Energy-efficient and reliable routing techniques for machine-to-machine communications

Zhao, Ming

2016

Zhao, M. (2016). Energy-efficient and reliable routing techniques for machine-to-machine communications. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/69202>

<https://doi.org/10.32657/10356/69202>

NANYANG
TECHNOLOGICAL
UNIVERSITY

**ENERGY-EFFICIENT AND RELIABLE
ROUTING TECHNIQUES FOR MACHINE
-TO-MACHINE COMMUNICATIONS**

ZHAO MING

School of Electrical & Electronic Engineering

2016

Energy-Efficient and Reliable Routing Techniques for Machine-to-Machine Communications

ZHAO MING

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2016

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor, Associate Professor Chong Han Joo Peter, for his patience and encouragement. Without his patience and guidance, I would never have been able to finish my research work and this dissertation.

I would like to give special thanks to Associate Professor Guan Yong Liang and Assistant Professor Rongxing Lu for their insightful comments, encouragement and valuable suggestions. Their expertise, understanding and patience added considerably to my graduate experience.

I appreciate Associate Professor Henry C.B. Chan and Assistant Professor Ivan Wang Hei Ho in Hong Kong Polytechnic University (HKPU) for their important advice and comments on this research.

I would like to thank Associate Professor Tseng King Jet for providing me the opportunity to join the Singapore-Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) Program.

I would also like to express my sincere gratitude and thanks to Dr. Arun Kumar for sharing his vast knowledge and skills with me. Moreover, Dr. Arun Kumar has inspired me to enjoy the beauty of scientific writing and patiently provided me with valuable suggestions at times of critical need.

Last but not least, I would also like to express my gratitude to all my family members for their support and encouragement during this time of study.

Table of Contents

Table of Contents	i
Abstract	iv
List of Figures	iv
List of Tables	v
List of Abbreviations	vi
List of Symbols	x
1 Introduction	1
1.1 Background	1
1.2 Motivation	5
1.3 Objectives	8
1.4 Summary of Contributions	8
1.5 Organization of the Thesis	11
2 An Overview of Machine-to-Machine Communications	12
2.1 Introduction	12

2.2	Key features of M2M communications	14
2.3	Capillary M2M Communications	15
2.3.1	Wireless Technologies for Capillary M2M Communications .	17
2.3.2	Standardization Activities of Capillary M2M Communications	19
2.4	Cellular M2M Communications	21
2.4.1	Architectural Enhancements of Cellular M2M Communications	22
2.4.2	Features of Cellular M2M communications in 3GPP	23
2.5	Active Research Areas of M2M communications	24
2.6	Summary	29
3	A Comprehensive Study on IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL)	30
3.1	Introduction	30
3.2	An Overview of Low-Power and Lossy Networks (LLNs)	32
3.2.1	Routing Protocols for LLNs	32
3.2.2	Applications of LLNs	37
3.3	An Overview of IPv6 Routing Protocol for Low Power and Lossy Networks (RPL)	40
3.3.1	DODAG Construction/Upward Routes	40
3.3.2	Downward Routes	41
3.3.3	Objective Functions	43
3.3.4	Trickle Algorithm	43
3.4	Implementation Framework of RPL in NS-3	45
3.4.1	Formation of Upwards Routes	47

3.4.2	Formation of Downwards Routes	48
3.4.3	Integration of Trickle Algorithm	49
3.5	Simulation Study	49
3.5.1	Simulation Setup	50
3.5.2	Performance Metrics	51
3.5.3	Performance Evaluation for Topology Initialization	51
3.5.4	Performance Evaluation of Traffic Patterns	54
3.6	Efficient Routing Support for LLNs	60
3.7	Summary	64
4	Point-to-point Hybrid RPL and Multipoint-to-point Cluster-parent-based RPL for LLNs	65
4.1	Introduction	65
4.2	Benchmarks	67
4.2.1	Point-to-point RPL (P2P-RPL)	67
4.2.2	Opportunistic RPL (ORPL)	69
4.3	Hybrid RPL Routing Protocol (HyRPL) for Point-to-point Traffic .	70
4.3.1	Routing Specification of HyRPL	71
4.3.2	Integrated Framework Design of HyRPL	72
4.4	Cluster-parent-based RPL (CRPL) for Multipoint-to-point Traffic .	77
4.4.1	Network Model	78
4.4.2	Selection of Cluster-Parent-Set	78
4.4.3	Coordination among Cluster Parent Set	82
4.5	Network Model and Performance Evaluation	83

4.5.1	Simulation Setup	84
4.5.2	Performance Evaluation of Point-to-point traffic	85
4.5.3	Performance Evaluation of Multipoint-to-point traffic	88
4.6	Summary	91
5	An Energy-Efficient Region-based RPL Routing Protocol for Low-Power and Lossy Networks	93
5.1	Introduction	93
5.2	System Architecture	96
5.2.1	Preliminary	96
5.2.2	P2P Traffic Modelling in LLNs	98
5.2.3	Energy Model	100
5.3	An Energy-efficient Region-based RPL for LLNs (ER-RPL)	100
5.3.1	Overview	101
5.3.2	Network Initialization Stage	102
5.3.3	Reactive P2P Route Discovery	108
5.4	Model Validation	117
5.5	Network Simulation	119
5.5.1	Experimental Setup	119
5.5.2	Performance Metrics	121
5.5.3	Performance Evaluation	121
5.6	Summary	127
6	An Energy-efficient and Cluster-parent-based RPL with Power-	

level Refinement for LLNs	128
6.1 Introduction	128
6.2 System Architecture	130
6.2.1 Network Model	131
6.2.2 Energy Model	132
6.2.3 Channel Model	133
6.3 An Energy-efficient and Cluster-parent-based RPL with Power-level Refinement for LLNs	134
6.3.1 Cluster-parent Set Selection	135
6.3.2 CPS Coordination for Packet Forwarding	142
6.3.3 Updating and Maintaining a Dynamic Topology	145
6.3.4 Supporting Generic Traffic Patterns	146
6.4 Simulation Study	148
6.4.1 Simulation Setup	148
6.4.2 Performance Metrics	150
6.4.3 Performance Evaluation	151
6.5 Summary	158
7 Conclusion and Future Works	159
7.1 Conclusion	159
7.2 Future Works	161
References	198

Abstract

In recent years, Machine-to-Machine (M2M) communications has emerged as ubiquitous communication among large number of embedded devices. M2M communications networks are typically composed of thousands of highly resource constraint devices connected with dynamic and lossy wireless links with predominantly unpredictable characteristics, categorized as Low-Power and Lossy networks (LLNs). Routing is of paramount importance for M2M communications, as the data has to be relayed via multiple hops before it reaches destination. The Internet Engineering Task Force (IETF) has standardized the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) as predominate candidate for M2M communications. In fact, the severe resource constraints and unreliable wireless medium of LLNs have driven the routing protocols towards an energy-efficient, reliable, stable, and scalable design. Power conservation and reliable data delivery are the two main primary goals to design routing protocols. Though, it is extremely challenging to achieve both simultaneously, especially for LLNs.

This thesis studies, designs, implements and evaluates the performance of energy-efficient and reliable routing techniques for M2M communications. Specifically, this thesis first presents a review of the state-of-the-art of M2M communications with focus on the enabling wireless communication technologies, standardization activities, and envisioned research opportunities. Furthermore, with a comprehensive study on RPL, a framework design of RPL is proposed. Aiming to simultaneously achieve high reliability and energy conservation in LLNs, suits of

routing protocols are proposed. Extensive experimental studies are conducted in NS-3 to validate the effectiveness and flexibility of proposed protocols.

This thesis proposes a hybrid RPL (HyRPL) aiming to provide reliable routing support with significant reduction of routing overhead. In HyRPL, the necessity for route discovery is determined by measuring the cost of pre-established route. With the design of a route constraint, HyRPL effectively balances the trade-off between utilizing the pre-established route and on-demand route discovery in LLNs.

This thesis also proposes a cluster-parent based RPL (CRPL), which effectively leverages path diversity so as to improve the reliability and reduce the number of retransmissions. With the objective to minimize the total number of transmissions, CRPL allows each node to select an optimal forwarder-set with a low-complexity algorithm. Moreover, an efficient loss recovery scheme for the detection and retransmission of lost data packets is designed to avoid duplicate transmissions.

This thesis also proposes a novel energy-efficient and region-based routing protocol, called ER-RPL. In contrast to traditional routing protocols, where all the nodes are required for route discovery, ER-RPL only requires a subset of nodes. By exploiting the region feature of static M2M communication networks, ER-RPL achieves energy conservation without compromising the reliability. Both the theoretical analysis and extensive simulation studies indicate the performance superiority of proposed ER-RPL.

Further enhancing the CRPL, a hybrid energy-efficient and cluster-parent-based RPL (HECRPL), which prolongs the lifetime of the network while achieving high reliability and fairness, is proposed. In addition, a novel routing metric to precisely measure the end-to-end transmission cost is also designed. At last, a distributed algorithm is proposed to select the optimal transmission power and forwarder list.

List of Figures

1.1	High-level architecture proposed by ETSI to support M2M communications.	2
1.2	The IPv6 protocol stacks standardized by IETF and IEEE to support capillary M2M communications.	3
1.3	Architecture of RPL routing domain.	3
2.1	A simple architecture of capillary M2M communications.	17
2.2	Illustration of spectrum usage in IEEE 802.11a and 802.11ah. . . .	18
2.3	Illustration of spectrum usage in IEEE 802.15.4 and 802.11b. . . .	20
2.4	A simple architecture of cellular M2M communications.	22
2.5	Cognitive gateway architectures: (a) Gateway connecting Non-IP based component and IP based component; (b) Gateway connecting capillary and cellular M2M communications.	26
3.1	Architecture of RPL routing support for building automation system.	38
3.2	Architecture of RPL routing support for smart meter system. . . .	39
3.3	DODAG topology constructed by RPL.	41
3.4	Simplified RPL class diagrams integrated with NS3 routing module.	46
3.5	Route processing diagram at the source node subject to different traffic patterns.	47
3.6	Packet format of the DODAG Information Solicitation.	47
3.7	Packet format of the DODAG Information Object.	48

3.8	Packet format of the Destination Advertisement Object.	49
3.9	Initialization of the DODAG Topology varying the network size under different root placement. (a) The maximum and average rank. (b) Routing control overhead. (c) Network convergence time.	53
3.10	Impact of link success rate under MRHOF and OF0 with MP2P traffic. (a) PDR. (b) Average path length. (c)End-to-end delay.	55
3.11	Impact of link success rate under MRHOF and OF0 with P2P traffic. (a) PDR. (b) Average path length. (c) End-to-end delay.	57
3.12	Impact of link success rate under MRHOF and OF0 with MP2P traffic. (a) PDR. (b) Average path length. (c)End-to-end delay.	59
4.1	P2P communication route supported by RPL and P2P-RPL.	69
4.2	The route processing diagrams of HyRPL at the source node subject to different traffic patterns.	73
4.3	Designed classes diagrams of HyRPL in NS-3.	74
4.4	Packet format of the measurement object.	76
4.5	Packet format of the Route Discovery Option.	76
4.6	Packet format of the Discovery Reply Object.	77
4.7	The process of cluster-parent set selection in CRPL.	81
4.8	Illustration of overhearing-based coordination approach with node H 's CPS.	83
4.9	Coordination among nodes in a CPS. (a). Node H attaches its CPS in the packet header following priority order. (b) A local forwarding table of node D containing the coordination information.	84
4.10	Impact of the link success rate at the edge on PDR.	85
4.11	Impact of the link success rate at the edge on average path length.	86
4.12	Impact of the link success rate at the edge on end-to-end delay.	87
4.13	Impact of the link success rate at the edge on routing control overhead.	88

4.14	The PDR performance of RPL, ORPL and CRPL. (a) root is randomly placed. (b) root is placed at the corner of the network.	89
4.15	Packet retransmission performance of RPL, ORPL and CRPL. (a) root is randomly placed. (b) root is placed at the corner.	90
4.16	Average path cost performance of RPL, ORPL and CRPL. (a) root is randomly placed. (b) root is placed at the corner.	91
5.1	Two examples of LLNs applications that exploit the region feature.	95
5.2	The Voronoi diagram of a network covered by the Reference Nodes.	96
5.3	An example of reference nodes and regions in the network.	98
5.4	The packet structure of RFO message.	102
5.5	Four temporary DODAGs rooted at each RN are built during network initialization stage.	103
5.6	The distributed self-regioning algorithm: (a) Step I. (b) Step II. . .	105
5.7	Packet structure of MRO.	110
5.8	Region-to-Region routing without route discovery. (a) The routing between <i>Region II_A</i> and <i>Region I_D</i> is enabled. (b) R2R routing is used for s_2 and d_2 . (c) The intermediate node, d_1 , can locates either in s_2 's region or d_2 's region. (d)R2R routing with route adjustment.	112
5.9	Illustration of Dead Zone problem.	115
5.10	Scenario I: The PDR comparison. (a) grid with random deployment. (b) random planar deployment.	121
5.11	Scenario I: Control overhead of simulation and theoretical results. (a) grid with random deployment. (b) random planar deployment. .	122
5.12	Scenario I: Energy consumption per successfully received data. (a) grid with random deployment. (b) random planar deployment. . . .	123
5.13	Scenario II: The PDR comparison. (a) grid with random deployment. (b) random planar deployment.	124
5.14	Scenario II: Hop count comparison.. (a) grid with random deployment. (b) random planar deployment.	124

5.15	Scenario II: Control overhead of simulation and theoretical results. (a) grid with random deployment. (b) random planar deployment.	125
5.16	Scenario II: Energy consumption per successfully received data . (a) grid with random deployment. (b) random planar deployment.	126
5.17	Scenario II: End-to-End delay comparison. (a) grid with random deployment. (b) random planar deployment.	126
6.1	Network formation with an optimal cluster-parent set formation.	136
6.2	The CPS formation control message format.	138
6.3	The optimal CPS selection at node j	139
6.4	Illustration of the adjustable transmission range.	140
6.5	Reduced-complexity design of the optimal CPS selection at node i	143
6.6	Scenario I: Different transmission ranges of RPL: (a) P_{pdr} . (b) e_n	151
6.7	Scenario II: Non-uniform distribution. (a) N_{alive} . (b) e_n	152
6.8	Scenario II: Non-uniform distribution. (a) T_{life} . (b) P_{pdr}	153
6.9	Scenario II: Non-uniform distribution. (a) e_n . (b) T_{life}	154
6.10	Scenario III (A) A uniform distribution with different link success rates. (a) P_{pdr} . (b) e_n	155
6.11	Scenario III (A) A uniform distribution with different link success rates. (a) N_{alive} . (b) T_{life}	155
6.12	Scenario III (A) A uniform distribution with different traffic flows. (a) P_{pdr} . (b) e_n	156
6.13	Scenario III (B) A uniform distribution with different network sizes. (a) T_{life} . (b) e_n . (c) O_r . (d) T_{delay}	157
7.1	The Firestorm mote.	162
7.2	The structure of Storm, Firestorm and embedded technologies.	163
7.3	The RPL network with Firestorms.	164

List of Tables

2.1	The distinctive characteristics of M2M communications	16
2.2	Standards for capillary M2M at different protocol layer.	19
3.1	Comparision of routing protocols for LLNs.	36
3.2	Main features of RPL routing protocol.	42
3.3	Current Routing Metric/Constraint Objects in LLNs.	43
3.4	Simulation parameters to evluate the performance of RPL.	50
4.1	Simulation parameters to evluate the performance of HyRPL and CRPL.	85
5.1	Simulation parameters to evluate the performance of ER-RPL. . . .	120
6.1	Simulation parameters to evluate the performance of HECRPL. . .	149
7.1	Summary of the proposed routing protocols.	161

List of Abbreviations

6LoWPAN	the IPv6 over low power wireless personal area network
ACK	acknowledgement
ARS	adaptive region selection
AMI	advanced metering infrastructure
CA	collision avoidance
CFM	cluster formation message
COAP	constrained application protocol
CPS	cluster-parent set
CRPL	cluster parent based RPL
CSMA	carrier sense multiple access
CTP	collection tree protocol
DAO	Destination Advertisement Object
DIO	DODAG Information Object

DIS	DODAG Information Solicitation
DODAG	destination-oriented directed acyclic graphs
DRC	region code of destination node
DRO	discovery reply object
ETSI	European Telecommunications Standards Institute
ETX	expected transmission count
FFD	full function device
H2H	human-to-human
HyRPL	hybrid RPL
ICT	information and communication technology
IoT	Internet of Things
LAN	local area network
LLNs	low-power and lossy networks
M2M	machine-to-machine
MAC	medium access control
MAN	metropolitan area network
MDP	markov decision process

MO	measurement object
MRO	message reply object
MP2P	multipoint-to-point
MTC	machine type communication
MTCD	machine type communication device
MTCG	machine type communication gateway
NC	network coding
OF	objective function
OR	opportunistic routing
ORPL	opportunistic RPL
P2MP	point-to-multipoint
P2P	point-to-point
P2P-RPL	point-to-point RPL
PDR	packet delivery ratio
PHY	physical layer
QoS	quality of service
R2R	region-to-region

RC	region code
RCM	region code map
RDO	route discovery option
REC	remaining energy cost
RFD	reduced function device
RN	reference node
RPL	IPv6 routing protocol for low-power and lossy networks
ROLL	routing over low-power and lossy network
RSSI	received signal strength indicator
SRC	region code of source node
SNR	signal-to-noise ratio
TCP	transmission control protocol
UDP	user datagram protocol
UDGM	unit disk graph model
WAN	wide area network
WPAN	wireless personal area network
WSN	wireless sensor network

List of Symbols

n	number of nodes
N	number of reference nodes
Ω	the set of reference nodes
ρ	network density
R	the transmission range of nodes
λ	the number of traffic flows
p_{ij}	the successful delivery probability from node i to j
E_{ij}	one hop ETX from node i to j
C_I	the <i>Coordinate</i> of RN I
$RNM(i, j)$	the entry in the i th row and j^{th} column of the matrix RNM
$RCM(m, n)$	the entry in the m th row and n^{th} column of the matrix RCM
M	the number of adjustable transmission ranges
W	the maximum size of the cluster-parent-set

N_i	the number of one-hop neighbors of node i
n_i	the size of node i 's CPS
w_i	the size of node i 's optimal CPS
wg_i	the link weight of node i
β_i	average link weight of node i
R_{o_i}	the optimal transmission range of node i
$CPS_o(i)$	the optimal CPS of node i
$E_{tx}(k, r)$	the energy consumed in transmitting a k -bit message with a transmission range r
$E_{rx}(k)$	the energy consumed in receiving a k -bit message
$E_{oh}(k)$	the energy consumed in overhearing a k -bit message
$\{nb_{i_r}\}$	the one-hop neighbor list of node i with a transmission range r
$\{nb_{i_h}\}$	the list of helpers of node i
Re_j	the residual energy of node j
$P_{CPS(i),j}$	the priority of node j in node i 's CPS
T_{tr}	the time for a packet to travel from one node to another node
T_{Aw}	the time for a node to wait for an ACK

Chapter 1

Introduction

1.1 Background

With the advancement and pervasiveness of information and communication technology (ICT), it is widely predicted that the number of embedded devices will increase tremendously and finally they will connect to the Internet as the Internet of Things (IoT) [1, 2]. As a fundamental enabler of IoT, Machine-to-Machine (M2M) communications [3] are cornerstones for the networked society with seamless connection. M2M communications refer to a large number of resource constraint embedded devices sharing information and making collaborative decisions with little or no human intervention. The autonomous characteristics of M2M communications, which decrease the cost for human resources significantly, give impetus for the research of M2M communications in both industry and academic areas. In order to put M2M communications into reality, European Telecommunications Standards Institute (ETSI) has defined a high-level architecture [4], as shown in Fig. 1.1. In particular, this architecture consists of three domains: *M2M devices domain*, *Network domain*, and *Application domain*. Specifically, *M2M devices domain* comprises large scale of low-power and embedded devices, the connectivity of which is typically supported by ad hoc networks. The convergent points such as network coordinators in *Network domain* serves to bridge the Internet and heterogeneous networks. Various control or management services are provided in *Application domain*.

Figure 1.1: High-level architecture proposed by ETSI to support M2M communications.

M2M communications can be classified into two categories based on communication technologies used, i.e., capillary M2M communications [5] and cellular M2M communications [6]. Different from cellular M2M communications, which rely on the cellular network infrastructure, capillary M2M communications are dominated by low-power, and low-cost embedded design with short-range communication technologies. Specifically, capillary M2M networks are typically composed of highly resource constraint devices interconnected with dynamic and lossy wireless links with unpredictable qualities, categorized as Low-Power and Lossy networks (LLNs). The dynamic and lossy nature of wireless links predominantly result from the co-channel interference, channel fading, and harsh environment (heat, dust, and moisture). Essentially, M2M communications require thousands of low-power and low cost embedded devices to be efficiently and seamlessly interconnected. The flourish of M2M communications eagerly seeks for wireless solutions to provide interoperability to network components because of the cost-effective, easy-to-install, scalable and flexible features of wireless communication. However, direct communication is rarely possible in large scale networks due to the limited range of wireless radio signal. Thus, data has to be relayed through multi-hops before it reaches the destination. Consequently, routing plays a crucial role in providing the interoperability for M2M devices so to finally enable M2M communications. Therefore, routing is an integral part of M2M communications. In order to put M2M communications into reality, IETF has standardized the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) [7], which is considered as the predominate standards of the network layer protocol for capillary M2M communications, as shown in Fig. 1.2.

Figure 1.2: The IPv6 protocol stacks standardized by IETF and IEEE to support capillary M2M communications.

As the protocol stack integration of capillary M2M deepens, the importance of RPL increases. RPL is an extensive IPv6-based distance-vector proactive routing protocol. Specifically, as a proactive routing protocol, RPL supports various traffic patterns, including multipoint-to-point (MP2P), point-to-point (P2P), and point-to-multipoint (P2MP). RPL forms the network as a direction oriented acyclic graph (DODAG), the architecture of which is shown in Fig. 1.3. The pervasiveness of M2M communications requires a comprehensive study on RPL to pave the way for researchers to understand and contribute in the relevant research area on routing for M2M communications. In particular, RPL is regarded as the most promising routing protocol to support LLNs applications due to its considerable flexibility.

Figure 1.3: Architecture of RPL routing domain.

LLNs applications, which emerges as the predominant paradigm for M2M communications, essentially require reliable and energy-efficient routing to support the connectivity of network utilities with tighter control and energy conservation. However, achieving high reliability and energy conservation simultaneously is a challenging issue while designing a routing protocol, especially for LLNs. Specif-

ically, the wireless channel generally experiences drastic variation due to channel fading, causing unreliable data transmission or prone to be a failure. In this situation, retransmission generally serves as a necessary way to mitigate the packet loss. Obviously, the more number of retransmission it requires, the more energy is depleted and the less efficient the bandwidth is. Moreover, nodes in LLNs are usually battery powered, have to run complex computational processes and face severe energy constraints when operating in a lossy environment. As a result, some of the connected nodes become isolated once their energy drain out. Importantly, power conservation and reliable data delivery are the two major goals while designing a routing protocol, especially for the network consisting of limited energy nodes which operate in harsh environments.

With severe energy constraint and limited memory, nodes in LLNs are typically self-organized, low-mobility, and densely deployed. This necessitates efficient and diverse solutions for some of the conventional wireless networking problems, such as energy-efficient medium access control, adaptive routing, dynamic bandwidth allocation, self-organization and so on. It is shown that traditional routing protocols face difficulties in coping the severe resource constraints and unreliable wireless links in LLNs [8]. In addition, the existing routing protocols for LLNs either are optimized for multipoint-to-point (MP2P) traffic or exclusively focus on the point-to-point (P2P) traffic paradigm in homogeneous networks. Motivated by the fact that the wide variety of LLNs applications are becoming more general, an ever growing support for a hybrid of these traffic paradigms in heterogeneous networks are expected.

In brief, the flourish of M2M communications still hinges on fully understanding and managing the challenging issues. In particular, the constraints of LLNs and routing challenges must be well addressed prior to the wide deployment of LLNs applications. This thesis presents a comprehensive study on M2M communications and RPL, well tackling their state-of-art statuses and the challenges abiding by the severe and inherent constraints of LLNs, and proposes suites of reliable and energy-efficient routing solutions to effectively boost the routing performance for LLNs.

1.2 Motivation

Owing to its autonomous characteristics, M2M communications have brought vast potential opportunities for designing real-time monitoring and remote control applications. Specifically, LLNs applications cover a wide scope of applications, including but not limit to, building automation, industrial control, urban environment and home automation [9, 10, 11]. However, before LLNs applications can be used as commercial products and to serve people's lives, the research developments and challenges in this area must be well tackled. In particular, reliable and energy-efficient routing is crucial for LLNs, because nodes are suffer from very limited battery power and operated in dynamic and lossy environment. Many research challenges still lie ahead, mostly due to severe resource constraints and stringent requirements imposed by diverse LLNs applications. The development of LLNs applications leads to an increased popularity of RPL. In fact, before a routing protocol is used for real world applications, simulation and evaluation of its performance is crucial in order to better understand its routing mechanism. The routing performance of RPL has been studied with the existing implementations in Omnet++ [12], Contiki [13] and TinyOS [14]. However, the function of those existing RPL implementations are limited, and the implementation platforms are difficult to add new features due to a lack of flexibility [15]. Although the work in [16] proposes a model of RPL in Network Simulator 3 (NS-3), it is still not accessible at current moment. The high demand and absence of a flexible implementation of RPL has driven this thesis to investigate, design and implement the framework of RPL on NS-3. As an open source discrete-event network simulator and primarily targeting for research and educational use, NS-3 is equipped with all the necessary IPv6 stack models [17]. Specifically, NS-3 is extremely flexible for users to add new features and configure parameters in various ways. Therefore, NS-3 becomes the promising candidate to implement RPL and examine its routing performance so as to pave the way for the research of RPL.

Point-to-point (P2P) communication, as an integral traffic paradigm in numerous LLNs applications, is becoming more and more important with the emergence of M2M communications. Due to the resource constraint, the optimal routing paths between arbitrary nodes usually are not provided by default. Two important routing methods, i.e., proactive and reactive are commonly applied to support

P2P communication. As a proactive routing protocol, RPL pre-establishes the DODAG and supports P2P communication by an intermediary. Specifically, in RPL, packets need to travel along the pre-established route, leading to long end-to-end delay and large numbers of unreliable links. In particular, the non-storing mode of RPL requires the root to act as the only intermediary. As a result, the root easily becomes the bottleneck when the traffic is high. A reactive routing protocol, i.e., point-to-point RPL (P2P-RPL) [18], is standardized as an enhancement of RPL focusing on the support for P2P communication. P2P-RPL disseminates the route discovery messages throughout the whole network to discover the best-quality route. As a result, it incurs exceeding control overhead, especially in dense and large scale networks. Transmitting large amount of control overheads is very expensive in terms of energy cost for LLNs. Importantly, the region information is a piece of important information in static M2M networks. In many LLNs applications, such as surveillance and building automation control application, the predominant communication usually takes place in certain regions at some period. As one of the most elemental structures in the computational geometry, the Voronoi diagram has extensively investigated the region feature and has various applications in a diverse of fields [19]. In particular, the concept of region defined in the Voronoi diagram can be adopted to improve the efficiency of route discovery. Specifically, the route discovery can be performed among a subset of regions in the network so as to significantly reduce the amount of control overhead, especially for large scale networks. In this way, energy-efficient route discovery can be attained without compromising the reliability.

Essentially, as most of the LLNs applications focus on a various type of data collection, the predominate communication paradigm for LLNs is multipoint-to-point (MP2P) traffic. RPL selects a single preferred parent and exclusively send packets through its preferred parent. The dynamic and lossy environments result in excessive packet retransmissions, which leads to a great deal of energy consumption and long occupation time of the shared wireless medium. Moreover, the power of nodes along some paths drain quickly in single path routing, so that some of the connected nodes become isolated. By leveraging path diversity, opportunistic routing (OR) [20], shows advantage in improving the network reliability. Some related works take advantage of this technique to provide reliable routing while reducing the number of transmissions [21, 22], which indirectly conserve energy. Although

the energy consumption during transmission is the major component of the total energy depletion of the network, some other factors also constitute a significant portion of the total energy consumption. The other major factors are the lossy rate of wireless medium, the energy consumption in reception and idle-listening, which significantly affect the total energy depletion of a dense lossy networks. In addition, it has been shown that in traditional networks the transmission power is usually unnecessarily high [23, 24, 25], which shortens the lifetime of the nodes and introduces excessive mutual interference. The trade-off is that the progress made in each transmission is more with higher transmit power. Recent developments have enabled the rapid advancement and pervasiveness of the multifunctional and energy-constraint nodes with continuous or discrete power levels [26]. To the best of our knowledge, the adjustable transmission power for optimal route selection in terms of energy-efficiency and reliability has not been thoroughly studied. Moreover, the uneven deployment of nodes and heterogeneous traffic patterns result in an unbalanced distribution of workload. So the nodes in the hotspot area quickly exhaust their energy, which is known as the “*Energy Hole Problem*”. Therefore, workload sharing is inherently important, as it prolongs the lifetime of the network and mitigate the occurrence of energy holes. With the goal of achieving reliability and energy efficiency, the routing metric that can precisely measure the transmission cost is in high demand. In particular, by exploring the benefit of OR, the mechanism to optimally select the forwarder lists should be well addressed to achieve the goal. Furthermore, the trade-off of the transmit power should be balanced with load balancing strategy.

Power conservation and reliable data delivery are the two main primary goals of designing routing protocols for LLNs. However, it is extremely challenging and critical to simultaneously achieving both reliability and energy-efficiency, especially for LLNs. Obviously, a practical solution to balance the trade-off between reliability and energy consumption is in high demand. The pervasiveness of LLNs applications call for a distributed, reliable, scalable and energy-efficient design of routing protocol.

1.3 Objectives

Putting up M2M communications for real-life use essentially requires thousands of low-power and low cost embedded devices to be efficiently and seamlessly interconnected. However, abiding by the severe and inherent constraints of LLNs, achieving energy-efficient and reliable routing simultaneously is an extremely challenging but critical issue in the design of routing protocols. Therefore, the overall objective of this dissertation is to study, design, and evaluate the performance of energy-efficient and reliable routing techniques for LLNs. It is envisioned that improving the reliability of the network as well as prolonging the lifetime of the network are the essential approach to make the promise of M2M communications into a reality. The above aim will be accomplished by fulfilling the following research objectives:

- To thoroughly review the state-of-the-art of M2M communications as well as the main research areas and envisioned opportunities associated with it.
- To conduct a comprehensive study on RPL so as to fully understand its limitations, categorise the research directions, and identify the challenges in LLNs.
- To propose a routing protocol that provides reliable data delivery while achieving significant reduction of control overhead.
- To develop a routing protocol that achieves significant improvement on the network reliability and reduction of packet retransmissions.
- To design a routing protocol that supports energy-efficient route discovery without compromising the reliability.
- To propose a routing protocol that can improve the reliability of the network as well as prolong the lifetime of the network.

1.4 Summary of Contributions

The research in this thesis focuses on developing suite of routing protocols to provide energy-efficient and reliable routing support for LLNs. Furthermore, extensive

simulation studies are conducted across various scenarios to validate the flexibility and effectiveness of the proposed routing protocols with comparison to the standardized protocols as the benchmarks. Specifically, the main contributions lie in the following aspects:

- First, this thesis presents the overall picture of M2M communications by categorizing it into two categories; capillary M2M communications and cellular M2M communications. With thorough investigation on the current standards of capillary M2M communications defined by IEEE and IETF and the enhancement architecture of cellular M2M communications defined by 3GPP, the recent development of wireless communication technologies and active research area of M2M communications are introduced.
- Second, by exploiting the trust-able, practicable, and flexible features of NS-3, a framework of RPL is designed and implemented in NS-3. This essentially unveils the ambiguities regarding the conformity and validation of RPL, and reduces its complexity in the implementation design. It is believed that this approach will benefit future research on routing protocols for LLNs. In addition, the main research challenges of RPL abiding by the severe and inherent constraints of LLNs are also critiqued. Furthermore, with a categorization of the main research directions of RPL, this thesis provides valuable insights that suggest enabling techniques to improve the performance and further increase the flexibility of RPL so as to encourage a wider adoption of it.
- Third, with a focus on P2P traffic support, the hybrid RPL (HyRPL) is proposed to support reliable data delivery with significant reduction of the amount of control overhead. HyRPL is a hybrid of proactive and reactive routing protocol developed based on RPL and P2P-RPL. In HyRPL, the necessity for route discovery is determined through measuring the cost of the route along the pre-established DODAG. With the design of route constraint, HyRPL effectively balances the trade-off between utilizing the pre-established route and on-demand route discovery.
- Fourth, with a focus on MP2P traffic support, the cluster-parent based RPL (CRPL) is proposed aiming at achieving significant improvement on the network reliability and reduction of packet retransmissions. By exploiting the

benefit of OR to leverage path diversity, the selection of optimal forwarder lists is formulated as a low-complexity algorithm with the objective to minimize the total number of transmissions. In addition, to mitigate the coordination error, an overhearing-based coordination scheme is employed to avoid packet duplications.

- Fifth, an energy-efficient region-based RPL (ER-RPL) that supports energy-efficient route discovery without compromising the reliability is proposed. In contrast of traditional routing protocols where all the nodes are required for route discovery, the proposed scheme only requires a subset of nodes to participate in the process, which is the key for energy conservation. In addition, ER-RPL inherits the mechanism from RPL to pre-establish the topology of the network such that it can support MP2P communication with an optimal topology. In ER-RPL, the region feature of static M2M networks is exploited, and a distributed algorithm that allows a node to estimate its region is designed. Furthermore, in order to reduce the control overhead for the route discovery, an efficient region-based route discovery is designed so that only a subset of nodes in some regions are required to join the route discovery. Furthermore, to make use of the region information, a region-to-region routing without route discovery scheme is designed. In addition, in order to mitigate the dead zone problem, an adaptive region selection is proposed.
- Sixth, by extending the previous work, i.e., CRPL, this thesis proposes a hybrid, energy-efficient cluster-parent based RPL (HECRPL), which further copes with the dynamic nature in LLNs and effectively prolong the lifetime of the network while achieving high reliability and fairness for the network. Specifically, a routing metric that can precisely measure the energy consumption of the end-to-end transmission of the network is designed. In addition, an algorithm that defines the way to optimally select the forwarder list and transmission range is proposed. In addition, based on a hybrid of residual energy and the lossy conditions of wireless channels, the priority in a forwarder set is dynamically updated to balance the workload of the network.

1.5 Organization of the Thesis

The organization of the remainder of the thesis is as follows. Chapter 2 presents an overview of M2M communications by reviewing state-of-the-art M2M technologies, the main research areas and envisioned opportunities with a categorization of capillary M2M communications and cellular M2M communications. Chapter 3 presents a comprehensive study on RPL with focus on the framework design, performance evaluation, research areas and envisioned opportunities. Chapter 4 proposes a hybrid RPL (HyRPL) and a cluster-parent based RPL protocol (CRPL) aiming to enhance the network performance under P2P and MP2P traffic patterns, respectively. Chapter 5 proposes an energy-efficient region-based routing protocol (ER-RPL) that is designed to achieve both reliable and energy-efficient data delivery for static networks. Chapter 6 introduces the design of a hybrid, energy-efficient, and cluster-parent based RPL routing protocol, called HECRPL, aiming is to simultaneously achieving high reliability as well as great energy conservation. Finally, conclusions and future research work are summarized in Chapter 7.

Chapter 2

An Overview of Machine-to-Machine Communications

2.1 Introduction

The concept of M2M communications has changed over time. Initially, M2M communications merely referred to a set of sensors, which collect the information and transmit this information to a specific application. Nowadays, the meaning of M2M communications has become much broader. Any machine or device, which generates data and is capable of transmitting this information to any other device via a wireless or wired communication network without human intervention, can be called M2M-enabled device and this process is called M2M communications. In addition, machines can make decision collaboratively without human intervention. M2M communications are characterized by the autonomous characteristics, which provide comprehensive solutions to various monitoring and control applications. The self-organizing M2M devices are tiny and low cost computing devices, including sensors, actuators and other communication devices, and usually operate in hash environment. Ubiquitous connectivity among M2M devices needs to be provided to support seamless communication among heterogeneous networks for M2M applications and services.

M2M communications can be classified into two categories based on commu-

nication technologies used; capillary M2M communications [5] and cellular M2M communications [6]. Capillary M2M communications refer to mostly embedded, low-power, and low-cost design with short-range communication systems. While insights from academic research on wireless sensor networks (WSNs) can be used, capillary M2M communications are dominantly supported by ad hoc networks. In cellular M2M communications, long-distance transmission technologies and licensed spectrum are used. For better penetration of M2M devices into the cellular market, it is envisaged that specific optimization for cellular M2M communications will be introduced in LTE-A [27].

In order to bring M2M communications into reality, it is crucial to understand the current technologies and standardization activities in this area. Over the past years, much research has been done on M2M communications [3, 28, 29, 30]. The IP-based standards for M2M communications has been reviewed and the challenges associated with each protocol stack are analyzed in [31]. The features and challenges of M2M communications in telecommunication networks are presented in [27, 32, 33]. The work in [34] introduces the architecture and the standards of home M2M networks, and a practical example for home area M2M communications can be found in [35]. Most of the existing research on M2M communications only focus on one specific type of M2M communications and the research issues associated with it. To illustrate M2M communications on a broader scale, this chapter presents the overall picture of M2M communications by introducing different types of M2M communications, the latest standards and active research areas associated with it.

This chapter presents a survey of different categories of M2M communications and recent development progress that have been made in M2M communications area. The remainder of this chapter is organized as follows. Section 2.2 introduces the distinguishing features of M2M devices. Then, two categories of M2M communications, including capillary M2M communications and cellular M2M communications, are discussed in detail. Specifically, Section 2.3 presents the current standards of capillary M2M communications defined by IEEE and IETF, and Section 2.4 presents the enhancement architecture of cellular M2M communications defined by 3GPP [36]. Section 2.5 reviews the main research focus of M2M communications by discussing the technologies and challenges associated with it. Finally,

this chapter is summarized in Section 2.6. Our aim is to provide a better understanding of current research and the progress till date in this field of research.

2.2 Key features of M2M communications

The main functions of M2M devices can be categorized as follows:

- *Data collection*: A group of augmented sensors or RFIDs collect the raw data and generate the information.
- *Data processing*: There should be a platform module responsible for analyzing and processing the data. After this procedure, only meaningful data will be transmitted.
- *Specific function*: Unlike sensor nodes, M2M devices are usually actuators with their specific functions rather than just collecting the surrounding raw data and information.
- *Certain storage*: The space of storage is determined by the proposed functionality of the device.
- *Distinctive address*: Each M2M device will be assigned an inimitable identifier.
- *Information transceiver*: This partition will connect M2M devices with the external world.
- *Power-supply system*: Devices can be powered either through a wire power supply or battery supply.

Some of the major characteristics of M2M communications are as follows:

- *Multitude and variety*: Owing to a large number of M2M devices and their enormously diverse Quality of Service (QoS) requirement, the huge amount of M2M traffic may lead the communication system overwhelmed. Meanwhile, the M2M traffic is usually sporadic and is device and application specific. These features bring unique challenges to the design of an efficient and reliable

M2M system. Beside this, distinct address must be assigned to each device, which requires huge address space.

- *Self-organization*: One of the main features of M2M communications is that the communication takes place with little or no human intervention. Thus, M2M devices are supposed to be self-sufficient which includes the capability of self-configuration, self-organization, self-healing, and self-management [34].
- *Reliability and robustness*: Without humans' intervention, the failure of a M2M network element can propagate, which can lead to serious problems for the whole system. Moreover, due to the dynamic environment in which M2M devices work in, the robust feature of M2M communications has to be to ensure a reliable transmission.
- *Security and privacy*: Highly sensitive and confidential data are handled by M2M network. However, it is difficult to detect network and system attacks because of the autonomous nature of M2M communications. Security vulnerabilities could result in privacy disclosure. An economical and convenient solution is required to ensure that M2M systems are secure.

In M2M communications, different access technologies and protocols are used to achieve ubiquitous connectivity among M2M devices fulfilling diverse QoS requirements, resulting in some unique characteristics, shown in Table 2.1.

