ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE
ENGINEERING AND TECHNOLOGY

PARENT-AWARE ROUTING ALGORITHM FOR RPL IN IOT NETWORKS

M.Sc. THESIS

Necip GOZUACIK

Department of Computer Engineering

Computer Engineering Programme

June 2015

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE
ENGINEERING AND TECHNOLOGY

PARENT-AWARE ROUTING ALGORITHM FOR RPL IN IOT NETWORKS

M.Sc. THESIS

Necip GOZUACIK
(504071528)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Prof. Dr. Sema OKTUĞ

June 2015

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

IOT AĞLARINDA KULLANILAN RPL İÇİN EBEVEYN TEMELLİ YÖNLENDİRME ALGORİTMASI

YÜKSEK LİSANS TEZİ

Necip GOZUACIK
(504071528)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Sema OKTUĞ

Haziran 2015

Necip GOZUACIK, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 504071528 successfully defended the thesis entitled **"PARENT-AWARE ROUTING ALGORITHM FOR RPL IN IOT NETWORKS"**, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Sema OKTUĞ** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. Cem ERSOY** Bogazici University	.....
	Yrd. Doç. Dr. Berk CANBERK Istanbul Technical University	.....
		.....

Date of Submission : **04 May 2015**
Date of Defense : **June 2015**

FOREWORD

Without the help and support of some wonderful people in my life, I would not able to complete this thesis.

During this thesis and my masters degree, Prof. Dr. Sema OKTUĞ has been an ideal advisor for me. Her support, ideas and comments always ignited me to work better.

I would like to thank to my colleague Dr. Oğuzhan Yavuz for his encouragement to return back to my thesis after a few years.

I would like to thank to my friends Ahmet Arış and Sıla Ozen from Department of Computer Engineering. They always supported me and advised.

I would like to thank to my family(my dad, mum, brother and sister) for their support.

Lastly, I would like to thank to my endless love Şule.

June 2015 Necip GOZUACIK

TABLE OF CONTENTS

Page

FOREWORD ..... **vii**
TABLE OF CONTENTS ..... **ix**
ABBREVIATIONS ..... **xi**
LIST OF TABLES ..... **xiii**
LIST OF FIGURES ..... **xv**
SUMMARY ..... **xvii**
ÖZET ..... **xix**
1. INTRODUCTION **1**
1.1 Motivation 1
1.2 Problem Statement 2
1.3 Thesis Structure 3
2. INTERNET OF THINGS (IOT) NETWORKS **5**
2.1 General Overview 5
2.2 IEEE 802.15.4 7
2.3 6LoWPAN 10
2.4 Routing Protocols 14
3. IPv6 ROUTING PROTOCOL FOR LOW-POWER AND LOSSY NETWORKS (RPL) **17**
3.1 Protocol Overview 17
3.2 Routing in RPL 20
3.3 RPL Messages 23
3.4 Routing Metrics and Constraints 27
3.5 Objective Function 29
3.6 Minimum Rank with Hysteresis Objective Function 30
4. PARENT-AWARE OBJECTIVE FUNCTION (PAOF) **33**
4.1 General Overview 33
4.2 Algorithm 34
4.3 Implementation 34
5. PERFORMANCE EVALUATION **39**
5.1 Simulation Environment: Contiki 39
5.1.1 System Overview 40
5.1.2 ContikiRPL 41
5.1.3 Cooja 42
5.2 Performance Metrics 43
5.3 Simulation and Network Setup 44
5.4 Results 46
6. CONCLUSIONS AND FUTURE WORKS **57**

REFERENCES **59**
APPENDICES **67**
APPENDIX A.1 69
CURRICULUM VITAE **75**

ABBREVIATIONS

6LoWPAN : IPv6 over Low power Wireless Personal Area Networks
ACK : Acknowledge
ACL : Access Control List
AODV : Ad-Hoc on Demand Distance Vector
APP : Appendix
BER : Bit Error Rate
CoAP : Constrained Application Protocol
CPU : Central Processing Unit
CSMA/CA : Carrier Sense Multiple Access with Collision Avoidance
DAG : Directed Acyclic Graph
DAO : Destination Advertisement Object
DIO : DODAG Information Object
DIS : DODAG Information Solicitation
DLL : Data Link Layer
DODAG : Destination-Oriented DAG
DODAGID : DODAG Identifier
DSDV : Destination Sequenced Distance Vector
DSR : Dynamic Source Routing
DTSN : Destination Advertisement Trigger Sequence Number
DV : Distance Vector
DYMO : Dynamic MANET On-demand Routing Protocol
ETX : Expected Number of Transmission
GHz : GigaHertz
GPRS : General Packet Radio Service
HTTP : HyperText Transfer Protocol
IANA : Internet Assigned Number Authority
ICMP : Internet Control Message Protocol
ICMPv6 : ICMP Version 6
IEEE : Institute of Electrical and Electronics Engineers
IETF : Internet Engineering Task Force
IoT : Internet of Things

IP : Internet Protocol
IPSO : IP Security Operating System
IPv4 : Internet Protocol version 4
IPv6 : Internet Protocol version 6
IS-IS : Intermediate System to Intermediate System
Kb : Kilobits
Kbps : Kilo Bits Per Second
LBR : Low Power and Lossy Border Router
LLC : Logical Link Control
LLN : Low Power and Lossy Network
LoWPAN : Low power Wireless Personal Area Networks
LR-WPAN : Low Rate WPAN
MAC : Media Access Control
MANET : Mobile Ad-Hoc Networks
MHz : MehaHertz
MOP : Mode of Operation
MQTT : Message Queue Telemetry Transport
MRHOF : Minimum Rank with Hysteresis Objective Function
MTU : Maximum Transmission Unit
ND : Neighbor Discovery
NSA : Node State Attribute
OF : Objective Function
OF 1 : Objective Function 1
OF0 : Objective Function 0
OLSRv2 : Optimized Link State Routing Protocol Version 2
OS : Operating System
OSI : Open System Interconnect
OSPF : Open Shortest Path First
PAN : Personal Area Network
PAOF : Parent Aware Objective Function
PDU : Protocol Data Unit
PER : Packet Error Rate
PHY : Physical Layer
RFC : Request for Comments
RIP : Routing Information Protocol
ROLL : Routing over Low Power and Lossy Links
RPL : IPv6 Routing Protocol for Low-Power and Lossy Networks
TBRPF : Topology Dissemination Based on Reverse-Path Forwarding
TCP : Transmisson Control Protocol
UDP : User Datagram Protocol
uIP : Micro IP
Wi-Fi : Wireless Fidelity
WPAN : Wireless Personal Area Network
WSN : Wireless Sensor Network

LIST OF TABLES

Page

Table 2.1 : Existing Routing Protocols Comparison 15
Table 3.1 : Metrics and Constraints 28
Table 3.2 : Conversion of Metric to Rank 30
Table 4.1 : DODAG Construction Performance 38
Table 5.1 : Simulation network parameters in Cooja 44
Table 5.2 : Preferred Parent Selection for Topology 1 47
Table 5.3 : Preferred Parent Selection for Topology 2 48
Table 5.4 : Preferred Parent Selection for Topology 3 49

LIST OF FIGURES

Page

Figure 1.1 : IoT: Connected World 1
Figure 1.2 : The IoT: Different Services and Technologies 2
Figure 1.3 : IoT Convergence with Too Many Nodes 3
Figure 2.1 : IoT Technology Evaluation 6
Figure 2.2 : IEEE 802.15.4 Layers 7
Figure 2.3 : IEEE 802.15.4 Frequency Distribution 8
Figure 2.4 : IEEE 802.15.4 General Frame Structure 9
Figure 2.5 : The relation of 6LoWPAN to related standards and alliances 10
Figure 2.6 : The 6LoWPAN architecture 11
Figure 2.7 : Extending the Internet Architecture 12
Figure 2.8 : Typical 6LoWPAN Header Stacks 14
Figure 3.1 : A RPL network with three DODAGs in two instances 18
Figure 3.2 : RPL Terminology 20
Figure 3.3 : Example for Building DODAG 21
Figure 3.4 : The operation of a router in a DODAG 23
Figure 3.5 : ICMPv6 Control Messages 24
Figure 3.6 : The DIO message format 25
Figure 3.7 : The DAO message format 26
Figure 3.8 : Trickle Timer Algorithm 27
Figure 3.9 : DODAG with ranks determined from ETX using the MRHOF. 31
Figure 4.1 : Flow Chart of PAOF 34
Figure 4.2 : Definition of rpldag type 35
Figure 4.3 : Implementation of BestParent Function for MRHOF 36
Figure 4.4 : Implementation of BestParent Function for PAOF 37
Figure 4.5 : DODAG Construction for MRHOF 38
Figure 4.6 : DODAG Construction for PAOF 38
Figure 5.1 : The Contiki System 39
Figure 5.2 : Contiki Protocol Stack 41
Figure 5.3 : ContikiRPL in protocol stack 42
Figure 5.4 : DIO sent by root after RPL starts 42
Figure 5.5 : Cooja, the network simulator that comes with Contiki 43
Figure 5.6 : Topology 1 45
Figure 5.7 : Topology 2 45
Figure 5.8 : Topology 3 45
Figure 5.9 : Average Parent Load Density for Negative Exponential 50
Figure 5.10 : Average Packet Delay for Negative Exponential 50
Figure 5.11 : Number of DIO Message for Negative Exponential 51

Figure 5.12: Number of DAO Messages for Negative Exponential 51
Figure 5.13: Parent Diversity for Negative Exponential 52
Figure 5.14: Average Parent Load Density for Pareto 52
Figure 5.15: Average Packet Delay for Pareto 53
Figure 5.16: Number of DIO Message for Pareto 53
Figure 5.17: Number of DAO Messages for Pareto 54
Figure A.1 : Topology 1 69
Figure A.2 : Topology 2 69
Figure A.3 : Topology 3 69
Figure A.4 : Topology 4 70
Figure A.5 : Topology 5 70
Figure A.6 : Topology 6 70
Figure A.7 : Topology 7 70
Figure A.8 : Topology 8 71
Figure A.9 : Topology 9 71
Figure A.10: Topology 10 71
Figure A.11: Topology 11 72
Figure A.12: Topology 12 72
Figure A.13: Topology 13 73
Figure A.14: Topology 14 73
Figure A.15: Topology 15 73
Figure A.16: Topology 16 74
Figure A.17: Topology 17 74

PARENT-AWARE ROUTING ALGORITHM FOR RPL IN IOT NETWORKS

SUMMARY

The number of smart devices connected to Internet is increasing and becoming important day by day. There exists several terminology and definitions for that like Internet of Things (IoT), Internet of Everything (IoE) , Web of Things (WoT), Web of Everything (WoE), Machine-to-Machine (M2M). We generally combine all these definitions under IoT.

IoT is a concept and paradigm that considers pervasive presence of the variety of things/objects that through wireless or wired connections and unique addressing schemes (via IPv6) are able to interact with each other and corporate with others to create new applications/services with a common goals.

The goal of IoT is to enable things to be connected anytime, anyplace, with anything and anyone using any path/network/infrastructure and any service. IoT is a new revolution of the Internet.

IoT is a network of physical objects that contain embedded technology to communicate, interact about their internal states or the external environment using efficient wireless protocols, powerful sensors and cheaper processors.

According to the industry analyst firm IDC, the installed IoT devices will grow up to approximately 212 billion devices by 2020, a number that includes 30 billion connected devices.

The deployment of wireless sensors networks (WSNs) accessible through the Internet is the result of the growing trend towards enabling the concepts of IoT. Wireless sensors are key elements in IoT networks to connect legacy devices/actuators to Internet. This is mostly achieved by introducing IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) technology standard on top of IEEE 802.15.4. Within this, wireless sensor node gained capability to talk to Internet via its IPv6 address through border router nodes running routing protocols to manage data traffic both in upward and downward.

IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) is the IETF proposed standard protocol for IPv6 constrained networks and designed to be used with 6LoWPAN to control routing traffic between sensor nodes and the Internet cloud.

In RPL architecture, Objective Function (OF) methodology determines how RPL nodes translate one or more metrics into ranks, and how to optimize and select routes in a network. In the literature, there are objective functions considering several metrics like OF_0 with hop count, Minimum Rank with Hysteresis Objective Function (MRHOF) with Expected Number of Transmission (ETX) (known, OF_1).

As the number of tiny devices are increasing day by day, to distribute data traffic through the load balanced networks gains importance and becomes crucial. None of the developed and/or implemented objective functions in the literature does consider the parent load density.

This thesis offers a solution to have a load balanced network based on a new Parent-Aware Objective Function (PAOF). The proposed objective function uses the ETX and the number of parent's values on each sensor node to compute the best path in order to route data packets across the network.

We implemented the proposed solution in Contiki OS and evaluated it by means of Cooja simulations, and compared, the results obtained with that of MRHOF.

Moreover, various network topologies and traffic types are studied to have an idea about the performance of the introduced approach.

Simulation results verified that the proposed solution provides promising results in term of parent load density, parent diversity and end-to-end delay.

IOT AĞLARINDA KULLANILAN RPL İÇİN EBEVEYN TEMELLİ YÖNLENDİRME ALGORİTMASI

ÖZET

İnternet'e bağlı akıllı cihazların sayısı günden güne artmakta ve önem kazanmaktadır. Bu konu ile ilgili çeşitli terminolojiler literatürde yer almaktadır : IoT (Nesnelerin İnterneti), IOE (Her Şeyin İnterneti), WoT (Nesnelerin Ağı), WoE (Her Şeyin Ağı), M2M (Makineden Makineye) vb. En popüler isimlendirme olarak IoT kullanılmaktadır.

IoT ortamındaki en yaygın paradigma ve dikkate alınması gereken konu hemen hemen bir çok cihazın/aygıtın kablosuz veya kablolu bağlantılar yoluyla internete erişmeleridir. Bunun tabi tam tersi de geçerlidir. İnternet üzerinden de bu cihazlara ulaşılabilir olunması. Kablosuz duyargaların internet ile kontrol edilmesinde IPv6 adresi üzerinden bağlantı kurulumu yapılmaktadır genellikle.

Nesnelerin internete bağlanması ile gerçeklik kavramında da bazı değişiklikler gündeme gelecektir. Gerçek ve sanal dünya oldukça birbirine yakınsar olacaktır. Enerji, ulaşım, şehircilik ve diğer birçok alanlarda yer alan uygulamalar, sistemler ve yapılar daha akıllı ortamlar yaratacaktır.

IoT'nin en önemli hedeflerinden biri her zaman her yerde internete bağlanacak nesneler sağlamaktır. Bunun için altyapı ve erişebilirlik durumumun da oldukça iyi olması gerekir. IoT teknolojisini özet olarak yeni internet devrimi olarak düsünebiliriz.

Bir endüstri analiz firması olan IDC'nin yapmış olduğu bir çalışmaya göre, 2020 yılına kadar 212 milyar sayısına ulaşacak cihaz olması ve bunların da 30 milyar kadarının internete bağlı olacağı öngörülmekte. Bu cihazlarin neler olduğuna baktığımızda karşımıza bir çeşitlilik çıkmakta: klasik bilgisayarlar, akıllı telefonlar, robotlar, insansız hava aracı (drone), canlılar, sensörler vb.

Klasik kablosuz sensör ağlarının internete bağlanabilme özellikleri kazanmaları ve bu konularda yapılan akademik ve sanayi çalışmaları IoT yaklaşımının getirdiği bir sonuçtur.

Kablosuz sensörler IoT sistemlerinin en önemli yapılarından biridir. Kablosuz sensörler yardımıyla eski tür cihazlar, aygıtlar ve eşyalar internet ağına dahil olabilme özellikleri kazanmalari sayesinde IoT ekosisteminin bir parçası olabilirler.

Kablosuz sensörlerin internete bağlanmaları ve taşınabilmeleri ile ilgili IETF organizasyonu tarafından bir çalışma grubu oluşturulmuştur. Bu çalışma grubuna 6LoWPAN ismi verilmiştir ve hedefi düşük güçlü kişisel kablosuz alan ağlarının IPv6 mimarisi ile uyumlu hale getirilmesi olmuştur. 6LoWPAN yapısı daha önce IEEE tarafından standartlarılmış 802.15.4 protokolü üzerine inşa edilmiştir. Standarda ve yığın yapısına getirilen özellikler ile IEEE 802.15.4 uyumlu çalışabilen kablosuz sensör yapıları internet protokolü ile artık iletişim kurabilmektedir.

IEEE 802.15.4 OSI mimarisinde birinci ve ikinci katmanlar olan fiziksel ve veri bağı katmanlarını gerçeklerken; 6LoWPAN adaptasyon katmanı da bu iki katman ile Ağ katmanı arasında bir ara yüz görevi üstlenmenktedir. Kablosuz sensör düğümlerinin kendi doğası gereği mevcut yönlendirme protokolleri ile çalışması pek mümkün değildir. Bundan dolayı IETF, ROLL (Routing over Low Power and Lossy Links / Düşük Güç ve Kayıplı Ağlar Üzerinde Yönlendirme) ismini verdiği başka bir çalışma grubu ile RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks / Düşük Güç ve Kayıplı Ağları için IPv6 Yönlendirme Protokolü) standartlarını IoT sistemleri için geliştirmiştir.

RPL protokolünün temel işlevi kablosuz sensörler ile internet bulutu arasındaki veri trafiğinin yönlendirilmesini sağlamaktır. Bunu yaparken de kablosuz sensörlerin düşük güç tüketimi, kayıplı olması gibi özelliklerini de göz önünde bulundurmaktadır.

RPL mimarisi içerisinde yönlendirme ve iletim yolunun belirlenebilmesi için objective function (amaç fonkisyonu) ismi verilen yaklaşım kullanılmaktadır. OF ile yapılan işlem, düğümlerin ya da link hatlarının çeşitli özelliklerine (hat kalitesi, atlama sayısı vb.) göre karar verip en uygun yolun iletim hattı için seçilmesidir.

Literatürde ve endüstride RPL gerçeklemelerinde en çok bilinen iki tane amaç fonksiyonu vardır. OF_0 olarak da adlandırılan ilk metot düğümler arasındaki atlama/uzaklık seviyesine bakarken; OF_1 ya da diğer bilinen isimlendirmesi ile MRHOF (Histeresiz Amaç Fonksiyonu ile Minimum Derece) ise hat kalitesine bakmaktadır.

Kablosuz sensör düğümleri yardımı ile internete bağlanabilen cihaz sayısı günden güne artmaktadır. Bu da IoT ağlarında, topolojilerinde yer alan sensör düğümlerin verimli kullanılabilmesi ihtiyacını doğurmaktadır. Mevcut amaç fonksiyonu çözümlerinden hiç biri sensör düğümünün yoğunluğu, iletişimde olduğu ebeveyn sayısı ile direkt olarak ilgilenmemektedir. Bir sensörün ebeveyn sayısının fazla olması onun veri iletiminde daha fazla çakışma yaşamasına neden olabilir. Link kalite değeri (ETX) birbirlerine yakın olan iki sensörden ebeyevn sayısı fazla olanın seçilmesi yukarıda bahsedilen probleme zaman zaman neden olabilir. Ayrıca topolojinin daha az ebeveyn ile temsil edilmesi dengesiz bir yapı oluşturmasına da neden olmuş olur.

Bu tez, ile yeni bir OF metodu önerilmiştir. PAOF (Parent Aware Objective Function / Ebeveyn Temelli Amaç Fonkiyounu) ismi verilen bu yöntem hem ETX değeri hem de ebeveyn sayısına bakarak bir sensörün yönlendirme için seçilmesine karar verir. Böylelikle seçilme şansı olmayan sensörler de tercih edilen ebeveyn olarak belirlenip ağdaki toplam yönlendirme yapacak olan sensör sayısı artırılır. Bu da daha dengeli bir yapı ile birlikte sensörlerin üzerine düşecek olan veri iletimi yoğunluğunu da azaltmış olur.

Tercih edilen ebeveyn sayısının artması, yaşanacak çarpışma ve çakışmalar da iyileştirme yapacağı için paketlerin merkezi düğüme iletilmesindeki gecikmeyi de genel anlamda azaltacaktır. PAOF yöntemi OSI mimarisinden aşina olduğumuz 3 numaralı katman olan Ağ Katmanında çalışmış olacaktır.

