LOAD BALANCING ENHANCEMENTS TO THE ROUTING PROTOCOL FOR LOW POWER AND LOSSY NETWORKS IN THE INTERNET OF THINGS

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

By

Hira Noor

November 2018

LOAD BALANCING ENHANCEMENTS TO THE ROUTING PROTOCOL FOR LOW POWER AND LOSSY NETWORKS IN THE INTERNET OF THINGS

By Hira Noor

November 2018

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Nail Akar(Advisor)

Ibrahim Korpeoglu

Mehmet Koseoglu

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

LOAD BALANCING ENHANCEMENTS TO THE ROUTING PROTOCOL FOR LOW POWER AND LOSSY NETWORKS IN THE INTERNET OF THINGS

Hira Noor
M.S. in Electrical and Electronics Engineering
Advisor: Nail Akar

November 2018

The internet today is shifting from the Internet of people to the Internet of Things (IoT). Particularly, in IoTs, wireless sensors connect edge devices to the Internet via a gateway that provides connectivity between wireless sensor networks (WSNs) and the Internet. IoT includes a variety of heterogeneous network applications ranging from smart grid automated metering infrastructures (AMIs), industrial and environmental monitoring networks to building automation. In WSNs, congestion causes a plenty of impairments such as increased packet losses, lower throughput, and energy wastage thus decreasing the lifetime and performance of wireless sensor applications. IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN) is envisioned to be used in the majority of IoT applications. Recently, the Internet Engineering Task Force (IETF) Routing over Low power and Lossy Networks (ROLL) working group has proposed a Routing Protocol for Low power and Lossy networks called RPL. RPL is often studied in a multipoint-to-point sink node (MP2P) scenarios. We investigate the load balancing and congestion problem of RPL. RPL suffers from congestion and unbalanced load distribution due to the use of a single path for multipoint-to-point traffic. In particular, we propose queue utilization-based multipath RPL (QU-MRPL). In QU-MRPL, multiple parents are selected based on their queue size information. We demonstrate that QU-MRPL achieves load balance in the network and thus increases the packet delivery ratio.

Keywords: Routing Protocol for Low power and Lossy Networks (RPL), Wireless Sensor Networks (WSNs), Internet of Things (IoT), Load Balancing.

ÖZET

NESNELERİN İNTERNETİ İÇİN DÜŞÜK GÜÇ KULLANAN KAYIPLI AĞLARDA YÖNLENDİRME PROTOKOLÜNE YÖNELİK YÜK DENGELEME İYİLEŞTİRMELERİ

Hira Noor
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Danışmanı: Akademik Ünvansız isim X
Kasım 2018

İnternet bugün insanların İnternet'inden Nesnelerin İnterneti'ne (IoT) doğru kaymaktadır. IoT konsepti, akıllı şebekeden otomatikleştirilmiş ölçüm altyapıları (AMI'ler), endüstriyel ve çevresel izleme ağları, bina otomasyonu gibi çeşitli heterojen ağ uygulamalarını içerir. IoT'lerde sensörler ve aktüatörleri barındıran kablosuz uç cihazlar, kablosuz sensör ağları ile erişebildikleri bir ağ geçidi üzerinden İnternet'e bağlanırlar. WSN'lerde, tıkanıklık önemli bir problem olup, bu nedenle artan paket kayıpları, daha düşük iş yapma yeteneği, ve artan enerji israfı dolayısıyla kablosuz sensör uygulamalarının ömür ve performans azalır. IoT uygulamalarının çoğunda, ağ protokolü olarak IPv6 6LoWPAN teknolojisinin kullanılması öngörülmektedir. Yakın geçmişde, İnternet Mühendislik Görev Gücünün (IETF) Düşük Güç ve Kayıplı Ağlar (ROLL) çalışma grubu, 6LoWPAN'lar için RPL isimli bir yönlendirme protokolü geliştirdi. Bu tezde, RPL protokülüne ait olan çok noktadan tek noktaya olarak adlandırılan senaryoda yük dengeleme ve tıkanıklık problemleri irdelenmektedir. Bu çalışmada, Kuyruk Uzunluk Tabanlı Birden Çok Yollu bir RPL yöntemi önerilmekte, bu önerilen yöntemde her düğüm, kök düğüme trafik göndermek için bir ebeveyn yerine birden çok ebeveyn seçmektedir. Simulasyonlar ile önerilen yöntem sınanmış ve farklı durum ve topolojilerde, yük dağılımının iyileşdiği ve paket teslimat oranının arttırıldığı gösterilmişdir.

Anahtar sözcükler: Düşük güç kullanan ve kayıplı ağlarin yönlendirme protokolü, Kablosuz sensör ağları, Nesnelerin interneti, Yük dengeleme.

Acknowledgement

I would like to express my deepest gratitude to my advisor Prof. Nail Akar for believing in my potential and giving me the opportunity to study in Bilkent. Thank you for guiding me throughout my studies. I find myself lucky to have a supervisor like Sir Nail.

I would like to acknowledge the TÜBİTAK support of my research as a scholar in ARDEB-1001 project number 115E360.

I also want to thank Prof. Ibrahim Korpeoglu and Prof. Mehmet Koseoglu for accepting to be part of my thesis jury, for their time and their useful suggestions.

A big thanks to all my friends in Bilkent. It was a great pleasure to be with you guys (I am not writing the names here because the list is very long :p). Bilkent wouldn't have been the same without you guys.

I also want to give huge thanks to my mother and my brothers ali and ahmad for their unconditional love, support and care. Mama, thank you for believing in me and letting me pursue my dream, and for handling so well the fact that your only daughter is thousands of miles away from you. My mama is my greatest inspiration and has taught me that love and kindness conquers all. None of this wouldn't have been possible without you. YOU are my whole world. I am lucky to have you in my life.

I also want to thank ALLAH Almighty for showering his countless blessings on me. I am blessed to have everything in my life that I don't deserve at all. HE is the kindest.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The advancement in wireless technologies has enabled the realization of new network architectures such as Cognitive Radio Networks (CRN), Wireless Sensor Networks (WSNs) and Mobile Adhoc Networks (MANETs) [4]. In particular, Wireless Sensor Networks (WSNs) connect edge devices with sensors and actuators to the Internet which is known as the Internet of Things (IoT) [5]. The number of interconnected devices is increasing exponentially [6]. According to an estimate by Cisco, the world has 20 billion interconnected devices today. However, by 2020, it is going to increase to 50 billion devices worldwide [7]. Internet of Things not only includes homogeneous WSNs but also heterogeneous WSNs that are a part of our daily lives [8]. WSNs are composed of a large number of sensor nodes which cooperatively interact with each other in short distances and utilize their processing capabilities to carry out complex monitoring tasks [9]. Since sensing applications generate large quantities of data, this data may be aggregated or fused together to lower the energy consumption. The source nodes transmit time series of the sensed phenomenon to the central node called sink node. The sink node serves as a gateway between the wireless sensor network and the Internet. It provides connectivity to the Internet and provides access to a central repository or a server where the user uses the sensed data for a particular application. Fig. 1.1 shows a typical WSN scenario. WSNs have advantages over traditional communication networks like low cost, ease of deployment, flexibility

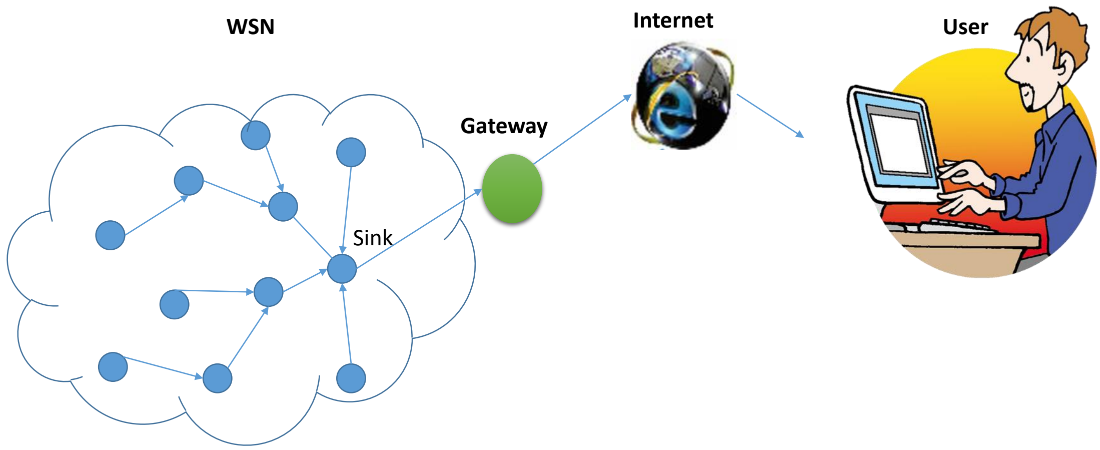


Figure 1.1: An illustration of a WSN

and parallel processing. In essence, WSNs with these capabilities may revolutionize the world and bring people together by providing better connectivity and ease of access.

Sensor nodes and smart devices in wireless sensor networks are interconnected by a variety of links, for instance, IEEE 802.15.4 [10], low-power PLC (power line communication), Low power Wi-Fi and Bluetooth. These links are lossy because they have limited resources. The devices in sensor networks are also resource-constrained because they have size constraints and moreover they need to be low power and low cost. These networks are also termed as Low power and Lossy Networks (LLNs).

Inherent properties of individual sensor nodes pose additional challenges in the design of communication protocols for LLNs. Sensor nodes are mostly battery-powered therefore it is vital that they consume energy very efficiently in terms of computation and communication. Communication activities are more power consuming as compared to computation activities, i.e., transmission and reception consume most part of the energy. The deployment of sensor nodes in WSNs is another aspect that influences the WSN protocols' design and development. WSN do not require the information about positioning of nodes. Therefore, they can be randomly deployed in inaccessible areas like underground monitoring applications or disaster relief operations. On the other hand, the routing protocols for WSNs should be self-organizing because of the random deployment. The density of sensor nodes also plays an important in WSN routing protocols. Considering the limited transmission range of sensor nodes, plenty of sensor nodes can be deployed in a WSN. Therefore, multi-hop communication is commonly used between sensor nodes. WSN has many applications ranging from the smart grid, smart cities, home automation, industrial automation, environmental monitoring to health monitoring. In industrial automation, sensors are deployed for fault detection and alert reporting. Common examples of WSNs are Body Area Network (BAN) and Personal Area Networks (PAN).

In order to meet the stringent and resource-constrained requirements of low power and lossy WSNs, extremely efficient and sophisticated communication protocols are required. In this respect, the Internet Engineering Task Force (IETF) has proposed 6LoWPAN [11] to enable the integration and communication between IPv6 devices and low power sensor nodes. This will allow the IPv6 based devices to connect with sensor nodes. However, an efficient and reliable communication stack is still required for seamless connectivity between WSNs and the Internet. The coexistence of WSN and WLANs also pose challenges for the MAC layer protocols. The transport layer protocols like TCP and UDP that are widely used in the Internet today, can not be used in WSNs. In most sensor networks, the sink or the central node is assumed to reside in the network or very close to it, therefore multi-hop communication is used in WSNs. Subsequently, new protocols are required for the transport layer to provide reliable transport of data through the WSN and the Internet. Moreover, the Internet protocols are not energy- and memory-efficient since these performance metrics have not received serious consideration throughout the development of the Internet. Therefore, we need effective mechanisms to connect today's Internet with the future Internet of Things.

Routing is of paramount importance in a multi-hop WSN as an LLN node not only transmits its own packets towards the destination (sink node) but also forwards the packets of neighbouring nodes in its subtree. An LLN typically contains several alternative paths towards a single destination but with different link qualities. It becomes a salient feature of the routing protocol to intelligently choose the routing paths from a source to a destination. Poor path selection leads to depletion of scarce resources quickly. Therefore it is crucial to analyze and estimate the resource consumption and efficiency of the routing protocol in these devices. Conventional link state routing protocols like OSPF, IS-IS and OLSR do not meet the resource constraints of LLNs. Contrarily, distance vector routing protocols like AODV, RIP and IGRP cannot provide fast recovery in case of frequent topology changes and link failures. In order to meet the requirements of LLNs, Internet Engineering Task Force (IETF) Routing over Low Power and Lossy Networks (ROLL) Working Group proposed a Routing Protocol for Low Power and Lossy Networks (RPL) [12]. The ROLL working group published the RFC 6550 in 2012, which specified the routing protocol for Low power and Lossy Networks (RPL).

RPL is an IPv6 based proactive distance vector routing protocol designed for low power and lossy networks. It is an emerging standard for routing in low power and lossy networks. RPL is a source routing protocol that is designed to operate on top of the 802.15.4 MAC layer and physical layer. RPL is designed for collection networks (multi-point to point traffic) where a central node coordinates and receives data from the overall network. RPL can also be used for point to point and point to multi-point traffic scenarios. RPL provides a mechanism to disseminate routing information over the dynamically formed network topology called DODAG (Direction-Oriented Directed Acyclic Graph). We can think of DODAG as a logical routing topology that is built by the routing protocol over the physical network.

RPL builds a tree-like structure called the Directed Acyclic Graph (DAG) in which all the traffic from a subtree is directed to a parent node. The *rank* defines the relative position of nodes in the graph with respect to the sink node. The rank of a node in the DODAG needs to be monotonically increasing in the downward direction and strictly decreasing in the upward direction to avoid loops in the network. Fig. 1.4 illustrates the node relationship set. RPL uses an objective function to calculate the rank of nodes. An Objective Function (OF) defines the rules for calculating the rank of the node, selecting the parent nodes and optimizing the routing paths based on the choice of certain routing metrics and

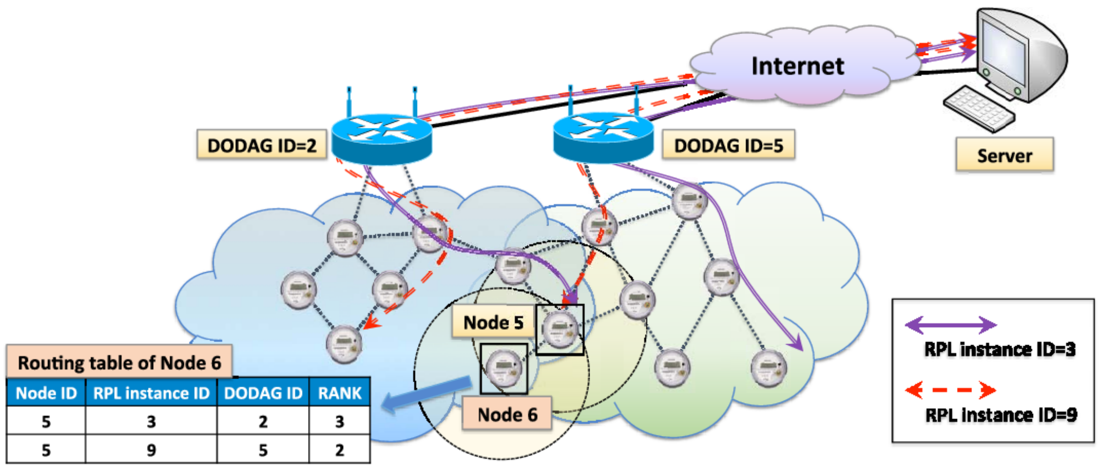


Figure 1.2: An illustration of RPL DAG structure with multiple DODAG roots and multiple instances [1]

constraints.

RPL uses DIO messages for constructing and maintaining upward routes toward the DODAG root (sink node). Upward routes are constructed to provide multi-point-to-point (MP2P) traffic support. Fig. 1.3 illustrates the exchange of RPL control messages during DODAG creation. The MP2P traffic pattern is a significant feature of data collection networks where a central node collects data from the overall network. This central node can serve as an LLN border router and transmit the collected data to an online storage space or a server.

We have considered a multi-point-to-point traffic pattern as it is most commonly used in wireless sensor applications for collecting data. In this thesis work, we aim at analyzing the performance of RPL and related issues in Contiki and propose an inter-operable and efficient implementation of RPL named QU-MRPL.