2.3 Capillary M2M Communications

Capillary M2M communications refer to mostly embedded, low-power, and low-cost design with short-range communication systems. In capillary M2M communications, devices in one area are usually organized with a tree-topology or mesh network. Typically they are connected wirelessly but it is not necessary. And, the capillary network connects to a cellular network via a gateway. A simple architecture of capillary M2M communications is shown Fig. 2.1. Capillary M2M communications system needs to fulfill the requirements to support large number of devices' interoperability with low power consumption, and handle burst and inter-

Table 2.1: The distinctive characteristics of M2M communications

Category	Features	Explanation
Device Type	Full Function Device (FFD) and Reduced Function Device (RFD)	FFD can serve as a leaf or the network coordinator and it can establish the communication link as a router. RFD cannot act as a network coordinator. It can only become the end node of the network.
Distribution Type	Decentralized distribution	The intelligent devices in M2M communications can generate the data and determine when to start the communication.
Transmission	Different rates among nodes	The data rates of different kinds of devices vary a lot. How to forward packets to next relay need to be efficiently designed.
Processing Ability	Memory utilization/Processor speed/ Database or knowledge-base	The intelligent device can process, aggregate, and compute the data before transmitting. This meaningful data can be transmitted directly to the application in the end-server without going through a gateway.
Communication Technology	One communication from different wireless technologies	Automatic communication among different technologies, such as Global System for Mobile Communication (GSM), WiFi, and Bluetooth and so on, can be achieved with the help of multiple physical (PHY) layer and medium access control (MAC) layer protocols.
Major Metrics	Energy consumption/ Throughput/End-to-end delay	The growing complexity of devices leads to increasing the energy consumption tremendously, which makes the critical usage of energy become essential.
QoS Requirements	Reliability/ Latency/Bit rate	It is challenging to guarantee the end-to-end QoS requirements in M2M networks, which comprise of different communication networks supported by different means of communication technologies.

mittent data transmission well in a high interference environment. These features need to be considered in the communication system design.

Figure 2.1: A simple architecture of capillary M2M communications.

2.3.1 Wireless Technologies for Capillary M2M Communications

Many factors have contributed in the rise of capillary M2M communications. For example, decreasing cost and increasing capabilities of sensors make wireless sensor networks (WSNs) very common these days. In addition, some of the existing wireless communication systems can also be utilized in M2M communications. The popular communication technologies such as UWB [37], WiFi (IEEE 802.11) [38], Bluetooth [39] and ZigBee (IEEE 802.15.4) [40], can be used for connecting the intelligent appliances and gateways in capillary M2M networks. Another important factor is the emergence of more advanced softwares. The software defined radio (SDR) [41] significantly improves the flexibility of wireless communication.

Currently, the IEEE 802.11 is one of the most popular wireless technology due to its support of high data rate, low cost and easy network deployment. However, with the rise of IoT and M2M communications, there are more and more competitors sharing the same spectrum. Since the increasing demand leads to high interference and low performance, technologies that can provide wireless local area access to support large-scale communication are in urgent need. Avoiding

the crowded 2.4 GHz ISM bands, IEEE 802.11a uses the relatively unused 5GHz, but the high carrier frequency results in a smaller effective range. Working at sub 1 GHz license-exempt bands gives the IEEE 802.11ah a significant advantage in providing large-coverage and low-power wireless connectivity to devices. Fig. 2.2 illustrates the spectrum usage of IEEE 802.11a, and the 802.11ah following the US channelization standards [42]. IEEE 802.11ah can more easily penetrate through a wall as it operates at low frequency. Its coverage range can be up to 1 km covering more devices that can be reached within one-hop transmission [43]. Besides, in many countries the spectrum below 1 GHz is not fully utilized yet; thus devices operating with IEEE 802.11ah can work at a less congested frequency band. Moreover, enhancement has been made in the design of IEEE 802.11ah PHY and MAC layers to increase the system throughput, support large number of devices with low energy consumption and to improve spectral efficiency [44]. These features make the IEEE 802.11ah a promising technology for M2M communications.

Figure 2.2: Illustration of spectrum usage in IEEE 802.11a and 802.11ah.

Visible light communication (VLC) [45] is another fast emerging technology that offers a large bandwidth and minimum electromagnetic interference. It can also achieve high data rate for short-range communications. These features of fast, safe, secure and no radio frequency radiation make the VLC another promising technology for M2M communications.

2.3.2 Standardization Activities of Capillary M2M Communications

To keep the installation simple and coverage seamless, many organizations are working to make standards for capillary M2M communications. A number of standards targeting to different protocol layers have been released. Here, we list the most popular ones as shown in Table 2.2.

Table 2.2: Standards for capillary M2M at different protocol layer.

Application Layer	COAP
Transport Layer	UDP
Network Layer	RPL
MAC Layer	IEEE 802.15.4e
PHY Layer	IEEE 802.15.4

IEEE makes enhancement for the PHY and MAC layers, which aim at enhancing the performances with respect to wireless communications, e.g., throughput and energy consumption, for M2M communications. Important features of IEEE 802.15.4 [46] include extremely low-power consumption techniques with low duty-cycle, collision avoidance using Carrier Sense Multiple Access-Collision Avoidance (CSMA-CA) scheme and reservation of guaranteed time-slots. The IEEE 802.15.4e working group (WG) caters for enhancement on the IEEE 802.15.4 MAC protocol to make it suitable for industrial requirements. Channel hopping is used to mitigate interference and multipath fading, increase the network capacity and improve reliability. Other important features of IEEE 802.15.4e are the extremely low operation costs and technological simplicity without sacrificing the flexibility. Currently IEEE 802.11b also operates in the ISM band of 2.4GHz, leading to interference from IEEE 802.15.4 [47]. Fig. 2.3 shows the spectrum usage in IEEE 802.15.4, 802.11b.

IETF makes effort to provide efficient M2M data management. IETF standardizes IPv6 over Low Power Wireless Personal Area Network (6LoWPAN) [48, 49] to bridge the gap between low-power embedded devices and IP networks. Specifically, 6LoWPAN defines how to use IPv6 in IEEE 802.15.4 radios [50]. The features of

6LoWPAN are much different from previous link technologies such as WiFi or Ethernet. In 6LoWPAN, an adaptation layer can be built up with an IP-based architecture. 6LoWPAN is characterized by small packet size, low bandwidth, star and mesh topologies and large number of devices. Specifically, as an IP-based technology for Low-power Wireless Personal Area Networks (LoWPANs), 6LoWPAN efficiently combines the IPv6 protocols and IEEE 802.15.4, so as to enable a full interoperability of low-power devices with the IP-based technologies. The two major focuses of 6LoWPAN WG are to find out how to bring IPv6 datagrams into IEEE 802.15.4 frames and perform IPv6 neighbor discovery functions in an overlapping broadcast domain network [51]. As usual, fragmentation will be done in the adaptation layer due to much larger frame size from upper layer compared to the maximum packet size in the lower layer. Header compression is designed to avoid the overhead of data and reduce energy consumption.

Figure 2.3: Illustration of spectrum usage in IEEE 802.15.4 and 802.11b.

In addition, routing over Low-Power and Lossy Links (ROLL) WG makes the necessary adaptations of IPv6 for network using IEEE 802.15.4. ROLL standardizes the IPv6 routing protocol for Low Power and Lossy Networks (RPL) [7, 51]. Low-Power and Lossy Networks (LLNs), as the typical network of M2M communications, is used to categorize the network environment where a large group of highly constrained devices are interconnected by predominantly wireless links of unpredictable quality. RPL is designed to support reliable and efficient interoperability among the devices in LLNs. With the help of RPL, the network topology is formed in an optimized way given particular objective. RPL is designed to support applications across various scenarios, such as home automation, urban environment, industrial control, and building automation.

IETF also defines the Constrained Application Protocol (CoAP) [52], which allows small, embedded and power limited devices to communicate interactively over the Internet. The CoAP is an application layer protocol which translates

to Hyper Text Transfer Protocol (HTTP) [53] to integrate with the web. The CoAP is an extremely important web protocol specialized to M2M communications. The IETF Constrained Restful environment (CoRE) WG [54] has done the major standardization work for this protocol.

2.4 Cellular M2M Communications

In cellular M2M communications, long-distance transmission technologies and licensed spectrum are used. Cellular based M2M communications have the advantage of better coverage and lower network deployment cost. However, existing cellular networks are designed for Human-to-Human (H2H) communications [55]; thus facilitating M2M communications into cellular networks creates diverse challenges. More specifically, data in M2M communications are usually small size and generated in burst transmission. In addition, applications that are applied in M2M communications are usually delay tolerant communication. These features determine that M2M communications do not require large channel bandwidth in cellular networks. In order to support massive connection of M2M devices with aforementioned features, 3GPP has standardized a new cellular technology, i.e., Narrow-Band IoT (NB-IoT) [56, 57, 58], which provides ubiquitous, stable, and robust connectivity for low-power devices so as to finally enable the IoT. Specifically, NB-IoT uses a licensed spectrum, the bandwidth requirement of which is 180kHz. NB-IoT is designed to run on existing operator networks and optimized for IoT. As a novel solution to support Low-power Wide-area (LPWA) applications, NB-IoT is compatible with existing LTE spectrum, services and networks. Two main features of NB-IoT are the wide area coverage and low power consumption. NB-IoT supports a various of M2M applications by offering very strong propagation of wireless signal and deep indoor penetration, for example, smart metering. As a cellular technology, NB-IoT requires low cost on device and deployment, and it provides long battery life and extended coverage, successfully supporting M2M communications in cellular network.

2.4.1 Architectural Enhancements of Cellular M2M Communications

Although, there are many standards defined by different organizations, the 3GPP [59] is the leading organization which standardizes the network architecture for cellular M2M communications. One of major contributions of 3GPP in standardizing cellular M2M communications is that the introduction of Machine Type Communication (MTC) [60]. For cellular M2M communications, two additional network elements in LTE-A are the MTC device (MTCD) and MTC gateway (MTCG). The MTCG is an important part of M2M communications as it is responsible for allocating bandwidth to MTCDs. It also manages the power consumption among MTCDs. A MTC user means a legal entity that uses cellular terminals for M2M communications.

Figure 2.4: A simple architecture of cellular M2M communications.

A simple architecture of cellular M2M communications is shown Fig. 2.4. The four stages of communications in MTCD-related transmissions are as follows:

Transmission between MTCD and eNB: The MTCDs use the licensed spectrum for direct communication with eNB. Therefore, MTCDs and UEs competes for this shared spectrum and it is also causes interference. Congestion is a big concern due to large number of devices joining the network. The eNB may overwhelm or even brake down due to huge amount of data transmission by MTCDs in case the cellular architecture is not well designed.

Transmission between MTCD and MTCG: The frame structure and the data rate of different MTCDs varies a lot. The MTCG usually comes with a certain computational capability, which aggregate and process the data sent by MTCDs. This prevents the eNB from becoming overwhelmed. For efficient data exchange, the MTCG can allocate bandwidth and power resource intelligently to MTCDs. The communication between MTCGs and MTCDs usually relay on the short-range communication technologies, such as ZigBee and Bluetooth.

Transmission between MTCG and eNB: The MTCG and eNB use licensed spectrum for communication. A bidirectional link is used for communication in this process. To minimize the interference during communication, Orthogonal and shared resource allocations techniques are proposed [61].

Peer-to-Peer (P2P) transmission between MTCDs: The communication technologies used in capillary M2M can also be applied to P2P communications, which will not cause interference to H2H communications.

2.4.2 Features of Cellular M2M communications in 3GPP

To avoid sophisticated hardware implementation, a promising solution is to make use of existing infrastructure of cellular networks. 3GPP has already developed the access class barring (ACB) and overload protection schemes to provide stable individual access to the eNB [62]. Some literatures also investigate other access control schemes to access the cellular channels for large number of MTCDs. 3GPP Release 10 [59] identifies the features required for cellular M2M communications, which includes low mobility, packet switching, addressing, time controlled and time tolerant, identifiers, congestion control and overload control, subscription control and security. Release 11 [63] and beyond makes improvements and enhancements on cellular M2M networks including the methods to support high-speed packet-switching, identify efficient addressing and messages routing. Some of the important features from Release 10 are as follows:

Low/No Mobility: Since most MTCDs are static, the tracking mechanism can be made less complicated to save the mobility signalling power.

Time Tolerant: Some MTC applications can tolerate latency of several mil-

liseconds to several minutes [27]. Diverse QoS performance perplexes the design of radio resource allocation schemes.

Time Controlled: The periods of reporting signal is predefined. The receiving and transmitting of MTCDs only take place within one access grant time interval (AGTI).

Small Data Transmission: Most MTCDs sporadically transmit small amount of data to eNB.

Group Based MTC: By grouping multiple MTCDs with similar QoS performance requirements, only a few physical resource blocks are required to transmit them, thereby alleviating congestion and overload problem.

2.5 Active Research Areas of M2M communications

The flourish of M2M communications results in an increasing attention from both research entities and industries. In recent years, the majority of research work on M2M communications aims at achieving an optimization of energy, total cost and reliability. As the perseverance of M2M communications increases, many research challenges arise. It is worthy to highlight the active research areas in this promising field while providing thoroughly analysis on the research challenges and valuable insights so as to finally support the design of an efficient, reliable, and robust M2M communications system. Some of the active research areas of M2M communications over the past years are summarized as follows:

Heterogeneous Network Design: M2M communications exchange data frequently between different networks such as WSNs, Wireless Personal Area Networks (WPAN) and cellular networks [64, 65]. This heterogeneity poses specific set of challenges that should be addressed for an efficient M2M communications system such as cloud computing, data fusion, data acquisition and data exchange between different networks needs to be seamless [66]. Till today, Internet is used as an intermediate network for the data exchange between different networks. For example, data collected by a node in a WSN cannot be passed directly to a ter-

terminal of a cellular network directly. It reaches to Internet via the sink node of WSN and then passed to the terminal of a cellular network. This way of data exchange reduces the M2M communications system efficiency. Today, hierarchical signal exchange is used among different networks and the heterogeneous networks should converge towards the architecture of a flat network in future to minimize the overhead of signaling [67]. Another research focus in M2M communications system is the access control schemes [64, 68]. The challenges are typically brought by the heterogeneous networks using different schemes, and it is further complexed by the coordination required. For example, scheduling is used in WSNs while access scheme of the cellular network is the random access control. These differences in the access control schemes need jointly optimized design of protocol stack for the coordination of time to access the channel among different networks.

Design of Cognitive Gateway: 1). *Networks with gateways that connect the Non-IP and IP based components:* In order to remotely monitor and control the M2M devices, a secure and efficient way is to interconnect the M2M devices with IP-based network [69]. Fig. 2.5(a) shows the interoperability of M2M devices that is provided with the multi-protocol gateway. The gateways can provide secure, scalable, energy efficient, and reliable capabilities to a telecommunication network. It can also be used to efficiently bridge telecommunication networks with local connections, such as Local Area Network (LAN), Wide Area Network (WAN), and Metropolitan Area Network (MAN). Conventional schemes that connect the Non-IP based devices with the external IP networks are not sufficient to provide robust, low-latency and reliable transmissions [70]. Therefore, traditional proxy-type gateway cannot bridge different networks to achieve a direct communication as the internal networks still use the native protocols. This problem is solved with a rather traditional solution in which every internal node is assigned an IP address [71]. This solution requires data fragmentation at the gateway due to diversity of the length of packets, especially with the utilization of the 6LoWPAN standards [72]. Further research is required on M2M gateways in order to make them cost and energy efficient and has seamless connection between internal nodes and external networks. 2). *Networks with gateways that connect capillary and cellular M2M communications:* Most of the M2M applications require data exchange between capillary and cellular M2M networks [73]. For example, smart grid needs efficient and reliable data exchange between these two networks [74, 47]. Interoperability of

Figure 2.5: Cognitive gateway architectures: (a) Gateway connecting Non-IP based component and IP based component; (b) Gateway connecting capillary and cellular M2M communications.

devices and network convergence are highly required for these heterogeneous networks. In both capillary and cellular networks the frame structures and signaling schemes varies a lot, which perplexes the complexity of the design of the heterogeneous network [75]. A cognitive gateway that is equipped to support multiple standards of wireless communication technologies can establish the communication link with the incompatible radio in an automatic manner [29]. Therefore, it is required to connect these networks in a way which is flexible and scalable for their seamless operation. Fig. 2.5(b) shows a gateway that connects capillary and cellular networks. A cognitive gateway can improve throughput, extend the convergence and help diverse devices to communicate smoothly between different networks. It can also help to provide better access control and QoS management services [73]. Therefore, it is essential for a gateway to be robust and designed carefully in order to be integrated with capillary and cellular M2M network to achieve greater energy conservation and better system scalability.

Group Control: In cellular M2M communications networks, a large number of MTC devices having diverse requirements on QoS need to get support from eNB [27, 76]. However, the structure of the current frame for LTE network is not well suited for small data transmission as it is designed for the transmissions of multimedia data with a large size with the goal of achieving high throughput. Over the past years, research on allocating radio resource to M2M communications in LTE networks has rapidly emerged [32, 75, 77, 78, 79]. Group control has been proposed in [80], which provides an efficient solution to solve the random access and efficient allocation of radio resource for MTCs in cellular M2M networks. However, further research needs to be done to explore optimal grouping which can save energy and cost.

Robust Connectivity: 1). *Dynamic and Lossy Networks:* In M2M communications, devices are usually operated in a dynamic and lossy environment. In other words, it comes with high interference and unreliable links. Moreover, due to the tremendous amount of M2M traffic, the uplink of M2M channel has a higher risk to become overloaded [67]. Both WLAN and cellular network can serve as backhauls for WSNs. However, such backhauling opportunity is not always available and it needs smart data delivery schemes among heterogeneous systems. Another research focus is the intermittent connectivity [81, 82], which leads to

uncertainty for network management and communication delays. The duration of connection and transmission delay limit the data delivery ability during a connection opportunity. In order to achieve robust connectivity, M2M communications needs solutions that can minimize the probability of collision and reduce communication delay. Thus, optimization of node discovery process, development time critical MAC protocol, and energy-efficient reliable routing protocol are the practical approaches to mitigate the effect of the dynamics and unreliability of wireless medium so as to essentially provide robust connectivity. 2). *Limited Storage*: Normally smart devices used in M2M communications are storage-constrained with a small buffer [83]. Thus, packets may be easily dropped from devices. Meanwhile, in order to save power, some smart devices may break their wireless link connectivity when they do not need to transmit data. Therefore, initiating communication with the sensor nodes at an arbitrary time is difficult. With high probability, data will be lost if the connectivity is not built up on time. Lots of research work has been done to accelerate the move toward efficient storing capabilities of M2M nodes [84, 85, 86]. One promising solution is to equip the smart node with the ability of data fusion and aggregation [87], which means pre-process the data in the smart node or smart sensor. In this way, nodes only need to store meaningful data. However, on the other hand, such ability might consume extra power.

Mitigating Co-channel Interference: H2H and M2M communications share the time and frequency resources, which results in serious co-channel interference, degrading the QoS [88]. This also brings the demand for efficient use of limited radio resources due to increasing number of M2M devices. Therefore, spectrum management becomes a major issue in realizing M2M communications [89, 90]. Cognitive radio technology [91, 92, 93] that can reuse the licensed spectrum holds a promising solution for this problem. However, it may still bring interference to each other. Another important research topic is interference mitigation schemes to avoid or minimize the co-channel interference [94]. Besides, the unreliability of wireless channels adds more difficulty in minimizing the co-channel interference. Channel fading, shadowing and multipath fading, results in large uncertainties of unreliability of wireless channels. In addition, conventional routing protocols, such as AODV[95], DSR [96], DSDV [97], are typically designed without considering the uncertainties of the channel conditions. The emergence of M2M communications attract the research attention on the dynamic and lossy wireless medium. There-

fore, suits of routing protocols with the consideration of the unreliable wireless links, such as Hydro [98], RPL [7] are proposed.

Energy Efficiency Support: M2M communications have severe constrains in terms of energy, memory and computation ability. Smart devices are usually battery-powered and have to run complex computational processes. The computational complexity perplexes the energy depletion problem, and the replacement of tremendous number of batteries needs expensive logistics. Decreasing the expense of power consumption becomes a major headache in M2M communications. Several energy saving schemes have been proposed for M2M communications, e.g., compressing the transmission messages to save the transmission power [99], energy-efficient routing protocols to conserve the energy by reducing transmission power [100, 101], sleep mode to minimize energy consumption [68].

2.6 Summary

The effectiveness of M2M communications requires ubiquitous connectivity among M2M devices and efficient transmission of meaningful information between intelligent devices and applications. In the future, M2M applications will become worldwide and an essential part of daily life. However, to make these promises a reality, M2M communications need to successfully deal with various technological challenges. This chapter has presented a comprehensive study on M2M communications with a focus on two categories of M2M, i.e., capillary M2M communications and cellular M2M communications. In addition, the standardization activities and active research areas of M2M communications are also introduced. Although this chapter introduces cellular M2M communications so as to provide an overall picture of M2M communications, further study is conducted with a focus on capillary M2M communications in the following chapters. Specifically, a suit of routing protocols will be proposed to support energy-efficient and reliable routing for capillary M2M in this thesis.

Chapter 3

A Comprehensive Study on IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL)

3.1 Introduction

In the previous chapter, we presented an overview of M2M communication in terms of capillary M2M communications, cellular M2M communications, and the active research areas. It has been shown that a suit of protocol stacks supporting capillary M2M communications are defined by standardization groups, i.e., IETF and IEEE. One of the milestone protocols in the stack is the IPv6 over Low power Wireless Personal Area Network (6LoWPAN) [72, 102], which bridges the gap between the low-power devices with the IP world. With the specifications of 6LoWPAN, the Routing Over Low-power and Lossy network (ROLL) working group standardizes the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) [7]. RPL, as a prominent candidate to support LLNs applications, is regarded as the most promising routing solution to support M2M communications. As a proactive IPv6 distance vector protocol, RPL is designed for large networks consisting of multitude of nodes operating in dynamic and lossy links (referred as LLNs) environments. RPL, which have the capabilities to fulfil the diverse requirements of various LLNs

application, have gained considerable attentions with the emergency of IPv6-based networks. As the protocol stack integration of capillary M2M deepens, the importance of RPL increases. Therefore, a comprehensive survey of RPL is crucial to pave the way for researchers to understand and contribute in the relevant research area of RPL. Essentially, the main contributions of this chapter are five-fold.

- Firstly, with a thorough analysis on the inherent constraints in LLNs, this chapter reviews the routing protocols that are designed for LLNs. The characteristics of several main LLNs routing protocols are clearly demonstrated for a comparative purpose with RPL.
- Secondly, this chapter presents a detailed overview of the RPL in terms of its routing specifications with focus on the routing methodologies of the attractive features, including the fast topology construction, low maintenance overhead, robustness, scalability, great flexibility.
- Thirdly, by exploiting the trust-able, practicable, and flexible features of NS-3, this chapter proposes and implements a lightweight RPL module, which essentially unveils the ambiguities regarding the conformity and validation of RPL, and reduces their complexity in the implementation design. This approach also benefits future research on RPL.
- Fourthly, the feasibility of RPL is clearly demonstrated through an experimental performance evaluation based on the implemented framework in NS-3. In order to understand the impact of the routing parameters and the lossy rate of wireless medium on the performance of RPL, extensive simulation studies are conducted with focus on the construction of topology and support for generic traffic paradigms, i.e., many-to-one, one-to-one, and one-to-many.
- Last but not the least, the main research focus on the routing support abiding by the severe and inherent constraints of LLNs are critiqued with analysis on the limitation of RPL. In addition, valuable insights are presented with suggestion of enabling techniques to improve the performance and further increase the flexibility of RPL so as to encourage a wider adoption of it.

The reminder of this chapter is organized as follows. Section 3.2 introduces the existing protocols designed for LLNs and suits of LLNs applications that adopt

RPL as the routing support. Section 3.3 presents a detailed overview of RPL protocol in terms of the routing specification. Then, the design and implementation of RPL module for NS-3 are introduced in section 3.4. Section 3.5 evaluates the performance of RPL with extensive simulation studies. The research focus on efficient routing support for LLNs are presented in Section 3.6. Finally, this chapter is summarized in Section 3.7.

3.2 An Overview of Low-Power and Lossy Networks (LLNs)

LLNs comprise of thousands of nodes which suffer from severe resource constraints (limited energy and memory resource) and operate in harsh environments. The dynamic and lossy nature of wireless links are usually suffer from the co-channel interference, channel fading, and a harsh environment (heat, dust, and moisture). In addition, the self-organizing property of LLNs requires the routing protocol to have the capability of auto configuration and self-healing with extremely low propagation and maintenance cost. Traditional routing protocols which are not able to support very dynamic topologies and are subject to link failure, could not address these issues and face difficulties in fulfilling the requirements of LLNs [8]. LLNs calls for a scalable, stable, reliable, and energy-efficient routing support. There are various quality-of-service (QoS) requirements for LLNs application, since the objective of LLNs applications are diverse. As a most promising candidate to support various LLN applications, RPL defines the general topology formation and maintenance mechanism with specialized and optional features. In this section, the most relevant routing protocols for LLNs are reviewed and compared. This section also introduces popular LLNs applications.

3.2.1 Routing Protocols for LLNs

In recent years, LLNs applications are gaining an ever increasing interest from both research entities and industries, which calls for an universal solution to enable the ubiquitous connectivity for tremendous numbers of low-cost and low-power em-

bedded devices. The key insight underlying this section is to survey and discuss a comprehensive set of routing protocols proposed for LLNs by describing the features of the main routing protocols and providing a comparison on their characteristics. The existing LLNs routing protocols can be categorized into: proactive routing, reactive (on-demand) routing and geographic routing. Based on Ad hoc On-demand Distance Vector Routing (AODV) [95], a number of reactive routing protocols have been proposed to provide the routing support for LLNs, including AODVjr [103], AODVbis [104], LoWPAN-AODV [105], LOAD(ng) [106, 107], TinyAODV [108], NST-AODV [109]. These routing protocols make adaptation/simplification on AODV to be low-footprint and suitable for the resource constraint and dynamic network environment. Another set of proactive routing protocols are also proposed, which includes, but not limit to, the Collection Tree Protocol (CTP) [110], Hydro [98], ZigBee Cluster Tree [111], RPL [7]. The third category is the geographic routing, in which the location or coordinates are known by each node or partial nodes either as a priori knowledge or through a self-configuration localization scheme. Beacon Vector Routing (BVR) [112] is one of the geographic routing protocol that is designed to support the routing for LLNs. The following of this section provides more details about the LLNs routing protocols. In addition, Table 3.2 presents a summary of the main features of LLNs routing protocols.

IEEE 802.15.4/ZigBee Cluster-Tree protocol [113] forms the network as a star-based topology. In particular, the synchronization in the ZigBee Cluster-Tree protocol is achieved through periodic beacon frames. In addition, multi-hop networking is supported by the Cluster-Tree protocol. As a hierarchical routing protocol, there is a Zigbee coordinator governing each cluster. However, collision easily takes place if the beacons are sent in an unorganized fashion. Furthermore, a time division approach is proposed in [114], which addresses the beacon collision problem. However, as a star-based networking, the Cluster-Tree protocol can not provide the scalability support as the number of nodes are restricted by limited coverage of the Zigbee coordinator. This is particularly unsuitable for LLNs which are commonly large-scale networks.

Lightweight On-demand Ad hoc Distance-vector Routing Protocol-Next Generation (LOADng) [107] is a reactive routing protocol derived from AODV for LLNs. The basic operation of LOADng is similar to that of AODV which makes lots of

simplifications. During the route discovery stage, route request (RREQ) messages are distributed throughout the network. If the destination receives the RREQ message, it replies with a route-reply (RREP) message through the reverse path. LOADng supports more traffic patterns, such as point-to-point (P2P), point-to-multipoint (P2MP), and multipoint-to-point (MP2P). But it shows disadvantage in terms of much overhead for MP2P traffic pattern compared with RPL [115]. The main reason behind this is that LOADng disseminates the control messages throughout the network for the best route discovery given a particular objective. The performance comparison and analysis of LOADng with RPL have been done with different traffic patterns [116, 117].

TinyAODV [108] is a simplified version of AODV, that has been implemented in the operating system of the MICAz's TinyOS. Similar to the LOADng, only the destination node generates the RREP messages. TinyAODV is implemented with a link failure detection mechanism, which is by default disabled. To target static networks, TinyAODV discards the undelivered data packets resulting from the link break. In addition, TinyAODV does not support the local repair, and it adopts hop counts as its routing metric. TinyAODV Release 2 supports multipoint-to-point (MP2P) traffic in which only the sink node can be the destination node. While TinyAODV Release 3 successfully enables the communication between two arbitrary nodes in the networks [109].

Based on the priori work in [118], Hydro [98], a hybrid routing protocol that integrates the local agility with the centralized control is proposed to provide a robust data collection and efficient point-to-point communication support. In Hydro, a distributed algorithm is used to form a direct-acyclic-graph to route the data from nodes in-network to the border routers. Specifically, in order to allow the border routers to have a knowledge of the global topology, nodes periodically send the topology reports to the border routers. In this way, a basic point-to-point routing is enabled with the border router, which works as an intermediary to relay packet up to the destination [98].

Beacon Vector Routing (BVR) [112] is a geographical routing protocol, proposed for LLNs. With the location knowledge of a set of landmarks, each node in the network obtains virtual coordinates. The packet is delivered to the destination in a greedy manner. Specifically, nodes chooses the node closest to the destination

as their next hop to deliver data packet towards the destination. Geographic routing has the advantages of a low overhead and the scalability, but it does not take the lossy nature of wireless links into consideration when it chooses the next hop. Consequently, geographic routing cannot cope well with the lossy wireless medium and provide reliable data delivery support for LLNs. In addition, some geographic routing protocols require nodes to maintain many states and have to exchange the one-hop or even two hop neighbour table periodically [119], which is very costly for a resource constrained network.

CTP [120], as a proactive gradient-based routing protocol, aims to support effective data collection. The collection-oriented, also known as multipoint-to-point/many-to-one traffic, is the predominant communication paradigm in LLNs applications. As a tree-based collection protocol, CTP uses the expected transmissions (ETX) as the routing metric for next-hop selection. Specifically, an accurate link estimator is developed in CTP, which uses the data and control traffic to notify the link estimations. In addition, based on the link lossy rate, an additive cost is computed in CTP. In CTP, a node chooses the node that provides the lowest accumulated ETX value as its next hop. It has been mentioned that the routing loops and packet duplication are the two problems that can occur in CTP [110]. RPL is an extension of CTP, which inherits and improves the data path validation and the adaptive beaconing from CTP.

The existing routing protocols for LLNs either optimize for data-collection traffic or exclusively focus on the point-to-point traffic paradigm in homogeneous networks. Motivated by the fact that the wide variety of LLNs applications are becoming more general, an ever growing support for a hybrid of these traffic paradigms in heterogeneous networks are expected. RPL is considered as the most promising routing protocol for LLNs because of its comprehensive features and great flexibility. Specifically, as a proactive routing protocol, RPL supports various traffic patterns, including multipoint-to-point (MP2P), point-to-point (P2P), and point-to-multipoint (P2MP). Thus, it avoids the usage of RREQ message and support the delivery of data with a low overhead. In addition, RPL provides the benefit of both flat and hierarchical topologies with a local and global repair support so as to be resilient to the dynamics of LLNs.

Table 3.1: Comparison of routing protocols for LLNs.

Protocol Name	Type	Topology	Algorithm	Local repair	Mobility	Scalability	Memory Usage	Energy Usage	Supported traffic	IPv6 Support
TinyAODV [121]	Reactive	Flat	Distance Vector	No	Static	High	Low	Low	MP2P, P2P	Yes
NST-AODV[109]	Reactive	Flat	Distance Vector	Yes	Static	High	Low	Low	MP2P, P2P	Yes
Hydro [98]	Reactive	Hierarchical /Flat	Source routing	No	Static	High	Medium	High	MP2P, P2P, P2MP	Yes
LOADng [117]	Reactive	Flat	Distance Vector	No	Static /Mobile	High	Low	Low	P2P	Yes
CTP [120]	Proactive	Hierarchical	Source routing	Yes	Static	High	Low	Low	MP2P, P2MP	No
RPL [7]	Proactive	Hierarchical /Flat	Distance Vector /Source routing	Yes	Static /Mobile	High	Low	Low	MP2P, P2P, P2MP	Yes
P2P-RPL [122]	Reactive	Flat	Distance Vector /Source routing	Yes	Static /Mobile	High	High	Low	P2P	Yes
ZigBee Cluster Tree [111]	Proactive	Hierarchical	Link State	No	Static	High	High	High	MP2P	No
BVR [112]	Geographical	Flat		No	No	Static	High	High	High	P2P

Other attractive features of RPL includes, but not limit to, a combination of source and distance-vector routing paradigms, the design of various routing metrics, and IPv6 address for large scale networks. These features turn RPL to be widely used for supporting LLNs applications across various scenarios. A detailed introduction of routing specifications of RPL are presented in the next section.

3.2.2 Applications of LLNs

As mentioned before, RPL is viewed as a standardized solution to efficiently provide reliable interoperability among arbitrary devices in different domains of IoTs and M2M networks. However, the requirements of different LLNs applications on routing vary from applications to applications, thus, the specific requirement on routing should be addressed in accordance to a particular application. The routing requirements on RPL to support different LLNs applications have been addressed in detail in existing literature, ranging from the urban networks [RFC5548], industrial control, [RFC5673], home automation [RFC5826], and building automation [RFC5867]. In this section, two popular LLNs applications with RPL are introduced.

Home Automation and Building Control

With the pervasiveness of wireless personal area networks (WPANs) and body sensor networks (BSNs), the home automation applications, e.g., healthcare, illuminance control with the window blind, and automated lighting, have attracted considerable interests recently. Specifically, a building management system consists of interrelated subsystem with different functions, such as HAVC system, lighting system and surveillance system. Moreover, it comprises of a large number of the actuators, controllers, sensors, and interactive devices, such as smart phones, laptops, and smart watch, etc. However, owing to the limited coverage of the radio, it is almost impossible to attain a direct communication in the large scale network. Furthermore, in the home and building environment, wireless signals typically suffer from a high interference and serious path loss due to the cement building structure and the dense usages of spectrum resource. Therefore, reliable connectivity is of paramount importance for efficient resource utilization with energy conservation

and a tighter control in home automation and building control.

The home and building network comprises of a large number of energy-constraint devices and unreliable wireless channels, which is considered as LLNs. Furthermore, RPL, as it is considered as the prominent candidate to support the home and building network, needs to fulfil important requirements [9, 10]. The requirements include mobility, manageability, stability, security, short convergence time, low configuration cost and flexible route selection in accordance to the application. Fig. 3.1 shows a routing topology that we define for the home and building networks with RPL.

Figure 3.1: Architecture of RPL routing support for building automation system.

Advanced Metering Infrastructure

Advanced Metering Infrastructure (AMI) systems [123] is an integrated system that is composed of the communication networks, smart meters, and the data management systems. Meanwhile, AMI supports the configuration, measurement, and the control, distribution and consumption of resources, e.g., gas, energy, and water through a two-way communication. In AMI networks, there are large number of endpoints, such as smart meters and home area network elements, etc., which are inter-connected by a wired or wireless communication network. In par-

particular, there is a back-haul network that provides the connectivity for software management in the utility control office. In addition, the field area router, which bridges the AMI system with the external networks, is integrated with multiple wireless interfaces, such as WiFi [124], Ethernet [125], IEEE 802.11ah [126], and IEEE 802.15.4 [127], and WiMax [128]. The AMI network calls for a large scale, self-configuration, efficient route selection, a low maintenance cost, multicast and anycast routing support. The applicability of RPL to AMI deployment has been defined in RFC work [123]. Specifically, RPL is designed with the energy-efficient routing mechanisms and energy aware metrics. With the capability to support various metrics and constraints, RPL can efficiently support heterogeneous networks, in which the nodes and the links are equipped with different characteristics. The challenges for the efficient data collection in AMI with RPL are addressed by [129], [130]. Fig. 3.2 shows the architecture of AMI with RPL routing support.

Figure 3.2: Architecture of RPL routing support for smart meter system.

3.3 An Overview of IPv6 Routing Protocol for Low Power and Lossy Networks (RPL)

RPL is an IPv6-based distance-vector proactive routing protocol and it provides the support for both multicast and unicast data transfer. In RPL, the topology of a network is constructed as a destination-oriented directed acyclic graphs (DODAG), in which the nodes form a tree without any loop. In particular, the DODAG is rooted at one sink node, called the root, which serves as a transit point to bridge the DAG with IPv6 network. Fig. 3.3 illustrates a RPL instance consisting of one DAG associated with two DODAGs, which are connected via a backbone link through their roots. In Fig.3.3, the direction of the arrow indicates the direction from the child node to its preferred parent. Importantly, RPL has four instance values, i.e., instance ID, DODAG ID, DODAG version number and rank. In particular, a DODAG topology is maintained by these four instances values. Furthermore, any node in RPL can be uniquely identified with these four instances values. The RPL instance is used to identify the network consisting of one or more DODAGs sharing the same service type (objective function), while the DODAG ID is used to describe nodes that are connected with the same root. The DODAG version number is updated as the topology of DODAG updates, so as to keep the node's state up-to-date with the current topology. Rank, which represents the relative distance towards the root, is a very important parameter of a node in a DODAG. Specifically, a node with a small rank indicate that it is close to the root. In RPL, the downward route is defined as the route from the root to a node, whereas the upwards is defined as the route from a node to the root. Table 3.2 summarizes other main features of RPL. The rest of this section introduces the important routing specifications defined for RPL, including the construction of the DODAG, the formation of the downward routes, the objective functions, and the trickle algorithm.

3.3.1 DODAG Construction/Upward Routes

The DODAG is formed in a way to optimize the routing support for the multipoint-to-point (MP2P) traffic paradigm. In RPL, a node has two types of status, i.e.,

Figure 3.3: DODAG topology constructed by RPL.

grounded node or floating node. The grounded nodes refer to the nodes which have already joined the DODAG, while the floating nodes refer to the nodes that have not yet joined any DODAG. When a floating node is trying to join the network, it multicasts DODAG Information Solicitation (DIS) message to a specific IPv6 multicast group. Upon receiving the DIS from its neighbour node, a grounded node unicasts back a DODAG Information Object (DIO) as the reply. The DIO message plays a key role in the formation of DODAG as it contains the indispensable configuration parameters, including RPL instance, objective function, trickle timers and other necessary information. According to the objective function enclosed in the DIO, a node will select one neighbour node as preferred parent and monotonically increase the rank based on preferred parent's rank. The value of the rank indicates the hop distance from a node to the root. In a DODAG, the root is with rank one, and all preferred parent has lower rank than its children. Only preferred parent helps to forward the packet coming from its children nodes. In this way, the delivery of packet will be towards the root.

3.3.2 Downward Routes

Some LLNs applications require a P2P (node to node) or P2MP (root to many nodes) routing support. The Destination Advertisement Object (DAO) is used to enable the downwards route for the DODAG. In RPL, two Mode of Operations

Table 3.2: Main features of RPL routing protocol.

Feature	Description
Loop avoidance and detection	In RPL, any node's rank has to be higher than its parent, resulting in the acyclic nature of the DODAG. Additionally, both local and global recovery mechanisms to detect the loop and recovery the topology.
Self-configuration	With IPv6 Neighbor Discovery mechanisms, the paths of network are typically dynamically discovered.
Communication paradigms	Multipoint-to-point (MP2P); Point-to-point (P2P); Point-to-multipoint (P2MP);
Target Network	Low-Power and Lossy Networks (LLNs); 6LowPAN networks; other IPv6 networks
Identifiers	RPL Instance ID, DODAG ID, DODAG Version, Rank
Security mode	Unsecured, Pre-installed, Authenticated
Mode of Operation (MOP)	MOP (0), no downward routes; MOP (1), non-storing; MOP (2), storing; and MOP(3), storing with multicast.

(MOPs) are used to define the maintenance mechanism of the downward routes, storing mode and non-storing mode. In the storing mode, the DAO message is sent to the parent node with multicast or unicast. The parent node will store the routing information carried by the DAO message before it sends a new DAO message with the aggregate routing information to its parent node. In non-storing mode, each node sends the DAO to the root, which is the only destination node. The intermediate nodes do not recode the reachability information in the DAO message, but insert their own addresses into the route list in the received DAO before forwarding the DAO message to its parent node. In this way, the root node can have a routing map of all nodes in the DODAG, so that RPL is able to provide the routing support for P2MP traffic patterns. RPL needs to pre-establish routes and can only route along the existing DAG. For P2P communication, the source node has to send the packet upwards until it reaches the ancestor node of the destination node. Then the common ancestor node will deliver the packet downwards towards the destination node. In the non-storing mode of RPL, the common ancestor has to be the root.

3.3.3 Objective Functions

In RPL, the selection of preferred parent is in accordance with the objective function (OF). The way of choosing objective functions has a significant impact on the network topology and performance of RPL [131]. In order to support diverse applications of LLNs, both node level and link level metrics are designed for RPL [132]. A detail introduction of which is given in Table. 3.3. The link and node metrics are usually (but not necessarily) additive along a path to the DODAG root. The routing metrics for LLNs can be categorized as 1) link versus node metrics, 2) qualitative versus quantitative, 3) dynamic versus status [132]. In RPL, the OF, such as the Objective Function Zero (OF0) [133] or the Minimum Rank with Hysteresis Objective Function (MRHOF) [134], etc., are advertised by DIO with a metric container. Then, a node chooses the best next hop as the preferred parent based on the qualitative value given the particular objective function.

Table 3.3: Current Routing Metric/Constraint Objects in LLNs.