Önerilen bu yeni yöntem kablosuz sensör ağları ve IoT için uzunca bir süredir kullanılagelen Contiki işletim sisteminde geliştirildi. Bu işletim sisteminde yer alan ContikiRPL mimarisi üzerinde yapılan yazılımsal geliştirmeler ile PAOF çalıştırılmış oldu. Simulasyon olarak da Contiki ile beraber gelen Cooja ortamı kullanıldı.

PAOF yöntemi simulasyon aracılığı ile literatürde yer alan ve en yaygın şekilde kullanılan MRHOF ile karşılaştırıldı. Simulasyonlar sırsında hem farklı topolojiler hem de farklı trafikler denendi. Böylelikle çözümün olabildiğince farklı sahalarda kullanılma deneyimi tecrübe edilmiş olundu.

Elde edilen sonuçlarda PAOF yönteminin özellikle ortalama ebeyeyn yük yoğunluğu, ebeveyn çeşitliliği ve de uçtan uca gecikme parametrelerinde MRHOF yönteminden genellikle daha başarılı ve verimli olduğu gözlendi.

1. INTRODUCTION

In this chapter, first of all motivation behind this thesis work is talked about. Existing problem statement in the describing area is identified generally. In the last section, thesis structure is given.

1.1 Motivation

WSN [1] is a distributed, self-organized network of small, energy-constrained [2] nodes that collect and generate data [3]. With the rising of IoT platforms [4], wireless sensor nodes are started to be used frequently in the field or different application types like transport, manufacturing, building, agriculture, biomedical [5].

IoT systems consist of different mediums residing in OSI (Open System Interconnect) [6] Layer 1 and Layer 2. That requires supporting inter-working and interoperability with also participation of upper layer protocols HTTP (HyperText Transfer Protocol), CoAP (Constrained Application Protocol) [7], MQTT (Message Queue Telemetry Transport) [8].

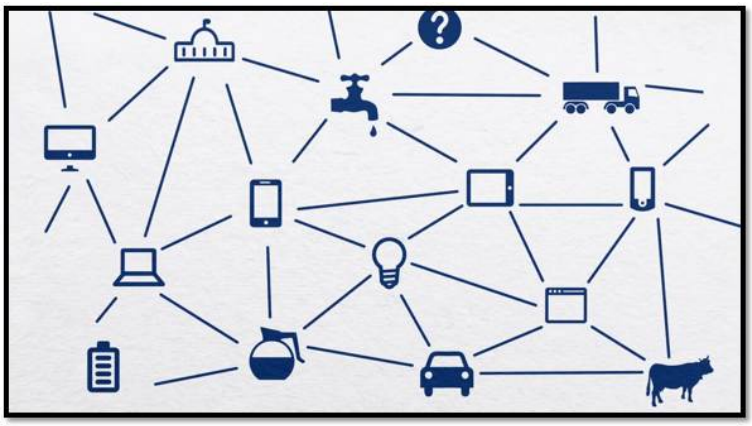


Figure 1.1: IoT: Connected World
[9]

6LoWPAN [10] is an important technology standard used within wireless sensor networks in IoT platforms. In 6LoWPAN, routing protocol at network layer has importance to send the generated data by the nodes towards to the Internet cloud

through sink/border router node. For 6LoWPAN, RPL [11] is the most preferred routing protocol at the network layer.

RPL determines the path routes based on objective function methodology [12] defined in RFC 6550. The mostly used objective function is MRHOF [13] which decides to select paths based on the ETX [14] value.

Different traffic types spanning on many applications need to use efficient routing algorithm to manage the data load properly. RPL is a very popular/hot protocol and still needs investigation and enhancement on several topics outlined in the standard. Objective function is one of the important topics in the item list. As sensor and application requirements need to collaborate mutually, an efficient objective function gains importance.

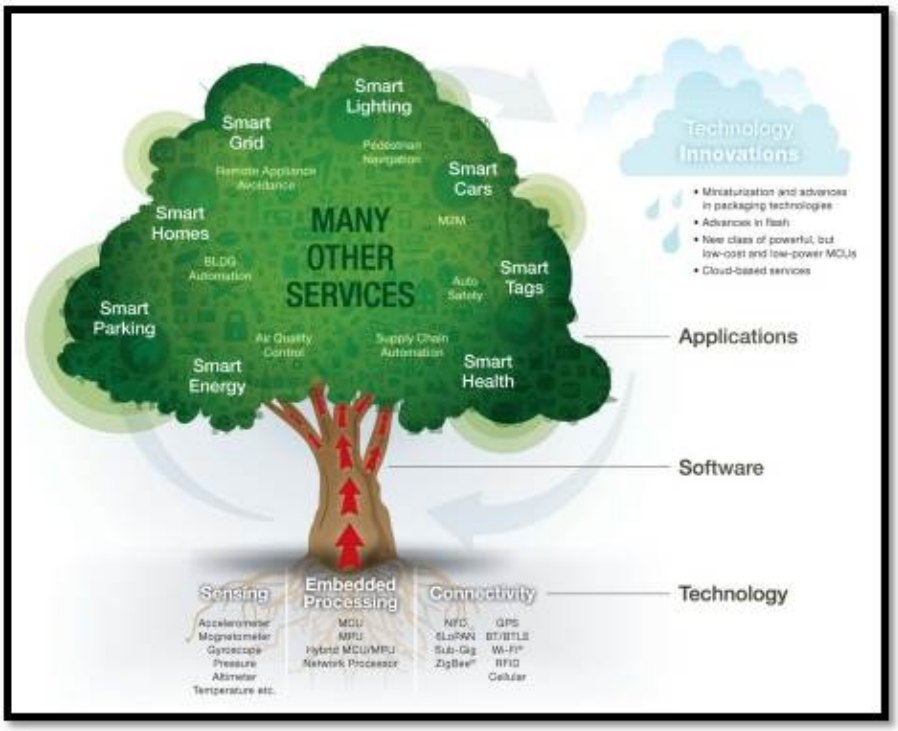


Figure 1.2: The IoT: Different Services and Technologies [15]

1.2 Problem Statement

As the number of requirements increase for the IoT networks in real life scenarios and applications [16], this situation requires to use too many sensor nodes to control all over the system.

Figure 1.3: IoT Convergence with Too Many Nodes [15]

Increasing in the number of sensor nodes in a network makes difficult to utilize all of the nodes effectively during data traffic through network lifetime [17]. For example in an IoT network, let's consider that sensor nodes are placed closely and have ETX values in similar range. In such a case, probably the same parent node list having better ETX value is always used to distribute the data traffic towards the sink node if MRHOF is used as objective function technique in RPL. In this situation, some nodes in the network would remain unemployed and this may lead to an unbalanced load distribution. Furthermore, in the long term, utilizing the same parent node list may cause energy efficiency problems, reduced network lifetime and congestion [18].

In order to achieve load balancing, we propose a new objective function called PAOF for RPL. Regarding different network topologies, the proposed PAOF ensures a better load balanced network, diversity of parent selection and reduce the end-to-end delay.

1.3 Thesis Structure

The rest of the thesis is organized as follows. Section 2 describes IoT networks and introduces IEEE 802.15.4, 6LoWPAN and Routing protocols. Section 3 describes RPL and gives information about routing, messaging, objective function and MRHOF. In Section 4, we present the proposed routing, objective function called PAOF in RPL. In Section 5, firstly we give information about Contiki OS and simulation tool Cooja and present an experimental performance evaluation to show the impact of the proposed technique against the MRHOF. Finally, Section 6 concludes the work and discusses future research items.

2. INTERNET OF THINGS (IOT) NETWORKS

In this chapter, firstly a general overview is introduced regarding to IoT. After then IEEE 802.15.4, 6LoPWPAN, the routing protocols for IoT networks are described.

2.1 General Overview

IoT is a recent communication paradigm that envisions a near future, in which the objects of everyday life will be equipped with micro controllers, transceivers for digital communication, and suitable protocol stacks that will make them able to communicate with one another and with the users, becoming an integral part of the Internet.

The IoT concept, hence, aims at making the Internet even more pervasive. Furthermore, by enabling easy access and interaction with a wide variety of devices for instance home appliances, surveillance cameras, monitoring sensors, actuators, displays, vehicles, and so on, the IoT will foster the development of a number of applications that make use of the potentially enormous amount and variety of data generated by such objects to provide new services to citizens, companies, and public administrations. This paradigm indeed finds application in many different domains, such as home automation, industrial automation, medical aids, mobile health care, elderly assistance, intelligent energy management and smart grids, automotive, traffic management, and many others [19].

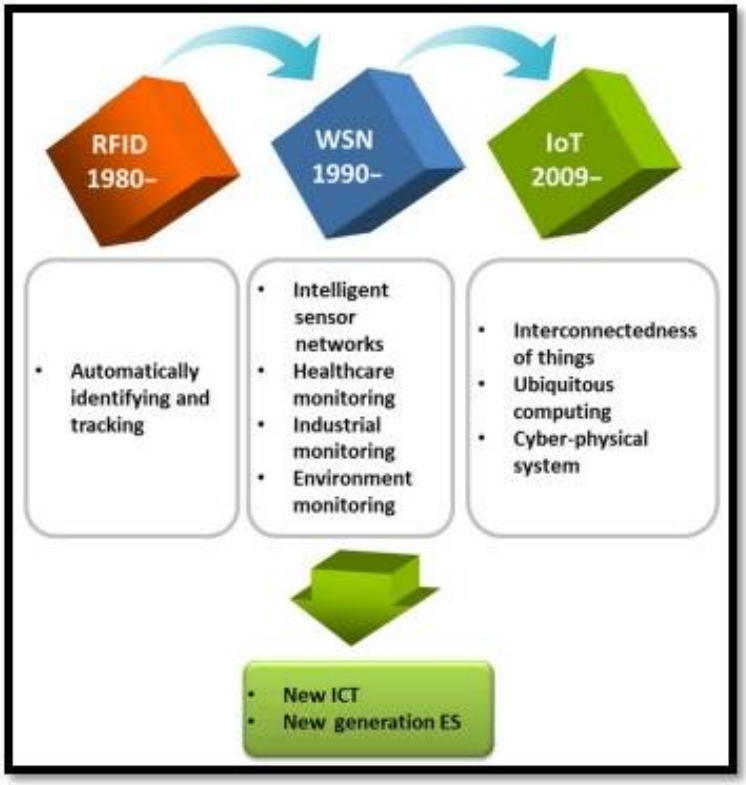


Figure 2.1: IoT Technology Evaluation [20]

A constrained/wireless sensor network also called LLN (Low Power and Lossy Network) [21] is a class of networks consisting of nodes to monitor physical or environmental conditions like also known as cyber-physical systems [22] nowadays. All of this system can be evaluated and summarized under the IoT umbrella.

Generally monitored parameters are temperature, humidity, pressure and power-line voltage, and vital body functions, etc. Sensor nodes are equipped with a transducer, microcomputer, transceiver and power source [23]. The size of these devices is generally very small and powered by either battery, energy scavenging like solar cells or mains powered. Based on these properties, the devices are resource constrained and therefore they need a very efficient use of the resources in term of power, battery, memory.

A wireless sensor/LLN network mostly support three types of traffic: Point-to-Point (between devices inside the LLN), Point-to-Multi Point (through sink node towards leaves) and Multi Point-to-Point (through leaf nodes towards sink) [24]. Since the devices have limited range of transmission, therefore Routing is required in these devices to communicate/reach each other. Routing is responsible for managing the

routes among sensor nodes and forwarding the packets on the most efficient route discovered.

The radio medium used by LLN/WSN devices is of short range and also very susceptible to bit errors. The lossy nature of LLN has a strong impact on the routing protocol design. Since the link failures are frequent and usually transient, therefore the routing protocol should not overreact in an attempt to converge the network as a result of temporary failures.

Due to these reasons, one of the challenging issues in IoT networks is finding the best routes for the delivery of data, which implies a very efficient routing mechanism for finding and keeping the routes in the network. The routing mechanism is subject to both the resource constraint nature of sensor nodes and lossy nature of the radio medium in LLN.

2.2 IEEE 802.15.4

The IEEE 802.15.4 standard [25] introduces physical and media access control (MAC) layers of low rate wireless personal area networks (LR-WPANs) [26].

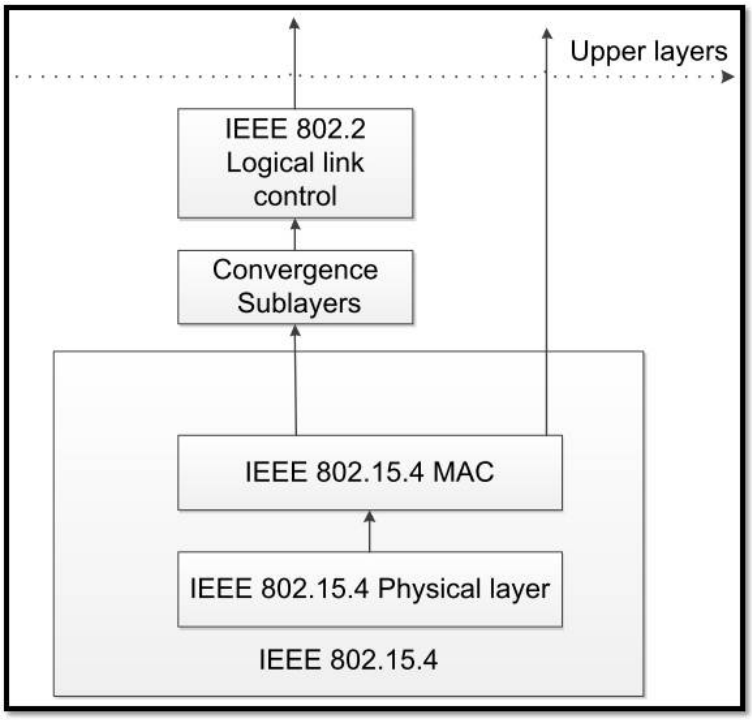
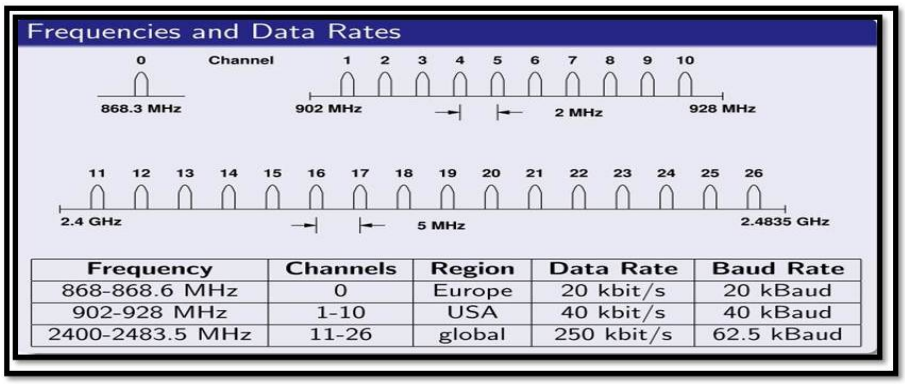


Figure 2.2: IEEE 802.15.4 Layers
[27]

This standard was developed considering characteristics of LR-WPANs such as low data rate, low power consumption, energy efficient. In order to suit these characteristics, the standard adopts techniques such as reduction in frequency and amount of data transfer, reduced frame overhead and strict power saving mechanisms such as duty cycling.

For the physical layer, the standard uses the unlicensed 2.4 GHz band for worldwide operation and the 868/915 MHz bands for Europe and United States respectively. Additional bands were added in later revisions. The 868/915 MHz and other bands would help in case of interference from other technologies associated with the 2.4 GHz band. The 2.4 GHz bands provide a transmission rate of 250 Kb/s while the 868/915 MHz bands provide the rate of 20 Kb/s and 40 Kb/s respectively. The 868/915 MHz bands, together support 11 channels while the 2.4 GHz band supports 16 channels, hence providing a total of 27 channels across the three bands. Frequency band diagram can be found in Figure 2.3.



Frequency	Channels	Region	Data Rate	Baud Rate
868-868.6 MHz	0	Europe	20 kbit/s	20 kBaud
902-928 MHz	1-10	USA	40 kbit/s	40 kBaud
2400-2483.5 MHz	11-26	global	250 kbit/s	62.5 kBaud

Figure 2.3: IEEE 802.15.4 Frequency Distribution [28]

The data link layer (DLL) is divided into two sub layers: the MAC and the logical link control (LLC). The logical link control layer is defined in other standards and the IEEE 802.15.4 defines only the MAC sublayer. The network topology in LR-WPAN can either be a star topology or some sort of extended connected topology such as mesh, ring or cluster. In order to allow for these type of topologies, the MAC frame also called the MAC protocol data unit (PDU) is kept very flexible.

There are four types of frames supported: data, beacon, acknowledgment and MAC command frames. The latter two frame types are used for MAC communication and only the data and beacon frames contain data from higher layers. The entire MAC

PDU should not exceed 127 bytes. A general frame structure can be seen in Figure 2.4:

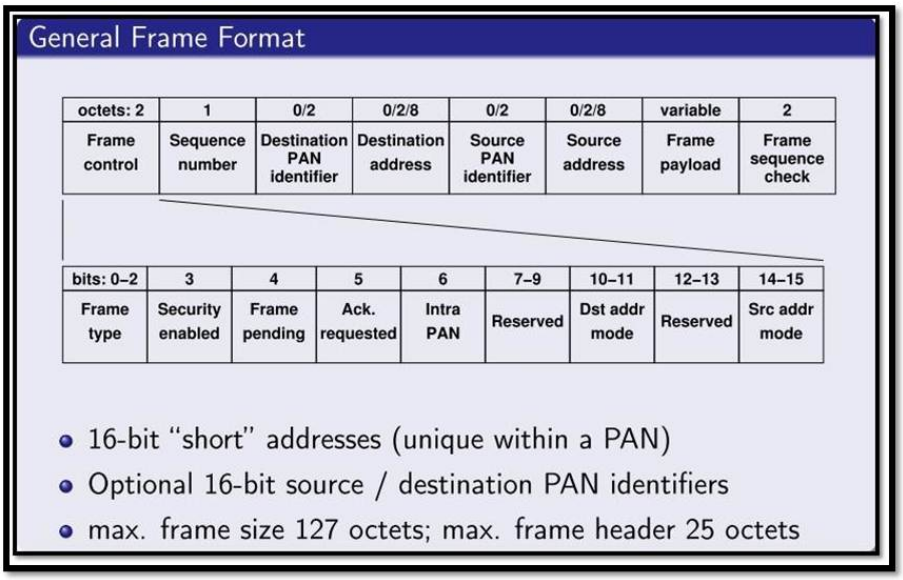


Figure 2.4: IEEE 802.15.4 General Frame Structure [28]

The standard supports two types of channel access mechanisms: non-beacon-enabled and beacon-enabled mode. In case of beacon enabled mode, the data frames are sent using the slotted carrier sense multiple access with collision avoidance (CSMA/CA) method [29]. In this method, the nodes are synchronized by the beacon frames sent by a special node called the coordinator. In non-beacon-enabled network, the data frames are sent using the un-slotted CSMA/CA method. In this case, whenever a collision is detected, the nodes back-off for a random time before retrying.

The standard supports various security suites which can be broadly categorized into four types: no security, encryption only security, authentication only security and security with both encryption and authentication. The radio chips have to implement access control lists (ACL) that contains information regarding which security suite has to be used. However, radio chip designers do not have to support all the security suites.

2.3 6LoWPAN

Using IPv6 on top of IEEE 802.15.4 has one major drawback in that 802.15.4 has an MTU of only 127 bytes, whereas the MTU for IPv6 is 1280 bytes [30]. As a consequence, IPv6 packets might have to be fragmented before they can be sent over a 802.15.4 link. Another problem with the small MTU is that encapsulated IPv6 packages, with their 128 bit addresses, take up a big portion of the available 127 bytes. A single IPv6 header is 40 bytes long (with no extended headers), and if TCP [31] is used as well, this header adds another 20 bytes. The overhead for TCP/IP communication then adds up to 60 bytes.

