotes de confirmação ou mensagens de confirmação sejam trocados de modo bidirecional entre qualquer nó sensor e a EB. XCTP permite que protocolos de transporte de dados com confirmação seja construído sobre ele, habilitando desta forma a entrega de dados confiável.

5.2 O Protocolo *eXtend Collection Tree Protocol*

Nesta seção será apresentada a arquitetura do protocolo XCTP bem como as principais características do CTP que foram exploradas pelo XCTP para a construção das rotas reversas. Além disso serão destacadas as principais alterações realizadas tanto na arquitetura quanto nos pacotes de dados. Também serão mostrados como os fluxos reversos são instalados, bem como o comportamento que o algoritmo apresenta em face às mudanças topológicas e o algoritmo interno do CRAL que efetua o encaminhamento dos pacotes no sentido reverso.

5.2.1 Arquitetura

Foram realizadas modificações na arquitetura do CTP, para que seja possível realizar a tarefa de encaminhar pacotes também na direção inversa à do fluxo padrão dos dados a saber Muitos-para-Um. Foram adicionadas e alteradas algumas regras do protocolo

para, bem como incrementos nos pacotes de dados. Os pacotes foram modificados minimamente adicionando somente um campo. Também foi criada uma nova tabela de rotas para armazenar os fluxos reversos. As regras do protocolo foram modificadas tanto no nível de plano de dados e de controle. O plano dados é definido como a parte da arquitetura que decide o que fazer com os pacotes de dados, assim ele foi modificado para consultar a tabela de rotas reversas. O plano de controle decide como construir e modificar o mapa da rede, assim ele foi modificado para manipular a tabela de fluxos reversos e também reagir apropriadamente a dois eventos principais, sendo eles:

1. **Fluxos reversos:** correta e eficiente instalação das regras dos fluxos reversos;
2. **Mudanças topológicas:** o nó deve reagir apropriadamente quando ocorre *loops* ou quando as rotas *unicast* do CTP mudam.

Na Figura 5.1 é mostrada a arquitetura do protocolo XCTP com os principais módulos e em destaque cinza os módulos que sofreram alterações ou foram adicionados. O Módulo de Roteamento é responsável pelo preenchimento das tabelas de Encaminhamento e de Fluxos Reversos. Essas tabelas indicam quem é o próximo salto para que os pacotes de dados possam ser transmitidos. Não foram feitas alterações no módulo de Estimador de Enlace, o qual utiliza o LQE *Four bit* (veja Tabela 2.2 do Capítulo 3). *Four bit* é um LQE baseado em software que faz uso de 4 bits de informações de camada diferentes para estimar os enlaces, além disso utiliza monitoramento ativo e passivo, necessitando que mensagens de sonda sejam usadas. O Módulo de Encaminhamento efetua consultas nas tabelas de Encaminhamento e Fluxos Reversos, além disso determina quaisquer inconsistências de rota e informa ao Módulo de Roteamento. O Módulo de Encaminhamento também mantém uma fila para transmissão e efetua análises de pacotes duplicados na rede. Abaixo na Figura 5.1 é mostrada a camada de enlace que contém recursos e características para utilizar o rádio de comunicação. Finalmente, na camada de Interface o protocolo XCTP provê interfaces para que componentes sejam construídos e utilizem os recursos do protocolo.

5.2.2 Modificação no Pacote de Dados

Para permitir a navegação de pacotes na direção reversa, foi adicionado um campo de 16 bits ao pacote de dados para representar o *destinatário* do pacote. A Figura 5.2a mostra o novo formato do pacote de dados. Os campos do pacote são: *P* que permite ao nó requisitar informações de roteamento para outros nós; *C* indica uma notificação de congestionamento de rota; *Time Has Live (THL)* em que cada nó quando recebe

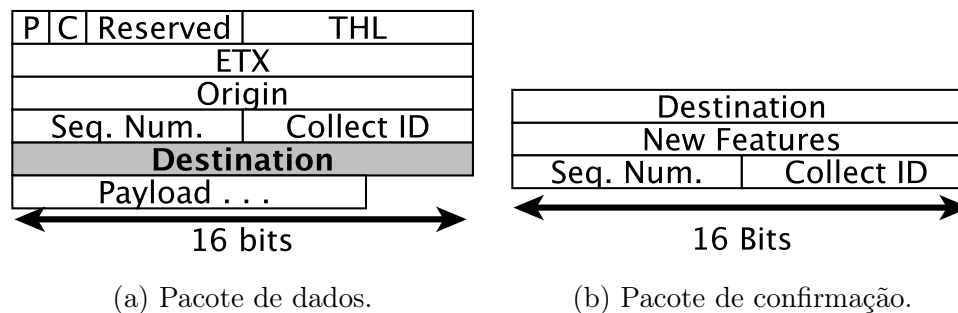


Figura 5.2: Formato dos pacotes para o protocolo XCTP.

um pacote incrementa este campo; campo para métrica *ETX* para estabelecimento das rotas e detecção de loops; Endereço *destination* para o endereço do nó de destino do pacote; endereço *origin* para o nó indicar o endereço do nó emissor do pacote; *collect ID* para identificar a árvore de coleta de dados.

Também foi criado um pacote de confirmação. Este pacote apresenta um subconjunto dos campos do pacote de dados, como ilustrado na Figura 5.2b. O campo de 16 bits *New Features* foi reservado para futuras melhorias. O pacote ACK é útil como uma mensagem de retroalimentação utilizados por protocolos da camada de transporte para implementar protocolos fim-a-fim sobre XCTP.

5.2.3 Fluxo Reverso

O plano de controle é responsável pela manipulação da tabela do XCTP para fluxos reversos. A Tabela de Fluxos Reversos tem os seguintes campos: *addr dest*: para endereços de descendentes na árvore de rotas do XCTP². *next hop*: para conter endereços de próximo salto para alcançar o destinatário do pacote; *ttl*: tempo de vida da rota na tabela, em que é possível implementar políticas para remoção dessas rotas.

O Módulo de Roteamento implementa as operações básicas de Criação, Leitura, Atualização e Remoção sobre a Tabela de Fluxos Reversos. A seguir serão descritas em detalhes cada uma das operações:

- *Criação*: no início a tabela encontra-se vazia. Quando um nó sensor encaminha uma mensagem para a EB, a rota reversa é instalada. Uma vez que o Módulo Estimador de Enlace armazena informações sobre os vizinhos diretos (1-salto), o Módulo de Roteamento não insere entradas na Tabela de Fluxos Reversos para vizinhos diretos. A Figura 5.3a ilustra essa situação: quando o nó C envia dados

²Vale ressaltar que foi realizada uma otimização em que este campo somente é preenchido para descendentes com mais de 1 salto de distância, já que os nós da vizinhança podem ser obtidos através da camada de enlace.

para a EB, o nó intermediário (que não está a 1 salto de C) intercepta o pacote oriundo do nó fonte C, o nó intermediário instala um fluxo reverso para C;

- *Leitura*: o Módulo de Roteamento fornece interfaces para que consultas sejam realizadas sobre a Tabela de Fluxos Reversos. Estas interfaces são utilizadas pelo Módulo de Encaminhamento para tomar decisões sobre os pacotes de dados. No item 5.2.5 desta seção são descritos os detalhes de uso dessa interface;
- *Atualização e Remoção*: o Módulo de Roteamento provê mecanismos para atualizar e remover entradas na Tabela de Fluxos Reversos. Essas funções são chamadas sempre que existe uma mudança na topologia da rede. Na Seção 5.2.4 são descritos os detalhes de atualização e remoção de entradas.

