Bachelor of Science in Computer Science & Engineering

Adaptation of LOADng Routing Protocol in Transmitting Aggregated Data to Reduce Network Traffic Load in IoT

by

Sayeda Suaiba Anwar

ID: 1504081

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

May, 2021

Adaptation of LOADng Routing Protocol in Transmitting Aggregated Data to Reduce Network Traffic Load in IoT

Submitted in partial fulfilment of the requirements for
Degree of Bachelor of Science
in Computer Science & Engineering

by

Sayeda Suaiba Anwar
ID: 1504081

Supervised by

Dr. Asaduzzaman
Professor
Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)
Chattogram-4349, Bangladesh.

May, 2021

The thesis titled '**Adaptation of LOADng Routing Protocol in Transmitting Aggregated Data to Reduce Network Traffic Load in IoT'** submitted by ID: 1504081, Session 2019-2020 has been accepted as satisfactory in fulfilment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering to be awarded by the Chittagong University of Engineering & Technology (CUET).

Board of Examiners

________________________________ Chairman

Dr. Asaduzzaman
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

________________________________ Member (Ex-Officio)

Dr. Md. Mokammel Haque
Professor & Head
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

________________________________ Member (External)

Dr. Md. Mokammel Haque
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Declaration of Originality

This is to certify that I am the sole author of this thesis and that neither any part of this thesis nor the whole of the thesis has been submitted for a degree to any other institution.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I am also aware that if any infringement of anyone's copyright is found, whether intentional or otherwise, I may be subject to legal and disciplinary action determined by Dept. of CSE, CUET.

I hereby assign every rights in the copyright of this thesis work to Dept. of CSE, CUET, who shall be the owner of the copyright of this work and any reproduction or use in any form or by any means whatsoever is prohibited without the consent of Dept. of CSE, CUET.

Signature of the candidate

Date:

Acknowledgements

First and foremost, I want to express my gratitude to Almighty Allah for providing me with the strength necessary to complete this task. The satisfaction that comes with completing this thesis would be incomplete if it weren't for the people whose unwavering cooperation made it possible, and whose constant guidance and encouragement ensured the success of all efforts. My honorable project supervisor, Dr.Asaduzzaman,Professor,Department of Computer Science and Engineering,Chittagong University of Engineering and Technology, was so helpful in the preparation of this project by providing guidelines, motivation, and constructive suggestions.Dr.Md.Mokammel Haque,Professor,Department of Computer Science and Engineering,Chittagong University of Engineering and Technology, deserves special thanks for his wise counsel. I'd also like to express my gratitude to all of my teachers for their invaluable assistance throughout the four years of my education. Finally, I'd like to express my gratitude to my friends, seniors, juniors, and department staff for their valuable suggestions and assistance in ensuring the project's success.

Abstract

The Internet of Things (IoT) is a modern concept that proposes linking devices to exchange data and accomplish a shared goal. When a large number of packets must be exchanged, a low power and lossy network (LLN) consisting of several nodes drains energy quickly. Each sensor's energy usage has a direct impact on the network's operational lifespan. Sensor nodes employ their limited energy to compute and transmit data in a wireless environment, but sending and receiving messages consumes the most of it. Only CHs process and aggregate data in cluster-based data aggregation approaches, with no processing at the node level. On the other hand, one of the most important features of tree-based networks is the ability to build an energy-efficient data aggregation tree while eliminating redundancy. Almost no research has been done on the energy consumption patterns of the Light-weight On-demand Ad-hoc Distance-vector - Next Generation (LOADng) routing protocol. This paper proposes a data aggregation strategy that can be applied in (LOADng). LOADng is designed specifically for LLN. The main purpose of this paper is to aggregate data to reduce data redundancy and transmit it using LOADng to analyse the performance of the routing protocol and compare it with the traditional LOADng. The data aggregation technique focuses on each sensor node, discarding duplicate data and updating route lifetime. Performance of the proposed technique is analysed and also compared with the traditional approach based on various parameters such as packet loss percentage, end-to-end delay, energy consumption, round trip time, jitter and simulation time. The new methodology outperforms the present one, according to the findings.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The Internet of Things (IoT) is a term that refers to the use of the internet to connect different physical devices and objects all over the world [1]. The Internet of Things (IoT), often known as the Internet of Everything or the Industrial Internet, is a new technological paradigm that envisions a worldwide network of interconnected equipment and gadgets. The Internet of Things is widely considered as one of the most essential areas of future technology, and it is attracting significant interest from a variety of industries [2]. The sensing domain, network domain, and application domain are the three domains that make up IoT architecture [3]. The three key characteristics of the Internet of Things are connectivity, sensing, and interactivity.

The exciting next-generation Internet, known as the Internet of Things, is enabled by wireless sensor networks (WSN), enhanced networking protocols, wireless radio frequency systems and a host of other technology and communication solutions [4]. WSNs disperse hundreds to thousands of low-cost micro-sensor nodes across their areas, and these nodes are essential components of the Internet of Things. The nodes in WSN-assisted IoT are resource constrained in a variety of ways, including storage, computation, control, and many more [5].

Network traffic is the amount of time it takes for the first node in the network to run out of resources. A significant number of nodes use multi-hop transmission to send their data to data sinks, causing network traffic congestion.

1.2 Data Aggregation in WSN

Several strategies for data aggregation have been suggested to save resources by reducing the volume of data sent from sensor nodes to their appropriate sink in WSN [6]. A well-known cluster-based protocol named LEACH, uses an election concept to choose the cluster's CH [7].

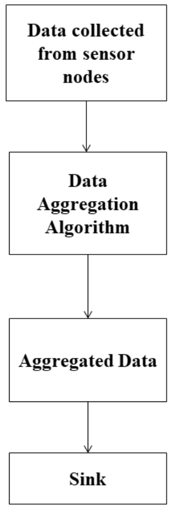

Figure 1.1: Data Aggregation

By minimizing the volume of data to be transmitted and best using network infrastructure by effective cooperation, MIMO and data aggregation strategies are combined to minimize energy usage per bit in WSNs [8].

1.3 Lightweight On-demand Ad-hoc Distance-vector Routing Protocol-Next Generation

IoT devices which are made up of many nodes that are bound by strict memory, computation, and, under certain cases, energy constraints,form a low power and lossy network(LLN). Under the oversight of the routing protocol, routes are discovered and data messages are enabled to be shared between nodes [9]. In low power IoT networks, two kinds of routing protocols, reactive and proactive, are used, both of which are based on path formation concepts [10].

In proactive routing protocols, the creation of routes between nodes begins without the need for data messages to be transmitted first. Gateway nodes, in common, gather data from other network devices and start the route creation process. In reactive routing protocols, routes are only defined when a node needs to send a message to a particular location. As a consequence, the node that wishes to send the message starts the path discovery procedure. The created routes are saved in the routing table of the node and used as needed. This protocol is suitable for applications of non-periodic traffic. The most prominent and widely used reactive protocol for LLN is LOADng [11].

1.4 Difficulties

The Internet Engineering Task Force (IETF) considers LOADng to be a fixed norm. LOADng has been through several revisions since its original proposal in October 2011, and is now in its fifteenth draft [9]. There hasn't been much study into how the various LOADng parameters influence a node's energy consumption patterns [12].

As a result, the main challenges can be summarized as follows:

1. It's a little difficult to articulate the standard because the protocol has undergone many changes.

2. In the LOADng routing protocol, hardly any research has been done on the energy usage behaviors of nodes under various parameters.

1.5 Motivation

The flooding of RREQ messages to establish a path between source and sink is less in the LOADng routing protocol than in other reactive routing protocols, resulting in an increased control message overhead.

For many successive time intervals, similar data sets are provided by sensor nodes, resulting in data redundancy [6]. Data aggregation eliminates redundant data, lowering energy consumption and enhancing node life.

When data from individual sensors is sent to the Internet, there is a lot of traffic load that can induce a failure of the packet. The proposed technique can be used to reduce network traffic congestion.

To ensure low control message overhead, we recommend using the LOADng routing protocol to relay packets between sensor nodes and the internet. To remove redundant data, data aggregation will be completed within the routing protocol.

1.6 Objectives of the work

Key objectives of this thesis work are:

1. To aggregate data for reducing data redundancy with a vision of decreasing network traffic load.
2. To adopt LOADng routing protocol for transmitting aggregated data and examining its performance with the help of different parameters.

1.7 Contribution of the thesis

The key goal is to modify the AODV routing protocol to incorporate the LOADng routing protocol, which is both a simplified and expanded variant of AODV [12]. Then, within the routing protocol, a data aggregation algorithm will be incorporated.

Traditional routing protocols produce a lot of control message overhead, which reduces the battery life by increasing the energy consumption of the node. In clustering-based networks, only CHs process and merge data, with no data processing at the node level [6]. In the proposed process, data aggregation will be done at the node stage.

The following are the major contribution of the thesis:

- As LOADng is still in its draft phase and the complete documentation of its implementation is not available, this protocol has been implemented here from the core. It provides improved QoS and stability by increasing packet transmission ratio and decreasing end-to-end delay for multiple messages shared by nodes in Iot systems.

- The number of control messages sent between nodes is decreased by implementing the SmartRREQ approach resulting in a more efficient network with less overhead.
- A data aggregation algorithm has been implemented within the routing protocol to reduce data redundancy.
- Hardly any work exists regarding energy consumption patterns of LOADng routing protocol [12]. This paper reduces the amount of energy consumed to route data messages, resulting in a more energy-efficient network.