Node Object	Link Object
<i>Node State and Attributes Object</i> , e.g., node workload, CPU, memory, etc.	<i>Throughput Object</i> , whose unit is bytes per second.
<i>Node Energy Object</i> , which estimates residue energy remaining	<i>Latency</i> , which can be used either a constraint or a metric.
<i>Hop Count Object</i> , which can be used either a constraint or a metric	<i>Link Reliability</i> , e.g., Link quality level reliability (LQL), Expected transmission count (ETX), etc.

3.3.4 Trickle Algorithm

In LLNs, there are two conflicting goals. First, the network topology should be constructed efficiently and quickly, which requires an effective forwarder selection scheme and a fast prorogation of routing information. Second, in order to conserve energy, the network topology should be maintained with minimum cost under the dynamic nature of wireless networks, such as the unpredictable and lossy wireless medium, and node states (alive or dead). However, these goals are further complicated by the low-power radios, which leads to transient disconnection and high loss, resulting in control packets getting lost. In order to address these challenges,

RPL adopts the trickle algorithm [135] to attain a rapid propagation and low cost for the maintenance of network topology. The goal of the trickle algorithm is to propagate the routing information quickly and maintain the network with a low cost. A local “*polite gossip*” is used by the trickle algorithm to self-regulate for exchanging packets with neighbor nodes. There are two possible states of the neighbor nodes. One state is that all of the nodes’ routing information is up to date. Another one is that there exists nodes with “old” information. Because the trickle algorithm is independent of the network density, it is scalable to different network topology.

With the trickle algorithm, RPL can control the transmission rate of the DIO messages. Three key parameters are used to define the interval of broadcasting control messages [135]; the minimum time interval size I_{min} , the maximum time interval size I_{max} , and the redundancy constant k . The trickle algorithm maintains three variables: 1) the current interval size, i.e., I , 2) t , which is a time within I and $I/2$, 3) a counter, c . These variables are updated with the dynamic of the network topology [136]. The broadcast of control message is scheduled after the duration of t . When timer t is out, the node will check whether the redundancy counter c is smaller than the redundancy constant k . If it is, the node will broadcast the control message, i.e., DIO message. If it is not, the node does not broadcast any control message. At time I , the node will reschedule the trickle timer by doubling the time interval I , and reselect the timer t , while keeping this time interval I smaller than I_{max} . There are two important definitions; consistent and inconsistent. A control message from a sender with a lower rank value that does not cause any changes to the recipient’s parent set, preferred parent, or rank is considered as consistent. Otherwise, the control message is inconsistent. When a node hears a transmission that is “consistent”, the redundancy counter c will be monotonically increased. Otherwise, c will be set to 0. Finally, the time interval increases during the network convergence time. When the interval I reaches I_{max} , the transmission rate of control messages, f_c , can be calculated as

$$f_c \approx \frac{1}{I_{min} 2^{I_{max}}}. \quad (3.1)$$

With the help of trickle algorithm, RPL can construct the DODAG at a high

speed with low routing overhead. In this way, RPL is able to exchange information in a simple, reliable and scalable manner in lossy environment with low energy consumption.

3.4 Implementation Framework of RPL in NS-3

It is a significant and difficult task to design and implement routing protocols in a network simulator. Before implementing routing protocols, we have conducted a systematic study on the architectural design of NS-3 and dynamic parameters of the network environment, such as the lossy rate of wireless medium, energy, channel contention and channel fading. Additionally, we carefully design and choose the network performance metrics because the evaluation of network performance requires high accuracy and representativeness of the simulation results. In this section, we describe the implementation design of proposed lightweight version of RPL. The decisions made during the designing of modules for RPL are also introduced. This section introduces the designing framework and choices made during the implementation of RPL. A through description of the DODAG construction is also provided.

Fig. 3.4 shows the class diagrams of RPL module, which complies with the existing IPv6 structure in NS-3. The main class of RPL routing protocol is defined as the namespace of “RPL”, which incorporates five major components, as shown in Fig. 3.4. All the control messages of RPL, such as DIS, DIO, DAO and DAO-ACK are defined in the class of *Control Message Header*. Furthermore, RPL requires the metric instance for the path selection. Two objective functions, i.e., OF0 [137] and MRHOF [138] are implemented in our design. With OF0, a node selects the shortest path in terms of hop counts whereas with MRHOF, a node selects the path that gives the minimum aggregate expect number of transmission (ETX). Moreover, The *Neighbor List* is defined to hold the information of neighbor nodes, e.g., rank and cost. Meanwhile, the *Neighbor List* also contains the list of parent set of a node. If a node detects there is a failure on its preferred parent, it will switch its preferred parent to another node in its parent set. The *Routing Table* is implemented to store the routing information and it is accessed with the *Routing Entry*. The *Trickle Algorithm* is designed to control the frequency of the

propagation and maintenance of the routes.

Figure 3.4: Simplified RPL class diagrams integrated with NS3 routing module.

Fig. 3.5 shows the routing support with our proposed RPL module at the source node in our implementation. The routing design is subject to different traffic patterns including MP2P, P2P and P2MP traffic. In our proposed RPL module, a source node first distinguish the traffic pattern according the the destination address. There are three cases in accordance to three types of traffic. In the first case, the source node forwards packets through its preferred parent if the destination node is the root. The address of root is globe information, which is enclosed in the DIO message. This case is identified as MP2P traffic. In the second case, the source node itself is the root, so that the packets are forwarded to the destination through a list of relays according to a source routing list. The source routing here is enable with the help of DAO message. The third case is the P2P traffic that requires an intermediary to support P2P routing. Specifically, in non-storing mode, packets will be first forwarded to the root through the preferred parent. Then, root attaches a header of source routing list with the packet. The subsequent forwarders relays the packets according to the source routing header. In storing mode, the first common ancestor of the source and destination node acts as the the intermediary to support the P2P communication.

Figure 3.5: Route processing diagram at the source node subject to different traffic patterns.

3.4.1 Formation of Upwards Routes

The packet format of DIS and DIO is shown in Fig. 3.6 and Fig. 3.7, respectively. In this work, each field in a packet format of control messages are designed by adopting the standardization defined in [RFC6550]. So we skip the introduction of each field and illustrate the flow of control messages.

Flag	Reserved	Option
------	----------	--------

Figure 3.6: Packet format of the DODAG Information Solicitation.

Four methods are designed namely, *SendDIS*, *RecvDIS*, *SendDIO* and *RecvDIO* for the formation of the upwards route in the DODAG. *SendDIS* defines a floating node to multicast the DIS message periodically. DIS message is used to solicit the DIO message. The joining of the DODAG is initialized with the help of DIS. The *RecvDIS* method is called once a node receives a DIS message, and it triggers the *SendDIO* to unicast a DIO if the node is a grounded node. *SendDIO* can either unicast or multicast DIO. After the unicast of DIO, the trickle timer activates the

route propagation and maintenance through adjusting the frequency of broadcast. Upon receiving the DIO from the grounded node, a node stops sending the DIS messages and cancels the DIS timer, which is defined in *RecvDIO*. Meanwhile, the source of DIS message updates its neighbor list and triggers a reset for trickle timer. Upon receiving unicast DIO message, a floating node will update its status, chooses the sender as its preferred parent and compute its rank. If the node is a grounded node, it will compare its cost towards root based on the objective function advertised in DIO message. The selection process of preferred parent is completed by the method of *RecvDIO*.

RPLInstanceID	MOP	DTSN	Rank	G	O	Cost
Version Number	Lifetime	RedundantK	Imin	Idouble	Reserved	
DODAG_ID						

Figure 3.7: Packet format of the DODAG Information Object.

3.4.2 Formation of Downwards Routes

The packet format of DAO is shown in Fig. 3.8. The implementation choice we made is to consider a node's preferred parent as the only one DAO parent of itself, so that the DAO message is sent in a unicast manner. The DAO message can be either entirely disabled or operated to support the storing/non-storing mode. We design a *GetStoring* method to configure the MOP of RPL in a globe manner, which means that all of the nodes in one DODAG are required to be operated in the same mode. The *SendDAO* and *RecvDAO* are defined to enable the downwards routing. It is worth to highlight that the *SendDAO* is triggered by the DIO message with the DTSN filed. In Storing mode, a node sends the DAO message to its DAO parent, which is its preferred parent in our design, with its downwards route information. A node stores the routing table entries of the routes, which are learnt from the DAO message. The Source Routing is used in the Non-Storing mode, in which the DAO is piggybacked with a vector of IP address. A node sends a DAO message to the root. The intermediate nodes do not store the downward route but add the address of itself to the DAO message before forwarding the message. The roots

store the source routing list with the address of the source node as the routing table entry. The *SendDAOAck* and *RecvDAOAck* are also applied.

RPLInstanceID	Flag	DAOSequence	Reserved
DODAG_ID			

Figure 3.8: Packet format of the Destination Advertisement Object.

3.4.3 Integration of Trickle Algorithm

Three methods are implemented in our RPL module to support the trickle timer, and they are *TrickleAlgorithm*, *ReScheduleTrickle*, and the *CounterReset*. As mentioned earlier, the DIO message advertises three configuration parameters of the trickle algorithm, i.e., I_{min} , $I_{doubling}$, and c . The I_{max} can be got through

$$I_{max} = \log_2 \frac{I_{doubling}}{I_{min}}, \quad (3.2)$$

The method of *TrickleAlgorithm* is used to schedule the control message transmission, such as the DIO in RPL, after time t . *TrickleAlgorithm* computes the time t as $I_{doubling}2^{I_{temp}}$, where the variable I_{temp} is randomly generated from the range of $[I/2, I]$. In addition, this method also schedules to call *ReScheduleTrickle* once I is reached. The method of *ReScheduleTrickle* checks the current interval I and increases the value of I to $2I$ or I_{max} , whichever is smaller, then calls *TrickleAlgorithm* method with the latest value of I . As mentioned earlier, once *RecvDIO* examines an inconsistent message, the counter c is reset to 0, so the *CounterReset* is called in this case..

3.5 Simulation Study

In order to evaluate the routing performance of RPL routing protocol under different network settings, we conduct extensive simulation studies in NS-3 simulator and evaluate the routing performance with a focus on two main categories: the

routing performance for the network initialization phase and the routing performance for different traffic patterns, including MP2P, P2P, and P2MP.

3.5.1 Simulation Setup

In this study, a unit disk graph model (UDGM) is used to model the lossy environment with different degrees of unreliability [139]. In this model, the packet loss probability is proportional to the square of Euclidean distance between two nodes. In addition, the packet reception probability at the edge of transmission range is also configurable. Data traffic starts once the network finishes initialization phase. The IEEE 802.11 standard is used to evaluate the performance of LLNs protocols [115, 140]. Other parameters used for simulations are shown in Table 5.1. Each scenario is run for 10 iterations and the results are the average values of those iterations. In this study, we study different positioning strategies of the root. The position of the root is very important, since it strongly affects the way that the network topology is created in RPL. We compare the routing performance of RPL using two objective functions, i.e., OF0 and MRHOF. In particular, we study the routing performance for the network initialization phase with different network sizes, and routing performance with different traffic patterns under link success rates.

Table 3.4: Simulation parameters to evaluate the performance of RPL.

Description	Value
Communication range	35m
Simulation duration	900s
Packet size	512 bytes
Traffic rate	1 pkt/s CBR flow
Retransmission limit	4
Traffic duration	200m
The minimum time interval size (I_{min})	50ms
Redundancy constant (c)	∞
Node deployment	random

3.5.2 Performance Metrics

In this study, the following metrics are adopted to evaluate the performance of RPL.

- *Network convergence time* represents the time at which the DAG is completely constructed and all the nodes have joined the network.
- *Average/Maximum rank* refers to the average or maximum hop count of nodes from the root in a DODAG.
- *Packet delivery ratio* refers to the ratio of the number of packets successfully delivered to the destination nodes and the number of packets generated by the source nodes. This metric illustrates the routing reliability.
- *Routing control overhead* refers to the total number/bytes of control messages used to build up network topology and support data delivery in the network during the simulation.
- *Average hop count* refers to the average number of hops for the discovered route between the source nodes and the destination nodes.
- *Average end-to-end delay* includes all possible delays during data transmission due to transmission time, retransmission caused by collision and queuing time.

3.5.3 Performance Evaluation for Topology Initialization

This section is to evaluate the performance of DODAG initialization in RPL. In this study, the MRHOF is adopted as the objective function in RPL. The link success rate at the edge is configured to be 0.5. The network size varies while node densities are intact. The number of nodes varies from 80, 120, 160, 200, to 240, and are uniformly distributed in the network areas $200\text{m} \times 200\text{m}$, $245\text{m} \times 245\text{m}$, $283\text{m} \times 283\text{m}$, $316\text{m} \times 316\text{m}$, $346\text{m} \times 346\text{m}$ respectively. The impact of different strategies of the root position and the tickle parameters with different network sizes are studied. In Fig .3.9(a), the “conner-avg-rank” refers to the average rank of the network, where the root is placed in the corner, whereas the “center-max-rank”

refer to the maximum rank of the network, when the root is placed in the center. The “conner-I8” refers to the routing performance of RPL with I_{max} as 8, when the root is placed in the corner. The “center-I12” refers to the routing performance of RPL with I_{max} as 12, when the root is placed in the center.

Fig. 3.9(a) compares the average and maximum rank of the network with varying network sizes under different root positions. It shows that both the average rank and maximum rank increases as the network size increases. Besides, when “center-avg-rank” compares with “conner-avg-rank”, the network with root placed at center has much less average number of hops than the network with root placed at conner. The rank value is important for a node in RPL as it indicates the hop distance of a node to the root. The maximum rank represents the size of a DODAG. The size of a DODAG is smaller if the root is in the center of the network than the root in the conner, which can be observed by comparing “center-max-rank” with “conner-max-rank”.

Fig. 3.9(b) shows that when the network size increases the number of routing control overhead also increases. The routing control overhead is composed of two parts, first, the overhead used for propagating and maintaining the upwards routes, i.e., DAO, and second, the downwards routes in a DODAG, i.e., DIO. Further observation reveals that the control overhead with center-root position is much lower than that with conner-root position. The reason behind this is that the overhead used to support the downwards routes increases significantly with an increase in the average ranks of nodes. More specifically, the source routing message, i.e., DAO, needs to travel more number of hops when the root is placed in the corner compared to it is in the center. Another reason is that, in non-storing RPL, all the intermediate nodes add their address into the DAO while they forward it, so that the size of DAO increases as a node’s hop distance to the root increases. Besides, the performance difference resulting from different value of I_{max} indicates that the control overhead with a higher value of I_{max} is smaller. The reason behind this is that the overhead used to support the downwards routes decreases as I_{max} increases. More specifically, the frequency for broadcasting the control message, i.e., DIO, is smaller with a larger maximum time interval. As a result, the routing control overhead becomes lesser with a larger maximum time interval.

Convergence is an important performance indicator for a routing protocol, as

(a)

(b)

(c)

Figure 3.9: Initialization of the DODAG Topology varying the network size under different root placement. (a) The maximum and average rank. (b) Routing control overhead. (c) Network convergence time.

one of the major goals in designing a routing protocol is to allow all nodes in the network to quickly and reliably converge. Fig. 3.9(c) shows that it takes longer time for the network topology to converge to a stable state as the network size increases. Furthermore, from the figure, we can observe that the topology converges faster with a smaller value of I_{max} . The reason behind this is that the frequency of broadcasting DIO, which is used for route propagation, is generally higher with a smaller value of I_{max} with the trickle algorithm implemented in RPL. Further observation on the performance difference on node positioning strategy reveals that the topology converges faster when the root is placed at the center. The faster converge time is due to a smaller average distance, which is indicated by the average rank of nodes to the root.

3.5.4 Performance Evaluation of Traffic Patterns

In this section, the performance of RPL and P2P -RPL with different traffic patterns, including MP2P, P2P, and P2MP, are studied. In a $245\text{m} \times 245\text{m}$ network area, 120 nodes are randomly deployed. Without loss of generality, the root is randomly deployed without specifying its position. Using the parameters in Table 5.1, the network performances with different link loss rates are investigated. Specifically, the link success rate at the edge of the network is varied from (0.1, 0.3, 0.5, 0.7). Two objective functions, i.e., MRHOF and OF0, are used as the routing metrics in this study. $\text{RPL}(\text{MRHOF})_{30 \text{ flows}}$ and $\text{RPL}(\text{OF0})_{30 \text{ flows}}$ denote the routing performance for RPL using MRHOF and OF0 as its routing metric with 30 traffic flows, respectively. Furthermore, the performance of RPL with 50 traffic flows are also examined with the objective function of MRHOF, which is denoted by $\text{RPL}(\text{MRHOF})_{50 \text{ flows}}$.

Multipoint-to-Point (MP2P) Scenarios

Fig. 3.10(a) shows that the packet delivery ratio (PDR) of RPL increases as the link success rate increases. Compared to $\text{RPL}(\text{OF0})_{30 \text{ flows}}$, $\text{RPL}(\text{MRHOF})_{30 \text{ flows}}$ achieves a better performance when the link success rate is low. Furthermore, the PDR of $\text{RPL}(\text{OF0})_{30 \text{ flows}}$ approaches to that of $\text{RPL}(\text{MRHOF})_{30 \text{ flows}}$ when the link success rate is 0.7. The reason behind this is that, MRHOF chooses the route

(a)

(b)

(c)

Figure 3.10: Impact of link success rate under MRHOF and OF0 with MP2P traffic. (a) PDR. (b) Average path length. (c) End-to-end delay.

with the consideration of the link loss rate. Specifically, MRHOF selects the route with a minimum number of transmissions to the root, whereas OF0 selects the route with minimum number of hop counts to the root without considering the link loss rate. Therefore, the benefit of choosing MRHOF as the objective function is more obvious when the link loss rate is high. Compared to $\text{RPL}(\text{MRHOF})_{50 \text{ flows}}$, $\text{RPL}(\text{MRHOF})_{30 \text{ flows}}$ achieves a better performance in terms of PDR as more packets will be dropped in the area around the root with a high traffic flow.

Fig. 3.10(b) shows that the average path length with MRHOF reduces as the link success rate increases, while the average path length with OF0 nearly remains the same as the link success rate increases. MRHOF defines a node to choose the route with a minimum number of transmissions. With a smaller hop distance, the progress that each transmission makes is greater. Meanwhile, the number of lossy links that a packet needs to travel to reach the destination is also smaller. In this way, a node tends to choose the node, which is closer to the root, as its preferred parent when the link success rate is getting better. However, with OF0 the selection of preferred parent is independent from the link loss rate. So the average path length is not affected by a variation of link loss rates.

Fig. 3.10(c) shows that the end-to-end delay decreases as the link success rate increases. Compared to $\text{RPL}(\text{MRHOF})_{30 \text{ flows}}$, $\text{RPL}(\text{MRHOF})_{50 \text{ flows}}$ suffers from much longer delay, especially when the link lossy rate is high. The reason behind this is that both link-level and node-level congestion takes place when the traffic flow is high, resulting in much more retransmission of packets and longer queuing time. A longer delay resulting from high traffic is more serious when the link lossy rate is high. Further observation on the difference between MRHOF and OF0 reveals that, $\text{RPL}(\text{OF0})_{30 \text{ flows}}$ suffers from longer delays compared to $\text{RPL}(\text{MRHOF})_{30 \text{ flows}}$. The longer end-to-end delay in $\text{RPL}(\text{OF0})_{30 \text{ flows}}$ results from more number of retransmissions, because more packets get dropped in a lossy network with OF0 than with MRHOF, as shown in Fig. 3.10(a).

Point-to-Point (P2P) Scenarios

Fig. 3.11(a) shows that the PDR of RPL increases as the link success rate increases. With the objective function of MRHOF, RPL achieves a better perfor-

(a)

(b)

(c)

Figure 3.11: Impact of link success rate under MRHOF and OF0 with P2P traffic. (a) PDR. (b) Average path length. (c) End-to-end delay.

mance than that with OF0 as the objective function, especially when the link success rate is low. It indicates that RPL can achieve higher reliability if the lossy rate of wireless links are considered in the path selection. In addition, compared to $\text{RPL}(\text{MRHOF})_{30 \text{ flows}}$, the performance of RPL drops when the number of traffic flows is 50, as shown by $\text{RPL}(\text{MRHOF})_{50 \text{ flows}}$. The reason behind this is that the root acts as the intermediary in order to support the P2P communication. The area around the root easily get congested under high traffic flows, so that large amount of packets get dropped in this case.

Fig. 3.11(b) shows that the average path length with MRHOF reduces as the link success rate increases, while the average path length with OF0 nearly remains the same as the link success rate increases. Compared Fig. 3.11(b) with Fig. 3.10(b), the average path length in P2P communication is much more than that in MP2P communication, the main reason of which is the usage of the intermediary. In non-storing mode, the intermediary has to be the root, resulting in an increased hop counts in average.

Fig. 3.11(c) shows that the end-to-end delay decreases as the link success rate increases. Further observation of Fig. 3.11(c) reveals that the end-to-end delay increases significantly with an increased number of flows. The reason behind this is that congestion occurs in the area around root, leading to much more retransmission of packets and longer queuing time. A longer delay resulting from high traffic is more serious when the link lossy rate is high. Compared with different objective functions applied in this work, $\text{RPL}(\text{OF0})_{30 \text{ flows}}$ has longer end-to-end delay compared to $\text{RPL}(\text{MRHOF})_{30 \text{ flows}}$. The longer end-to-end delay results from more number of retransmissions. Specifically, more packets get dropped in a lossy network with OF0 than with MRHOF, as shown in Fig. 3.11(a).

Point-to-Multipoint (P2MP) Scenarios

Fig. 3.12(a) shows that the PDR increases as the link success rate increases. As mentioned earlier, the DODAG is constructed in a way that the MP2P communication is optimized. However, the downwards route of the DODAG, which is used to support the P2MP traffic, is not selected optimally. Hence, the performance of RPL with P2MP traffic is worse than that of with MP2P traffic as shown in Fig.

(a)

(b)

(c)

Figure 3.12: Impact of link success rate under MRHOF and OF0 with MP2P traffic. (a) PDR. (b) Average path length. (c) End-to-end delay.

3.12(a) and Fig. 3.10(a). Furthermore, Fig. 3.12(a) also demonstrates that the PDR performance is better with light traffic when $\text{RPL}(\text{MRHOF})_{30 \text{ flows}}$ is compared with $\text{RPL}(\text{MRHOF})_{50 \text{ flows}}$. The reason behind this is two fold. First, a large number of traffic flows leads to serious transmission collision. Second, the area nearby the root gets congested easily under high traffic.

Fig. 3.12(b) shows that the average path lengths of $\text{RPL}(\text{MRHOF})_{30 \text{ flows}}$ and $\text{RPL}(\text{MRHOF})_{50 \text{ flows}}$ reduces as the link success rate increases while the average path length of $\text{RPL}(\text{OF0})_{30 \text{ flows}}$ is not affected by different link loss rates. The route used to support P2MP traffic is the reverse route used to support the MP2P traffic. So the average path length with P2MP traffic patterns is close to that with MP2P traffic patten, as shown in Fig. 3.10(b).

Fig. 3.12(c) shows that the end-to-end delay of $\text{RPL}(\text{MRHOF})_{50 \text{ flows}}$ is higher compared with $\text{RPL}(\text{MRHOF})_{30 \text{ flows}}$. In P2MP traffic scenario, the root acts as the only source node in the network, so that congestion takes place easily in the root area, especially when there is burst data to be sent by root node. Therefore, $\text{RPL}(\text{MRHOF})_{50 \text{ flows}}$ suffers from long end-to-end delay.

3.6 Efficient Routing Support for LLNs

Owing to its great flexibility, RPL has emerged as a prominent candidate for routing in IoTs and M2M communications. The routing specifications of RPL are designed to meet various requirements of diverse LLNs applications. However, it is a difficult task to provide efficient routing support in LLNs due to its severe resource constraints and lossy environment. There are many challenging issues that need to be adequately addressed for the pervasive utilization of LLNs applications in real-word. Power conservation and reliable data delivery are the two main primary goals to design routing protocols for LLNs. In recent years, research studies of RPL with focus on enhancing network performance in terms of reliability and energy conservation have emerged [141, 142]. However, it is a challenging and crucial task to achieve energy-efficient and reliable routing support for LLNs. In order to address the challenges, this thesis proposes routing solutions by specifically tackling the following key research issues:

- **Reliable route discovery in an energy-efficient manner.**

In LLNs, the optimal route between two arbitrary nodes is not provided by default due to limited memory and buffer. As mentioned earlier, RPL constructs DODAGs in a way that the MP2P routing support is optimized. However, RPL cannot provide efficient routing support for other traffic patterns, such as P2P and P2MP, because data needs to travel along the pre-established DAG. As a result, packets have to travel through more numbers of lossy links, resulting in higher latency and lower packet delivery ratio. Specifically, P2P communication, as a fundamental requirement of various LLNs applications, needs to be supported by an intermediary in RPL. In particular, the root is the only intermediary in the non-storing mode. As a result, the root easily becomes the bottleneck when the traffic is high. In order to efficiently support P2P communication, route discovery is the common approach to discover the best-quality route in the networks. Traditional reactive routing protocols discover the best-quality route by disseminating control messages throughout the network, resulting in large amount of control overhead, especially in large scale and dense networks. During the process of route discovery, all nodes in the network need to participate in the route discovery, leading to considerable energy consumption. In LLNs, it is vital to discover the optimal route in an energy-efficient manner. Therefore, achieving energy-efficient route discovery without compromising the reliability is a crucial issue of protocol designs.

Another important nature of wireless links is the asymmetric nature, which has been investigated by numerous empirical studies. The main reason of link asymmetry is that transmitter power and receiver sensitivity are different from nodes to nodes [144]. In addition, some other factors, such as the reflectors, absorbers, etc, also result in link asymmetry. The asymmetric nature of wireless links has significant influence on the performance of routing protocols. Protocols without considering the link asymmetry sometimes fail when it is encountered in the real deployments, especially in LLNs [145]. Therefore, it is critical to take the link asymmetry into consideration while designing routing protocols to support energy-efficient route discovery without compromising the reliability.

- **Reliable data collection while prolonging the lifetime of the network.**

Initially, as most of the LLNs applications focus on a various type of data collection, the predominate communication paradigm is multipoint-to-point (MP2P) routing. It is shown that a paradigm shift for routing of traditional ad hoc networks to the large convergecast networks have taken place [146]. RPL selects a single preferred parent and exclusively forwards packets through its preferred parent. In particular, single path routing cannot cope well with the dynamic and lossy wireless channels. The unreliability of wireless channels leads to packets loss and extensive retransmission, which results in high energy consumption and long channel occupancy time. It is difficult to achieve energy efficiency and reliability simultaneously, especially in LLNs.

Opportunistic routing (OR) [20] is proposed for wireless mesh networks. Owing to the broadcast nature of wireless channels, OR can combine multiple “poor” wireless links into a more reliable link. Another advantage of OR is that it defers the selection of the next forwarder after the packet transmission. In OR, the data packet is sent concurrently to multiple nodes, which become the candidate relays. The forwarder of the packet is selected according to the actual results of the packet reception, which effectively leverages the packet delivery. OR exploits the path diversity to improve performance of networks. Much work has been done to explore the potential of OR to improve network reliability [147, 148]. However, coordination errors can easily occur in a lossy environment, leading to duplicate transmissions and the occurrence of high energy costs in LLNs. Therefore, further study is required to adopt OR and exploit its benefits for LLNs.

Based on OR, anypath routing was proposed in [22], which offers a conventional method to compute the routing cost in terms of the total number of transmissions. Traditional routing protocols usually select forwarders in the way that the total number of transmissions is minimized [22, 148]. Although the energy depleted in transmitting is the major component of the energy consumption for a network, other factors, such as the link lossy rate, energy depletion in receiving and overhearing also have significant impact

on the lifetime of the network, especially for a dense network with dynamic and lossy wireless links. Thus, in practice, the traditional approaches usually face difficulties in selecting an optimal route for energy constraint nodes. The network connectivity decreases significantly once the energy of some nodes starts to drain out, leading to performance degradation. Consequently, for a network composing of nodes with a limited energy resource, the optimal routes should be selected in the way that the energy consumption of the network is minimized.

In LLNs, a large number of energy constraint nodes may be potentially inaccessible in real environment. Conserving energy to prolong the lifetime of the network is crucial for LLNs. Owing to the broadcast nature of wireless channels, nodes may overhear the transmissions even if they are not the intended recipients of these transmissions. This can constitute a significant portion of the total energy cost in dense networks [149], so the energy expended in receiving and overhearing should also be considered part of the total amount of energy consumed as a whole in one transmission. In addition, the unnecessary energy expenditure also results from unnecessarily high power transmissions [23], which shortens the lifetime of the network and introduces mutual interference in a shared wireless medium. Recent developments have enabled the advancement of the multifunctional nodes, which are low-power, low-cost, tiny and have continuous or discrete power levels [23, 24]. Existing research studies on the adjustable transmission range have two main categories. The first group is to explore the minimum transmission range that can maintain the network connectivity [23]. Another group is the clustering approach, which focuses on how to select the optimal transmission range for the cluster head [24, 25]. To the best of our knowledge, the subject of adjustable transmission power for optimal route selection in terms of energy efficiency and reliability has not been thoroughly studied. RPL can be enhanced in a way that each node can transmit data packets at an optimal power level to optimize the energy consumption of the network.

In LLNs, the random deployment of nodes and heterogeneous traffic patterns result in an unbalanced distribution of workload, making the nodes in the hotspot areas to exhaust their energy quickly. For example, in a data collec-

tion application, the nodes close to the gateway usually deplete their energy more quickly than the other nodes, known as “Energy Hole Problem” [150]. In order to prolong the lifetime of the network and to solve the problem of energy holes, workload sharing is inherently important for RPL since the desired reliability cannot be achieved once the network becomes disconnected.

3.7 Summary

Routing is of paramount importance for M2M communications. It is envisioned that LLNs applications will be worldwide and an essential part of our life in the near future. However, it is a difficult task to provide reliable, energy-efficient, scalable and robust routing support for LLNs. The flourish of LLNs application hinges on fully understanding the constrains and state-of-the-art routing technologies. This chapter has reviewed the main routing protocols proposed for LLNs and presented a comprehensive study on RPL. Furthermore, by implementing RPL framework in NS-3, an experimental evaluation of RPL has been conducted. The performance metrics in terms of network convergence time, packet delivery ratio, average path length and end-to-end delay during the network initialization phrase and data transmission phrase (subject to different traffic patterns including P2P, MP2P and P2MP) are studied to validate and analysis the flexibility and limitations of RPL. Furthermore, the key research challenges of achieving energy-efficient and reliable routing in LLNs have been addressed, which is the motivation for further research work in the thesis.

Chapter 4

Point-to-point Hybrid RPL and Multipoint-to-point Cluster-parent-based RPL for LLNs

4.1 Introduction

In the previous chapter, a comprehensive study on RPL was conducted. Specifically, an implementation framework of RPL was presented. In addition, the performance under different traffic patterns including P2MP, P2P and MP2P has been extensively evaluated. As these traffic patterns become more popular with the pervasiveness of LLNs applications, an ever growing demand for power conservation and reliable data delivery is expected, while still abiding by the severe resource constraints and unreliability of LLNs. This motivates us to further investigate efficient routing support under different routing patterns in LLNs. By addressing the limitation of RPL, this chapter proposes two routing protocols, i.e., hybrid RPL (HyRPL) and cluster-parent-based RPL (CRPL) that aim to provide reliable routing support while achieving reduction of energy consumption.

Point-to-point (P2P) communication, as a fundamental communication paradigm of many LLNs applications, is becoming more and more important due to its widely and crucial usage in practice. Many home and building automation appli-

cations, such as the remote control applications, require an efficient P2P routing support [9, 10]. However, in LLNs, the optimal route between two arbitrary nodes is not provided by default due to limited memory and buffer size. RPL uses pre-established routes along the directed acyclic graph (DAG) for P2P routing support. As the construction of DAG is only optimized for MP2P, RPL cannot provide efficient routing support for P2P communication. Therefore, point-to-point RPL (P2P-RPL) [122] is standardized as an enhancement of RPL to provide efficient P2P communication support in a reactive manner. P2P-RPL discovers the route with best quality route by disseminating control message throughout the network. Although reliable route can be discovered with P2P-RPL, the large amount of control messages results in considerable energy consumption, which is an extremely expensive cost for LLNs. In order to tackle the aforementioned challenges, this chapter proposes a hybrid design of proactive and reactive protocol based on RPL and P2P-RPL, namely HyRPL. HyRPL efficiently balances the trade-off between energy-efficiency and reliability. Without compromising the reliability, HyRPL achieves significant reduction in control overhead. Specifically, HyRPL determines the necessity of route discovery by measuring the cost of route along pre-established DAG so as to attain the performance superiority.

In practice, as many of the deployments focus on various types of monitoring, the predominant communication paradigm is *Collection-Oriented*, or *many-to-one routing*, regarded as MP2P traffic. Meanwhile, it has been shown that a paradigm shift of routing for traditional wireless sensor networks and other small scale networks to large convergecast networks consisting of multitude nodes had taken place [146]. In RPL, a single preferred parent is selected to forward packet, which face difficulties in coping well with the lossy link. Packet retransmission will occupy bandwidth, consume energy and shorten network lifetime. Besides, the power of nodes along good paths drain quickly with single relay, so that some of the connected nodes become isolated. Opportunistic routing (OR) [20] shows advantages in improving the reliability of the networks by leveraging path diversity. By exploiting the benefit of OR, the cluster-parent-based RPL protocol (CRPL) is proposed. CRPL is a highly robust and energy-efficient routing protocol for LLNs and shows significant advantages in terms of the network reliability, reduction of packet retransmissions and much lower delivery cost. CRPL allows each node to select an optimal cluster-parent set (CPS) in a distributed manner with a low complexity

algorithm. The coordination among CPS is based on overhearing. Besides, the network load is shared by multipath so that the power of nodes along good paths will not drain out quickly.

The *Dissemination-Oriented* (or P2MP) traffic occurs when there is a need to send commands by the root/gateway to other nodes in the network. P2MP is a minor component in the total LLNs traffic, and it can be efficiently supported by the existing DODAG architecture of RPL. Therefore, this thesis does not conduct further investigation on P2MP traffic.

This chapter proposes HyRPL and CRPL with a focus on the enhancement of P2P and MP2P traffic patterns, respectively. Extensive simulation studies are conducted to validate the effectiveness and flexibility of HyRPL and CRPL. In addition to RPL, P2P-RPL and Opportunistic RPL (ORPL) [142] are also adopted and implemented in NS-3 as the benchmark protocols. The performance metrics in terms of packet delivery ratio, routing control overhead, number of retransmission and average path cost are measured under various scenarios. Simulation results demonstrate the significant performance superiority of the proposed HyRPL and CRPL over the benchmarks.

The reminder of this chapter is organized as follows. The benchmark routing protocols are introduced in Section 4.2. Section 4.3 presents the proposed HyRPL with its implementation design. Section 4.4 introduces the proposed CRPL, followed by the analysis and performance evaluation in Section 4.5. Finally, this chapter is summarized in Section 4.6.

4.2 Benchmarks

This section presents a latest and standardized addition of RPL, i.e, P2P-RPL and a recent enhancement of RPL, i.e., ORPL.

4.2.1 Point-to-point RPL (P2P-RPL)

In LLNs, the routing path between two arbitrary nodes is usually not provided by default in order to keep the routing table size small. RPL shows inefficiency in

P2P communication support due to the usage of an intermediary. Therefore, the route discovery is the common approach used to discover the best quality route between a source and destination pair. In order to facilitate the end-to-end traffic transmission in LLNs, P2P-RPL constructs the temporary DODAG when there is a P2P communication requirement. The source node of the traffic flow acts as the temporary root for the temporary DODAG. The lifetime of the DODAG is strictly restricted according to the lifetime of the route request. Specifically, P2P-RPL piggybacks the P2P Route Discovery Option (P2P-RDO) with the DIO discover routes for the source and destination pairs. The source node originates the route discovery message and disseminates P2P-RDO throughout the network with the trickle algorithm. In P2P-RPL, the routing metric can be specified according to the application objective, so that the optimal end-to-end path will be selected in terms of the metrics. Upon receiving the P2P-RDO message, nodes will check the destination address enclosed in RDO. If the node's IP address is not the destination address, the node will compute its rank, choose a "temporary preferred parent" node to the source node and join the temporary DODAG, and propagate this information. In this way, a temporary DODAG will be formed across the network. Once the destination node receives this information, it replies with a P2P Discovery Reply Object (P2P-DRO) through the discovered route to the source node. The reverse route will be used for the P2P transmission. Upon receiving a P2P-DRO, a node will check the origin of this route discovery process. If the node is not the source node that initiates this route discovery, it will consider itself as an intermediate node and further forward the packet to its "temporary preferred parent", and the reverse route is recorded for data packet forwarding. When the source node receives the P2P-DRO message, it starts to transmit data packet through the discovered routing path to destination.

An example of RPL and P2P-RPL for supporting P2P communication can be found in Fig. 4.1. It shows the communication from node S to node D , in which the dotted line represents route provided by RPL and the solid line represents the route provided by P2P-RPL.

Figure 4.1: P2P communication route supported by RPL and P2P-RPL.

4.2.2 Opportunistic RPL (ORPL)

The recent work in [142] introduces an opportunistic forwarding scheme, namely Opportunistic RPL (ORPL). ORPL takes advantage of cooperative relays to improve the reliability of advanced metering infrastructure (AMI) networks. In ORPL, the formation of DODAG works in the same way as RPL. And ORPL inherits the preferred parent concept of RPL. Each node has a candidate parents set advertised by its preferred parent. The preferred parent selects its neighbour nodes in the order of increasing distance, and constitutes a relay set for its children nodes. With the knowledge of preferred parents' neighbour nodes, nodes will choose candidate parents set consisting of their preferred parents and the ones sufficiently close to their preferred parents. Then nodes send data packets to their candidate parents, which will coordinate with each other to forward data packets. With the help of candidate parents set, ORPL shows advantage in improving the reliability of the network. However, during the process of candidate parents selection, ORPL does not consider the end-to-end cost and works in a decentralized way, while our CRPL chooses the cluster-parent set through a top-down approach to minimize the end-to-end cost for each node. Besides, ORPL inherits the preferred parent from RPL by fixing the candidate parents set to be the neighbour of preferred parent. But the preferred parent is chosen in single path forwarding scheme, which may not be the optimal choice in multipath routing. Meanwhile, in ORPL, the top priority to forward packets is always assigned to the preferred parent. The delivery cost can be reduced if other nodes with lower end-to-end cost in candidate parents set can send packets first. In other words, higher priority

should be assigned to nodes with lower cost to the root, which is considered in the design of CRPL.

4.3 Hybrid RPL Routing Protocol (HyRPL) for Point-to-point Traffic

RPL uses the routes along the pre-established DAG. It cannot provide efficient P2P routing support because the DODAG is optimized to for data collection at the root [143]. In P2P-RPL, the control messages disseminate across the network, resulting in large amount of control overhead and energy consumption. Additionally, P2P-RPL can find better P2P route than RPL in most case. But sometimes the route discovered by P2P-RPL may not be much better than the existing route in RPL, which is through the DAG root from the source to the destination. The improvement of route quality depends on the network topology, such as the distance between source and destination. Moreover, the cost associated with P2P route discovery is very expensive in LLNs, because all nodes in the network needs to participate in temporary DODAG formation for the route discovery, which leads to significant cost in energy-constraint networks.

HyRPL is designed to balance the trade-off between energy-efficiency and reliability. HyRPL measures the cost of routes along the pre-established DAG before discovering the optimal route for a source-destination pair. With the proper selection of the route constraint, HyRPL achieves significant reduction in control overhead without compromising the reliability. Furthermore, before a routing protocol is put up for real-life use, simulating and evaluating the performance is crucial in order to have a better understanding of the routing algorithms. By exploiting the trust-able, practicable, and flexible features of NS-3, an integrated framework that comprise RPL, P2P-RPL and HyRPL is designed and implemented. After introducing the details of the proposed HyRPL, this section presents the details of implementation framework of HyRPL.

4.3.1 Routing Specification of HyRPL

In order to achieve efficient P2P routing while minimizing the control overhead, HyRPL, as a hybrid of proactive and reactive routing protocol, efficiently combines the mechanism of DODAG formation and route discovery of RPL and P2P-RPL. Importantly, HyRPL employs a cost measurement mechanism from the work in [151] to achieve the goal of efficient P2P communication support with low routing overhead. HyRPL inherits the existing DODAG construction of RPL in addition to its efficient support for P2P route discovery. Therefore, HyRPL is designed with the capability to support the generic traffic patterns. HyRPL allows a node to determine whether or not it should discover the best-quality route. HyRPL has an important routing parameter, i.e., the routing constraint, which can be in terms of node level or link level metric. Node level metrics take the condition of each node along the path into consideration during the path cost calculation, whereas link level metrics calculate the path cost according to the condition of wireless links along the path. Specifically, node level metrics includes node energy, node state and hop count. Link level metrics includes throughput, latency and reliability. More details in terms of node level and link level metrics can be referred from Table 3.3. It is worthy to highlight that the necessity of route discovery in HyRPL is determined by whether or not the cost of the existing route fulfills the routing constraints.