The Contiki operating system is an open source operating system for smart objects. It provides both the networking implementation as well as application support for motes [13]. Contiki provides mechanisms for programming smart object applications and assisting communication between them. Contiki oprsting system and its applications are developed in the C programming language, which makes it highly portable. The applications running Contiki OS can directly communicate with other IP-based applications, web service and Internet-based

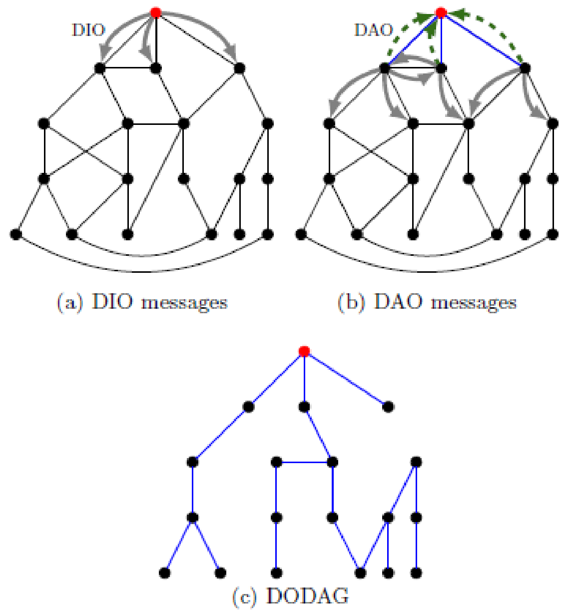


Figure 1.3: RPL control message exchange

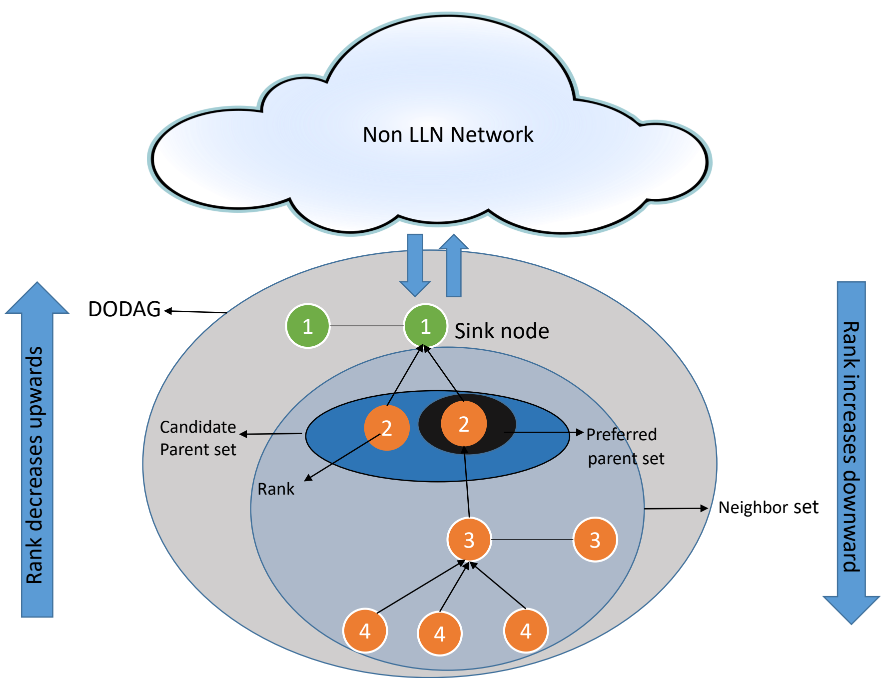


Figure 1.4: RPL DODAG example with node set relationships

services because Contiki contains the entire IP stack. Contiki has a number of features including radio duty cycling, ContikiMAC, IPv6 stack and COOJA simulator. Contiki provides an implementation of various IETF standards for smart objects including IETF 6LowPAN, IETF RPL (Routing protocol for low power and lossy networks) and IETF COAP. Contiki operating system provides a cross-layer implementation which means there are separate modules for every layer of the protocol stack, for example, the routing module is present in a separate directory "contiki/core/net/rpl" that provides complete features of the IETF Routing protocol for low-power and lossy networks.

1.1 Motivation

The topology of a WSN is unstable due to low power and lossy links. Thus the link quality in this network might be varying. Efficient data distribution gained a lot of attention, especially for low power and lossy networks as the number of connected devices is increasing day by day. IETF's ROLL working group has proposed RPL for routing in low power and lossy networks. Evidently, a lifetime of a network and reliability of data are two key features of network quality. To make communication effective, data should be reliably delivered to a central data collector node. However, in a multihop network some of the nodes in the network may have to take part in many communication tasks, which drains out their energy quickly, therefore cutting down the lifetime of the network. In particular, if some nodes are always selected to transmit data, their energy drains out faster and they become the bottleneck nodes. On the other hand, since wireless sensors are deployed in dense environments, they need to transmit heavy traffic and the node queues will probably overflow. This causes unreliable transmission of data. The buffer overflow is a critical factor leading to packet drops in the network as the node can only receive the data but cannot store it because of the limited buffer capacity [14] [15]. In order to balance the load distribution among nodes in the network, we should design a suitable transmission mechanism to avoid bottleneck nodes in the network. In this case, also the node becomes the bottleneck node even if it has sufficient energy. Hence, we should balance the node buffers to increase reliability and reduce packet drops.

1.2 Problem Statement

RPL is designed for LLNs. However, load balancing is missing in the standard RPL. RPL allows nodes to save a list of candidate parent nodes but only one preferred parent node is selected to forward data while others are kept as backup for fault tolerant purposes. Considering a network with asymmetric node distribution and irregular data traffic, only one preferred parent may result in significant load imbalance. In a real network deployment, the sensor nodes that have better link quality might have more child nodes in their subtree than other nodes and thus the energy consumption of these nodes gets accelerated. They will have higher energy consumption than nodes with lighter traffic. These nodes easily become the bottleneck nodes thus disconnecting a part of the network. The load and buffer imbalance of these nodes severely affects the network lifetime and performance. Therefore, preventing bottleneck nodes in the network is a crucial problem to be solved. We need to design an efficient data distribution and load balancing mechanism. RPL separates the load balancing problem from the standard (RFC6550) and assumes that the OF will deal with it. However, both the objective function proposed by RPL aim to reduce the re-transmission in the network by considering the link quality. Consequently, this results in uneven energy consumption among nodes and congestion at some bottleneck nodes. In order to achieve load balancing and congestion avoidance, we propose a multipath routing scheme based on queue utilization information. The proposed implementation performs better load distribution and alleviates congestion in the network.

1.3 Contribution

The goal of this thesis is to propose a multipath routing solution that will mitigate congestion and balance the load in the network. The algorithm that we have proposed increases the packet delivery ratio and maximizes the lifetime of bottleneck nodes. We have implemented the algorithm in Contiki operating system and simulations are done in the Cooja simulator. The modifications to the operating system makes it possible to logically route to multiple parent nodes in case of congestion in the network. This feature is currently missing in RPL. Filling this gap existent in RPL is advantageous for high data rate applications.

1.4 Thesis Organization

This thesis is organized as follows. Chapter 2 presents the background of WSNs, the protocol stack and its implementation with RPL used in this work. Chapter 3 describes the proposed algorithm for multipath routing in RPL. It continues with the implementation and modification of the Contiki operating system for the proposed work. Chapter 4 validates the effectiveness of the proposed work through extensive simulations. Finally, we conclude our work in Chapter 5. Some future aspects of the work are also described.

Chapter 2

Background

2.1 Low Power and Lossy Networks (LLNs)

Low-power and lossy networks (LLNs) are the networks with resource-constrained devices. The routers and the interconnects in an LLN have memory constraints and they are battery operated. Because of the low-power and short range, the links are usually “lossy”. They are characterized by unstable connectivity. The traffic in LLNs is usually multipoint-to-point. They used composed of thousands of nodes.

2.2 IPV6 over Low Power Wireless Personal Area Networks (6LowPAN)

A LoWPAN is a particular instance of an LLN in which the devices are interconnected with IEEE 802.15.4 links. The main difference between an LLN and a LoWPAN is that LowPAN is restricted to IEEE 802.15.4 network whereas an LLN can have other links as well such as power line communication (PLC) and WiFi. The IPv6 over Low-power wireless personal area networks (6LowWPAN) working group was formed to work on a mechanism for IPv6 communication over IEEE 802.15.4 protocol stack for LowPAN networks. 6LoWPAN [16] acts as an adaptation layer between the standard IPv6 world and the low power and lossy communications wireless media offered by IEEE 802.15.4 [17]. A LoWPAN is typically composed of devices that collaborate with each other to connect the physical deployment with the real world applications e.g wireless sensors. The key characteristics of LoWPANs include:

1. **Small packet size:** Given that maximum packet size at the physical layer is 127 bytes, the available maximum frame size at the medium access control (MAC) layer is 102 octets. Depending on the link layer security mechanism, it adds further overhead. This leaves 81 octets available for data packets, which is far below the minimum MTU size of the IPv6 packet according to the standard specification.

2. **IEEE 802.15.4 addressing modes**: There are several addressing modes defined for IEEE 802.15.4 devices. The devices can either use 16-bit short addresses in the personal area network (PAN) or 64-bit extended addresses.

3. **Low bandwidth:** The LLN links have low bandwidths for the currently defined physical layers i.e, 2.4 GHz, 915 MHz and 868 MHz and consequently low data rates ranging from 20 kbps to 250 kbps respectively

4. **Topologies:** support for star and mesh topologies.

5. **Large scale networks:** LLNs are considered to be deployed on a large scale for monitoring purposes. The location of these LLN devices mostly wireless sensors is not predefined.

6. **Unreliability**: The devices within an LLNs tend to be unreliable because of the low power and memory constraints.

7. **Duty cycling:** The traffic in an LLN is usually bursty and therefore the devices remain in sleep (power saving) mode for longer periods of time to save energy.

2.3 Routing in WSNs

There are various channel access control mechanisms in a wireless medium. The control mechanism that defines the rules by which the nodes get access to the shared medium and the right to transmit the data. Design of MAC protocols is of paramount importance as it aims to minimize the collisions among multiple nodes, trying to access the shared channel. Energy consumption and fairness among nodes are also some major concerns in wireless transmissions. The next crucial step is to choose among available paths, the right path for message transmission between a source and destination pair. Furthermore, questions like what are the routing possibilities and how to determine the shortest path are also very important. In this chapter, the basic concepts and concerns about routing in WSN are discussed.

Formally, Routing is defined as a mechanism of finding the best possible path for data transmission upon request between a source and destination pair. In WSNs, the network layer is used to implement the routing mechanisms. Traditionally, routing tables are used to populate the network information. Routing tables list the most suitable neighbours from every node in the network. Nodes choose their next hop according to the available routing information for reliable data transmission. In the case of single-hop networks, the sender node can directly transmit the data to the destination as its next hop. However, in case of large-scale multihop deployments, the destination is not directly reachable via the sender. Therefore, intermediate nodes are used to relay data to the neighbour node until it reaches the destination. The intermediate node has to take routing decision when choosing a neighbour for forwarding the incoming data packet, which is not destined to itself. Efficient routing algorithms are used to populate the routing tables and provide a path for every destination. The formation of routing tables and keeping them up to date is very crucial for both distributed and centralized routing protocols in WSNs. The routing tables basically list the path from a node to reach a destination. How the routing table is built and updated is discussed in this chapter.

Requirements and Challenges

Routing is one of the main challenges in WSNs due to the intrinsic properties of sensors that differentiate them from traditional wireless networks such as cellular networks or mobile ad hoc networks. The existing routing protocols for wireless networks cannot be deployed in wireless sensor networks due to the inherent characteristics of WSNs. For the design of a routing protocol for WSNs the limited processing capabilities of sensor nodes, their memory constraints, the dynamic changes in network topology, and the unreliability of wireless links should be considered. These factors pose several challenges for routing in WSNs. Therefore an efficient and reliable routing protocol must overcome these challenges. When proposing a new protocol, it is always enticing to begin with the protocol specification directly. But without a distinctive understanding of the routing requirements, this becomes certainly a cumbersome task to adjust the protocol as new requirements are included. To avoid such situations, the IETF working group usually defines a requirement document first before standardizing a protocol.

The routing requirements defined below does not specify a link layer to use; they just list the common requirements of the LLN networks.

1. **Support of unicast/multicast/anycast:** Support for unicast, multicast and anycast traffic has been listed in many requirement documents. The support for multicast traffic is exclusively listed in the ROLL working group.

2. **Adaptive routing:** It is very important for routing protocols to adapt paths according to the network changes. As the network conditions (e.g link node/node energy drainage/ node mobility change) the nodes should be able to recompute the routing paths.

3. **Constraint-based routing:** In IP networks, typically routers are not constrained. Core routers have powerful CPUs and several gigabytes of RAM. Although smart objects now have reasonable memory and processing power these are still more constrained than a router. Another crucial constraint

is node energy. Nodes in LLNs are mostly battery operated and therefore energy consumption is a major constraint. The routing protocol for LLNs should be able to support constraint-based routing based on node constraints like energy, CPU and memory, application requirements such as finding a path without main powered nodes or link constraints such as link latency.

4. **Traffic patterns:** A large number of LLNs focus on data collection networks (e.g smart grids) where most of the traffic is from the leaf nodes to a data collector or sink. This kind of traffic is known as multipoint-to-point (MP2P) traffic. Point to multipoint (P2MP) traffic is also often required as the sink or the root node may want to send a command to multiple leaf nodes in the network. Therefore a routing protocol should be able to reliably transmit MP2P and P2MP traffic. A routing protocol should be able to find multiple paths as well in case of bursty traffic.

5. **Scalability:** As explained earlier, LLNs are large-scale networks with hundreds of nodes. The size of the network may increase depending on the application requirements. The routing protocol should be scalable according to network size. The information each node obtains about the network is limited because all the nodes are not reachable to each other. The routing protocol should be fully distributed and it should be able to gather information of all nodes in the network for full connectivity. In the case of dense deployments, the routing protocol should regularize the routing information to increase energy efficiency.

6. **Configuration:** WSNs are deployed in environments where human intervention is not practical all the time or the tasks are very cumbersome for human's e.g monitoring water level in the soil, smart grids etc. Therefore, it is expected that the sensors have a self-healing process and they can adapt to failures without human intervention. Therefore the routing protocol should be able to configure the network with minimal resources. In case, the user wants to deploy a new node in the network, the routing protocol should be able to adjust the configuration automatically. Most of the WSN

are highly distributed. In such large scale and distributed networks, a centralized routing protocol cannot work efficiently. Therefore decentralized routing should be done. The sensor nodes should be able to locally repair a failure. This decentralized approach may not be optimal but it is energy efficient.

7. **Node Behavior:** Due to resource constraints, the nodes in a network can be in the sleeping mode most of the time. The routing protocol must be able to identify the behaviour of the node and the node should be able to act as a proxy. This means that the packet can be delivered to a proxy and that will relay the packet to the destination once it wakes up.

8. **Node Deployment:** Nodes in WSN can either be deployed deterministically or they can be placed randomly. In deterministic deployment, the position of nodes is known. The messages and data packets are routed through pre-determined paths. However, in random deployments, the position of nodes is not determined. During the initial stages of topology formation, the nodes are unaware of the neighbouring nodes and the network topology. The topology also keeps changing due to several reasons during the network lifetime. It is solely the responsibility of the routing protocol to provide routing information and the information of the positioning of the nodes as the nodes are discovered.

9. **Node mobility:** Some applications require mobile nodes for monitoring or sensing purposes. The mobility of nodes introduces changes in network topology and some neighbour nodes become unreachable via other nodes that they were connected to before. An optimal routing protocol should be able to find alternative routes in case some of the nodes are not reachable temporarily. Route instability is an important issue in WSNs and the routing protocol should be able to handle network changes.

10. **Energy consumption:** Energy consumption is one of the major challenges in WSNs. The sensors are mostly battery operated. Their energy drains out quickly while performing computational and routing tasks. Nodes consume a lot of energy during exchange of control messages for topology creation.

Therefore the routing protocol should be energy efficient and it should be able to minimize energy consumption. Routing metrics such as remaining energy of the nodes can be used as a constraint to compute shortest paths.

11. **Robustness:** Routing in WSNs is done in a multi-hop manner. Several sensor nodes have to relay a data packet between a source to destination. Unlike the internet, where dedicated routers are used for routing, the sensor nodes also take part in routing in WSNs. These sensor nodes have power and CPU constraints. Therefore these nodes may run out of battery or an unexpected failure may make the non-functional. The routing protocol should be able to handle node failures and provide repair mechanisms.

12. **Application aware routing:** In WSNs, the support for various applications is very important. Therefore the routing protocol should be able to make a routing decision based on the application requirement. Static routes can be used for monitoring application whereas event based may require an efficient wake up mechanism and lower latency. Similar to the internet, the LLN protocol may require the support of multi-topology routing (MTR) and quality of service (QoS).

13. **Network stability in LLNs:** Network stability is a very difficult task for the routing protocols. There is a trade-off between convergence time and network stability. It is expected that an ideal routing protocol will have fast convergence i.e finding alternate paths, in case of a node failure. There is a serious risk of network instability and routing loops in case of frequent node failures, particularly among distributed routing protocols. This is why the compromise between fast failure recovery and network stability is quite challenging in LLNs.

14. **Security:** Security is an important requirement in many LLNs. Some applications e.g smart homes, temperature sensing etc may not require security but in many cases, e.g industrial automation, smart grid and building automation etc require very strict security. Therefore the routing protocol should have authentication and encryption services.

How to deal with conflicting objectives? Considering requirements from several applications with different constraints is very challenging. The basic approach is taking a combination of all requirements. However, this is not considered as a realistic or desirable approach. Considering the union of all requirements may not be possible because of the constraints on energy consumption and minimizing the complexity of the protocol. In some cases the requirements of various application may be conflicting with each other. Even if all of the requirements are fulfilled by a single routing protocol, it may not be beneficial. Why would a routing protocol for urban networks have to support features of a network operating in buildings? It will be more beneficial to only include the features required for a particular application. It will help to reduce power consumption and computation complexity. An efficient approach can be to design a modular routing protocol. The second approach was adopted by the Routing over low power and lossy network (ROLL) working group as explained in the next section. The aim of the ROLL working group was to design a modular routing protocol in which the core components of the application will be specified by the routing protocol in particular whereas the optional features will only be activated when required.

2.4 IPv6 Routing Protocol for Low power and Lossy Networks (RPL)

The IETF formed a Routing over low power and lossy networks (ROLL) working group to deal with the routing challenges in WSN and specify the requirements of low power and lossy networks. One of the main challenges for the working group was to determine the scope of the protocol to be defined. LLNs greatly vary from each other in contrast with traditional IP networks. For example a mobile delay tolerant network used to study wildlife differs a lot from a dense network for industrial automation. Thus it was decided to limit the scope of the standard to four main applications: home automation, building automation, urban networks (including smart grids) and industrial automation. These applications also represent other networks and it was urgently needed to define a routing protocol for these applications. Thus by designing a routing protocol for these applications should suffice the routing requirements of many smart object networks. The ROLL working group published the RFC 6550 in 2012, which specifies the Routing protocol for low power and lossy networks (RPL). RPL become the defacto routing standard for routing in IPv6 compliant devices. RPL is a proactive IPv6 based distance vector routing protocol for low power and lossy networks. RPL is a source routing protocol that is designed to operate with other link layers like 802.15.4 MAC layer and physical layer. RPL is designed for collection networks (multipoint to point traffic) where a central node coordinates and receives data from the overall network. RPL can also be used for point to point and point to multipoint traffic scenarios. Since RPL is a proactive routing protocol it provides a mechanism to disseminate routing information over the dynamically formed network topology called DODAG. We can think of DODAG as a logical routing topology that is built by routing protocol over a physical network. The mechanism of DODAG construction is explained in the rest of this section. RPL uses trickle timer algorithm to periodically generate control messages in the network.