1.8 Thesis Organization

The following is the structure of the rest of this thesis report:

- Chapter 2 provides a synopsis of previous studies in the field of data aggregation and the various routing protocols used in IoT networks.
- The suggested data aggregation algorithm using the LOADng routing protocol is defined in Chapter 3.
- The suggested solution is implemented in details in Chapter 4.
- The proposed solution is compared to the current approach in Chapter 5, which includes the results and performance analysis.
- The future suggestions for the paper are presented in Chapter 6.

1.9 Chapter Summary

The Lightweight On-Demand Ad-hoc Distance-vector Routing Protocol - Next Generation and Data Aggregation in Wireless Sensor Networks (WSN) are discussed in this chapter. It also provides an overview of the Data Aggregation mechanism and its challenges. This section also contains information about the work's motivation as well as the contributions made. In the following chapter, the context and current state of the problem will be addressed.

Chapter 2

Literature Review

2.1 Introduction

Various routing protocols have been adopted to enhance the performance of an IoT scenario by optimising energy consumption and saving battery life in a LLN. Various approaches regarding data aggregation have also been adopted to decrease data redundancy. A complete overview on these things is provided in this chapter.

2.2 Overview of Routing Protocols used in IoT

The Internet Engineering Task Force (IETF) introduced a series of protocols and open standards, including Constrained Application Protocol (CoAP) and 6LoWPAN, to make applications and services more accessible to wireless and resource-constrained devices. MQTT, Advanced Message Queuing Protocol (AMQP), and Data Distribution Service (DDS) are examples of application layer protocols and specifications for IoT [13]. LEACH and other hierarchical protocols aim to increase the network's scalability and reliability. To prevent fake signals and redundant data flow to the base station, they conduct data aggregation. They are, however, vulnerable to HELLO flooding and selective forwarding attacks [14]. Network Flow and QoS-aware protocols, such as the minimum cost forwarding protocol, place a premium on lowering the cost of information flow between nodes. Sink-hole attacks, on the other hand, are a possibility [15]. A number of novel security problems remain to be resolved when these protocols are deployed in the IoT environment.

Since the Optimized Connection State Routing (OLSR) protocol and its successor, OLSRv2, are proactive routing protocols that broadcast neighbor discovery and topology control packets on a regular basis, they are unlikely to be suitable for the Internet of Things. They maintain a comprehensive list about all immediate neighbors and are routed across the entire network. This creates both protocol overhead on the air and storage overhead by draining batteries through unwanted transmissions [16][17].

RPL was developed to serve as the LLN and IoT's routing protocol. It lowers excessive control overhead while increasing control traffic and memory consumption [17].

This work employs the LOADng framework, which enables it to search Internet connected nodes in a diverse and on-demand manner while taking into account the shortcomings of existing literature and the needs of IoT low-power networks.

2.3 LOADng Routing Protocol Overview

LOADng is a simpler version of AODV because an RREQ can only be responded to by the destinations and a LOADng router does not keep a precursor list, which permits an RERR to be sent only to the source. LOADng is also a more advanced variant of AODV, with optimized RREQ flooding and support for various address lengths [12]. In all P2MP and P2P traffic cases, LOADng outperformed RPL [18]. LOADng is developed with networks made up of devices that have severe hardware limitations. It also supports IPv6, IPv4 and Rime addressing schemes [19].

2.3.1 LOADng Operation

Being a reactive routing protocol, LOADng discovers routes by transmitting path request and reply messages. When a node has to deliver a data message but is unsure how to get there, it can initiate a new route discovery process. By broadcasting a route request (RREQ) packet, the node searches for a path to the desired destination. Each node that receives an RREQ must process the request and determine whether or not to forward it. This process is repeated before the RREQ arrives at its destination. The destination should give a path reply (RREP) request in response to the received RREQ. In unicast mode, the RREP is forwarded to the RREQ originator, forming a route between the two nodes participating in the message exchange [9].

If the message is an RREP, the node must search to see if the field ”ackrequired” is set. If the RREP message is valid, an RREP ACK message must be sent to the previous hop. Finally, the node determines if it is the destination of the RREP packet. If the outcome is positive, the route development process has been completed, and data packets will now be submitted. If this is not the scenario, the message is routed to the next hop [20].

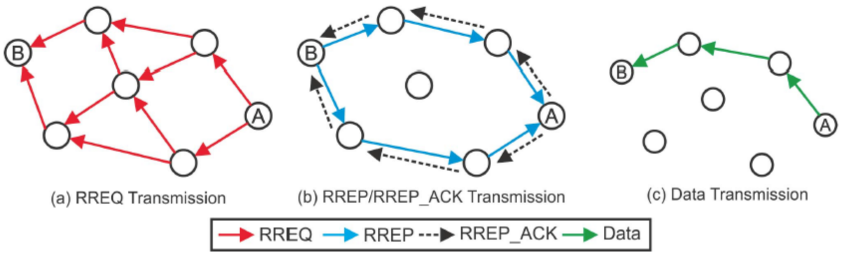

Figure 2.1: Operation of LOADng

The SmartRREQ enhancement was proposed for LOADng to decrease the number of control messages exchanged in the route discovery process [21]. A node can conduct the SmartRREQ relevant processing after completing all of the initial processing and checking that the message is correct for forwarding. As a result, the node determines if it owns a route to the message destination on its Routing Entries. If this condition is fulfilled, the node should send the SmartRREQ message to the next address found in unicast mode [9].

2.3.2 LOADng Control Messages

The control messages used in LOADng are discussed below :

2.3.2.1 Route Request

When a node has to transmit a data to another node, this message is frequently utilized. The sequence number (seq-num), the kind of metric used in the routing process (metric-type), a field carrying the value of the utilized metric (route-metric), a hop count (hop-count), and a hop limit (hop-limit), as well as the message source and destination addresses, are among the numerous fields in RREQ.

2.3.2.2 Route Reply

It is employed by a destination that receives an RREQ in response to a route call. RREP contains similar fields as RREQ. The distinction is that RREP has a field named “ackrequired” that indicates whether the RREP must or must not be replied with an acknowledgment response.

2.3.2.3 RREP Acknowledgement

It’s used to respond to an RREP message with the field "ackrequired" set to real, which the node receives. RREP ACK generally has two fields: sequence number (seq-num) and destination. It’s worth noting that an RREP ACK is only submitted to the originator of the previously issued RREP and can never be forwarded.

2.3.2.4 Route Error

It is used when a node fails when transmitting a data message to the next hop. It can also be used where the node does not know the destination of a data packet. RERR includes an “errorcode” field that specifies the error that happened during the routing procedure, the address of the node that cannot be reached (unreachableAddress), a hop limit (hop-limit), the message’s originator, and its destination.

2.3.3 LOADng Information Base

2.3.3.1 Routing Set

It is made up of a set of routing tuples that store information about the neighbour nodes, such as the next hop and the number of hops required to reach a destination. Furthermore, the routing package stores the tuple's valid time. When a routing tuple's valid time expires, it must be marked as invalid or deleted from the Routing Collection.

2.3.3.2 Blacklisted Neighbour Set

It saves the addresses of nodes with potential faults that prevent communication. In general, it stores the addresses of nodes that are unable to send a necessary acknowledgment response in a communication fault series. Each stored address has a valid timestamp that shows when the data should expire.

2.3.3.3 Pending Acknowledgement Set

It keeps track of the RREP messages that were sent with the field "ackrequired" set to true. Each tuple in the collection includes details about the RREP's next hop, its originator, a sequence number, a flag showing whether or not the RREP ACK was sent and a valid time. Whether the valid time expires and the RREP ACK has not yet been sent, the next hop's address should be added to the Blacklisted Neighbor list.

2.4 Overview of Data Aggregation in Routing

The cluster heads (CHs) in the Single-hop Energy-Efficient Clustering Protocol (S-EECP) are chosen using a weighted probability based on the ratio of each node's remaining energy to the network's average energy, while the elected CHs in the Multi-hop Energy-Efficient Clustering Protocol (M-EECP) use a multi-hop communication approach to propagate data packets to the base station [22].

A time-based technique and spatial aggregation degrees are implemented along with an adaptive data aggregation (ADA) scheme for clustered sensor networks.

The reporting frequency at sensor nodes controls them and the aggregation ratio at CHs, respectively [23]. Tree on DAG (ToD) is a semi-structured technique that employs Dynamic Forwarding on an inherently constructed system composed of several shortest path trees to aid network scalability. The core hypothesis behind ToD was that neighboring nodes in a graph would have low stretching in one of these trees in ToD, leading to early packet aggregation [24][25].

A data cleaning pre-processing method for reducing transmitted packet size and preparing data for an efficient data mining technique based on k-means and FP-tree [26].

2.5 Background and Present State of the Problem

The majority of current hierarchical data aggregation techniques, whether cluster-based or tree-based networks, are committed to event-driven data models. Data aggregation protocols that focus primarily on the collection of CHs and data transfer to the sink can be used in cluster-based networks. Signal processing and physical layer techniques are suggested by other methods [6]. To the contrary, creation of an energy efficient data aggregation tree without eliminating redundancy is considered as one of the key features of tree-based networks.