In order to find a solution to the above problems, a working group named 6LowPAN [32] was appointed, by the IETF, to create an adaptation layer between IPv6 and IEEE 802.15.4. The fundamentals of this layer is specified in [33], and the main purpose is to make communication over 802.15.4 links fulfill the requirements stated by IPv6.

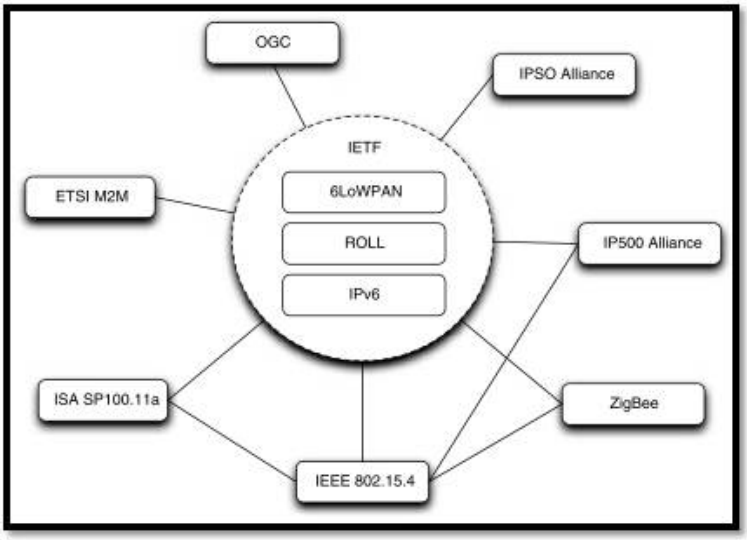


Figure 2.5: The relation of 6LoWPAN to related standards and alliances [32]

6LoWPAN radically makes improvement by introducing an adaptation layer between the IP stack's link and network layers to enable efficient transmission of IPv6 datagrams over 802.15.4 links, dramatically reducing IP overhead [34]. The adaptation layer is an IETF proposed standard and provides header compression to reduce the transmission overhead, fragmentation to support the IPv6 minimum MTU requirement, and support for layer-two forwarding to deliver and IPv6 datagram over multiple radio

hops [33]. 6LoWPAN achieves low overhead by applying cross-layer optimizations; it uses information in the link and adaptation layers to compress network- and transport-layer headers. Drawing on IPv6 extension headers, it employs the header stacking principle to separate the orthogonal concepts and keep the header small and easy to parse.

A general architectural view of 6LoWPAN can be seen in Figure 2.6.

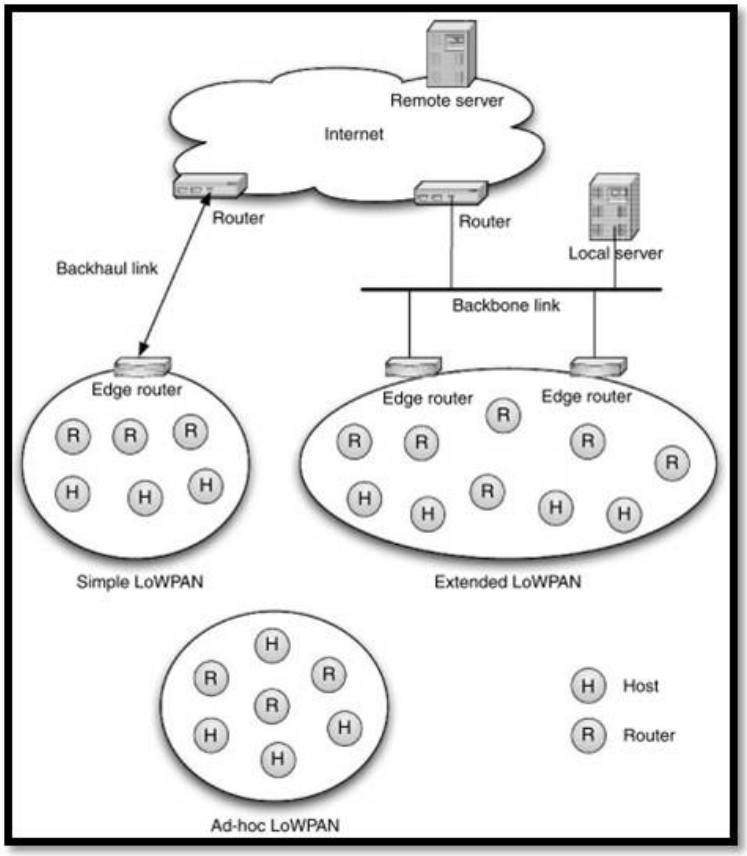


Figure 2.6: The 6LoWPAN architecture
[32]

By communicating natively with IP, LoWPAN networks are connected to other IP networks simply by using IP routers. LoWPANs will typically operate on the edge, acting as stub networks. The LoWPAN may be connected to other IP networks through one or more border routers that forward IP datagrams between different media. Connectivity to other IP networks may be provided through any arbitrary link, including Ethernet, Wi-Fi, GPRS, or satellite. Because 6LoWPAN only specifies operation of IPv6 over IEEE 802.15.4, border routers may also implement Stateless IP/ICMP Translation [35] or other IPv6 transition mechanisms to connect 6LoWPAN

networks to IPv4 networks. Above connection properties can be summarized in Figure 2.7.

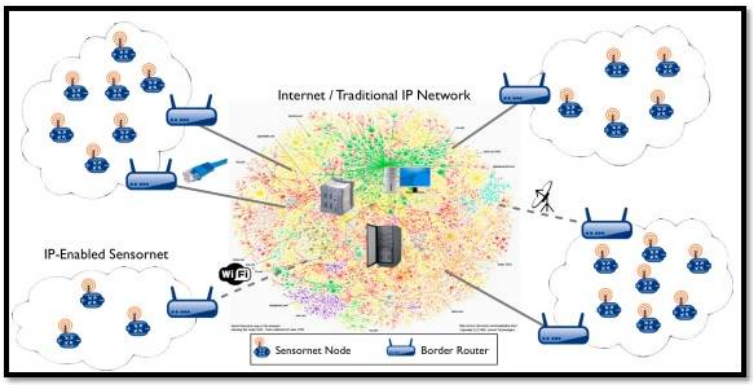


Figure 2.7: Extending the Internet Architecture [36]

6LoWPAN Adaptation Layer

The 6LoWPAN format defines how IPv6 communication is carried in 802.15.4 frames and specifies the adaptation layer's key elements [36]. 6LoWPAN has three primary elements:

Header Compression: IPv6 header fields are compressed by assuming usage of common values. Header fields are elided from a packet when the adaptation layer can derive them from link-level information carried in the 802.15.4 frame or based on simple assumptions of shared context.

Fragmentation: IPv6 packets are fragmented into multiple link-level frames to accommodate the IPv6 minimum MTU requirement.

Layer-2 Forwarding: To support layer-2 forwarding of IPv6 datagrams, the adaptation layer can carry link-level addresses for the ends of an IP hop. Alternatively, the IP stack might accomplish intra-PAN routing via layer-3 forwarding in which each 802.15.4 radio hop is an IP hop.

The key concept applied throughout the 6LoWPAN adaptation layer [37] is the use of stateless or shared-context compression to elide adaptation, network and transport layer header fields, - compressing all three layers down to a few bytes, combined. We can see that it is possible to compress header fields to a few bits when we observe that they often carry common values, reserving an escape value for when less-common ones appear. Common values occur due to frequent use of a subset of IPv6 functionality (such as UDP, TCP, and ICMPv6 as Next Header values) and simple assumptions

of shared context (for example, a common network prefix assigned to the entire LoWPAN). 6LoWPAN also absorbs redundant header information across protocol layers (for instance, UDP and IPv6 length fields and IPv6 addresses are derived from lower-layer headers).

Traditional IP header compression techniques are stateful and generally focus on optimizing individual flows over a highly constrained link. These methods assume that the compressor and de-compressor are in direct and exclusive communication and compress both network and transport layer headers together. They optimize for long-lived flows by exploiting redundancies across packets within a flow over time, requiring the endpoints to initially send packets uncompressed.

Flow-based compression techniques are poorly suited for LoWPANs. Traffic in many LoWPAN applications is driven by infrequent readings or notifications, rather than long-lived flows. Communication over multiple hops requires hop-by-hop compression and decompression and per-flow state at each intermediate node.

Many LoWPAN routing protocols obtain receiver diversity via rerouting, which would require state migration and reduce compression effectiveness. In contrast, stateless and shared-context compression in 6LoWPAN does not require any per-flow state and lets routing protocols dynamically choose routes without affecting compression efficiency. Looking at 6LoWPAN's specifics, we can see how extensively it employs stateless compression.

Encapsulation Header Format

6LoWPAN uses header stacking to keep orthogonal concepts separate and enforce a well-defined method for expressing its capabilities [36]. Analogous to IPv6 extension headers, 6LoWPAN expresses each capability in a self-contained sub-header: mesh addressing, fragmentation, and header compression. Mesh addressing supports layer-2 forwarding, and fragmentation supports the IPv6 minimum MTU requirement. 6LoWPAN identifies all header formats using a header type field placed at the beginning of each header.

The header stack is simple to parse and allows elision of headers when unneeded. The fragmentation header is elided for small datagrams, indicating that a single frame carries the entire payload. Similarly, the mesh header is elided when 6LoWPAN frames

are delivered over a single radio hop, so the path source and destination are identical to those in the link-layer header. Figure 2.8 shows typical header stacks.

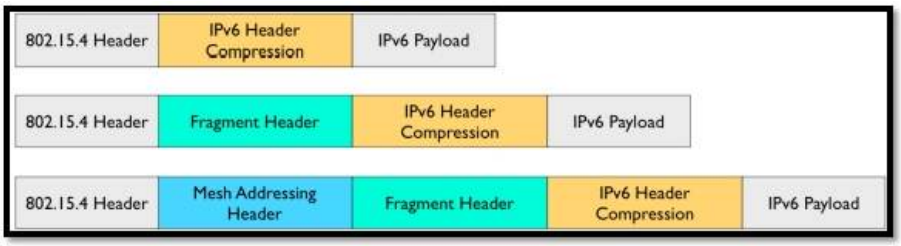


Figure 2.8: Typical 6LoWPAN Header Stacks [36]

Mesh Under vs. Route Over Two important architectural issues for IPv6 over LoWPAN are how link-level factors inform routing and at what layer datagram forwarding occurs within the LoWPAN. Traditionally, IP routing occurs at the network layer in a manner largely independent from the underlying links that implement the individual hops. 6LoWPAN, in its role as an adaptation between the link (layer two) and the network (layer three), can support routing at either layer. Their performance may differ based on network topology and application requirements.

2.4 Routing Protocols

Limited memory and communication capabilities constrain the routing state at each node as well as the routing information that might be communicated. These restrictions preclude using protocols that rely on complete link-state information.

Traditional distance vector mobile ad-hoc networks (MANET) protocols [38] are also ill-suited because they assume a high rate of mobility for all nodes in the network, whereas LoWPAN nodes are better characterized by more structured mobility within a set of stationary nodes. Consequently, MANET protocols use frequent floods to discover and maintain routes. Caches used to optimize communication only trade memory for communication. In addition, most of these protocols exchange route maintenance information at rates that far out typical LoWPAN communication and react to link fading with expensive route-repair actions.

Instead, LoWPAN routing protocols must operate using incomplete information and tolerate some inconsistency. Interestingly, we are returning to scalability issues similar to those encountered with the early Internet, but this time in a wireless setting. The

new Routing over Low Power and Lossy Links (ROLL) working group within the IETF routing directorate has already addressed these challenges.

Routing is considered as one of the critical items in 6LoWPAN networks There are many academic works on this area [39] [40]. In the past, there have been several routing protocols for 6LoWPAN-compliant LLNs, such as Hydro [41] and Hilow [42]. Unfortunately, these solutions are not able to fulfill every requirement expected from IoT networks.

IETF ROLL working group investigated and compared existing routing protocols such as OSPF [43], OLSRv2 [44], TBRPF [45], RIP [46], AODV [47], DSDV [48], DYMO[-low] [49], DSR [50]. Based on comparison, none of them met the IoT requirements as shown in Table 2.1.

Table 2.1: Existing Routing Protocols Comparison

Name	Table Size	Loss Response	Control Cost	Link Cost	Node Cost
OSPF	fail	fail	fail	pass	fail
OLSRv2	fail	fail	fail	pass	pass
TBRPF	fail	pass	fail	pass	?
RIP	fail	fail	fail	?	fail
AODV	pass	?	pass	fail	fail
DSDV	fail	fail	fail	?	fail
DYMO	pass	fail	pass	fail	fail
DSR	fail	?	pass	fail	?

To address the most part of the requirements and open items, the IETF ROLL working group [51] proposed a routing protocol, referred to as RPL, RPL is designed for networks with lossy links, which are those exposed to high Packet Error Rate (PER) and link outages. This property meets the base requirement from the WSNs and IoT applications.

3. IPv6 ROUTING PROTOCOL FOR LOW-POWER AND LOSSY NETWORKS (RPL)

In this chapter, firstly protocol overview is described for RPL. In latter parts, detailed information about RPL is noted including routing and messaging details. Routing metrics and constraints are described and then Objective Function is introduced. Lastly, we mention about MRHOF which is the method to be compared with our proposed technique.

3.1 Protocol Overview

RPL is a distance–vector (DV) and a source routing protocol that is designed to operate on top of several link layer mechanisms including IEEE 802.15.4 PHY and MAC layers [52]. It targets collection-based networks, where nodes periodically send measurements to a collection point, as well as point-to-multi-point traffic from the central point to the devices inside the LLN. Point-to-point traffic is also supported in RPL.

A key feature in RPL is that it represents a specific routing solution for low power and lossy networks, which stand for network with very limited resources in terms of energy, computation and bandwidth turning them highly exposed to packet losses. In fact, it has been specifically designed to meet the requirements of resource-constrained nodes as mentioned in the routing requirement terminology document. In particular, RPL-enabled LLNs take into account two main features (i) the prospective data rate is typically low (less than 250 kbps), and (ii) communication is prone to high error rates, which results in low data throughput. A lossy link is not only characterized by a high Bit Error Rate (BER) but also the long inaccessibility time, which strongly impacts the routing protocol design. In fact, the protocol was designed to be highly adaptive to network conditions and to provide alternate routes, whenever default routes are inaccessible.

RPL is based on the topological concept of Directed Acyclic Graphs (DAGs). The DAG defines a tree-like structure that specifies the default routes between nodes in the LLN. More specifically, RPL organizes nodes as Destination-Oriented DAGs (DODAGs), where most popular destination nodes (i.e. sinks) or those providing a default route to the Internet (i.e. gateways) act as the roots of the DAGs.

A network may consist of one or several DODAGs shown below, which form together an RPL instance identified by a unique ID, called RPLInstanceID [11]. A network may run multiple RPL instances concurrently; but these instances are logically independent. A node may join multiple RPL instances, but must only belong to one DODAG within each instance.

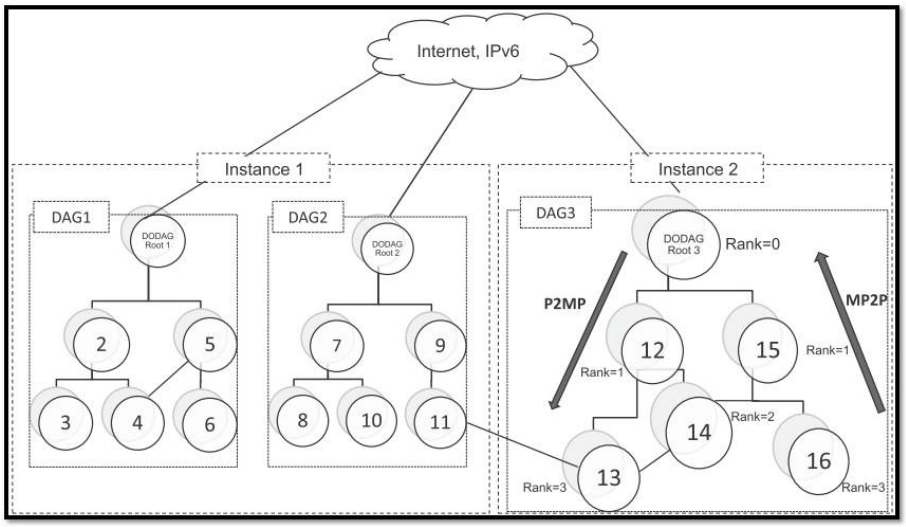


Figure 3.1: A RPL network with three DODAGs in two instances [52]

RPL defines three types of nodes:

- Low Power and Lossy Border Routers (LBRs): it refers to the root of a DODAG that represents a collection point in the network and has the ability to construct a DAG. The LBR also acts as a gateway (or edge router) between the Internet and the LLN.
- Router: it refers to a device that can forward and generate traffic. Such a router does not have the ability to create a new DAG, but associate to an existing one.
- Host: it refers to an end-device that is capable of generating data traffic, but is not able to forward traffic.

The basic topological component in RPL is the DODAG, a Destination Oriented DAG, rooted in a special node called DODAG root, as illustrated in Figure 3.1. The DODAG root has the following properties:

- typically acts as an LBR,
- represents the data sink within the directed acyclic graph, (iii) it is typically the final destination node in the DODAG, since it acts as a common transit point that bridges the LLN with IPv6 networks, (iv) it has the ability to generate a new DODAG that trickles downward to leaf nodes.

Each node in the DODAG is assigned a rank. The rank of a node is defined in [52] as the node's individual position relative to other nodes with respect to a DODAG root. It is an integer that represents the location of a node within the DODAG. The rank strictly increases in the downstream direction of the DAG, and strictly decreases in the upstream direction. In other words, nodes on top of the hierarchy receive smaller ranks than those in the bottom and the smallest rank is assigned to the DODAG root.

The architecture of a DODAG is similar to a cluster-tree topology where all the traffic is collected in the root. However, the DODAG architecture differs from the cluster-tree in the sense that a node can be associated not only to its parent (with higher rank), but also to other sibling nodes (with equal ranks). The rank is used in RPL to avoid and detect routing loops, and allows nodes to distinguish between their parents and siblings in the DODAG. In fact, RPL enables nodes to store a list of candidate parents and siblings that can be used if the currently selected parent loses its routing ability.

In the construction process of network topology, each router identifies a stable set of parents on a path towards the DODAG root, and associates itself to a preferred parent, which is selected based on the Objective Function defined in standard [11].

The Objective Function defines how RPL nodes translate one or more metrics into ranks, and how to select and optimize routes in a DODAG. It is responsible for rank computation based on specific routing metrics (e.g. delay, link quality, connectivity, etc.) and specifying routing constraints and optimization objectives. The design of efficient Objective Functions is still an open research issue. In [53], the draft proposes to use the ETX [54] and required to successfully transmit a packet on the link as the

path selection criteria in RPL routing. The route from a particular node to the DODAG root represents the path that minimizes the sum of ETX from source to the DODAG root. In [55], the draft proposes objective function 0 (OF_0), which is only based on the abstract information carried in an RPL packet, such as Rank. OF_0 is agnostic to link layer metrics, such as ETX, and its goal is to foster connectivity among nodes in the network.

A general view about RPL terminology including above definitions can be seen and summarized in Figure 3.2.

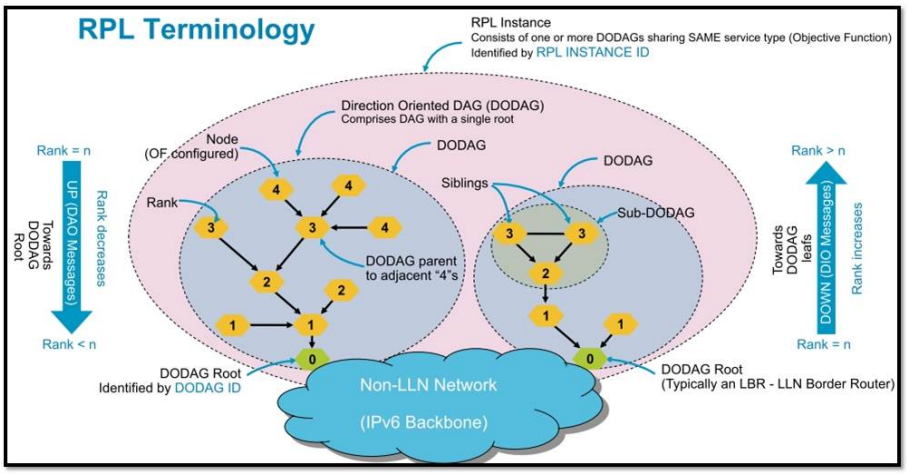


Figure 3.2: RPL Terminology
[56]

Supported traffic flow types are Point-to-Point, Point-to-Multi Point and Multi Point-to-Point.