RPL builds tree like structure called Directed Acyclic Graph (DAG) in which all the traffic from a subtree is directed to a parent node and it forms loop free topology. In case of upward routing, all the traffic is directed towards a single node therefore it is called destination oriented directed acyclic graph (DODAG). A node's rank defines the node's particular position relative to the other nodes in the network with respect to the sink node. The rank of a node is calculated using an objective function. The rank of a node need to be monotonically increasing in the downward direction and strictly decreasing in the upward direction to avoid loops in the network. RPL instance is a set of one or more DODAGs that have the same RPLINSTANCEID. RPL instance is identified by a unique RPLINSTANCEID. There can be multiple DODAGs within one instance that share the same instance id. Each instance uses one objective function. There can be multiple RPL instances of the same network with different objective functions that operate independent of each other. The significance of having multiple instances is to find routes with routing metrics and constraints. For example one instance can be used to find routes for delay constraint applications using minimum hop objective function while the other instance can be used to find routes with better link quality for reliability constraint applications.

There are three types of icmpv6 control messages that are uniquely identified by a code. RPL uses control messages to propagate routing information in the network.

- **Destination information object message (DIO):** The DODAG uses DIO messages to DODAG information for upward routing. It contains RPL instance id, RPL configuration parameters and objective function information that allows a node to select its preferred parent based on the routing metric.
- **Destination advertisement object message (DAO):** The DAO messages are used to maintain downward routes along the DODAG. They can only be propagated once the topology is built by using DIO messages. In non-storing mode, the DAO message is unicast by the leaf node to the DODAG root. In storing mode, the DAO message is unicast by the child node to their preferred parents.
- **Destination information solicitation message (DIS):** DIS messages are used by any node to explicitly solicit DIO messages from the neighboring nodes. It can be triggered by node if that node doesn't receive the DIO message.

Finding shortest possible paths from a source to destination and populating the routing table is one of the most important tasks of a routing protocol. The size of routing table is usually measured by the number of entries in its routing table instead of bytes or Kilobytes. RPL supports P2P and MP2P traffic patterns. RPL uses DIO messages for constructing and maintaining upward routes toward the DODAG root (sink node). Upward routes are constructed to provide multipoint-to-point (MP2P) traffic support. MP2P traffic pattern is a significant feature of data collection networks where a central node collects data from the overall network. This central node can serve as a LLN border router and transmit the collected data to an online storage space or a server. The root node initiates the topology construction by sending multicast DIO messages to other nodes. The nodes close to the root, receive the DIO and select it as their preferred parent since it has lower rank and further multicast DIO to neighboring nodes. Each node has a set of one-hop neighbor nodes called candidate neighbor set. Upon receiving the DIO message from the candidate neighbors, the node selects a set of nodes which have a rank strictly less than the node's own rank according to the objective function. This group of nodes is called as candidate parent set. Finally the node selects one preferred parent from the candidate parent set. Whenever a node wants to send data to the root node, it immediately sends it to the preferred parent. The parent node sends it to its own parent and this process continues until it reaches the DODAG root.

In RPL, downward routes are maintained by DAO and DAO-ACKs messages. Downward routes are required for Point-to Point (P2P) and Point-to-Multipoint (P2MP) traffic scenarios. It is an optional feature of RPL. DAO messages are sent in the upward direction after the topology is built using DIO messages. The routing tables are populated according to one of the modes of operation (MOP) that are described in next section.

RPL supports two modes of operation for supporting P2MP and P2P communication: storing mode and non-storing mode according to the resources. Both MOPs use DAO messages for building downward routes. These two modes of operation are incompatible. A given RPL Instance can either be storing or non-storing. P2P routes are by default incorporated by having the sensor node transmit the data packet via its preferred parent all the way up to the DODAG root. Once the root receives the message it transmits it to the destination either by appending the source route to the data packet or by simple hop-by-hop routing down the DODAG. This depends on whether the mode of operation (MOP) is set to be storing mode or non-storing mode. Fig. 2.1 shows an illustration of storing and non-storing mode. In storing mode, all nodes maintain a routing table with entries of all nodes in the tree. The size of the routing table depends on the size of network. For P2P communication, the packets travels up until it reaches a node which is an ancestor of the destination. The packet is then directed downwards by

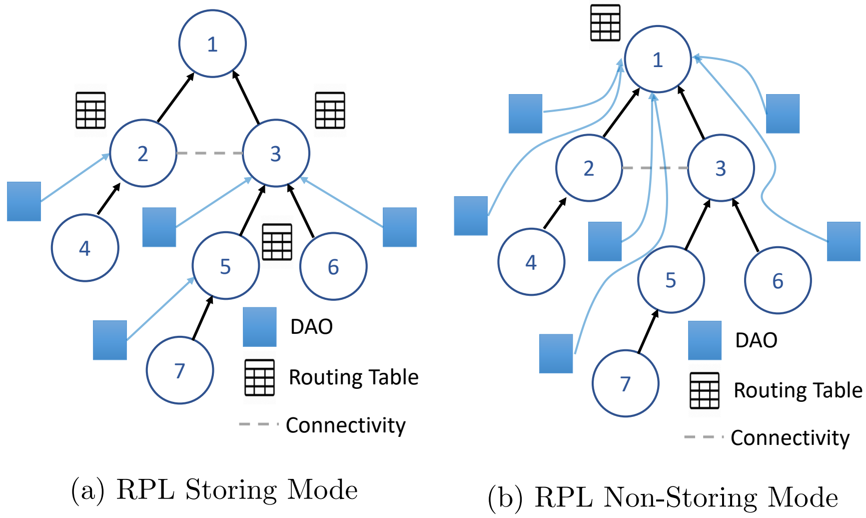


Figure 2.1: RPL Storing and Non-Storing Mode of Operation

the ancestor node. In storing mode, each node checks its routing table to decide its next hop for downward routing. In non-storing mode, the DIO parent nodes do not store the routing table. The DODAG root stores the routing information of the overall DAG in its routing table. In non-storing mode, a node sends data all the way up to the DODAG root by recursively passing on the messages to DIO parents. At the DODAG root, the packet is source-routed to the required destination. Source routing is used to route packets in downward direction. The routing information is contained in the source routing header.

An objective function (OF) defines the rules for calculating the rank of node, selecting the parent nodes and optimizing the routing paths based on the some routing metrics and constraints. The significance of objective function can be explained by considering a physical network with several links and different link qualities such as latency, reliability and different type of nodes like main-powered or battery operated. If the network runs two different type of applications then it might be useful to use links according to the application requirements. For example one of the objective functions can be used to achieve lower latency while other one can be used to have high reliability. ContikiRPL implements two OFs as described by IETF i.e. Objective function 0 (OF0) [18] and Minimum rank hysteresis objective function (MRHOF) [19].

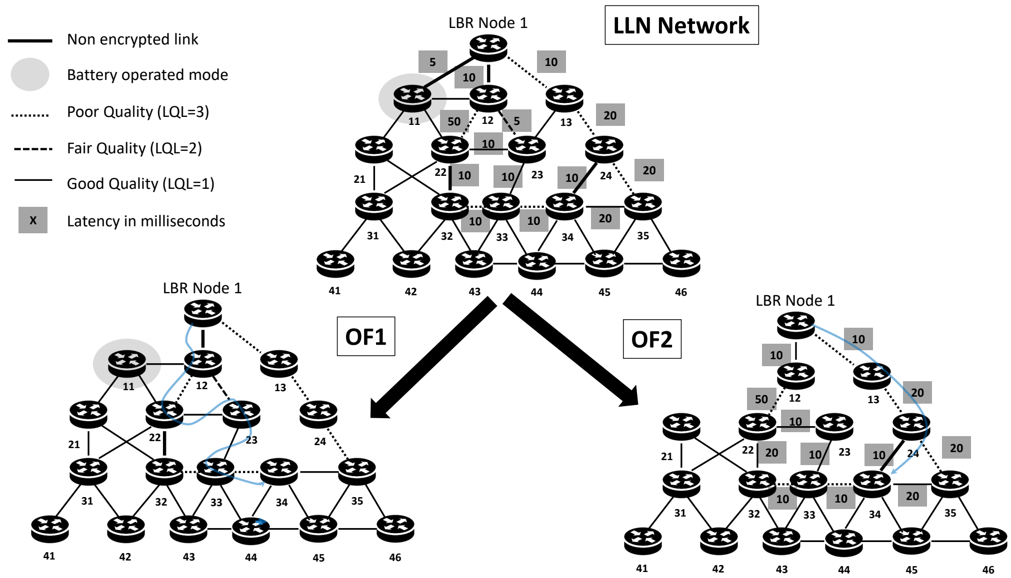


Figure 2.2: An illustrantion of DODAG with two different objective functions [2]

1. **OF0:** OF0 uses hop count as routing metric.

2. **MRHOF:** MRHOF selects routes that minimize a metric and use hysteresis to reduce churn in the network in case of small changes in the network. MRHOF uses additive routing metric. Expected transmission count (ETX) is one of the traditional routing metrics. It is possible for a node to join multiple DODAGs based on the application requirements and mark the traffic according to the objective functions specified. Fig. 2.2 shows two DODAGs built on two different objective functions.

Routing metric is a quantitative value that is used by a routing protocol to find shortest path with minimum cost. It helps in making routing decision. Traditional routing protocols like Open shortest path first (OSPF) and IS-IS use static routing metric like link bandwidth or a linear combination of some metrics. However, LLNs require dynamic link matrices as well as static link metrics because of the wide variety of applications and resource constraints. Also, both link and node metrics are required. To understand the significance of dynamic routing metrics and constraints for LLNs, let's consider two network scenarios.

1. An application requires lower latency and quick delivery of packets to the

destination. Therefore ETX will be used to find shortest paths with minimum end-to-end delay.

2. A network may include some resource constrained nodes. Therefore the objective will be to find paths that traverse only main-powered nodes and avoid battery operated nodes. In the first example ETX is used as routing metric whereas in the second scenario battery-operated nodes can be used as constraint to exclude from the routing path. Contrary to traditional wireless networks, Node metrics are also used in LLNs.

Two commonly used routing metrics are hop count and ETX.

1. **Hop Count:** Hop count determines the number of intermediate nodes between a source and destination pair. A hop count of 4 indicates that there are 4 intermediate links between the source and destination.

2. **Expected Transmission Count (ETX):** ETX is the measure of quality of link between a sender and destination pair. ETX of a wireless link is the expected average number of transmissions required to successfully send a data packet and receive ACK frame over the same link. The path ETX is the sum of the ETX of all the links along the path. ETX varies from zero to infinity. An ETX of one indicates a perfect link.

2.5 Related Work on RPL

There are two aspects of my work. One is the load balancing in dense WSNs and the other aspect is congestion mitigation in high data rate applications. Each of which has its own representative body of literature. The core of this work is the use of multipath routing in wireless sensor networks for high data rate applications in which single path routing is not feasible because of congestion in the network. The main contribution of my work is to introduce multiple paths for load balancing in case of congestion in the network. A lot of work has been done on performance evaluation of RPL in Contiki [20]. In order to highlight the load balancing and congestion problem of RPL and its impact on the network performance, several studies have been performed on performance analysis of RPL [21]. In [22], the authors performed the performance analysis of RPL in contiki and showed that most of the packet losses occur at the node level due to congestion in the network. In [22], the authors investigated the load balancing problem of RPL. They showed that most of the packet losses in a network are due to congestion and ETX cannot be used to detect congestion in the network. They performed the experiments in an indoor testbed and evaluated it in TinyRPL. A number of studies [23], [24] have shown that RPL has a load imbalance problem as it finds paths with good link quality which leads to uneven load distribution and congestion at some bottleneck nodes. Therefore these studies show that congestion has undeniably significant impact on network performance. Thus congestion control mechanisms should be proposed.

Increasing advances in WSN applications have revealed various issue of routing in WSNs. Researcher's are giving attention to these issues in order to improve the performance of routing protocols in WSNs. Some critical wireless sensor applications result in huge data load which leads to congestion in the network due to limited storage capacity. Congestion occurs because of two reasons i.e., link collisions and buffer overflows. When several sensor nodes transmit data at the same time at high rate to a single relay node. Basically the node received more packets than it can froward and the incoming traffic rate overcomes the buffer capacity and the buffer overflows. This overflow causes congestion thus increasing packet loss and decreasing the throughput of the wireless channel. Therefore, congestion is one of the most critical issues to deal with in WSNs. Indeed, handling and controlling congestion in WSNs is considered to be a remarkable research gap which has attracted researcher's attention. Recently, many studies have been done on the performance analysis of RPL. These studies show that RPL suffers from unbalanced load distribution and congestion in the high data rate applications. Load imbalance and buffer overflow are the main causes of congestion in network running RPL.

Multipath routing has been widely used in literature for balancing the load and improving quality of service for both wireless sensor and ad hoc networks. Multipath routing can be used to achieve multi-fold objectives, including higher reliability, increase in throughput, fault tolerance, congestion mitigation and hole avoidance. The main idea of multi-path routing is to provide alternate paths for information to reach the destination. A node saves alternate paths to the sink node and then uses them to distribute the traffic load. In order to support high data rate and maximize network lifetime, RPL can be enhanced to support multipath routing.

Multipath routing reduces the probability that the communication is disrupted in case a node runs out of energy or a link fails. A lot of research has been done on load balancing and multipath routing in RPL. In [25] the authors have implemented multipath using opportunistic routing over IEEE 802.15.4. They consider the RPL implementation on top of IEEE 802.15.4. They forward data packets to multiple nodes opportunistically instead of using single preferred parent for relaying data towards root node. This implementation supports delay-sensitive applications like alarms that need to be delivered to the root node before the deadline. This approach provides somewhat better results than RPL in terms of end-to-end packet reliability (PDR) and delay. In [26] three multipath routing schemes were proposed using energy as a routing metric, local repair mechanism and a combination of them. They have modified the IPv6 stack in OMNET++ simulator and integrated these schemes in IPv6 stack for IoT applications. Traditional RPL uses single path for upward routing. They provide multiple paths by considering nodes with same rank to forward data. They utilize nodes with same rank In the event of node failures also. In particular the neighbor/ sibling nodes have the same rank. They have used residual energy to switch routes among parent nodes and use neighboring node with the same rank in case of local repair. Their approach has lower overhead and higher packet delivery ratio as compared to RPL. In this way they have achieved load balancing and fast local repair.

Multipath opportunistic routing ORPL is also proposed in [27], which supports any-to-any routing and forward traffic by just considering the information of subtree. This implementation supports applications like building automation and smart cities that not only send data upward to the sink node but also send from one node to another node (Point-to-point) for coordination purposes. Point-to-point traffic in RPL is supported by first sending the data upward along the gradient to a common ancestor and then downward using hop-by-hop routing or source routing. They have proposed anycast routing for downward traffic over MAC duty cycling. Any node that wakes up first and is closer to the root, receives the packet, acknowledges it and forwards it. This implementation has lower latency, robust and interoperable as compared to RPL.

Tan et.al proposed multipath routing using a cost function that considers the remaining energy of nodes and the hop count [28]. The path discovery process finds two paths between source and destination reactively that minimize this cost function. They also consider the interfering nodes and exclude them from the multipaths. This approach reduces the interference in the wireless medium. The algorithm is implemented in NS2 simulator. They have reduced energy consumption in the network and increased network lifetime. However they do not consider the convergecast traffic pattern.

Marwa et.al [29] used the spanning tree algorithm on RPL protocol to do load balancing in the network (MD-RPL). The algorithm modifies tree formation by minimizing the degree of the tree formed by RPL to achieve load balancing. They have used routing metric and the node rank as parameters.

Boubekeur et al. [30] proposed BD-RPL which limits the size of the subtree of each node in the network to relieve congestion. They have defined a threshold for the size of subtree and nodes are allowed to select a particular parent only if its subtree size is smaller than a predefined threshold. The authors have evaluated their proposed implementation against ContikiRPL through simulations and testbed experiments.

Liu et al. [31] proposed a load balancing RPL called LB-RPL to handle the load unbalancing problem in RPL. Thy exploit the queue utilization information of a node to achieve load balance. The queue utilization value is detected from how the delay in DIO. Therefore the node that is congested, delays its DIO transmission. A node probabilistically selects its parent node for each data packet transmission by using the queue utilization information of its candidate parent set. Workload balance is obtained while making routing decision. More data is transmitted to nodes with lighter workload and less packets are sent to nodes with heavier workload. The detection of queue utilization information via DIO is problematic because of DIO packet losses and DIO reception time do not exactly reflect the queue utilization because of trickle timer. Also, it has a slow recovery process as the transmission interval of DIO is very long. This can also have additional overhead by always eliminating the congested parent node from the parent set. The authors evaluated their work in NS-2 simulator. Lodhi et al.[32] designed M-RPL using Contiki OS and COOJA simulator. They detect traffic congestion through queue utilization and provides alternate parent nodes for each node to distribute traffic load.

Several other works evaluate schemes through testbed experiments in congested scenarios. To address the congestion issue and load imbalance, kim et al. [33] proposed Queue utilization based QU-RPL.Every node selects their parent node based on queue utilization routing metric of its neighboring nodes and their hop count to the location border router. They have provided more insights about their work on another test best in [24]. They performed all the experiments during night-time to focus on load balancing effect They did all experiments during the night-time to give attention to load balancing effect rather than link dynamics and therefore the behavior of RPL and QU-RPL in fluctuating link environments is still not studied.

Al-Kashoash et al. addressed the congestion problem of RPL and proposed congestion aware Objective function (CA-OF) in [34]. They used buffer occupancy as a routing metric and introduced a new RPL objective function called congestion aware objective function (CA-OF). Since ETX does not consider node congestion. CA-RPL tries to mitigate congestion by avoiding congested paths and selecting the paths with least delay. However, the proposed method may result in route instability.