Existe um compromisso entre agilidade e eficiência considerando a manutenção de rotas em L2Ns. Agilidade aqui refere-se a quão rápido a rede reage a mudanças topológicas, enquanto eficiência é a energia consumida e o número de mensagens enviadas para manter a rede operacional. A rede requer alta frequência de envio de mensagens de sonda para manter as rotas atualizadas. Isto aumenta a agilidade da rede, por outro lado, reduz a eficiência. XCTP usa o algoritmo Trickle Levis et al. [2003b] para aumentar o número de mensagens de sonda quando a rede é instável e reduz exponencialmente o número de sondas quando a rede é estável. Assim mantém-se o compromisso balanceado entre agilidade e eficiência. XCTP faz uso dos pacotes de dados para construir rotas reversas, assim, não existe custo adicional para preencher a Tabela de Fluxos Reversos.

5.2.4 Reação a Mudanças na Topologia

Um sistema de roteamento deve saber quando e onde alterar suas entradas na tabela para refletir corretamente às mudanças na dinâmica topológica da rede. Estas mudanças fazem com que o plano de dados decida corretamente os caminhos pelo quais os pacotes devem trafegar. O plano de controle do XCTP age e altera a Tabela de Fluxos Reversos quando existem ocorrência de loops e falhas nos enlaces.

Para manter a consistência das rotas, cada nó sensor mantém o custo estimado da rota para a EB. Além disso, esta informação é anexada aos pacotes de dados e de controle (vide Figura 5.2a). XCTP usa o ETX como métrica para calcular o custo da rota. Este custo é sempre crescente na direção dos nós folhas da árvore de roteamento e esta invariante deve ser sempre mantida. Loops são detectados quando este invariante é violado. Neste caso, a Tabela de Fluxos Reversos deve ser removida.

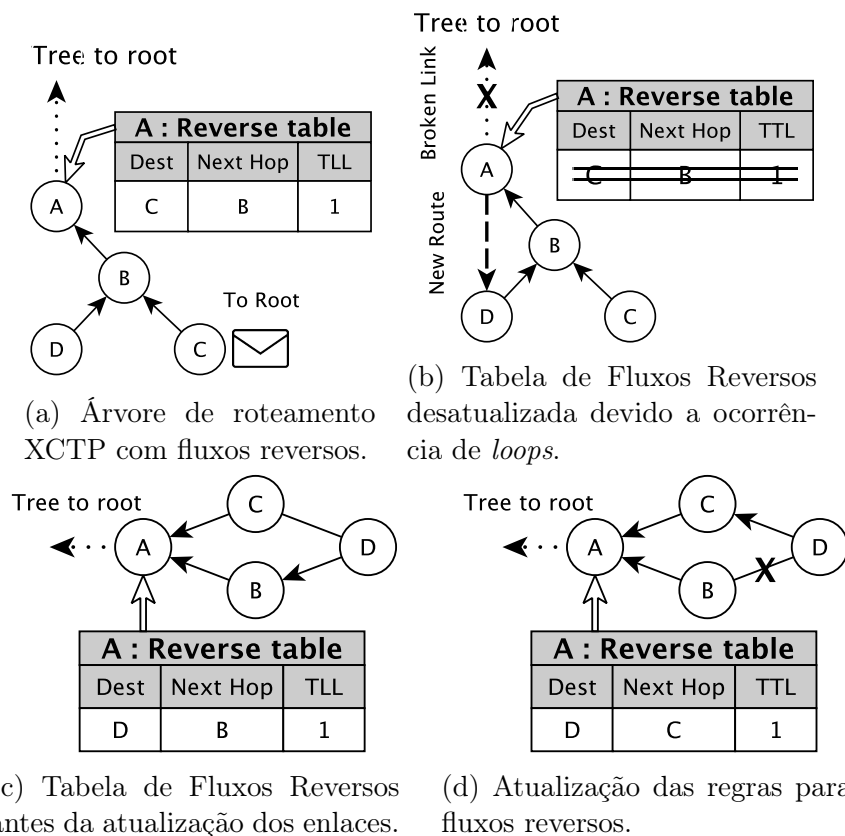


Figura 5.3: Reação do plano de controle sobre o plano de dados, quando ocorre detecção de *loop* e atualização de caminhos.

As Figuras 5.3a e 5.3b ilustram essa situação. Na Figura 5.3a, é mostrado o estado inicial da Tabela de Fluxos Reversos. Então, como mostrado na Figura 5.3b, existe uma falha no enlace que tem por consequência a formação de um loop entre os nós A, B e D. Quando acontece um loop, o plano de dados pode ler da Tabela de Fluxos Reversos, nós que eram descendentes e que agora estão em um nível mais alto na árvore de roteamento. Portanto, a ação tomada quando loops são detectados pelo plano de dados do XCTP é sinalizar ao plano de controle que houve uma detecção de loop. Assim, o plano de controle removerá apropriadamente as entradas da tabela. Novos fluxos reversos são reconstruídos quando existem novos pacotes de dados são enviados pela rede.

No caso de troca de enlaces devido à dinâmica da qualidade dos enlaces, o plano de controle deve atualizar a Tabela de Fluxos Reversos para que o plano de dados encaminhe pacotes pelos novos caminhos que refletem a nova configuração da árvore de roteamento. A Tabela de Fluxos Reversos é atualizada quando um pacote de dados de um fluxo já instalado é interceptado, mas este pacote foi roteado através de um

diferente vizinho. A Figura 5.3c e 5.3d ilustram esse caso. Os pacotes de dados do nó D enviados para a EB eram encaminhados pelo nó B e as mudanças nas qualidades dos enlaces fizeram com que os dados fossem encaminhados agora pelo nó C. Assim, o plano de dados do nó A deve ser atualizado para refletir essa nova configuração: o fluxo reverso deve ser encaminhado agora pelo nó C (Figura 5.3d).

5.2.5 Interfaces do XCTP

Aqui será descrita a Application Programming Interface (API) do protocolo XCTP. O protocolo CTP não requer um endereço de destinatário para as mensagens, já que todas são destinadas à EB. XCTP por outro lado, necessita que um endereço de destino para prover roteamento *unicast* para um específico nó sensor da rede. XCTP integra uma interface que inclui o endereço de destinatário, bem como rotinas para manipulação para as Tabelas de Encaminhamento e de Fluxos Reversos. Estas rotinas são:

- **addr sendTo(target, pkt):** onde *target* é o endereço de destino do pacote de dados do XCTP;
- **addr nextHop(target):** onde *target* é um parâmetro opcional. Se *target* for especificado, a rotina **nextHop(target)** consulta a tabela de fluxos reversos, em caso contrário a mensagem será enviada diretamente para a EB.
- **loopDetect():** esta rotina sinaliza ao plano de controle quando um loop é detectado (vide Seção 5.2.4).
- **snoopNewPkt(pkt):** quando um pacote de dados é interceptado de um novo fluxo reverso, o plano de controle deve ser sinalizado para atualizar as tabelas de Encaminhamento e Fluxos Reversos.

Assim, a interface **sendTo(target, pkt)** deve ser utilizada quando a EB necessita enviar um pacote para um específico nó sensor da rede.

O Algoritmo 6 descreve a rotina da interface **sendTo(target, pkt)**. Na linha 2 é verificado se o pacote recebido é de dados ou de confirmação, pois somente esses dois tipos de pacote devem trafegar pelos caminhos reversos. Após essa verificação, o endereço de destinatário é extraído do pacote. Em caso do destinatário ser o próprio nó (linha 3), o pacote alcançou seu destino e deve ser processado adequadamente. Caso o endereço de destino seja um dos nós descendentes (linha 5), o pacote é encaminhado. Em caso contrário, ou seja, o destinatário não está presente em nenhuma das tabelas existem duas abordagens a se fazer: descartar o pacote ou encaminhá-lo à EB (linha 9).