HyRPL comprise three phases: 1) network initialization phase for DODAG construction; 2) on-demand route cost measurement; 3) reactive route discovery. In the network initialization phase, HyRPL forms the DODAG in the same manner as RPL. The detailed process of DODAG initialization has been demonstrated in the previous, is therefore not addressed here. When there is a P2P routing request for an arbitrary source-destination pair, HyRPL first sends a Measurement Object (MO) from the source node to the destination node through the available route via root¹. According to the objective function, intermediate nodes record the accumulated cost along the routing path towards the destination. Upon receiving the MO message, the destination node unicasts a Message Reply Object (MRO) to the source node with the accumulated cost of the P2P route that has been measured.

¹This route is the pre-established route along the DAG in RPL during the network initialization phase

If the accumulated cost value is not more than the the routing constraints, the source node immediately send data using the existing P2P route instead of discovering another one. Otherwise, a source node broadcasts the control messages to discover the optimal route throughout the network. HyRPL adopts the similar route discovery method defined in P2P-RPL. The detail of the process for route discovery, which has been introduced in P2P-RPL, is therefore not addressed in this section.

The routing design of HyRPL is subject to different traffic patterns including MP2P, P2P and P2MP traffic. In the proposed HyRPL, a source node needs to first distinguish the traffic pattern according to the destination address. There are three cases in accordance to three types of traffic, as shown in Fig. 4.2. In the first case, the source node forwards packets through its preferred parent if the destination node is the root. The address of root is a piece of globe information, which is enclosed in the DIO message. This case is identified as MP2P traffic. In the second case, the source node itself is the root, so that the packets are forwarded to the destination through a list of relays according to a source routing list. The source routing here is enabled with the help of DAO message. The third case is the P2P traffic that requires an intermediary to support P2P routing. HyRPL has two types of configurations, i.e., *P2P-enabled* and *P2P-disabled*. Specifically, when the communication is identifies as P2P traffic, the source node first checks whether it is the mode of *P2P-enabled*. If so, the packet is attached with a Tag of *DeferredRouteOutputTag* and inserted into a buffer. Then, the mechanism to measure the routing cost is triggered to measure the routing cost of the existing path via root. If the routing cost is not within the route constraint, the best quality P2P route will be discovered using P2P-RPL.

4.3.2 Integrated Framework Design of HyRPL

The implementation of HyRPL comprises three main components: 1) DODAG construction; 2) mechanism for routing cost measurement; 3) on-demand route discovery. Fig. 4.3 shows the class diagrams of the proposed HyRPL module, which complies with the existing structure of NS-3 IPv6 routing protocol. As shown in the figure, the class of *HyRPL* mainly consists of five components: *Control Message*

Figure 4.2: The route processing diagrams of HyRPL at the source node subject to different traffic patterns.

Header, Routing Table, Packet Buffer, Objective Function and Trickle Timer. In particular, in order to manage the frequency of control messages for each flow, an enhanced version of trickle algorithm is implemented in HyRPL, in which each timer is identified by a pair of source node and sequence number.

Figure 4.3: Designed classes diagrams of HyRPL in NS-3.

Framework Design for DODAG Construction

Four methods are designed namely, *SendDIS*, *RecvDIS*, *SendDIO* and *RecvDIO* for the formation of the upwards route in the DODAG. *SendDIS* defines a floating node to multicast the DIS message periodically. The joining of the DODAG is initialized with the help of DIS. The *RecvDIS* method is called once a node receives a DIS message, and it triggers the *SendDIO* to unicast a DIO if the node is a grounded node. Upon receiving the DIO from the grounded node, a node stops sending the DIS messages and cancels the DIS timer, which is defined in *RecvDIO*. Meanwhile, the source of DIS message updates its neighbor list and triggers a reset for trickle timer. Upon receiving unicast DIO message, a node will update its status, choose the sender as its preferred parent and compute its rank. The selection process of preferred parent is completed by the method of *RecvDIO*.

The DAO message can be either entirely disabled or operated to support the storing/non-storing mode. The *SendDAO* and *RecvDAO* are defined to enable the downwards routing. In Storing mode, a node sends the DAO message to its DAO parent, which is its preferred parent in our design, with its downwards route information. The recipient stores the routing table entries of the routes, which are learnt from the DAO message. The source routing is used in the Non-Storing mode, in which the DAO is piggybacked with a vector of IP address. A node sends a DAO message to the root. Instead of storing the downward routes, the intermediate nodes add their addresses into the DAO message before forwarding the message. The roots store the source routing list with the address of the source node as the routing table entry.

Framework Design for Routing Cost Measurement

The format of the measurement object is shown in Fig. 4.4. The measurement object carries the routing metrics, source address and destination address, as the start point and end point, respectively. The *type* field is used to distinguish message request object and the message reply object. For a P2P traffic flow, if the source node does not have a valid route to the destination, it adds a *DeferredRouteOutputTag* to the data packet when the *RouteOutput* is called. Before the data is delivered, the *RouteInput* is called, and the packets with the *DeferredRouteOutputTag* will be sent by using of the *DeferredRouteOutput* method. The method of *DeferredRouteOutput* is designed to store packets into the buffer and to initiate the route discovery process in order to enable the P2P communication. The mechanism for routing cost measurement is implemented. The cost of the route along pre-established DAG is measured to determine the necessity for route discovery². The important methods to measure the routing cost along an existing route are *SendMRO* and *RecvMRO*, which handle the operation of the message request object and the message reply object. The *SendMRO* is activated in the *DeferredRouteOutput* method, and the message request object is used to carry the accumulated cost given a particular objective function. The route for data delivery can be achieved as either hop-by-hop route or the source route with the message

²In the non-strong mode, this existing route is along the pre-established DAG and via the root.

reply object.

Type	Start Point Address	End Point Address	Sequence	Hop Count	Aggregate Cost
------	---------------------	-------------------	----------	-----------	----------------

Figure 4.4: Packet format of the measurement object.

Framework Design for Route Discovery

According to the standard, the Route Discovery Option (RDO), the structure of which is shown in Fig. 4.5, is defined as an option field of the DIO message. Two methods, *SendP2PRDO* and *RecvP2PRDO* are defined to propagate the route discovery information and build the temporary DODAG. The *RecvP2PRDO* method checks the field of the *DestAddr*. If it is not the destination for this route discovery process, a node adds the sender of this RDO message to its neighbour list, calculates its cost, chooses its preferred parent and get the rank value. Then the node broadcast its RDO message by calling the method of the *SendP2PRDO*. In this way, a temporary DODAG rooted at the source node is constructed.

DODAG_ID	Target Address	Sequence	Flag	Lifetime
----------	----------------	----------	------	----------

Figure 4.5: Packet format of the Route Discovery Option.

The operation of Discovery Reply Object (DRO), the structure of which is shown in Fig. 4.6, is defined in the methods of *SendP2PDRO* and *RecvP2PDRO*. The method of *SendP2PDRO* is called by the *RecvP2PRDO* method if the receiver is the destination node of this route discovery process. The DRO is delivered through the discovered route to the source node, which is the root of the temporary DODAG. The *RecvP2PRDO* method records the reverse route of the DRO message for the transmission of the data packet. We have also designed a source routing object, essentially enabling the source routing. The source routing object are used to recode the address of the intermediate nodes, so that the source node is able to deliver the data packets along the route to destination that has been discovered. The size of the source routing object is dynamically allocated. The source node will send the data packet with either hop-by-hop routing or source routing once the *RecvP2PRDO* method is called.

DODAG_ID	Hop Count	Sequence	Flag
Route Discovery Option(s) ...			

Figure 4.6: Packet format of the Discovery Reply Object.

It is worth to highlight that there may be multiple concurrent traffic flows in the network, and the propagation and maintenance of each of the temporary DODAG is based on one trickle timer. The harsh map is applied to maintain the trickle timer subject to each of the traffic flow. The key of each row is defined by a pair of the source node address and the sequence number. The logic of defining the sequence number is like this: the sequence number increase only if the destination node does not exists in the list.

4.4 Cluster-parent-based RPL (CRPL) for Multipoint-to-point Traffic

In the previous section, we have introduced the defined P2P-RPL that support P2P routing in a reactive manner. In this section, we investigate the issue of reliable data collection by adopting opportunistic routing in RPL for MP2P traffic support. Traditional routing protocols deliver packets through single relay and cannot cope well with the unreliable wireless medium. Traditional routings select single next-hop based on one metric, and packets are sent exclusively following that path. In RPL, a single preferred parent is used for packet forwarding, which cannot cope well with the lossy link. Packet retransmission will occupy bandwidth, consume energy and shorten network lifetime. Besides, the power of nodes along good paths drain quickly with single relay, so that some of the connected nodes become isolated. Opportunistic routing (OR) [20] takes advantage of the broadcast nature of wireless channel and defers the forwarder selection based on the receiving outcome. In opportunistic routing, packets are sent to multiple relays concurrently. The decision of next forwarder is based on the result of receiving outcome. Upon receiving the data packet, the node will check whether it should participate in packet forwarding or not. And if so, the node starts to schedule packet forwarding

while cooperating with other forwarding nodes to avoid duplicate transmission. Plenty of literatures have studies about how cooperative diversity schemes can improve network performances [20, 22, 148].

In this section, we propose a cluster-parent-based RPL protocol (CRPL), which is an opportunistic coordination forwarding scheme. In CRPL, through a top-down approach, an optimization problem is formulated. The optimal cluster-parent set is selected to minimize the end-to-end cost for individual node. Priority-based scheduling mechanism is applied to coordinate the cluster-parent set for packet forwarding.

4.4.1 Network Model

Consider that n number of stationary nodes with limited energy are dispersed in an area, the network can be described as a non-cyclic graph, denoted by $G = (V, E)$, where V represents the set of nodes in the network, and E represents the set of edges in the network. In CRPL, a node chooses the best combination of neighbor nodes its cluster-parent-set (CPS), which helps to forward packets to the destination node. In order to define the most cost efficient CPS, we conduct a mapping for every node to a positive, real-valued energy cost for routing as $c_e : V \rightarrow \mathbb{R}^+$. Furthermore, a mapping for each link to a link cost, which is $c_l : E \rightarrow \mathbb{R}^+$ is performed. which makes the network becomes a weighted graph. In this way, the cost of the edge $(i, j) \in E$ is denoted as $c(i, j)$. The CPS of node i is denoted by $CPS(i)$, and $c(i, CPS(i))$ is used to denote the cost for node i to successfully deliver one packet to the root node. Node j is an arbitrary node in the CPS of node i , which is denoted by $j \in CPS(i)$.

4.4.2 Selection of Cluster-Parent-Set

Anycasting routing [152] is used to enhance the network performance with path diversity. However, packet retransmission and redundant transmission are with high expense in terms of energy expenditure. Therefore, when adopting OR for LLNs, we have to carefully consider how to select relay set and coordinate the members in one set. Considering the traffic pattern of RPL, most of the data traffic is des-

minated at the root as multipoint-to-point traffic. Besides, the actual transmission cost is the end-to-end cost instead of one hop transmission cost, so that the routing path in LLNs should be chosen in the way that the end-to-end transmission cost is minimized. Therefore, minimizing the end-to-end cost is an important criterion to select the cluster-parent set. Without considering this feature, packets may be forwarded to some nodes with “poor” connectivity to root. With the objective to minimize end-to-end transmission cost for every node in the network, CRPL selects the optimal cluster set for each node in the network through a top-down approach. Another important issue is the inter-link quality of members in the CPS. Poor inter-link quality causes coordination error, leading to lots of duplicated packets. The duplication of packets may even overweight the advantage of path diversity. Thus, we can only select the nodes with “good” enough inter-link quality to be one set. In our design, the link success rate among the members in one CPS has to be above 0.5.

Due to the lossy nature of wireless medium, the packet may get lost and cannot successfully be delivered through the wireless channel. In this work, the end-to-end transmission cost refers to the number of transmission that a node takes to successfully deliver one packet to the root. In CRPL, the end-to-end transmission cost for a node through its CPS is called CCPS. The accumulated cost value, CCPS, is calculated based on the link quality and the cost value of members in CPS. In CRPL, every node in the network has an optimal cluster-parent set and a CCPS value.

The topology set-up in CRPL protocol includes neighbour discovery, cluster-parent set selection and the topology optimization. The neighbour discovery works in a similar way as RPL. The root initiates the topology construction. The cluster-parent configuration information can be enclosed in DIO message as the objective function. One-hop neighbour information with inter-link quality is advertised with DIO messages. Upon receiving DIO message from neighbours, a two-hop neighbour table can be built up with the information of the link qualities among them. How to measure the link quality is out of the scope, and we assume that the link layer will handle this issue. Based on neighbours’ cost and their inter-link quality information, the node will check all the possibilities of the subset of neighbours. As aforementioned, “good” inter-link quality is required to select nodes into the subset,

so the subsets only consist of members with “good” link qualities among each others. And node will examine the cost through different combination of neighbours by calculating CCPS value regarding to different CPSs. After that, nodes will choose the optimal subset, which gives minimum end-to-end transmission cost to root, as their CPSs and update this information. When receiving any inconsistent information from neighbours, nodes will re-compute the cost value and update the information if necessary. Based on the control message exchanging and cost value updating, the network will converge into a stable topology after some time.

When multiple relays are used for packet forwarding, joint ETX [22] is calculated as the anypath routing cost. Joint ETX is the average number of transmission that at least one of the nodes in the CPS is able to successfully receive the packet, defined in Eq. (4.1), where p_{ij} is the successful transmission probability between node i with node j . We use $CPS(i)$ to represent the cluster-parent set for node i , and node j is an arbitrary node in $CPS(i)$. The $ETX_{CPS(i)}$ is the transmission cost for node i to successfully deliver one packet to at least one node of the $CPS(i)$, so that it is the value of $c(i, j)$, where $j \in CPS(i)$.

$$ETX_{CPS(i)} = \frac{1}{p_{i,CPS(i)}} = \frac{1}{1 - \prod_{j \in CPS(i)} (1 - p_{i,j})}. \quad (4.1)$$

Eq. (4.2) is used to calculate the aggregated cost through CPS to root [148], where RC represents the remaining path cost. It is assumed that the relay set is sorted by end-to-end cost, in which $C_1 \leq C_2 \leq \dots \leq C_{|CPS(i)|}$, where C_j is equivalent to $c(j, CPS(j))$.

$$RC_{i,CPS(i)} = \frac{C_1 p_{i1} + \sum_{j=2}^{|CPS(i)|} C_j p_{ij} \prod_{k=1}^{j-1} (1 - p_{i,k})}{1 - \prod_{j \in CPS(i)} (1 - p_{i,j})}. \quad (4.2)$$

The accumulate end-to-end cost for a node consists of one hop joint ETX and the remaining path cost associated with the cluster-parent-set. Hence, the end-to-end cost for a node through CPS, which is the CCPS value, can be computed through

$$CCPS_{i,CPS(i)} = ETX_{CPS(i)} + RC_{i,CPS(i)}. \quad (4.3)$$

Figure 4.7: The process of cluster-parent set selection in CRPL.

Based on the CCPS and inter-link quality information from neighbours, nodes compute and get different CCPS values with different cluster sets. Then the node chooses the parent set that provides the minimum CCPS value as its CPS. Once receiving the CCPS update from neighbours, nodes compute possible CCPS values again. If the minimum CCPS value is lower than previous value, nodes will advertise this information to its neighbours immediately. Besides, if node A is in the CPS of node B, node A will not let node B to participate in node A's CPS selection. The network will converge into a stable topology after a while.

The way to select the best CPS for an arbitrary node i in the network is to select the $CPS(u)$ that $c(i, CPS(j))$ is minimized, that is

$$\begin{aligned}
 & \underset{x}{\text{minimize}} && CCPS_{i, CPS(i)} \\
 & \text{subject to} && c(i, CPS(i)) = c(i, j) + c(j, CPS(j)), \quad j \in CPS(i), \\
 & && c(i, j) = ETX_{CPS(i)} \frac{1}{p_{i, CPS(i)}} = \frac{1}{1 - \prod_{j \in CPS(i)} (1 - p_{i, j})}, \\
 & && c(j, CPS(j)) = RC_{i, CPS(i)} = \frac{C_1 p_{i1} + \sum_{j=2}^{|CPS(i)|} C_j p_{ij} \prod_{k=1}^{j-1} (1 - p_{i, k})}{1 - \prod_{j \in CPS(i)} (1 - p_{i, j})}, \\
 & && 0 \leq p_{i, j} \leq 1, \\
 & && 0 < C_j.
 \end{aligned}$$

The way to select the best CPS for an arbitrary node i in the network is to

select the $CPS(u)$ that gives the minimum cost to node u , which is $c(u, d) = \min \{c(u, CPS_o(u)) | CPS_o(u) \text{ is optimal CPS for node } u \text{ to } d\}$. In particular, $c(i, CPS(i))$ is composed of two parts: the cost that node i spends to deliver the packet to at least one node in its CPS, and the cost that the packet reception in $CPS(i)$ spends to deliver the packet to the destination through its own CPS, so $c(i, CPS(i)) = c(i, j) + c(j, CPS(j))$, where $j \in CPS(i)$.

Fig. 4.7 illustrates the procedure of node J 's CPS selection. Node E, F and G join the network and advertise their CCPSs. We assume their inter-link qualities are above 0.5, and C_E is 8, C_F is 3 and C_G is 2. And the link qualities between node J with them are $p_{JE}=0.8$, $p_{JF}=0.6$, $p_{JG}=0.7$. Upon receiving neighbour's CCPSs, node J computes the cost through different combination of neighbour set and can get $CCPS_{J,E}$, $CCPS_{J,F}$, $CCPS_{J,G}$, $CCPS_{J,EF}$, $CCPS_{J,EG}$, $CCPS_{J,FG}$, $CCPS_{J,EFG}$ by substituting the values into Eq. 4.3. By comparing these CCPSs, minimum cost for node J is achieved with the CPS containing F and G.

4.4.3 Coordination among Cluster Parent Set

The coordination of CPS is based on a cooperative overhearing scheme. After a node sends data packets, the nodes from its CPS will participate in packet forwarding. Instead of immediately replying acknowledgement (ACK) upon receiving data packets, nodes in CPS follow a priority list to schedule the ACK transmission. Higher priority will be assigned to the node with lower CCPS. If a node fails to receive any ACK from its CPS, it will retransmit the packet. The coordination process can be explained using Fig. 4.8. Node H's CPS consists of node B, C and D, assuming $C_C \leq C_D \leq C_B$. So node C has the top priority and node B has the lowest priority. Node H broadcasts a packet and starts a resend timer. If node C receives the packet, it will reply ACK to node H immediately, and forward the packet. If node D and B receive the packet from node H, they will keep the packet and start to overhear ACK from other nodes with higher priority in the CPS. Thus, node D will start to overhear ACK from node C. Node B has to overhear the ACK from node C and D. If nodes overhear the ACK from the overhearing list, they will discard the packet and stop the ACK timer. Once the ACK waiting time is timeout without overhearing the ACK from the overhearing list, nodes reply ACK

and send out the packet. The ACK scheduling timer is based on the priority in the CPS. In this example, Node D will schedule its ACK transmission after node C, and node B start to schedule after node D. After receiving ACK from any one node of CPS, node H will cancel the packet resend timer. If the resend timer is reached without receiving the ACK from CPS, node H will retransmit the packet.

Figure 4.8: Illustration of overhearing-based coordination approach with node H's CPS.

Upon receiving a data packet, the node needs to know whether it belongs to the CPS of last hop or not. If the node is the member of last hop's CPS, it has to know the overhearing list, which consists of other nodes with higher priority in the CPS. There are two ways to achieve that, shown in Fig. 4.9. Node H can enclose its CPS information in the packet header in the priority order. Any node receiving the data packet will check the packet header first, to decide whether it should participate in the packet forwarding or not and determine its priority in CPS. Another way is that every node keeps a local forwarding table, which records the relevant cluster set. For example, when node D receives a packet from node H, it will check its forwarding table to acquire the necessary knowledge, shown in Fig. 4.9(b). There is tradeoff in both solutions. Attaching the CPS header to packets consumes more energy, especially when the packet size is small and the size of CPS is large. A local forwarding table can efficiently store the CPS information, but it requires additional memory. Considering the constraint and objective, we adopt the second solution in our implementation.

4.5 Network Model and Performance Evaluation

In order to validate the effectiveness of HyRPL and CRPL routing protocol under different network settings, simulation studies are conducted in NS-3 under two scenarios. In the first scenario, P2P traffic pattern is studied and the performance

Data payload				C	D	B
--------------	--	--	--	---	---	---

(a)

From node H	C	D	B
From node I	D	C	...
From node L	D	...	...

(b)

Figure 4.9: Coordination among nodes in a CPS. (a). Node H attaches its CPS in the packet header following priority order. (b) A local forwarding table of node D containing the coordination information.

of HyRPL is compared with RPL and P2P-RPL. In the second scenario, MP2P traffic pattern is studied and the performance of CRPL is compared with RPL and ORPL. The routing performance in terms of PDR, control overhead, and end-to-end delay are evaluated in this study.

4.5.1 Simulation Setup

In order to model the lossy environment and examine the performance under different degree of unreliability, this study uses a unit disk graph model (UDGM) to model the lossy environment with different degrees of unreliability [139]. In this study, the link success rate at edge is varied to model different levels of lossy environment. The link success rate at edge is configured to be 10%, 30%, 50% and 70%. ACK is assumed not to suffer from link loss, because ACK can usually be recovered with strong coding techniques due to its small size. Data traffic is not sent during the network initialization period. The IEEE 802.11 standard is used to evaluate the performance of LLNs protocols [115, 140]. In $245\text{m} \times 245\text{m}$ network area, 120 nodes are randomly distributed. The objective function of MRHOF is adopted to study the impact of link success rate on the routing performance. This study comprises of two scenarios. The first scenario focuses on P2P traffic, in which the performance of HyRPL, P2P-RPL and RPL are evaluated, and the root is randomly deployed without specifying the exact location. In the second scenario, the performance of RPL, OPRL and CRPL are evaluated under multipoint-to-point traffic. In CRPL, the maximum size of CPS is set to be three. Simulation studies are conducted when the root is placed at the corner and randomly deployed, respectively. Other simulation parameter settings are shown in Table 5.1. In this study,

Table 4.1: Simulation parameters to evaluate the performance of HyRPL and CRPL.

Parameter	Value
Communication range	35m
Simulation duration	1000s
Packet size	512 bytes
Traffic rate	1 pkt/s CBR flow
Traffic flow	30
Link success rate at edge	0.1/0.3/0.5/0.7
Mobility scenario	No mobility, randomly distribute
The maximum interval size (I_{max})	8
The minimum time interval size (I_{min})	50ms

different deployments are generated in each run and the results is the average value from 10 runs.

4.5.2 Performance Evaluation of Point-to-point traffic

Figure 4.10: Impact of the link success rate at the edge on PDR.

Fig. 4.10 shows that the packet delivery ratio (PDR) in RPL, P2P-RPL and HyRPL increases as the link success rate increases. In addition, it also reveals that HyRPL can attain a better performance than RPL when the link lossy level is high. As mentioned earlier, P2P-RPL discovers the best quality route given a particular objective function. RPL uses the route along the pre-established DODAG, and requires that the root acts as intermediary to support the P2P communication. As a

result, the data delivery usually takes more number of lossy links and suffers from longer delay. Furthermore, the root becomes a bottleneck in the network when the traffic is high. Compared to RPL, P2P-RPL achieves about 20% improvement when the link success rate is low. In addition, RPL can achieve higher PDR compared to P2P-RPL and HyRPL when the link success rate at the edge is 0.7. The reason is that, in RPL, retransmission scheme in MAC layer can efficiently recover the lost packets when the link success rate is high. However, the control messages used in both P2P-RPL and HyRPL occupy the shared wireless channel resource, resulting in more packet collisions and reduce the probability for success packets delivery. Furthermore, HyRPL sends a routing cost measurement object from the source node to the destination node for each source-destination node pair. However, if the routing cost measurement object gets lost during the transmission, all traffic for this source-destination node pair cannot be successfully delivered. This is the main reason that HyRPL cannot achieve better performance than RPL when the link success rate is 0.7. In order to overcome this issue, in our later work, we implement a routing layer acknowledgement for the reliable delivery of the routing cost measurement object to guarantee its successful delivery from the source node to the destination node. In this study, HyRPL uses the hop count as the route constraint, which is set to be 5 in this study. In HyRPL, the cost measurement is conducted before discovering the best quality route. More specifically, in this case, the existing route along the pre-established DAG will be used for data delivery, if it is not more than 5 hops. Otherwise, HyRPL initiates route discovery to find the best quality route for that source-destination pair.

Figure 4.11: Impact of the link success rate at the edge on average path length.

Fig. 4.11 shows that the average path length in RPL, P2P-RPL and HyRPL

decreases as the link success rate increases. P2P-RPL discovers the best route throughout the network for a source-destination pair and attains a much shorter average path length for the routing pair than that of RPL. It indicates that the routing constraint in HyRPL needs to be carefully chosen, in order to maximize its routing performance. The distribution of nodes should also be considered when selecting the routing constraint for the cost measurement mechanism in HyRPL. Specifically, the average hop counts or average aggregate expected transmission counts of two arbitrary nodes in the networks are good indications for the routing constraints. Therefore, those values can be studied first according to network deployment and used to determine the routing constraints. Furthermore, the routing constraints should be set close to the value of the average hop counts or average aggregate expected transmission counts of two arbitrary nodes in the networks.

Figure 4.12: Impact of the link success rate at the edge on end-to-end delay.

Fig. 4.12 shows that the end-to-end delay decreases as the link success rate increases. Compared to RPL, P2P-RPL and HyRPL achieve much shorter end-to-end delay. Compared to RPL, it takes a much shorter time for P2P-RPL and HyRPL to deliver packets to the destination, due to a short average path length in P2P-RPL and HyRPL.

Fig. 4.13 shows the routing control overhead for the P2P route discovery with 30 traffic flows. In LLNs, the control overhead not only occupies the shared spectrum but also leads to higher cost of energy. Thus, the control overhead is an important performance metric for routing protocols in LLNs. Since RPL uses the pre-established routes along the DAG to deliver packets, it does not use additional control messages to discover the P2P routes. Therefore, Fig. 4.13 does not include the performance of RPL in terms of control overhead. Fig. 4.13 shows

Figure 4.13: Impact of the link success rate at the edge on routing control overhead.

that the control overhead decreases as the link success rate increases. Furthermore, compared to P2P-RPL, HyRPL achieves significant reduction of control overhead. When the cost measurement is enabled, the route discovery is not performed as long as the hop counts of existing route is less than 6. Further observation of Fig. 4.10, 4.11, 4.12 and 4.13 reveals that our proposed HyRPL can effectively balance the trade-off between efficient data delivery and low routing overhead.

4.5.3 Performance Evaluation of Multipoint-to-point traffic

PDR refers to the ratio of the total number of received packets to the total number of transmitted packets. Fig. 4.14 shows that CRPL outperforms RPL and ORPL in terms of PDR with different root positioning strategy, and the benefit is more significant when the link quality is getting worse. Less packets get dropped in CRPL because of the increased successful transmission probability. And the top-down approach shows advantages in finding better route towards root, because it considers the link condition of preceding nodes to root. Comparing Fig. 4.14(a) with 4.14(b), the PDR performance degrades when root is located at the corner. When the root moves to the corner, there will be increased transmission cost for other nodes. So more packets get dropped in this case. However, the decrease of PDR is less serious in CRPL than in ORPL and RPL. In Fig. 4.14(b), Compared to RPL, CRPL can achieve an average 40% improvement in terms of packet delivery ratio (PDR), and ORPL can achieve an average 20% improvement in terms of PDR. Thus, the improvement on PDR achieved by CRPL (i.e., 40%) is double than that by ORPL, (i.e., 20%) in comparison with RPL. With CRPL routing

Figure 4.14: The PDR performance of RPL, ORPL and CRPL. (a) root is randomly placed. (b) root is placed at the corner of the network.

protocol, nodes can select the optimal path, which alleviates the effect of root position on the transmission cost. This result shows that CRPL can effectively overcome the lossy link and improve the network reliability with arbitrary position of root.

Packet retransmission takes place when the relay set or the preferred parent fails to receive the packet. Fig. 4.15 depicts that the number of packet retransmission in RPL, ORPL and CRPL decreases logarithmically with the increasing link quality, and CRPL achieves much lesser number of retransmission (15-20% lower than that of RPL and around 10% lower than that of ORPL). With the help of CPS selection and coordination in CRPL, several lossy links are combined and used to delivery packets, so that the packet reception probability increases and links become more reliable. Consequently, the average number of transmission required to successfully deliver one data packet become less. Besides, CRPL shows more benefit with the increased path length in the network by comparing Fig. 4.15(a) with 4.15(b). Packet retransmission results in energy depletion. CRPL significantly reduces packet retransmission numbers, resulting in significant reduction of energy

Figure 4.15: Packet retransmission performance of RPL, ORPL and CRPL. (a) root is randomly placed. (b) root is placed at the corner.

consumption. Therefore, our CRPL can achieve greater energy saving than RPL and ORPL.

The average path cost is the average total number of packet transmissions required to successfully deliver one packet to the destination. RPL regards the accumulated ETX value as every node's cost. That cost in ORPL is the delivery cost through the candidate parents and its preferred parent. In our CRPL, CCPS counts for the node's transmission cost through its CPS to the root. Fig. 4.16 shows that CRPL achieves much lower average end-to-end path cost for individual node, outperforming RPL and ORPL. Meanwhile, CRPL can effectively alleviate the effect of root position. CRPL incorporates the effect of link loss and the cost of preceding nodes, resulting in better forwarding path selection with the optimal CPS. It is evident that CRPL is able to find low-cost and reliable paths on a lossy network. The reduction of transmission cost leads to great energy saving. As we would expect, CRPL yields more benefit under more traffic flows and longer path over lossy links.

Figure 4.16: Average path cost performance of RPL, ORPL and CRPL. (a) root is randomly placed. (b) root is placed at the corner.

4.6 Summary

This chapter studied a standardized enhancement protocol of RPL, i.e., P2P-RPL, and proposed HyRPL and CRPL.

First of all, to provide flexible and efficient P2P communication support, a hybrid routing protocol, called HyRPL that supports both proactive and reactive routing has been proposed. For P2P communication, HyRPL performs route discovery when the cost of the route along a pre-established DAG does not fulfil the route constraints. The framework design of HyRPL essentially unveils the ambiguities regarding the conformity and validation of P2P communication in LLNs, and shows the benefit for future research on P2P communication. To validate the effectiveness of the proposed protocol, extensive simulation study have been conducted and network performances of the proposed HyRPL are compared with benchmark protocols in terms of packet delivery ratio (PDR), average path length, end-to-end delay and control overhead. Specifically, the performance of PDR shows that, in comparison with RPL, the proposed HyRPL can provide more reliable routing

support when the link success rate is low. In addition, the performance in terms of average path length and end-to-end delay demonstrates that, by delivering packets through shorter routing path, HyRPL can achieve lower latency compared to RPL. Meanwhile, less number of routing control overhead are used in HyRPL compared to that of P2P-RPL.

Subsequently, we have further investigate the impact of lossy wireless medium on routing for the MP2P traffic traffic. Specifically, the proposed CRPL selects the CPS in the way to minimize the end-to-end cost for each node through a top-down approach. Instead of using one preferred parent, an optimal CPS is used to opportunistically relay the packets. A priority forwarding and overhearing scheme is used for the coordination among nodes in the CPS. Extensive simulation results show that CRPL can efficiently cope with unreliable link to improve the network reliability and reduce packet retransmission significantly.

Chapter 5

An Energy-Efficient Region-based RPL Routing Protocol for Low-Power and Lossy Networks

5.1 Introduction

In the previous chapter, a detailed study on RPL-based routing protocols for P2P communication has been conducted. Different from RPL, which uses the route along the pre-established DAG, P2P-RPL discovers the best quality route and delivers packets with the discovered route. However, the control messages are disseminated throughout the network for optimal P2P route discovery, resulting in considerable energy consumption. HyRPL was proposed to determine the necessity for route discovery by comparing the cost of pre-established route with route constraint. Additionally, as mentioned before, LLNs essentially requires scalable routing support. To address this issue, further investigation on the distributed, reliable and energy-efficient routing approach has been carried out. Particularly, this chapter proposes an energy-efficient region-based routing protocol (ER-RPL), which provides scalable and energy-efficient P2P routing support without compromising the reliability.

Point-to-point (P2P) communication is a fundamental requirement of many

LLNs applications. However, the routing paths between arbitrary M2M devices are not provided by default due to severe resource constraints. The cost associated with route discovery is very expensive in LLNs. Specifically, the route discovery in P2P-RPL and other conventional approaches require all nodes in the network to participate in the temporary DODAG formation, leading to significant cost for resource constrained networks. Owing to the lossy nature and highly energy constraints, it is inherently very challenging to attain high reliability and energy conservation simultaneously. As mentioned in the previous chapter, LLNs suffer from severe constraints such as power consumption, limited memory, unreliable wireless links. Meanwhile, LLNs applications require reliable and stable routing support for generic traffic patterns with heterogeneous node capabilities. It is a critical task to find a reliable routing path in an energy-efficient manner, especially for LLNs. Particularly, energy can be saved by reducing the amount of control overhead, especially in dense networks.

Aiming at 1) supporting generic traffic patterns with the limited memory, 2) achieving reliable routing in dynamic and lossy environment, 3) discovering reliable routing path with severe energy constraints. 4) fulfilling the scalability requirement for large scale networks, 5) resisting asymmetry nature of wireless links in real-world scenarios, ER-RPL is designed to provide reliable and energy-efficient routing support for static networks. In contrast to traditional routing protocols, where all nodes in the network participate in the route discovery process, ER-RPL only requires a subset of nodes in some regions to join the route discovery, which is the key for energy conservation. ER-RPL can discovery reliable routing paths in an energy-efficient manner. Furthermore, with the inheritance of proactively constructing DODAG from RPL, ER-RPL is equipped with the capability to support generic traffic patterns. In addition, ER-RPL takes the asymmetric nature of wireless links into consideration so that it is robust to different wireless channel conditions. In this chapter, the performance of ER-RPL is evaluated with both theoretical analysis and simulation studies. Two conventional routing protocols are used as the benchmarks in the simulation study. In comparison with the benchmarks, ER-RPL can achieve more reliable data delivery with significant energy conservation. Therefore, ER-RPL is proposed with the objective to achieve reliable data delivery in an energy-efficient manner.

The proposed ER-RPL utilizes the region information to support efficient P2P communication. For static networks, such as M2M networks and WSNs, the area where a node resides is a piece of important information. Many LLNs applications employ this region feature. For example, in an event-triggered application, all sensors within the area can capture the event and deliver this information to the end-user, as shown in Fig. 5.1(a). Another typical example is that, in automatic control systems, the control command can be sent to all devices in one level or to a region/room of a building, which is shown in Fig. 5.1 (b). The region information can be used to efficiently discover the routing paths.

Figure 5.1: Two examples of LLNs applications that exploit the region feature.

The Voronoi diagram, which is one of the most elemental structures in the computational geometry, has been extensively investigated and has various applications in a diverse of fields [19]. In the Voronoi diagram, a set of w points partition the plane into w Voronoi regions, as shown in Fig. 5.2. Essentially, the design of ER-RPL employs the concept of region defined in the Voronoi diagram.

Specifically, the contribution of this chapter is fourfold.

- First, a self-regioning algorithm is proposed to allow a node to estimate the region it resides in a distributed manner.
- Second, a P2P traffic model that considers the routing decision and lossy nature of wireless networks is proposed. In addition, theoretical analysis on the reduction of control overhead that ER-RPL can attain is provided.
- Third, a region-based route discovery approach is proposed to discover the optimal route by only requiring a portion of nodes in the network to partici-

Figure 5.2: The Voronoi diagram of a network covered by the Reference Nodes.

pate in the route discovery. Based on the region knowledge, region-to-region routing is designed to support P2P communication without route discovery. Furthermore, the Adaptive Region Selection is proposed to effectively solve the Dead Zone Problem.

- Fourth, to confirm and validate the effectiveness of the proposed protocol, ER-RPL is implemented in NS-3 and experimental studies are carried out to substantially show that ER-RPL can achieve high packet delivery ratio with low control overhead and energy consumption.

The reminder of this chapter is organized as follows. The system model is presented in Section 5.2. Section 5.3 describes the proposed protocol ER-RPL. Theoretical analysis is presented in Section 5.4, followed by simulation study and performance evaluation in Section 5.5. Finally, this chapter is summarized in Section 5.6.

5.2 System Architecture

5.2.1 Preliminary

This network model considers a 2-dimensional space, in which n numbers of stationary nodes are densely deployed on the plane of the network area. The transmission

range of nodes is R . The network can be described as a non-cyclic graph, denoted by $G = (V, E)$, where the node set is denoted by V and the edge set is denoted by E . In the network, N number of nodes with location-awareness capability (e.g., with GPS) are called Reference Nodes (RNs). The nodes without position knowledge are regarded as normal nodes (“nodes” in this chapter). The set of RNs, denoted by Ω , subdivide the plane of the network area into N Voronoi regions, one for each node in Ω . The Voronoi region of the RN I is denoted as $V(I)$. A node i in the network area lies in $V(I)$, if and only if RN I is the closest RN to the node i , and it is denoted by $i \in V(I)$. Furthermore, each of the Voronoi region is divided into a configurable number of non-overlapping regions.

In order to define the most reliable P2P route in LLNs, we perform a mapping for each of the network edges to a positive, real-valued routing cost, which is $c : E \rightarrow \mathbb{R}^+$ to edges. In this way, the network becomes a labelled graph. Furthermore, the cost of the edge $(u, v) \in E$ is denoted as $c(u, v)$. A P2P routing path, Q , with a length of h from the node s to the node d is a sequence of edges with a set of nodes, which is denoted by $\langle u_0, \dots, u_h \rangle$ with $s = u_0, d = u_h$, and $(u_{i-1}, u_i) \in e$ for $1 \leq i \leq h$. In addition, we also define the cost of the path as $c(Q) = \sum_{i=1}^h c(u_{i-1}, u_i)$. Specifically, our work employs the Expected Transmission count (ETX) [153] as the routing metric. Meanwhile, we also denote the successful delivery probability from node u to node v as p_{uv} . One hop ETX means the average number of transmissions required to successfully deliver one packet to the next hop, which is defined as $E_{uv} = 1/p_{uv}p_{vu} \approx 1/p_{uv}$, by assuming that ACK does not suffer from link loss¹. In this way, we have $c(u, v) = E_{uv}$. The cost of a shortest path between two nodes u and v is defined as $c(s, d) = \min \{c(Q) | Q \text{ is a path from } s \text{ to } d\}$, which is the most reliable route between the node u and v . If there is no path from s to d , we define $c(s, d) = \infty$. Essentially, the goal of our proposed ER-RPL is to discover the Q with the minimum cost for a source-destination pair in an energy-efficient manner.

ER-RPL makes use of a small amount of location-aware RNs, but it differs from the geographic routing in three fundamental ways. 1) Different from geographic routing, ER-RPL establishes the best quality route in terms of reliability in a reactive manner. 2) Nodes do not have the unique geographic coordinates

¹ACK normally has small size, and it can be recovered with the strong coding techniques.

Figure 5.3: An example of reference nodes and regions in the network.

knowledge. The region information is only used to determine whether the node should participate the route discovery or not. 3) In ER-RPL, the data delivery relies on the node's address and routing table instead of the virtual or real coordinates. These features make the proposed ER-RPL essentially different from geographic routing.

Two scenarios are considered in our study. In the first scenario, the nodes are nearly uniformly distributed so that the value of ρ is a constant. In the second scenario, the value of ρ is likely to change due to the irregular node deployment. Fig. 5.3 shows four RNs with location-awareness capability are deployed in the network.

5.2.2 P2P Traffic Modelling in LLNs

Packets drop takes place in both node level and link level [144]. The link level channel contention and node level resource limitation will affect the communication quality. In this research study, we assume that there is enough buffer space at the relay node, hence the packet drop rate at an arbitrary node i is approximately 0.

Assume that the link loss is independent and identically distributed (i.i.d). The successful delivery probability from node i to node j can be denoted by p_{ij} . We assume that wireless links have bidirectional readability and have asymmetric

nature, which means p_{ij} may not be equal to p_{ji} for the delivery of packets with the same size. Retransmission is implemented in the Medium Access Control (MAC) layer to improve the network reliability.

The traffic load from source node i to destination node j is denoted by L_{ij} , where $i \neq j$. Then the traffic load distribution matrix of the network is

$$L = \begin{bmatrix} L_{00} & \cdots & L_{0N} \\ \vdots & \ddots & \vdots \\ L_{N0} & \cdots & L_{NN} \end{bmatrix}, \quad (5.1)$$

where for $i \in n$, $L_{ii} = 0$.