In recent years, the IETF (Internet Engineering Task Force) has published a series of RFCs (Request for Comments) with the aim of resolving interoperability issues that have arisen in emerging Internet of Things (IoT) scenarios [27].

RPL is a proactive routing protocol that uses a guided acyclic graph to exchange messages between nodes. RPL, to the contrary, has been shown to perform poorly in certain forms of network traffic in recent studies [28][29]. The Lightweight On-demand Ad Hoc Distance-vector Routing Protocol Next Generation (LOADng) has been suggested as an alternative to RPL. Unlike RPL, LOADng is only in the draft phase [20].

2.6 Chapter Summary

This chapter presented an overview of routing protocols used in IoT. An overview of the LOADng routing protocol including it's functioning, control messages and information base has also been provided. A review is done on data aggregation techniques within routing in IoT. Background and present state of the problem has also been analysed.

Chapter 3

Methodology

3.1 Introduction

When a large volume of redundant data is sent from sensor nodes to the destination, there is a risk of significant energy loss and network traffic interference, which could result in packet loss. One of the disadvantages of sensor nodes used in IoT is that they discharge quickly, and once discharged, it is difficult and expensive to reload them. The suggested methodology's key goal is to eliminate network traffic congestion by using a data collection technique to reduce redundant data and transmitting packets using the LOADng routing protocol.

3.2 Representation of Proposed Framework

The entire procedure is split into two phases:

- Data Aggregation Phase
- Routing Phase

3.2.1 Data Aggregation Phase

Any sensor node in the Internet of Things is directly connected to the internet, causing traffic congestion in the network when a vast number of nodes relay data at the same time, as well as a rise in packet loss percentage. We already mentioned in the previous section that other data aggregation approaches that use clustering for aggregation are only concerned with the CH node. However, in order to minimize traffic, we have discarded redundant data in each sensor node.

The working procedure of our adopted algorithm is discussed here.

3.2.1.1 Proposed Algorithm for Data Aggregation

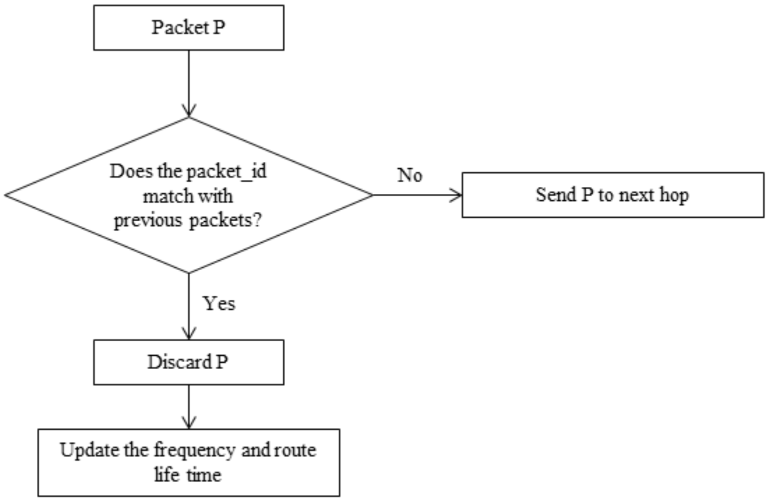

Figure 3.1: Flowchart 1 : Proposed Algorithm for Data Aggregation

When a sensor node generates a packet, it can proceed to its destination by passing through other nodes. Each node keeps a cache file in which it stores the packet ids of previous packets that have passed through it. When a node receives a new packet, it compares the packet-id of the current packet to the packet-id of all previous packets that have gone through the current node. If it fits, the current packet is discarded, and the frequency of the packet, i.e. the number of times an identical packet has arrived at this node, as well as the path lifetime, are changed. If it does not fit, the packet is forwarded to the next hop.

3.2.2 Routing Phase

The LOADng routing protocol will be used to send aggregated data to the destination. Since LOADng is being drafted for a variety of uses, we have used the most generalized version of LOADng here.

The LOADng routing protocol presents an AODV simplification based on RREQ and RREP messages [30]. The SmartRREQ concept has also been suggested for LOADng [21]. To minimize the number of control messages, we introduced the SmartRREQ principle for the LOAdng routing protocol in this approach. Reducing the number of control messages exchanged saves battery life and allows a longer network lifespan.

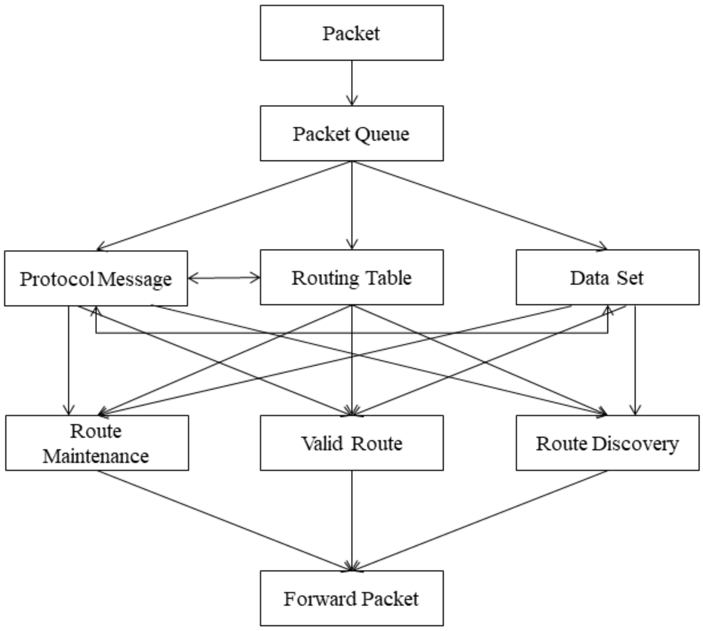

Figure 3.2: Conceptual Model for LOADng

The conceptual model offers a foundation for implementing LOADng. It depicts the movement of data. When a packet arrives at a node, LOADng initiates. Every node in the network uses it to make choices on incoming packets. This conceptual model is used to handle a packet that arrives at the interface of a node and has to be sent across the network. A data packet or a control packet might be used. Unused routes are not saved by this protocol. As a result, when a packet arrives at a node, the node may not be aware of the route to the destination. As a result, the *Packet Queue* module is used to queue the packet for transmission.It is in charge of adding and removing packets from the queue. Before being deleted from the queue, each packet is kept for a set period of time.

To determine whether or not to process the packet, the node reviews its data set. Any transmission from a blacklisted neighbour, for example, is ignored by LOADng. Based on the information in the incoming packet, the node additionally changes its *Routing Table* module and *data set*. The node continues depending on the kind of packet (data or control). If the packet is a data packet, the node looks up a viable route to the destination in its routing table. The packet is sent if a valid route is available.

When there isn't a viable route available, a route discovery is started. The *Protocol Message* module is used to produce protocol messages, which are then forwarded for route discovery. This module is invoked when the arriving packet is a control message. If the control packet is for route discovery, the *Protocol Message* module invokes the Route Discovery module. If the control packet is connected to route discovery, it invokes the *Route Maintenance* module. The *Route Maintenance* module also keeps track of the current state of the node's active connections.Likewise, the *Route Maintenance* module is called when a node fails to identify a route to destination or when a route to destination fails. This paradigm is versatile in the sense that it may be used to create any reactive routing protocol as its foundation. As an example, we use this model for the LOADng routing protocol to explain the implementation.

3.2.2.1 Procedure for Route Discovery

The first step is to determine whether a packet is a data packet or not. If the packet is a data packet with a known destination address, it should be forwarded to the destination in unicast mode. The RREQ for the data packet must be broadcasted if the destination address is unknown but the current node is the source. If this is not the case, an RERR message is sent to the node's source address. If the packet comes from a blacklisted node, it must be discarded.

If the packet isn't a data packet, it should be examined to see if it's a control packet. The routing table must be updated if P is an RREQ. If the current node is the packet's destination, an RREP should be sent to the source in unicast; otherwise, the node should verify whether the packet's destination address is in the routing table. If the address is present then the RREQ is to be unicast to the destination, else, the RREQ is broadcasted .