Algoritmo 6: Operação interna da interface `sendTo(...)`

```

[1] Procedimento sendTo(target, pkt)
[2]   se isDataXCTP(pkt) ∨ isAckXCTP(pkt) então
[3]     se pkt.destination = my.addr então
[4]       | // Processa o pacote localmente.
[5]     senão se pkt.nextHop = nexHop(pkt.destination) então
[6]       | // Envia unicast para endereço na Tabela de Fluxos
[7]       |     Reversos
[8]     fim
[9]     senão
[10]    | // Descarta pacote ou encaminha para a estação base
[11]    fim
[12]    retorna
[13]  senão
[14]    | // Encaminha pacote normalmente pela árvore XCTP
[15]    | pkt.nextHop = nextHop()
[16]    | encaminha(pkt)
[17]    retorna
[18]  fim

```

No segundo caso, uma vez que a EB conhece toda a topologia da rede, a EB pode encaminhar o pacote pela rota *unicast* apropriada ou simplesmente descartar a mensagem. No caso do pacote não apresentar endereço de destinatário válido, o nó com XCTP encaminha as mensagens diretamente para a EB (linhas 13–15).

O protocolo XCTP permite comunicação sobre o paradigma ponto-a-ponto. Isto é possível devido ao modo como as rotas reversas são construídas e se a decisão tomada na linha 9 do Algoritmo 6 seja a de encaminhar para a EB. Se um nó X quer se conectar com um nó Y, o qual não está diretamente ligado, X pode utilizar a rotina *sendTo(Y, pkt)*. O nó Y receberá os pacotes através de um ancestral de X ou, no pior caso, os pacotes irão até a EB e depois percorrerão a rota *unicast* até o nó Y.

5.2.6 O Protocolo *Transport Automatic Piggyback Protocol*

Usado a API do provida pelo protocolo XCTP foi implementado um protocolo de transporte confiável, o qual é chamado de Transport Automatic Piggyback Protocol (TAP2). TAP2 usa um mecanismo de *piggyback* e Automatic Repeat Request para controle de erros e retransmissões de pacotes. Diversos protocolos de transporte para RSSF foram propostos na literatura [Paek & Govindan, 2007; Kim et al., 2007; Wang et al., 2008] que também podem ser implementados sobre o XCTP. Contudo, os

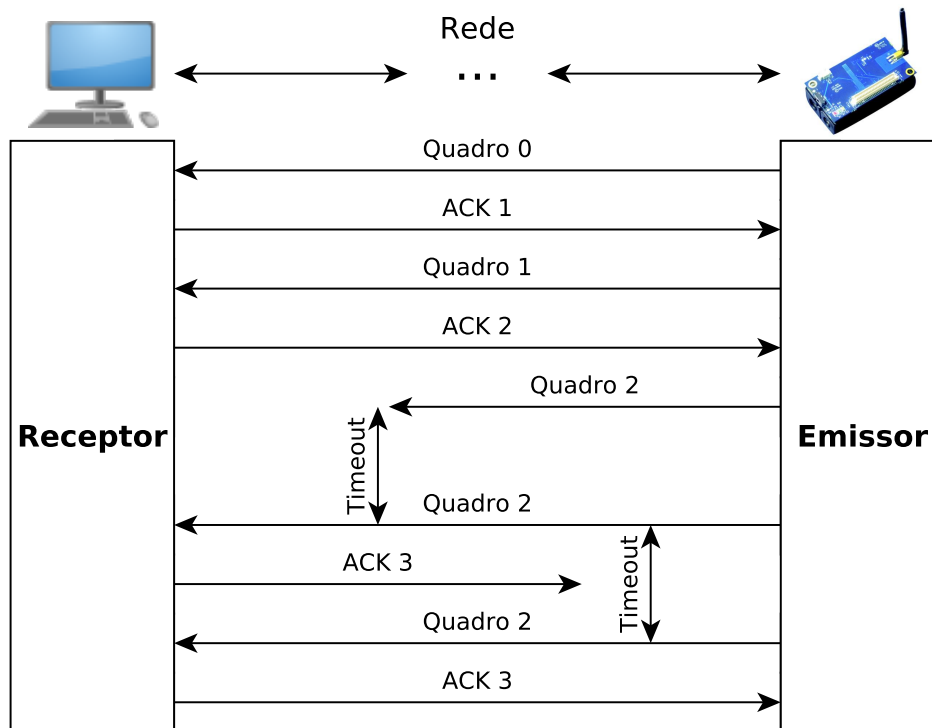


Figura 5.4: Operação do protocolo Transport Automatic Piggyback Protocol.

pré-requisitos de poucos recursos computacionais e implementação simplista do TAP2 foram as razões da escolha de seu uso neste trabalho.

A Figura 5.4 apresenta os mecanismos de operação do protocolo de transporte TAP2. O emissor envia mensagens com número de sequência, o receptor por sua vez ao receber um quadro envia uma mensagem ACK contendo o número de sequência do próximo quadro que deseja receber. Caso um quadro ou ACK seja perdido durante a transmissão dos dados um *timeout* instalado no lado do emissor irá se esgotar e então a última mensagem solicitada será transmitida novamente. Percebe-se na Figura 5.4 os dois momentos em que um quadro é reenviado devido a término do *timeout*. Deste modo, TAP2 entrega os dados com garantia para estação base³.

5.3 Avaliação

Esta seção apresenta a avaliação experimental do protocolo XCTP. Também será apresentada a metodologia utilizada para validar e avaliar o protocolo. A divisão desta seção se dá em duas partes: Metodologia e Resultados. Na seção de metodologia são descritas todas as ferramentas utilizadas, bem como os cenários de simulação para realizar

³Note que TAP2 também pode operar no sentido reverso, isto é, a estação base enviando dados de modo seguro para os nós sensores.

os experimentos. Na segunda parte da seção, os resultados obtidos dos experimentos são exibidos e comentados em detalhes.

5.3.1 Metodologia

Aqui será descrita a metodologia utilizada para analisar o protocolo XCTP. Dos protocolos apresentados na Tabela 3.2, o protocolo Direct Diffusion Intanagonwiwat et al. [2003], Deluge Chlipala et al. [2004], (DIP, DRIP, DHV) Levis et al. [2005] são usados apenas para a disseminação de dados e não servem para comparação com XCTP. CTP Gnawali et al. [2009] é um dos mais novos protocolos para coleta de dados em RSSF e apresentam melhores resultados que MultiHopLQI Philip Buonadonna [2007] e MintRoute Woo et al. [2003b]. No melhor dos nossos conhecimentos, não existem implementações estáveis e abertas à comunidade dos protocolos Dymo C. Perkins [2013], DSR Johnson et al. [2007] e Hydro Dawson-Haggerty et al. [2010]. Portanto as comparações aqui apresentadas utilizam os protocolos RPL, AODV e CTP.

XCTP será comparado com três outros protocolos da literatura: CTP, RPL e AODV. O principal objetivo desta análise é mostrar que o protocolo XCTP está operando apropriadamente, bem como avaliar o desempenho do XCTP quando comparado com os atuais protocolos de roteamento em RSSF.

XCTP, CTP e AODV foram implementados sobre a linguagem de programação para nesC para o sistema operacional para RSSF TinyOS Levis et al. [2005]. Foi adotada a implementação RPL Contiki Dunkels et al. [2004], pois a versão do protocolo RPL para TinyOS não se encontra estável. Também foram realizados experimentos com Tymo, uma versão do protocolo Dymo para TinyOS. Contudo, Tymo não se mostrou estável como reportando em TinyOS [2008].