For P2P communication, a node needs to choose the next hop among its neighbouring nodes for packet forwarding. This routing decision has significant impact on the network performance. To elucidate this, let δ_{ij} be the routing choice made by node i to node j , where $j \in N(i)$ and $N(i)$ denotes node i 's neighbour node set. If δ_{ij} is equal to 1, it means that node i chooses node j as its next hop. If the value of δ_{ij} is 0, it means that node i does not choose node j as its next hop. We use $r_{i,s}$ to denote the traffic generation rate at node i , and use $r_{i,d}$ to denote the arrival rate of the traffic destined at node i . Assume that the traffic receiving rate of node i from node j is f_{ji} , and the traffic departure rate at node i to node j is f_{ij} . In this way, the P2P traffic model for node i can be formulated as

$$r_{i,s} + \sum_{j \in N(i)} \delta_{ji} f_{ji} - \sum_{j \in N(i)} \delta_{ij} f_{ij} p_{ij} = r_{i,d}, \quad (5.2)$$

where $0 \leq f_{ji} \leq C_{ji}$, $\forall i, j \in n$ and $i \neq j$.

Let us define the network running time as T . At time t , the traffic generation rate of node i is $r_{i,s,t}$. The traffic arrival rate of node i as destination is represented

by $r_{i,d,t}$. The duration of traffic flow is Δt . In this way, the traffic load of node i is shown in Eq. (5.3) and (5.4).

$$\int_0^T r_{i,s,t} \Delta t_i = L_{i0} + L_{i1} + \dots + L_{iN} = \sum_{0 \leq j \leq N} L_{ij}, \quad (5.3)$$

$$\int_0^T r_{i,d,t} \Delta t_i = L_{0i} + L_{1i} + \dots + L_{Ni} = \sum_{0 \leq j \leq N} L_{ji}. \quad (5.4)$$

5.2.3 Energy Model

We model nodes with four basic states (transmit, receive, idle, sleep) and a transition state among them [154]. The energy consumption in transmitting and receiving states is the key issue to address in this research study. We use the First Order Radio Model [155], which has been widely used to measure the energy dissipation for wireless sensor networks (WSNs) [24, 156]. In this model, the radio consumes E_{elec} to run the transmitter or receiver circuitry. The transmitting amplifier is ϵ_{amp} , which is used to achieve the acceptable signal-to-noise (SNR) ratio. We assume the propagation loss exponent is 2. In this way, the energy consumption for transmitting a k -bit message with a transmission range r is modelled as

$$E_{tx}(k, R) = kE_{elec} + kr^2\epsilon_{amp}, \quad (5.5)$$

The energy consumption of the receiver is modeled as

$$E_{rx}(k) = kE_{elec}. \quad (5.6)$$

5.3 An Energy-efficient Region-based RPL for LLNs (ER-RPL)

This chapter proposes a hybrid of proactive and reactive routing protocol, namely ER-RPL, to achieve a reliable data delivery in an energy-efficient manner. This

section presents an overview of ER-RPL, and then demonstrates the key stages in ER-RPL for efficient P2P data delivery.

5.3.1 Overview

ER-RPL inherits the mechanism from RPL to pre-establish the DODAG such that it can support the multipoint-to-point (MP2P) with the optimized topology. Additionally, ER-RPL discovers the best P2P route with the region information in a reactive manner. Both the RNs and nodes are stationary in this work. In ER-RPL, each RN belongs to an area, and that area is divided into a configurable number of non overlapping regions. The region number associated with each RN can be configured with the self-regioning algorithm. Then each node starts to estimate which region it resides in. During the route discovery for a source-destination node pair, ER-RPL requires a node to make a judgement whether it should participate or not with the region knowledge of the source and destination nodes. In this way, the route discovery is only performed among a subset of nodes in the network, which leads to significant control overhead reduction. Basically, ER-RPL includes two main stages.

Network initialization stage: Each RN computes its *Coordinates*, which is defined as the average distance per hop count in this work. Then, each node estimates the distances to all RNs based on RNs' *Coordinates* values and their hop counts to the corresponding RNs. A region code (RC), which represents the particular region a node resides in the network, is obtained by a node with the distributed self-regioning algorithm.

Route discovery stage: According to the RC of source node and destination node, the route discovery is only performed among a subset of nodes in the network with the region-based route discovery. Besides, the Region-to-Region (R2R) routing without route discovery is designed as an enhancement to the region-based route discovery.

Details of these two stages are discussed in the following sections.

5.3.2 Network Initialization Stage

A series of positioning algorithms have been proposed, which are well known as the Ad Hoc Positioning System (APS). APS includes six algorithms: DV-Distance, DV-Hop, Euclidean, DV-Bearing, DV-Coordinate, and DV-Radial [157]. Different from the work in [157], where the *Coordinates* is defined as a correction factor, our work defines the *Coordinates* as the average distance per hop of RNs. ER-RPL uses an enhanced DV-Hop algorithm, which uses the RNs' *Coordinates* and the number of hops to RNs to estimate the distances between a node to the RNs. It is worthy to highlight that the number of RNs in a network is known in advance by all RN nodes. RNs play a critical role in the network initialization stage, but they do not perform any task during the reactive P2P route discovery and data delivery stage.

Coordinates of Reference Node

DODAG_RN	Rank	CFlag	Coordinate	Position
----------	------	-------	------------	----------

Figure 5.4: The packet structure of RFO message.

Each RN computes its *Coordinates* value based on its relative distances and hop counts towards other RNs in the network. Upon receiving the topology formation information initiated by the root, each RN serves as a temporary “root” and builds up a temporary DODAG using the Minimum Hop Count [133] as the routing metric. The construction of temporary DODAG is achieved by disseminating the Region Formation Objec (RFO) message, and the regions are initialized during this process. Fig. 5.4 shows the packet structure of the RFO message. The *DODAG_RN* is the IP address of the RN. The *Rank* is used to update the hop distance to this RN. The *Position* carries the geographical position of this RN. There are two stages of the RFO operation, which is indicated by the *CFlag*. When the *CFlag* is set to be zero by the RN, it indicates that the current stage is the *Coordinates Computation* stage of this RN. When the *CFlag* is set to be one, it is in the *Euclidean Distance Calculation* stage. The *Coordinate* field records the *Coordinates* of the corresponding RN. The temporary DODAG rooted at RN *I* is denoted as *DODAG_RN_I*, where *I* is an arbitrary RN in the network area. A

node will first check the *CFlag* field when it receives an RFO message from its one hop neighbour node.

Figure 5.5: Four temporary DODAGs rooted at each RN are built during network initialization stage.

During the *Coordinates Computation* stage, a node will join the temporary DODAG, choose the next hop, update its rank, and broadcast its status with an RFO message. Similar to RPL basic DODAG set up, the rank is monotonically increased based on next hop towards that RN. The RFO operation also follows the Trickle Algorithm. In this way, the information will be dismissed throughout the whole network. Based on the control message exchange and rank value update, each RN is able to collect the information including other RNs' positions and the distances between other RNs in the network. N temporary DODAGs will be constructed with N RNs in the network, as shown in Fig. 5.5. Assume that the position of RN I is (x_I, y_I) . The Euclidean distance between two arbitrary RN I and RN J , is calculated using $d_{IJ} = \sqrt{(x_I - x_J)^2 + (y_I - y_J)^2}$. We use h_{IJ} to denote the hop count of RN I to RN J . When an RN joins the DODAG rooted by another RN, it records the position information as well as the hop count associated with that RN. Once an arbitrary RN I has the distance and hop count information

of the other $(N - 1)$ RNs, its *Coordinates* can be calculated through

$$C_I = \frac{\sum_{J \in \Omega, J \neq I} d_{IJ}}{\sum_{J \in \Omega, J \neq I} h_{IJ}}. \quad (5.7)$$

After a RN get its *Coordinates* value, it update *CFlag* field as one and broadcast the RFO message with its *Coordinates* value.

During the *Euclidean Distance Calculation* stage, a node is able to get each RN's *Coordinates* value and the hop count to that RN through joining the temporary DODAGs, the information of which is used for the self-regioning algorithm. The *Coordinates* measurement of RN does not depend on the transmission range of nodes, the advantage of which is the simplicity. However, the accuracy depends on the isotropy property of the network deployment. The network graph of an isotropic network has similar distribution in different directions. So the estimation of distance per hop works more accurately when nodes are deployed evenly.

The Distributed self-regioning Strategy

Based on the update of RFO message during the *Euclidean Distance Calculation* stage, a node is able to collect the information of each RN's *Coordinates* value and the hop count associated with that RN. With that information, the distances of a node to all RNs can be calculated. For an arbitrary node i , the Euclidean distances to every RN in the network are denoted by $\{d_{iI}\}$, where $I \in \Omega$. The distance between node i to RN I can be estimated with

$$d_{iI} = C_I h_{iI}, \quad (5.8)$$

Through comparing its estimate distances to all RNs, a node can find out its closest RN. We use *RegionNumber_{RN}* to represent the particular region under one RN. There are multiple regions covered by each RN. When the area of one RN is segmented into four regions, whose locations are the upper left segment, lower left segment, upper right segment, and lower right segment, the region number will be *Region* I, II, III, IV, respectively, as shown in Fig. 5.6. In Fig. 5.6(a), d_{iA} is the minimum one among the distances to four RNs ($d_{iA}, d_{iB}, d_{iC}, d_{iD}$), so node i is inferred to reside in the region under RN A , which is denoted by $i \subseteq R_A(I, II, III, IV)$.

Then a self-regioning algorithm using the Law of Cosines identifies which region the node belongs to, and a RC will be assigned to the node accordingly. Fig. 5.6

Figure 5.6: The distributed self-regioning algorithm: (a) Step I. (b) Step II.

shows an example of the distributed self-regioning algorithm with four RNs in the network. The Law of Cosines is used to compute the values of $\angle\alpha$, $\angle\beta$, and $\angle BAC$ as follows.

$$\angle\alpha = \angle iAB = \arccos \frac{d_{iA}^2 + d_{AB}^2 - d_{iB}^2}{2d_{iA}d_{AB}} \quad (5.9)$$

$$\angle\beta = \angle iAC = \arccos \frac{d_{iA}^2 + d_{AC}^2 - d_{iC}^2}{2d_{iA}d_{AC}} \quad (5.10)$$

$$\angle BAC = \arccos \frac{d_{AB}^2 + d_{AC}^2 - d_{BC}^2}{2d_{AB}d_{AC}} \quad (5.11)$$

Through comparing the value of $|360 - \alpha - \beta - BAC|$, $|\beta - \alpha - BAC|$, $|\alpha - \beta - BAC|$, $|\alpha + \beta - BAC|$, node i is considered to reside in IV_A because $|\alpha + \beta - BAC|$ gets the minimum value. The region code (RC) of IV_A , refers to the lower right segment of RN A , will be assigned to node i . The RC is very important in ER-RPL, which is a 8-bits code. The first four bits are represented by the region number and the last four bits are represented by the RN's ID. The ID of an RN is an unique integer. For example, 0000, 0001, 0010, and 0011 can be used to represent *Region* I, II, III, and IV, respectively. If node i is located at *Region* IV_A and the ID of RN A is 5, which is 0101 in binary, then node i 's RC will be 00110101, as shown in Fig. 5.6.

With the planned deployment of N RNs in the network, a 2D tessellation with N cells can be formed. Each cell contains one RN. The number of RNs is a

composite number, $r \times t = N$, $\forall r, t \in \mathbb{Z}_+$, where r is the row number and t is the column number of the tessellation.

Definition 1 *The reference node map, $RNM = (RN_{ij})$, is defined as a $r \times t$ matrix with RNs' ID, where $0 \leq i \leq r$, $0 \leq j \leq t$.*

When four regions are associated with each RN, four RCs can be generated from every element of a Reference Node Map (RNM). In this way, the Region Code Map (RCM), which consists of all RCs in the network, can be computed based on the RNM according to the *Def. 2*.

Definition 2 *Based on the RNM, the RCM is defined as a $2r \times 2t$ matrix. $RCM = (RC_{mn})$. Each element of RNM, RN_{ij} , is used to generate four elements of RCM, which are $RC_{2i,2j}$, $RC_{2i,2j+1}$, $RC_{2i+1,2j}$, and $RC_{2i+1,2j+1}$.*

The RNM represents RNs' relative position of the cells. In the network initialization period, nodes obtain the geographical location of each RN to build the RNM. In Fig. 5.6(a), the ID of RN *A*, *B*, *C*, and *D* in binary are 0101, 0110, 0111, and 1110, respectively. The RNM is

$$RNM = \begin{bmatrix} 0101, 0111 \\ 0110, 1110 \end{bmatrix}, \quad (5.12)$$

The *Region* I, II, III, and IV are represented by 0000, 0001, 0010, and 0011, respectively. Therefore, the RCM of this network is

$$RCM = \begin{bmatrix} 00000101, 00100101, 00000111, 00100111 \\ 00010101, 00110101, 00010111, 00110111 \\ 00000110, 00100110, 00001110, 00101110 \\ 00010110, 00110110, 00011110, 00111110 \end{bmatrix}, \quad (5.13)$$

The pseudo-code of the distributed self-regioning algorithm is shown in Algorithm 1.

Algorithm 1: Distributed Self-regioning Algorithm.

Data: $\{d_{iI}\}, \{d_{IJ}\}$, where $i \in N$, $RN\ I, J \in K$, and $I \neq J$. $RN\ A, B, C$, and D is $RN_{i,j}, RN_{i+1,j}, RN_{i,j+1}$, and $RN_{i+1,j+1}$ element of RNM , respectively.

```
1 if  $\{d_{iA}\} = \min\{d_{iI}\}$  then
2    $i \subseteq R_A(I, II, III, IV)$ ;
3   if  $(d_{iA} + d_{iB} > d_{AB})$  then
4     if  $(d_{iA} + d_{iC} > d_{AC})$  then
5       return  $compare\_angles(\angle iAB, \angle iAC, \angle BAC)$ 
6     else if  $\angle iAB > \angle BAC$  then
7       return  $III_A$ 
8     else
9       return  $IV_A$ 
10  else if  $(d_{iA} + d_{iC} > d_{AC})$  then
11    if  $\angle iAC > \angle BAC$  then
12      return  $II_A$ 
13    else
14      return  $IV_A$ 
15  else
16    return  $IV_A$ 
17 else if  $\{d_{iA}\} = \{d_{iB}\} = \min\{d_{iI}\}$  then
18    $i \subseteq R_A(II, IV) \cup R_B(I, III)$ ;
19   if  $d_{iC} < d_{iD}$  then
20     if  $\angle iAC > \angle BAC$  then
21       return  $II_A$ 
22     else
23       return  $IV_A$ 
24   else if  $\angle iAC > \angle BAC$  then
25     return  $I_B$ 
26   else
27     return  $III_B$ 
28 else if  $\{d_{iA}\} = \{d_{iB}\} = \{d_{iC}\} = \min\{d_{iI}\}$  then
29   return  $IV_A$  ( $d_{DA} > d_{DB}, d_{DA} > d_{DC}$ )
30 else
31   return  $IV_A$  or  $III_B$  or  $II_C$  or  $I_D$ 
32 Function  $compare\_angles(\alpha, \beta, \gamma)$ 
33    $\delta = \min\{|360 - \alpha - \beta - \gamma|, |\beta - \alpha - \gamma|, |\alpha - \beta - \gamma|, |\alpha + \beta - \gamma|\}$ 
34   if  $\delta = |360 - \alpha - \beta - \gamma|$  then
35     return Region I
36   else if  $\delta = |\beta - \alpha - \gamma|$  then
37     return Region II
38   else if  $\delta = |\alpha - \beta - \gamma|$  then
39     return Region III
40   else
41     return Region IV
```

Rational of Reference Node and Region

It is important to notice that the number of regions per RN covers may vary. There is no overlapping regions, which means one region can only belong to one RN. More regions can be created if more RNs are involved in the self-regioning algorithm. Besides, if a node has equal distance to two or more RNs, an additional RN needs to get involved in the self-regioning algorithm. Algorithm 1 includes the equal-distance case and addresses how to select the additional node.

Consider each region is with the same node density. After a node finds out its closest RN, four regions per RN can be formed if the node uses another two RNs for the self-regioning algorithm. And the shape of each region is approximately a rectangle. Similarly, six regions per RN can be formed if three neighbour RNs are used, and the region shape is close to a hexagon. Octagonal regions will be formed if another four neighbour RNs are used for the self-regioning algorithm. After identifying the closest RN, if another ε neighbour RNs are involved in the self-regioning algorithm, each RN can have 2ε regions. In this way, there are $2\varepsilon N$ regions when there exists N RNs in the network.

Generally speaking, with more numbers of RNs used in the self-regioning algorithm, there will be more regions in the network. ER-RPL explores the optimal P2P route among only a subset of regions in the network. When the network is segmented to more regions, each region will have a smaller size and there tends to be a smaller subset of nodes participating in the route discovery on average. In this way, a greater reduction of control overhead can be achieved with an increased number of RNs used in the self-regioning algorithm. The impact of the number of regions on the reduction of overhead that can be achieved by ER-RPL, the analysis of which is presented in Section 5.4.

5.3.3 Reactive P2P Route Discovery

In this section, we present the routing design of the proposed protocol. ER-RPL comprises three main components for the reliable and energy-efficient data delivery.

1) During the region-based route discovery, a submatrix of the region code map (RCM) is computed based on the source node's region code (SRC) and the desti-

nation node's region code (DRC). Upon receiving a route request, a node can make a decision whether it should join or ignore a route discovery process by checking its RC's existence in that submatrix of the RCM. Then the temporary DODAG is formed among only a portion of nodes in the network. 2) The Region-to-Region (R2R) routing without route discovery is designed as an enhancement. The R2R route refers that there is an available route for any node in one region to any node in another region. The R2R route needs to go through an intermediate node, which is located in one of these two regions. The R2R routes are enable during the a route discovery for one source-destination node pair, which are the routes between nodes in the source node's region and nodes in the destination node's region. Besides, the route adjustment on-the-fly is implemented to shorten the R2R routes. 3) Since nodes are randomly deployed in the network plane, the node density may vary from region to region. Sometimes nodes are sparsely deployed in some region. Besides, the transmission range of nodes are limited. In this case, nodes in some regions maybe not able to provide the connectivity to their neighbour regions, which is defined as the dead zone problem. We propose an Adaptive Region Selection (ARS) solution for the dead zone problem.

Region-based P2P Route Discovery

Given a source-destination node pair, a submatrix of RCM is generated and defined as the IRCM, which composes a subset of the RCs in the network. The IRCM is used to determine whether a node should join the route discovery or not. Nodes, whose RC is the element of IRCM, are supposed to participate the route discovery for the source-destination node pair. Compared to the network size, ER-RPL selects a smaller area for the route discovery with the help of IRCM. The size of that area is not more than the network area. The regions of the source and destination nodes are located in the diagonal of the selected area. The shortest (most reliable) routing path for the source-destination pair will be chosen for the data delivery. Assume that RC_{m_s, n_s} and RC_{m_d, n_d} denote the SRC and DRC in the RCM, respectively. Then the IRCM is a $(|m_s - m_d| + 1) \times (|n_s - n_d| + 1)$ matrix, and the $IRCM(u, v)$ has equal entry to $RCM(u + \min(m_s, m_d), v + \min(n_s, n_d))$, where $0 \leq u \leq (|m_s - m_d| + 1)$, $0 \leq v \leq (|n_s - n_d| + 1)$.

With the RCM shown in Eq. (5.13). If the SRC is 00110101 and the DRC is

00011110, the *IRCM* is

$$IRCM = \begin{bmatrix} 00110101, 00010111 \\ 00100110, 00001110 \\ 00110110, 00011110 \end{bmatrix}. \quad (5.14)$$

The Message Request Object (MRO) is designed as the control message for ER-RPL, the packet structure of which is shown in Fig. 5.7. The *Start Point Address* and *End Point Address* record the source and destination IP addresses, respectively. The *Lifetime* indicates how long the temporary DODAG will exist. But if the source node receives a hint from the upper layer, the lifetime can be extended or shortened. The end-to-end cost in terms of ETX is recorded in the *Accumulated ETX* field. The hop count towards the destination is stored in the *Hop* field. Different types of MRO messages are distinguished through the *type* field. MRO(0) denotes the Message Request Object with type 0. MRO(1) denotes the Message Reply Object with type 1. MRO(2) denotes the Region-to-Region MRO with type 2. MRO(3) denotes the Region-based DAG MRO with type 3.

Type	Start Point Address	End Point Address	Sequence	Hop Count	Aggregate Cost
------	---------------------	-------------------	----------	-----------	----------------

Figure 5.7: Packet structure of MRO.

For a P2P route request, the source node will send a MRO(0) through the existing DAG (via the root) to the destination node. MRO(0) records the accumulated cost and hop count along this path. Upon receiving MRO(0), the destination node can decide whether to use this existing route or explore a better route based on the route constraints². If the existing route's cost is within the route constraints, MRO(1) is sent from the destination node to the source node. Otherwise, this route is deemed as unsatisfactory. Then the destination node will check whether the routes between the source node's region and the destination node's region are available or not. If the R2R route has been enabled, there exists a route from the

²The work in [151] describes how to measure the P2P routing metrics in detail.

destination node to the source node through an intermediate node. The source node will send the MRO(2) to the intermediate node through the best quality path. That intermediate node has the routing path to any node in the source node's region, and it will forward the MRO(2) to the source node through the optimal routing path. Once the source node receives the MRO(2), the reverse route will be used for data delivery. However, if the existing route is unsatisfactory and there is no R2R route available, the destination node will initiate the temporary DODAG construction as the root with MRO(3), in which the SRC and the DRC are piggybacked.

Once a node receives the MRO(3) from its neighbour nodes, a node computes the IRCM based on the SRC and DRC first. Then a node checks whether its RC exists in the IRCM or not. If it is so, the node will join the route discovery with the following steps. 1) The node temporarily records the sender's ID with its end-to-end cost and hop count corresponding to the destination node in its local table. 2) The node chooses the neighbour node that gives the minimum accumulated cost to the destination as its preferred parent. 3) The node computes its cost towards the destination, update its hop count and broadcast the MRO(3) to its neighbours. In this way, a temporary DODAG is constructed in the way that nodes always select the best quality path towards the destination. During route discovery, nodes in a subset of regions choose the best route to the destination given particular routing metrics. Once the source node receives the route discovery message, it will send data to the destination node through the optimal route discovered by the DODAG construction.

Assume that M denotes the total number of regions, which is equivalent to $(2r \times 2t)$, and K denotes the number of the subset of regions that involve in the route discovery, which is equivalent to $(|m_s - m_d| + 1) \times (|n_s - n_d| + 1)$. The complexity of our proposed protocol is $\mathcal{O}(M + M + K)$, where $K < M$. As N is very small ($N \ll n$), the complexity of our algorithm is low, so that our proposed algorithm is comparable with the existing protocols, such as RPL and P2P-RPL.

Region-to-Region Routing without route discovery

During the region-based route discovery process, the best quality path towards the destination is selected by each node in the regions identified by the IRCM. Meanwhile, nodes in the same region with the source or destination node, send their routing information (hop-by-hop or source routing) to the destination node. In this way, the downwards route from the temporary root (destination node in this routing pair) to all nodes in the source node's region and destination node's region will be enabled.

Figure 5.8: Region-to-Region routing without route discovery. (a) The routing between *Region II_A* and *Region I_D* is enabled. (b) R2R routing is used for s_2 and d_2 . (c) The intermediate node, d_1 , can locate either in s_2 's region or d_2 's region. (d) R2R routing with route adjustment.

The DODAG root, which is the destination node in this P2P pair, is used as an intermediary to support the communication between the source node's region to the destination node's region. Besides, in order to enable the R2R routing, every

node keeps a R2R table, each row of which stores one node id, two RCs and the lifetime for this row. In order to keep the R2R table small, only nodes with SRC or DRC create the entry. Upon receiving the MRO(3) that carries the SRC and DRC, a node will store the SRC and DRC pair corresponding to the destination node (temporary root) in its R2R table. Upon receiving a P2P routing request, the destination node will examine its R2R routing table before the route discovery. If the route to source node's region is indicated as available, the destination node will send a reply MRO(2) via the corresponding DODAG root to the source node.

Fig. 5.8 illustrates an example for the R2R routing with ER-RPL. In this figure, the solid arrow is used to indicate the route which MRO(2) travels along. The dash arrow indicate the route for data delivery. As shown in Fig. 5.8(a), the routing between nodes in *Region II_A* and *Region I_D* are enabled during the route discovery from destination node d_1 to source node s_1 . All nodes in *Region II_A* and *Region I_D* update their route information to node d_1 and store the information of (d_1, I_D, II_A) with its expire time in this R2R table. Node d_1 will temporarily record the routes to all nodes in *Region IV_A* and *Region I_D*.

Fig. 5.8(b) shows a source-destination node pair, i.e., s_2 - d_2 . The destination node d_2 checks its R2R table, in which the entry (d_1, I_D, II_A) indicates that there is an existing route to nodes in *Region I_D* via the intermediary node d_1 . Instead of route discovery, the destination node d_2 replies with a MRO(2) to the node d_1 . With the routing information to s_2 , d_1 will forward the MRO(2) to node s_2 . Once node s_2 receives the MRO(2), the reverse route of MRO(2) will be used for data delivery.

The R2R routing without route discovery can generally be applied no matter the intermediate node is located in the source or destination node's region. The example can be found in Fig. 5.8(b) and 5.8(c). In both cases, the entry for the source node region in R2R is (d_1, I_D, II_A) , and node d_1 performs as the intermediate node. Although the routing paths from the source and destination node to the intermediate node are the best quality paths, the R2R route may not be the optimal one due to the existence of the intermediary. There occurs some extra hops within one region when adopting R2R routing. But it is lesser significant when the route tends to be long between the source region and the destination region. With a fixed network size, the number of nodes in each region is lesser

when there are more numbers of regions. In this way, the sub-optimal route will be closer to the optimal route and the R2R table size become smaller. However, to get more regions in the network, it requires more number of RNs leading to an increased computation complexity.

Route Adjustment On-the-fly: We propose a route adjustment “on-the-fly” scheme to shorten the R2R route and alleviate the sub-optimal routing path effect. When a node receives or overhears a MRO(2), it always chooses the one that provides smaller hop count as the next hop to the destination node. Without bringing any additional control message, this scheme allows a node to choose its preceding node instead of the one it receives the MRO(2) from. There is no additional control messages in this scheme.

In Fig. 5.8(b), node m overhears the MRO(2) from node n , whose hop count to d_2 is one. After node m receiving the MRO(2) from node d_1 , whose hop count to d_2 is two. Node m will choose node n as the next hop, so that its hop count to d_2 is two. Comparing 5.8(b) with 5.8(d), the hop count is effectively reduced with this scheme.

Adaptive Region Selection for the Dead Zone Problem

In the previous section, we assume that the density of nodes in each region is approximately a constant. However, due to the irregular node deployment, the node density may vary from region to region. Let ρ_A denote the node density of *Region A*. It is known that a sufficient density is a critical requirement to ensure almost surely (a.a.s.) network connectedness [158].

Let us consider n nodes placed uniformly and independently in a 2D network area. For the sake of simplicity, the region shape is square with a size of $L_I \times L_I$. Then each region is considered to be divided into square size cells, each edge of which is $\sqrt{2}R/2$. For region I, the node density required for a.a.s. connectedness is defined in the following theorem:

Theorem 1 $\rho_I \geq \frac{\alpha \ln L_I}{r^2}$, for some $\alpha > 0$, $\lim_{L_I \rightarrow \infty} P_{conn}(L_I) = 1$, $P_{conn}(L_I)$ denotes the probability that *Region I* is connected.

This theorem provides the necessary conditions to ensure asymptotically a.a.s.

Figure 5.9: Illustration of Dead Zone problem.

region connectivity. A similar theorem is proved in [156]. So the proof is omitted here. Although this theorem theoretically provides the required density for the network connectedness, an effective routing solution to check the connectedness is also needed in practice.

As we know, the region connectivity is impaired if the network density is too low. When some regions are with sparse node distribution, these regions may not be able to provide the connectivity for their neighbor regions. The dead zone problem arises when the P2P route is discovered among such regions. In Fig. 5.9, when the destination node needs to discover a route to the source node, nodes in *Region B*, *Region E*, and *Region H* will participate in the route discovery process. Due to the limited transmission range and sparse node deployment, nodes in *Region E* are not able to establish a route for *Region B* and *Region H*. Therefore, *Region E* is considered as a dead zone for *Region B* and *Region H*. However, the routing path can be successfully provided, if nodes in *Region D* join the route discovery process.

We propose an Adaptive Region Selection (ARS) scheme to provide an efficient solution for the dead zone problem. During the network initialization stage, nodes update their RCs and one hop neighbor information through their preferred parent nodes to the root, so that the root node has a global knowledge of the network topology. A similar approach is defined in RPL [7], where the Destination Advertisement Object (DAO) is used to propagate upwards along the DAG to the root, so that the downwards route from the root is enabled. The root usually has a strong computation capability and large memory space, which can store RNs'

Coordinates, nodes' RCs and their one hop neighbour list. Upon receiving the MRO(0) for a P2P routing request, the root will compute the IRCM and perform a neighbor list checking, so that it can determine whether the source node and destination node are reachable within the regions from IRCM. If the root detects that those regions cannot provide the connectivity, it will use a flag to indicate the dead zone problem in MRO(0). Upon receiving the MRO(0) with the dead zone indication, the destination node will piggyback this information in the route discovery message (MRO(3)). Instead of IRCM, the Expanded IRCM (EIRCM) will be used in this case.

Definition 3 Assume that SRC and DRC are the RC_{m_s, n_s} element and the RC_{m_d, n_d} element in the RCM, respectively, where the RCM is a $2r \times 2t$ matrix. The EIRCM is defined as a $(m_{dd} - m_{ss} + 1) \times (n_{dd} - n_{ss} + 1)$ submatrix of RCM, where m_{ss} , n_{ss} , m_{dd} , and n_{dd} can be computed according to Eq. (5.15). $EIRCM(u, v)$ has entry equal to $RCM(m, n)$, where $m = u + m_{ss}$, and $n = v + n_{ss}$.

$$\begin{aligned}
 m_{ss} &= \begin{cases} \min(m_s, m_d) - 1, & \text{if } \min(m_s, m_d) \neq 0 \\ 0, & \text{otherwise,} \end{cases} \\
 n_{ss} &= \begin{cases} \min(n_s, n_d) - 1, & \text{if } \min(n_s, n_d) \neq 0 \\ 0, & \text{otherwise,} \end{cases} \\
 m_{dd} &= \begin{cases} \max(m_s, m_d) + 1, & \text{if } \max(m_s, m_d) \neq m \\ m, & \text{otherwise,} \end{cases} \\
 n_{dd} &= \begin{cases} \max(n_s, n_d) + 1, & \text{if } \max(n_s, n_d) \neq n \\ n, & \text{otherwise.} \end{cases}
 \end{aligned} \tag{5.15}$$

During the region-based route discovery process with ARS, a node will compute EIRCM instead of IRCM after receiving MRO(3) with the dead zone problem indication. If a node finds its RC is an element in the EIRCM, it will join this temporary DODAG for the P2P route discovery. In this way, more regions are involved in the route discovery process so that it greatly increases the chance for identifying an available routing path from the source node to the destination node.

5.4 Model Validation

The trickle algorithm is used in RPL, P2P-RPL and ER-RPL to achieve both rapid propagation and low maintenance overhead. Three key parameters are used to define the interval of broadcasting control messages [135]; the minimum time interval size I_{min} , the maximum time interval size I_{max} , and the redundancy counter c . The current communication interval is denoted as I . When a node starts a trickle timer, it resets c to 0 and the control messages update time, which is denoted by t , to a random value between $I/2$ to I . Whenever a node hears a transmission that is “consistent”³, it increments the counter c . The trickle algorithm doubles the current communication interval, I , when the interval I expires. Any “inconsistent” message will cause trickle timer to reset. Until the interval length is greater than I_{max} , I is set to be I_{max} . In other words, the time interval increases during the network convergence time. When the interval I reaches I_{max} , the transmission rate of control messages, f_c , follows

$$f_c \approx \frac{1}{I_{min}2^{I_{max}}}. \quad (5.16)$$

Assume that the number of traffic flows is denoted by λ . In P2P-RPL, all nodes will participate in the route discovery. When λ traffic flows require the route discovery, where $\lambda < n(n-1)/2$, the control overhead of P2P-RPL for time t is

$$O_{p2p-rpl} = n\lambda f_c t = \frac{n\lambda t}{I_{min}2^{I_{max}}}. \quad (5.17)$$

The route discovery may not be necessary under certain circumstances. Nodes can have direct communication to their one-hop neighbours. By assuming that nodes’ one-hop neighbours number is $\pi R^2\rho$, the ratio of one-hop data delivery is

$$\varphi_{nb} \approx \frac{C_n^1(\pi R^2\rho - 1)}{n^2} = \frac{\pi R^2\rho - 1}{n}. \quad (5.18)$$

Nodes’ routing tables usually contain some valid routes. The ratio of the number of valid routes to the number of nodes in the network is denoted as φ_{rt} . The value of φ_{rt} is affected by many factors, such as the routing table size, traffic flows,

³A control message (DIO, P2P-DRO, and MRO(3) in RPL, P2P-RPL, and ER-RPL, respectively) from a sender with a lesser DODAG rank that does not cause any changes to the recipient’s parent set, preferred parent, or rank is considered consistent. Otherwise, the control message is inconsistent.

and the routing lifetime. The number of one-hop neighbours can be approximately considered as the number of valid routing entries due to the limited memory space of nodes. With the i.i.d. source-destination node pairs, the control overhead of P2P-RPL for λ traffic flows during time t is

$$O_{p2p-rpl} \approx \frac{(1 - \varphi_{rt})n\lambda t}{I_{min}2^{I_{max}}} \approx \frac{(1 - \varphi_{nb})n\lambda t}{I_{min}2^{I_{max}}}. \quad (5.19)$$

In ER-RPL, only a subset of nodes in the networks will participate in the route discovery. For simplicity, we consider a dense network, in which the node density, ρ , of each region is the same. We also assume the route discovery is needed to support the communication between two arbitrary nodes in the network. For a network containing k^2 regions, the ratio of nodes that participate in route discovery and consume overhead to total nodes is denoted as $\varphi(k, \rho, n)$, where $0 < \varphi(k, \rho, n) \leq 1$. For a $k \times k$ RCM, the SRC is $RCM(m, n)$, and the DRC is $RCM(a, b)$, where $a \in [0, k-1]$, $b \in [0, k-1]$. Then IRCM is a $(|m-a|+1) \times (|n-b|+1)$ matrix. In this case, the value of $\varphi(k, \rho, n)$ is $(|m-a|+1)(|n-b|+1)/k^2$. When the source node and destination node are i.i.d., $\varphi(k, \rho, n)$ will no longer depend on ρ and n as the number of traffic flows λ tends to ∞ , which is represented by $\varphi(k)$. In this case, we can have

$$\varphi(k) \approx \frac{\sum_{a=0}^{k-1} \sum_{b=0}^{k-1} \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} (|m-a|+1)(|n-b|+1)}{k^6}. \quad (5.20)$$

Similarly, it is not always necessary to discover routes because of the one-hop neighbours and valid routes in the routing table. The expectation value of $\varphi(k, \rho, n)$ is $\varphi(k)$, which does not consider the valid routing entries in the routing table. So we can get the following,

$$\varphi(k, \rho, n) \approx \varphi(k) + \varphi_{rt} \approx \varphi(k) + \varphi_{nb}. \quad (5.21)$$

So theoretically, the control overhead for ER-RPL based on the previous conditions is

$$O_{er-rpl} \approx \frac{\varphi(k, \rho, n)n\lambda t}{I_{min}2^{I_{max}}}. \quad (5.22)$$

Therefore, with the comparison of P2P-RPL, the reduction of control messages that ER-RPL can be obtained by comparing Eq. (5.19) and Eq. (5.22). Thus, the

performance ratio of the reduction achieved by ER-RPL to P2P-RPL, denoted by δO , is

$$\delta O = \frac{O_{p2p-rpl} - O_{er-rpl}}{O_{p2p-rpl}} = \frac{1 - \varphi(rt) - \varphi(k, \rho, n)}{1 - \varphi(rt)}. \quad (5.23)$$

5.5 Network Simulation

5.5.1 Experimental Setup

We implement RPL, P2P-RPL and ER-RPL from scratch on Network Simulator 3 (NS3) [17]. The error erasure channel is used to model the lossy environment. Each wireless link is assigned with a random probability p_{ij} , where $0.3 \leq p_{ij} \leq 0.8$. Many research studies have provided solutions to measure the wireless link quality [159, 160], which is out of the scope of this work. We assume that the link layer will handle this issue. Four RNs are used in this study. Each RN will have four regions with the distributed self-regioning algorithm. Both symmetric links ($p_{ij} = p_{ji}$) and asymmetric links ($p_{ij} \neq p_{ji}$) are considered, which are denoted by s and as , respectively, in the performance analysis section. For instance, for a particular protocol, $ER-RPL_s$ and $ER-RPL_{as}$ denote the routing performance with symmetric and asymmetric links, respectively. RPL is in non-storing mode. Our simulation studies are conducted with the IEEE 802.11. It is worth to highlight that IEEE 802.11 is usually not considered as the best candidate for LLNs, while IEEE 802.15.4 [161] is viewed as the optimal choice for LLNs. But the routing protocols' general behavior, such as path-length, packet delivery, control overhead, etc, can be concluded from the simulation study with IEEE 802.11 [115], especially for networks with a low data rate. A detail performance comparison of IEEE 802.11 and IEEE 802.15.4 can be found in the work of [162].

The efficiency of route discovery and efficiency of data delivery are two main performance to evaluate the effectiveness of the routing protocols for P2P communication. Specifically, the frequency of control message broadcasting significantly affect the amount of routing overhead used in the route discovery and the latency for the route discovery. Thus, the performance under different frequency of control message broadcasting is studied in Scenario I to evaluate the efficiency of route

discovery. More specifically, in Scenario I, the maximum time interval I_{max} are varied from 5 to 8 and traffic flows are set to 30. The routing performance under different P2P traffic flows is important to evaluate the effectiveness of route discovery. To validate whether or not the discovered routes can reliably deliver packets, it is essential to evaluate the routing protocol's performance under different traffic flows. Thus, the performance under different traffic flows is studied in Scenario II to evaluate the efficiency of data delivery. In Scenario II, the number of P2P traffic flows are varied from 10 to 40, and I_{max} is set to 8. In both scenarios, two types of node deployment strategies are studied. Without loss of generality, 100 nodes are randomly deployed in a $180m \times 180m$ network area with the random planar deployment strategy, and we also study the 10×10 grid deployment with some randomness for the constant node density scenario. The node's position is $(20b \pm \Delta x, 20r \pm \Delta y)$, where $-5 \leq \Delta x \leq 5, -5 \leq \Delta y \leq 5$. Assume that the node ID is U , the value of b and r are obtained through $U = 10b + r$, where b, r is $\mathbb{N}$, and $r < 10$.

In this simulation study, the source and destination node pair is randomly selected for each traffic flow. Each traffic flow lasts for 90 seconds and then another P2P flow starts. In this way, the number of concurrent traffic flows are kept the same but there are different source and destination routing pairs generated during the simulation. Other parameters are displayed in Table 5.1. A different random topology is generated in each run and each data of the results is the average value from ten runs.

Table 5.1: Simulation parameters to evaluate the performance of ER-RPL.

Parameter	Value	Parameter	Value
Number of nodes	100	Number of reference nodes	4
Retransmission limit	5	Communication range	35m
Packet size	512 bytes	Traffic rate	4 pkt/s, CBR flow
Routing metric	ETX	Simulation time	1000s
Transmitter electronics (E_{elec})	50nJ/bit	Transmit amplifier (ϵ_{amp})	100pJ/bit/ m^2
The Minimum time interval size (I_{min})	50ms	Redundancy constant (c)	∞

5.5.2 Performance Metrics

The following metrics are adopted to evaluate the performance of the proposed protocol and the benchmarks.

1. *Packet delivery ratio* refers to the ratio of the number of packets successfully delivered to the destination nodes to the number of packets generated by the source nodes. This metric illustrates the routing reliability.
2. *Normalized routing control overhead* refers to the ratio of the number of control messages to the number of data successfully delivered to destinations.
3. *Energy consumption per data successfully delivered* is the ratio of the total energy consumption to the number of data packets that are successfully delivered to the destination nodes. The total energy consumption of the network includes the energy spent for all phases of the network during the simulation.
4. *Average hop count* refers to the average number of hops for the discovered route between the source nodes and the destination nodes.
5. *Average end-to-end delay* includes all possible delay during data transmission due to transmission time, retransmission and queuing time.

5.5.3 Performance Evaluation

Figure 5.10: Scenario I: The PDR comparison. (a) grid with random deployment. (b) random planar deployment.