Chapter 3

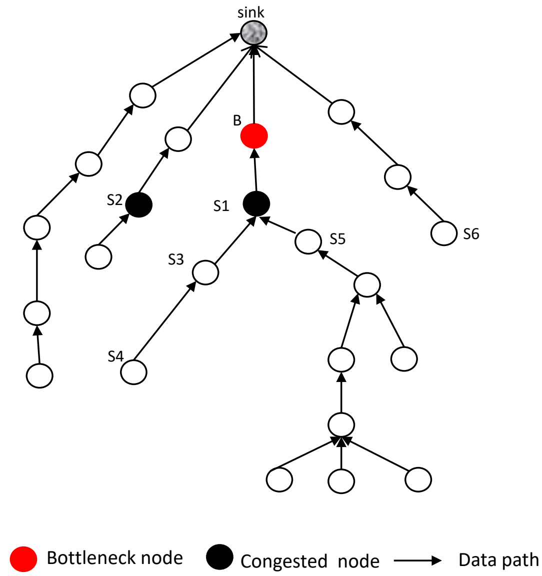
QU-MRPL and Implementation

3.1 Algorithm

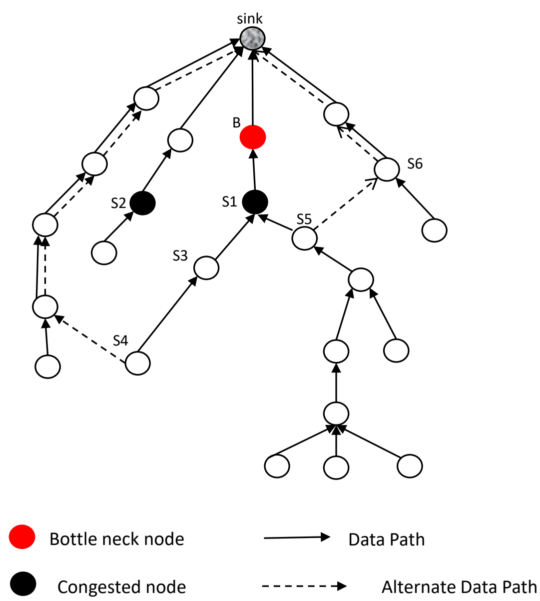
We exploit the average queue size of preferred parent to detect congestion in the network. We introduce multiple paths for load balancing and to mitigate congestion in the network. Basically, following steps have been taken (1) inform the child node about the average queue size (QU) value via DIO (2) Check If the queue is above the threshold (3) Do multipath routing to mitigate congestion

Queue Utilization Based Congestion Detection

Every node informs its child node about its average queue size information via DIO control messages. DIO messages are transmitted using trickle timer algorithm to attain balance between control messages overhead and fast recovery in case of inconsistencies in the network. Generally the DIO contains the RPL version number, RPL instance ID, DODAG Preference, DODAG identifier, mode of operation (MOP), rank, the objective function and an options field. We use the reserved field of the DIO messages to multicast the queue utilization information of the nodes. We take the queue size of a node at the arrival epochs of every



(a) An example topology with unbalanced trafficload distribution in sensor networks. The dark solid nodes are the congested nodes as they have more data packets to froward than the other nodes



(b) An illustrative example of packet forwarding of a sensor node according to multipath routing in QU-MRPL. S5 selects S6 as an alternate parent

Figure 3.1: An illustrative example of RPL and QU-MRPL

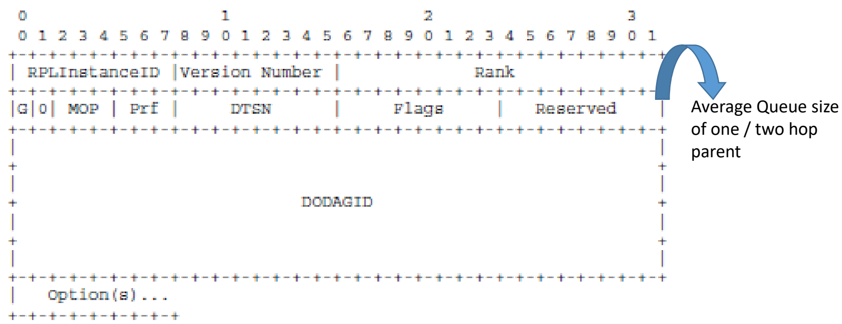


Figure 3.2: DIO Base Object

data packet and increment it by one. Every node calculates its average queue size according to the algorithm 1 and multicasts it to the child nodes when the trickle timer expires. Once the child node receives the parent's average queue size, it processes it and checks whether the average queue size is above a threshold or not. If the average queue size is above a predefined threshold, it indicates congestion in the network. We let the parent node inform its average queue size information for up to two-hop child nodes by using the reserved byte of the DIO message.

Fig. 3.2 shows the DIO message base format. It is important that when we indicate congestion in the network and implement multipath routing. To alleviate this problem, we set the threshold t to 85% in our simulations. When the queue of a node is above the threshold we trigger multipath routing.

The child nodes select a second parent based on ETX as a potential next hop for forwarding some of its data. Since we have an unbalanced DODAG, it is possible that there is congestion on neighboring nodes as well. Therefore we check the queue utilization information P_{i2} of the neighboring nodes as well. If there is no congestion on the neighboring node that we select as alternate parent to forward some of the data, the nodes sends *alpha* percent of data packets to its parent P_{i2} and rest of data to the preferred parent. C_{max} is the maximum queue size of a node

$$\alpha = 1 - \left(\frac{1}{C_{max}}\right) * P_{i2} \tag{3.1}$$

In Fig. 3.1 when S3 and S5 receive their parent's queue size they start multipath routing. S5 selects S6 as an alternate parent and sends a fraction of its traffic to S6. S3 selects S2 as an alternate parent but S2 also suffers from congestion. Therefore it sends its parent's queue size information to own child node S4 so that it can start multipath routing.

Algorithm 1 Queue utilization calculation procedure

1: Initialize queue utilization counter QU_{count} and Packet counter $P_{ktcount}$
2: when a data packet P_{kt} arrives, check the packet destination address
3: **if** packet destination address is not equal to current node's address **then**

   a. Increment received packet counter

   b. Obtain current queue utilization value

   c. Calculate cumulative queue utilization value $QU_{cumcounter}$

   d. Calculate average queue utilization before trickle timer expires

4: **end if**
5: When trickle timer expires, multicast a DIO message with average queue utilization value of parent node to the child nodes
6: Reset QU_{count} and $P_{ktcount}$

Algorithm 2 QU-MRPL Queue utilization based Multipath RPL

```
A source node listens to the radio channel
When a message is received, check the type of the message
if A DIO message is received then
Check the reserved bit of the DIO message to get the
parent's average queue utilization information
    if average queue utilization value carried in the
message is the maximum value then
        Choose a second parent node form the parent table as a next hop for
forwarding some of the data
    end if
else if A DAO message is received then
    Process it according to RPL
else if A Data message is received then
    Invoke queue utilization calculation procedure
    Forward this data message on the default route if there
is no congestion on the default route else use multipath
if the queue utilization counter has maximum value.
end if
```

3.2 Performance metrics

We have used three performance metrics for qualitative analysis and performance comparison of RPL and QU-MRPL. The first metric is cumulative throughput. The second metric is PDR. Third one is latency and the fourth one is energy consumption.

Cumulative Throughput: In this metric the aim is to compute the number of bits per sec delivered to the sink node. This value is computed as follows:

$$\text{Throughput} = \frac{\text{received packets} * \text{MTU} * 8}{10 * 1000}. \tag{3.2}$$

Packet Delivery Ratio: PDR is the ratio of number of packets delivered to the number of packets sent to the sink node. PDR represents link quality since the number of packet drops increases in case of a bad quality link. PDR is calculated as:

$$\text{PDR} = \frac{\Sigma(\text{Number of received packets})}{\Sigma(\text{Number of sent packets})} \tag{3.3}$$

Latency: Latency is the different between the packet generation and its reception at the sink node. It is considered only for delivered packets. Latency is calculated as:

$$\text{Latency} = \text{Packet generation time - Packet arrival time.} \tag{3.4}$$

3.3 Operating systems for WSNs

The base that combines together the upper and lower layers of a protocol stack is called the operating system. Since RPL is the standard routing protocol for WSNs, it has been implemented in many operating systems. The choice of OS greatly effects the performance of a routing protocol. Contiki OS is the most widely used operating system for WSNs. We have implemented our algorithm in Contiki and simulated in its COOJA simulator.

Contiki (Operating system):

The Contiki operating system is an open source operating system for networked embedded systems in general, and smart objects in particular. It provides both the networking implementation as well as application support for motes [13]. Contiki was the first operating system that enabled uIP TCP/IP communication stack for IP communication. The world's smallest IPV6 communication stack – uipv6 was incorporated in Contiki in 2008.

Contiki provides mechanisms for programming smart object applications and assisting communication between them. Contiki system and its applications are developed in C programming language, making it highly portable. The application running Contiki OS can directly communicate with other IP-based applications, web service and internet-based services because Contiki Contains IP-stack. Contiki provides instant Contiki. Instant contiki is a linux based software implementation that provides all the necessary features for software development and simulation. Instant Contiki runs on virtual machine. Contiki has a number of features including radio duty cycling, ContikiMAC, IPV6 stack and COOJA simulator. COOJA simulator as explained in the next section, has MSP emulator, which makes development and debugging easier for the developer. Contiki provides implementation of various IETF standards for smart objects including IETF 6LowPAN, IETF RPL (Routing protocol for low power and lossy networks) and IETF COAP. Contiki operating system provides cross layer implementation which means there are separate modules for every layer of the protocol stack for example the routing module is present in a separate directory "contiki/core/net/rpl" that provides complete features of the IETF Routing protocol for low power and lossy networks.

3.3.1 ContikiMAC

The MAC layer provides mechanisms for channel access and allocation. Contiki OS Medium Access Control Protocol takes care of channel access mechanisms for wireless networks [3]. They can be interpreted as rules that coordinate when each node is going to transmit/receive packets. ContikiMAC is an implementation of Carrier Sense Multiple Access (CSMA) protocol. CSMA is based on Carrier Sensing for detecting medium activity and are prone to collisions and lower efficiency. CSMA protocol keep a list of packets to each of the neighbors and calculate statistics such as number of re-transmissions, collisions, deferrals, etc. CSMA uses Clear Channel Assessment (CCA) to check whether the channel is available for transmission or not. Fig. 3.7 shows the timeline of data transmission and reception.

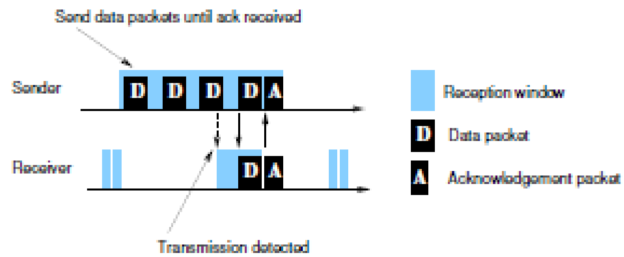


Figure 3.3: ContikiMAC [3]

3.3.2 ContikiRPL

The RPL directory in the net folder of Contiki has several files for the implementation of RPL in Contiki. The functionality of each file is described below:

Table 3.1: Contiki OS files

File	**Functions**
rpl-dag.c	Contains functions for initializing and maintaining DAG
rpl-of0.c	Contains implementation of objective function zero with hop count routing metrics
rpl-mrhof.c	Contains implementation of minimum rank hysteresis objective function with ETX routing metric
rpl-icmp6.c	Contains the implementation of ICMPV6 control messages
rpl-ext-header.c	Contains the extension headers for implementation of RPL in Contiki
rpl.c	It has functions for maintaining routes once the DAG is created.
rpl-timers.c	It has the implementation of trickle timer algorithm and has parameters for sending updates.

3.4 Modification in Contiki 2.7 for multipath routing using queue utilization

When the node receives a DIO message, it calls rpl_dio_process function from the rpl_dagċ file to check the DIO information. If the DIO is not received from the same node before , it is added in the candidate parent table. Now the rpl_find_parent function is called to find the parent for the node receiving the DIO. Now if the DIO receiver node selects the DIO sender node as a preferred

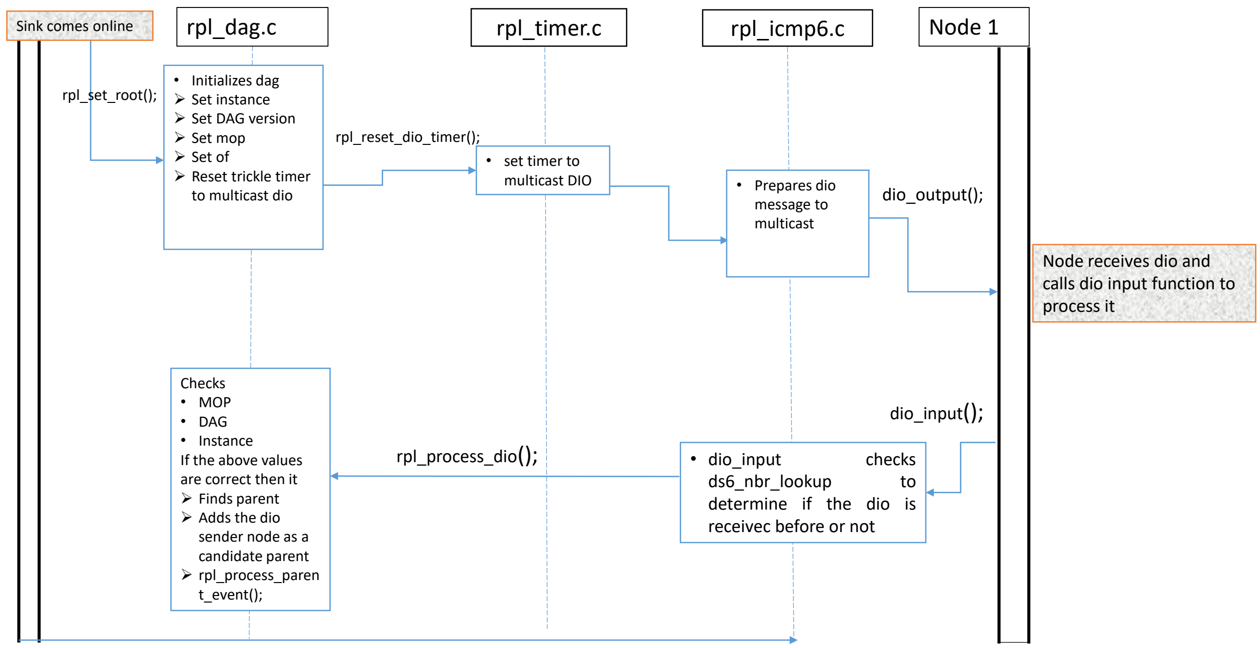


Figure 3.4: A demonstrating scenario of ContikiRPL

parent, we save it's average queue occupancy information in the parent_avg variable.

If the DIO sender is not selected as a parent, we check the neighbor table to find an alternate parent node. The alternate_nbr pointer checks if the DIO is received from the neighbor node. If true then it saves the average queue occupancy information in the alternate_parent_avg variable.

Modifications in RPL control messages

To disseminate the average queue information of the parent nodes we have utilized the two reserved bytes of the DIO message. When the dio_input function in the rplicmp6.c is called, it processes the incoming message and stores it in the DIO. When the dio_output function is called it extracts the average queue information of the node and checks if the average is above a threshold

Modifications in RPL routing files

One of the problems with RPL is that it has single path for upward routing. The solution we have proposed is multipath routing in case of congestion in the network. We have made modifications in the uipds6route.c file for finding the alternate route. The function traverses the neighbor table to find the alternate parent node in case of congestion in the network.

3.5 COOJA Simulator

Contiki OS provides COOJA simulator. The simulator is implemented in Java. COOJA provides emulation of sensor motes and allows the sensor nodes software to be written in C. COOJA allows simulations at three different levels simultaneously: Network level, Operating system level and Machine code instruction level.

The following tools have been used during the development of this thesis:

1. Instant Contiki: It is a Ubuntu image of Contiki OS with all tools pre-installed.
2. VMware Player: Instant Contiki image has been mounted in VMWare Player.
3. Wireshark: It is a network protocol analyzer tool. It is used a packet sniffer.
4. MATLAB: It is used for satistical analysis and to generate plots.

COOJA provides several plug-ins and tools for saving the various network features. Some of the ones that we have used in my simulations are explained below.

- **Simulation visualizer**: It gives options for visualizing network traffic, radio environment, grid and mote types.

- **Time line**: It provides timeline of data transmission, turnaround time ,ack time and shows if there are any collisions.

- **Radio logger**: It captures the network packets in a pcap file.

- **Moteoutput**: The moteoutput window displays any print outs from the motes. It can be very useful to check the data flow and other messages.

Fig. 3.5 shows a snapshot of COOJA being run on Instant Contiki 2.7. COOJA stores the simulation in an xml file with extension 'csc' (COOJA simulation configuration). This file contains information about the simulation environment, the nodes and its positions,random seed and radio medium.

The Radio logger tool of COOJA generated pcap files. Several tools are available for analyzing and interpreting the network behaviour. We have used Wireshark to analyze pcap files. Wireshark is one of most popular and widely used tool for displaying IPv6 files. Fig. 3.6 shows a capture of wireshark which includes the RPL protocol and is properly interpreted.The displayed figure shows the set of message exchanged between nodes. The type column shows ICMPv6 control messages, IEEE 802.15.4 packets and other control messages.