Todas as execuções dos experimentos foram realizadas sobre o simulador para L2Ns TOSSIM Levis et al. [2003a]. Nestas simulações, foi considerado que a EB é um PC sem restrições de memória e que pode manter informações sobre toda a topologia da rede. Também foi utilizada a ferramenta LinkLayerModel Zuniga [2015] do próprio TinyOS para gerar os modelos de topologia e conectividade. Os parâmetros não triviais possuem o seguinte significado Jain [2007]: i) Expoente Path Loss diz respeito à taxa de decaimento do sinal, ii) Shadowing Std Dev (dB) refere-se a partes do sinal que são perdidas por absorções, reflexões e difrações pelo fato da existência de objetos ao longo do caminho do sinal, iii) Taxa de Decaimento (db) é a decaimento do sinal em dB a uma referida distância do emissor do sinal. Os valores escolhidos para estes parâmetros emulam um cenário de campo de futebol Zuniga [2015]

Foram consideradas 10 topologias diferentes, em cada cenário foram executadas

Tabela 5.1: Parâmetros padrão de simulação

Parâmetro	Valor
Estação Base	1 centro
Número de nós sensores	100
Alcance do Rádio (m)	100
Densidade da rede ($nós/m^2$)	10
Número de Experimentos	100
Bytes transmitidos	1024
Expoente Path Loss	4.7
Taxa de Decaimento (dB)	55.4
Shadowing Std Dev (dB)	3.2

10 simulações, totalizando 10 experimentos. Nos gráficos apresentados na Seção de resultados, as curvas representam a média e as barras de erro representam o intervalo de confiança de 95%. A Tabela 5.1 apresenta os parâmetros utilizados por padrão nos cenários de simulação.

Para quantificar e qualificar os experimentos com o intuito de avaliar e validar os protocolos, as seguintes métricas foram utilizadas:

1. *Construção de protocolos de transporte*: em que será analisado se o protocolo de roteamento favorece a implementação de protocolos de transporte de dados fim-a-fim confiáveis;
2. *Robustez*: serão provocadas falhas na rede e então se observa o comportamento dos protocolos, bem como o correto funcionamento do protocolo em diferentes topologias. Desta forma, avalia-se a robustez de operação do protocolo;
3. *Escalabilidade*: neste quesito será analisado a escalabilidade dos protocolos no que tange o consumo de memória dos nós sensores para armazenar rotas;
4. *Custo adicional de mensagens*: neste item será verificado o custo adicional gerado pelos protocolos para criar e manter as rotas;
5. *Consumo de memória RAM e ROM*: neste quesito é avaliado o espaço de memória requerido pelos protocolos de roteamento para sua operação.

5.3.2 Resultados de Simulação

1. *Construção de protocolos de transporte*: inicialmente serão mostrados os resultados dos protocolos objetos de análise a respeito do favorecimento a implementação

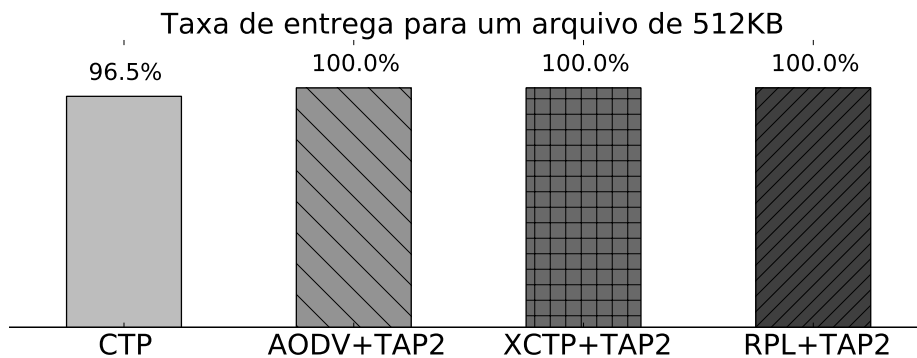


Figura 5.5: Transferência de um arquivo de 512KB de um nó a 5 saltos de distância da EB.

de protocolos de transporte confiáveis fim-a-fim. A Figura 5.5 mostra a porcentagem de entrega de um arquivo de 512KB de tamanho enviado de um nó a 5 saltos de distância da estação base. Os protocolos XCTP+TAP2, AODV+TAP2 e RPL+TAP2 alcançam 100% de entrega do arquivo, porque estes protocolos permitem que mensagens de confirmação sejam enviadas, para cada parte recebida do arquivo, entre os dois envolvidos na comunicação. É notório que CTP consegue transferir a maior parte dos dados, a saber 96.5% dos 512KB, porém não atinge os 100%, porque não é possível requisitar dados que foram perdidos ou confirmar partes já recebidas do arquivo. A impossibilidade de requisitar os fragmentos remanescentes do arquivo resulta em mau funcionamento de aplicações que são intolerantes a perda de dados. Conclui-se então que XCTP, AODV e RPL favorecem ao desenvolvimento de protocolos de transporte, provendo rotas que permitem que mensagens de retroalimentação sejam trocadas entre os nós sensores e a estação base de modo bidirecional;

2. *Robustez*: para avaliar a robustez foram elaborados experimentos com diferentes quantidades de fluxos ativos (nós sensores transmitindo dados para a EB) e experimentos que inserem falhas na rede:
 - a) Os protocolos XCTP e AODV foram comparados a respeito das suas reações quando acontecem falhas na rede. O seguinte cenário foi empregado neste experimento: 5 dos 100 nós sensores transferem, cada um, 1KB de dados de modo confiável para a EB, após 10s do início da transmissão dos dados 25% dos nós da rede foram desligados sem causar componentes desconectados na rede. Este desligamento, faz com que o protocolo crie novas rotas para a EB e reajam adequadamente a essa nova configuração da rede. A Figura 5.6 mostra no eixo Y a porcentagem de fluxos que terminaram a transferência

de 1KB de dados, já o eixo X mostra o tempo percorrido. Nos primeiros segundos de simulação as duas abordagens são similares, XCTP inicia pouco mais rápido devido a suas rotas serem construídas pró-ativamente, enquanto AODV apresenta construções de rota sob demanda, ou seja, de modo reativo a necessidade dos nós. Depois de desligar parte da rede no tempo de 10s, XCTP reage rapidamente ao encontrar novas rotas para a EB, os 5 nós sensores operando com XCTP completam a transferência em aproximadamente 25s. O protocolo AODV reage lentamente a mudanças topológicas. AODV em média leva 60s para completar a transferência de todos os dados, em alguns cenários, AODV supera os 200s para terminar as transferências;

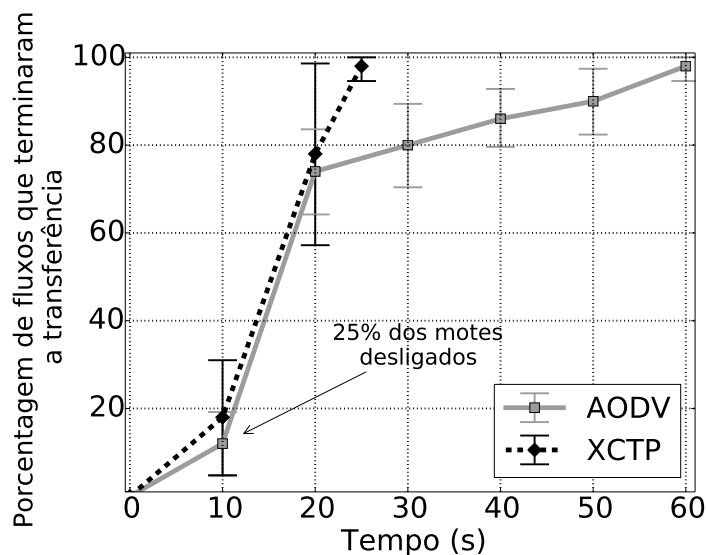


Figura 5.6: Reação dos protocolos XCTP e AODV na ocorrência de falhas na rede.