As shown in Fig. 5.10, ER-RPL improves the packet delivery ratio (PDR) by 150% compared to RPL. ER-RPL_s achieves a very close performance to P2P-RPL with symmetric links (P2P-RPL_s), which is the optimal value with symmetric links. It shows that although only partial nodes are involved in the P2P route discovery process, ER-RPL can still choose the nearly optimal route to provide reliable routing, which is one of our goals in this work. With the region-based route discovery, ER-RPL_{as} can maintain the performance and outperform P2P-RPL with asymmetric links (P2P-RPL_{as}) by around 10%. In P2P-RPL, the P2P route is discovered throughout the whole network. Therefore, the optimal route from the destination node to the source node will be selected by P2P-RPL. The PDR degrades in P2P-RPL if the wireless link becomes asymmetric. Because in P2P-RPL, the temporary DODAG is rooted at the source node, so that the route for data delivery may not be the optimal one from the source to the destination under asymmetric links. Different from P2P-RPL, whose temporary DODAG is rooted at the source node, ER-RPL forms the temporary DODAG at the destination node, so the best route from the source to destination is always used.

Figure 5.11: Scenario I: Control overhead of simulation and theoretical results. (a) grid with random deployment. (b) random planar deployment.

Fig. 5.11 shows that the routing control overhead for route discovery decreases with the increment of the maximum time interval size (I_{max}). ER-RPL achieves average 60% less control overhead compared to P2P-RPL. The control overhead of P2P-RPL_{as} is larger than P2P-RPL_s due to fewer packets received by the destinations. RPL uses the pre-established route for the data delivery, so the route discovery is not required and we consider that no additional control messages oc-

cur in this case. Fig. 5.11 also depicts the theoretical results with Eq. (5.19) and (5.22). With a short maximum time interval, the network reaches the convergence stage quickly. So the effect of high frequency during the network initialization stage is negligible. Meanwhile, due to the valid routing entries, the overhead is slightly lower in the simulation than the theoretical results, which regards the number of one-hop neighbours as the number of valid routes. However, with the increment of I_{max} , it takes more time for the network to converge, thus it occurs higher overhead.

Figure 5.12: Scenario I: Energy consumption per successfully received data. (a) grid with random deployment. (b) random planar deployment.

Fig. 5.12 depicts the energy consumption decreases with the decreasing frequency of control messages. ER-RPL achieves around 66% and 60% energy conservation compared to RPL on average, as shown in Fig. 5.12(a) and Fig. 5.12(b), respectively, because the longer routes in RPL consume more energy. The impact of the frequency of control message is more significant on P2P-RPL than ER-RPL. P2P-RPL disseminates the control messages throughout the network, and the large amount of control messages constitute a significant portion of the total energy cost. So the variation of the frequency of control messages leads to a obvious difference on the energy consumption performance in P2P-RPL. With a great reduction of the control overhead, ER-RPL achieves a significant energy conservation compared to P2P-RPL. The energy consumption of ER-RPL only changes slightly with a variation of the frequency of control messages. Apart from the frequency of the control messages, many other factors also affect the energy performance, for instance, the size of data packets and control messages.

Fig. 5.13 shows the PDR performance drops as the traffic increases. In RPL,

Figure 5.13: Scenario II: The PDR comparison. (a) grid with random deployment. (b) random planar deployment.

the root becomes the bottleneck of the network when the traffic is heavy, resulting in a serious performance drop. The dead zone problem may take place in the random planner deployment. But ER-RPL still achieves a nearly optimal value, as represented by the PDR performance of P2P-RPL_s. It shows that the proposed ARS is an efficient solution for the dead zone problem.

Figure 5.14: Scenario II: Hop count comparison.. (a) grid with random deployment. (b) random planar deployment.

Fig. 5.14 depicts that the average hop count of P2P routes selected by ER-RPL is very close to P2P-RPL, which is 40% less than that of RPL. In RPL, the data packets are delivered through the pre-established route via the root, thus it suffers from long route. The average hop count is slightly more in Fig. 5.14(a) compared to Fig. 5.14(b). Because the node density of the grid with randomness

deployment is lower than that of the random planar deployment in this study.

Figure 5.15: Scenario II: Control overhead of simulation and theoretical results. (a) grid with random deployment. (b) random planar deployment.

Fig. 5.15 depicts that the control overhead increases with the increasing of P2P traffic flows. In Fig. 5.15(a), ER-RPL achieves about 59%, 66% less control overhead than P2P-RPL with symmetric and asymmetric link, respectively. In Fig. 5.15(b), ER-RPL_s and ER-RPL_{as} achieve about average 55%, 58% less overhead compared with P2P-RPL_s and P2P-RPL_{as}, respectively. This is because more regions join the route discovery with the ARS to solve the dead zone problem of the random planar network. Results from the simulation study are close to that from the theoretical analysis. The subtle difference is due to the randomness of control frequency in the route discovery stage that are considered in the simulation model but are not considered in the analytical model with Eq. (5.19) and Eq. (5.22). Specifically, during the route discovery stage, the transmission frequency of route discovery messages broadcasting are higher. And the frequency of broadcast control message decreases as the nodes join the temporary DODAG. Finally, the transmission interval of control message broadcasting reach to the maximum interval and it is maintained in this maximum interval for the route maintenance. Since in the analytical model, the transmission interval is considered as the maximum interval. The frequency of broadcasting control messages is much higher during the route discovery stage, which is not considered in the analytical model. This is the reason that the normalized control messages of theoretical analysis is smaller than the simulation results. In addition, R2R routing further reduces the control overhead in ER-RPL, which is reflected by the smaller gap between the theoretical

and simulation results than P2P-RPL.

Figure 5.16: Scenario II: Energy consumption per successfully received data . (a) grid with random deployment. (b) random planar deployment.

Figure 5.17: Scenario II: End-to-End delay comparison. (a) grid with random deployment. (b) random planar deployment.

Fig. 5.16 shows that the energy consumption increases with the increase of P2P traffic flows. ER-RPL achieves a great energy conservation compared to RPL and P2P-RPL. In RPL, all traffic flows are delivered through the root. The area near the root will become congested with high traffic, and a large amount of packets get dropped in that case. So the increase of P2P traffic flows has a more significant impact on RPL than ER-RPL and P2P-RPL. ER-RPL and P2P-RPL shows a more robust energy performance with the increase of network traffic flows. In addition, ER-RPL has similar energy performances with symmetric and asymmetric links,

while P2P-RPL_{as} consumes more energy than P2P-RPL_s due to a less amount of successfully delivered data packets.

Fig. 5.17 depicts that the end-to-end delay increases as the traffic increases. ER-RPL achieves a similar performance as P2P-RPL. RPL suffers from long delay due to the longer route used for the data delivery. Additionally, the increasing traffic load has a significant impact on the delay performance of RPL, because of the bottleneck in the network.

5.6 Summary

This chapter has proposed ER-RPL, which is a distributed energy-efficient region-based routing protocol. ER-RPL is a hybrid of proactive and reactive routing protocol, and it exploits the region information of networks with stationary nodes. The proposed routing protocol provides a generic traffic pattern support, which inherits the DODAG network structure optimized for MP2P traffic pattern from RPL, and discovers the best quality routing path within the regions between the source and destination nodes for P2P communication. ER-RPL can achieve great energy conservation without compromising network reliability. In addition, theoretical analysis of control overhead reduction is also provided. Simulation results in terms of PDR and energy consumption show that, in comparison with RPL, ER-RPL can achieve significant improvement on network reliability and energy conservation. In addition, compared to P2P-RPL, ER-RPL can achieve significant reduction of routing control overhead and energy consumption. Furthermore, the performance in terms of PDR also shows that ER-RPL can improve network reliability compared to P2P-RPL when the wireless links are asymmetric. Therefore, the proposed ER-RPL is robust to different wireless channel conditions.

Chapter 6

An Energy-efficient and Cluster-parent-based RPL with Power-level Refinement for LLNs

6.1 Introduction

Chapter 4 proposed the cluster-parent-based RPL (CRPL), which attains significant improvement on reliability and reduction of the number of transmissions. In particular, in order to conserve energy, CRPL selects the optimal CPS in a way that the total number of transmissions are minimized. Although, the energy consumption during transmission is the major component of the total energy depletion of the network, some other factors including the lossy rate of wireless medium, the energy consumption in reception and idle-listening, also have significant impact on the total energy depletion of the network, especially for dense lossy networks. Therefore, the optimal forwarder list should be selected in the way that directly minimizes the energy consumption of the network. In order to address this issue, this chapter proposes a hybrid, energy-efficient, and cluster-parent-based RPL routing protocol, called HECRPL, with the aim of simultaneously achieving high reliability as well as great energy conservation.

Reliability is of paramount importance for a network. However, the unrelia-

bility of wireless medium easily results in excessive packet retransmissions, which leads to a great deal of energy consumption and long occupation time of the shared wireless medium. Particularly, there is a trade-off between power conservation with a reliable and efficient data delivery. In traditional protocols, unnecessarily high transmission power is used, and it shortens the lifetime of nodes and introduce excessive mutual interference, which decreases the possibility of spectrum reuse. A lower level of transmit power has advantage in spatial reuse and energy conservation, whereas a higher transmission power means that there will be more neighbours and that more progress will be made with every transmission, resulting in fewer lossy links that packets need to travel over to reach their destination. As addressed earlier, achieving both reliability and energy-efficiency at the same time is extremely challenging but critical to the design of a routing protocol, especially for LLNs. Recent developments have enabled the rapid advancement and pervasiveness of the multifunctional and energy-constraint nodes with continuous or discrete power levels. This chapter investigates the factors that have great impact on energy consumption and reliability of the network to optimally balance the trade-off. Moreover, the uneven deployment of nodes and heterogeneous traffic patterns result in an unbalanced distribution of workload. So the nodes in the hotspot area quickly exhaust their energy, which is known as the “*Energy Hole Problem*”. Therefore, workload sharing is inherently important, as it prolongs the lifetime of the network and solves the problem of energy holes, thus it is also considered in the protocol design of this chapter.

In order to address the aforementioned challenges, a proactive opportunistic adaptive routing protocol, namely HECRPL, is proposed to support energy-efficient, robust, reliable, and scalable routing with load balancing capability. Specifically, HECRPL uses a top-down approach to select the optimal forwarder set, so as to minimize the energy depletion of the network, leverage path diversity, and improve the robustness. In particular, a novel routing metric is proposed to precisely measure the end-to-end transmission cost in the networks. Importantly, a distributed algorithm is proposed to select the optimal forwarder list and the optimal transmission range. Furthermore, in a forwarder list, priority is based on a hybrid of residual energy and the lossy conditions of wireless channels to balance the workload of the network. In addition, HECRPL also incorporates a local loss recovery scheme to efficiently detect and retransmit the lost packets, and an

overhearing-based coordination scheme to avoid duplicate transmissions. Extensive simulation results demonstrate that our proposed approach effectively prolongs the lifetime of the network and achieve more reliable data delivery compared to the benchmark, RPL, across a wide range of scenarios. The proposed HECRPL is implemented in the NS-3 platform to study and demonstrate its feasibility and effectiveness under a wide range of scenarios.

Essentially, HECRPL incorporates the following five major features to effectively prolong the lifetime of the network, while achieving a high reliability and fairness for the network:

- First, a top-down approach for optimal cluster-parent set selection to minimize the energy depletion of the network and leverage path diversity, while achieving the global goal.
- Second, an overhearing-based coordination among nodes in a cluster-parent-set to avoid duplicate transmissions.
- Third, the priority in a cluster-parent-set is based on a hybrid of residual energy and the lossy conditions of wireless channels.
- Fourth, an efficient loss recovery scheme for the detection and retransmission of the lost data packets.
- Fifth, the optimal transmission power is selected to increase the network capacity (increase spatial reuse) and further reserve energy.

The rest of this chapter is organized as follows. Section 6.2 presents the system architecture. The design of our proposed HECRPL is described in Section 6.3. Section 6.4 demonstrates the performances evaluation of HECRPL with comparison to the benchmarks. Finally, Section 6.5 summarises this chapter.

6.2 System Architecture

This section presents the formalization of the network model, the energy model, and channel model. Specifically, Graph theory is employed to illustrate the network model of the proposed HECRPL. With the network model, the method for

the selection of the optimal forwarder list is formalized as a recursive procedure. The energy model takes the energy consumption in receiving and overhearing into consideration. The channel model is defined in accordance with the impact of transceivers' power and sensitivity on the packet reception rate.

6.2.1 Network Model

Consider that n number of stationary nodes with limited energy are dispersed in an area, the network can be described as a non-cyclic graph, denoted by $G = (V, E, W)$, where V represents the set of nodes in the network, E represents the set of edges in the network, and W is the upper bound of a outdegree of a node. In HECRPL, a node chooses an optimal forwarder list as its optimal cluster-parent-set (CPS), which helps to forward packets to the destination node. In order to define the most cost efficient CPS, we conduct a mapping for every node to a positive, real-valued energy cost for routing as $c_e : V \rightarrow \mathbb{R}^+$. Furthermore, a mapping for each link to a link cost, which is $c_l : E \rightarrow \mathbb{R}^+$ is performed. which makes the network becomes a weighted graph. In this way, the cost of the edge $(u, v) \in E$ is denoted as $c(u, v)$. The CPS of node u is denoted by $CPS(u)$, and $c(u, CPS(u))$ is used to denote the cost for node u to successfully deliver one packet to the destination node d . Node v is an arbitrary node in the CPS of node u , which is denoted by $v \in CPS(u)$. $c(u, CPS(u))$ is composed of two parts: the cost that node u expends to deliver the packet to at least one node in its $CPS(u)$, and the cost that the packet reception in $CPS(u)$ spends to deliver the packet to the destination through its own CPS, so $c(u, CPS(u)) = c(u, v) + c(v, CPS(v))$, where $v \in CPS(u)$. The way to select the optimal CPS for an arbitrary node u in the network is to select the $CPS(u)$ that gives the minimum cost to node u , which is $c(u, d) = \min \{c(u, CPS_o(u)) | CPS_o(u) \text{ is optimal CPS for node } u \text{ to } d\}$. In this work, we consider that nodes are capable of adjusting their transmitting power to a certain number of discrete values. The transmission range of a node is determined by the level of its transmit power, the relation of which will be introduced later. The number of adjustable levels of transmission power/ranges is considered to be M , where $M \geq 1$, and the available transmission ranges is denoted by $(R_1, R_2, \dots, R_M)$, where $(R_1 < R_2 < \dots < R_M)$. The ultimate goal of HECRPL is to extend the lifetime of the network and improve its reliability. In

HECRPL, a node chooses an optimal transmit power level and an optimal set of forwarders as its CPS, denoted as R_{o_i} and $CPS_o(i)$ for node i , respectively.

6.2.2 Energy Model

A node has four basic operating states (transmitting, receiving, idling, and sleeping) and one transition state among different states [154]. The energy consumption in the idling and sleeping states is the minor component, and is therefore not discussed in this study. The First Order Radio Model [155] has been widely used to measure the dissipation of energy for wireless sensor networks (WSNs) [156, 24]. In this model, the radio takes E_{elec} to run its circuitry of the transmitter or the receiver. The transmitting amplifier is used to achieve an acceptable signal-to-noise (SNR) ratio, which is denoted by ϵ_{amp} . In this study, the propagation loss exponent is considered to be 2. In this way, the energy consumed to transmit a k -bit message with a transmission range r at the transmitter side is modelled as

$$E_{tx}(k, r) = E_{elec}k + \epsilon_{amp}r^2. \quad (6.1)$$

The energy consumption of the receiver is modelled as

$$E_{rx}(k) = E_{elec}k. \quad (6.2)$$

The energy depletion for overhearing k bits of packets is denoted by $E_{oh}(k)$, which can be obtained through

$$E_{oh}(k) = \alpha E_{elec}k = \alpha E_{rx}(k), \quad (6.3)$$

where $0 < \alpha \leq 1$.

It is observed that overhearing contributes a significant portion of the total energy cost in dense networks, due to a large number of overhearing recipients. We denote node i 's one-hop neighbor number as N_i , and the size of its forwarder set (its CPS) as n_i . The energy consumption for node i 's one transmission of a k -bit message with a transmission range r is denoted by $E_i(k, r)$, which includes three components, namely the energy consumed in transmitting, receiving, and overhearing, and it is defined as

$$\begin{aligned}
E_i(k, r) &= E_{tx}(k, r) + n_i E_{rx}(k) + (N_i - n_i) E_{oh}(k) \\
&= E_{tx}(k, r) + n_i E_{rx}(k) + (N_i - n_i) \alpha E_{rx}(k),
\end{aligned} \tag{6.4}$$

Although additional energy savings can be achieved by introducing a mechanism for encoding and decoding in the header of the packet, the price in terms of hardware complexity is high, and it leads to delay for the protocol. For simplicity, this research work considers that the overhearing receives the full length of the packets, where α becomes 1. Therefore, Eq. (6.4) can be simplified to

$$E_i(k, r) = E_{tx}(k, r) + n_i E_{rx}(k) + (N_i - n_i) \alpha E_{rx}(k) = E_{tx}(k, r) + N_i E_{rx}(k). \tag{6.5}$$

6.2.3 Channel Model

It is observed that the transmitter power and receiver sensitivity of different nodes usually differ, and that this affects the channel condition and packet reception rate [145]. In this work, in order to model the lossy channel of the network, each node is assigned with a random number, called *weight*, which is denoted as wg_i for node i and $0 \leq wg_i \leq 1$. The *weight* represents the power and sensitivity of the node for transmitting and receiving, and it is used to calculate the probability of a successful delivery of data between nodes. The probability of a successful delivery from node i to j is denoted by p_{ij} , which is the average of the weights of nodes i and j in this study.

$$p_{ij} = \frac{wg_i + wg_j}{2}. \tag{6.6}$$

We define an average degree of link loss for each node, which is denoted by β_i for node i . It is the probability that node i 's transmission can, on average, be successfully received by the neighbor node, and is defined as

$$\beta_i = \frac{\sum_{j \in N_i} p_{ij}}{N_i}. \tag{6.7}$$

In a lossy network, the communication cost of the network for node i 's one transmission is composed of the energy cost in transmitting, receiving and overhearing. When the average degree of link loss for node i is β_i , an average of $\beta_i N_i$ neighbor nodes will receive the transmission of node i . In this way, the average

communication cost of the network for node i 's one transmission can be calculated as

$$E_i(k, r) = E_{tx}(k, r) + \beta_i N_i E_{rx}(k). \quad (6.8)$$

6.3 An Energy-efficient and Cluster-parent-based RPL with Power-level Refinement for LLNs

The ultimate goal of HECRPL is to extend the lifetime of the network and improve its reliability. The design of HECRPL mainly consists of two stages: the selection of the optimal cluster-parent-set (CPS) during the network initialization stage and the coordination of CPS for packet forwarding during the packet delivery stage.

During the network initialization stage, each node selects an optimal CPS in conjunction with power-level refinement in HECRPL, aiming at minimizing the end-to-end energy cost. HECRPL uses a novel routing metric, which considers a combination of energy cost and the lossy nature of wireless channels. In HECRPL, the network topology is formed in the way that each node selects its optimal CPS and transmits at the possible lowest power level in a distributed manner.

During the data delivery stage, the actual forwarder in one CPS is selected opportunistically based on the actual reception result. The priority of nodes in a CPS for forwarding data is based on a combination of residual energy and the lossy condition of wireless channels to efficiently balance the energy depletion of the network graph. HECRPL adopts the coordination based on overhearing and loss recovery for data retransmissions to avoid duplicate transmissions and provide reliable support for the delivery of data. In addition, a node's priority in a CPS is dynamically updated for balancing the workloads of the network.

HECRPL dynamically updates the topology of the network and maintains the connectivity with the local and global recovery mechanisms. The design of HECRPL for supporting generic traffic patterns is addressed in detail.

6.3.1 Cluster-parent Set Selection

In order to optimize the lifetime of a network, HECRPL selects the optimal CPS and the right transmission range for each node in a distributed manner. A node chooses an optimal set of forwarders as its CPS, which can efficiently improve the reliability of the network by leveraging path diversity. The trade-off of the protocol design is that, with a higher transmit power, a node has more neighbors to choose from for its CPS selection and more progress can be made in one transmission, whereas with a lower transmit power, the energy expended in transmitting can be conserved, and the energy depleted in overhearing can be reduced. Two important issues are addressed in our design: 1) how to select the optimal CPS to minimize the energy consumption for delivering data packets; and 2) how to select the optimal transmission range to balance the aforementioned tradeoff. We will describe our approaches for each of the issues.

A top-down approach is adopted by HECRPL to balance the aforementioned tradeoff and select the optimal CPS to support reliable data delivery and minimize the energy cost of the network. In HECRPL, all of the transmission ranges are considered while selecting the optimal CPS. A node selects the optimal CPS, with the help of which the energy cost for delivering the data packets to the destination is minimized. With different node distributions, a node chooses an optimal transmission range, denoted by R_{o_i} for node i , after it is deployed randomly. The optimal CPS and power level are selected by a node so that the total energy that the network takes to successfully deliver a data packet through the CPS to the destination is minimized.

The End-to-end Transmission Cost Calculation

The actual cost in energy of delivering data packets is the end-to-end transmission cost of the network instead of the one-hop transmission cost. Without considering the end-to-end transmission cost, it is possible to forward packets to nodes with a “poor” connection to the destination. Therefore, our criterion for selecting the optimal cluster-parent-set (CPS) is to minimizing the end-to-end energy cost, which is calculated based on the total number of transmissions and on the energy consumption associated with each transmission. In this work, we propose a novel

routing metric to optimally select the best CPS.

One important issue relating to the selection of the optimal CPS is the inter-link quality of the members of one CPS. The “poor” link qualities between members can lead to errors of coordination, resulting in duplicated packets. Without taking the inter-link quality into consideration, duplicated transmissions may sometimes overweight the gain in path diversity. Hence, only nodes with good inner-link qualities can be selected and placed into one CPS. In our design, the inner-link quality of the members of one CPS has to be above 0.5. As shown in Fig. 6.1, the wireless link success rate among r_1 , r_2 , and r_3 should be more than 0.5 so that they can fulfil the requirement of being selected as one CPS.

Figure 6.1: Network formation with an optimal cluster-parent set formation.

Researchers has proposed many solutions to measuring link quality [159], a topic that is beyond the scope of this work. In this work, it is assumed that this issue is handled by the link layer. The probability of successful transmission between node i with node j is denoted by p_{ij} . In this case, the one hop ETX can be calculated as $E_{ij} = 1/p_{ij}p_{ji} \approx 1/p_{ij}$, where p_{ji} in this equation refers to the probability of successful delivery of an ACK from node j to node i , which is approximate to one. Because the ACK usually has very small size, which can be recovered with the strong coding techniques, we assume ACK does not suffer from link loss in this study. The joint ETX [148] is the average number of transmissions that a node takes so that at least one of its forwarders can successfully receive the data packet, which is defined in Eq. (6.9), and node j is an arbitrary node in the CPS of node i . The joint ETX of node i is defined as

$$ETX_{CPS(i)} = \frac{1}{p_{i,CPS(i)}} = \frac{1}{1 - \prod_{j \in CPS(i)} (1 - p_{ij})} \quad (6.9)$$

We define the Anycast Energy Cost (AEC) as representing the transmission cost to the network for one packet to be successfully received by at least one node among its CPS. Thus, the AEC of node i for transmitting a k -bit packet with the transmission range r is defined as

$$AEC_{i,CPS(i)} = \frac{E_i(k, r)}{1 - \prod_{j \in CPS(i)} (1 - p_{ij})} = \frac{E_{tx}(k, r) + \beta_i N_i E_{rx}(k, r)}{1 - \prod_{j \in CPS(i)} (1 - p_{ij})} \quad (6.10)$$

The actual cost of packet transmission is the end-to-end transmission cost instead of the one-hop transmission cost. The energy cost to the network for a CPS to deliver a packet to its destination is defined as the Remaining Energy Cost (REC). Meanwhile, we denote the end-to-end transmission cost for a node i through its optimal CPS as $ECCPS_{i,CPS_o(i)}$, the simplified version of which is $ECCPS_i$. It is assumed that the nodes in a CPS are sorted according to their end-to-end energy cost, in which ($ECCPS_1 \leq ECCPS_2 \leq \dots \leq ECCPS_{|CPS(i)|}$). The Remaining Energy Cost (REC) of node i 's CPS is defined as

$$REC_{i,CPS(i)} = \frac{ECCPS_1 p_{i1} + \sum_{i=2}^{|CPS(i)|} ECCPS_j p_{ij} \prod_{k=1}^{j-1} (1 - p_{i,k})}{1 - \prod_{j \in CPS(i)} (1 - p_{i,j})} \quad (6.11)$$

The accumulated end-to-end transmission cost of a node is composed of its AEC and the REC associated with its optimal CPS. Hence, the end-to-end transmission cost for an arbitrary node i through its CPS can be computed through

$$ECCPS_{i,CPS(i)} = \min(AEC_{CPS(i)} + REC_{i,CPS(i)}) \quad (6.12)$$

With information on the ECCPS values and inter-link quality of its neighbour nodes, a node computes all possible combinations of the CPS and obtains the corresponding ECCPS values. After a while, the network topology will converge into a stable stage. An example structure of the network is shown in Fig. 6.1. In that example, the optimal CPS of node α consists of r_1 , r_2 , and r_3 .

Routing Design for HECRPL

HECRPL builds the network topology using three steps: neighbor discovery, optimal CPS selection, and topology optimization. The step of neighbor discovery in HECRPL works in a similar way to that of RPL. In addition, nodes update their one-hop neighbor list during the network initialization stage. The root node starts to form the topology. We design the cluster-formation message (CFM) as the control message for the propagation of nodes' cost and the maintenance of the network topology in HECRPL. The packet structure of the CFM is shown in Fig. 6.2. We use two bytes for each field in the CFM message, and each frame is composed of three fields. The CFM message carries a node's ECCPS cost and its optimal CPS list, which includes the ID of nodes in the CPS and their *Priority* and *Cost* associated with that *NodeID*. The size of a CFM message is determined by a node's optimal CPS size. For an arbitrary node i , w_i , which denotes the size of $CPS_o(i)$, has to be at least equal to or greater than one to maintain the node's connectivity, and $w_i \leq N_i$. The CFM message broadcast by node i is with the size of $w_i + 1$. In the last frame of the CFM message, the node encloses the destination address, its residual energy, and its end-to-end transmission cost in the fields of *DstID*, *ReEnergy*, and *Cost*, respectively. The broadcasting of the CFM object is also based on the trickle algorithm, so that HECRPL can quickly set up the topology of the network with a low overhead.

Figure 6.2: The CPS formation control message format.

A node chooses its CPS and computes its transmission cost upon receiving the CFM. Fig. 6.3 illustrates the process of optimal CPS selection at node j . Once a node receives the CFM from its neighbors, it first checks whether or not the sender's CPS list contains the ID of itself. If it is, it conducts the following actions: 1) record the node list and the corresponding priority list, 2) record the sender ID as its child node with this set, 3) include the sender in its list of helpers. The list of helpers of node i is denoted as $\{nb_{i_h}\}$. While computing the CPS set, a node will not choose nodes from its list of helpers into its CPS set. In another case, if the recipient's ID does not exist in the CFM, the recipient updates this

neighbour's cost. If it is the first time that a node obtains the EECPS value of this neighbor or the value is different from the historical value, the node will select its optimal CPS and choose the level of transmit power. A nodes check all possibilities of the subsets of neighbor nodes, and calculating the cost with different possible CPSs regarding to different transmission ranges. The node then selects the CPS that gives the lowest ECCPS value, which represents the end-to-end transmission cost, as its optimal CPS. The detail of this computing process is introduced in Section 6.3.1. If the latest cost value is different from the previous value, the node broadcasts its ECCPS value and the information about its latest optimal CPS to its neighbor.

Figure 6.3: The optimal CPS selection at node j .

Refining the Level of the Transmission Power

The relative distances between a node and its neighbors is an important factor in our design. There have been many studies on how to estimate this distance using the Received Signal Strength Indicator (RSSI) [163, 164]. In this work, we assume that the relative distance between a node i to its neighbor node j , which is denoted by d_{ij} , can be estimated by the RSSI values. In this way, a node can obtain different neighbor lists corresponding to different transmission ranges. The

set $\{nb_{i_r}\}$ denotes the neighbor list for node i with transmission range r , where an arbitrary node $j \in \{nb_{i_r}\}$ satisfies $d_{ij} \leq r$. Therefore, the end-to-end transmission cost with different CPSs given the particular neighbor list (subject to different power levels) is calculated. An example structure of the network is shown in Fig. 6.1. In that example, the optimal CPS of node α consists of r_1 , r_2 , and r_3 .

Figure 6.4: Illustration of the adjustable transmission range.

The number of one-hop neighbors usually increases as the transmission range increases, which is a result of increased power levels. With the adoption of a smaller transmission range, the number of overhearing recipients is also reduced. With each adjustable transmission range, there is a corresponding neighbor list for an arbitrary node i . In order to avoid the routing loop, the candidate nodes for the selection of the CPS should exclude the list of helpers from its neighbor list. We use Ω to denote the set of nodes that can be used for the selection of the optimal CPS. Then, the different possible combinations of nodes, the size of which is varied from 1 to W , are chosen from Ω to examine the energy cost for the node i through that set. It is worth to highlight that the inter-link qualities between the nodes in one possible forwarder set needs to be not less than 0.5. Each power level is substituted into Eq. (6.10) and (6.11). Then, the CPS with lowest ECCPS value at the current transmission range can be obtained. Among those CPSs with a different transmission range, the one giving the lowest ECCPS value is considered as the optimal CPS for the node. Meanwhile, the transmit power corresponding to that transmission range is adopted for the packet transmission. The pseudo-code of the distributed optimal CPS selection with the transmit power

refinement algorithm is shown in Algorithm 1.

The transmission range of a node is determined by the level of its transmit power, the relation of which can be obtained from Eq. (6.1). The adjustable transmission range is mainly meant for data packets. Due to the small size of the control message, the transmit power required for broadcasting control messages is much less than that required for delivering data. Therefore, we assume that the control message can be delivered within R_M with any level of transmit power.

Except for nodes in $\{nb_{i_h}\}$, node i consider all nodes within its transmission range as its candidate neighbor node set. The combination of nodes that can give the node a minimum cost towards the gateway will be selected as its optimal CPS. For simplicity, here, we do not exclude $\{nb_{i_h}\}$ from $\{nb_{i_r}\}$ when computing complexity. Let k_r denote the size of $\{nb_{i_r}\}$ with the transmission range r , and $1 \leq W \leq k_{R_M}$. The number of candidates to examine is

$$\sum_{r=R_1}^{R_M} \sum_{l=1}^W \binom{k_r}{l} \quad (6.13)$$

Eq. (6.13) will lead to a complexity of $O(\sum_{r=R_1}^{R_M} k_r^W)$. Although the computational processing capability of today's nodes keeps increasing with the rapid development of technologies, the complexity of the computations should be reduced for widely deployment purpose. In [148], a greedy heuristic was put forward to solve the issue of reducing the complexity. We have enhanced that approach in our design to further reduce the complexity of HECRPL. The calculation of the complexity considers that nodes are uniformly distributed in the network with the density, ρ . In this way, k_r is a function of the transmission range the node, which is $k_r = \rho\pi r^2$. With the design of reduced complexity, the detailed procedure of optimal CPS selection of node i is described in Fig. 6.5. Instead of randomly combining the nodes, node i sorts the list $\{nb_{i_{R_M}}\}$ according to their ECCPS in an increasing order. Then node i first selects the node with the least cost and tries to combine it with the next best node. If this process decreases the cost of node i , node i will take these two nodes as its CPS and continue to combine them with the third best node (the node with the least cost in the remaining node list). A node keeps repeating this process until it reaches the following conditions: one is that the combination does not decrease the cost, and another case is that its CPS size

has reached the maximum value. Then, the node checks the minimum available transmission range that can cover these nodes in the optimal CPS set as its R_{o_i} . In this way, the heuristic makes the complexity to $(k_{R_M} + 2W - 1)$, resulting in a complexity of $O(k_{R_M})$.

Algorithm 2: The Distributed Optimal CPS Selection with the Transmit Power Refinement Algorithm.

Data: The selection of an optimal CPS and transmit power level for node i . ;

Assume Ω and Φ denotes two sets of nodes and are initialized to empty.;

Initialize C_{min} to 999.

```

1 forall  $r \rightarrow R_1$  to  $R_M$  do
2   for Any node  $j \in \{nb_{i_r}\}$  and  $j \notin \{nb_{i_h}\}$  do
3      $\Omega \leftarrow \Omega \cup j$ ;
4   for  $w \rightarrow 1$  to  $W$  do
5     if  $w = 1$  then
6       for each  $j \in \Omega$  do
7          $ECCPS_{i,j} \leftarrow$  call Eq. (6.10) and (6.11);
8         if  $C_{min} > ECCPS_{i,j}$  then
9            $C_{min} \leftarrow ECCPS_{i,j}$ ;
10           $R_{o_i} \leftarrow r$ ;
11           $CPS_o(i) \leftarrow \Phi$ ;
12     else
13       for  $\Phi \leftarrow$  any  $w$  number of nodes  $\in \Omega$  do
14         if  $p_{u,v} \geq 0.5$ , where  $u, v \in \Phi$  then
15            $ECCPS_{i,\Phi} \leftarrow$  call Eq. (6.10) and (6.11);
16           if  $C_{min} > ECCPS_{i,\Phi}$  then
17              $C_{min} \leftarrow ECCPS_{i,\Phi}$ ;
18              $R_{o_i} \leftarrow r$ ;
19              $CPS_o(i) \leftarrow \Phi$ ;
20  $ECCPS_{i,CPS_o(i)} \leftarrow C_{min}$  ;

```

6.3.2 CPS Coordination for Packet Forwarding

HECRPL uses an adaptive cooperation scheme for forwarding data after selecting the optimal CPS. A distributed dynamic priority scheduling algorithm is implemented in HECRPL to balance the workload of the network, which significantly prolongs the lifetime of the network. Cooperation based on overhearing takes place

 Figure 6.5: Reduced-complexity design of the optimal CPS selection at node i .

according to the priority list of the nodes in a CPS, which is calculated during the selection of the CPS. In addition, HECRPL uses the per-hop network layer retransmission with the acknowledgement (ACK) in order to provide best-effort reliability.

Dynamic Priority Updating for Workload Balancing

HECRPL optimizes the route selection process and sorts the priority list for nodes in a CPS for forwarding data by combining the residual energy and the end-to-end transmission cost. As mentioned before, each node has a ECCPS value, which is the end-to-end transmission cost in terms of energy consumption. The value of ECCPS is calculated by jointly considering the distribution of both nodes and the communication qualities of the link layers. Less energy in the network is consumed if the packet is forwarded by a node with a lesser ECCPS value. In traditional routing protocols, nodes with a low cost will be chosen for forwarding data. Due to a heavy traffic load, the energy of nodes with a low cost usually drains out quickly, which is the common phenomenon in traditional routing protocols. In

order to share the workload and prevent nodes with a low energy cost value from draining out their energy, nodes in one CPS will cooperate with each other to opportunistically relay data packets in HECRPL. That is achieved by considering the residual energy of the nodes when assigning the priority to nodes in a CPS. In this way, a node with a smaller ECCPS value in one CPS will not bear much of a traffic load than the other nodes. We use Re_j to denote the residual energy of node j . In HECRPL, the priority of a node in one CPS is assigned according to the value of $Re_j/ECCPS_j$, where $j \in CPS(i)$. We denote node j 's priority in node i 's CPS using $P_{CPS(i),j}$, where $j \in CPS(i)$ and $P_{CPS(i),j} \in \mathbb{R}$, $P_{CPS(i),j} = 1, 2, \dots, w_i$. Members of one CPS schedule their transmission following a priority list. The node with top priority in carrying out the packet forwarding is the one with the greatest value in terms of the quotient of its residual energy by its ECCPS value. The node with the top priority schedules the transmission first. The priority list is updated dynamically when the CFM is received from its CPS, which from time to time carries a different ECCPS and residual energy.

Overhearing-based Scheduling and Loss Recovery

In order to avoid duplicate transmissions, the members of a CPS follow a cooperative overhearing scheme to participate in forwarding data. A timer-based ACK is used for reliable data delivery and coordination among one CPS. Instead of immediately replying the ACK, the members in a CPS schedule their ACK transmission following the priority list after receiving a data packet. The ACK waiting time for members of a CPS is defined as T_{Aw} . For an arbitrary node j , its schedule for forwarding data is $(P_{CPS(i),j} - 1)T_{Aw}$ after receiving the data packet from node i . As a member of the CPS, every node maintains an overhearing list consisting of the ID of other nodes that have higher priorities in the same CPS. During its ACK waiting time, a node keeps listening to the channel to check if there is any ACK transmission from a node in its overhearing list. If there is, this means that the data packet has already been successfully delivered by another member with a higher priority in the CPS, and the node will cancel the transmission for this packet. We define the time it takes for a packet to traverse from a node to another node as T_{tr} . Without receiving the ACK from any member of its CPS within the duration of $2T_{tr}$, a node will retransmit the data packet.

After nodes receive the data packet, they need to know whether or not it should participate in the data forwarding process. In other words, the node needs to check whether or not it itself belongs to the sender's CPS. If it does, it needs to have the information on the overhearing list. Here, we introduce two possible approaches. One approach is to let a node enclose the information about its CPS in the header of data packet, following the order in the priority list. Once receiving data packets, a node needs to examine the packet header, so that the node can decide whether or not it should be involved in forwarding the packet, determine its priority in the CPS, and obtain the overhearing list. Another approach relies on a local forwarding table. A node's local forwarding table stores the information of the CPS that it belongs to. The information includes the child node corresponding to each CPS and the priority associated with each node in the CPS. In this way, an overhearing list can be obtained based on the priority of the nodes' IDs in the CPS.

6.3.3 Updating and Maintaining a Dynamic Topology

With the help of the CPS, HECRPL forms the topology much more robustly compared to the conventional protocols. Considering that nodes in LLNs usually rely on a limited energy source, nodes will become "dead" once they are drained out of their energy. Then, some of the nodes will become disconnected and the network connectivity decreases. Therefore, the network topology needs to be reconstructed from time to time. Although an event-driven (e.g., a node's death) topology update can maintain the topology in real time, it introduces too much overhead. The advantage of a timer-based topology update is the limited overhead, but the route will become inefficient or be disconnected if the periods of time are too long. Transmitting the control message is very expensive in terms of energy consumption in LLNs, and the system reliability and routing performance will drop if the network connectivity cannot be efficiently maintained.

Our proposed HECRPL is designed with *Local Recovery* and *Global Recovery* to support the updating of the topology with quick route repair and low routing overhead. *Local Recovery* is used to recover the local connectivity with a quick response time once some nodes die. *Global Recovery* is used to periodically op-

timize the selection of the CPS towards the root. With a design that combines event-driven and timer-based topology updates, the route can be recovered quickly, and global optimization can be achieved with low amount of routing overhead in HECRPL.

Local Recovery: We consider a node is to be dead when it consumes 99% of its total energy. A node will then broadcast a short message (with a one bit flag) to inform its neighbor node with a *Death Notification* message. The recipients of the *Death Notification* will remove the sender from their one-hop neighbor lists. In RPL, upon receiving the *Death Notification* from its preferred parent, a node will immediately choose another lower-ranking node from its set of candidate parents set to be its preferred parent. In HECRPL, upon receiving the *Death Notification* from nodes in its CPS, a node will calculate the ECCPS value with its living neighbors, and broadcast its latest ECCPS value and CPS information with the CFM message. In this way, HECRPL can quickly recover and optimize the connectivity locally if some of their forwarders die.

Global Recovery: Although *Local Recovery* is applied, it is necessary to select the optimal forwarders with a globe scope, so that the end-to-end transmission cost can be minimized. In our implementation, RPL increments the DODAG Version Number periodically for the *Global Recovery*. In RPL, a node selects its preferred parent and computes its rank according to the latest DODAG Version Number enclosed in the DIO message. A similar approach is implemented in HECRPL. HECRPL selects the optimal CPS based on the living nodes and their cost through a top-down approach during the *Global Recovery*. The root node is the only node that can increase the DODAG Version Number. Any node that receives the control message with a bigger DODAG Version Number will start to reconstruct the topology. In this way, the optimal network topology can be constructed with the *Global Recovery*.

6.3.4 Supporting Generic Traffic Patterns

Considering that the major traffic pattern of an LLNs application is multipoint-to-point (MP2P) traffic, this work focuses on designing the kind of proactive routing that supports traffic whose destination is the root. But the design of HECRPL can

also be extended to support other traffic patterns, e.g., point-to-point (P2P) traffic. We first briefly review the existing schemes for P2P routing support in LLNs. Then, we introduce how HECRPL can be used to support an energy-efficient and reliable P2P routing in a reactive manner.

RPL has to pre-establish the DODAG and the data packets can only be forwarded along the existing route. For P2P communication, the source node needs to send packets upwards until the packets reach the destination node's ancestor, and this ancestor node will forward the packets downwards to the destination node. There is a storing mode and a non-storing mode in RPL [7]. In RPL's non-storing mode, the root node is the only one that can serve as the common ancestor of the source and destination. The packet delivery for P2P communication suffers from long end-to-end latency and more unreliable links in RPL.