3.2 Routing in RPL

Topology formation in RPL starts with designating one node as root node. The root node determines the configuration parameters for the network. The configuration is packed into a DODAG Information Object (DIO) message [11], which is then used to disseminate the information in the network. There are many options which can be configured in a DIO to tailor the network configuration to the application's requirements. The compulsory information contained in a DIO comprises amongst others:

- RPLInstanceID for which the DIO is sent,
- the DODAGID of the RPLInstance of which the sending node is part,

- the current DODAG version number, and
- the node's rank within the DODAG.

The RPLInstanceID is a unique identifier of an RPL Instance in a network. The DODAGID serves the same purpose: to uniquely identify a DODAG in an RPL Instance. A node's rank describes its logical distance from the root node within the DODAG. When traversing the DODAG from the root node towards the leaf nodes, the rank of nodes is monotonically increasing. When forming the DODAG, each node is required to select parent nodes from its neighbors. Afterwards, when the node is calculating its rank, this has to be larger than the rank of all its parents. In this way, the formation of loops in the routing structure is prevented. Note that rank is not necessarily related to the physical distance, nor to the distance in hops between a node and the root node, but a metric determining a node's desirability (in terms of application goals, which might, e.g., be load balancing for energy preservation) as a next hop on a route to the root node. A node's rank is calculated based on the Objective Function (OF), which is specified according to the DODAG's application goals.

The objective function is therefore one of the hooks which can be used to tailor RPL closely to serve a specific application. Not only does the objective function contain the parameters for calculating a node's rank, it is also responsible for selecting a node's parents by describing the desirability of a neighboring node to be chosen as parent and therefore be part of a route towards the root node. To give an example, a node's energy level or its type of power resource could be used in the objective function to calculate its rank 2. For the sake of simplicity, the hop count distance between a node and the root node are chosen as determining parameter for the objective function in the following example step by step.

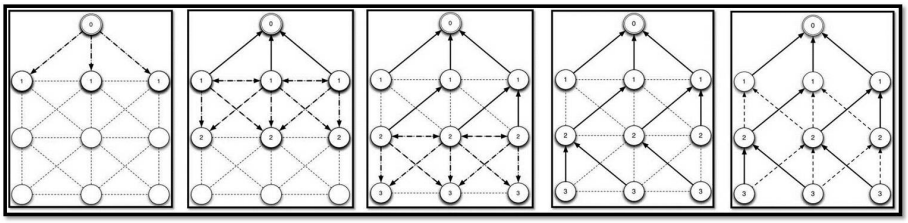


Figure 3.3: Example for Building DODAG

The root node triggers the DODAG formation by broadcasting a DIO message to its neighbors. Note that only the root node of a DODAG is allowed to initiate the

discussion of DIOs. Whilst the RPLInstanceID and the DODAGID remain unchanged throughout the whole topology formation, the rank field is updated, as the DIO messages are traversing the network. Since the root node has a distance of 0 to itself, its rank is set to 0. Each neighbor receiving the DIO calculates its rank according to the objective function by computing its hop count distance to the root node and sets its rank to 1.

After calculating its rank, each node updates the DIO and broadcasts it to its neighbors. Each node retains a candidate neighbor set, in which it keeps track of the neighbors with lower or equal rank it has heard of (i.e., from which it has received a DIO message). Out of this candidate neighbor set, each node selects parent nodes, which have to have a lower rank than the node itself.

From the parent set, the node picks a so-called preferred parent, which serves as the node's next hop when routing a data packet towards the root. This choice is determined by the objective function. In the example, the neighbors of the root node only know of one node fulfilling this condition, so they pick the root as their preferred parent.

All nodes of the network have received DIO messages and joined the DODAG by calculating their rank, whilst the nodes with hop count distance 2 have picked their preferred parents. Note that the preferred parent must be unique.

Each node keeps its set of parent nodes, to which it can resort as next hops in case its preferred parent becomes unreachable.

With all nodes having joined the DODAG, the topology formation is complete. For this iteration which was initiated by the root node. It can happen that node failures or changing environmental conditions create the need to rebuild the routing topology. To help the nodes keep track of which DODAG iteration they are in, and to determine whether it is the newest one, a version number is written in the DIO message. Note that only the root node is allowed to increment the version number in order to trigger a rebuild of the DODAG. So whenever a node receives a DIO message containing a newer version number than the one it recorded, it can add the sender of this DIO to its candidate neighbor set and might even select it as parent.

The overall routing operation in DODAG can be seen in Figure 3.4.

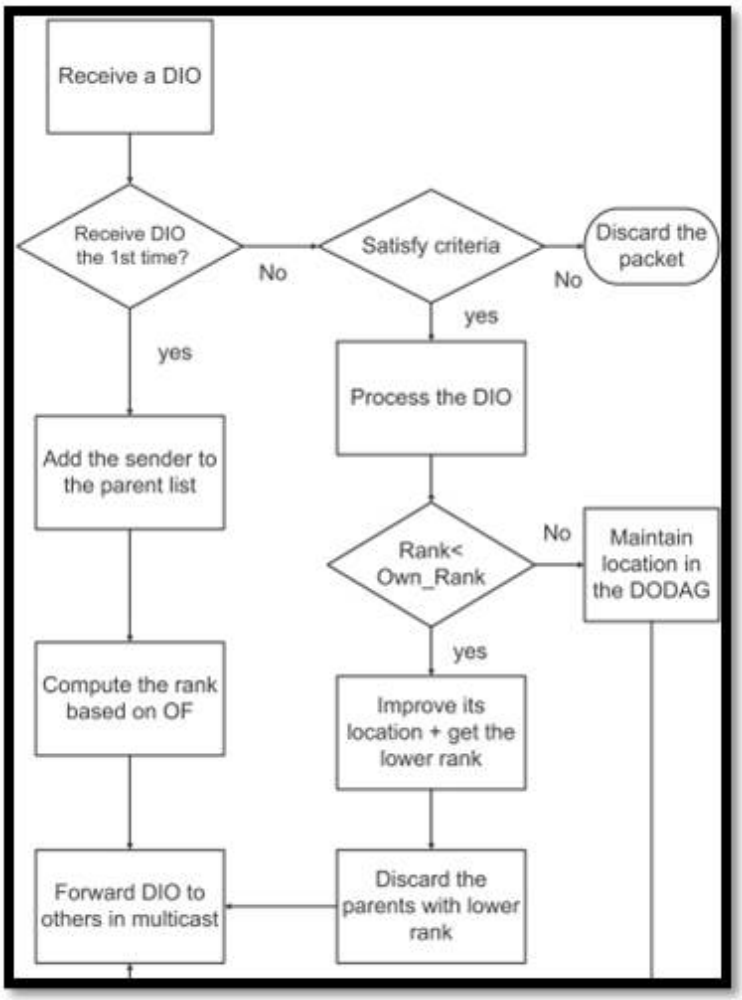


Figure 3.4: The operation of a router in a DODAG [52]

3.3 RPL Messages

RPL messages are specified as a new type of ICMPv6 control messages are shown in Figure 3.5. According to [57] , the RPL control message is composed of

- an ICMPv6 header, which consists of three fields: Type, Code and Checksum
- a message body comprising a message base and a number of options.

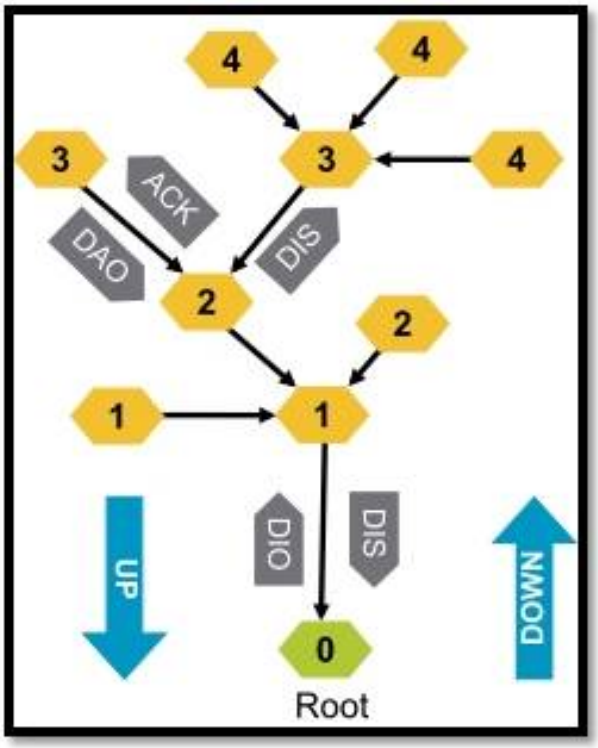


Figure 3.5: ICMPv6 Control Messages
[56]

Four RPL message can be described below.

DODAG Information Solicitation (DIS): The DIS message is mapped to 0X01 (ICMPv6 control message), and is used to solicit a DODAG Information Object (DIO) from an RPL node [11]. The DIS may be used to probe neighbor nodes in adjacent DODAGs. The current DIS message format contains non-specified flags and fields for future use.

DODAG Information Object (DIO): The DIO message [11] is mapped to 0x01 (ICMPv6 control message), and is issued by the DODAG root to construct a new DAG and then sent in multi-cast through the DODAG structure. The DIO message carries relevant network information that allows a node to discover a RPL instance, learn configuration parameters, select a DODAG parent set, and maintain the DODAG. The format of the DIO Base Object is presented in Figure 3.6.

The main DIO Base Object fields are:

- RPLInstanceID, is an 8-bit information initiated by the DODAG root that indicates the ID of the RPL instance that the DODAG is part of

- Version Number, indicates the version number of a DODAG that is typically incremented upon each network information update, and helps maintaining all nodes synchronized with new updates
- Rank, a 16-bit field that specifies the rank of the node sending the DIO message
- Destination Advertisement Trigger Sequence Number (DTSN) is an 8-bit flag that is used to maintain downward routes
- Grounded (G) is a flag indicating whether the current DODAG satisfies the application-defined objective
- Mode of Operation (MOP) identifies the mode of operation of the RPL instance set by the DODAG root. Four operation modes have been defined and differ in terms of whether they support downward routes maintenance and multi cast or not. Upward routes are supported by default. Any node joining the DODAG must be able to cope with the MOP to participate as a router, otherwise it will be admitted as a leaf node
- DODAGPreference (Prf) is a 3-bit field that specifies the preference degree of the current DODAG root as compared to other DODAG roots. It ranges from 0x00 (default value) for the least preferred degree, to 0x07 for the most preferred degree
- DODAGID is a 128-bit IPv6 address set by a DODAG root, which uniquely identifies a DODAG
- DIO Base Object may also contain an Option field.

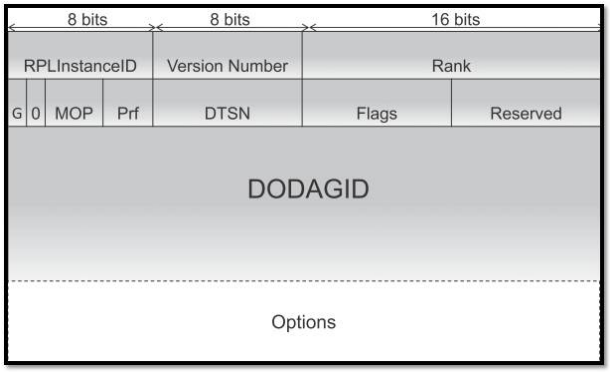


Figure 3.6: The DIO message format
[52]

*Destination Advertisement Object (DAO):*The DAO message is mapped to 0x02 (ICMPv6 control message), and is used to propagate reverse route information to

record the nodes visited along the upward path. DAO messages are sent by each node, other than the DODAG root, to populate the routing tables with prefixes of their children and to advertise their addresses and prefixes to their parents. After passing this DAO message through the path from a particular node to the DODAG root through the default DAG routes, a complete path between the DODAG root and the node is established.

As shown in Figure 3.7, the main DAO message fields are:

- RPLInstanceID, is an 8-bit information indicates the ID of the RPL instance as learned from the DIO
- K flag that indicates whether and acknowledgment is required or not in response to a DAO message
- DAO-Sequence is a sequence number incremented at each DAO message
- DODAGID is a 128-bit field set by a DODAG root which identifies a DODAG. This field is present only when flag D is set to 1

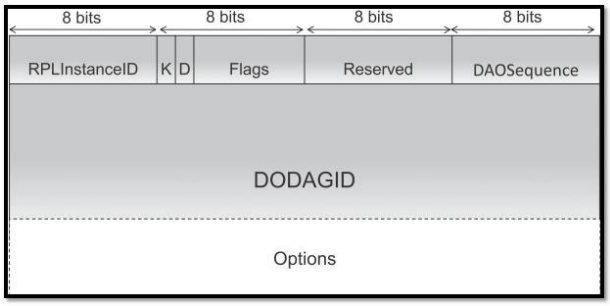


Figure 3.7: The DAO message format [52]

Destination Advertisement Object (DAO-ACK): The DAO-ACK message is sent as a unicast packet by a DAO recipient (a DAO parent or DODAG root) in response to a unicast DAO message. It carries information about RPLInstanceID, DAOSequence, and Status, which indicate the completion. Status code are still not clearly defined, but codes greater than 128 mean a rejection and that a node should select an alternate parent.

The DODAG construction is based on the Neighbor Discovery (ND) process referenced within several aspects in the papers performance [58], analyze [59], duty-cycle [60].

RPL specifies two modes of operations to maintain downward routes in an RPL instance:

Storing mode: in the storing mode, a DAO message is sent in unicast by the child to the selected parent, which is able to store DAO messages received by its children before sending the new DAO message with aggregate reachability information to its parent. The storing mode can enable or disable multi-cast mode.

Non-storing mode: in the non-storing mode, the DAO message is sent in unicast to the DODAG root, thus, intermediate parents do not store DAO messages, but only insert their own addresses to the reverse route stack in the received DAO message, then forwards it to its parent.

To maintain the DODAG, each node periodically generates DIO messages triggered by a trickle timer. The key idea of the trickle timer technique is to optimize the message transmission frequency based on network conditions. In a nutshell, the frequency is increased whenever an inconsistent network management information is received for faster recovery from a potential failure, and decreased in the opposite case. Trickle timer algorithm can be seen in Figure 3.8.

T: Timer value. T is in the range [I, I/2]
C: Redundancy counter
K: Redundancy constant
$\mathbf{I_{min}}$**:** Smallest value of I
$\mathbf{I_{doubling}}$: The number of times is doubled before maintaining a constant multicast rate.
$\mathbf{I_{max}}$: Largest value of $\mathbf{I_{max}} = \mathbf{I_{min}} * 2^{\mathbf{Idoubling}}$.
When T fires, if C>K, then send DIO, then upon expiration of I, compute new(I) and T.

Figure 3.8: Trickle Timer Algorithm [56]

3.4 Routing Metrics and Constraints

A routing metric is a quantitative value used to find the cost of a path and helps in making the routing decision in case there are different routes available. In LLN a metric is a scalar used to find the best path according to the objective function. Routing metrics are a critical component to the routing strategy. Most of the IP routing protocols such as OSPF [61] and IS-IS [62] used in traditional network use static metrics (interface bandwidth) or some static value based on interface speed for

instance. But LLN has a wide variety applications and constraints which strongly appeal for dynamic metrics.

To better understand the need of dynamic metrics and difference between a metric and constraint for LLN, let's consider the following examples.

1. An application requires a quick delivery of packets using a short path and therefore the goal will be to use ETX metric for routing.

2. An application may require encrypted communication and therefore the goal will be to avoid non-encrypted links in the path.

3. A node may be energy constrained and the objective will be to minimize energy consumption [63] by using as many connected nodes along the path as possible.

The metrics can be categorized as node metric and link metrics as stated below [64].

Node metrics: Node State Attribute (NSA), Node Energy and Hop count *Link metrics:* Throughput, Latency, Link Quality Level, ETX and Link Color

ContikiRPL [65] implements two routing metrics: hop count and ETX.

Hop Count: This metric counts the number of hops from the source to the destination. A hop count of 3 means there are 3 intermediate links between the source and destination.

ETX : ETX of a link is the expected number of transmissions required to send a packet over that link.

A constraint is used to either include or exclude links from the routing path that do not meet the criteria specified in the objective function.

A general list about routing metric and constraint can be seen in Table 3.1.

Table 3.1: Metrics and Constraints

Node Metric	Link Metric
State/Energy/Hop Count	Throughput/Latency/ETX

3.5 Objective Function

The creation of a DODAG's topology is guided so that the selected objective function is minimized without violating any of its constraints. Variables for objective functions are defined in [64] and they consists of different link and node metrics such as throughput, latency, ETX, energy (type of power source or remaining energy), and hop-count. The specification of an objective function defines how the objective function value is determined from available metrics, how this value is used to determine a nodes rank, and how it can be used to select a parent set and a preferred parent from a node's neighbors. At present, only two different objective functions have been defined: (1) OF_0 in [66] and (2) the Minimum Rank with Hysteresis Objective Function (MRHOF) in [13].

OF_0 is meant as a basic objective function that does not require any metric to be measured and it will, using default configurations, end up minimizing hop-count. MRHOF is slightly more complicated and can compute a nodes rank based on the additive metrics hop-count, latency, and ETX (default). The RPL implementation in Contiki uses MRHOF as its default objective function.

Unfortunately the concept of rank has a slightly different interpretation when viewed upon in the context of an objective function or a DODAG topology. In its most basic form, rank is a 16 bit number that is computed by the objective function. Within the DODAG topology, however, this raw rank value is divided by a number called MinHopRankIncrease [11]. The new computed value is DAGrank. The implementation of MinHopRankIncrease ensures that rank will increase for every step downwards in the DODAG. Using the default value of 256 for MinHopRankIncrease.

Every root node is required to have a rank of MinHopRankIncrease which will, using the definition above, give them a DAGrank of 1. Similarly, every node beneath the root node will have DAGrank larger than 1. The rank and DAGrank of 0 is reserved for situations where an LLN has several border routers. In these cases, the coordination between different

An objective function defines how a RPL node selects and optimizes routes within a RPL instance based on the information objects available. Consider a physical network made of several links with different qualities such as throughput, Latency and nodes with different qualities such as battery operated, mains-powered. If the network carries

different types of traffic it might be useful to carry the traffic based on different objective functions which are optimizing different metrics or fulfilling constraints. Thus an objective function is used to steer the traffic to different paths according to the requirements. These requirements are actually encoded in a programming logic what we call objective function and used by RPL during routing operations which is explained next.

ContikiRPL implements two objective functions: OF_0 and ETX. OF_0 uses hop count as routing metric where as ETX uses ETX metric as a routing metric for selecting the best path.

This separation of objective functions from the core protocol specification allows RPL to be adopted to meet the different optimization criteria required for a wide range of deployments, applications and network designs.

3.6 Minimum Rank with Hysteresis Objective Function

The Minimum Rank with Hysteresis Objective Function, MRHOF [11] [13], is designed to find the paths with the smallest path cost while preventing excessive churn in the network. It does so by using two mechanisms. First, it finds the minimum cost path, i.e., path with the minimum Rank. Second, it switches to that minimum Rank path only if it is shorter (in terms of path cost) than the current path by at least a given threshold. This second mechanism is called "hysteresis" [13].