- b) No experimento #2 apresenta um cenário com 50 fluxos ativos com a EB. Após 10s de simulação 25% dos nós da rede são desligados. A Figura 5.7 mostra o comportamento do protocolo XCTP com e sem a presença de falhas na rede, na figura são mostradas as porcentagens de nós sensores que concluíram a transferência dos dados pelo tempo. XCTP, mesmo após o desligamento parcial da rede, rapidamente reconstrói as rotas entre reversas e para a EB e continua a transferir os dados. Em média com 2 minutos de simulação, todos os nós terminam a transferência. XCTP apresenta pouca diferença de comportamento na presença ou não de falhas na rede. Isto mostra que XCTP é ágil mesmo na presença de falhas e com muitos fluxos ativos concorrentes na rede. AODV não é mostrado nesta figura, pois ele

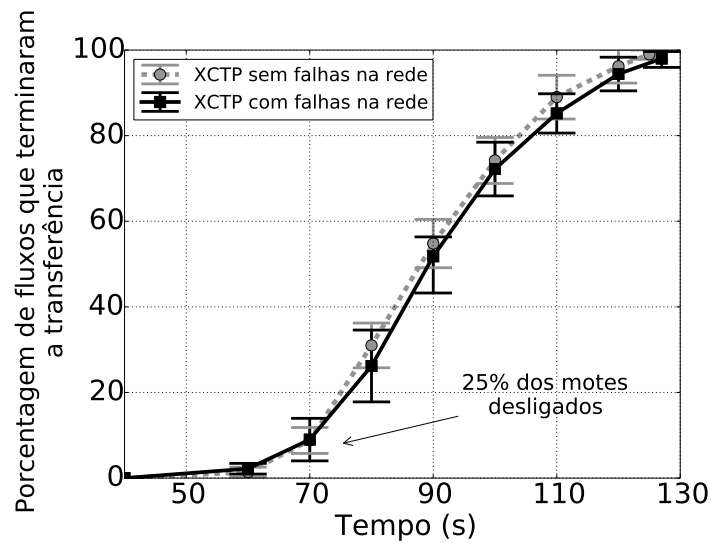


Figura 5.7: Reação do protocolo XCTP na presença de falhas na rede. 50 fluxos estão ativos.

não opera com mais de 5 fluxos concorrentes, o próximo quesito explica esse fato;

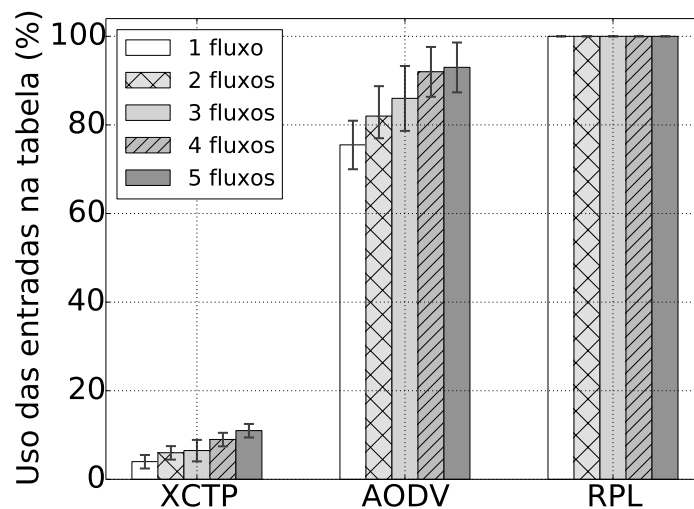


Figura 5.8: Consumo de memória da tabela de roteamento pelo número de fluxos para os protocolos XCTP, RPL e AODV.

3. *Escalabilidade*: para mostrar que XCTP é escalável, foram comparados os tamanhos da tabela de roteamento dos protocolos XCTP, AODV e RPL. Não foram feitas comparações com CTP, pois sua tabela tem tamanho constante e armazena somente o próximo salto em direção a EB. A Figura 5.8 mostra a comparação entre os protocolos no uso da tabela de roteamento variando o número de fluxos

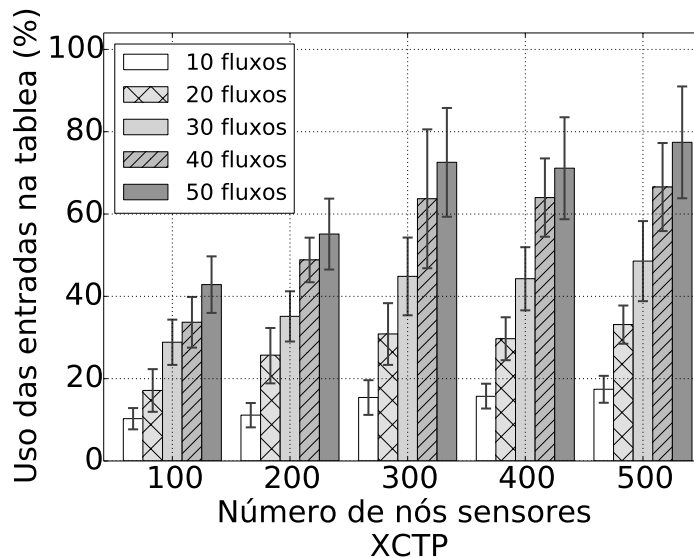


Figura 5.9: Tabela de Fluxos Reversos do protocolo variando o número de fluxos ativos e nós sensores na rede.

ativos. Fica evidente que com 5 fluxos ativos o protocolo AODV consome 100% da tabela de rotas. Quando muitos fluxos coexistem e a tabela de rotas encontra-se cheia, AODV é obrigado a rejeitar requisições para novas rotas. Assim, AODV necessita esperar que os temporizadores de rotas antigas expirem para então instalar novas rotas, este procedimento resulta em alto tempo de reação a falhas na rede e impede grande número de fluxos concorrentes. Ao contrário do AODV, XCTP consome aproximadamente 82% a menos de entradas na tabela de rotas que AODV para a mesma quantidade de fluxos. RPL sempre instala todas as possíveis rotas reversas, independentemente da demanda. Alguns nós intermediários não são capazes de armazenar todas as rotas para os nós em níveis mais baixos da árvore de roteamento, isto causa desconexões para algumas rotas. Ao contrário do protocolo RPL, XCTP mitiga esse problema com políticas baseadas em TTL em rotas subutilizadas e somente armazena rotas reversas sob demanda; A Figura 5.9 mostra que XCTP é robusto e escalável. XCTP sempre sob o limite da tabela de rotas reversas para diferentes quantidades de fluxos concorrentes, com diferentes topologias e quantidades de nós sensores na rede.

4. *Custo adicional de mensagens*: a Figura 5.10 apresenta o tráfego de mensagens de controle sobre um experimento de 5 horas de duração. Os protocolos XCTP e RPL apresentam alto número de mensagens de controle quando a rede é iniciada, mas eles decrescem e estabilizam sobre o tempo. XCTP envia menor quantidade de mensagens de controle que RPL, porque XCTP não necessita enviar mensagens

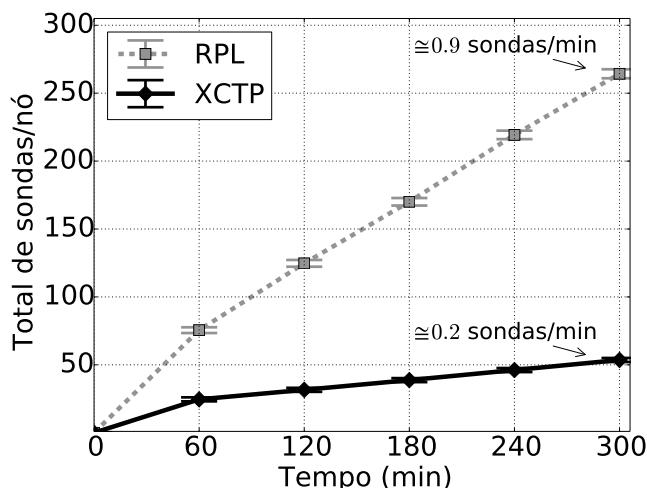


Figura 5.10: XCTP requer menor quantidade de mensagens de controle que RPL.

de sonda adicionais para construir as rotas reversas, enquanto RPL sim;

5. *Consumo de memória RAM e ROM*: a Tabela 5.2 apresenta o consumo de memória RAM e ROM dos componentes da pilha de protocolos avaliados em com e sem o protocolo de transporte TAP2. XCTP adiciona pouco mais que 1KB de código ao protocolo CTP, estes protocolos requerem menor quantidade de RAM quando comparados com AODV. O protocolo AODV entre todos é o que consome menor quantidade de ROM, ao passo que o RPL é o que mais consome RAM e ROM. No que tange os protocolos em conjunto com TAP2, XCTP apresenta menor consumo de RAM que AODV.