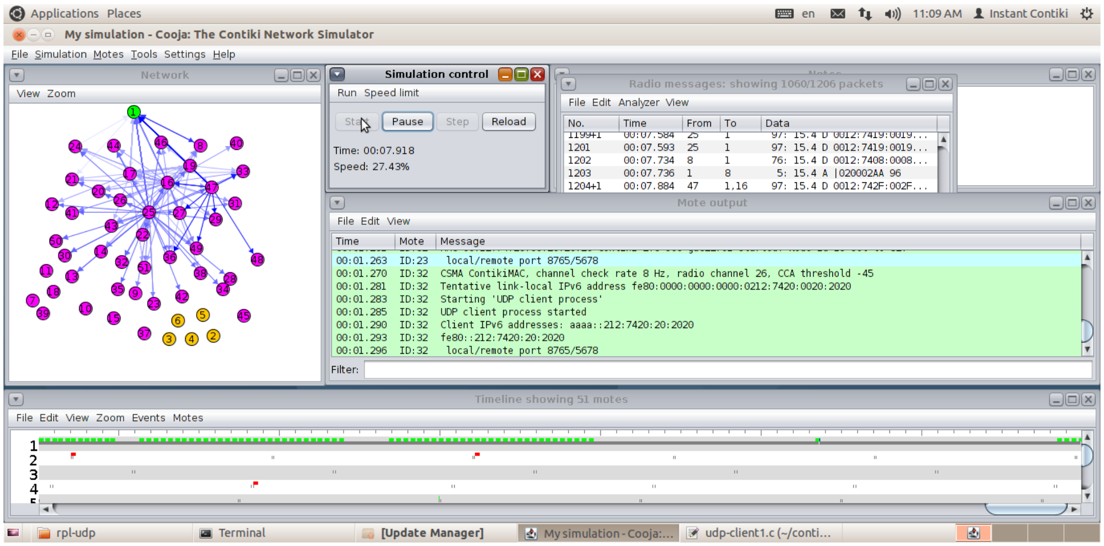


Figure 3.5: A snapshot of COOJA

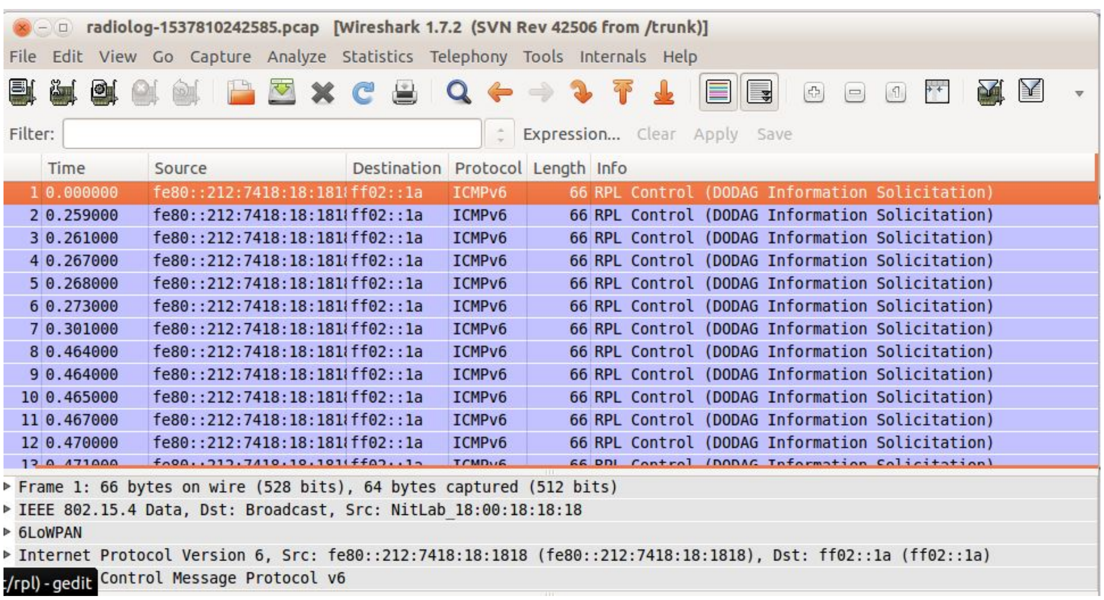


Figure 3.6: A snapshot of Wireshark

3.5.0.1 Trickle Timer

The Contiki OS uses trickle timer [35] for RPL control messages dissemination. The trickle has three tuneable parameters. DIO Interval Minimum represents the initial interval for DIO message dissemination. When the RPL network starts the DIO's are transmitted at a rate I_{min} and it is doubled each time the DIO is fired until it reches the upper limit I^{max}. I_{min} is calculated as

$$I_{min} = 2^{DIOIntervalMinimum} \tag{3.5}$$

DIO Interval doubling represents the number of times the timer can be doubled until it reaches the maximum value I_{max}. I_{max} is calculated as

$$I_{max} = I_{min} * 2^{DIOIntervalDoublings} \tag{3.6}$$

In the Contiki OS, the default value of DIO Interval Minimum is 12 and DIO Interval Doubling is 8. In Fig. 3.7 we have calculated the DIO messages overhead with variations of Trickle timer.

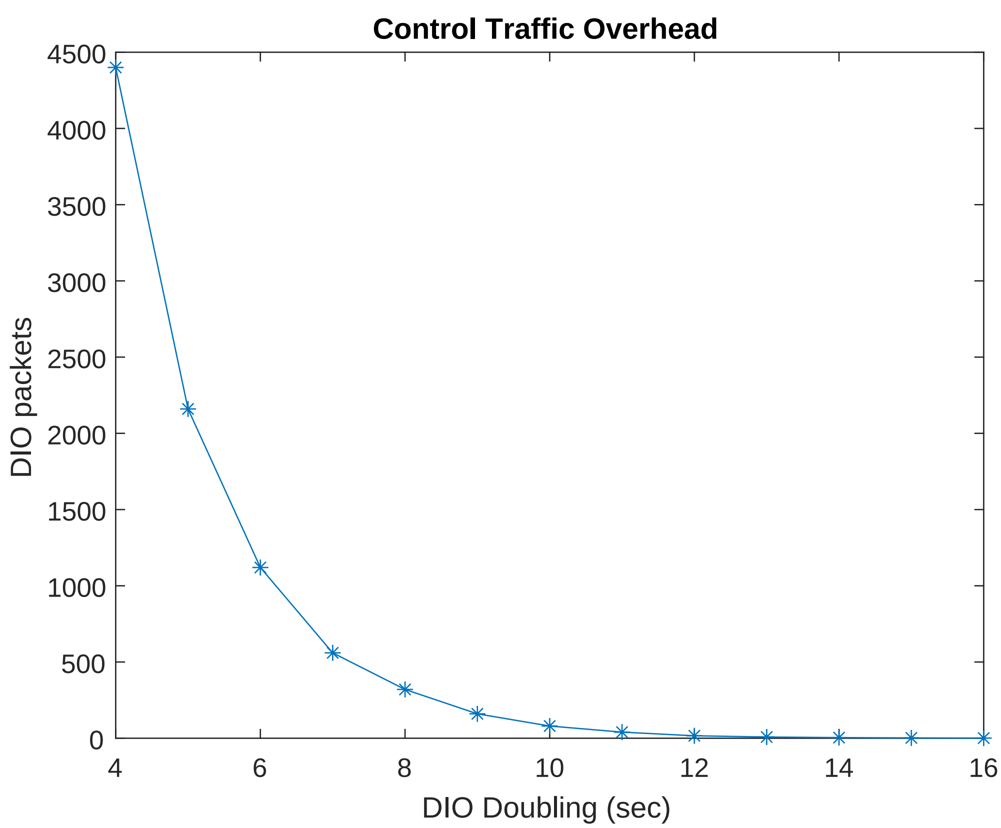


Figure 3.7: Control Overhead

3.6 Simulation environment

The simulation parameters are shown in Table 3.2. COOJA provides implementation of various wireless channel models such as Unit Disk Graph Medium (UDGM), Directed Graph Radio Medium (DGRM) and Multi-path Ray tracer Medium (MRM). We have used Unit Disk Graph Medium (UDGM) with Distance Loss as a propagation model. This model considers two parameters :interference among nodes and success rate of transmission and reception. If there is interference among nodes, the packets are lost. In UDGM the transmission range is modeled as a disk. All nodes inside the disk are in the radio range of each other and they can transmit and receive packets. We use the sky platform which is an emulation of TelosB mote [36]. There can be two types of congestion in the network. One in which the nodes transmit data at a very high rate but there may be or may not be congestion in the network. The second scenario is the one in which the traffic rate is very high such that congestion is unavoidable. We aim to analyze the former case. In order to analyze the performance and behavior of network under congestion we have used different data rates i.e 30,40,60 and 120 packets per minute (ppm). The DIO Interval Doubling is set to 4. As shown in Fig. 3.7, the control overhead is 4400 packets per hour which is very less as compared to the data traffic of all nodes in the network. For the performance analysis, we simulated two topologies as shown in Fig. 3.8 and 3.9. In topology 1, the sink node is placed at the top. The topology is established in a way that upward routing works in a multihop manner. In topology 2, the sink is placed in the center.

Table 3.2: Simulation Parameters Used in COOJA

Parameters (Units)	Value
Topology	Random/Grid
Area (m)	100*100
Source nodes	50
TX range (m)	50
Interference range (m)	75
Packet size (bytes)	30
Initial Energy of sensor nodes (Joule)	4
Packet transmission power consumption (Watts)	0.0174
Packet reception power consumption (Watts)	0.0188
MAC layer	IEEE 802.15.4
Network layer	RPL / QU-MRPL
Communication model	UDGM
Data rate (packets per minute)	30,40,60,120

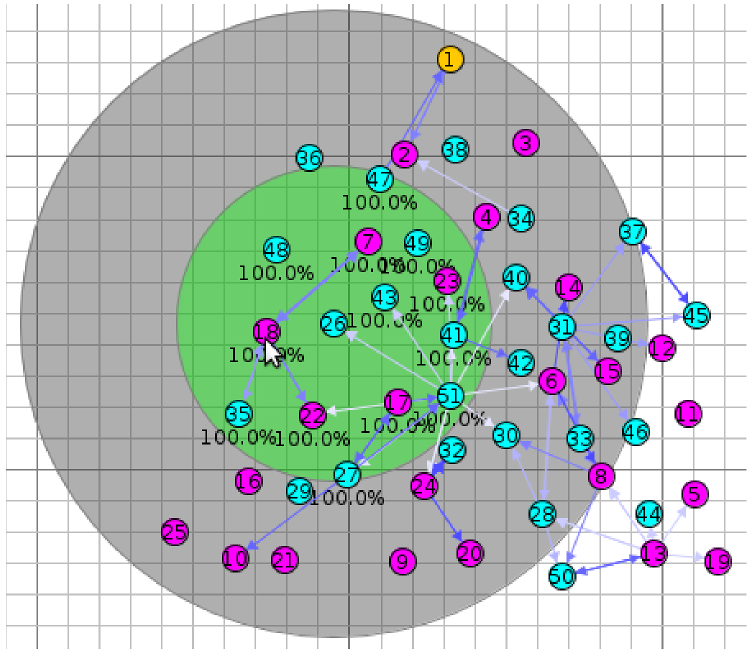


Figure 3.8: Sample RPL network topology 1 built in COOJA. The node 1 is the sink node while all other are source nodes which transmit data to sink node. The arrows represent the flow of radio messages. The green circle represents transmission area and grey circle shows the radio collision region. The percentages represent the reception ratio.

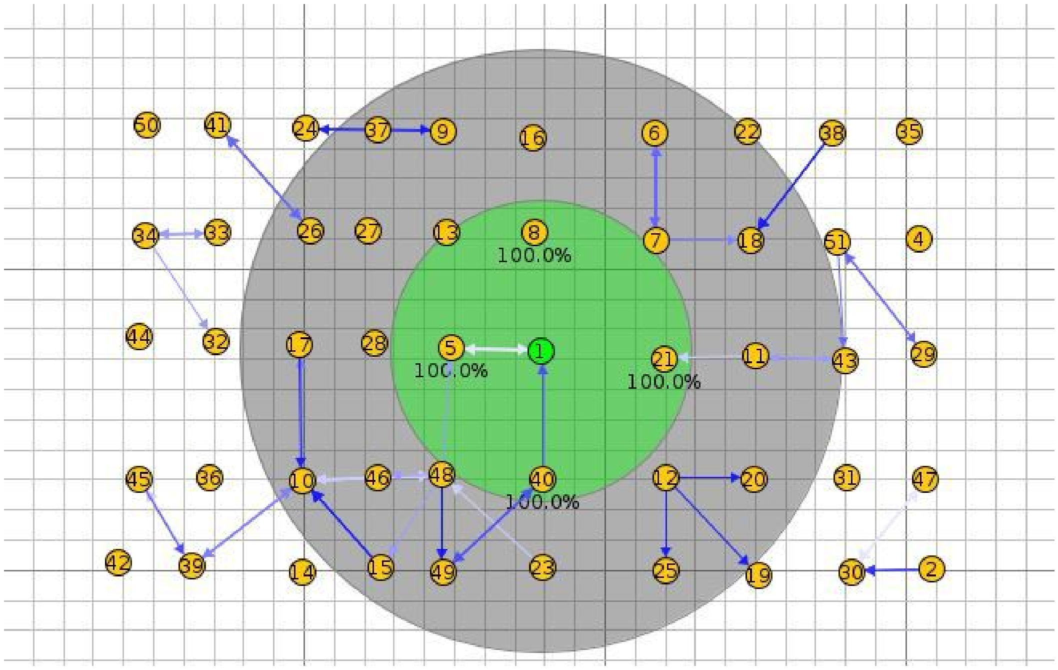


Figure 3.9: Sample RPL network topology 2 built in COOJA. The node 1 is the sink node while all other are source nodes which transmit data to sink node. The arrows represent the flow of radio messages. The green circle represents transmission area and grey circle shows the radio collision region. The percentages represent the reception ratio.

Chapter 4

Results

The cumulative throughput, packet delivery ratio (PDR) and control overhead results for topology 1 and topology 2 are summarized in this section. We have proposed a generalized approach for congestion mitigation that can be applied with both objective functions, i.e., OF0 and MRHOF.

4.1 Topology 1

In this section, the results for topology 1 as shown in Fig. 3.8 are summarized for both objective functions.

4.1.0.1 Cumulative Throughput Results

The cumulative throughput is calculated according to (3.2). Fig. 4.1 shows the cumulative throughput of OF0 and Fig. 4.2 shows the cumulative throughput for MRHOF with data rate of 120 ppm. In both graphs the QU-MRPL performs better than RPL most of the times.

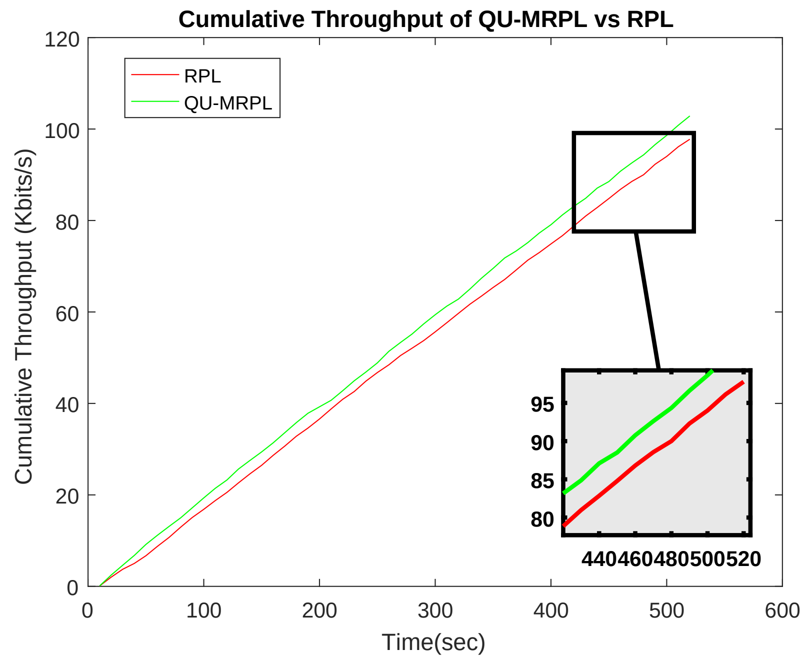


Figure 4.1: OF0: Cumulative Throughput of RPL vs QU-MRPL in Topology 1 with data rate 120 ppm

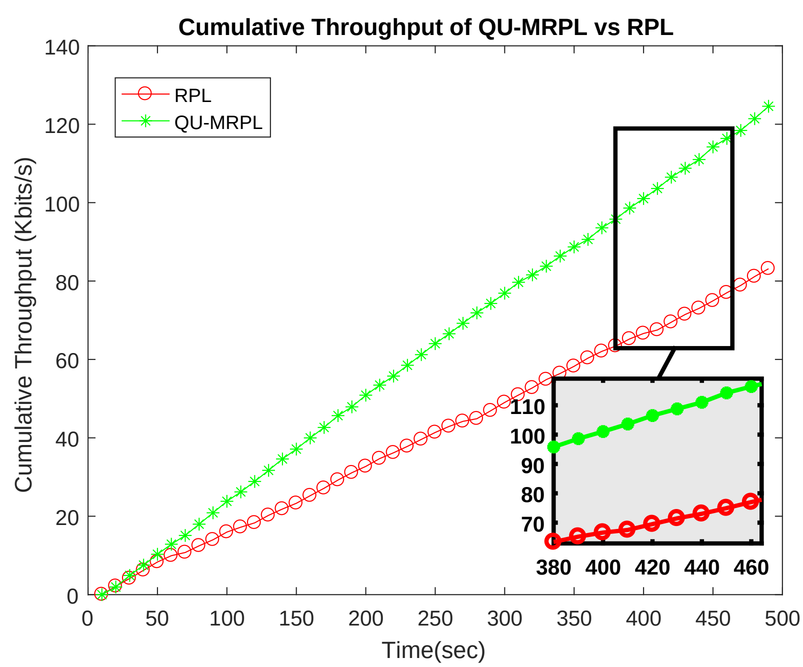


Figure 4.2: MRHOF: Cumulative Throughput of RPL vs QU-MRPL in Topology 1 with data rate 120 ppm

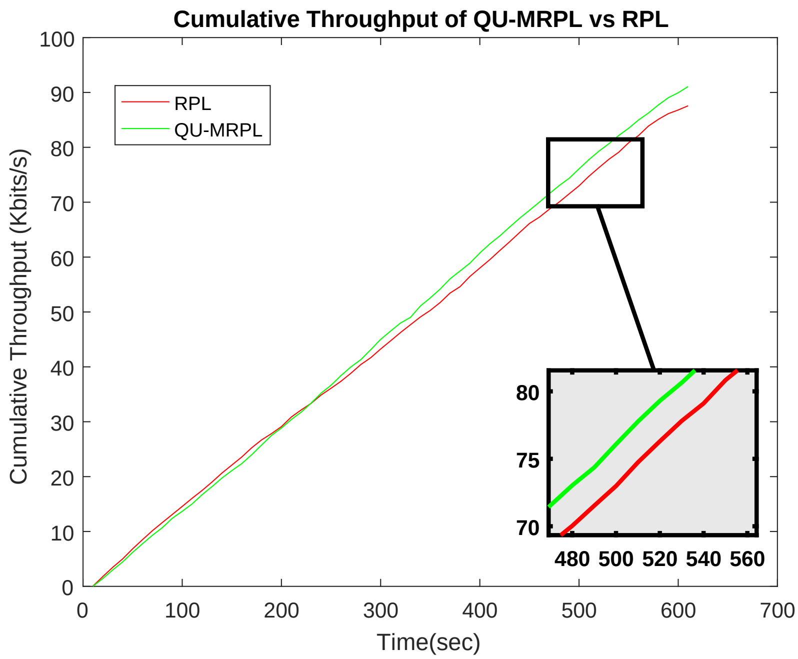


Figure 4.3: OF0: Cumulative Throughput of RPL and QU-MRPL in topology 1 with data rate 60 ppm

Fig. 4.3 shows the cumulative throughout of OF0 and Fig. 4.4 shows the cumulative throughput for MRHOF with data rate of 60 ppm.