If P is a RREP, then the routing table needs to be updated. The packet is

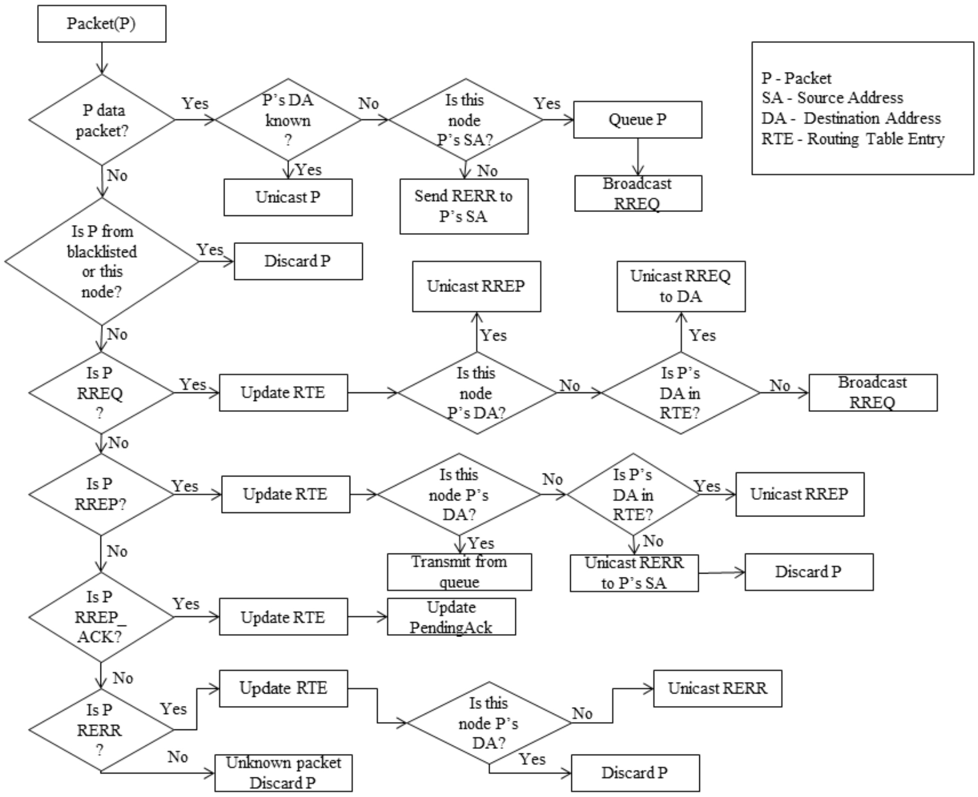

Figure 3.3: Flowchart 2 : Procedure for Route Discovery

dispatched from the packet queue if the present node is P's destination. P is unicast to its destination if the destination address for P is in the routing table of the present node. A RERR signal is unicast to P's source if the packet is not discarded.

The routing table and PendingAck are both updated if the packet is an RREP ACK. If this is not the case, it must be determined whether the packet contains an error message. The routing table will be changed if this is the case. The packet is discarded if its destination is the current node; otherwise, RERR is unicast to the source.

3.2.2.2 Processing Common Message

In this framework, hop count has been taken as the route metric for the development of the routing protocol.The node should update the field hop-count during common processing. It should look through its Routing Set in order to find a route entry for the message originator. A new route record for the message originator is produced if the route cannot be located. After this, the fields of

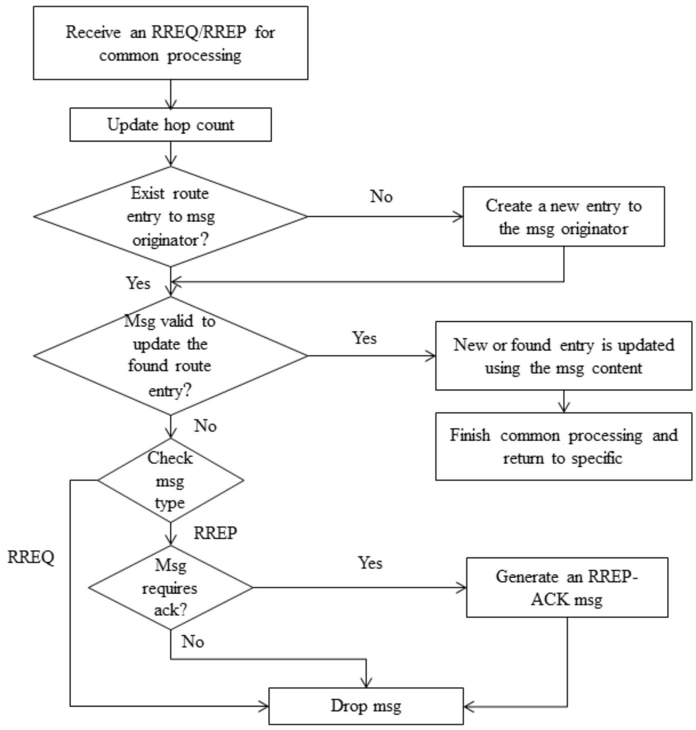

Figure 3.4: Flowchart 3 : Precessing Common Message

the received message are compared to the constructed or discovered route entry to see if the message may be used to update the route entry. If the message is correct, the route entry is modified, the common processing is completed and the message is returned to the original processing. If the message isn't being used to update the route, the node should check the message type and, if necessary, send an RREP ACK.

3.2.2.3 Processing of RREQ

If the received packet is an RREQ, the current node's Routing Table is updated, and the packet's validity for processing is verified. Then common message processing is carried out. A RREP to the source address is produced if the present node is the destination node. If it is not the destination node, it determines if the message is appropriate for forwarding. If it is correct, the node should carry out the specified processing based on the SmartRREQ principle. The node scans the routing table to see if there is a routing entry to the message path. If the condition is met, the node can broadcast the message to the next address located

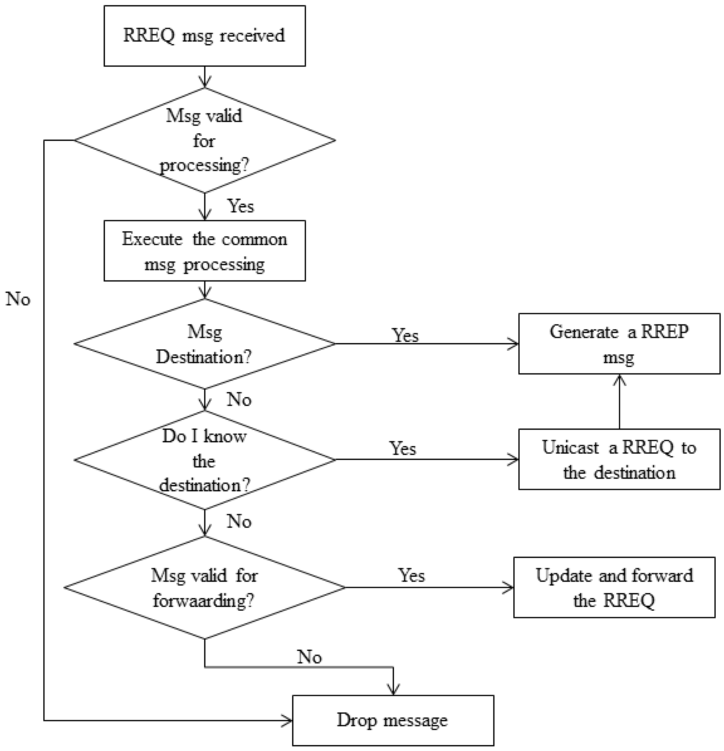

Figure 3.5: Flowchart 4 : Processing of RREQ

in unicast mode.

The message can be processed again by the next address that receives it before it finds its final destination. If the node cannot locate a path entry to the destination, the message should be broadcast. As a result, the SmartRREQ enhancement principle will minimize the number of transmitted broadcasts, thus reducing the control message overhead needed to find a new path and lowering network energy usage.

3.2.2.4 Processing of RREP

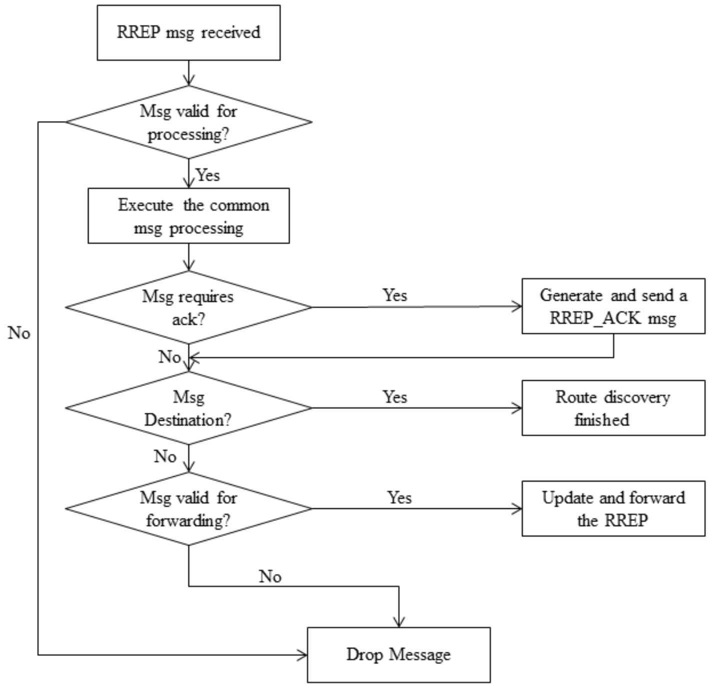

Figure 3.6: Flowchart 4 : Processing of RREP

Validation of message processing and execution of common message processing must both be performed for RREP. If the acquired packet is an RREP and the present node is the destination, the route discovery process is complete. An RREP ACK response is created and delivered if an acknowledgment is required. The routing table of the current node is modified and the message is forwarded if the current node is not the destination but the message is appropriate for forwarding. If not, the packet will be dropped.

3.3 Data Aggregation within Routing Protocol

The whole data aggregation algorithm is implemented within the routing protocol.For duplicate file detection two files named loadng-dpd.cc and loadng-dpd.h have been implemented. In these files, the task of detecting duplicate files along with updating route lifetime has been done. Another two files named loadng-cache.cc and loadng-cache.h are maintained to keep a record of the packets which pass through a node.loadng-dpd.h and loadng-cache.h are included in the main file. When a packet passes through a node, it matches the packet-id with the previous packet-ids which are recorded in the cache. If the packet-ids match, then the packet is discarded,else, the packet is forwarded next.