Tabela 5.2: Consumo de memória RAM e ROM em Bytes.

	CTP	RPL	XCTP	XCTP+TAP2	AODV	AODV+TAP2
RAM	1505	6516	1812	1968	2119	2545
ROM	16204	46454	17942	18435	13868	14562

5.4 Conclusões do Capítulo

Neste capítulo foi apresentado o eXtend Collection Tree Protocol um protocolo para coleta de dados em RSSF confiável, robusto, escalável e eficiente. XCTP se propõe a resolver o problema de rotas que permitam troca de informações bidirecionais entre as EBs e os nós sensores sendo aqui classificado como coleta de dados confiável. XCTP cria essas rotas explorando as características de um dos protocolos de roteamento mais avançados em coleta de dados em RSSF denominado CTP. XCTP como uma extensão

Capítulo 6

Considerações Finais

Nesta dissertação, duas variações de protocolos de roteamento para Redes de Sensores Sem Fio foram estudadas. O primeiro protocolo considerado, intitulado Centrality-based Routing Aware for L2Ns, combina duas características principais (centralidade, estimador de qualidade de enlace) para efetuar fusão de dados. O segundo problema dedicou-se a mitigar a questão de confiabilidade na coleta de dados, a este protocolo deu-se o nome eXtend Collection Tree Protocol.

Para o protocolo CRAL, um algoritmo distribuído que calcula a centralidade Sink Betweenness para cada nó sensor e ajusta as rotas para atravessarem nós intermediários comuns foi implementado. Deste modo, pode-se realizar a fusão de dados eficiente. Para justificar e avaliar a implementação deste algoritmo, experimentos foram realizados sobre os protocolos CRAL, CT e SPT. Estes experimentos mostraram resultados favoráveis ao protocolo CRAL em comparação aos demais protocolos, sugerindo que CRAL é de fato um protocolo que deve ser considerado ao se realizar coleta de dados em RSSF de modo eficiente nos quesitos de consumo de energia, vazão e confiabilidade.

Já o protocolo XCTP constrói rotas bidirecionais entre a estação base e os nós sensores da rede para permitir que mensagens de retroalimentação, controle ou dados trafeguem. Um estudo comparativo do XCTP foi feito com os protocolos mais atuais. Diante dessas comparações, XCTP mostrou-se favorável a construção de protocolos de transporte confiável, mais robusto que AODV tanto na presença de falhas, quanto no uso da tabela de rotas. Além disso, XCTP apresenta uma abordagem que é mais escalável ao passo que mantém baixo o custo do envio de sondas quando comparado com RPL e AODV. Tal como para CRAL, os resultados das comparações efetuadas com XCTP sugerem que o protocolo é de fato uma alternativa para coleta de dados Confiável em RSSF.

Em suma, esta dissertação apresenta as bases para a construção de dois protocolos

de roteamento para coleta de dados em RSSF. Estes protocolos foram submetidos a um conjunto de experimentos, os quais mostram que os protocolos aqui propostos são de fato soluções que apresentam resultados competitivos em relação aos já conhecidos e consolidados protocolos da literatura. A seguir, são listadas as principais contribuições deste trabalho:

- Proposição do algoritmo Centrality-based Routing Aware for L2Ns e uma avaliação experimental sobre o desempenho do protocolo,
- Proposição do algoritmo XCTP para RSSF e uma avaliação experimental sobre o desempenho do protocolo.

6.1 Trabalhos Futuros

Como trabalhos futuros, existem dois caminhos a serem trilhados com os protocolos aqui apresentados. Para o protocolo CRAL, pretende-se refiná-lo e avaliá-lo através de *testbed* reais, para então ter uma validação empírica do protocolo. De modo concorrente, planeja-se a construção de alternativas que visem mitigar os problemas causados pelo protocolo CRAL (vide Seção 4.2.4 do Capítulo 4), como o de drenar a energia dos nós centrais mais rapidamente. Além disso, explorar questões de densidade, localização e integração com a camada MAC também são objetos de estudo futuro do protocolo CRAL.

No que se refere a trabalhos futuros sobre o XCTP, pretende-se realizar experimentos em *testbed* reais tais como os disponibilizados em [Doddavenkatappa et al., 2012; Fambon et al., 2014]. Com base nestes experimentos, será possível validar o protocolo XCTP e refiná-lo. Outra tarefa a ser realizada compreende a extensão do protocolo TAP2 para operar em modo de janela deslizante, isto viabilizaria o aumento da vazão entre o nó sensor e a estação base. Além disso, pretende-se disponibilizar o XCTP como um projeto de software livre sob a licença GPL.

Referências Bibliográficas

- Aguayo, D.; Bicket, J.; Biswas, S.; Judd, G. & Morris, R. (2004). Link-level measurements from an 802.11 b mesh network. Em *ACM SIGCOMM Computer Communication Review*, volume 34, pp. 121--132. ACM.
- Awerbuch, B.; Holmer, D. & Rubens, H. (2004). High throughput route selection in multi-rate ad hoc wireless networks. Em *Wireless on-demand network systems*, pp. 253--270. Springer.
- Baccour, N.; Koubaa, A.; Jamaa, M.; Youssef, H. & Zuniga, M. (2009). A comparative simulation study of link quality estimators in wsn. Em *MASCOTS '09*. IEEE.
- Baccour, N.; Koubâa, A.; Mottola, L.; Zúñiga, M. A.; Youssef, H.; Boano, C. A. & Alves, M. (2012). Radio link quality estimation in wireless sensor networks: A survey. *ACM Trans. Sen. Netw.*
- Baccour, N.; Koubâa, A.; Youssef, H.; Jamâa, M. B.; Do Rosário, D.; Alves, M. & Becker, L. B. (2010). F-lqe: A fuzzy link quality estimator for wireless sensor networks. Em *Wireless Sensor Networks*, pp. 240--255. Springer.
- Balasubramanian, N.; Balasubramanian, A. & Venkataramani, A. (2010). Energy consumption in mobile phones: a measurement study and implications for network applications. Em *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pp. 280--293. ACM.
- Basu, P. & Redi, J. (2004). Effect of overhearing transmissions on energy efficiency in dense sensor networks. Em *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pp. 196--204. ACM.
- Borgatti, S. P. & Everett, M. G. (2006). A graph-theoretic perspective on centrality. *Social networks*, 28(4):466--484.