Fig. 4.5 shows the cumulative throughout of OF0 and Fig. 4.6 shows the cumulative throughput for MRHOF with data rate of 40 ppm.

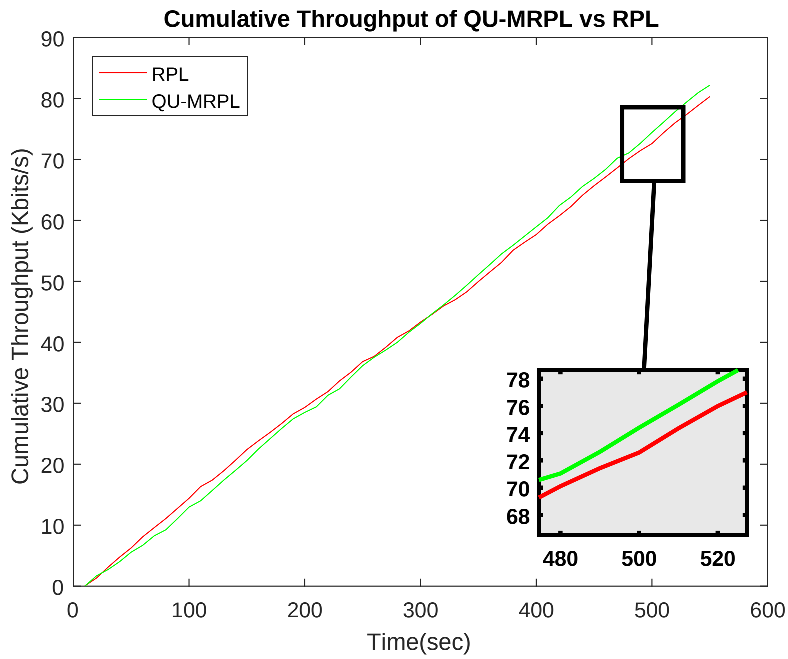


Figure 4.4: MRHOF: Cumulative Throughput of RPL and QU-MRPL in topology 1 with data rate 60 ppm

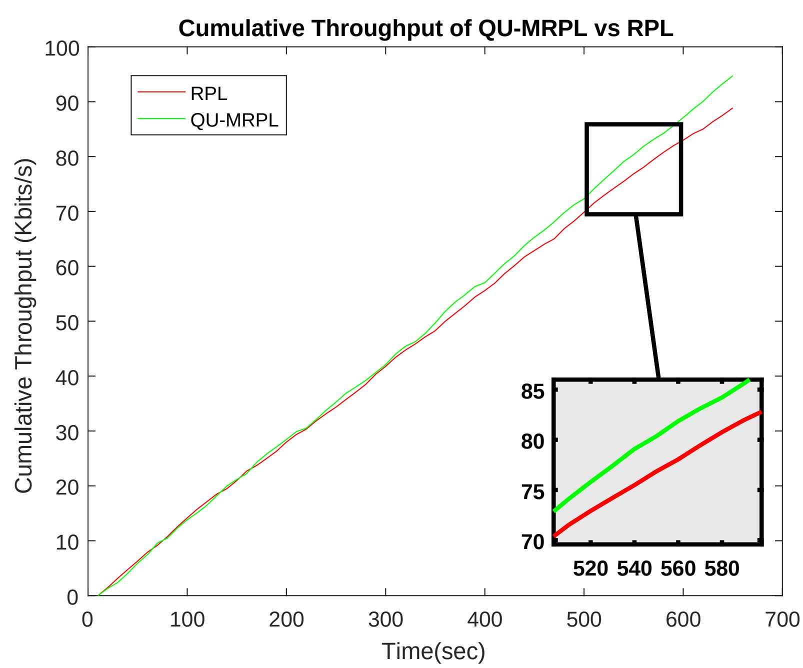


Figure 4.5: OF0: Cumulaitve Throughput of RPL and QU-MRPL in topology 1 with data rate 40 ppm

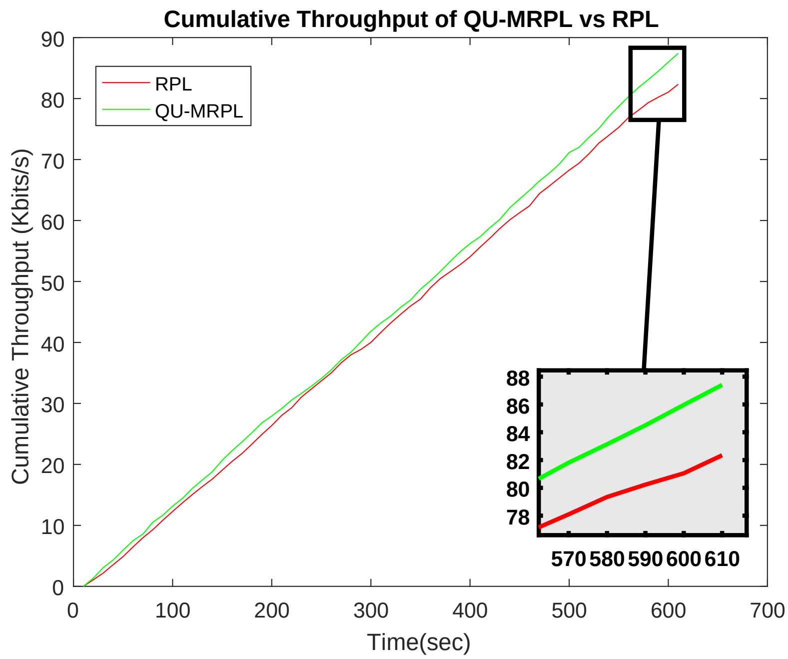


Figure 4.6: MRHOF: Cumulaitve Throughput of RPL and QU-MRPL in topology 1 with data rate 40 ppm

Fig. 4.7 shows the cumulative throughout of OF0 and Fig. 4.8 shows the cumulative throughput for MRHOF with data rate of 30 ppm.

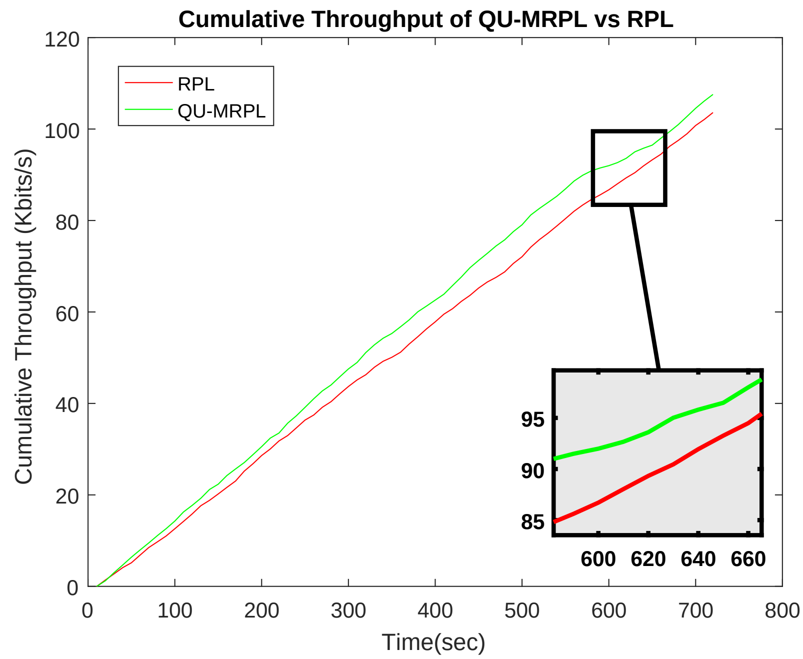


Figure 4.7: OF0: Cumulative Throughput of RPL and QU-MRPL in topology 1 with data rate 30

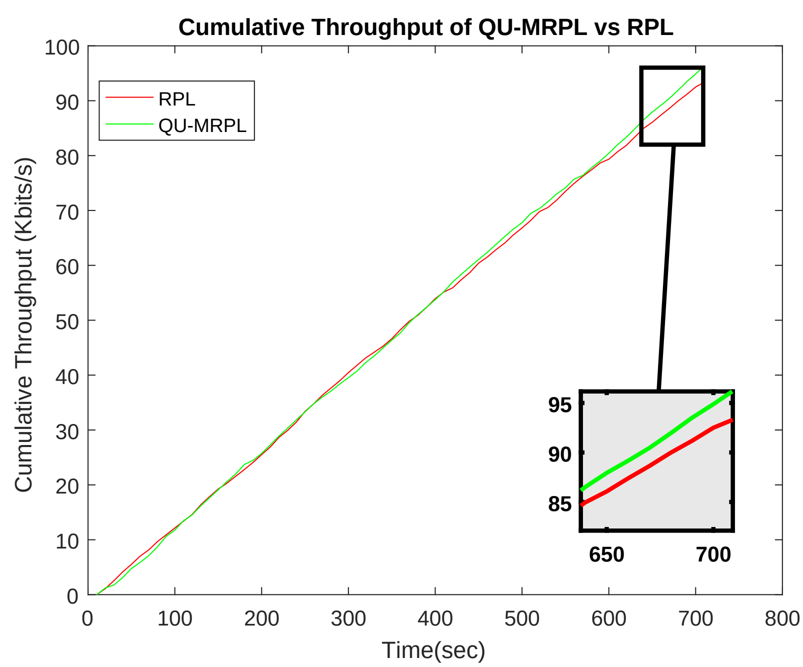


Figure 4.8: MRHOF: Cumulative Throughput of RPL and QU-MRPL in topology 1 with data rate 30

From the above results we can conclude that QU-MRPL performs better than RPL in most of the cases as it increases the throughput. However the throughput is very less in congested scenarios as compared to light traffic. It can also be observed that MRHOF performs better than OF0.

Packet Delivery Ratio Results

The PDR is calculated according to equation 3.3. In Fig. 4.9 and Fig. 4.10 the Packet delivery ratio of OF0 and MRHOF in Topology 1 is shown is shown respectively with varying data rates. The graph shows that QU-MRPL performs better than RPL. As shown in the graphs when the traffic rate is very high the PDR drops as the nodes become more congested.

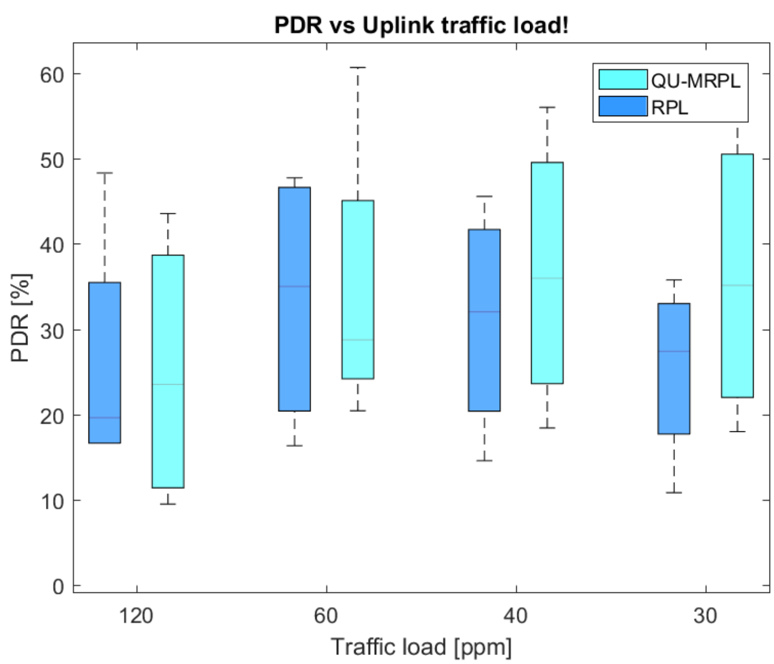


Figure 4.9: PDR of RPL and QU-MRPL in topology 1 with OF0

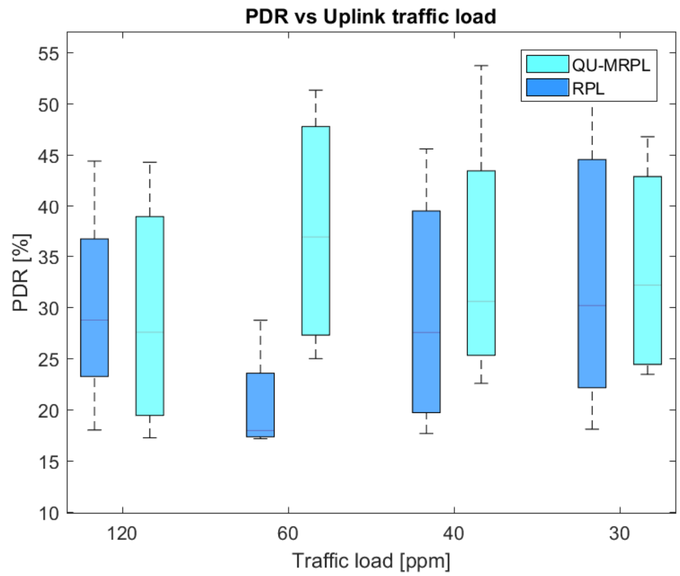


Figure 4.10: PDR of RPL and QU-MRPL in topology 1 and MRHOF

4.1.0.2 Latency Results

We have calculated the latency according to (3.4). Fig. 4.11 and Fig. 4.12 shows the latency of the network with OF0 and MRHOF. The latency increases as the traffic rate increases since the paths become congested. The latency of QU-MRPL is slightly higher than RPL becuase it may take sometime to find alternate paths in case of congestion in the network.

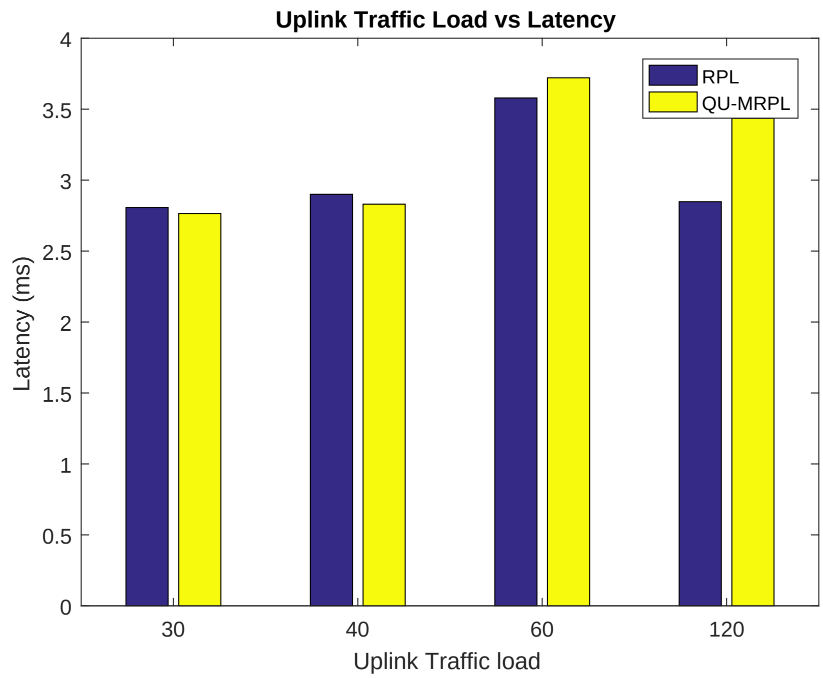


Figure 4.11: Latency of RPL and QU-MRPL in Topology 1 with varying data rates with OF0

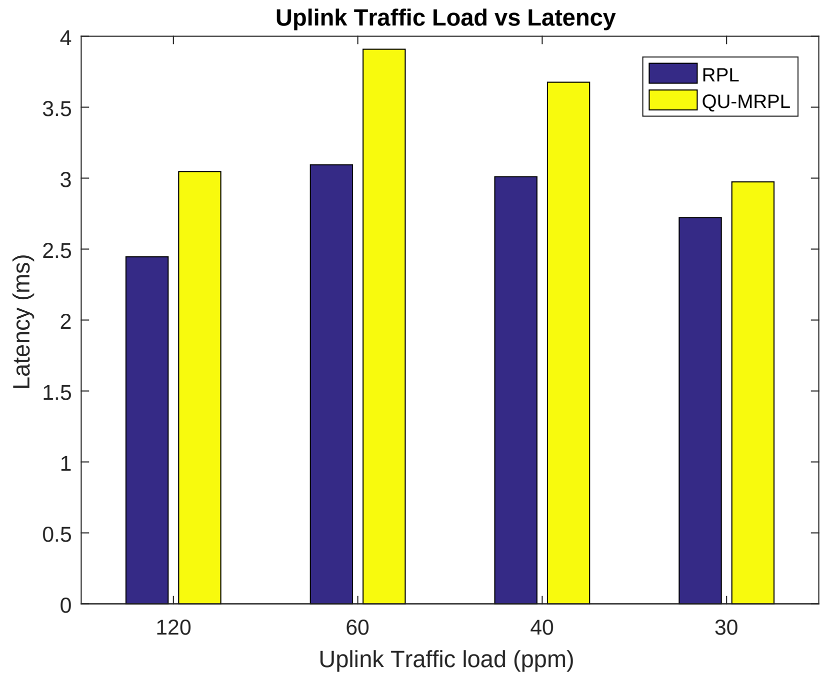


Figure 4.12: Latency of RPL and QU-MRPL in Topology 1 with varying data rates with MRHOF

4.2 Topology 2

This section compares the performance of RPL and QU-MRPL in topology 2 network.