An enhanced protocol, P2P-RPL [122, 165], is proposed to discover the shortest P2P route for LLNs in a reactive manner. P2P-RPL builds the temporary DAG rooted at the source node. The source node initiates the route discovery process. Upon receiving the route request message, the node selects the preferred parent given the particular objective function, which works in the same way with the DODAG formation by RPL. Once the destination node receives the route discovery information, it replies with a message to the source node through the route available in the temporary DAG. The data will then be delivered through the reverse route. Sometimes the P2P route discovered by P2P-RPL is not much shorter than the available P2P route of RPL. The superiority of P2P-RPL over RPL depends on the network topology, e.g., the distance between the source node and the destination node .

HECRPL can be applied to discover the best quality routing paths in terms of reliability and energy-efficiency for P2P communication. HECRPL first measures the cost of the P2P route along the DODAG before discovering the optimal route. The source node sends a control message via the DODAG's root to the destination node, which records the path cost of this route. Upon receiving this control message, the destination node needs to decide whether to explore another route or to use this available route. It makes this determination by checking whether or not the cost of the existing route is within the route constraint¹. If the cost of the

¹The work in [151] introduces a mechanism about how to measure the cost of the P2P route

existing route is not within the route constraint, the destination node will initiate the route discovery process by broadcasting the cluster-formation message (CFM), the format of which has been described earlier. Upon receiving the CFM, the node records its neighbors' ECCPS cost. With the neighbors' cost, the node computes its optimal CPS towards the destination node by applying the Algorithm 1, and chooses the optimal transmission range. Then, the node broadcasts this information to its neighbors. In this way, each node finds the optimal route through a CPS towards the destination with the optimal transmission range. Not until the source node receives this information, it will send the data packets through its optimal CPS to the destination node. For the P2P routing support, the optimal CPS temporarily exists, the lifetime of which is strictly restricted by the information in the route request, unless the source node receives the lifetime extension information from upper layer. In HECRPL, the routing paths for a source-destination pair are optimized to minimize the end-to-end energy cost, and the CPSs are selected to leverage path diversity to improve the reliability of the network. In this way, HECRPL can provide efficient routing support for P2P communication.

6.4 Simulation Study

6.4.1 Simulation Setup

As we know, the simulation of routing protocols is of great importance for a better understanding and evaluation for the routing algorithms. Although there exist a similar RPL implementation on Omnet++ with Castalia [12], but its RPL implementation is limited currently. Contiki [13] and TinyOS [14] are used to test RPL with the real embedded node. The built-in simulators of Contiki and TinyOS are the Cooja and TOSSIM, respectively, but it is difficult to modify their frameworks to add new features. To the best of our knowledge, although a model of RPL on NS-3 [17] has been proposed in [16], it is still not available to access at current moment. Owing to its flexibility and wide usage, we choose NS-3 as our simulation platform, and we have implemented RPL and HECRPL from scratch.

along the DODAG in detail.

The lossy wireless medium is modelled using the error erasure channel. With the channel model proposed in this work, the lossy rate of each link can be obtained with the *weight* assigned to each node. Considering that MP2P traffic is the major traffic pattern in LLNs applications, this simulation study focused on the MP2P traffic pattern. The position of the root node has a great impact on the network topology, which is part of our study. In our study, the maximum number of CPSs is set to be 3 ($W = 3$), due to the consideration of the computational complexity. The simulation parameters of our experiments, which are commonly used in other literatures, are displayed in Table 6.1. It is assumed that ACK does not suffer from the link loss, because current coding techniques can easily recover the ACK due to its small size. Our simulation studies are conducted with the IEEE 802.11. As we know, IEEE 802.15.4 [161] is considered as the best candidate for LLNs. Although IEEE 802.11 is usually not regarded as the optimal choice for LLNs, the general behaviour of routing protocols, e.g., packet delivery, path-length, control overhead, etc., can be concluded from the simulation study with IEEE 802.11, especially for networks with a low data rate. So IEEE 802.11 is also commonly used to evaluate the performance of protocols for LLNs currently [140, 115]. The work in presents a detail performance comparison of IEEE 802.11 and IEEE 802.15.4 [162].

Table 6.1: Simulation parameters to evaluate the performance of HECRPL.

Parameter	Value
Retransmission Limit	3
Traffic Duration	200s
Packet size	512 bytes
Traffic rate	1 pkt/s CBR flow
Initial Energy	5000000000 nJ
Transmitter Electronics (E_{elec})	50nJ/bit
Transmit Amplifier (ϵ_{amp})	100pJ/bit/ m^2
The Minimum Time Interval Size (I_{min})	50ms

For HECRPL, the adjustable transmission range is set to be [30, 35, 40, 45]. Because RPL adopts a fixed transmission range, we first explore the optimal transmission range for RPL in terms of the reliability and energy efficiency in Scenario I. Then, nodes in RPL adopt the optimal transmission range, and the performance

of RPL is used as the benchmark to be compared with that of HECRPL in the following experiment, i.e., both in Scenario II and Scenario III. Since the node distribution strategies significantly affect the network performance in static networks, the node distribution strategies are used in Scenario II and Scenario III to examine the performance of HECRPL and compare that with RPL. In Scenario II, a non-uniform distribution in a circular area is studied, in which the density of nodes increases as their distance to the root decreases. In Scenario III, nodes are uniformly distributed in a square network area. In this scenario, two cases are studied. The first case investigates the impact of the link lossy level on the network performance of HECRPL and RPL, and the second case examines the performance of HECRPL and RPL as the network sizes change with the same density of nodes. In this study, a different random topology is generated in each run and every result is the averaged value of ten runs.

6.4.2 Performance Metrics

The study of HECRPL is compared to the benchmark protocol, RPL, in terms of the performance metrics:

- *Number of nodes alive*, N_{alive} , records the number of nodes whose residual energy is more than 1% of its initial energy.
- *Packet delivery ratio (PDR)*, P_{pdr} , refers to the ratio of the number of successfully received packets by the destination to the number of packets sent by the source nodes, which illustrates the reliability of the network.
- *Network lifetime*, T_{life} , is captured as the time that the first node, 10%, 20%, 30%, 40%, and 50% of nodes become dead in the network.
- *Energy consumption of the network*, e_n , is the total energy consumption of the network, which includes the energy cost in transmitting, receiving, and overhearing.
- *Energy consumption per data received*, e_d , is the average amount of energy consumed for one successful piece of data delivered to the destination.

- *End-to-end delay*, T_{delay} , consists of all of the delays that occurred during the transmission of the packet.
- *Routing control overhead*, O_r , refers to the size of the control overhead in terms of bytes in the routing layer.

6.4.3 Performance Evaluation

Different Transmission Ranges of RPL

In Scenario I, 180 nodes are uniformly distributed in a square area ($240\text{m} \times 240\text{m}$). The root is placed in the center of the network. The simulation time is 1000 seconds. The node's *weight* is randomly assigned as being between 0.1 to 0.7, and 0.3 to 0.9, resulting in a mean link success rate of 0.4 and 0.6, respectively. The transmission ranges of RPL of 30m, 35m, 40m, 45m are investigated, so as to select the best one for use below.

(a)

(b)

Figure 6.6: Scenario I: Different transmission ranges of RPL: (a) P_{pdr} . (b) e_n .

Fig. 6.6(a) and 6.6(b) show that the network has better performance in terms of reliability and energy efficiency with better link quality. With an increased transmission range, a higher PDR and a reduction in energy consumption can be achieved with RPL. This is because each transmission can make more progress with a higher transmission range, so that the number of lossy links that a node needs to travel through in order to reach the destination is reduced. In this way, the total number of transmissions is reduced, resulting in a reduction in energy cost with improved reliability. Therefore, we adopt 45m as the transmission range for RPL in the discussion below.

Non-uniform Distribution in a Circular Area

In this scenario, 180 nodes follow an exponential distribution in the circular area, where the mean and bound is set to be 85 and 120, respectively. A node's *weight* is randomly assigned between 0.2 to 0.8. The root is placed in the center of the network. The network performance with different traffic flows (20, 30, 40, 50, 60) and different simulation times is studied in this scenario.

Figure 6.7: Scenario II: Non-uniform distribution. (a) N_{alive} . (b) e_n .

Fig. 6.7(a), 6.7(b), and 6.8(a) show the N_{alive} , e_n and T_{life} with different numbers of traffic flows, respectively. Compared to RPL, HECRPL significantly prolongs the lifetime of the network, which have many more numbers of alive nodes after the simulation was run for the same duration. In Fig. 6.7(a), a 66% improvement is achieved with HECRPL compared to RPL after the simulation

was run for 700 seconds. When the simulation time is 1000 seconds, HECRPL outperforms RPL at rates starting from 87% and dropping to 50% as the traffic flows increased. The decrease in gain with the increase in traffic flows is because more data packets are delivered with HECRPL, which consumes energy and causes some more nodes to die. HECRPL can maintain good network connectivity and the data packets are still delivered in HECRPL, while the network with RPL lost most of its connectivity once too many nodes died.

Figure 6.8: Scenario II: Non-uniform distribution. (a) T_{life} . (b) P_{pdr} .

Fig. 6.7(b) also shows this feature, where energy conservation of up to 40% is achieved at 700 seconds of simulation time, and there is less energy is conserved at 1000 seconds of simulation time. More energy is consumed as the traffic flows increase. Fig. 6.8(a) shows that HECRPL can efficiently delay the appearance of the first dead node by 35% to 15% longer than RPL as traffic flows increase. The time at which 10% of the nodes dead is also studied and is shown using Fig. 6.8(a), in which HECRPL shows a superiority of performance over RPL of about 46% to 25% as the traffic flow increases. The interruption of communication because of the death of partial nodes' can be efficiently alleviated with HECRPL, by making use of path diversity for efficiently workload balance.

Fig. 6.8(b), 6.9(a), and 6.9(b) show the P_{pdr} , e_n and T_{life} with different network simulation times, respectively. In this case, 30 and 50 traffic flows are studied. Fig. 6.8(b) shows that HECRPL can achieve improvements in reliability of up to 85%, and 50% over RPL when the traffic flows are 30 and 50, respectively. HECRPL can achieve greater energy savings than RPL, which is shown in Fig. 6.9(a). But

Figure 6.9: Scenario II: Non-uniform distribution. (a) e_n . (b) T_{life} .

the improvement is less significant when many nodes in the network become dead nodes. Fig. 6.9(b) shows that HECRPL significantly delays the death of nodes. HECRPL provides much better network connectivity, resulting in better network performance compared to RPL.

Uniform Distribution in a Square Area

In Scenario III (A), 180 nodes are uniformly distributed in a square area ($240\text{m} \times 240\text{m}$). The root is placed in the center of the network. The simulation time is 1500s. The network performances with different link success rates (the mean values of which are 0.3, 0.4, 0.5, 0.6, 0.7) studied by varying the range of the node's *weight*. Different numbers of traffic flows (20, 30, 40, 50, 60, 70) are also investigated in this scenario.

Fig. 6.10(a), 6.10(b), and 6.11(a) show the P_{pdr} , e_n , and N_{alive} respectively with the different mean link success rates. In Fig. 6.10(a), the PDR performance drops in both RPL and HECRPL at the simulation time of 1000 seconds compared with that at 700 seconds. This is a result of more nodes becoming dead as the simulation time runs on, but HECRPL can effectively alleviate the death of an energy constrained node, and shows great improvement compared to RPL, as shown in Fig. 6.10(a). In Fig. 6.10(b), the energy consumption increases slightly as the link success rate increases, because more packets can be successfully delivered with less lossy links. Fig. 6.11(a) shows that slightly more nodes become dead with a

Figure 6.10: Scenario III (A) A uniform distribution with different link success rates. (a) P_{pdr} . (b) e_n .

better link quality, but a great improvement in PDR is achieved at the same time. This is because more packets can be delivered to the destination with a better channel quality, and more nodes will be involved in forwarding packets. Delivering data causes the energy of a node to become depleted, so the number of dead nodes increase slightly.

Figure 6.11: Scenario III (A) A uniform distribution with different link success rates. (a) N_{alive} . (b) T_{life} .

Fig. 6.11(b) shows the time at which 10%, 20%, 30%, 40% and 50% of the nodes in the network become dead in the process of 30 and 50 traffic flows while the mean of the link success rate is 0.6. Fig. 6.11(b) shows that HECRPL can significantly delay the death of nodes. Compared to RPL, HECRPL can delay

the death of nodes by about 50% with 30 traffic flows, and by about 25% with 50 traffic flows. Therefore, better connectivity can be maintained with HECRPL than with RPL after the simulation has been running for a period of time. Fig. 6.12(a) and 6.12(b) show the P_{pdr} and e_n with different numbers of traffic flows at 700 seconds and 1000 seconds, respectively. With the increase in traffic flow numbers, the PDR performance of HECRPL and RPL drops, but HECRPL shows superiority over RPL, as shown in Fig. 6.12(a). Fig. 6.12(b) depicts the e_n of RPL and HECRPL converging with the increase in traffic flow numbers. This is because HECRPL can maintain better network connectivity than RPL. The smaller gain in energy consumption with the increase in traffic flows is because many more packets are delivered with HECRPL than with RPL.

Figure 6.12: Scenario III (A) A uniform distribution with different traffic flows. (a) P_{pdr} . (b) e_n .

In Scenario III (B), we study the impact of the root's position on the routing performance. In this case, the size of the network varies with the same node density. The number of nodes ranges from 60, 100, 140, to 180, and the nodes are uniformly distributed in the network areas, which are $139\text{m} \times 139\text{m}$, $180\text{m} \times 180\text{m}$, $213\text{m} \times 213\text{m}$, and $241\text{m} \times 241\text{m}$, respectively. The number of traffic flows corresponding to each network is from 12, 20, 28, to 36. The simulation time is 1500s. The node's *weight* is randomly assigned to be between 0.3 and 0.9. The $\text{RPL}_{\text{corner}}$ and $\text{HECRPL}_{\text{corner}}$ denote that the root is placed in the corner of the network area with RPL and HECRPL, respectively. The $\text{RPL}_{\text{center}}$ and $\text{HECRPL}_{\text{center}}$ denote that the root is placed in the center of the network area.

Figure 6.13: Scenario III (B) A uniform distribution with different network sizes. (a) T_{life} . (b) e_n . (c) O_r . (d) T_{delay} .

Fig. 6.13 shows the T_{life} , e_n , O_r , and the T_{Delay} with different numbers of nodes. Fig. 6.13(a) shows that it takes less time for 10% of nodes to become dead in RPL_{conner} and HECRPL_{conner} than in RPL_{center} and HECRPL_{center}. When the root is placed in the corner, the number of lower-ranking nodes tends to be smaller than that at the center position of the root. In another words, there will tend to be more nodes near the root if the root is placed in the center rather than in the corner. Those nodes close to the root usually deplete their energy much more quickly than other sensors, which is known as the problem of the energy hole around the root. HECRPL_{conner} and HECRPL_{center} outperform RPL_{conner} and RPL_{center} respectively in terms of alleviating the energy hole problem. Fig. 6.13(b) shows the e_n performance when 10% of the nodes are dead. A great deal of energy is conserved with the help of HECRPL. Fig. 6.13(c) shows that the routing control

overhead increases as the number of nodes increases. Compared to RPL, HECRPL has more overhead because of the use of the cluster formation message (CFM). In Fig. 6.13(d), RPL has a shorter latency than HECRPL, because the coordination of members in a CPS is designed with a waiting timer to avoid duplicate transmissions. In this design, T_{tr} is set to be 0.4 seconds, so that the packet retransmission time is 0.8 seconds, and T_{Aw} is set to be 0.015 seconds. The setting of the timers depends on the application's objective. These timers can be adjusted to further decrease the latency, taking network traffic flows into consideration.

6.5 Summary

This chapter proposed a distributed, reliable, and energy-efficient routing protocol, namely HECRPL. The optimal CPS is selected through a top-down approach in conjunction with the power-level refinement. In HECRPL, the routing decision is made with the goal of minimizing the network's energy consumption. Extensive simulation studies show that HECRPL achieves reliability and energy efficiency simultaneously. Meanwhile, HECRPL significantly prolongs the lifetime of the network and provides robust network connectivity. In addition, HECRPL is effective for one-to-any, many-to-one, or one-to-all types of communication.

Chapter 7

Conclusion and Future Works

This chapter concludes the thesis by summarizing the contributions, and recommending some directions for further research.

7.1 Conclusion

This thesis has studied, designed, implemented and evaluated the performance of energy-efficient and reliable routing solutions for M2M communications. Specifically, this thesis has proposed an framework design that effectively integrates RPL and its latest standardized addition, i.e., P2P-RPL, and this approach substantially paves the way for future research on routing protocol for LLNs. In addition, this thesis also proposed suits of routing protocols that simultaneously achieve reliability and energy-efficiency in LLNs.

This thesis first presented an overview of M2M communications. The technologies, architectures and protocols associated with cellular M2M communications and capillary M2M communications were reviewed. With further investigation on the protocol stacks of capillary M2M communications, the importance of RPL was addressed. It was revealed that the prevalence of LLNs applications essentially require energy conservation and reliable data delivery for efficient routing support.

Routing is of paramount importance in supporting the current and prospective LLNs applications. In this thesis, a comprehensive study on RPL was conducted.

The framework design of RPL in NS-3 was introduced in detail. Furthermore, numerous experimental studies on RPL were carried out. This thesis also identified the importance and difficulties in achieving the energy-efficient and reliable data delivery in LLNs, abiding by severe energy constraints and unreliable wireless medium.

The importance of P2P communication support is increasing due to its widely and crucial usage in current and prospective LLNs applications. To address these applications' requirements, this thesis proposes a hybrid of proactive and reactive routing protocol, i.e., HyRPL. The HyRPL determines the necessity of route discovery by first measuring the cost of route along pre-established DAG. HyRPL provides reliable routing support with significant reduction of routing overhead. Furthermore, by exploiting the region-feature of static M2M networks, ER-RPL was proposed in this thesis. ER-RPL achieved significant energy conservation without compromising the reliability. A distributed self-regioning algorithm was developed to allow a node to estimate its region. With the region-based route discovery developed in ER-RPL, only a subset of nodes in the network is required to participate in the route discovery, which is the key to energy conservation.

The MP2P traffic is a predominant communication paradigm for LLNs applications, as most of them focus on various type of monitoring. In this thesis, CRPL was proposed to enhance the network reliability and reduce the number of re-transmissions. By leveraging path diversity, CRPL selected the optimal CPS that can minimize the number of retransmissions. Subsequently, HECRPL was proposed as an extension of CRPL. A novel routing metric was proposed to precisely measure the energy consumption for end-to-end data delivery. Furthermore, a distributed algorithm that can optimally select the CPS and transmission range was also developed. In addition, an adaptive priority updating scheme was proposed to effectively balance the workload. In the protocol design, both energy consumption and lossy nature of wireless links were taken into consideration. With significant improvement on reliability, HECRPL can efficiently prolong the lifetime of the network.

Extensive simulation studies were conducted to validate the flexibility and effectiveness of proposed protocols. It has been successfully demonstrated that proposed routing protocols can simultaneously achieve higher energy-efficiency and

reliability than the benchmarks.

As a summary, this thesis has proposed four routing protocols, i.e., HyRPL, CRPL, ER-RPL, and HECRPL, with the aim of achieving energy-efficient and reliable routing support. To present their relationship and the applicable conditions, these four proposed routing protocols have been summarized in an overlooked view point in Table 7.1 as below.

Table 7.1: Summary of the proposed routing protocols.

Routing protocol	Type	Based on	Key algorithms proposed	Objective	Applicable conditions on network requirement	Traffic pattern support
HyRPL	Reactive	RPL	Routing cost measurement	Improve network reliability and reduce routing control overhead	No specific requirement on networks	P2P communication
ER-RPL	Reactive	HyRPL	Distributed self-regioning scheme, Region-based route discovery	Reduce energy consumption without compromising the reliability	Network with several nodes with location-awareness capability	P2P communication
CRPL	Proactive	RPL	CPS selection, Cooperative packet forwarding	Improve network reliability and reduce retransmission	Devices with one hop link quality knowledge	MP2P communication
HECRPL	Proactive	CRPL	Optimal CPS and transmit power selection, Energy routing metric	Improve network reliability and reduce energy consumption	Devices with adjustable transmission power	MP2P communication

7.2 Future Works

In this thesis, significant progress has been made in designing energy-efficient and reliable routing protocols for M2M communications. However, since M2M communications is a promising technique in pervasive environments and applications, there are several research directions that need to be explored to complement this thesis. Therefore, this thesis identifies four research topics to be investigated in the future.

Hardware Implementation in Real-life Scenarios

The routing protocols covered in this thesis are implemented and evaluated in the simulation platform of NS-3. Additionally, implementing these protocols on the actual hardware and evaluating them in a real environment is an essential approach to transfer research into practice. Driven by the prevalence of wireless communications and advancement of sociological developments, wireless embedded networks have reached new avenues with hardware devices stabilizing 16-bit micro-controllers, IPv6 addressing, IEEE 802.15.4 radio modules and innovations in the design of routing protocols. The research team at the University of California, Berkeley (UCB) has developed a low-power embedded mote that enables the rapid prototyping with a high degree of synergy [179]. A new module incorporating the first generation of ARM Cortex M4 and LoWPAN radio was developed, named *Storm*. Furthermore, *Storm* is a tiny ($16\text{mm} \times 26\text{mm}$) and combines a 32-bit Cortex M4 micro-controller with 802.15.4 radio, and 64-Mbit flash memory. In addition, the ultra low-power module exports 80 pins via castellated edges, and is designed to be soldered into a larger PCB that contains additional sensors or actuators.

Figure 7.1: The Firestorm mote.

The Firestorm mote is shown in Fig. 7.1. The Firestorm is a development board for *Storm*. Apart from the features offered by *Storm*, the Firestorm adds an NRF 51822 Bluetooth Low Energy (BLE) System on the chip and includes many on-board sensors (temperature, light, and accelerometer, etc.), IPv6 radio, bluetooth, I2C compatibility, Arduino, DI/O pins, etc. A high level overview of *Storm* and Firestorm is shown in Fig. 7.2, referenced from [179].

The fast prototyping of Firestorm is attained with the Lua [166], a simple in-

Figure 7.3: The RPL network with Firestorms.

avoidance using a CSMA-CA scheme. The research on cross layer design of IEEE 802.15.4 and RPL has emerged in recent years. The strategy for efficient topology construction with IEEE 802.15.4 to support RPL was proposed in [161]. In addition, an enhancement of IEEE 802.15.4 is also proposed to allow RPL to own multiple parent nodes concurrently so as to achieve high throughput [174].

It requires necessary adaptations to ensure that RPL can achieve high performance while working with IEEE 802.15.4. As introduced earlier, RPL builds a directed acyclic graph, while IEEE 802.15.4 supports a cluster-tree graph. In addition, duty cycling (also known as sleep scheduling) is a strategy for energy conservation and prolonging the lifetime of the network. In order to conserve energy, IEEE 802.15.4 is designed with low duty cycling feature. However, duty cycling brings challenges to the design of efficient distributed routing protocols. Specifically, duty-cycling has significant influence on end-to-end performance of the network layer protocols, such as high latency, high control overhead, low packet delivery ratio, and low throughput [175]. Existing routing approaches that take duty-cycling into

consideration, normally require network-wide synchronization, resulting in overhead, and it is not suitable for LLNs [176]. In addition, the dynamic duty-cycling introduces critical challenges in handling the dynamic and lossy networks due to intermittent connectivity. Therefore, both duty-cycling and unreliable wireless links significantly complicate the optimization of network performance. The issue of link reliability is mostly neglected while studying the impact of duty cycling [177].

In order to achieve high throughput, a cross layer design is essential to optimize the performance of RPL while working with IEEE 802.15.4. The main objective of a cross-layer protocol design is to achieve short latency, low routing overhead and high packet delivery ratio. Specifically, opportunistic routing takes advantages of the broadcast nature of wireless channel. Each node maintains a forwarder list. Nodes in a forwarder list opportunistically forward the packets, the coordination of which is based on overhearing. In a cross layer design, apart from the link loss rate, the neighbors' sleep schedule should also be taken into consideration for the optimal forwarder list selection. By maintaining the number of awake nodes in each time slot with a maximization of joint link success rate for packet forwarding, high network performance can be achieved abiding by low duty-cycling. It is worthwhile to consider the features of MAC layer while designing routing protocols in future endeavours.

Markov Decision Process Utilization for LLNs

M2M devices operate with limited buffer size in dynamic and lossy environment. The research conducted in this thesis reveals that congestion easily occurs in LLNs as a result of limited buffer size, large amount of retransmissions and burst network load. Packet drop occurs in both node-level and link-level. It is essential to minimize the packet loss, in order to maximize the network throughput. The dynamic and lossy environment, limited memory and buffer size, and severe energy constraints of LLNs make it difficult to mitigate the congestions. This thesis studied the multipath routing to conserve energy and improve network reliability. In accordance with multipath routing, an interesting avenue of research would be to properly model the mutual influence of packet forwarding among nodes in forwarder list on buffer length, with the aim of achieving optimal throughput and minimizing the packet loss.

The Markov Decision Process (MDP) [178] is considered as a superior choice for stochastic optimization to obtain the best action given particular objectives with constraints. In particular, MDP can be used to optimize the routing decision for LLNs so as to achieve the optimal throughput with the consideration of buffer status and link dynamics. Specifically, the routing decision should be made according to the dynamic nature of LLNs. However, traditional routing optimization approaches with MDP typically consider the scenario of single path routing. The widely adopted opportunistic routing techniques has opened up new doors for the design of routing protocols. An interesting avenue of research would be to consider multicast instead of unicast in the network. Mathematical models can also be developed, while considering multicast. It is worthwhile to investigate the optimal routing decision for opportunistic routing in LLNs with MDP.

List of Publications

Journal Papers

1. Ming Zhao, Ivan W.H. Ho and Peter H.J. Chong, “An Energy-efficient Region-based RPL Routing Protocol for Low-Power and Lossy Networks”, *IEEE Internet of Things Journal*, to be published, DOI: 10.1109/JIOT.2016.2593438, 2016.
2. Ming Zhao, Arun Kumar, Peter Han Joo Chong, Rongxing Lu, “A Comprehensive Study of RPL and P2P-RPL Routing Protocols: Implementation, Challenges and Opportunities”. *Journal of Peer-to-Peer Networking and Applications*, DOI: 10.1007/s12083-016-0475-y, 2016.
3. Ming Zhao, Peter Han Joo Chong and Henry C.B. Chan, “An Energy-Efficient and Cluster-parent based RPL with Power-level Refinement for Low-Power and Lossy Networks”, *Journal of Computer Communications*, to be published, DOI: 10.1016/j.comcom.2016.10.007, 2016
4. Ming Zhao, Arun Kumar, Peter Han Joo Chong and Rongxing Lu, “A Reliable and Energy-efficient Opportunistic Routing Protocol for Dense Lossy Networks”, *IEEE Wireless Communication Letter*, to be published, DOI 10.1109/LWC.2016.2625279, 2016.
5. Ming Zhao, Arun Kumar, Tapani Ristaniemi, and Peter Han Joo Chong, “Machine-to-Machine Communication and Research Challenges: A Survey”, *Wireless Personal Communications*, (Submitted).

Conferences

1. Ming Zhao, H. Y. Shwe, and Peter Han Joo Chong, "Cluster-parent based rpl for low-power and lossy networks in building environment", In the Proceedings of the 12th IEEE Annual Consumer Communications and Networking Conference (CCNC), pp. 779-784, 2015.
2. Ming Zhao, G. G. Md. Nawaz Ali, Rongxin Lu, and Peter Han Joo Chong, "An Energy-Efficient and Self-regioning based RPL for Low-power and Lossy Networks", To be appear in the 84th IEEE Vehicular Technology Conference, 2016. (Invited Paper)
3. Ming Zhao, Arun Kumar, G. G. Md. Nawaz Ali, and Peter Han Joo Chong, "A Cloud-based Network Architecture for Big Data Services", To be appear in IEEE International Conference on Big Data Intelligence and Computing (DataCom), 2016.
4. Cherry Ye Aung, G. G. Md. Nawaz Ali, Ming Zhao, Ruo Jun Cai, and Peter Han Joo Chong, "Information Epidemics Control for Data Delivery in Opportunistic Networks", To be appear in the IEEE International Conference on Communications (ICC), 2016.
5. Ruo Jun Cai, G. G. Md. Nawaz Ali, Cherry Ye Aung, Ming Zhao, and Peter Han Joo Chong, "Simulation Study of Routing Attacks Under Network Coding Environment", To be appear in the IEEE International Conference on Communications (ICC), 2016.

Appendix

```
#include<libmsgpack.c>
#include<time.h>
#include<stdlib.h>
#define RPL_SYMBOLS \
{ LSTRKEY("root_func"), LFUNCVAL(rpl_root_func)}, \
{ LSTRKEY("float_func"), LFUNCVAL(rpl_float_func)}, \

#define disrecvport 49152
#define diorecvport 49153
#define dismcastport 49154
#define diobcastport 49155
int dismcast_callback(lua_State *L);
int diobcast_callback(lua_State *L);
int diorecv_callback(lua_State *L);
int disrecv_callback(lua_State *L);
int t_timeout(lua_State *L);

int dio_init(lua_State *L)
{
    /*DIO = { Rank, G/F, DODAG ID, OF(etx), Node ID, Version }*/
    lua_getglobal(L, "DIO");
    int table_index=lua_gettop(L);
    lua_pushstring(L, "rank");
    lua_pushvalue(L, lua_tonumber(L, 1)+1);
    lua_settable(L, table_index);
}
```

```
    lua_pushstring(L, "GF");
    lua_tonumber(L, 2);
    lua_settable(L, table_index);
    lua_pushstring(L, "dodag_id");
    lua_pushvalue(L, 3);
    lua_settable(L, table_index);
    lua_pushstring(L, "etx");
    lua_pushvalue(L, 4);
    lua_settable(L, table_index);
    lua_pushstring(L, "node_id");
    lua_pushlightfunction(L, libstorm_os_getnodeid);
    lua_call(L, 0, 1);
    lua_settable(L, table_index);
    lua_pushstring(L, "version");
    lua_pushvalue(L, 5);
    lua_settable(L, table_index);
    lua_pushvalue(L, table_index);
    lua_setglobal(L, "DIO");
    return 0;
}

int dis_init(lua_State *L)
{
    /*DIS = {Node ID}*/
    lua_createtable(L, 0, 2);
    int table_index=lua_gettop(L);
    lua_pushstring(L, "node_id");
    lua_pushlightfunction(L, libstorm_os_getnodeid);
    lua_call(L, 0, 1);
    lua_settable(L, table_index);
    lua_setglobal(L, "DIS");
    return 0;
}
```

```

int bcast_dio(lua_State *L)
{
    /*multicast address: "ff02::1"*/
    libstorm_net_sendto(socket, payload, ip, port)
    lua_pushlightfunction(L, libstorm_net_sendto);
    lua_getglobal(L, "diobcast_sock");
    lua_pushlightfunction(L, libmsgpack_mp_pack);
    lua_getglobal(L, "DIO");
    lua_call(L, 1, 1);
    lua_pushstring(L, "ff02::1");
    lua_pushnumber(L, diorecvport);
    lua_call(L, 4, 1);
    return 0;
}

int mcast_dis(lua_State *L)
{
    lua_pushlightfunction(L, libstorm_net_sendto);
    lua_getglobal(L, "dismcast_sock");
    lua_pushlightfunction(L, libmsgpack_mp_pack);
    lua_getglobal(L, "DIS");
    lua_call(L, 1, 1);
    lua_pushstring(L, "ff02::1");
    lua_pushnumber(L, disrecvport);
    lua_call(L, 4, 1);
    lua_getglobal(L, "dismflag");
    if (lua_tonumber(L, -1)==1)
    {
        cord_set_continuation(L, mcast_dis, 0);
        return nc_invoke_sleep(L, 5 * SECOND_TICKS);
    }
    else
    {
        return cord_return(L, 0);
    }
}

```

```
    }
}
int i_timeout(lua_State *L)
{
    int t, imin, imax;
    int i= lua_tonumber(L, lua_upvalueindex(1));
    int l_inst= lua_tonumber(L, lua_upvalueindex(2));
    srand(time(NULL));
    lua_getglobal(L, "TRICKLE_INSTANCE");
    int g_inst = lua_tonumber(L, -1);
    if(g_inst!= l_inst)
    {
        return cord_return(L, 0);
    }
    else
    {
        /*i timed out, double i*/
        lua_getglobal(L, "IMIN");
        imin = lua_tonumber(L, -1);
        lua_getglobal(L, "IMAX");
        imax = lua_tonumber(L, -1);
        if((2*i)<(imin*(2^imax)))
        {
            i*=2;
        }
        else
        {
            i = imin * (2^imax);
        }
        t = (rand() % ((i)-(i/2))) + (i/2) + 1;
        lua_pushnumber(L, t);
        lua_pushnumber(L, i);
        lua_pushnumber(L, l_inst);
        cord_set_continuation(L, t_timeout, 3);
    }
}
```

```
        return nc_invoke_sleep(L, (t) * SECOND_TICKS);
    }
}

int t_timeout(lua_State *L)
{
    int t= lua_tonumber(L, lua_upvalueindex(1));
    int i= lua_tonumber(L, lua_upvalueindex(2));
    int l_inst= lua_tonumber(L, lua_upvalueindex(3));
    lua_getglobal(L, "TRICKLE_INSTANCE");
    int g_inst = lua_tonumber(L, -1);
    if(g_inst!= l_inst)
    {
        return cord_return(L, 0);
    }
    else
    {
        lua_getglobal(L, "C");
        lua_getglobal(L, "k");
        if(lua_tonumber(L, -2)<lua_tonumber(L,-1)) // if (c<k)
        {
            lua_pushlightfunction(L, bcast_dio);
            lua_call(L, 0, 0);
        }
        lua_pushnumber(L, i);
        lua_pushnumber(L, l_inst);
        cord_set_continuation(L, i_timeout, 2);
        /*call i_timeout after (i-t) ticks*/
        return nc_invoke_sleep(L, (i-t) * SECOND_TICKS);
    }
}

int trickle_timer(lua_State *L)
{
    int imin, imax,i,t;
```

```
    srand(time(NULL));
    lua_getglobal(L, "IMIN");
    imin = lua_tonumber(L, -1);
    lua_getglobal(L, "IMAX");
    imax = lua_tonumber(L, -1);
    i = (rand() % ((imin * (2^imax)) - imin)) + 1;
    int c=0;
    lua_pushnumber(L, c);
    lua_setglobal(L, "C");
    t = (rand() % ((i) - (i/2))) + (i/2) + 1;
    lua_pushnumber(L, 0);
    lua_setglobal(L, "t_timeout_flag");
    lua_setglobal(L, "i_timeout_flag");
    /*socket for listening to data: eport 49154*/
    lua_pushnumber(L, t);
    lua_pushnumber(L, i);
    lua_getglobal(L, "TRICKLE_INSTANCE");
    cord_set_continuation(L, t_timeout, 3);
    return nc_invoke_sleep(L, t * SECOND_TICKS);
}

/*actions to be performed upon receipt of msg on DIO bcast socket*/
int diobcast_callback(lua_State *L)
{
    char *pay= (char *)lua_tostring(L, 1);
    char *srcip=(char *)lua_tostring(L, 2);
    uint32_t srcport = lua_tonumber(L, 3);
    printf("Received %s from %s on port %u (DIS mcast)\n",
           pay, srcip, (unsigned int)srcport);
    return 0;
}

/*actions to be performed upon receipt of msg on DIS mcast socket*/
int dismcaster_callback(lua_State *L)
```

```
{
    char *pay= (char *)lua_tostring(L, 1);
    char *srcip=(char *)lua_tostring(L, 2);
    uint32_t srcport = lua_tonumber(L, 3);
    printf("Received %s from %s on port %u (DIS mcast)\n",
          pay, srcip, (unsigned int)srcport);
    return 0;
}

/*used for DIO broadcast according to trickle timer*/
int create_diobcast_socket(lua_State *L)
{
    lua_pushlightfunction(L, libstorm_net_udpsocket);
    lua_pushnumber(L, diobcastport);
    lua_pushlightfunction(L, diobcast_callback);
    lua_call(L,2,1);
    lua_setglobal(L, "diobcast_sock");
    return 0;
}

/*create socket for receiving DIS messages*/
int create_disrecv_socket(lua_State *L)
{
    lua_pushlightfunction(L, libstorm_net_udpsocket);
    lua_pushnumber(L, disrecvport);
    lua_pushlightfunction(L, disrecv_callback);
    lua_call(L,2,1);
    lua_setglobal(L, "disrecv_sock");
    return 0;
}

/*create socket for multicasting DIS messages*/
int create_dismcast_socket(lua_State *L)
{

```



```
    lua_pushlightfunction(L, libstorm_net_udpsocket);
    lua_pushnumber(L, dismcasterport);
    lua_pushlightfunction(L, dismcaster_callback);
    lua_call(L, 2, 1);
    lua_setglobal(L, "dismcast_sock");
    return 0;
}

/*create socket for receiving DIO messages*/
int create_diorecv_socket(lua_State *L)
{
    lua_pushlightfunction(L, libstorm_net_udpsocket);
    lua_pushnumber(L, diorecvport);
    lua_pushlightfunction(L, diorecv_callback);
    lua_call(L, 2, 1);
    lua_setglobal(L, "diorecv_sock");
    return 0;
}

/*actions to perform upon receipt of DIO*/
int diorecv_callback(lua_State *L)
{
    lua_pushlightfunction(L, libmsgpack_mp_unpack);
    lua_pushvalue(L, 1);
    lua_call(L, 1, 1);
    int tab_index=lua_gettop(L); //DIO received
    char *srcip = (char *)lua_tostring(L, 2);
    lua_getglobal(L, "dismflag");
    if(lua_tonumber(L, -1)==1)
    {
        lua_pushlightfunction(L, libstorm_net_close);
        lua_getglobal(L, "dismcast_sock");
        lua_call(L, 1, 0);
        lua_pushnumber(L, 0);
    }
}
```

```
        lua_setglobal(L, "dismflag");
    }
    lua_getglobal(L, "DIO");
    int self_dio= lua_gettop(L);
    lua_pushstring(L, "rank");
    lua_gettable(L, self_dio);
    if(lua_tonumber(L, -1)>1) // not a root node
    {
        int s_rank = lua_tonumber(L, -1);
        //add to parent list
        lua_getglobal(L, "parent_table");
        lua_pushstring(L, "node_id");
        lua_gettable(L, tab_index);
        lua_pushstring(L, srcip);
        lua_settable(L, -3);
        lua_setglobal(L, "parent_table");
        lua_pushstring(L, "rank");
        lua_gettable(L, tab_index);
        int p_rank = lua_tonumber(L, -1);
        if((s_rank==-1) || (s_rank > (p_rank+1)))
        {
            lua_pushvalue(L, tab_index);
            lua_setglobal(L, "PrefParent");
            lua_pop(L, 1);
            /*update rank, gf, dodag_id, etx and version*/
            lua_pushlightfunction(L, dio_init);
            lua_pushstring(L, "rank");
            lua_pushnumber(L, lua_tonumber(L, -1)+1);
            lua_pushnumber(L, 1); //GF=1
            lua_pushstring(L, "dodag_id");
            lua_gettable(L, tab_index);
            lua_pushstring(L, "etx");
            lua_gettable(L, tab_index);
            int etx=lua_tonumber(L, -1);
```

```
        lua_pop(L, 1);
        lua_pushnumber(L, etx+1);
        lua_pushstring(L, "version");
        lua_gettable(L, tab_index);
        lua_call(L, 5, 0);
        /*inconsistent state reached, reset trickle timer*/
        lua_getglobal(L, "TRICKLE_INSTANCE");
        lua_pushnumber(L, lua_tonumber(L, -1)+1);
        lua_setglobal(L, "TRICKLE_INSTANCE");
        lua_pushlightfunction(L, trickle_timer);
        lua_call(L, 0, 0);
    }
    else
    {
        /*consistent state, increment trickle counter C*/
        lua_getglobal(L, "C");
        lua_pushnumber(L, lua_tonumber(L, -1)+1);
        lua_setglobal(L, "C");
    }
}

return 0;
}

/*actions to perform upon receipt of DIS*/
int disrecv_callback(lua_State *L)
{
    /* floating node does not respond DIO*/
    lua_getglobal(L, "DIO");
    lua_pushstring(L, "grounded");
    lua_gettable(L, -2);
    if(lua_tonumber(L, -1)==0)
    {
        return 0;
    }
    /*get DIS msg from payload*/
```

```

    char *srcip = (char *)lua_tostring(L, 2);
    lua_pushlightfunction(L, libmsgpack_mp_unpack);
    lua_pushvalue(L, 1);
    lua_call(L, 1, 1);
    /*unicast back DIO*/
    lua_pushlightfunction(L, libstorm_net_sendto);
    lua_getglobal(L, "disrecv_sock");
    lua_pushlightfunction(L, libmsgpack_mp_pack);
    lua_getglobal(L, "DIO");
    lua_call(L, 1, 1);
    lua_pushstring(L, srcip);
    lua_pushnumber(L, diorecvport);
    lua_call(L, 4, 1);
    lua_getglobal(L, "TFLAG");
    if (lua_tonumber(L, -1) == 0)
    {
        lua_pushnumber(L, 1);
        lua_setglobal(L, "TFLAG");
        lua_pushlightfunction(L, trickle_timer);
        lua_call(L, 0, 0);
    }
    return 0;
}

/*actions that take place in a root node*/
int rpl_root_func(lua_State *L)
{
    /*socket for listening to DIS and responding with DIO unicast*/
    lua_pushlightfunction(L, create_disrecv_socket);
    lua_call(L, 0, 0);
    /*socket for broadcasting DIO*/
    lua_pushlightfunction(L, create_diobcast_socket);
    lua_call(L, 0, 0);
    /*initialize DIO table*/
    lua_pushlightfunction(L, dio_init);

```

```
lua_pushnumber(L, 1);
lua_pushnumber(L, 1);
lua_pushlightfunction(L, libstorm_os_getnodeid);
/*dodag id is same as node id for root*/
lua_call(L, 0, 1);
lua_pushnumber(L, 0);
lua_pushnumber(L, 1);
lua_call(L, 5, 0);
/*initialize DIS table*/
lua_pushlightfunction(L, dis_init);
lua_call(L, 0, 0);
lua_pushnumber(L, 0);
lua_setglobal(L, "TFLAG");
/*set trickle timer parameters*/
lua_pushnumber(L, 5);
lua_setglobal(L, "IMIN");
lua_pushnumber(L, 16);
lua_setglobal(L, "IMAX");
lua_pushnumber(L, 1);
lua_setglobal(L, "K");
/*set global trickle instance flag*/
lua_pushnumber(L, 0);
lua_setglobal(L, "TRICKLE_INSTANCE");
/*set flag to show that DIS is not being multicast*/
lua_pushnumber(L, 0);
lua_setglobal(L, "dismflag");
/*set empty parent table*/
lua_newtable(L);
lua_setglobal(L, "parent_table");
/*set empty routing table*/
lua_newtable(L);
lua_setglobal(L, "routing_table");
return 0;
}
```

```
int rpl_float_func(lua_State *L)
{
    /*socket for listening to DIS and responding with DIO unicast*/
    lua_pushlightfunction(L, create_disrecv_socket);
    lua_call(L, 0, 0);
    /*socket for multicasting DIS*/
    lua_pushlightfunction(L, create_dismcast_socket);
    lua_call(L, 0, 0);
    /*socket for receiving DIO unicast / broadcast*/
    lua_pushlightfunction(L, create_diorecv_socket);
    lua_call(L, 0, 0);
    /*socket for broadcasting DIO*/
    lua_pushlightfunction(L, create_diobcast_socket);
    lua_call(L, 0, 0);
    /*initialize DIO table*/
    lua_createtable(L, 0, 6);
    lua_setglobal(L, "DIO");
    lua_pushlightfunction(L, dio_init);
    lua_pushnumber(L, -1); /*rank of floating node*/
    lua_pushnumber(L, 0); /*floating*/
    lua_pushnumber(L, -1); /*not associated to any dodag*/
    lua_pushnumber(L, 10000); /*infinite cost*/
    lua_pushnumber(L, -1); /*no version yet since node is floating*/
    lua_call(L, 5, 0);
    /*initialize DIS table*/
    lua_pushlightfunction(L, dis_init);
    lua_call(L, 0, 0);
    /*set flag to show that DIS is being multicast*/
    lua_pushnumber(L, 1);
    lua_setglobal(L, "dismflag");
    /*start DIS multicast*/
    lua_pushlightfunction(L, mcast_dis);
    lua_call(L, 0, 0);
    /*set TFLAg to 0 showing that trickle timer is not currently running*/
```

```

    lua_pushnumber(L, 0);
    lua_setglobal(L, "TFLAG");
    /*set trickle timer parameters*/
    lua_pushnumber(L, 5);
    lua_setglobal(L, "IMIN");
    /*number of doublings of imin*/
    lua_pushnumber(L, 16);
    lua_setglobal(L, "IMAX");
    /*number of consistent messages for no transmission*/
    lua_pushnumber(L, 1);
    lua_setglobal(L, "K");
    /*set global trickle instance flag*/
    lua_pushnumber(L, 0);
    lua_setglobal(L, "TRICKLE_INSTANCE");
    /*set empty parent table*/
    lua_newtable(L);
    lua_setglobal(L, "parent_table");
    /*set empty routing table*/
    lua_newtable(L);
    lua_setglobal(L, "routing_table");
    return 0;
}

```

References

- [1] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [2] A. Whitmore, A. Agarwal, and L. Da Xu, “The internet of things-a survey of topics and trends,” *Information Systems Frontiers*, vol. 17, no. 2, pp. 261–274, 2015.
- [3] J. Kim, J. Lee, J. Kim, and J. Yun, “M2M service platforms: survey, issues, and enabling technologies,” *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 61–76, 2014.
- [4] D. Boswarthick, O. Elloumi, and O. Hersent, *M2M communications: a systems approach*, John Wiley and Sons, 2012.
- [5] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, and M. Dohler, “Standardized protocol stack for the internet of (important) things,” *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1389–1406, 2013.
- [6] T. Taleb and A. Kunz, “Machine type communications in 3gpp networks: potential, challenges, and solutions,” *IEEE Communications Magazine*, vol. 50, no. 3, pp. 178–184, 2012.
- [7] O. Gaddour and A. Koubâa, “Rpl in a nutshell: A survey,” *Computer Networks*, vol. 56, no. 14, pp. 3163–3178, 2012.
- [8] P. Levis, A. Tavakoli, and S. Dawson-Haggerty, “Overview of existing routing protocols for low power and lossy networks,” *Internet Engineering Task Force, Internet-Draft draftietf-roll-protocols-survey-07*, 2009.
- [9] J. Martocci, P. Mil, N. Riou, and W. Vermeulen, “Building automation routing requirements in low-power and lossy networks,” 2010.
- [10] A. Brandt and J. Buron, “Home automation routing requirements in low-power and lossy networks,” 2010.