- Boukerche, A.; Loureiro, A. A.; Nakamura, E. F.; Oliveira, H. A.; Ramos, H. S. & Villas, L. A. (2014). Cloud-assisted computing for event-driven mobile services. *Mobile Networks and Applications*, 19(2):161--170.
- Brandes, U. (2008). On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, 30(2):136--145.
- C. Perkins, S. Ratliff, J. D. (2013). Dynamic manet on-demand (aodvv2) routing draft-ietf-manet-dymo-26. <http://tools.ietf.org/html/draft-ietf-manet-dymo-26>.
- Camp, T.; Boleng, J. & Davies, V. (2002). A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5):483--502.
- Cerpa, A.; Wong, J. L.; Kuang, L.; Potkonjak, M. & Estrin, D. (2005a). Statistical model of lossy links in wireless sensor networks. Em *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pp. 81--88. IEEE.
- Cerpa, A.; Wong, J. L.; Potkonjak, M. & Estrin, D. (2005b). Temporal properties of low power wireless links: modeling and implications on multi-hop routing. Em *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pp. 414--425. ACM.
- Chen, P.-Y.; Cheng, S.-M. & Chen, K.-C. (2014). Information fusion to defend intentional attack in internet of things. *Internet of Things Journal, IEEE*, 1(4):337--348. ISSN 2327-4662.
- Chen, Y.; Shu, J.; Zhang, S.; Liu, L. & Sun, L. (2009). Data fusion in wireless sensor networks. Em *Electronic Commerce and Security, 2009. ISECS '09. Second International Symposium on*, volume 2, pp. 504--509.
- Chlipala, A.; Hui, J. & Tolle, G. (2004). Deluge: data dissemination for network reprogramming at scale. *University of California*.
- Clark, B. N.; Colbourn, C. J. & Johnson, D. S. (1991). Unit disk graphs. *Annals of Discrete Mathematics*, 48:165--177.
- Couto, D. S. J.; Aguayo, D.; Bicket, J. & Morris, R. (2005). A high-throughput path metric for multi-hop wireless routing. *Wirel. Netw.*
- Dawson-Haggerty, S.; Tavakoli, A. & Culler, D. (2010). Hydro: A hybrid routing protocol for low-power and lossy networks. Em *Smart Grid Communications (Smart-GridComm)*. IEEE.

- Doddavenkatappa, M.; Chan, M. C. & Ananda, A. L. (2012). Indriya: A low-cost, 3d wireless sensor network testbed. Em *Testbeds and Research Infrastructure. Development of Networks and Communities*, pp. 302--316. Springer.
- Dolev, S.; Elovici, Y. & Puzis, R. (2010). Routing betweenness centrality. *JACM*.
- Draves, R.; Padhye, J. & Zill, B. (2004). Routing in multi-radio, multi-hop wireless mesh networks. Em *Proceedings of the 10th annual international conference on Mobile computing and networking*, pp. 114--128. ACM.
- Dunkels, A.; Gronvall, B. & Voigt, T. (2004). Contiki - a lightweight and flexible operating system for tiny networked sensors. Em *Local Computer Networks, 29th IEEE International Conference*.
- Fambon, O.; FLEURY, É.; Harter, G.; Pissard-Gibollet, R. & Saint-Marcel, F. (2014). Fit iot-lab tutorial: hands-on practice with a very large scale testbed tool for the internet of things. *10èmes journées francophones Mobilité et Ubiquité, UbiMob2014*.
- Fasolo, E.; Rossi, M.; Widmer, J. & Zorzi, M. (2007). In-network aggregation techniques for wireless sensor networks: a survey. *Wireless Communications, IEEE*.
- Fonseca, R.; Gnawali, O.; Jamieson, K. & Levis, P. (2007). Four-bit wireless link estimation. Em *HotNets*.
- Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, pp. 35--41.
- Freeman, L. C. (1979). Centrality in social networks conceptual clarification. *Social networks*, 1(3):215--239.
- Freeman, L. C.; Borgatti, S. P. & White, D. R. (1991). Centrality in valued graphs: A measure of betweenness based on network flow. *Social networks*, 13(2):141--154.
- Gnawali, O.; Fonseca, R.; Jamieson, K.; Moss, D. & Levis, P. (2009). Collection tree protocol. Em *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*.
- Hwang, F. K.; Richards, D. S. & Winter, P. (1992). *The Steiner tree problem*. Elsevier.
- IEEE 802.11b-1999 (1999). <http://standards.ieee.org/>.
- IEEE 802.15.4 (2003). <http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf>.

- Instruments, T. (2006). CC2420: 2.4 GHz IEEE 802.15. 4/ZigBee-ready RF Transceiver.
- Intanagonwiwat, C.; Govindan, R.; Estrin, D.; Heidemann, J. & Silva, F. (2003). Directed diffusion for wireless sensor networking. *Networking, IEEE/ACM Transactions*.
- Jackson, A. W.; Milliken, W.; Santiváñez, C. A.; Condell, M. & Strayer, W. T. (2007). A topological analysis of monitor placement. Em *Network Computing and Applications, 2007. NCA 2007. Sixth IEEE International Symposium on*, pp. 169--178. IEEE.
- Jain, K.; Padhye, J.; Padmanabhan, V. N. & Qiu, L. (2005). Impact of interference on multi-hop wireless network performance. *Wireless networks*, 11(4):471--487.
- Jain, R. (2007). Channel models a tutorial1.
- Javaid, N.; Javaid, A.; Khan, I. & Djouani, K. (2009). Performance study of etx based wireless routing metrics. Em *2nd IC4 2009*.
- Jiang, P.; Huang, Q.; Wang, J.; Dai, X. & Lin, R. (2006). Research on wireless sensor networks routing protocol for wetland water environment monitoring. Em *Innovative Computing, Information and Control, 2006. ICICIC'06. First International Conference on*, volume 3, pp. 251--254. IEEE.
- Johnson, D.; Hu, Y.; Maltz, D. et al. (2007). The dynamic source routing protocol for mobile ad hoc networks. Relatório técnico, RFC 4728.
- Kim, S.; Fonseca, R.; Dutta, P.; Tavakoli, A.; Culler, D.; Levis, P.; Shenker, S. & Stoica, I. (2007). Flush: A reliable bulk transport protocol for multihop wireless networks. Em *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*. ACM.
- Krishnamachari, B.; Estrin, D. & Wicker, S. (2008). Modelling data-centric routing in wireless sensor networks. Em *IEEE infocom*, volume 2, pp. 39--44.
- Kuhn, F.; Moscibroda, T. & Wattenhofer, R. (2004). Unit disk graph approximation. Em *Proceedings of the 2004 Joint Workshop on Foundations of Mobile Computing, DIALM-POMC '04*, pp. 17--23, New York, NY, USA. ACM.
- LaI, D.; Manjeshwar, A.; Herrmann, F.; Uysal-Biyikoglu, E. & Keshavarzian, A. (2003). Measurement and characterization of link quality metrics in energy constrained wireless sensor networks. Em *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*, volume 1, pp. 446--452. IEEE.

- Lee, J. & Kim, S. (2010). Exploring the role of social networks in affective organizational commitment: Network centrality, strength of ties, and structural holes. *The American Review of Public Administration*.
- Levis, P.; Lee, N.; Welsh, M. & Culler, D. (2003a). Tossim: Accurate and scalable simulation of entire tinyos applications. Em *Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM.
- Levis, P.; Madden, S.; Polastre, J.; Szewczyk, R.; Whitehouse, K.; Woo, A.; Gay, D.; Hill, J.; Welsh, M.; Brewer, E. et al. (2005). Tinyos: An operating system for sensor networks. Em *Ambient intelligence*. Springer.
- Levis, P. A.; Patel, N.; Culler, D. & Shenker, S. (2003b). *Trickle: A self regulating algorithm for code propagation and maintenance in wireless sensor networks*. Computer Science Division, University of California.
- Li, J. & Mohapatra, P. (2005). An analytical model for the energy hole problem in many-to-one sensor networks. Em *VTC-2005-Fall IEEE 62nd*.
- Ma, Y. (2005). Improving wireless link delivery ratio classification with packet snr. Em *Electro Information Technology, 2005 IEEE International Conference on*, pp. 6--pp. IEEE.
- Macias, E.; Suarez, A. & Lloret, J. (2013). Mobile sensing systems. *Sensors*, 13(12):17292--17321.
- Moscibroda, T.; Wattenhofer, R. & Weber, Y. (2006). Protocol design beyond graph-based models. Em *Proc. of the ACM Workshop on Hot Topics in Networks (HotNets-V)*, pp. 25--30.
- Oliveira, E.; Ramos, H. & Loureiro, A. (2010). Centrality-based routing for wireless sensor networks. Em *Wireless Days, 2010 IFIP*.
- Orinoco wireless networks (2014). <http://www.orinocowireless.com/>.
- Paek, J. & Govindan, R. (2007). Rcrdt: Rate-controlled reliable transport for wireless sensor networks. Em *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*. ACM.
- Pastor-Satorras, R. & Vespignani, A. (2002). Immunization of complex networks. *Physical Review E*, 65(3):036104.