3.4 Chapter Summary

The proposed approach, which involves the routing phase and data aggregation phase, has been reviewed in this chapter. It's also explained how the data aggregation process is used in the routing protocol.The implementation details are analysed in the next chapter.

Chapter 4

Implementation Details

4.1 Introduction

The primary goal of this paper is to reduce network traffic congestion and node energy usage. To compare our proposed solution to the current approach, a virtual environment was developed.

4.1.1 Software Specificification

- Operating System : Linux
- Software used for simulation : Network Simulator - 3 (Version 3.29)
- Programming Language Used : C++

4.2 Routing Protocol Module Implementation

Implementation of LOAADng is done in ns3 based on Ipv4. To tackle Ipv4 routing, there are built-in classes in ns3 called ns3:: Ipv4RoutingProtocol. These groups can be extended by a new routing protocol to incorporate Ipv4-based routing functionalities. This paper describes the design of an on-demand routing system based on Ipv4. The following virtual member functions are available in the class:

4.2.1 SetIpv4

This feature connects an Ipv4 entity to the routing protocol.

4.2.2 PrintRoutingTable

The routing table entries (RTEs) of the node are printed using this feature.

4.2.3 RouteOutput

When an outbound packet is received and a correct route to the destination is needed, this feature is called. It looks at the RTEs to see if there is a valid Ipv4 path to the destination. This feature invokes the path discovery procedure's start function. After the route is defined, the RouteInput function receives the corresponding RTE.

4.2.4 RouteInput

This feature is used to forward a packet and offers three choices for forwarding. Unicast, local delivery, and error callback are the three options.

4.2.5 NotifyInterfaceUp

This feature can receive messages if the status of an interface changes. It's used to set up a user interface. The interface's Ipv4 address is shared with a socket that receives and forwards packets through it.

4.2.6 NotifyInterfaceDown

When the interface goes down, this feature will be notified.

4.2.7 NotifyAddAddress

When an interface incorporates a new address, this feature is called. The role connects the sockets to the interface addresses and updates the routing table with new paths.

4.2.8 NotifyRemoveAddress

When an interface excludes an address, this feature is called. The feature deletes routes from the routing table and unbinds sockets connected with interface addresses.

4.2.9 NotifyAddRoute

When the Ipv4 stack adds a route, this feature is called.

4.2.10 NotifyRemoveRoute

When a path is disabled, this feature is named.

4.3 Packet Forwarding

Inbound and outbound data/control packets are handled by the functions RouteOutput() and RouteInput(). The RouteInput() function handles packets that arrive at a node and determines the mode of action. The packet is forwarded via unicast or locally depending on the type of packet and the destination address. If the RouteInput() function is unable to forward the packet to its intended destination, an error callback may be created. The LocalDeliverCallback() method is used to deliver packets intended for local distribution. A valid route to the destination is required by the node that has a packet to forward to another node. RouteOutput() looks for a correct route entry in its routing table. If a correct RTE is detected, the packet will be forwarded using the information in the RTE. A source initiated on-demand routing protocol initiates a route discovery if a valid RTE is not available, as long as the packet under consideration is provided by this node. A routing protocol's route discovery process is distinctive and complex.

4.4 Path Discovery Module

The protocol-specific features are handled by the path discovery module. The following are significant architecture considerations:

4.4.1 Data Structure of Routing Table

The routing table should include all of the information needed to correctly route a packet to its destination. The data structure used to execute the routing table is essential. The routing table's primary operations are search, add, remove, and change. Since the nodes are limited in terms of resources, the computational complexity of these operations should be minimized.

4.4.2 Control Messages

A control message's fields are protocol-specific. Control messages should be designed and implemented with extreme caution, since they are the keys to establishing a path to a certain destination.

4.4.3 Queue for Packet

Until forwarding the packet to the destination, the packet queue holds the incoming packet. It is needed during route exploration.

4.4.4 Additional Data Structures

For a packet to be routed correctly, additional information such as adjacent nodes and the blacklisted node is necessary. The data structure used to store the additional information is determined by how it will be used during route exploration.

4.5 Classes

The LOADngHelper class assists a node in implementing LOADng routing. RoutingProtocol extends the abstract class ns3::Ipv4RoutingProtocol which represents the core routing logic. The QueueEntry class generates a queue entry for an incoming packet, which is then queued in the RequestQueue class. When a source node has any packets to send to a destination node, it calls the RoutingProtocol class to start the routing process. It looks up a correct path to the destination in its routing table. To find a correct entry to the node, the protocol uses the RoutingTable class. This RoutingProtocol needs additional route information to be stored. As a result, a special RoutingTableEntry class is used, which extends the ns3::Ipv4Route class. MANET's well-known UDP port number 269 is used by LOADng[18].

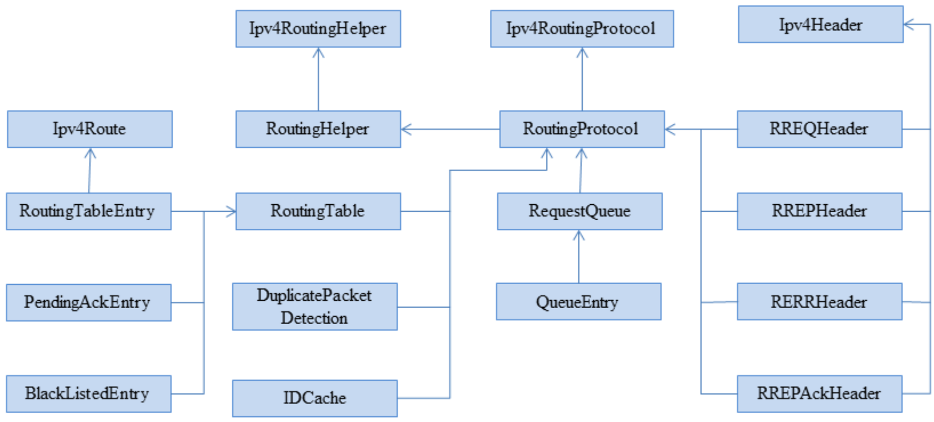

Figure 4.1: Class Diagram

4.6 Routing Parameters

Table 4.1 lists all of the important parameters as well as the values that were used in the simulation. The NetDiameter parameter specifies the maximum number of hops available to complete a network traversal. The average time it takes for a packet to travel one hop distance is the NodeTraversalTime. The NetTraversalTime is the time it takes for an RREQ to cross the entire network.

The maximum number of RREQ retries a node may render before making a destination unreachable is RreqRetries. The MaxDist field specifies the metric's highest potential value. If the hop count is the metric to consider, then MaxDist is equal to NetDiameter.

Parameters for LOADng	Value Used
NetDiameter	35
NetTraversalTime	40 ms
NodeTraversalTime	2.8 s
Rreqretries	2
TimeOutBuffer	2
BlackListTimeOut	5.6 s
NextHopWait	50 ms
RrepAckRequired	TRUE
UseBidirectionalLinkOnly	TRUE
MaxHopCount	255
MaxQueueLength	64

Table 4.1: Routing Parameters

4.7 Chapter Summary

This chapter summarizes all of the prerequisites for implementing the proposed strategy. The results of the suggested solution are discussed in the following sections.

Chapter 5

Results and Discussions

5.1 Introduction

Data aggregation efficiency in LOADng has been analyzed in this chapter. Several packets with a different number of nodes have been transmitted using the LOADng routing protocol, with or without data aggregation, for both cases. The results for various parameters have been correlated.

5.2 Parameters for Simulation

Simulations were conducted using 10 to 90 nodes, all of which were equipped with the 802.11 Wi-Fi Physical Layer, which works at 2.4 GHz and gives a data rate of up to 11 Mbps. The nodes are distributed at random, with a constant density. As a result, as the number of nodes increases, the network's coverage area expands, increasing the average number of hops between any two nodes in the network. By adjusting the total number of nodes in the network, the influence of traffic on network performance and energy efficiency can be observed.

Each packet is 64 bytes in size, with a data rate of 2048 bits per second. The mobility model is RandomWayPointMobilityModel, with a maximum speed of 20 m/s.

Parameters for the simulation are summed up below:

Parameter	NS-3 simulation platform
Position of sensor nodes	Random with constant node density
Number of sensor nodes	10 to 90 nodes
Propagation Delay Model	ConstantSpeedPropagationModel
Application Model	On-off application
Mobility Model	RandomWaypointMobilityModel
Propagation Loss Model	FriisPropagationLossModel
Simulation Time	300 seconds
Wifi Remote Station Manager	ConstantRateWifiManager
Energy Source	BasicEnergySourceHelper
Device Energy Model	WifiRadioEnergyHelper

Table 5.1: Parameters for Simulation

5.3 Experimental Result

We have focused on following parameters to validate our experiment :

1. Packet Loss Percentage
2. End to end Delay
3. Energy Consumption
4. Average Round Trip Time
5. Jitter and
6. Simulation Time

5.3.1 Packet Loss Percentage and No. of Nodes

Packet loss happens if one or more data packets are unable to meet their destination in a computer network. This measure is calculated by dividing the number of packets dropped by the total number of packets and multiplying the result by 100. The number of packets lost or dropped during transmission must be maintained to a minimum for smooth and uninterrupted transmission.