Table 3.2: Conversion of Metric to Rank

Node/Link Metric	Rank
Hop Count	Cost
Latency	Cost/65536
ETX	Cost

MRHOF may be used with any additive metric listed in [64] as long as the routing objective is to minimize the given routing metric. Nodes must support at least one of these metrics: hop count, latency, or ETX. Nodes should support the ETX metric. MRHOF does not support non-additive metrics. An example for MRHOF can be seen in Figure 3.9.

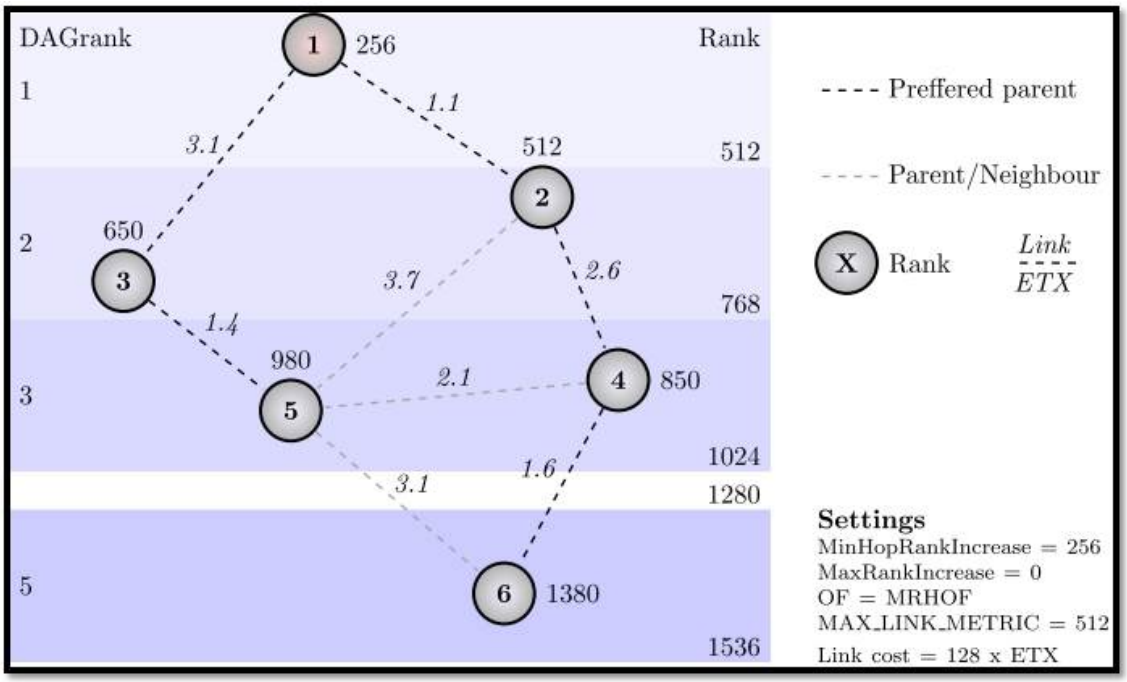


Figure 3.9: DODAG with ranks determined from ETX using the MRHOF.

4. PARENT-AWARE OBJECTIVE FUNCTION (PAOF)

In this chapter, firstly the general overview is described. In the second section algorithm of new proposed OF technique is introduced and in the last section implementation details are talked.

4.1 General Overview

RPL identifies the best paths to route the data through the network according to the objective function and a set of metrics as described in the previous section. These metrics can be either node attributes, such as hop-count, node remaining energy; or link attributes, such as link quality, latency, and ETX.

Among these metrics, the ETX is widely used to design reliable routing protocols for WSNs since it reflects the quality of the paths used to transmit data. In addition to this, hop count, energy level are also used metrics/constraints. However, none of the existing objective functions consider parent count as a node metric. Based on RPL, the routes are used intermediate nodes towards sink node to carry generated data. Due to this, some nodes are identified as preferred parent so they are responsible to transmit the payload from the leaf nodes.

One of the important critical aspects in IoT network is load balancing. As the number of nodes and data traffic increase in the network topology day by day, it is worth to manage the network resources such as distributed work load, longer life time. efficiently. There are also many academic research papers and books related to load distribution [67], balance [68] and imbalance areas [69] for IoT and WSN networks.

In order to achieve load balancing, in this thesis we propose a new method called Parent Aware Objective Function (PAOF) for RPL. This method is planned to work based on RPL requirements and specification. Regarding different network topologies, the

proposed PAOF technique ensures a better load balanced network, diversity of parent selection [70] and reduce the end-to-end delay.

4.2 Algorithm

The proposed technique PAOF combines the ETX value as link metric and parent count as node metric (the number of candidate parents) and then compute the cost of the path towards the sink. In this objective function, the ETX value is still the key point being considered. Regarding to this, we use MinHopRankIncrease parameter defined in RPL Control message DIO [11] as a reference point.

Proposed technique considers the parent count metric only if delta between two candidate preferred parents ETX is smaller than MinHopRankIncrease value. If so, the algorithm compares the number of parent counts and selects the minimum one. Hence, we are able to utilize more sensor nodes as preferred parent, the data traffic is shared by more nodes in the network. Flowchart of the algorithm can be seen in Figure 4.1.

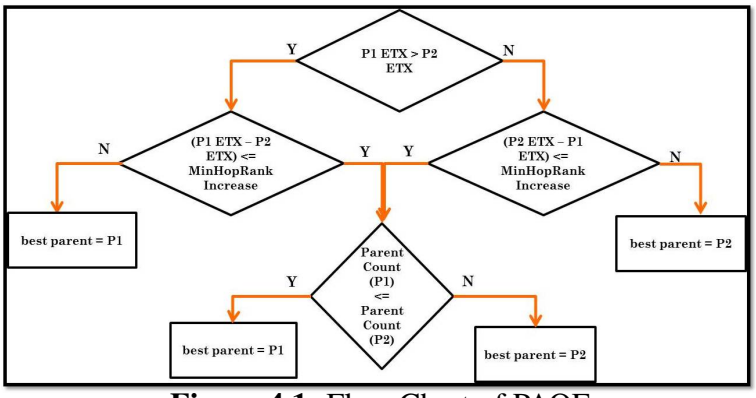


Figure 4.1: Flow Chart of PAOF

4.3 Implementation

The algorithm is implemented into ContikiRPL [71] source code via extending existent source code files designed for MRHOF [72]. RPL related source code files are investigated and reviewed in detailed. MRHOF algorithm is implemented in rpl-mrhof.c code file [72].

Key types can be seen in the following list.

- rplparent

- rpldag
- rplof
- rplinstance
- CalculateRank
- BestDag
- BestParent
- UpdateMetricContainer

From the above list, mainly parents field of rpldag type is used for PAOF. Existing code snippet belonging for this type can be seen in Figure 4.2.

```
/* Directed Acyclic Graph */
struct rpl_dag {
  uip_ipaddr_t dag_id;
  rpl_rank_t min_rank; /* should be reset per DAG iteration! */
  uint8_t version;
  uint8_t grounded;
  uint8_t preference;
  uint8_t used;
  /* live data for the DAG */
  uint8_t joined;
  rpl_parent_t *preferred_parent;
  rpl_rank_t rank;
  struct rpl_instance *instance;
  LIST_STRUCT(parents);
  rpl_prefix_t prefix_info;
};
```

Figure 4.2: Definition of rpldag type

Key functions/procedures can be seen in the following list.

- UpdateMetricContainer
- CalculatePathMetric
- Reset
- NeighborLinkCallback
- CalculateRank
- BestDag
- BestParent
- UpdateMetricContainer

From the above list, mainly bestparent function is redesigned and touched for PAOF. Existing code snippet running for MRHOF and PAOF can be found in Figures 4.3 and 4.4.

best_parent_of1 - /home/user/tez1/code/

File Edit Search Preferences Shell Macro Windows Help

```
{
  rpl_dag_t *dag;
  rpl_path_metric_t min_diff;
  rpl_path_metric_t p1_metric;
  rpl_path_metric_t p2_metric;

  dag = p1->dag; /* Both parents are in the same DAG. */

  min_diff = RPL_DAG_MC_ETX_DIVISOR /
             PARENT_SWITCH_THRESHOLD_DIV;

  p1_metric = calculate_path_metric(p1);
  p2_metric = calculate_path_metric(p2);

  /* Maintain stability of the preferred parent in case of similar ranks. */
  if(p1 == dag->preferred_parent || p2 == dag->preferred_parent) {
    if(p1_metric < p2_metric + min_diff &&
       p1_metric > p2_metric - min_diff) {
      PRINTF("RPL: MRHOF hysteresis: %u <= %u <= %u\n",
             p2_metric - min_diff,
             p1_metric,
             p2_metric + min_diff);
      return dag->preferred_parent;
    }
  }

  return p1_metric < p2_metric ? p1 : p2;
}
```

Figure 4.3: Implementation of BestParent Function for MRHOF

best_parent_of2 - /home/user/tez1/code/

File Edit Search Preferences Shell Macro Windows

```
static rpl_parent_t *
best_parent(rpl_parent_t *p1, rpl_parent_t *p2)
{
  rpl_dag_t *dag;
  rpl_path_metric_t min_diff;
  rpl_path_metric_t p1_metric;
  rpl_path_metric_t p2_metric;

  rpl_path_metric_t p1_1_metric;
  rpl_path_metric_t p2_2_metric;
    rpl_path_metric_t delta;

  dag = p1->dag; /* Both parents are in the same DAG. */

  min_diff = RPL_DAG_MC_ETX_DIVISOR /
             PARENT_SWITCH_THRESHOLD_DIV;

  p1_metric = calculate_path_metric(p1);
  p2_metric = calculate_path_metric(p2);

  if (p1_metric >= p2_metric)
  {
    delta = p1_metric - p2_metric;

    if (delta <= 256)
    {
     p1_1_metric = RPL_PARENT_COUNT(p1->dag); /* OF2 */
     p2_2_metric = RPL_PARENT_COUNT(p2->dag); /* OF2 */

     if (p1_1_metric == p2_2_metric)
     {
      return p2; /* OF 2 */
     }
     return p1_1_metric < p2_2_metric ? p1 : p2;  /*   OF2 */
    }

  }
  else
  {
    delta = p2_metric - p1_metric;

    if (delta <= 256)
    {
     p1_1_metric = RPL_PARENT_COUNT(p1->dag); /* OF2 */
     p2_2_metric = RPL_PARENT_COUNT(p2->dag); /* OF2 */

     if (p1_1_metric == p2_2_metric)
     {
       return p1; /* OF 2 */
     }
     return p1_1_metric < p2_2_metric ? p1 : p2;  /*   OF2 */
    }

  }
```

Figure 4.4: Implementation of BestParent Function for PAOF

The implemented algorithm is evaluated for a simple DODAG graph scenario and the DODAG is constructed differently for MRHOF and PAOF. Graph results can be seen in Figures 4.5 and 4.6.

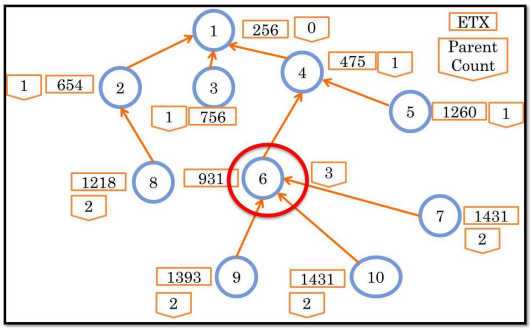


Figure 4.5: DODAG Construction for MRHOF

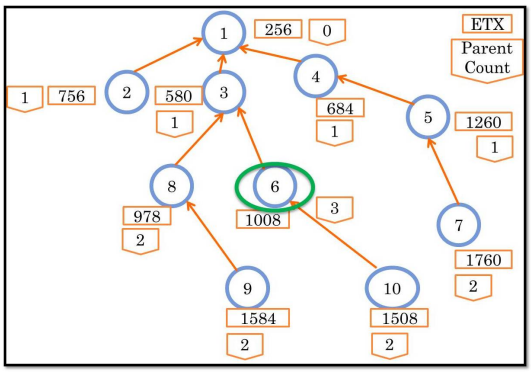


Figure 4.6: DODAG Construction for PAOF

It can be seen that more nodes are selected as preferred parent node in PAOF technique. This ensures that load density is better in PAOF (6/10) than MRHOF (4/10). Details about simulation results will be described in the next section.

Table 4.1: DODAG Construction Performance

	Number of Preferred Parent Nodes
MRHOF	4
PAOF	6

5. PERFORMANCE EVALUATION

In this chapter, firstly Contiki OS is introduced. After then performance metrics are described and simulation environment is identified. Lastly, measurements and results are evaluated.

5.1 Simulation Environment: Contiki

Contiki [73] is a wireless sensor network operating system and consists of the kernel, libraries, the program loader and a set of processes. It is used in networked embedded systems and smart objects [74].

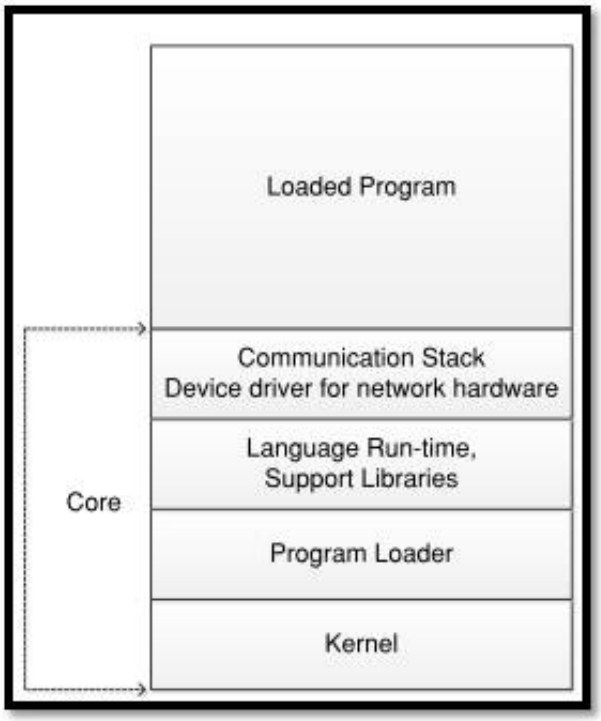


Figure 5.1: The Contiki System
[26]

Contiki provides mechanisms that assist in programming the smart object applications. It provides libraries for memory allocation, linked list manipulation and communication abstractions. It is the first operating system that provided IP communication. It is developed in C, all its applications are also developed in C programming language, and therefore it is highly portable to different architectures like Texas Instruments MSP430.

Contiki is an event-driven system in which processes are implemented as event handlers that run to completion. A Contiki system is partitioned into two parts: the core and the loaded programs. The core consists of the Contiki kernel, the program loader, the language run-time, and a communication stack with device drivers for the communication hardware.

The Program loader loads the programs into the memory and it can either obtain it from a host using communication stack or can obtain from the attached storage device such as EEPROM.

The Contiki operating system provides modules for different tasks (layers). It provides the routing modules in a separate directory "contiki/core/net/rpl" [75] and consists of a number of files. These files are separated logically based on the functionalities they provide for instance rpl-dag.c [75] contains the functionality for Directed Acyclic Graph (DAG) formation, rpl-icmp6.c [75] provides functionality for packaging ICMP messages.

5.1.1 System Overview

Contiki provides a wide range of features not necessarily expected in such a low footprint operating system, such as an interactive shell, a web browser and a flash-based file system.

More importantly, it provides two communication stacks: uIP [76] and Rime [77]. uIP is a small RFC-compliant TCP/IP stack that makes it possible for Contiki to communicate over the Internet. Rime is a lightweight communication stack designed for low-power radios. Rime provides a wide range of communication primitives, from best-effort local area broadcast, to reliable multi-hop bulk data flooding. General protocol stack can be seen in Figure 5.2.

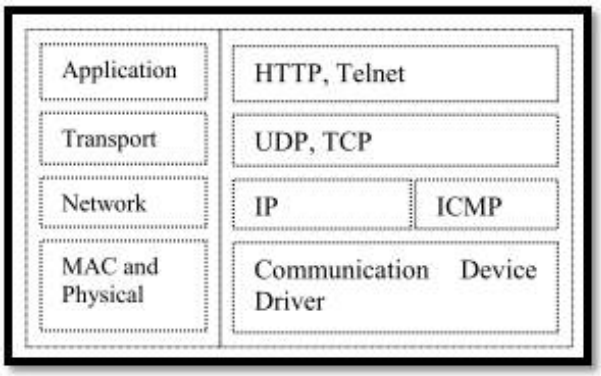


Figure 5.2: Contiki Protocol Stack
[23]

The Rime communication stack provides a set of basic communication primitives ranging from best-effort single-hop broadcast and best-effort single-hop unicast to best-effort network flooding and hop-by-hop reliable multi-hop unicast. It has been designed to map onto typical sensor network protocols: data dissemination, data collection, and mesh routing.

The major components of the Rime protocol stack are shown in below figure. The Rime stack builds on top of the physical layer and the MAC layer. The physical layer is handled by the radio driver. The MAC layer is a sublayer of the data link layer, and a common requirement for any shared medium communication.

Duty Cycling evaluated also in [78] is the technique of keeping the radio off as much as possible and switch it on only when needed. The WSN devices are small and operate with very small batteries [79] that provide power for only a very limited time. However, duty cycling can significantly reduce energy consumption.

Contiki has three duty cycling mechanisms: ContikiMAC [80], X-MAC [81] and LPP. ContikiMAC is a protocol based on the principles behind low-power listening but with better power efficiency. During these thesis, The ContikiMAC radio duty cycling mechanism is used.

5.1.2 ContikiRPL

ContikiRPL implements the RPL protocol, as specified in version 18 of the RPL specification [11], and two objective functions OF_0 and the Minimum Rank Objective Function with Hysteresis (MRHOF). ContikiRPL has been successfully tested for interoperability through the IPSO Alliance program, where it was used on three

different platforms and ran over two different link layers, IEEE 802.15.4 and the Watteco low-power power-line communication module.

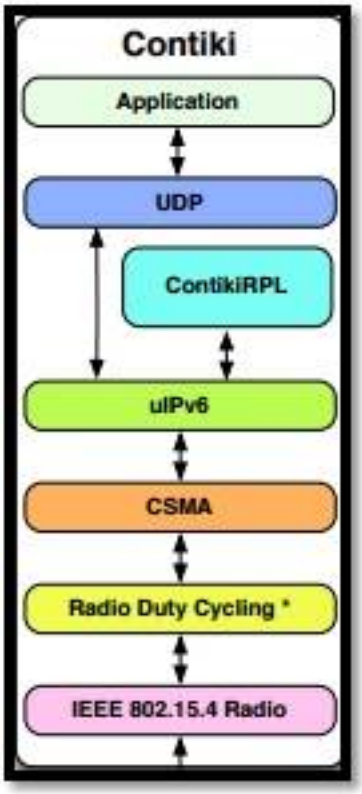


Figure 5.3: ContikiRPL in protocol stack
[82]

ContikiRPL implements two routing metrics, hop count and ETX. ContikiRPL includes two objective functions i.e. OF_0 and ETX. OF_0 uses hop count as routing metric whereas ETX uses ETX metric as a routing metric for selecting the best path.

When the RPL network starts, the root of the DAG starts sending out the DIO messages to let the neighbors know the parameters of the network like DAG-ID, objective function, routing metric, rank etc. as shown in Figure 5.4.

153	4.361000	fe80::212:7401:1:101	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)
154	4.585000	fe80::212:7401:1:101	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)
155	4.595000	fe80::212:7401:1:101	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)
156	4.596000	fe80::212:7401:1:101	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)
157	4.598000	fe80::212:7401:1:101	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)
158	4.600000	fe80::212:7401:1:101	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)
159	4.601000	fe80::212:7401:1:101	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)
160	4.604000	fe80::212:7401:1:101	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)
161	4.606000	fe80::212:7401:1:101	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)

Figure 5.4: DIO sent by root after RPL starts
[23]

5.1.3 Cooja

Cooja [83] is a Java-based simulator designed for simulating sensor networks running the Contiki sensor network operating system. The simulator is implemented in Java but allows sensor node software to be written in C.