Cumulative Throughput Results

The cumulative throughput is calculated according to equation 3.2. Fig. 4.13 shows the cumulative throughout of OF0 and Fig. 4.14 shows the cumulative throughput for MRHOF with data rate of 120 ppm.

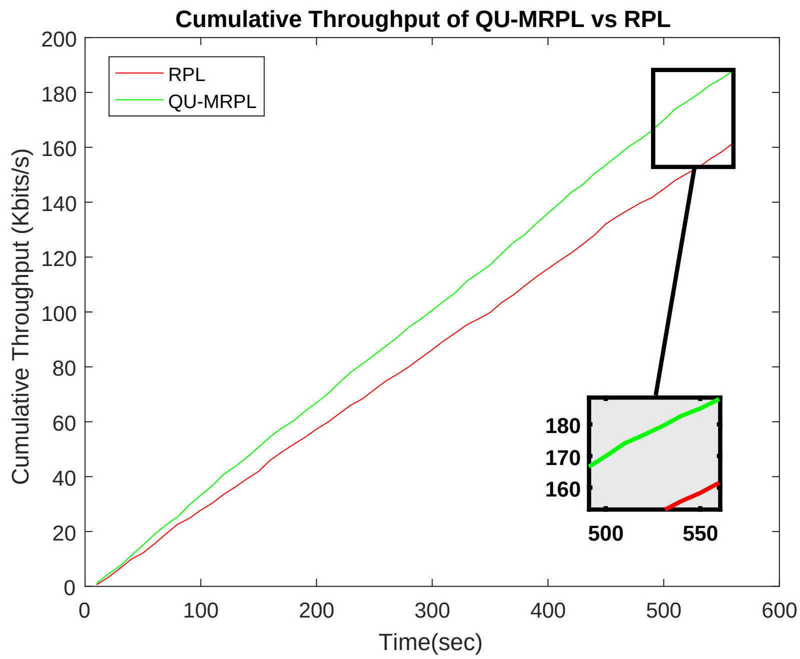


Figure 4.13: OF0: Cumulative Throughput of RPL and QU-MRPL in topology 2 with data rate 120 ppm

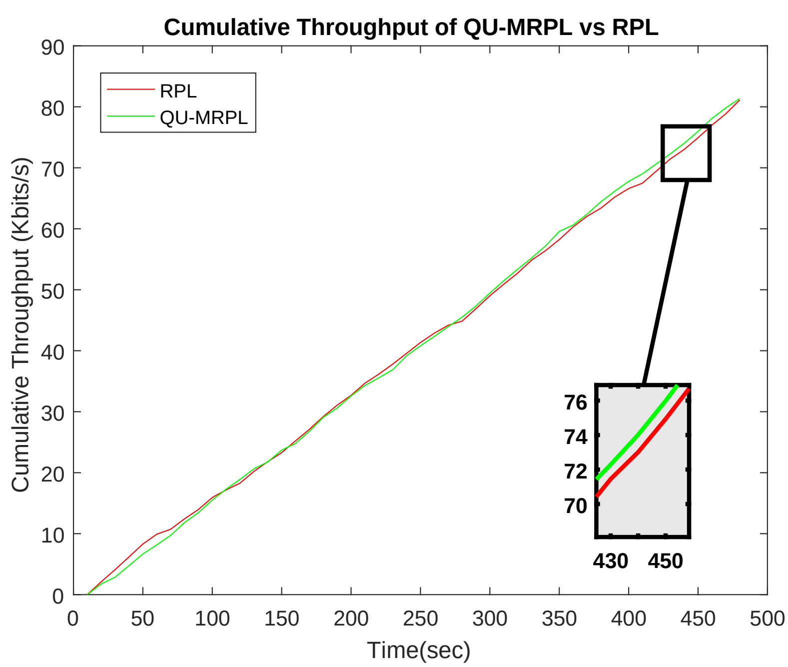


Figure 4.14: MRHOF: Cumulative Throughput of RPL and QU-MRPL in topology 2 with data rate 120 ppm

Fig. 4.15 shows the cumulative throughout of OF0 and Fig. 4.16 shows the cumulative throughput for MRHOF with data rate of 60 ppm.

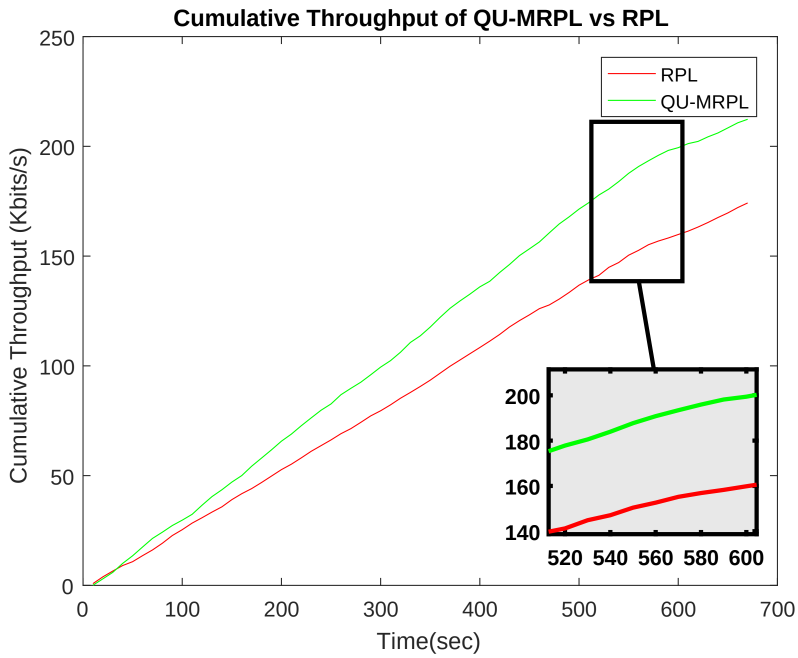


Figure 4.15: OF0: Cumulative Throughput of RPL and QU-MRPL in topology 2 with data rate 60 ppm

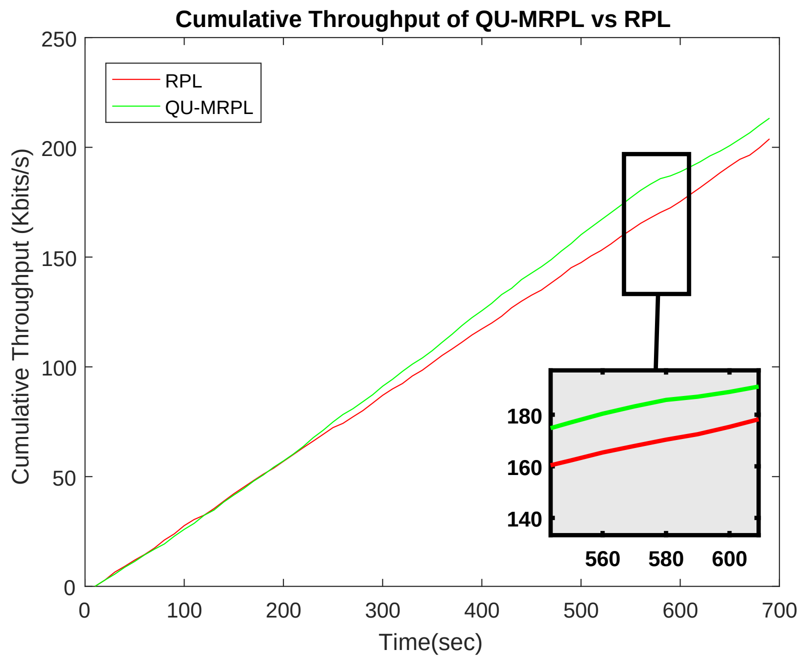


Figure 4.16: MRHOF: Cumulative Throughput of RPL and QU-MRPL in topology 2 with data rate 60 ppm

Fig. 4.17 shows the cumulative throughout of OF0 and Fig. 4.18 shows the cumulative throughput for MRHOF with data rate of 40 ppm.

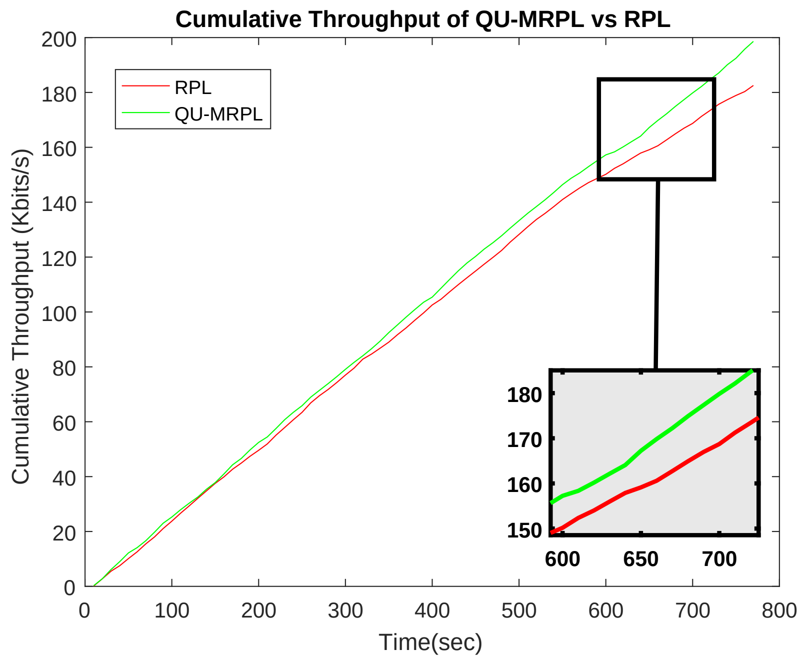


Figure 4.17: OF0: Cumulative Throughput of RPL and QU-MRPL in topology 2 with data rate 40 ppm

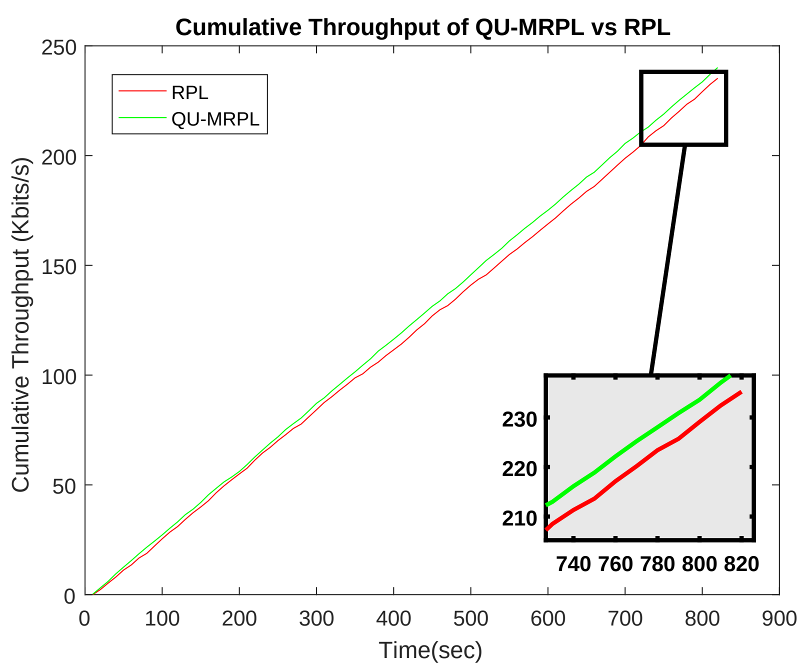


Figure 4.18: MEHOF: Cumulative Throughput of RPL and QU-MRPL in topology 2 with data rate 40 ppm

Fig. 4.19 shows the cumulative throughout of OF0 and Fig. 4.20 shows the cumulative throughput for MRHOF with data rate of 30 ppm.

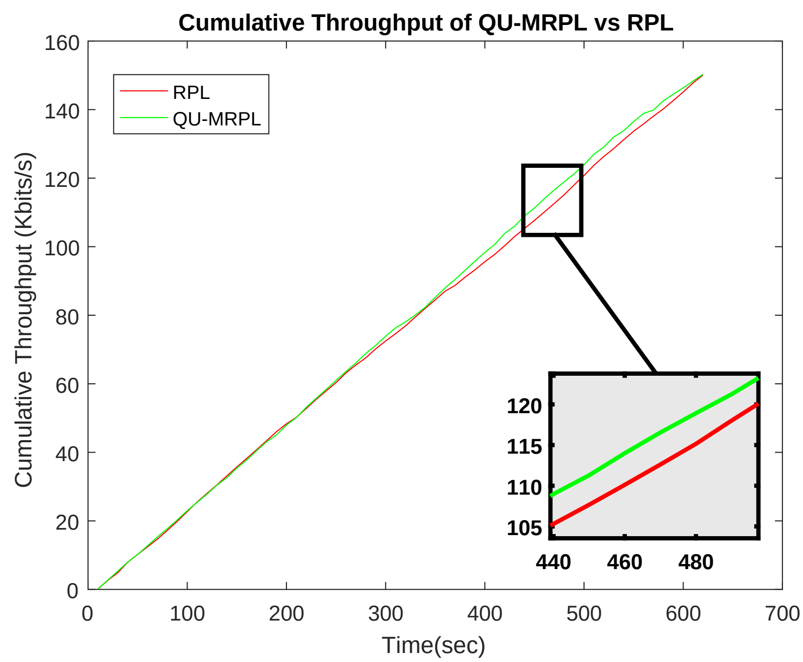


Figure 4.19: OF0: Cumulative Throughput of RPL and QU-MRPL in topology 2 with data rate 30 ppm

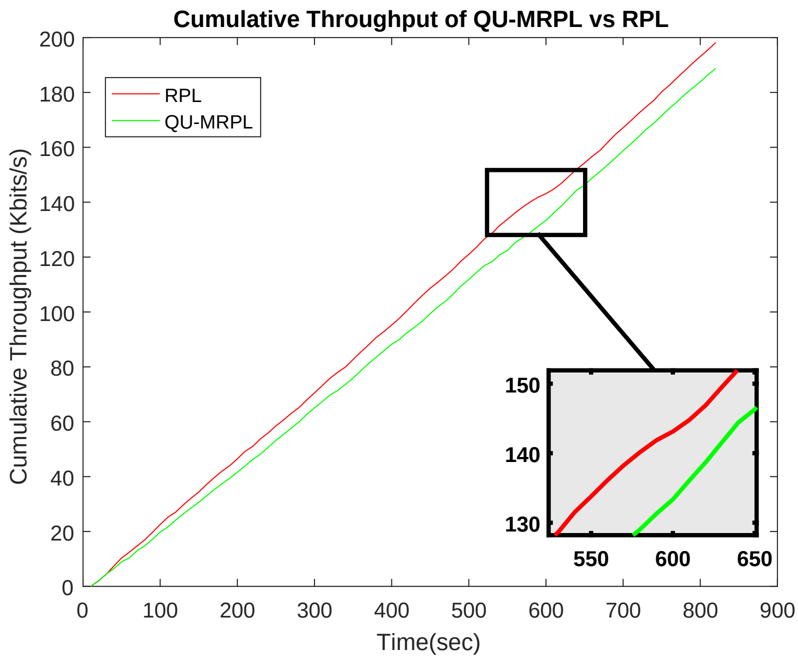


Figure 4.20: MRHOF: Cumulative Throughput of RPL and QU-MRPL in topology 2 with data rate 30 ppm

Packet Delivery Ratio Results

The PDR is calculated according to equation 3.3 for topology 2.