- Perkins, C. E. & Royer, E. M. (1999). Ad-hoc on-demand distance vector routing. Em *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*.
- Philip Buonadonna, Crossbow, G. T. (2007). MultiHopLQI. <http://www.tinyos.net/tinyos-2.x/tos/lib/net/lqi/>.
- Power, A. A. L. (2010). 2.4 GHz Transceiver for ZigBee, IEEE 802.15. 4, 6LoWPAN, RF4CE and ISM Applications, ATMEL Corporation, San Jose, CA.
- Qin, X. & Gu, Y. (2011). Data fusion in the internet of things. *Procedia Engineering*, 15:3023--3026.
- Ramos, H. S.; Frery, A. C.; Boukerche, A.; Oliveira, E. M. & Loureiro, A. A. (2014). Topology-related metrics and applications for the design and operation of wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 10(3):53.
- Reijers, N.; Halkes, G. & Langendoen, K. (2004). Link layer measurements in sensor networks. Em *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pp. 224--234. IEEE.
- Sabidussi, G. (1966). The centrality index of a graph. *Psychometrika*, 31(4):581--603.
- Senel, M.; Chintalapudi, K.; Lal, D.; Keshavarzian, A. & Coyle, E. J. (2007). A kalman filter based link quality estimation scheme for wireless sensor networks. Em *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, pp. 875--880. IEEE.
- Sinalgo, D. C. G. (2014). Simulator for network algorithms (sinalgo).
- Solis, I. & Obraczka, K. (2006). In-network aggregation trade-offs for data collection in wireless sensor networks. *International Journal of Sensor Networks*, 1(3):200--212.
- Srinivasan, K.; Dutta, P.; Tavakoli, A. & Levis, P. (2006). Understanding the causes of packet delivery success and failure in dense wireless sensor networks. Em *Proceedings of the 4th international conference on Embedded networked sensor systems*, pp. 419--420. ACM.
- Srinivasan, K.; Dutta, P.; Tavakoli, A. & Levis, P. (2010a). An empirical study of low-power wireless. *ACM Transactions on Sensor Networks (TOSN)*, 6(2):16.
- Srinivasan, K.; Dutta, P.; Tavakoli, A. & Levis, P. (2010b). An empirical study of low-power wireless. *ACM Transactions on Sensor Networks (TOSN)*, 6(2):16.

- Srinivasan, K.; Kazandjieva, M. A.; Agarwal, S. & Levis, P. (2008). The β -factor: measuring wireless link burstiness. Em *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM.
- Srinivasan, K. & Philip, L. (2006). Rssi is under appreciated. Em *Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets)*.
- Takahashi, H. & Matsuyama, A. (1980). An approximate solution for the steiner problem in graphs. *Math. Japonica*, 24(6):573--577.
- Tang, L.; Wang, K.-C.; Huang, Y. & Gu, F. (2007). Channel characterization and link quality assessment of ieee 802.15. 4-compliant radio for factory environments. *Industrial Informatics, IEEE Transactions on*, 3(2):99--110.
- TinyOS (2008). TYMO: DYMO on TinyOS. <http://tinyos.stanford.edu/tinyos-wiki/index.php/Tymo>.
- Villas, L.; Boukerche, A.; de Araujo, R. B. & Loureiro, A. A. F. (2010). Highly dynamic routing protocol for data aggregation in sensor networks. Em *ISCC 2010*. IEEE.
- Wang, X.-S.; Zhan, Y.-Z. & min Wang, L. (2008). Stcp: Secure topology control protocol for wireless sensor networks based on hexagonal mesh. Em *WiCOM '08*.
- Wang, Y.; Martonosi, M. & Peh, L.-S. (2007). Predicting link quality using supervised learning in wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(3):71--83.
- Winter, P. (1987). Steiner problem in networks: a survey. *Networks*, 17(2):129--167.
- Winter, T. (2012). Rpl: Ipv6 routing protocol for low-power and lossy networks.
- Woo, A. & Culler, D. E. (2003). *Evaluation of efficient link reliability estimators for low-power wireless networks*. Computer Science Division, University of California.
- Woo, A.; Tong, T. & Culler, D. (2003a). Taming the underlying challenges of reliable multihop routing in sensor networks. Em *Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 14--27. ACM.
- Woo, A.; Tong, T. & Culler, D. (2003b). Taming the underlying challenges of reliable multihop routing in sensor networks. Em *International Conference on Embedded Networked Sensor Systems*. ACM.

- Xu, Y. & Lee, W.-C. (2006a). Exploring spatial correlation for link quality estimation in wireless sensor networks. Em *Pervasive Computing and Communications, 2006. PerCom 2006. Fourth Annual IEEE International Conference on*, pp. 10 pp.–211.
- Xu, Y. & Lee, W.-C. (2006b). Exploring spatial correlation for link quality estimation in wireless sensor networks. Em *Pervasive Computing and Communications, 2006. PerCom 2006. Fourth Annual IEEE International Conference on*, pp. 10--pp. IEEE.
- Zhang, L.; Gao, D.; Zhao, W. & Chao, H.-C. (2013). A multilevel information fusion approach for road congestion detection in vanets. *Mathematical and Computer Modelling*, 58(5):1206–1221.
- Zhou, G.; He, T.; Krishnamurthy, S. & Stankovic, J. A. (2004). Impact of radio irregularity on wireless sensor networks. Em *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pp. 125--138. ACM.
- Zuniga, M. (2015). Building a network topology for tossim. <http://www.tinyos.net/tinyos-2.x/doc/html/tutorial/usc-topologies.html>.

Apêndice A

Publicações

Durante o mestrado, foram publicados dois artigos reportando os protocolos construídos ao longo dos estudos realizados. Abaixo são listadas as publicações:

- Bruno P. Santos, Marcos A. M. Vieira and Luiz F. M. Vieira. eXtend Collection Tree Protocol. *in IEEE WIRELESS AND COMMUNICATIONS AND NETWORKING CONFERENCE – WCNC, 2015.*
- Bruno P. Santos, Luiz F. M. Vieira, Marcos A. M. Vieira. CRAL: um algoritmo de roteamento baseado em centralidade e energia para Redes de Sensores Sem Fio. *in 33^o Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos – SBRC, 2015.*
- Bruna Soares Peres, Otavio Augusto de Oliveira Souza, Bruno Pereira Santos, Edson Roteia Araujo Junior, Olga Goussevskaia, Marcos Augusto Menezes Vieira, Luiz Filipe Menezes Vieira, Antonio Alfredo Ferreira Loureiro. Matrix: Multihop Address Allocation and Dynamic Any-to-Any Routing for 6LoWPAN. *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems - MSWiN, 2016.*
- Bruno P Santos, Lucas AM Silva, Clayson SFS Celes, João B Borges, Bruna S Peres Neto, Marcos Augusto M Vieira, Luiz Filipe M Vieira, Olga N Goussevskaia, Antonio AF Loureiro. Internet das Coisas: da Teoria à Prática. *XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos – SBRC, 2016*