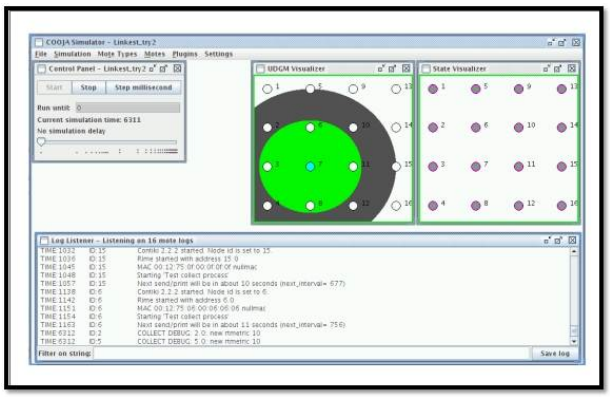


Figure 5.5: Cooja, the network simulator that comes with Contiki [84]

One of the differentiating features is that Cooja allows for simultaneous simulations at three different levels: Network Level, Operating System Level and Machine code instruction level. Cooja can also run Contiki programs either compiled natively on the host CPU or compiled for MSP430 emulator [85].

In Cooja all the interactions with the simulated nodes are performed via plug-ins like Simulation Visualizer, Time line, and Radio logger. It stores the simulation in an xml file with extension 'csc' (Cooja simulation configuration). This file contains information about the simulation environment, plug ins, the nodes and its positions, random seed and radio medium.

Cooja Simulator runs the Contiki applications whose files are placed in another directory and may also contain a "project-conf.h" file which provides the ability to change RPL parameters in one place.

5.2 Performance Metrics

In order to make a qualitative analysis and compare the results obtained for MRHOF and PAOF, the following metrics were employed:

Average Parent Load Density: In this metric the aim is to compute the average load density [86] on all selected preferred parents. This value is computed shown below.

$$\sum Delivered\ Successful\ Packets / \sum Preferred\ Parents$$

Average Packet Delay: To measure the delay between times the packet generated and reached to the sink node. This value is computed as shown below.

$$\sum((Packet\,Arrival\,Time) - (Packet\,Generation\,Time))/\sum Reached\,Packets$$

Number of DIO Messages: As the proposed technique interacts on directly DIO message generation, it is useful to compare the number of generated DIO messages to have an idea from the point of introduced overhead.

Number of DAO Messages: As our proposed technique aims to increase the number of preferred parents, the number of these control messages shows us if there is an overhead in the total messaging.

Parent Diversity: This metric shows us how many different nodes can be selected as preferred parent in a network topology. This value is computed as shown below.

$$\sum Preferred\,Parents/\sum Nodes$$

5.3 Simulation and Network Setup

Table 5.1 gives a detailed description of the network and the simulation in term of the parameters employed in Cooja and Contiki.

Table 5.1: Simulation network parameters in Cooja

Parameter Name	Parameter Value
Number of Nodes	1 sink and 24 sensors
Radio Range	50m
Network Layer	IPv6 with 6LoWPAN
Transport Layer	UDP
Routing Protocol	RPL
Channel Check Rate	8
RPL Mode	Storing Mode
Network Setup Time	60 seconds
Simulation Time	960 seconds

In this work, we employed mainly on three network topologies which are given in following figures in order to study the performance of the introduced method. In all of these three topologies, there exist 1 sink node and 24 sensor nodes. Sink node is placed at the center in the first topology while it is located at the middle top position

in the other two topologies. Prior to selection of these three network topology, many topology types are experienced. Detail can be seen in Appendix section.

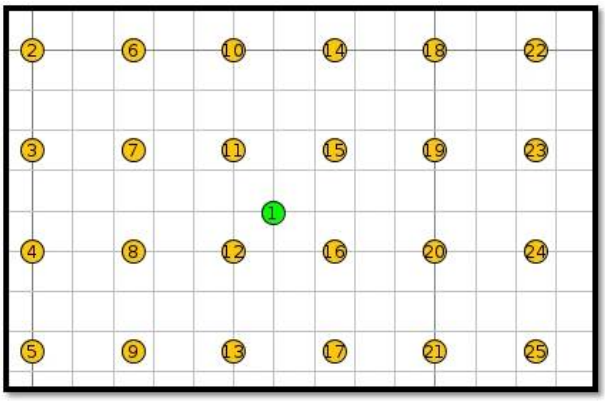


Figure 5.6: Topology 1

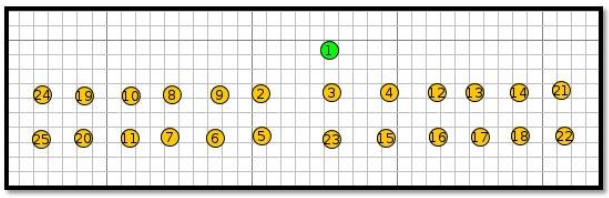


Figure 5.7: Topology 2

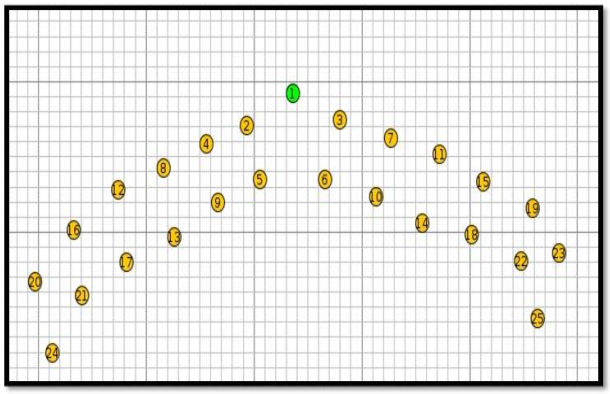


Figure 5.8: Topology 3

In the traffic scenario, each node generated a payload data having the length of 30 bytes at the time intervals determined by Negative Exponential distribution [87], given with four different values of λ 0.2;0.5;0.7;0.9, Pareto distribution with α 5 β 1 [88].

Probability density function of Negative Exponential Distribution can be seen below.

$$\left[f(x;\lambda) = \begin{cases} \lambda e^{(-\lambda x)} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \right]$$

Probability density function of Pareto Distribution can be seen below.

$$\left[f(x) = \begin{cases} \frac{\alpha \beta^{\alpha}}{x^{\alpha+1}} & \text{if } x \geq \beta \\ 0 & \text{if } x < \beta \end{cases} \right]$$

Data traffic started out after 60 seconds. The first minute is left for the RPL control messages DIO, DAO and DIS traffic to be able to setup a stable DODAG graph. After than each node generated 20 packets for each traffic scenario.

Negative Exponential distribution simulates Query-driven applications where the sensing nodes trigger the sending after the event detection.

Pareto distribution simulates Continuous sensing applications (Time-driven) where some critical applications require continuous sending of sensing values. This distribution also generates some bursty traffic.

Simulation were run three times for each topology and traffic scenario and the results show the average of them. Confidence interval is 95%.

5.4 Results

Preferred parent selection results can be seen in Tables 5.2, 5.3 and 5.4.

Table 5.2: Preferred Parent Selection for Topology 1

OF_1		OF_2	
Node No	Preferred Parent	Node No	Preferred Parent
2	10	2	7
3	8	3	11
4	8	4	8
5	8	5	8
6	14	6	8
7	1	7	1
8	1	8	1
9	1	9	1
10	1	10	1
11	1	11	1
12	1	12	1
13	1	13	1
14	1	14	1
15	1	15	1
16	1	16	1
17	1	17	1
18	14	18	19
19	1	19	1
20	1	20	1
21	17	21	13
22	14	22	19
23	19	23	19
24	19	24	19
25	17	25	17
Number of Preferred Unique Parents: 6		Number of Preferred Unique Parents: 7	

In Table 5.2, it can be seen that number of preferred count in OF_2 (7) method is higher than OF_1 (6).

Table 5.3: Preferred Parent Selection for Topology 2

OF_1		OF_2	
Node No	Preferred Parent	Node No	Preferred Parent
2	3	2	3
3	1	3	1
4	3	4	3
5	2	5	23
6	2	6	2
7	6	7	9
8	6	8	9
9	2	9	2
10	8	10	8
11	8	11	8
12	4	12	15
13	12	13	12
14	17	14	13
15	3	15	3
16	4	16	15
17	12	17	16
18	17	18	13
19	10	19	11
20	10	20	11
21	18	21	18
22	18	22	14
23	3	23	3
24	19	24	19
25	19	25	19
Number of Preferred Unique Parents: 11		Number of Preferred Unique Parents: 14	

In Table 5.3, preferred parent count in OF_2 is three plus OF_1. Let's make a deep dive analysis for node 21 and 22 from the point of preferred patent selection. We can see that both of node 21 and 22 select node 18 as preferred parent. However, in OF_2, node 22 selects node 14 while node 21 selects node 18. As the difference in ETX values is getting smaller between nodes, their selection probability increases if their parent count is lower.

Table 5.4: Preferred Parent Selection for Topology 3

OF_1		OF_2	
Node No	Preferred Parent	Node No	Preferred Parent
2	1	2	1
3	1	3	1
4	2	4	2
5	2	5	2
6	3	6	3
7	3	7	3
8	4	8	4
9	4	9	5
10	6	10	6
11	7	11	7
12	8	12	8
13	8	13	9
14	10	14	10
15	11	15	11
16	12	16	12
17	12	17	13
18	14	18	14
19	15	19	15
20	16	20	16
21	17	21	17
22	18	22	18
23	19	23	19
24	21	24	21
25	22	25	22
Number of Preferred Unique Parents: 18		Number of Preferred Unique Parents: 21	

In Table 5.4, preferred parent count result and behavior seem similar to Table 5.3.

Results for overall Negative Exponential distributed traffic scenario can be seen below.

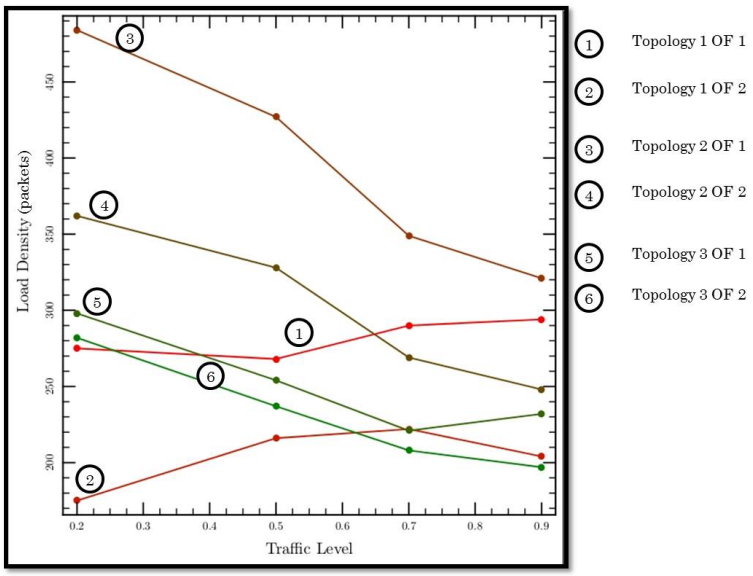


Figure 5.9: Average Parent Load Density for Negative Exponential

In Figure 5.9, average parent load density in OF_2 is better than OF_1 for all of the traffic level points. This is due to preferred count in OF_2 is always higher than OF_1. Traffic level in here means inter arrival probabilities calculated by Negative Exponential function distribution. For example if traffic level is 0.2, this means that inter arrival value is on the average 5 seconds. While λ value is increased, inter arrival time is getting smaller.

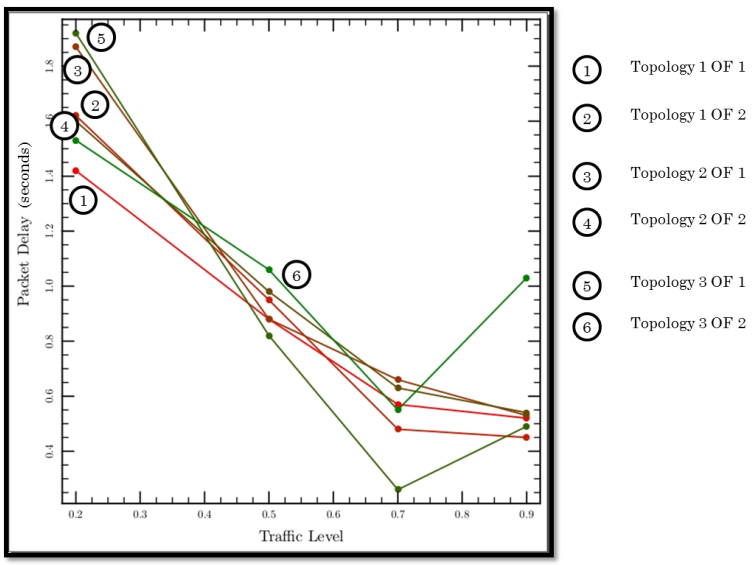


Figure 5.10: Average Packet Delay for Negative Exponential

In Figure 5.10, we can see that average packet delay is not too much different for OF_1 and OF_2 in general.

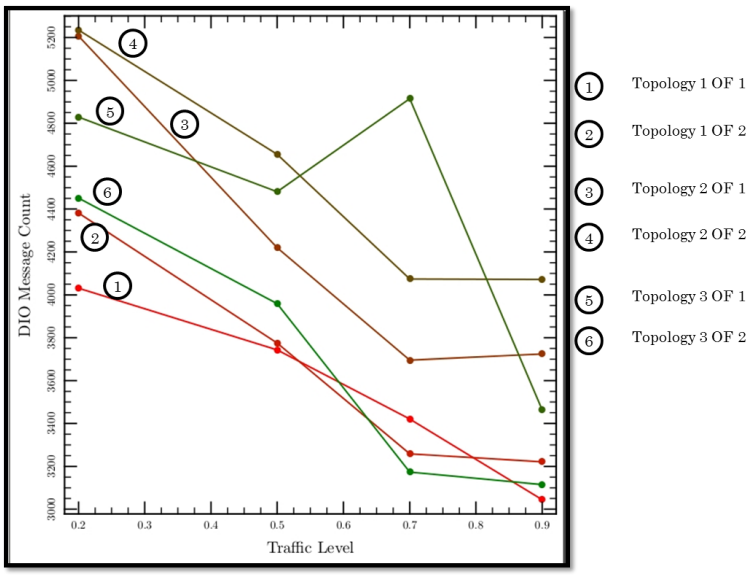


Figure 5.11: Number of DIO Message for Negative Exponential

In Figure 5.11, we can see that number of DIO message is in better position for OF_1 than OF_2 in general. Specially for Topology 3, OF_2 performs better than OF_2. Also DIO message generation acts in decreasing manner while traffic level changes. This is due to staying less time in active traffic.

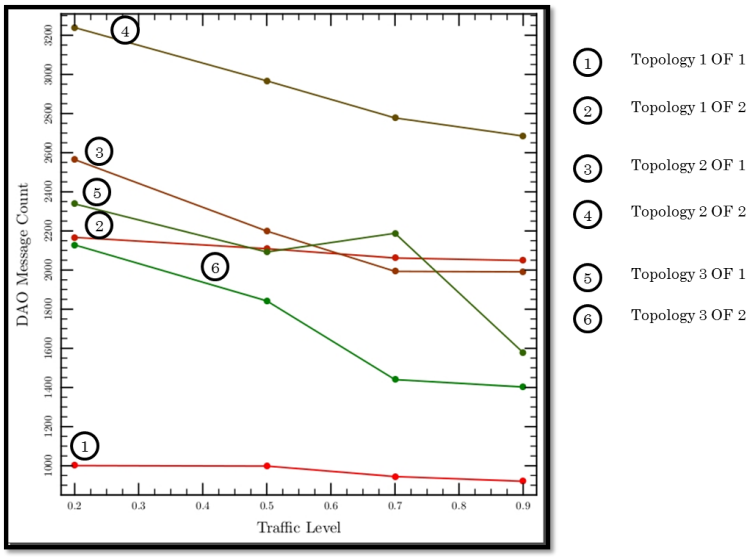


Figure 5.12: Number of DAO Messages for Negative Exponential

In Figure 5.12, we can see that number of DAO message is in better position for OF_1 than OF_2 in general. Specially for Topology 3, OF_2 performs better than OF_2. This is due to increasing number of preferred count which means adding a new DAO message sender into the DODAG. Also DAO message generation acts in decreasing manner while traffic level changes. This is due to staying less time in active traffic.

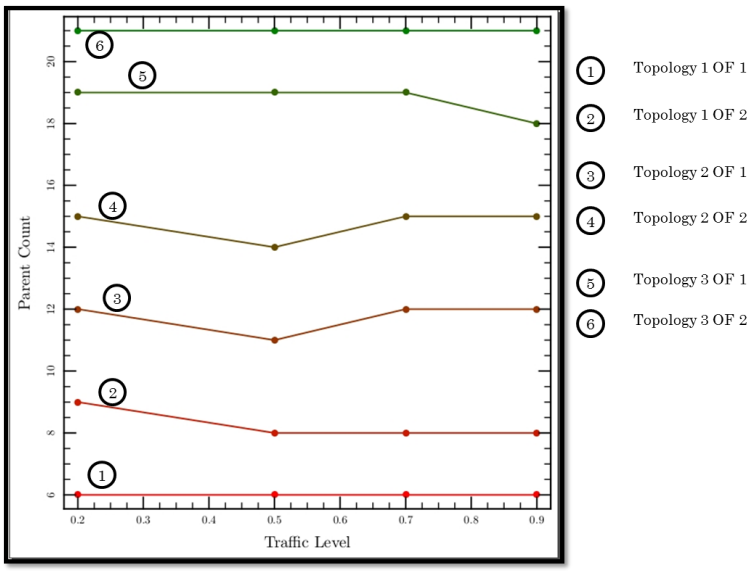


Figure 5.13: Parent Diversity for Negative Exponential

In Figure 5.13, parent diversity in OF_2 is better than OF_1 for all of the traffic level points and topologies. Furthermore, traffic level change seems not affecting this metric. This is due to constructing majority of graph at the beginning.

Results for Pareto traffic scenario:

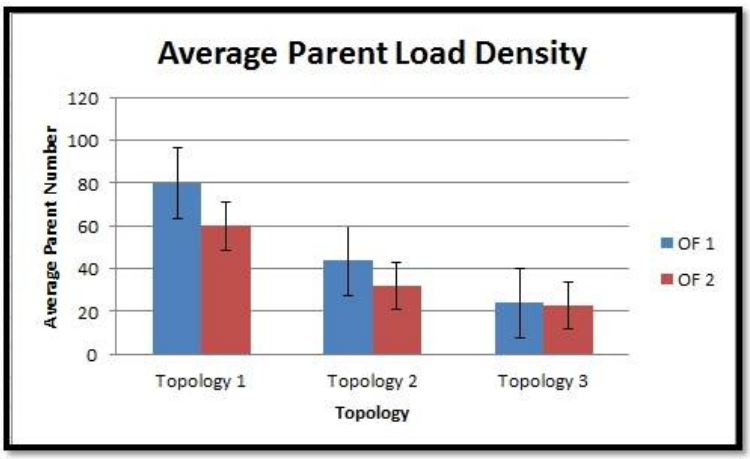


Figure 5.14: Average Parent Load Density for Pareto

In Figure 5.14, average parent load density in OF_2 is better than OF_1 for all of the topologies. This is due to preferred count in OF_2 is always higher than OF_1.

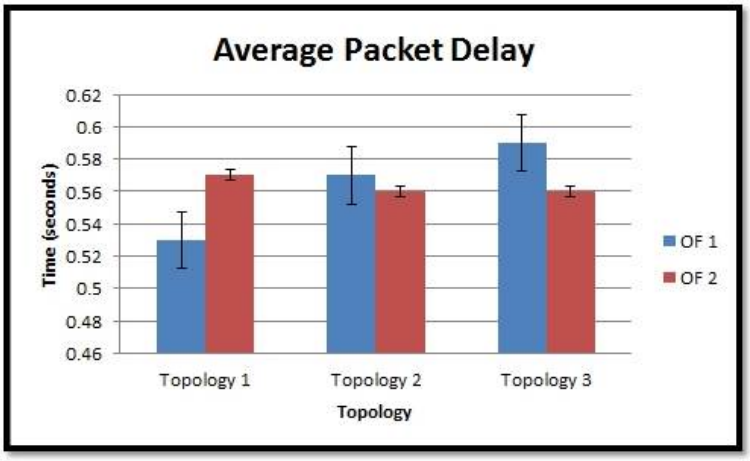


Figure 5.15: Average Packet Delay for Pareto

In Figure 5.15, average packet delay in OF_2 is better than OF_1 for Topology 2 and 3.