Packet loss percentage has been measured for a different number of nodes to observe the performance of the data aggregation algorithm within LOADng and

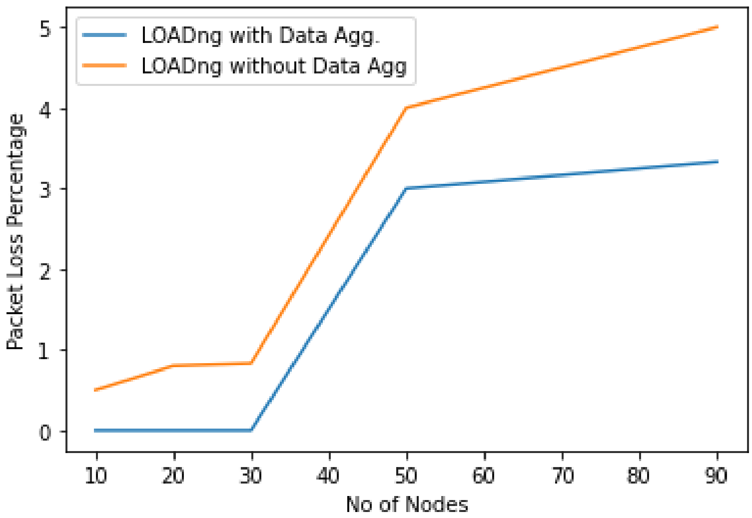

Figure 5.1: Packet loss percentage vs. number of nodes

to compare with the existing LOADng routing protocol. In this paper, the percentage of packet loss against nodes of between 10 and 90 is evaluated. In both the cases, packet loss percentage increases with the increasing number of nodes, but for the proposed approach, packet loss percentage is less than the existing approach. The "LOADng with Data Aggregation" curve almost flattened after 50 nodes.

5.3.2 End To End Delay and No. of nodes

The time it takes for a packet to travel from source to destination through a network is referred to as end-to-end delay. It's the sum of a packet's Transmitting Delay (at the MAC layer), Propagation Delay, and Queuing Time. Delay is determined by the number of hops and network congestion. Higher delays may cause routers to discard packets because the TTL limit has been exceeded. To evaluate the performance of the suggested approach, end-to-end delay is measured for various numbers of nodes. The graph of end-to-end delay vs. number of nodes shows that the curve for "LOADng without Data Aggregation" increases dramatically after node 40, while the curve with data aggregation does better.

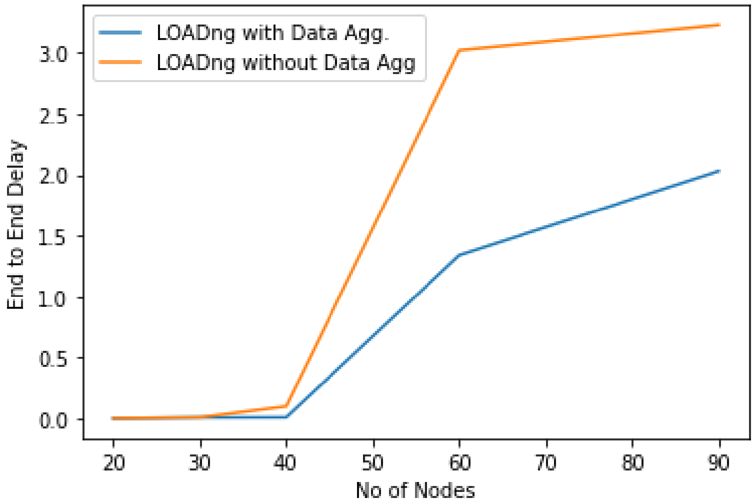

Figure 5.2: End to end delay vs. number of nodes

5.3.3 Energy Consumption and No. of Nodes

In IoT, network lifetime is important for applications. Energy consumption of each sensor directly affects the network operational lifetime. Sensor itself supplies necessary operation with limited battery energy. The battery consumption of a sensor node is significantly correlated with its longevity because it is the only low energy source available, and charging it is difficult and expensive in most circumstances. In a wireless environment, sensor nodes use their limited energy to compute and transmit data, but sending and receiving messages consumes the most energy.

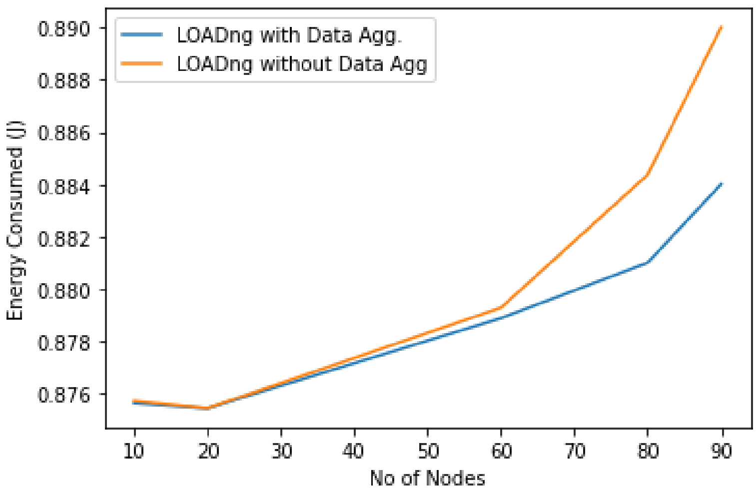

Figure 5.3: Energy Consumption vs. number of nodes

In ns-3, the energy model consists of 2 components, namely, energy source and device energy model. On each node, there will be just one energy source, which will indicate the node's overall energy store. The energy spent by each device will be reported to the energy source, which will update the source's remaining energy. The energy source will inform all linked device energy models when the energy is totally depleted. BasicEnergySourceHelper and WifiRadioEnergyHelper, which are used in ns-3 as Energy Source Model and Device Energy Model, have been used to analyze the energy usage of a node. The Data Aggregation solution uses less resources than the current one , ensuring a longer network lifespan.

5.3.4 Avg. Round Trip Time and No. of Nodes

The round trip time is the time it takes for a signal to be transmitted as well as the time it takes for the signal to be acknowledged as having been received. The transmission times between the two contact endpoints are included in this time delay. Lower the round trip time for a different number of nodes, better the performance of the proposed approach. The graph is almost stable at first, but after 50 nodes, the curves grow, and we can see that the curve with data aggregation performs better.

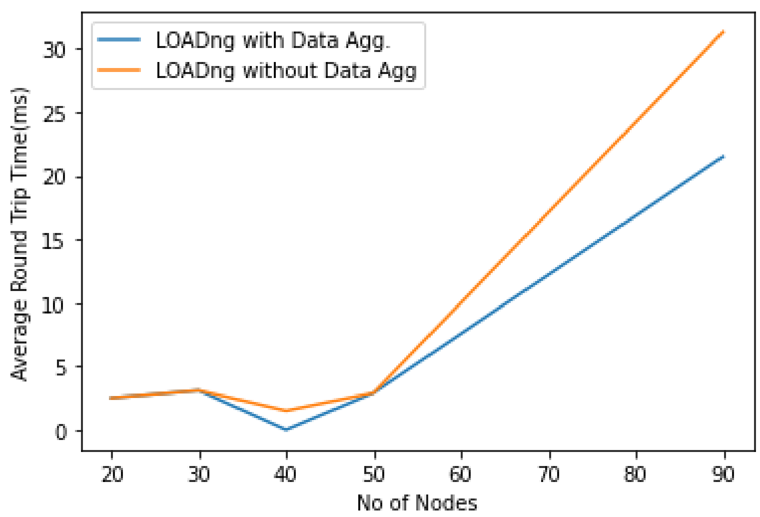

Figure 5.4: Avg round trip time vs. number of nodes

5.3.5 Jitter vs No. of Nodes

Data packets are used to convey information across the internet. They are often dispatched at regular intervals and require a specific period of time to complete.

When data packets are sent across a network, there is a temporal delay known as jitter. Jitter is the difference in time between arriving data packets induced by

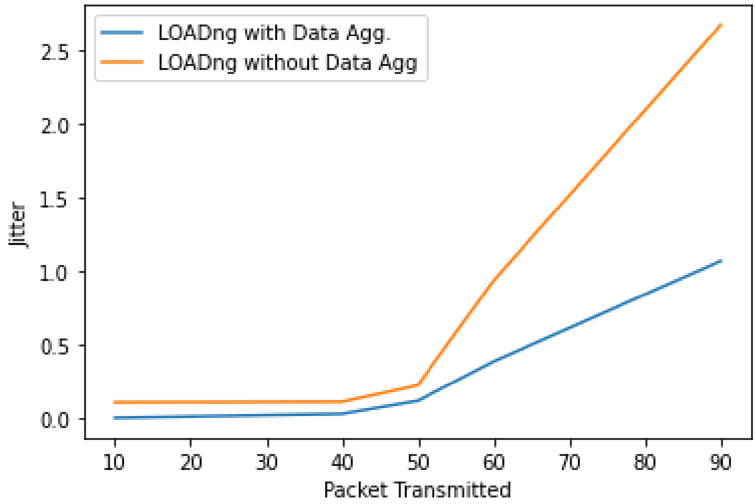

Figure 5.5: Jitter vs. number of nodes

network congestion or path changes. In other words, the longer it takes for data packets to arrive, the more jitter might degrade the information quality. When data is aggregated, congestion across the network decreases. As a result, jitter should also decrease compared to the traditional approach. Here jitter is analysed against an increasing number of nodes ranging from 10 t0 90 nodes. Initially the graph remains almost constant, after 50 nodes, the curves rise, where we can see a better performance for the curve with data aggregation.