- [11] K. Pister, P. Thubert, S. Dwars, and T. Phinney, “Industrial routing requirements in low-power and lossy networks,” Technical Report, 2009.
- [12] A. Boulis, “Castalia: revealing pitfalls in designing distributed algorithms in wsn,” *In the Proceedings of the 5th ACM international conference on Embedded networked sensor systems*, pp. 407–408, 2007.
- [13] K. D. Korte, A. Sehgal, and J. Schönwälder, “A study of the rpl repair process using contikirpl,” *Dependable Networks and Services*, Springer, pp. 50–61, 2012.
- [14] D. Han and O. Gnawali, “Performance of rpl under wireless interference,” *IEEE Communications Magazine*, vol. 51, no. 12, pp. 137–143, 2013.
- [15] Y.B. Chen, K.M. Hou, J.P. Chanet, and K. El Gholami, “A rpl based adaptive and scalable data-collection protocol module for ns-3 simulation platform,” *In NICST 2103 New Information Communication Science and Technology for Sustainable Development: France-China International Workshop*, pp. 8–16, 2013.
- [16] L. Bartolozzi, T. Pecorella, and R. Fantacci, “ns-3 rpl module: Ipv6 routing protocol for low power and lossy networks,” *In the Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques*, pp. 359–366, 2012.
- [17] G. F. Riley and T. R. Henderson, “The ns-3 network simulator,” *Modeling and Tools for Network Simulation*, Springer, pp. 15–34, 2010.
- [18] M. Goyal, E. Baccelli, M. Philipp, A. Brandt, and J. Martocci, “Reactive discovery of point-to-point routes in low power and lossy networks (draft-ietf-roll-p2p-rpl-09),” Internet Draft, Work in Progress, Technical Report, 2012.
- [19] B. She, X. Zhu, X. Ye, W. Guo, K. Su, and J. Lee, “Weighted network voronoi diagrams for local spatial analysis,” *Computers, Environment and Urban Systems*, vol. 52, pp. 70–80, 2015.
- [20] S. Biswas and R. Morris, “Exor: opportunistic multi-hop routing for wireless networks,” *In the Proceedings of ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 133–144, 2005.
- [21] N. Chakchouk, “A survey on opportunistic routing in wireless communication networks,” *IEEE Communications Surveys and Tutorials*, vol. 17, no. 4, pp. 2214–2241, 2015.
- [22] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, “Least-cost opportunistic routing,” Technical Report, 2007.

- [23] S. Panichpapiboon, G. Ferrari, and O. K. Tonguz, "Optimal transmit power in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 10, pp. 1432–1447, 2006.
- [24] C. Song, M. Liu, J. Cao, Y. Zheng, H. Gong, and G. Chen, "Maximizing network lifetime based on transmission range adjustment in wireless sensor networks," *Computer Communications*, vol. 32, no. 11, pp. 1316–1325, 2009.
- [25] A. Jarry, P. Leone, O. Powell, and J. Rolim, "An optimal data propagation algorithm for maximizing the lifespan of sensor networks," *Distributed Computing in Sensor Systems*, Springer, pp. 405–421, 2006.
- [26] T. Wang, R. C. De Lamare, and A. Schmeink, "Alternating optimization algorithms for power adjustment and receive filter design in multihop wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 1, pp. 173–184, 2015.
- [27] S.Y. Lien and K.C. Chen, "Massive access management for qos guarantees in 3gpp machine-to-machine communications," *IEEE Communications Letters*, vol. 15, no. 3, pp. 311–313, 2011.
- [28] C. Lai, R. Lu, D. Zheng, H. Li *et al.*, "Toward secure large-scale machine-to-machine communications in 3gpp networks: challenges and solutions," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 12–19, 2015.
- [29] A. Aijaz and A. H. Aghvami, "Cognitive machine-to-machine communications for internet-of-things: a protocol stack perspective," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 103–112, 2015.
- [30] K.C. Chen and S.Y. Lien, "Machine-to-machine communications: Technologies and challenges," *Ad Hoc Networks*, vol. 18, pp. 3–23, 2014.
- [31] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. Mccann, and K. Leung, "A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities," *IEEE Wireless Communications*, vol. 20, no. 6, pp. 91–98, 2013.
- [32] K. Zheng, F. Hu, W. Wang, W. Xiang, and M. Dohler, "Radio resource allocation in lte-advanced cellular networks with m2m communications," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 184–192, 2012.
- [33] R. Ratasuk, A. Prasad, Z. Li, A. Ghosh, and M. Uusitalo, "Recent advancements in m2m communications in 4g networks and evolution towards 5g," *In the Proceedings of the 18th*

- IEEE International Conference on Intelligence in Next Generation Networks (ICIN)*, pp. 52–57, 2015.
- [34] Y. Zhang, R. Yu, S. Xie, W. Yao, Y. Xiao, and M. Guizani, “Home m2m networks: architectures, standards, and qos improvement,” *IEEE Communications Magazine*, vol. 49, no. 4, pp. 44–52, 2011.
- [35] N.H. Nguyen, Q.T. Tran, J.M. Leger, and T.P. Vuong, “A real-time control using wireless sensor network for intelligent energy management system in buildings,” *In IEEE Workshop on Environmental Energy and Structural Monitoring Systems (EESMS)*, pp. 87–92, 2010.
- [36] F. Ghavimi and H.H. Chen, “M2m communications in 3gpp lte/lte-a networks: Architectures, service requirements, challenges, and applications,” *IEEE Communications Surveys and Tutorials*, vol. 17, no. 2, pp. 525–549, 2015.
- [37] S. Prabhakaran and N. Bhaskaran, “Uwb antennas with band notch characteristics-a study,” *Journal of Network Communications and Emerging Technologies (JNCET)*, vol. 4, no. 2, 2015.
- [38] F. Gebali, “Modeling ieee 802.11 (wifi) protocol,” *Analysis of Computer Networks*, Springer, pp. 371–401, 2015.
- [39] B. Qiao and K. Ma, “An enhancement of the zigbee wireless sensor network using bluetooth for industrial field measurement,” *In the IEEE MTT-S International Microwave Workshop Series on Advanced Materials and Processes for RF and THz Applications (IMWS-AMP)*, pp. 1–3, 2015.
- [40] N. Mahajan and J. Kaur, “A review of 2.4 ghz transmitters for ieee 802.15. 4 low rate wpans,” *In the Proceedings of the Second IEEE International Conference on Advances in Computing and Communication Engineering (ICACCE)*, pp. 28–33, 2015.
- [41] R. G. Machado and A. M. Wyglinski, “Software-defined radio: Bridging the analog–digital divide,” *Proceedings of the IEEE*, vol. 103, no. 3, pp. 409–423, 2015.
- [42] E. Khorov, A. Lyakhov, A. Krotov, and A. Guschin, “A survey on ieee 802.11 ah: An enabling networking technology for smart cities,” *Computer Communications*, vol. 58, pp. 53–69, 2015.
- [43] S. Aust, R. V. Prasad, and I. G. Niemegeers, “Ieee 802.11 ah: Advantages in standards and further challenges for sub 1 ghz wi-fi,” *In the Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 6885–6889, 2012.

- [44] W. Sun, M. Choi, and S. Choi, "Ieee 802.11 ah: A long range 802.11 wlan at sub 1 ghz," *Journal of ICT Standardization*, vol. 1, no. 1, pp. 83–108, 2013.
- [45] M. Togashi, "Visible light transmitter, visible light receiver, visible light communication system, and visible light communication method," Jan. 5 2016, US Patent 9,232,202.
- [46] N. Torabi, K. Rostamzadeh, and V. Leung, "Ieee 802.15. 4 beaconing strategy and the coexistence problem in ism band," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1463–1472, 2015.
- [47] V. B. Mišić, J. Mišić, X. Lin, and D. Nerandzic, "Capillary machine-to-machine communications: the road ahead," *Ad-hoc, Mobile, and Wireless Networks*, Springer, pp. 413–423, 2012.
- [48] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler, "Transmission of IPv6 Packets Over IEEE 802.15.4 Networks," *Internet Proposed Standard RFC 4944*, September 2007.
- [49] J. W. Hui and D. E. Culler, "Extending ip to low-power, wireless personal area networks," *IEEE Internet Computing*, vol. 12, no. 4, pp. 37–45, July 2008.
- [50] I. Group *et al.*, "Standard for part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (lr-wpans)," *ANSI/IEEE*, vol. 802, 2003.
- [51] J. Ko, A. Terzis, S. Dawson-Haggerty, D. E. Culler, J. W. Hui, and P. Levis, "Connecting low-power and lossy networks to the internet," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 96–101, 2011.
- [52] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (coap)," 2014.
- [53] S. Gritzalis and D. Spinellis, "Addressing threats and security issues in world wide web technology," *Communications and Multimedia Security*, Springer, pp. 33–46, 1997.
- [54] Z. Shelby, "Constrained restful environments (core) link format," 2012.
- [55] A. Banerjee, B. Nguyen, V. Gopalakrishnan, S. Kasera, S. Lee, and J. Van der Merwe, "Efficient, adaptive and scalable device activation for m2m communications," *In the Proceedings of the 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 399–407, 2015.
- [56] A. Rico-Alvarino, M. Vajapeyam, H. Xu, X. Wang, Y. Blankenship, J. Bergman, T. Tirronen, and E. Yavuz, "An overview of 3gpp enhancements on machine to machine communications," *IEEE Communications Magazine*, vol. 54, no. 6, pp. 14–21, 2016.

- [57] A. Ijaz, L. Zhang, M. Grau, A. Mohamed, S. Vural, A. U. Quddus, M. A. Imran, C. H. Foh, and R. Tafazolli, "Enabling massive iot in 5g and beyond systems: Phy radio frame design considerations," *IEEE Access*, vol. 4, pp. 3322–3339, 2016.
- [58] Y.-P. E. Wang, X. Lin, A. Adhikary, A. Grövlén, Y. Sui, Y. Blankenship, J. Bergman, and H. S. Razaghi, "A primer on 3gpp narrowband internet of things (nb-iot)," *arXiv preprint:1606.04171*, 2016.
- [59] C. Anton-Haro and M. Dohler, *Machine-to-machine (M2M) Communications: Architecture, Performance and Applications*, Elsevier, 2014.
- [60] H. Shariatmadari, R. Ratasuk, S. Iraji, A. Laya, T. Taleb, R. Jäntti, and A. Ghosh, "Machine-type communications: current status and future perspectives toward 5g systems," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 10–17, 2015.
- [61] S. Vassaki, G. Pitsiladis, S. E. Sagkriotis, and A. D. Panagopoulos, "Future m2m communication networks: Spectrum sharing, random," *Handbook of Research on Next Generation Mobile Communication Systems*, pp. 149, 2015.
- [62] S.-Y. Lien, T.-H. Liau, C.-Y. Kao, and K.-C. Chen, "Cooperative access class barring for machine-to-machine communications," *IEEE Transactions on Wireless Communications*, vol. 11, no. 1, pp. 27–32, 2012.
- [63] N. Zhang, N. Cheng, A. T. Gamage, K. Zhang, J. W. Mark, and X. Shen, "Cloud assisted hetnets toward 5g wireless networks," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 59–65, 2015.
- [64] Y. Liu, C. Yuen, X. Cao, N. U. Hassan, and J. Chen, "Design of a scalable hybrid mac protocol for heterogeneous m2m networks," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 99–111, 2014.
- [65] C. Ide, B. Dusza, M. Putzke, C. Müller, and C. Wietfeld, "Influence of m2m communication on the physical resource utilization of lte," *In the Proceedings of IEEE Wireless Telecommunications Symposium (WTS)*, pp. 1–6, 2012.
- [66] Z. Wang and X. Xu, "Smart home m2m networks architecture," *In the Proceedings of Ninth IEEE International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, pp. 294–299, 2013.
- [67] U. Phuyal, A. T. Koc, M.-H. Fong, and R. Vannithamby, "Controlling access overload and signaling congestion in m2m networks," *In the Proceedings of IEEE Conference Record of*

- the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pp. 591–595, 2012.
- [68] R. Lu, X. Li, X. Liang, X. S. Shen, and X. Lin, “Grs: The green, reliability, and security of emerging machine to machine communications,” *IEEE Communications Magazine*, vol. 49, no. 4, pp. 28–35, 2011.
- [69] Z. M. Fadlullah, M. M. Fouda, N. Kato, A. Takeuchi, N. Iwasaki, and Y. Nozaki, “Toward intelligent machine-to-machine communications in smart grid,” *IEEE Communications Magazine*, vol. 49, no. 4, pp. 60–65, 2011.
- [70] R. Giuliano, F. Mazzenga, A. Neri, and A. M. Vegni, “Security access protocols in iot networks with heterogenous non-ip terminals,” *In the Proceedings of IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 257–262, 2014.
- [71] J. G. Babbitt and H. Abu-Amara, “System and method for dynamic ip address management,” Sep. 9 2003, US Patent 6,618,757.
- [72] G. Moritz, F. Glatowski, C. Lerche, and D. Timmermann, “Beyond 6lowpan: Web services in wireless sensor networks,” *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 1795–1805, 2013.
- [73] S. Singh and K.L. Huang, “A robust m2m gateway for effective integration of capillary and 3gpp networks,” *In the Proceedings of the 5th IEEE International Conference on Advanced Networks and Telecommunication Systems (ANTS)*, pp. 1–3, 2011.
- [74] G. Lu, D. Seed, M. Starsinic, C. Wang, and P. Russell, “Enabling smart grid with etsi m2m standards,” *In IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 148–153, 2012.
- [75] M. Hasan, E. Hossain, and D. Niyato, “Random access for machine-to-machine communication in lte-advanced networks: issues and approaches,” *IEEE Communications Magazine*, vol. 51, no. 6, pp. 86–93, 2013.
- [76] A. G. Gotsis, A. S. Lioumpas, and A. Alexiou, “M2m scheduling over lte: Challenges and new perspectives,” *IEEE Vehicular Technology Magazine*, vol. 7, no. 3, pp. 34–39, 2012.
- [77] C.-Y. Tu, C.-Y. Ho, and C.-Y. Huang, “Energy-efficient algorithms and evaluations for massive access management in cellular based machine to machine communications,” *In the Proceedings of IEEE Vehicular Technology Conference (VTC Fall)*, pp.1–5, 2011.

- [78] N. Zhang, G. Kang, J. Wang, Y. Guo, and F. Labeau, "Resource allocation in a new random access for M2M communications", *IEEE Communication Letters*, vol. 19, no. 5, pp. 843-846, May. 2015.
- [79] C. Oh, D. D. Hwang, and T. Lee, "Joint access control and resource allocation for concurrent and massive access of M2M devices," *IEEE Transactions on Wireless Communications*, vol. 14, no. 8, pp. 4182-4192, August 2015.
- [80] S.-Y. Lien, K.-C. Chen, and Y. Lin, "Toward ubiquitous massive accesses in 3gpp machine-to-machine communications," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 66-74, 2011.
- [81] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. S. Jubert, M. Mazura, M. Harrison, M. Eisenhauer *et al.*, "Internet of things strategic research roadmap," *Internet of Things-Global Technological and Societal Trends*, pp. 9-52, 2011.
- [82] I. Stojmenovic, "Machine-to-machine communications with in-network data aggregation, processing, and actuation for large-scale cyber-physical systems," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 122-128, 2014.
- [83] F. Li, X. F. Chi, J. S. Zhang, and X. D. Li, "Network modeling and performance analysis under time-varying channel and general arrival of m2m services," *IEEE Advanced Materials Research*, Trans Tech Publications, vol. 756, pp. 2687-2692, 2013.
- [84] N. Afrin, J. Brown, and J. Y. Khan, "An adaptive buffer based semi-persistent scheduling scheme for machine-to-machine communications over lte," *In the Proceedings of Eighth International Conference on Next Generation Mobile Apps, Services and Technologies (NG-MAST)*, pp. 260-265, 2014.
- [85] Afrin, Nusrat and Brown, Jason and Khan, Jamil Y, "Design of a buffer and channel adaptive lte semi-persistent scheduler for m2m communications," *In the Proceedings of IEEE International Conference on Communications (ICC)*, pp. 5821-5826, 2015.
- [86] Y.-B. Chen, G.-Y. Lin, and H.-Y. Wei, "Dynamic estimation of unsaturated buffer in context-aware m2m wifi network," *In the Proceedings of IEEE International Conference on Internet of Things (iThings), Green Computing and Communications (GreenCom), and Cyber, Physical and Social Computing (CPSCoM)*, pp. 308-311, 2014.
- [87] S.-Y. Tsai, S.-I. Sou, and M.-H. Tsai, "Effect of data aggregation in m2m networks," *In the Proceedings of 15th IEEE International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pp. 95-99, 2012.

- [88] S. Hamdoun, A. Rachedi, and Y. Ghamri-Doudane, "Radio resource sharing for mtc in lte-a: An interference-aware bipartite graph approach," *In the Proceedings of IEEE Global Communication Conference (IEEE Globecom)*, 2015.
- [89] G. Wu, S. Talwar, K. Johnsson, N. Himayat, and K. D. Johnson, "M2m: From mobile to embedded internet," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 36–43, 2011.
- [90] J. Zander, L. K. Rasmussen, K. W. Sung, P. Mahonen, M. Petrova, R. Jantti, and J. Kro-nander, "On the scalability of cognitive radio: assessing the commercial viability of sec-ondary spectrum access," *IEEE Wireless Communications*, vol. 20, no. 2, pp. 28–36, 2013.
- [91] J. Mitola, "Cognitive radio—an integrated agent architecture for software defined radio," 2000.
- [92] Y. Zhang, R. Yu, M. Nekovee, Y. Liu, S. Xie, and S. Gjessing, "Cognitive machine-to-machine communications: visions and potentials for the smart grid," *IEEE Network*, vol. 26, no. 3, pp. 6–13, 2012.
- [93] E. Z. Tragos and V. Angelakis, "Cognitive radio inspired m2m communications," *In the Proceedings of 16th IEEE International Symposium on Wireless Personal Multimedia Com-munications (WPMC)*, pp. 1–5, 2013.
- [94] B. H. Lee and S.-L. Kim, "Mobility control for machine-to-machine lte systems," *In the Proceedings of 11th European Wireless Conference Sustainable Wireless Technologies (Eu-ropean Wireless)*, VDE, pp. 1–5, 2011.
- [95] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing," Technical Report, 2003.
- [96] Johnson, David B and Maltz, David A, "Dynamic source routing in ad hoc wireless net-works," *Mobile computing*, Springer, pp. 153–181, 1996.
- [97] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," *ACM SIGCOMM computer communication review*, vol. 24, no. 4. pp. 234–244, 1994.
- [98] S. Dawson-Haggerty, A. Tavakoli, and D. Culler, "Hydro: A hybrid routing protocol for low-power and lossy networks," *In the Proceedings of the First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 268–273, 2010.
- [99] S. Ajah, A. Al-Sherbaz, S. Turner, and P. Picton, "Machine-to-machine communications energy efficiencies: the implications of different m2m communications specifications," *International Journal of Wireless and Mobile Computing*, vol. 8, no. 1, pp. 15–26, 2015.

- [100] A. Aijaz, M. Tshangini, M. R. Nakhai, X. Chu, and A.-H. Aghvami, "Energy-efficient uplink resource allocation in lte networks with m2m/h2h co-existence under statistical qos guarantees," *IEEE Transactions on Communications*, vol. 62, no. 7, pp. 2353–2365, 2014.
- [101] C.-Y. Lee and C.-S. Yang, "Distributed energy-efficient topology control algorithm in home m2m networks," *International Journal of Distributed Sensor Networks*, vol. 2012, 2012.
- [102] G. Moritz, F. Golatowski, C. Lerche, and D. Timmermann, "Beyond 6lowpan: Web services in wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 1795–1805, 2013.
- [103] I. D. Chakeres and L. Klein-Berndt, "Aodvjr, aodv simplified," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 3, pp. 100–101, 2002.
- [104] C.-C. Ooi and N. Faisal, "Implementation of geocast-enhanced aodv-bis routing protocol in manet," *In the Proceedings of IEEE Region 10 Conference TENCON*, pp. 660–663, 2004.
- [105] G. Montenegro and N. Kushalnagar, "Aodv for ieee 802.15. 4 networks," *draft-montenegro-lowpan-aodv-00, IETF Internet Draft (Work in progress)*, 2005.
- [106] K. Kim, S. D. Park, G. Montenegro, S. Yoo, and N. Kushalnagar, "6lowpan ad hoc on-demand distance vector routing (load)," *Network WG Internet Draft (work in progress)*, vol. 19, 2007.
- [107] T. H. Clausen and A. C. De Verdière, "The lln on-demand ad hoc distance-vector routing protocol-next generation (loadng)," 2011.
- [108] T. source code repository, "Tymo: DYMO implementation for TinyOS," [Online December 2007] Retrieved from <http://tymo.sourceforge.net/>, .
- [109] C. Gomez, P. Salvatella, O. Alonso, and J. Paradells, "Adapting aodv for ieee 802.15. 4 mesh sensor networks: theoretical discussion and performance evaluation in a real environment," *In the Proceedings of the International Symposium on on World of Wireless, Mobile and Multimedia Networks*, IEEE Computer Society, pp. 159–170, 2006.
- [110] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," *In the Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pp. 1–14, 2009.
- [111] H.-f. Xie, F. Zeng, G.-q. Zhang, and D.-l. Su, "Simulation research on routing protocols in zigbee network," *In the Proceedings of the 6th International Asia Conference on Industrial Engineering and Management Innovation*, Springer, pp. 891–898, 2016.

- [112] R. Fonseca, S. Ratnasamy, D. Culler, S. Shenker, and I. Stoica, “Beacon vector routing: Scalable point-to-point in wireless sensor networks,” *IRB-TR-04-012, Berkeley: Intel Research*, vol. 14, 2004.
- [113] J. Zheng and M. J. Lee, “Will ieee 802.15. 4 make ubiquitous networking a reality?: a discussion on a potential low power, low bit rate standard,” *IEEE Communications Magazine*, vol. 42, no. 6, pp. 140–146, 2004.
- [114] A. Koubaa, A. Cunha, and M. Alves, “A time division beacon scheduling mechanism for ieee 802.15. 4/zigbee cluster-tree wireless sensor networks,” *In the Proceedings of the 19th IEEE Euromicro Conference on Real-Time Systems (ECRTS’07)*, pp. 125–135, 2007.
- [115] J. Yi, T. Clausen, and Y. Igarashi, “Evaluation of routing protocol for low power and lossy networks: Loadng and rpl,” *In the Proceedings of the IEEE Conference on Wireless Sensor (ICWISE)*, pp. 19–24, 2013.
- [116] U. Herberg and T. Clausen, “A comparative performance study of the routing protocols load and rpl with bi-directional traffic in low-power and lossy networks (lln),” *In the Proceedings of the 8th ACM Symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pp. 73–80, 2011.
- [117] M. Vucinic, B. Tourancheau, and A. Duda, “Performance comparison of the rpl and loadng routing protocols in a home automation scenario,” *In the Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)* , pp. 1974–1979, 2013.
- [118] J. W. Hui and D. E. Culler, “Ip is dead, long live ip for wireless sensor networks,” *In the Proceedings of the 6th ACM conference on Embedded network sensor systems*, pp. 15–28, 2008.
- [119] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, “Geographic routing without location information,” *In the Proceedings of the 9th ACM annual international conference on Mobile computing and networking*, pp. 96–108, 2003.
- [120] N. T. Long, N. De Caro, W. Colitti, A. Touhafi, and K. Steenhaut, “Comparative performance study of rpl in wireless sensor networks,” *In the Proceedings of the 19th IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*, pp. 1–6, 2012.
- [121] N. N. Pham, J. Youn, and C. Won, “A comparison of wireless sensor network routing protocols on an experimental testbed,” *In the Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, vol. 2, pp. 276–281, 2006.

- [122] M. Goyal, E. Baccelli, M. Philipp, A. Brandt, and J. Martocci, “Reactive discovery of point-to-point routes in low-power and lossy networks,” Technical Report, 2013.
- [123] N. Cam-Winget, L. Toutain, J. Hui, D. Popa, M. Gillmore, and K. Monden, “Applicability statement for the routing protocol for low power and lossy networks (rpl) in ami networks,” 2015.
- [124] Perahia, Eldad and Gong, Michelle X, “Gigabit wireless LANs: an overview of IEEE 802.11 ac and 802.11 ad,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 15, no. 3, pp. 23–33, 2011.
- [125] D. Cavendish, “Operation, administration, and maintenance of ethernet services in wide area networks,” *IEEE Communications Magazine*, vol. 42, no. 3, pp. 72–79, 2004.
- [126] M. Park, “Ieee 802.11 ah: sub-1-ghz license-exempt operation for the internet of things,” *IEEE Communications Magazine*, vol. 53, no. 9, pp. 145–151, 2015.
- [127] F. Cuomo, A. Abbagnale, and E. Cipollone, “Cross-layer network formation for energy-efficient ieee 802.15. 4/zigbee wireless sensor networks,” *Ad Hoc Networks*, vol. 11, no. 2, pp. 672–686, 2013.
- [128] B. A. Al-Omar, T. Landolsi, and A. Al-Ali, “Evaluation of wimax technology in smart grid communications,” *Journal of Communications*, vol. 10, no. 10, 2015.
- [129] S. S. S. R. Depuru, L. Wang, and V. Devabhaktuni, “Smart meters for power grid: Challenges, issues, advantages and status,” *Renewable and sustainable energy reviews*, vol. 15, no. 6, pp. 2736–2742, 2011.
- [130] V. C. Güngör, D. Sahin, T. Kocak, S. Ergüt, C. Buccella, C. Cecati, and G. P. Hancke, “Smart grid technologies: communication technologies and standards,” *IEEE transactions on Industrial informatics*, vol. 7, no. 4, pp. 529–539, 2011.
- [131] A. Brachman, “Rpl objective function impact on llns topology and performance,” *Internet of Things, Smart Spaces, and Next Generation Networking*, Springer, pp. 340–351, 2013.
- [132] J.-P. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel, “Routing metrics used for path calculation in low-power and lossy networks,” Technical Report, 2012.
- [133] P. Thubert, “Objective function zero for the routing protocol for low-power and lossy networks (rpl),” 2012.
- [134] O. Gnawali, “The minimum rank with hysteresis objective function,” 2012.

- [135] P. A. Levis, N. Patel, D. Culler, and S. Shenker, “Trickle: A self regulating algorithm for code propagation and maintenance in wireless sensor networks,” *Computer Science Division, University of California*, 2003.
- [136] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, “The trickle algorithm,” *Internet Engineering Task Force, RFC6206*, 2011.
- [137] P. Thubert, “Rpl objective function 0,” *draft-ietf-roll-of0-03 (work in progress)*, 2010.
- [138] O. Gnawali and P. Levis, “The minimum rank objective function with hysteresis, draft-ietf-roll-minrank-hysteresis-of-10 work in progress,” *The Minimum Rank Objective Function with Hysteresis, draft-ietf-roll-minrank-hysteresis-of-10 work in progress*, 2011.
- [139] J. Ko, J. Eriksson, N. Tsiftes, S. Dawson-Haggerty, J.-P. Vasseur, M. Durvy, A. Terzis, A. Dunkels, and D. Culler, “Industry: beyond interoperability: pushing the performance of sensor network ip stacks,” *In the Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, pp. 1–11, 2011.
- [140] U. Herberg and T. Clausen, “Study of multipoint-to-point and broadcast traffic performance in the ipv6 routing protocol for low power and lossy networks,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 2, no. 4, pp. 293–305, 2011.
- [141] B. Pavkovic, F. Theoleyre, and A. Duda, “Ieee 802.15. 4 and rpl crossoptimization for reliable opportunistic routing in wsn,” *In the Proceedings of the ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2011.
- [142] S. Gormus, F. Tosato, Z. Fan, Z. Bocus, and P. Kulkarni, “Opportunistic rpl for reliable ami mesh networks,” *Wireless Networks*, pp. 1–18, 2012.
- [143] M. Zhao, H. Y. Shwe, and P. H. J. Chong, “Cluster-parent based rpl for low-power and lossy networks in building environment,” *In the Proceedings of the 12th IEEE Annual Consumer Communications and Networking Conference (CCNC)*, pp. 779–784, 2015.
- [144] F. Ren, T. He, S. Das, and C. Lin, “Traffic-aware dynamic routing to alleviate congestion in wireless sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 9, pp. 1585–1599, 2011.
- [145] P. Misra, N. Ahmed, D. Ostry, and S. Jha, “Characterization of asymmetry in low-power wireless links: an empirical study,” *Distributed Computing and Networking*, Springer, pp. 340–351, 2011.

- [146] T. Watteyne, A. Molinaro, M. G. Richichi, and M. Dohler, "From manet to ietf roll standardization: A paradigm shift in wsn routing protocols," *IEEE Communications Surveys and Tutorials*, vol. 13, no. 4, pp. 688–707, 2011.
- [147] E. Rozner, J. Seshadri, Y. A. Mehta, and L. Qiu, "Soar: Simple opportunistic adaptive routing protocol for wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 12, pp. 1622–1635, 2009.
- [148] G. Schaefer, F. Ingelrest, and M. Vetterli, "Potentials of opportunistic routing in energy-constrained wireless sensor networks," *Wireless Sensor Networks*, Springer, pp.118–133, 2009.
- [149] P. Basu and J. Redi, "Effect of overhearing transmissions on energy efficiency in dense sensor networks," *In the Proceedings of the 3rd ACM international symposium on Information processing in sensor networks*, pp. 196–204, 2004.
- [150] A. Liu, X. Jin, G. Cui, and Z. Chen, "Deployment guidelines for achieving maximum lifetime and avoiding energy holes in sensor network," *Information Sciences*, vol. 230, pp. 197–226, 2013.
- [151] M. Goyal, A. Brandt, and E. Baccelli, "A mechanism to measure the routing metrics along a point-to-point route in a low-power and lossy network," 2013.
- [152] S. Jain and S. R. Das, "Exploiting path diversity in the link layer in wireless ad hoc networks," *Ad Hoc Networks*, vol. 6, no. 5, pp. 805–825, 2008.
- [153] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, no. 4, pp. 419–434, 2005.
- [154] Q. Wang, M. Hempstead, and W. Yang, "A realistic power consumption model for wireless sensor network devices," *In the Proceedings of the 3rd IEEE Annual Communications Society on Sensor and Ad Hoc Communications and Networks (SECON'06)*, vol. 1, pp. 286–295, 2006.
- [155] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," *In the Proceedings of the 33rd annual Hawaii international conference on System sciences*, pp. 10–20, 2000.
- [156] O. Younis and S. Fahmy, "Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

- [157] D. Niculescu and B. Nath, "Dv based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1 4, pp. 267–280, 2003.
- [158] D. M. Blough and P. Santi, "Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc networks," *In the Proceedings of the 8th ACM annual international conference on Mobile computing and networking*, pp. 183–192, 2002.
- [159] N. Baccour, A. Koubaa, L. Mottola, M. A. Zuniga, H. Youssef, C. A. Boano, and M. Alves, "Radio link quality estimation in wireless sensor networks: a survey," *ACM Transactions on Sensor Networks (TOSN)*, vol. 8, no. 4, pp. 34, 2012.
- [160] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis, "Four-bit wireless link estimation," *HotNets*, 2007.
- [161] B. Pavkovic, A. Duda, W.-J. Hwang, and F. Theoleyre, "Efficient topology construction for rpl over ieee 802.15. 4 in wireless sensor networks," *Ad Hoc Networks*, vol. 15, pp. 25–38, 2014.
- [162] J. Zheng and M. J. Lee, "A comprehensive performance study of ieee 802.15. 4," 2004.
- [163] A. Awad, T. Frunzke, and F. Dressler, "Adaptive distance estimation and localization in wsn using rssi measures," *In the Proceedings of the 10th IEEE Euromicro Conference on Digital System Design Architectures, Methods and Tools (DSD)*, pp. 471–478, 2007.
- [164] K. Benkic, M. Malajner, P. Planinsic, and Z. Cucej, "Using rssi value for distance estimation in wireless sensor networks based on zigbee," *In the Proceedings of 15th IEEE international conference on Systems, signals and image processing (IWSSIP)*, pp. 303–306, 2008.
- [165] E. Baccelli, M. Philipp, and M. Goyal, "The p2p-rpl routing protocol for ipv6 sensor networks: Testbed experiments," *In the Proceedings of the 19th IEEE International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 1–6, 2011.
- [166] R. Ierusalimschy, L. H. De Figueiredo, and W. Celes Filho, "Lua-an extensible extension language," *Software Practice and Experience*, vol. 26, no. 6, pp. 635–652, 1996.
- [167] D. E. C. Michael P Andersen, Gabe Fierro, "Main Repositories of Firestorm," [Online 2015] Retrieved from https://github.com/UCB-IoET/ioet_contrib.
- [168] M. P. Andersen, "Storm embedded Lua," [Online 2015] Retrieved from https://github.com/SoftwareDefinedBuildings/storm_elua.

- [169] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.
- [170] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, *Trading structure for randomness in wireless opportunistic routing*, ACM, vol. 37, no. 4, 2007.
- [171] Y. Yan, B. Zhang, J. Zheng, and J. Ma, "Core: a coding-aware opportunistic routing mechanism for wireless mesh networks [accepted from open call]," *IEEE Wireless Communications*, vol. 17, no. 3, pp. 96–103, 2010.
- [172] Y. Lin, B. Li, and B. Liang, "Codeor: Opportunistic routing in wireless mesh networks with segmented network coding," *In the Proceedings of IEEE International Conference on Network Protocols (ICNP)*, pp. 13–22, 2008.
- [173] D. Koutsonikolas, C.-C. Wang, and Y. C. Hu, "Efficient network-coding-based opportunistic routing through cumulative coded acknowledgments," *IEEE/ACM Transactions on Networking (TON)*, vol. 19, no. 5, pp. 1368–1381, 2011.
- [174] B. Pavković, F. Theoleyre, and A. Duda, "Multipath opportunistic rpl routing over ieee 802.15. 4," *In the Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pp.179–186, 2011.
- [175] Y. Sun, O. Gurewitz, S. Du, L. Tang, and D. B. Johnson, "Adb: an efficient multihop broadcast protocol based on asynchronous duty-cycling in wireless sensor networks," *In the Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pp. 43–56, 2009.
- [176] J. Hao, B. Zhang, and H. T. Mouftah, "Routing protocols for duty cycled wireless sensor networks: A survey," *IEEE Communications Magazine*, vol. 50, no. 12, pp. 116–123, 2012.
- [177] K. Han, J. Luo, Y. Liu, and A. V. Vasilakos, "Algorithm design for data communications in duty-cycled wireless sensor networks: A survey," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 107–113, 2013.
- [178] M. Abu Alsheikh, D. T. Hoang, D. Niyato, H.-P. Tan, and S. Lin, "Markov decision processes with applications in wireless sensor networks: A survey," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 3, pp. 1239–1267, 2015.
- [179] M.P. Andersen, G. Fierro, and D. Culler, "System Design for a Synergistic, Low Power Mote/BLE Embedded Platform", *In the Proceedings of the 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 1–12, 2016.