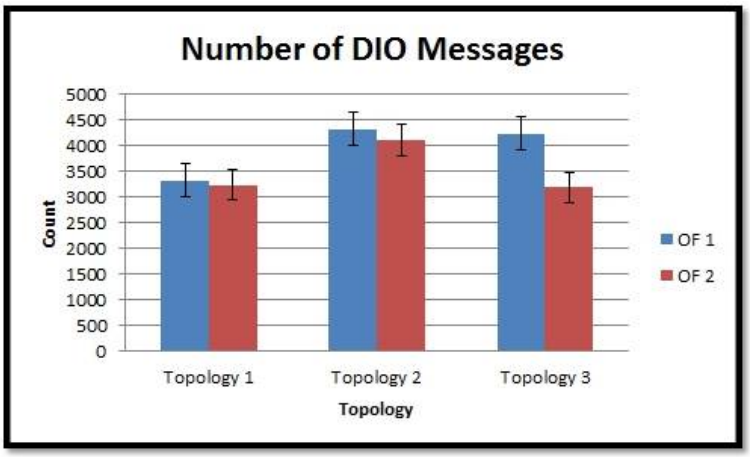


Figure 5.16: Number of DIO Message for Pareto

In Figure 5.16, number of DIO message is similar range for OF_1 and OF_2.

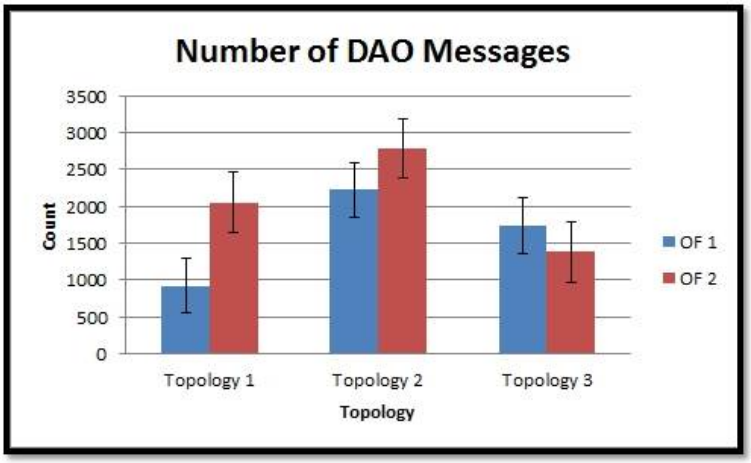


Figure 5.17: Number of DAO Messages for Pareto

In Figure 5.17, number of DAO message is better in OF_1 than OF_2. This is due to increasing number of preferred count which means adding a new DAO message sender into the DODAG.

Here we would like to give some general comments regarding the results obtained.

We got lower parent load density and higher parent diversity because we increased the possibility of some intermediate nodes to be selected as preferred parent with the proposed PAOF solution. Therefore, more nodes are tagged as preferred parent in the network where all of them are used during carrying of the data towards the sink node.

In addition to, average packet delay shows better result in general due to distributing load and reducing collision and congestion in the intermediate nodes.

Number of DIO and DAO messages is higher in our solution because we give chance to more nodes to be considered as preferred parent when the nodes have similar ETX values. This results to generate more DAO messages as the preferred parent node count increases. However, this is not a big deal because DIO and DAO message generation mostly are populated at the initial phase of the network topology setup. Based on Trickle timer, time interval of DIO/DAO message generation is expanded so the network becomes sane and stable.

From the point of topology classification, we can say that PAOF works better in Topology 2 and Topology 3. Topology 1 is not a good environment for PAOF because the sink is in the center position and nodes are very close to each other. PAOF performs better in such an IoT network where

- sink node is located at the corner
- intermediate nodes are distributed in mid-level sparse manner
- topology is equivalent to tree

From the point of traffic classification, we can say that PAOF works better under bursty traffic (Pareto Distribution) as the collision probability is high.

The main drawback of our proposed routing technique is that it leads more DIO and especially DAO message generation. However, as DIO transmission is governed by Trickle Timer [11], DIO/DAO message generation will be still under control and stable in the latter phase of the traffic.

6. CONCLUSIONS AND FUTURE WORKS

In this thesis, we introduce a new objective function, calling **PAOF** for RPL to be used in IoT networks. This new objective function works based on link metric **ETX** and node metric **parent count**. PAOF aims to distribute the workload balanced thorough the intermediate nodes. Parent count of a node gives an idea about messaging load and interference possibility. While ETX value of a node is in a good shape, it may fall in congestion and suffer from battery if it is chosen as preferred parent by the bottom level child nodes. Within PAOF, we give a chance to select another node having similar range of ETX value.

We compare the proposed technique with MRHOF using Cooja simulation environment with Contiki OS running ContikiRPL implementation. MRHOF is default objective function in ContikiRPL and decides using the ETX value only.

From the simulation results, we can say that proposed routing technique PAOF frequently performs better than MRHOF. The best results for PAOF are observed such in medium sparse topology, where the sink node is located in the corner.

The results obtained show that PAOF makes significant improvements in parent load density, parent diversity and end-to-end delay compared to MRHOF using ETX. PAOF ensures that the network will become load balanced, hence, have longer network lifetime and behave tolerantly in case of congestion.

The future work on the proposed technique will be to focus on the effectiveness of different type of IoT applications generating **hybrid** traffic. Also P2P and P2MP traffic flows can be evaluated.

In this work, as a new approach, parent count is considered as a Layer 3 metric. Similarly, child count can also be considered as a new candidate node metric. This may need to define a **cross layer** algorithm including Layer 2 and Layer 3.

As another future work, a hybrid PAOF MRHOF approach can be studied as an **Adaptive Objective Function**. This new mechanism will behave **proactively** and manage routing according to the network topology dynamically.

REFERENCES

[1] **Bhende, M.**, **Wagh, S. and Utpat, A.** (2014). A Quick Survey on Wireless Sensor Networks, *Communication Systems and Network Technologies (CSNT), 2014 Fourth International Conference on*, pp.160–167.

[2] **Abedin, S.**, **Alam, M.**, **Haw, R. and Hong, C.S.** (2015). A system model for energy efficient green-IoT network, *Information Networking (ICOIN), 2015 International Conference on*, pp.177–182.

[3] **Akyildiz, I.**, **Melodia, T. and Chowdury, K.** (2007). Wireless multimedia sensor networks: A survey, *Wireless Communications, IEEE*, ***14***(6), 32–39.

[4] **Milinkovic, A.**, **Milinkovic, S. and Lazic, L.** (2014). Some experiences in building IoT platform, *Telecommunications Forum Telfor (TELFOR), 2014 22nd*, pp.1138–1141.

[5] **Chen, S.**, **Xu, H.**, **Liu, D.**, **Hu, B. and Wang, H.** (2014). A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective, *Internet of Things Journal, IEEE*, ***1***(4), 349–359.

[6] **Wetteroth, D.** (2001). *OSI Reference Model for Telecommunications*, McGraw-Hill Professional; 1st edition.

[7] **Kovatsch, M.** (2011). Demo abstract: Human-CoAP interaction with Copper, *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*, pp.1–2.

[8] **Gupta, V. and Jayaraghavendran** (2015). Invited Talk: IoT Protocols War and the Way Forward, *VLSI Design (VLSID), 2015 28th International Conference on*, pp.28–28.

[9] `http://secondnexus.com/technology-and-innovation/the-internet-of-things-is-creating-a-smarter-world/`, [Online; accessed 13-April-2015].

[10] `https://tools.ietf.org/html/rfc4944`, [Online; accessed 13-April-2015].

[11] `https://tools.ietf.org/html/rfc6550`, [Online; accessed 13-April-2015].

[12] **Gaddour, O. and Koubâa, A.** (2012). {RPL} in a nutshell: A survey, *Computer Networks*, ***56***(14), 3163 – 3178, `http://www.sciencedirect.com/science/article/pii/S1389128612002423`.

[13] `https://tools.ietf.org/html/rfc6719`, [Online; accessed 13-April-2015].

[14] **Javaid, N.**, **Javaid, A.**, **Khan, I. and Djouani, K.** (2009). Performance study of ETX based wireless routing metrics, *Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on*, pp.1–7.

[15] **Vermesan, O. and Friess, P.** (2014). *Internet of Things Applications - From Research and Innovation to Market Deployment*, River Publishers.

[16] **Sheng, Z.**, **Yang, S.**, **Yu, Y.**, **Vasilakos, A.**, **McCann, J. and Leung, K.** (2013). A survey on the ietf protocol suite for the internet of things: standards, challenges, and opportunities, *Wireless Communications, IEEE*, ***20***(6), 91–98.

[17] **Colistra, G.**, **Pilloni, V. and Atzori, L.** (2014). Objects that agree on task frequency in the IoT: A lifetime-oriented consensus based approach, *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pp.383–387.

[18] **Kafi, M.**, **Djenouri, D.**, **Ben-Othman, J. and Badache, N.** (2014). Congestion Control Protocols in Wireless Sensor Networks: A Survey, *Communications Surveys Tutorials, IEEE*, ***16***(3), 1369–1390.

[19] **Cenedese, A.**, **Zanella, A.**, **Vangelista, L. and Zorzi, M.** (2014). Padova Smart City: An urban Internet of Things experimentation, *A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on*, pp.1–6.

[20] **Xu, L.D.**, **He, W. and Li, S.** (2014). Internet of Things in Industries: A Survey, *Industrial Informatics, IEEE Transactions on*, ***10***(4), 2233–2243.

[21] **Yi, J.**, **Clausen, T. and Igarashi, Y.** (2013). Evaluation of routing protocol for low power and Lossy Networks: LOADng and RPL, *Wireless Sensor (ICWISE), 2013 IEEE Conference on*, pp.19–24.

[22] **Sehgal, V.**, **Patrick, A. and Rajpoot, L.** (2014). A comparative study of cyber physical cloud, cloud of sensors and internet of things: Their ideology, similarities and differences, *Advance Computing Conference (IACC), 2014 IEEE International*, pp.708–716.

[23] **Ali, H.** (2012). A Performance Evaluation of RPL in Contiki, *Ph.D. thesis*, SICS, Stockholm Sweden.

[24] **Accettura, N.**, **Palattella, M.**, **Boggia, G.**, **Grieco, L. and Dohler, M.** (2013). Decentralized Traffic Aware Scheduling for multi-hop Low power Lossy Networks in the Internet of Things, *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, pp.1–6.

[25] **Gutierrez, J.**, **Naeve, M.**, **Callaway, E.**, **Bourgeois, M.**, **Mitter, V. and Heile, B.** (2001). IEEE 802.15.4: a developing standard for low-power low-cost wireless personal area networks, *Network, IEEE*, ***15***(5), 12–19.

[26] **Devadiga, K.** (2013). Development of a Multicast Routing Protocol for Low power and Lossy Networksi, *Ph.D. thesis*, Aalto University.

[27] **Michopoulos, V.** (2012). Congestion and medium access control in 6LoWPAN WSN, *Ph.D. thesis*, Loughborough University.

[28] `http://cnds.eecs.jacobs-university.de/slides/2010-aims-6lowpan.pdf`, [Online; accessed 13-April-2015].

[29] **Wang, F.**, **Li, D. and Zhao, Y.** (2011). Analysis of CSMA/CA in IEEE 802.15.4, *Communications, IET*, ***5***(15), 2187–2195.

[30] `https://tools.ietf.org/html/rfc2460`, [Online; accessed 13-April-2015].

[31] `http://tools.ietf.org/html/rfc7414`, [Online; accessed 13-April-2015].

[32] **Zach Shelby, C.B.** (2009). *6LoWPAN: The Wireless Embedded Internet*, Wiley.

[33] `http://tools.ietf.org/html/rfc4944`, [Online; accessed 13-April-2015].

[34] `http://www.ietf.org/html.charters/6lowpan-charter.html.`, [Online; accessed 13-April-2015].

[35] `https://tools.ietf.org/html/rfc2765`, [Online; accessed 13-April-2015].

[36] **Hui, J.**, **Culler, D. and Chakrabarti, S.**, `http://www.ipso-alliance.org/wp-content/media/6lowpan.pdf`, [Online; accessed 13-April-2015].

[37] **Hui, J. and Culler, D.** (2008). Extending IP to Low-Power, Wireless Personal Area Networks, *Internet Computing, IEEE*, ***12***(4), 37–45.

[38] **Bansal, B.**, **Tripathy, M.R.**, **Goyal, D. and Goyal, M.** (2015). Improved Routing Protocol for MANET, *Advanced Computing Communication Technologies (ACCT), 2015 Fifth International Conference on*, pp.340–346.

[39] **Park, S.**, **Crespi, N.**, **Park, H. and Kim, S.H.** (2014). IoT routing architecture with autonomous systems of things, *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pp.442–445.

[40] **Iyer, G.**, **Agrawal, P. and Cardozo, R.** (2013). Performance comparison of routing protocols over smart utility networks: A simulation study, *Globecom Workshops (GC Wkshps), 2013 IEEE*, pp.969–973.

[41] **Dawson-Haggerty, S.**, **Tavakoli, A. and Culler, D.** (2010). Hydro: A Hybrid Routing Protocol for Low-Power and Lossy Networks, *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pp.268–273.

[42] **Yu, H. and He, J.** (2011). Improved hierarchical routing over 6LoWPAN, *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, pp.377–380.

[43] **Ke, X.**, **SHEN, M.**, **Liu, H.**, **Liu, J.**, **Li, F. and Li, T.** (2015). Achieving Optimal Traffic Engineering Using a Generalized Routing Framework, *Parallel and Distributed Systems, IEEE Transactions on*, ***PP***(99), 1–1.

[44] **Yi, J.**, **Clausen, T. and Herberg, U.** (2015). Depth First Forwarding for Unreliable Networks: Extensions and Applications, *Internet of Things Journal, IEEE*, ***PP***(99), 1–1.

[45] **Clausen, T.**, **Yi, J. and de Verdiere, A.** (2012). LOADng: Towards AODV Version 2, *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, pp.1–5.

[46] **Li, H.**, **Ma, Y.**, **Liu, W. and Fu, Y.** (2015). Improved analysis of greedy block coordinate descent under RIP, *Electronics Letters*, ***51***(6), 488–490.

[47] **Kabir, T.**, **Nurain, N. and Kabir, M.** (2015). Pro-AODV (Proactive AODV): Simple modifications to AODV for proactively minimizing congestion in VANETs, *Networking Systems and Security (NSysS), 2015 International Conference on*, pp.1–6.

[48] **Gupta, M. and Kumar, S.** (2015). Performance Evaluation of DSR, AODV and DSDV Routing Protocol for Wireless Adhoc Network, *Computational Intelligence Communication Technology (CICT), 2015 IEEE International Conference on*, pp.416–421.

[49] **Ledy, J.**, **Hilt, B.**, **Boeglen, H.**, **Poussard, A.M.**, **Drouhin, F.**, **Vauzelle, R. and Lorenz, P.** (2015). Impact of realistic simulation on the evaluation of mobileAd-hoc routing protocols, *Emerging Topics in Computing, IEEE Transactions on*, ***PP***(99), 1–1.

[50] **Bin Zikria, Y.**, **Nosheen, S. and Kim, S.W.** (2015). Quality of service analysis for multimedia traffic using DSR, AODV and TORA over Wi-Media ultra wide band, *Applied Sciences and Technology (IBCAST), 2015 12th International Bhurban Conference on*, pp.539–546.

[51] `https://datatracker.ietf.org/wg/roll/charter`, [Online; accessed 13-April-2015].

[52] **Gaddour, O. and KoubíA, A.** (2012). Survey RPL in a Nutshell: A Survey, *Comput. Netw.*, ***56***(14), 3163–3178, `http://dx.doi.org/10.1016/j.comnet.2012.06.016`.

[53] `http://tools.ietf.org/html/draft-gnawali-roll-etxof-00`, [Online; accessed 13-April-2015].

[54] **Akyildiz, I. and Wang, X.** (2009). *Wireless Mesh Networks*, Wiley.

[55] `http://tools.ietf.org/html/draft-ietf-roll-of0-12`, [Online; accessed 13-April-2015].

[56] http://meetings.apnic.net/__data/assets/pdf_file/0012/31800/IP-in-Smart-Object-Networks_V1-8_APRICOT.pdf, [Online; accessed 13-April-2015].

[57] https://tools.ietf.org/html/rfc4443, [Online; accessed 13-April-2015].

[58] **Grajzer, M. and Glabowski, M.** (2014). Performance evaluation of Neighbor Discovery x002B; x002B; protocol for the provisioning of self-configuration services in IPv6 mobile ad hoc networks, *Telecommunications Network Strategy and Planning Symposium (Networks), 2014 16th International*, pp.1–6.

[59] **Fang, S.**, **Berber, S. and Swain, A.** (2008). Analysis of Neighbor Discovery Protocols for Energy Distribution Estimations in Wireless Sensor Networks, *Communications, 2008. ICC '08. IEEE International Conference on*, pp.4386–4390.

[60] **Chen, L.**, **Li, Y.**, **Chen, Y.**, **Liu, K.**, **Zhang, J.**, **Cheng, Y.**, **You, H. and Luo, Q.** (2015). Prime-set-based neighbour discovery algorithm for low duty-cycle dynamic WSNs, *Electronics Letters*, ***51***(6), 534–536.

[61] **Liu, P.**, **Sun, R. and Yan, Z.** (2011). Dynamic OSPF protocol, *Broadband Network and Multimedia Technology (IC-BNMT), 2011 4th IEEE International Conference on*, pp.51–55.

[62] **Rusu, O. and Vraciu, V.** (2010). IS-IS metric optimization, *Roedunet International Conference (RoEduNet), 2010 9th*, pp.282–286.

[63] **Capone, S.**, **Brama, R.**, **Accettura, N.**, **Striccoli, D. and Boggia, G.** (2014). An Energy Efficient and Reliable Composite Metric for RPL Organized Networks, *Embedded and Ubiquitous Computing (EUC), 2014 12th IEEE International Conference on*, pp.178–184.

[64] https://tools.ietf.org/html/draft-ietf-roll-routing-metrics-19, [Online; accessed 13-April-2015].

[65] **Guan, L.**, **Kuladinithi, K.**, **Potsch, T. and Goerg, C.** (2014). A deeper understanding of interoperability between TinyRPL and ContikiRPL, *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pp.1–6.

[66] https://tools.ietf.org/html/rfc6552, [Online; accessed 13-April-2015].

[67] **Samanta, M. and Banerjee, I.** (2014). Optimal load distribution of cluster head in fault-tolerant wireless sensor network, *Electrical, Electronics and Computer Science (SCEECS), 2014 IEEE Students' Conference on*, pp.1–7.

[68] **Moghadam, M. and Taheri, H.** (2014). High throughput load balanced multipath routing in homogeneous wireless sensor networks, *Electrical Engineering (ICEE), 2014 22nd Iranian Conference on*, pp.1516–1521.

[69] **Lenoir, V.**, **Lattard, D.**, **Dehmas, F.**, **Morche, D. and Jerraya, A.** (2014). Computational load reduction by downsampling for energy-efficient digital baseband, *New Circuits and Systems Conference (NEWCAS), 2014 IEEE 12th International*, pp.333–336.

[70] **Perumal, T.**, **Sulaiman, M.**, **Mustapha, N.**, **Shahi, A. and Thinaharan, R.** (2014). Proactive architecture for Internet of Things (IoTs) management in smart homes, *Consumer Electronics (GCCE), 2014 IEEE 3rd Global Conference on*, pp.16–17.

[71] `https://www.sics.se/media/news/contikirpl-the-new-default-contiki-ipv6` [Online; accessed 13-April-2015].