5.3.6 Packet Loss Percentage and No. of Packets

For a better understanding of the performance , the proposed approach is analysed in depth. Here the packet loss percentage has been observed with respect to the number of packets for a varying number of nodes such as 10,60 and 90.

When the number of nodes is 10, Packet loss percentage is null for proposed approach whereas it is proportionally increasing for existing approach.

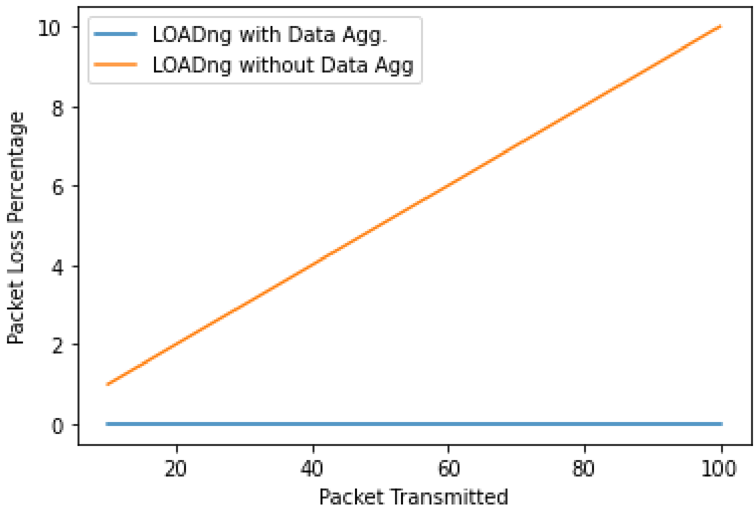

Figure 5.6: Packet Loss Percentage vs. No. of Packets for n=10

When node number is 60, Both the graphs are almost showing the same pattern of result initially. Later on,it is observed that the curve for the existing approach is rising steeper than the proposed approach.

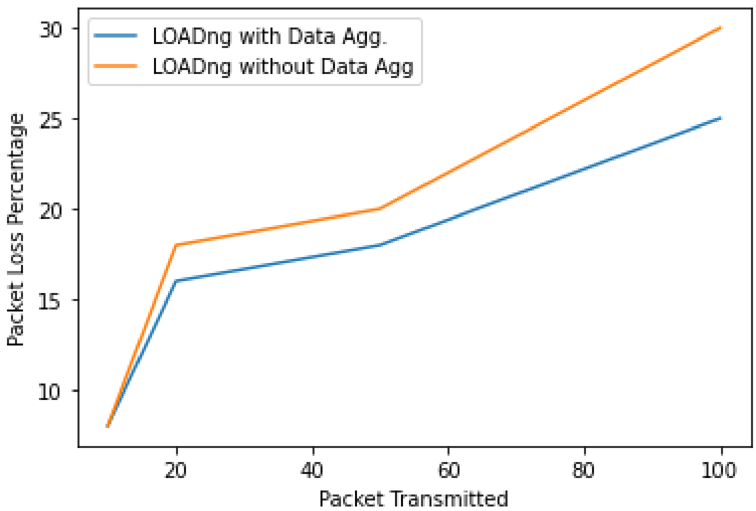

Figure 5.7: Packet Loss Percentage vs. No. of Packets for n=60

When the simulation is done for 90 nodes, then it is seen that the curve for the existing approach is steeper than the proposed approach which ensures a better performance for the existing approach.

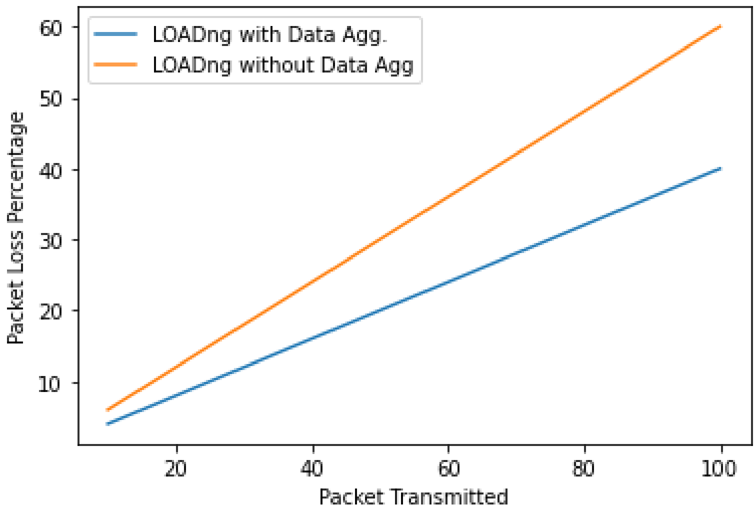

Figure 5.8: Packet Loss Percentage vs. No. of Packets for n=90

From the above analysis it is to be understood that when the number of transmitted packets increases the packet loss also increases but it varies with the number of nodes. And packet loss percentage with respect to the number of packets has decreased in data aggregation approach.

5.3.7 Packet Loss Percentage and Simulation Time

The amount of time spent simulating a model is referred to as simulation time. This parameter assesses the simulator's performance. In this work, LOADng is implemented based on IPv4 in ns-3 for both standard and suggested approaches.The total simulation time is considered as 300 seconds. Packet Loss Percentage is also analysed for a varying simulation time for a better understanding. It has been seen that for an increasing simulation time, packet loss percentage has been decreased.

When number of nodes, n= 10 :

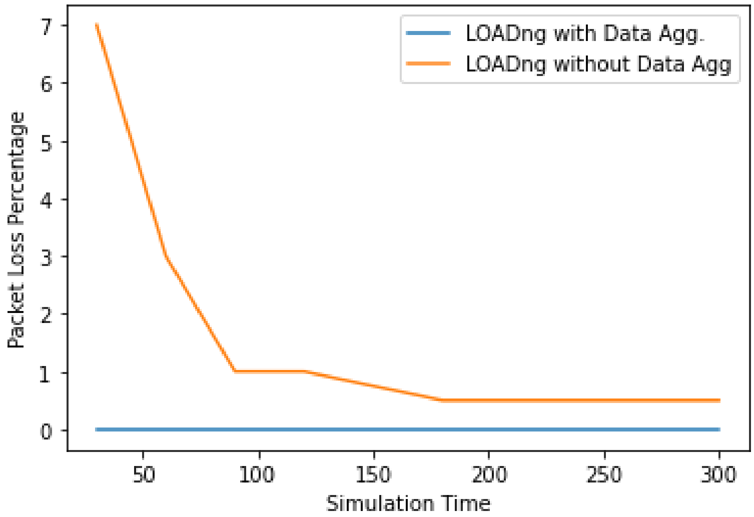

Figure 5.9: Packet Loss Percentage vs. Simulation Time for n=10

When number of nodes, n= 60 :

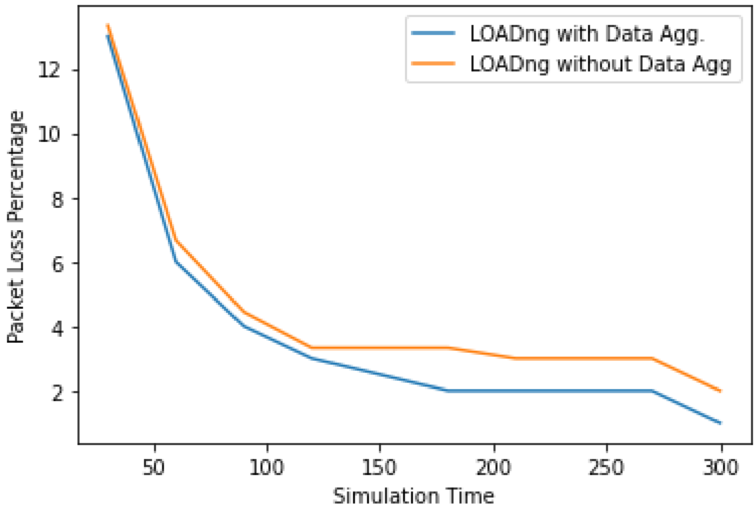

Figure 5.10: Packet Loss Percentage vs. Simulation Time for n=60

When number of nodes, n= 90 :

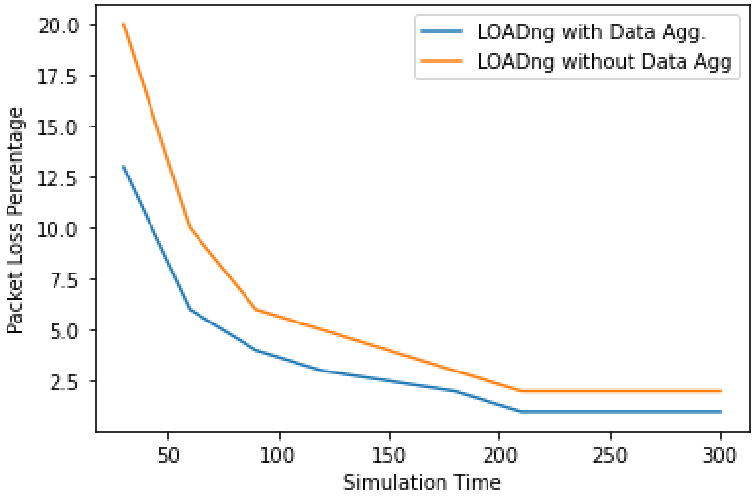

Figure 5.11: Packet Loss Percentage vs. Simulation Time for n=90

It has been seen that for an increasing simulation time, packet loss percentage has been decreased.