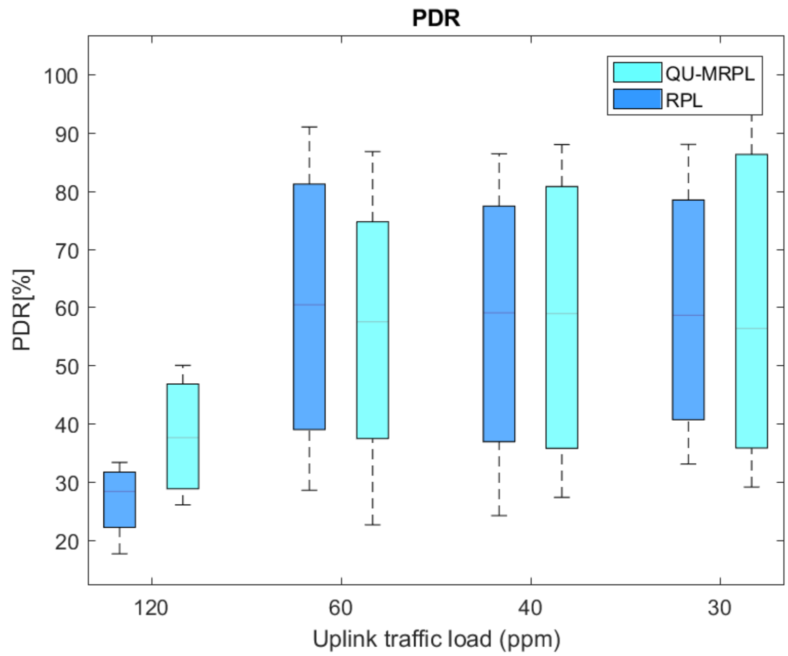


Figure 4.21: PDR of RPL and QU-MRPL in topology 2 with varying data rates with OF0

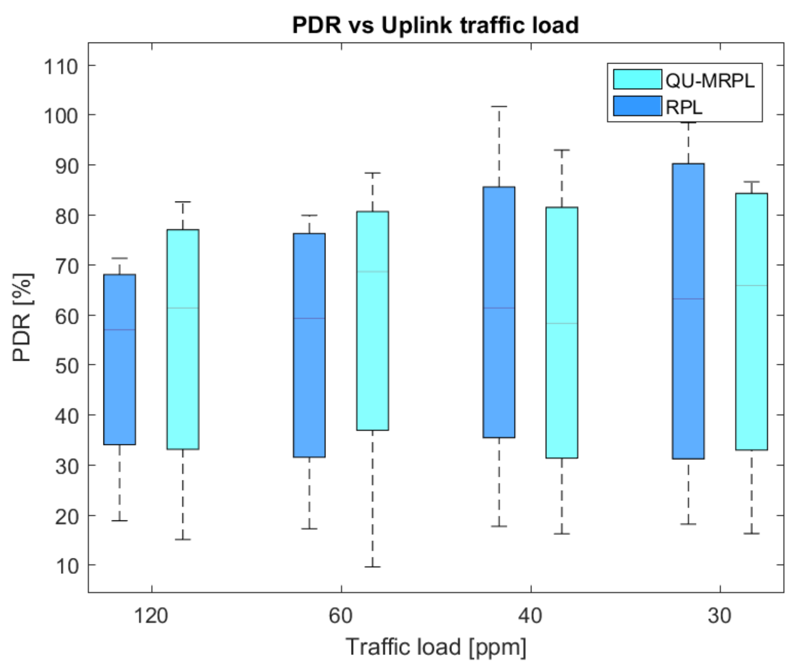


Figure 4.22: PDR of RPL and QU-MRPL in topology 2 with varying data rates with MRHOF

In Fig. 4.21 and Fig. 4.22 the Packet delivery ratio of OF0 and MRHOF in Topology 2 is shown is shown with varying data rates. The graph shows that QU-MRPL performs better than RPL. As shown in the graph when the traffic increases the PDR drops as the nodes become more congested.

Latency Results

The latency of the network is calculated according to (3.4). Fig. 4.23 an Fig. 4.24 shows the latency vs the uplink traffic load with varying data rates. The latency in the network increases as the data rate increases because of the more congested paths.

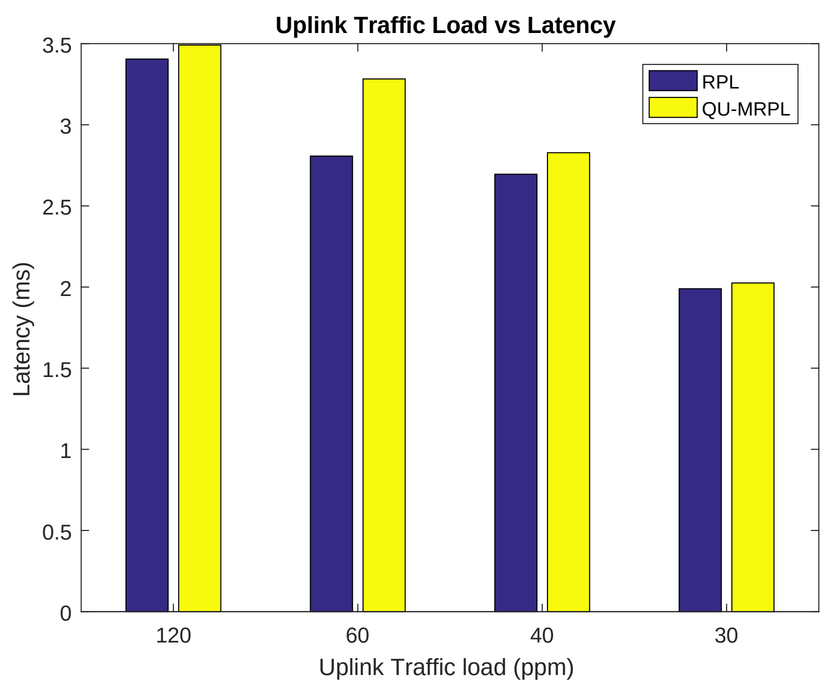


Figure 4.23: Latency of RPL and QU-MRPL in topology 2 with varying data rates with OF0

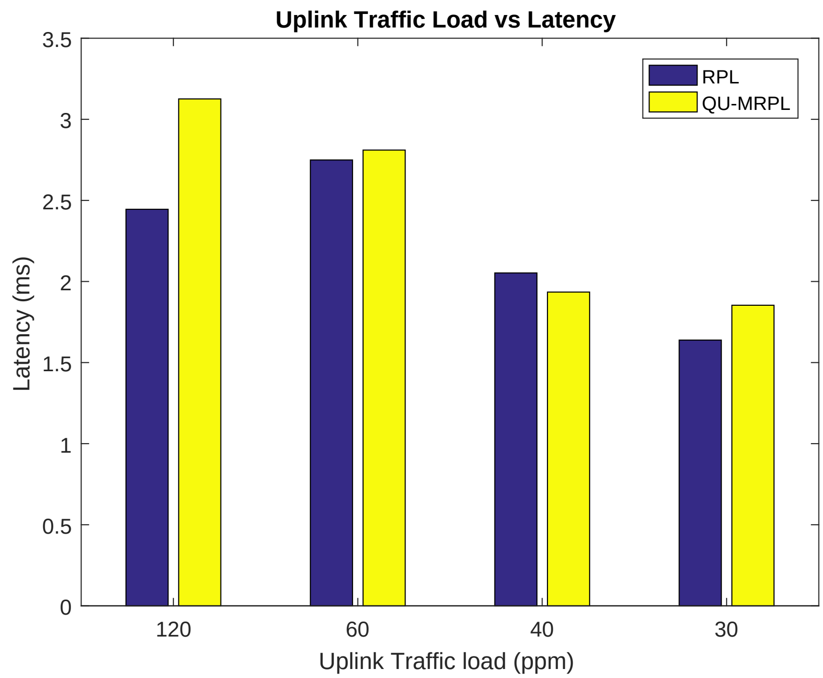


Figure 4.24: Latency of RPL and QU-MRPL in topology 2 with varying data rates with MRHOF

From the above generated results we conclude that QU-MRPL performs better than RPL in most of the cases. Topology 2 performs better than Topology 1. This is because there can be multiple paths for nodes in topology 2 as compared to topology 1. Also the bottleneck nodes will have to forward less traffic in topology 2 as compared with topology 1. However with respect to objective functions, MRHOF performs better than OF0 because it selects routes based on link quality and it provides more reliable routes.

The performance metrics at the end of simulation are summarized in Tables 4.1 - 4.6 below.

Table 4.1: Cumulative Throughput with varying data rates in Topology 1

Cumulative Throughput (Kbps) for Topology 1				
Data rate(ppm)	RPL/ MRHOF	QU-MRPL/ MRHOF	RPL/ OF0	QU-MRPL/ OF0
120	83.13	124.58	98.92	104.73
60	80.28	82.15	87.57	91.08
40	82.34	87.40	88.87	94.72
30	93.38	96.33	103.60	107.56

Table 4.2: Cumulative Throughput with varying data rates in Topology 2

Cummulative Throughput (Kbps) for Topology 2				
Data rate (ppm)	RPL/ MRHOF	QU-MRPL/ MRHOF	RPL/ OF0	QU-MRPL/ OF0
120	81.12	81.36	161.52	187.99
60	203.85	213.28	174.21	212.32
40	238.77	240.09	182.54	198.60
30	188.78	198.16	150.09	150.28

Table 4.3: Packet Delivery Ratio with varying data rates in Topology 1

Packet Delivery Ratio [%] for Topology 1				
Data rate (ppm)	RPL/ MRHOF	QU-MRPL/ MRHOF	RPL/ OF0	QU-MRPL/ OF0
120	9.11	21.68	9.88	20
60	24.20	30.74	12.90	19.03
40	20.86	36.59	26.75	30.82
30	41.47	44.36	47.59	56.55

Table 4.4: PDR with varying data rates in Topology 2

Packet Delivery Ratio [%] for Topology 2				
Data rate (ppm)	RPL/ MRHOF	QU-MRPL/ MRHOF	RPL/ OF0	QU-MRPL/ OF0
120	13.08	19.12	21.26	24.60
60	38.55	54.8	26.20	46.96
40	72.78	90.85	52.25	69.18
30	96.8	101.69	66.93	73.55

Table 4.5: Latency with varying data rates in Topology 1

Latency (msec) for Topology 1				
Data rate (ppm)	RPL/ MRHOF	QU-MRPL/ MRHOF	RPL/ OF0	QU-MRPL/ OF0
120	2.4	3	2.8	3.4
60	3	3.9	3.5	3.7
40	3	3.6	2.9	2.8
30	2.7	2.9	2.8	2.7

Table 4.6: Latency with varying data rates in Topology 2

Latency (msec) for Topology 2				
Data rate (ppm)	RPL/ MRHOF	QU-MRPL/ MRHOF	RPL/ OF0	QU-MRPL/ OF0
120	2.445	3.126	3.4	3.5
60	2.7	2.8	2.8	3.2
40	2.05	1.93	2.6	2.8
30	1.63	1.85	1.9	2

From Table 4.3 and 4.4 we can observe that the OF0 with hop count routing metric presents worse reliability in terms of PDR as compared to MRHOF with ETX routing metric in most of the cases. ETX selects best links to route data packets. Finally our implementation of QU-MRPL provides higher reliability than RPL by increasing the PDR mostly.

Table 4.5 and 4.6 represents the overall end-to-end delay in the network. QU-MRPL has slightly higher delay than RPL. This is because we are delivery more number of data packets to the sink and a data packet is buffered for sometime before the node wakes up to forward it. QU-MRPL presents a maximum delay of 3.4 msec in Topology 1. This kind of delay is tolerable in some applications for example, smart grids generate regular metering traffic that requires reliable delivery.

Chapter 5

Conclusions

Routing is an important paradigm in WSNs because of resource constraints like energy and memory in wireless sensors. RPL is a recently standardized routing protocol for low power and lossy networks. The goal of this thesis was to solve the congestion and load balancing problem in RPL. We proposed a queue utilization based multipath routing approach on top of RPL. Therefore, our work is in line with the standard routing protocol for WSNs and it can be deployed in any WSN environment. We implemented the QU-MRPL in Contiki operating system and its COOJA simulator. Contiki is an open source operating system for WSNs and it provides emulation of a variety of wireless sensors like TelosB. RPL builds a destination oriented directed acyclic graph structure using ICMPV6 control messages. RPL allows nodes to save a list of candidate parent nodes but only one preferred parent node is selected to forward data while others are kept as backup for fault tolerant purposes. We started by monitoring the queue size information of all nodes in the DODAG. Each node sends a destination information object message (DIO) to its neighbor nodes to inform its rank in the DODAG. We utilized the reserved bits in the DIO message to disseminate the queue size information. We enable multi-path routing on nodes where the queue size is above a threshold in order to balance the load and avoid congestion. The aim is to alleviate congested paths in case of unbalanced load distribution in the network. We demonstrate the performance of QU-MRPL as compared with RPL for two Objective Functions MRHOF and OF0. For this purpose we studied two different topologies for analyzing the performance. Experiments are performed with varying data rates. Through extensive simulations we have shown that QU-MRPL outperforms RPL in terms of throughput and packet delivery ratio. Results showed that packet delivery ratio increases as the data rate decreases. We have observed upto 50% increase in throughput in QU-MRPL but the delay is slightly higher than RPL. It is because the alternate path may not be the best path. Further research can be done to validate the effectiveness of QU-MRPL for denser deployment scenarios. An other future aspect of our work will be adaptive QU-MRPL which will manage routing in a dynamic network topology for mobility purposes.

Bibliography

[1] H.-S. Kim, J. Ko, D. E. Culler, and J. Paek, "Challenging the ipv6 routing protocol for low-power and lossy networks (rpl): A survey," *IEEE Commun. Surv. Tutor*, vol. 19, no. 4, pp. 2502–2525, 2017.

[2] J. Vasseur, N. Agarwal, J. Hui, Z. Shelby, P. Bertrand, and C. Chauvenet, "Rpl: The ip routing protocol designed for low power and lossy networks," *Internet Protocol for Smart Objects (IPSO) Alliance*, vol. 36, 2011.

[3] A. Dunkels, "The contikimac radio duty cycling protocol," 2011.

[4] L. Mainetti, L. Patrono, and A. Vilei, "Evolution of wireless sensor networks towards the internet of things: A survey," in *Software, Telecommunications and Computer Networks (SoftCOM), 2011 19th International Conference on*, pp. 1–6, IEEE, 2011.

[5] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[6] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[7] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," *CISCO white paper*, vol. 1, no. 2011, pp. 1–11, 2011.

[8] J.-P. Vasseur and A. Dunkels, *Interconnecting smart objects with ip: The next internet.* Morgan Kaufmann, 2010.

[9] D. Minoli, *Building the internet of things with IPv6 and MIPv6: The evolving world of M2M communications*. John Wiley & Sons, 2013.

[10] J. T. Adams, "An introduction to ieee std 802.15. 4," in *Aerospace Conference, 2006 IEEE*, pp. 8–pp, IEEE, 2006.

[11] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of ipv6 packets over ieee 802.15. 4 networks," tech. rep., 2007.

[12] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. P. Vasseur, and R. Alexander, "Rpl: Ipv6 routing protocol for low-power and lossy networks," tech. rep., 2012.

[13] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pp. 455–462, IEEE, 2004.

[14] M. A. Kafi, D. Djenouri, J. Ben-Othman, and N. Badache, "Congestion control protocols in wireless sensor networks: a survey," *IEEE communications surveys & tutorials*, vol. 16, no. 3, pp. 1369–1390, 2014.

[15] A. Ghaffari, "Congestion control mechanisms in wireless sensor networks: A survey," *Journal of network and computer applications*, vol. 52, pp. 101–115, 2015.

[16] Z. Shelby and C. Bormann, *6LoWPAN: The wireless embedded Internet*, vol. 43. John Wiley & Sons, 2011.

[17] J. A. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, and B. Heile, "Ieee 802.15. 4: a developing standard for low-power low-cost wireless personal area networks," *IEEE network*, vol. 15, no. 5, pp. 12–19, 2001.

[18] P. Thubert, "Objective function zero for the routing protocol for low-power and lossy networks (rpl)," tech. rep., 2012.

[19] O. Gnawali and P. Levis, "The minimum rank with hysteresis objective function," tech. rep., 2012.

[20] H. Ali, "A performance evaluation of rpl in contiki," 2012.

[21] R. Ullah, Y. Faheem, and B.-S. Kim, "Energy and congestion-aware routing metric for smart grid ami networks in smart city," *IEEE Access*, vol. 5, pp. 13799–13810, 2017.

[22] H. A. Al-Kashoash, Y. Al-Nidawi, and A. H. Kemp, "Congestion analysis for low power and lossy networks," in *Wireless Telecommunications Symposium (WTS), 2016*, pp. 1–6, IEEE, 2016.

[23] H.-S. Kim, J. Paek, D. E. Culler, and S. Bahk, "Do not lose bandwidth: Adaptive transmission power and multihop topology control," in *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 99–108, IEEE, 2017.

[24] H.-S. Kim, H. Kim, J. Paek, and S. Bahk, "Load balancing under heavy traffic in rpl routing protocol for low power and lossy networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 4, pp. 964–979, 2017.

[25] B. Pavković, F. Theoleyre, and A. Duda, "Multipath opportunistic rpl routing over ieee 802.15. 4," in *Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pp. 179–186, ACM, 2011.

[26] Q. Le, T. Ngo-Quynh, and T. Magedanz, "Rpl-based multipath routing protocols for internet of things on wireless sensor networks," in *Advanced Technologies for Communications (ATC), 2014 International Conference on*, pp. 424–429, IEEE, 2014.

[27] S. Duquennoy, O. Landsiedel, and T. Voigt, "Let the tree bloom: Scalable opportunistic routing with orpl," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, p. 2, ACM, 2013.

[28] T. Ming-hao, Y. Ren-lai, L. Shu-jiang, and W. Xiang-dong, "Multipath routing protocol with load balancing in wsn considering interference," in *Industrial Electronics and Applications (ICIEA), 2011 6th IEEE Conference on*, pp. 1062–1067, IEEE, 2011.

[29] M. Mamdouh, K. Elsayed, and A. Khattab, "Rpl load balancing via minimum degree spanning tree," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2016 IEEE 12th International Conference on*, pp. 1–8, IEEE, 2016.

[30] F. Boubekeur, L. Blin, R. Leone, and P. Medagliani, "Bounding degrees on rpl," in *Proceedings of the 11th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pp. 123–130, ACM, 2015.

[31] J. Guo, X. Liu, G. Bhatti, P. Orlik, and K. Parsons, "Load balanced routing for low power and lossy networks," July 24 2014. US Patent App. 13/746,173.

[32] M. A. Lodhi, A. Rehman, M. M. Khan, and F. B. Hussain, "Multiple path rpl for low power lossy networks," in *Wireless and Mobile (APWiMob), 2015 IEEE Asia Pacific Conference on*, pp. 279–284, IEEE, 2015.

[33] H.-S. Kim, J. Paek, and S. Bahk, "Qu-rpl: Queue utilization based rpl for load balancing in large scale industrial applications," in *Sensing, Communication, and Networking (SECON), 2015 12th Annual IEEE International Conference on*, pp. 265–273, IEEE, 2015.

[34] H. A. Al-Kashoash, Y. Al-Nidawi, and A. H. Kemp, "Congestion analysis for low power and lossy networks," in *Wireless Telecommunications Symposium (WTS), 2016*, pp. 1–6, IEEE, 2016.

[35] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The trickle algorithm," tech. rep., 2011.

[36] T. Datasheet, "Crossbow inc," 2013.