[72] `https://github.com/contiki-os/contiki/blob/master/core/net/rpl/rpl-mrhof.c`, [Online; accessed 13-April-2015].

[73] `http://www.contiki-os.org/`, [Online; accessed 13-April-2015].

[74] **Preiss, T.**, **Sherburne, M.**, **Marchany, R. and Tront, J.** (2014). Implementing dynamic address changes in ContikiOS, *Information Society (i-Society), 2014 International Conference on*, pp.222–227.

[75] `https://github.com/contiki-os/contiki/tree/master/core/net/rpl`, [Online; accessed 13-April-2015].

[76] `https://totalueberwachung.de/blog/files/seminar-contiki.pdf`, [Online; accessed 13-April-2015].

[77] `http://dunkels.com/adam/dunkels07rime.pdf`, [Online; accessed 13-April-2015].

[78] **Ozen, S. and Oktug, S.** (2014). Forwarder set based dynamic duty cycling in asynchronous wireless sensor networks, *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*, pp.2432–2437.

[79] **Tang, H.**, **Sun, C.**, **Liu, Y. and Fan, B.** (2013). Low-latency asynchronous duty-cycle MAC protocol for burst traffic in wireless sensor networks, *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pp.412–417.

[80] `https://github.com/contiki-os/contiki/wiki/Radio-duty-cycling#The_Contiki_XMAC`, [Online; accessed 13-April-2015].

[81] **Ullah, A.**, **Ahn, J.S. and Kim, G.** (2013). X-MAC protocol with Collision Avoidance algorithm, *Ubiquitous and Future Networks (ICUFN), 2013 Fifth International Conference on*, pp.228–233.

[82] `http://www.cs.berkeley.edu/~culler/papers/ko11contikirpl.pdf`, [Online; accessed 13-April-2015].

[83] `https://github.com/contiki-os/contiki/wiki/An-Introduction-to-Cooja`, [Online; accessed 13-April-2015].

[84] **Heddeghem, W.V.** (2009). Cross-layer Link Estimation For Contiki-based Wireless Sensor Networks, *Ph.D. thesis*, Vrije University, Brussel Belgium.

[85] `http://ijcsi.org/papers/IJCSI-11-5-1-27-30.pdf`, [Online; accessed 13-April-2015].

[86] **Chitlange, M.M. and Deshpande, V.S.** (2015). Effect of node density on congestion in WSN, *Pervasive Computing (ICPC), 2015 International Conference on*, pp.1–3.

[87] **Rahmani, A.M.**, **Kamali, I.**, **Lotfi-Kamran, P.**, **Afzali-Kusha, A. and Safari, S.** (2009). Negative Exponential Distribution Traffic Pattern for Power/Performance Analysis of Network on Chips, *VLSI Design, 2009 22nd International Conference on*, pp.157–162.

[88] **Xu, B.**, **Chen, H.**, **Zhu, W. and Zhu, X.** (2013). Multi-objective cost-sensitive attribute reduction, *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint*, pp.1377–1381.

APPENDICES

APPENDIX A.1 : Topologies

APPENDIX A.1

All of the below topologies were simulated for DODAG construction with OF 1 (MRHOF) and OF 2 (PAOF) as prepraration work. Topology 1, 6 and 8 were selected for performance evaluation.

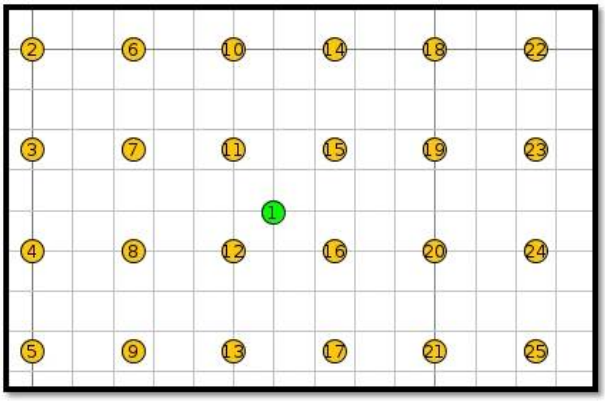


Figure A.1: Topology 1

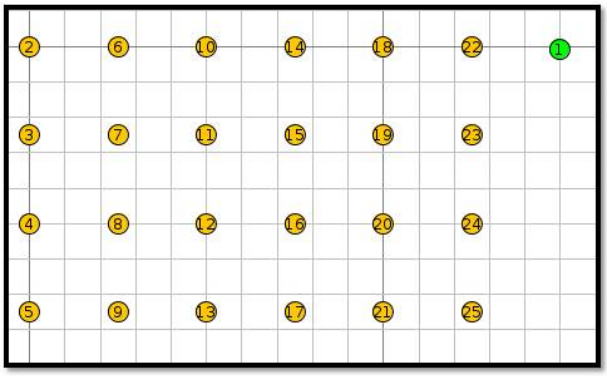


Figure A.2: Topology 2

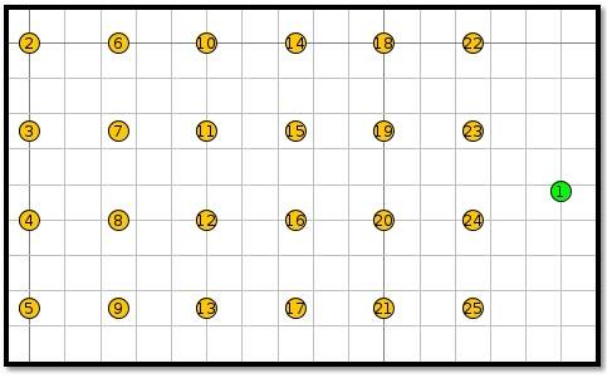


Figure A.3: Topology 3

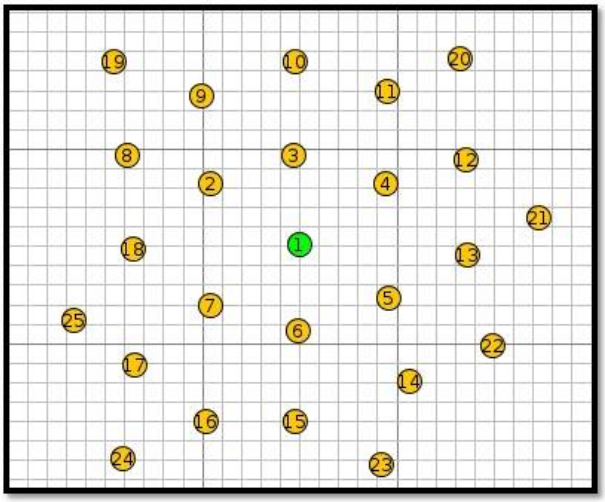


Figure A.4: Topology 4

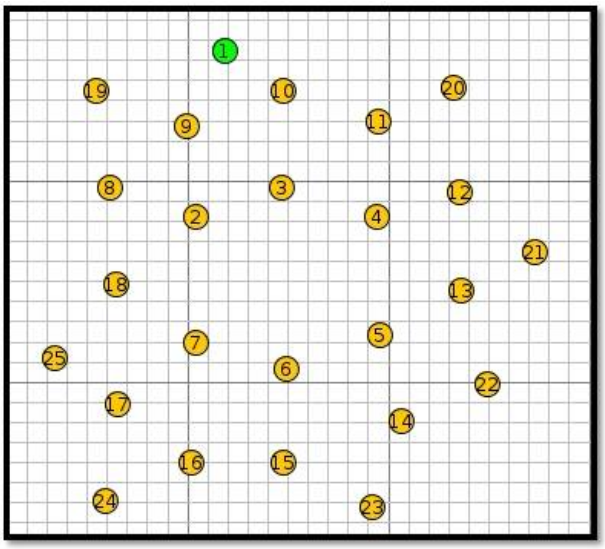


Figure A.5: Topology 5

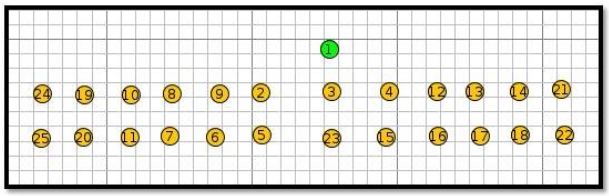


Figure A.6: Topology 6

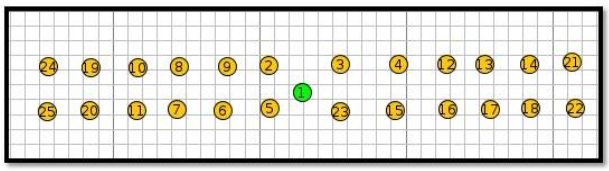


Figure A.7: Topology 7

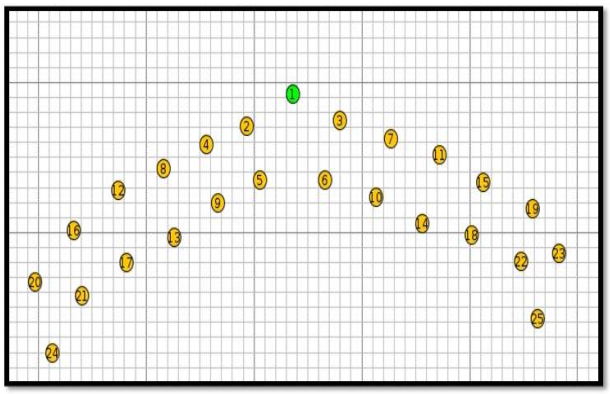


Figure A.8: Topology 8

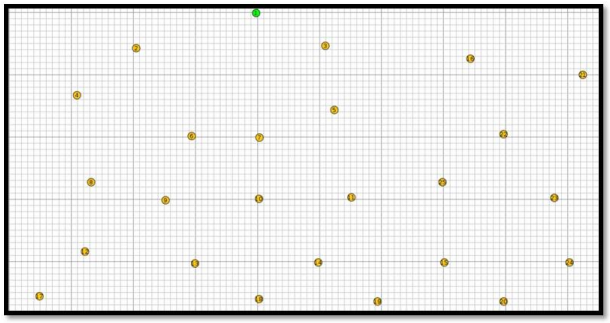


Figure A.9: Topology 9

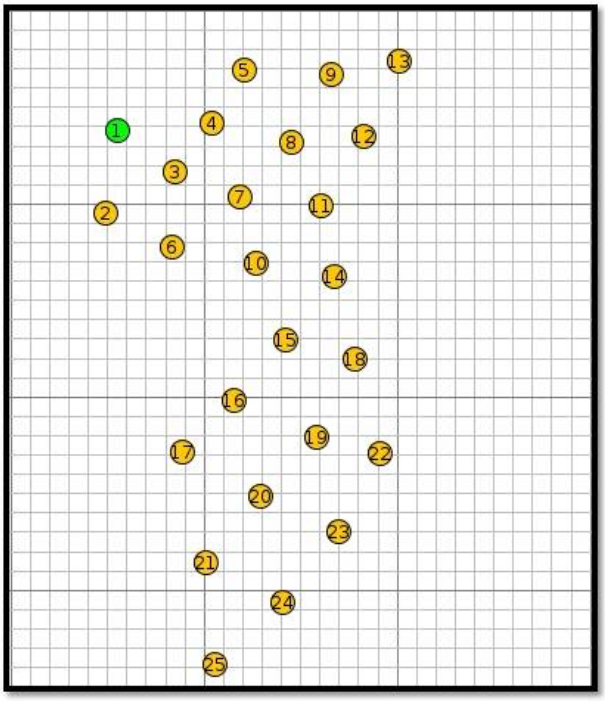


Figure A.10: Topology 10

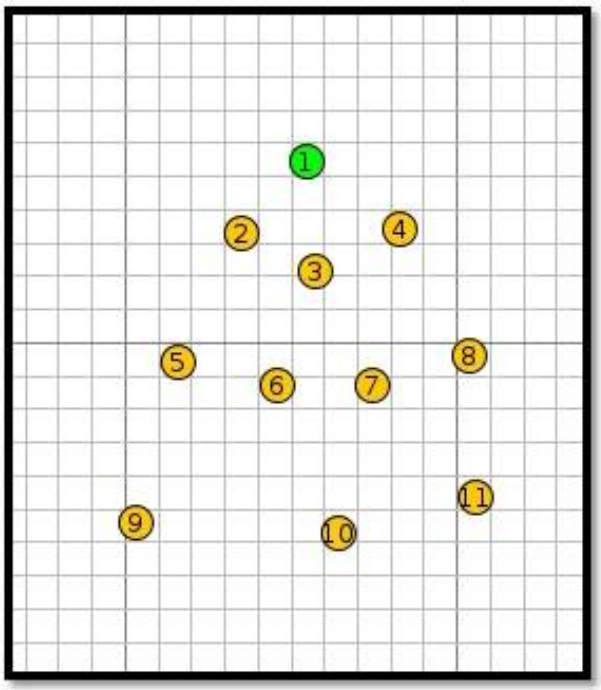


Figure A.11: Topology 11

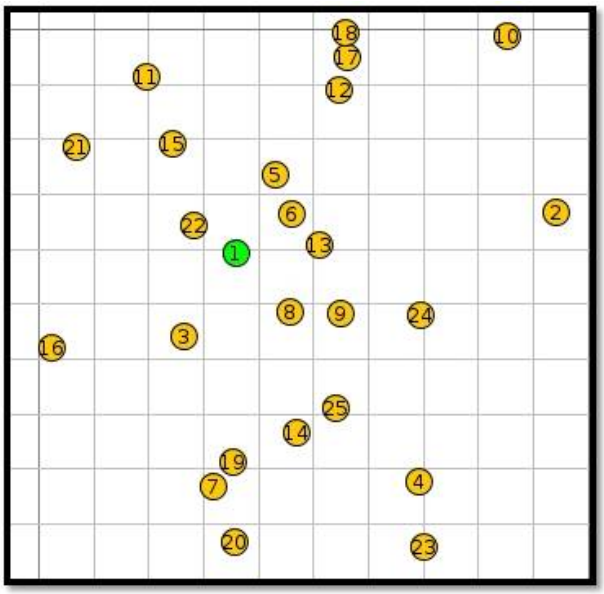


Figure A.12: Topology 12

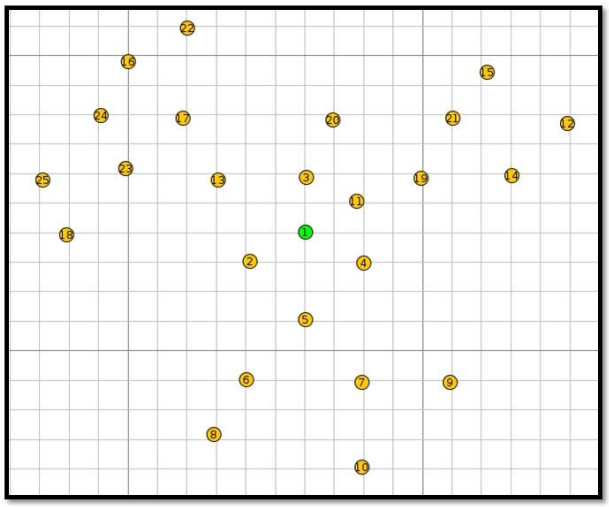


Figure A.13: Topology 13

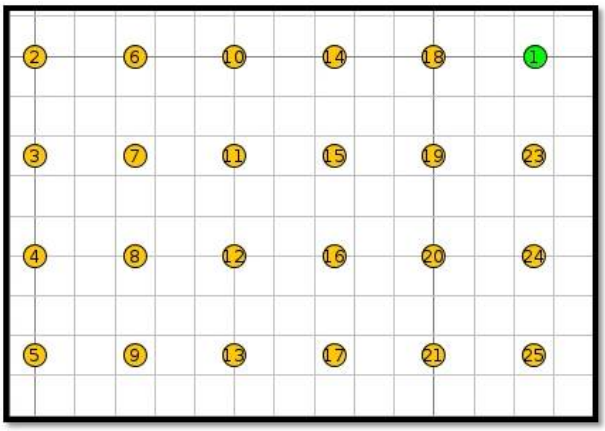


Figure A.14: Topology 14

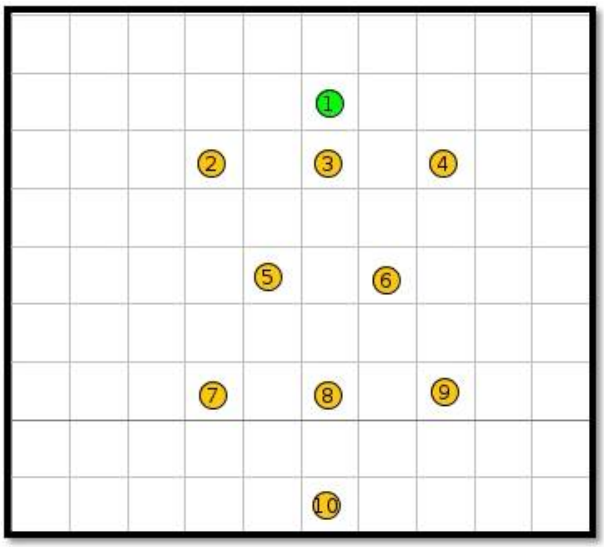


Figure A.15: Topology 15

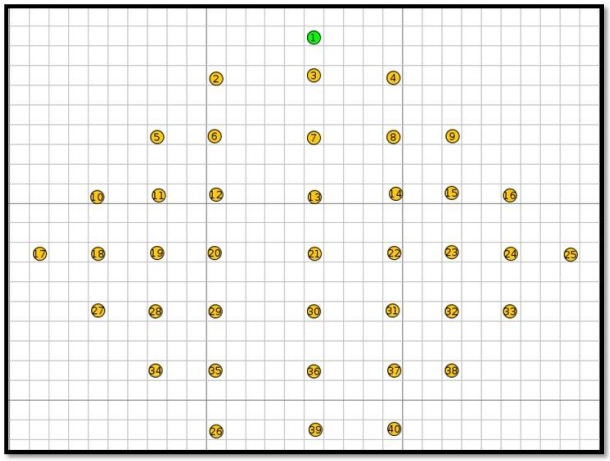


Figure A.16: Topology 16

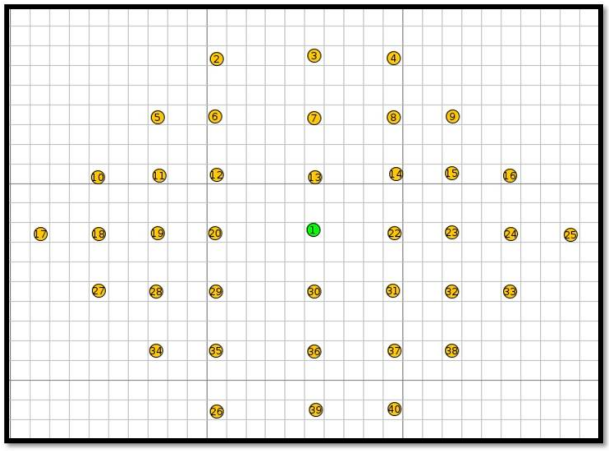


Figure A.17: Topology 17

CURRICULUM VITAE

Name Surname: Necip Gozuacik

Place and Date of Birth: Sakarya, 25.01.1985

Adress: Çamçeşme Mah. Aydınlı Cad. Tuğcular Sok. No:10/1 Pendik/İSTANBUL

E-Mail: gozuacik@gmail.com

B.Sc.: Yıldız Technical University, Electronic and Communication Engineering, 2006
Yıldız Technical University, Computer Engineering, 2007

Professional Experience and Rewards: Senior Software Architect in Netas (2006 -)

List of Publications and Patents: VoIP Santral Çekirdek Bileşeninde Yazılım Yaması, Proceedings of the 8th Turkish National Software Engineering Symposium, Güzelyurt, KKTC, Turkey, September 8-10, 2014

PUBLICATIONS/PRESENTATIONS ON THE THESIS

- **Gozuacik N.**, Oktug S., 2015: Parent Aware Routing for IoT Networks *The 8th conference on Internet of Things and Smart Spaces ruSMART*, August 26-August 28, 2015 St.-Petersburg, Russia.