5.4 Chapter Summary

Two vital requirements for an IoT network are low energy consumption and fast packet transport. In this part, we compared the performance of the proposed solution to that of the current approach under various criteria. It can be inferred that both methods co - exist at first, but after 50 nodes, the proposed solution outperforms the others, which is significant given the vast number of nodes in an IoT network.

Chapter 6

Conclusion

6.1 Conclusion

A data aggregation algorithm for Light-weight On-demand Ad-hoc Distance-vector Routing Protocol-Next Generation(LOADng) for Low Power and Lossy Networks is discussed in this paper. Data Aggregation decreases data redundancy, resulting in lower energy usage and reliability, as well as a longer network lifespan. Different clustering-based data aggregation methods concentrate on the cluster head (CH), while the suggested method focuses on each sensor node. The proposed solution achieves better outcomes than the current one.

6.2 Future Work

Following are some of the project's future recommendations:

1. The route metric used in this paper is hop count. Other route metrics may be used to determine whether or not it improves efficiency.

2. The simulation module's code will be optimized as part of future work. The module includes notes, test suites, and illustrations. The paper is unfinished since the LOADng Internet-draft has not yet been finalized, and work on it is still ongoing.

References

[1] P. Gokhale, O. Bhat and S. Bhat, ‘Introduction to iot,’ *International Advanced Research Journal in Science, Engineering and Technology*, vol. 5, no. 1, pp. 41–44, 2018 (cit. on p. 1).

[2] I. Lee and K. Lee, ‘The internet of things (iot): Applications, investments, and challenges for enterprises,’ *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015 (cit. on p. 1).

[3] H. Chen, X. Jia and H. Li, ‘A brief introduction to iot gateway,’ in *IET international conference on communication technology and application (ICCTA 2011)*, IET, 2011, pp. 610–613 (cit. on p. 1).

[4] M. Elappila, S. Chinara and D. R. Parhi, ‘Survivable path routing in wsn for iot applications,’ *Pervasive and Mobile Computing*, vol. 43, pp. 49–63, 2018 (cit. on p. 1).

[5] J. Shen, A. Wang, C. Wang, P. C. Hung and C.-F. Lai, ‘An efficient centroid-based routing protocol for energy management in wsn-assisted iot,’ *Ieee Access*, vol. 5, pp. 18 469–18 479, 2017 (cit. on p. 1).

[6] H. Harb, A. Makhoul, R. Tawil and A. Jaber, ‘Energy-efficient data aggregation and transfer in periodic sensor networks,’ *IET Wireless Sensor Systems*, vol. 4, no. 4, pp. 149–158, 2014 (cit. on pp. 2–4, 11).

[7] R. C. Shah and J. M. Rabaey, ‘Energy aware routing for low energy ad hoc sensor networks,’ in *2002 IEEE Wireless Communications and Networking Conference Record. WCNC 2002 (Cat. No. 02TH8609)*, IEEE, vol. 1, 2002, pp. 350–355 (cit. on p. 2).

[8] Q. Gao, Y. Zuo, J. Zhang and X.-H. Peng, ‘Improving energy efficiency in a wireless sensor network by combining cooperative mimo with data aggregation,’ *IEEE transactions on vehicular technology*, vol. 59, no. 8, pp. 3956–3965, 2010 (cit. on p. 2).

[9] J. V. Sobral, J. J. Rodrigues, R. A. Rabêlo, K. Saleem and V. Furtado, ‘Loadng-iot: An enhanced routing protocol for internet of things applications over low power networks,’ *Sensors*, vol. 19, no. 1, p. 150, 2019 (cit. on pp. 2, 3, 8).

[10] J. Tripathi, J. C. De Oliveira and J.-P. Vasseur, ‘Proactive versus reactive routing in low power and lossy networks: Performance analysis and scalability improvements,’ *Ad Hoc Networks*, vol. 23, pp. 121–144, 2014 (cit. on p. 2).

[11] A. C. d. V. T Clausen, A. N. J Yi and Y. Igarashi, 'The lightweight on-demand ad hoc distance-vector routing protocol - next generation (loadng),' 2017 (cit. on p. 3).

[12] D. Sasidharan and L. Jacob, 'A framework for the ipv6 based implementation of a reactive routing protocol in ns-3: Case study using loadng,' *Simulation Modelling Practice and Theory*, vol. 82, pp. 32–54, 2018 (cit. on pp. 3–5, 7).

[13] B. Gupta and M. Quamara, 'An overview of internet of things (iot): Architectural aspects, challenges, and protocols,' *Concurrency and Computation: Practice and Experience*, vol. 32, no. 21, e4946, 2020 (cit. on p. 6).

[14] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, 'Energy-efficient communication protocol for wireless microsensor networks,' in *Proceedings of the 33rd annual Hawaii international conference on system sciences*, IEEE, 2000, 10–pp (cit. on p. 6).

[15] C. Karlof and D. Wagner, 'Secure routing in wireless sensor networks: Attacks and countermeasures,' *Ad hoc networks*, vol. 1, no. 2-3, pp. 293–315, 2003 (cit. on p. 6).

[16] T. Clausen, P. Jacquet, C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum and L. Viennot, 'Optimized link state routing protocol (olsr),' 2003 (cit. on p. 7).

[17] T. Clausen, U. Herberg and M. Philipp, 'A critical evaluation of the ipv6 routing protocol for low power and lossy networks (rpl),' in *2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, IEEE, 2011, pp. 365–372 (cit. on p. 7).

[18] J. Yi, T. Clausen and Y. Igarashi, 'Evaluation of routing protocol for low power and lossy networks: Loadng and rpl,' in *2013 IEEE Conference on wireless sensor (ICWISE)*, IEEE, 2013, pp. 19–24 (cit. on pp. 7, 26).

[19] T. Clausen, J. Yi and U. Herberg, 'Lightweight on-demand ad hoc distance-vector routing-next generation (loadng): Protocol, extension, and applicability,' *Computer Networks*, vol. 126, pp. 125–140, 2017 (cit. on p. 7).

[20] J. V. Sobral, J. J. Rodrigues, K. Saleem and J. Al-Muhtadi, 'Performance evaluation of loadng routing protocol in iot p2p and mp2p applications,' in *2016 International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*, IEEE, 2016, pp. 1–6 (cit. on pp. 8, 11).

[21] J. Yi, T. Clausen and A. Bas, 'Smart route request for on-demand route discovery in constrained environments,' in *2012 IEEE International Conference on Wireless Information Technology and Systems (ICWITS)*, IEEE, 2012, pp. 1–4 (cit. on pp. 8, 14).

[22] D. Kumar, 'Performance analysis of energy efficient clustering protocols for maximising lifetime of wireless sensor networks,' *IET Wireless Sensor Systems*, vol. 4, no. 1, pp. 9–16, 2013 (cit. on p. 10).

[23] H. Chen, H. Mineno and T. Mizuno, ‘Adaptive data aggregation scheme in clustered wireless sensor networks,’ *Computer Communications*, vol. 31, no. 15, pp. 3579–3585, 2008 (cit. on p. 11).

[24] K.-W. Fan, S. Liu and P. Sinha, ‘Dynamic forwarding over tree-on-dag for scalable data aggregation in sensor networks,’ *IEEE Transactions on Mobile Computing*, vol. 7, no. 10, pp. 1271–1284, 2008 (cit. on p. 11).

[25] G. Prakash, M. Thejaswini, S. Manjula, K. Venugopal and L. Patnaik, ‘Tree-on-dag for data aggregation in sensor networks,’ *World Academy of Science, Engineering and Technology*, vol. 37, pp. 727–733, 2009 (cit. on p. 11).

[26] J. M. Bahi, A. Makhoul and M. Medlej, ‘Energy efficient in-sensor data cleaning for mining frequent itemsets,’ *Sensors & Transducers*, vol. 14, no. 2, p. 64, 2012 (cit. on p. 11).

[27] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia and M. Dohler, ‘Standardized protocol stack for the internet of (important) things,’ *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1389–1406, 2012 (cit. on p. 11).

[28] Y. Chen, J.-P. Chanet, K.-M. Hou, H. Shi and G. De Sousa, ‘A scalable context-aware objective function (scaof) of routing protocol for agricultural low-power and lossy networks (rpal),’ *Sensors*, vol. 15, no. 8, pp. 19 507–19 540, 2015 (cit. on p. 11).

[29] C. H. Barriquello, G. W. Denardin and A. Campos, ‘A geographic routing approach for ipv6 in large-scale low-power and lossy networks,’ *Computers & Electrical Engineering*, vol. 45, pp. 182–191, 2015 (cit. on p. 11).

[30] C. Perkins, E. Belding-Royer and S. Das, *Rfc3561: Ad hoc on-demand distance vector (aodv) routing*, 2003 (cit. on p. 14).