# Queue Length Aware Dynamic Source Routing Protocol for Mobile Ad Hoc Networks

Agonafir Ayele Wosene

A Thesis Submitted to The department of Computer Science and Engineering
School of Electrical Engineering and Computing

Presented in Partial Fulfillment for the Degree of Master of Science in Computer Science and Engineering

Office of Graduate Studies
Adama Science and Technology University

September, 2024
Adama, Ethiopia

# Queue Length Aware Dynamic Source Routing Protocol for Mobile Ad Hoc Networks

Agonafir Ayele Wosene

Major Advisor: Dr.-Ing. Frezewd Lemma

A Thesis Submitted to The department of Computer Science and Engineering
School of Electrical Engineering and Computing

Presented in Partial Fulfillment for the Degree of Master of Science in Computer Science and Engineering

Office of Graduate Studies
Adama Science and Technology University

September, 2024
Adama, Ethiopia

# DECLARATION

I hereby declare that this Master Thesis entitled **"Queue Length Aware Dynamic Source Routing Protocol for Mobile Ad Hoc Networks"** is my own work and has not been submitted to any university for similar purposes. All sources of materials that are used for this thesis have been duly acknowledged through citation

Agonafir Ayele Wosene ________________ ________________

Name of student Signature Date

# RECOMMENDATION

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled **"Queue Length Aware Dynamic Source Routing Protocol for Mobile Ad Hoc Networks"** prepared under my guidance by **Agonafir Ayele**. submitted in partial fulfillment of the requirements for the degree of Masters of Science in the Department of Computer Science and Engineering Therefore, I recommend the submission of revised version of the thesis to the department following the applicable procedures.

| | | |
|---|---|---|
| Frezewd Lemma (Dr.-Ing.) | ____________________ | ______________ |
| Major Advisor | Signature | Date |

# Approval Page

I, the advisors of the thesis entitled **"Queue Length Aware Dynamic Source Routing Protocol for Mobile Ad Hoc Networks"** by **Agonafir Ayele**, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

| | | |
|---|---|---|
| ______________________ | ______________________ | ________________ |
| Major Advisor | Signature | Date |
| ______________________ | ______________________ | ________________ |
| Co-advisor | Signature | Date |

We, the undersigned, members of the Board of Examiners of the thesis by **Agonafir Ayele** have read and evaluated the thesis entitled **"Queue Length Aware Dynamic Source Routing Protocol for Mobile Ad Hoc Networks"** and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science and Engineering.

| | | |
|---|---|---|
| ______________________________ | ______________________ | ____________________ |
| Chairperson | Signature | Date |
| ______________________________ | ______________________ | ____________________ |
| Reviewer 1 | Signature | Date |
| ______________________________ | ______________________ | ____________________ |
| Reviewer 2 | Signature | Date |

Finally, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

| | | |
|---|---|---|
| ______________________________ | ______________________ | ____________________ |
| Department Head | Signature | Date |
| ______________________________ | ______________________ | ____________________ |
| School Dean | Signature | Date |
| ______________________________ | ______________________ | __________________ |
| Office of Postgraduate Studies, Dean | Signature | Date |

# ACKNOWLEDGEMENT

First and foremost, I thank the Almighty God who provided me with health and strength to make all of this possible. Secondly, I am grateful to Dr. Ing Ferezewed Lemma for his time and unceasing efforts in advising, checking, and providing motivation, helpful, and insightful comments and suggestions based on his outstanding professional experience. Lastly, I thank my family for their support.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ACRONYMS

| | |
|---|---|
| AODV | Ad-hoc on-Demand Distance Vector |
| CBR | Continuous Bit Rate |
| DSDV | Destination Sequenced Distance Vector Routing |
| DSR | Dynamic Source Routing |
| DSR-PM | Dynamic Source Routing Protocol Based on Path Monitoring |
| FTP | File Transfer Protocol |
| GUI | Graphic User Interface |
| HT-AODV | Hop Count and Time based Ad Hoc on Demand Distance Vector Routing |
| MANET | Mobile Ad-hoc Network |
| MET-MFO | Minimum Execution Time Scheduling and Moth Flame Optimization |
| NAM | Network Animator |
| NS-3 | Network Simulator Version Three |
| OLSR | Open Link State Routing |
| OS | Operating System |
| OTCL | Object-Oriented Variant of Tool Command Language |
| PDR | Packet Delivery Ratio |
| QL-AODV | Queue Length Ad-Hoc on-Demand Distance Vector |
| QLA-DSR | Queue Length Aware Dynamic Source Routing |
| QoS | Quality of Service |
| TCP/IP | Transport Control Protocol/Internet Protocol |
| TORA | Temporally Ordered Routing Algorithm |
| UDP | User Datagram Protocol |
| ZRP | Zone Routing Protocol |

# ABSTRACT

*Mobile Ad-hoc Network was developed for short-term data communication in places with poor communication infrastructure; they do not rely on a centralized access point, infrastructure, or administration to meet the requirements for spontaneous network setup. They are distinguished by the use of wireless networks, constantly changing topologies, multi-hop connectivity, and decentralized routing and decision-making. The two most frequently researched on-demand ad hoc routing protocols are DSR and AODV. Previous research has indicated the limitations of these protocols in specific network conditions. Dynamic Source Routing (DSR) is a well-known routing protocol that has gained much attention due to its efficiency in MANETs. However, it does not take the queue length of the nodes into account when selecting a route in the next-hop node which leads to packet drop, congestion, and high delay in the network that reduces network performance. To address this issue, a Queue Length Aware Dynamic Source Routing Protocol (QLA-DSR) chooses next-hop nodes based on their queue lengths before forwarding packets are designed. In addition to the shortest path to the destination, the developed protocol considers the queue length of the intermediate nodes based on their buffer space availability and traffic load while selecting the next hop. In doing so, QLA-DSR aims to minimize the congestion and delay of the network, improving overall performance. Network Simulator (NS-3) is a simulating tool implemented to evaluate and compare its performance with other existing routing protocols DSR and AODV and the performance such as packet delivery ratio, and end-to-end delay, are used to compare QLA-DSR with other existing protocols. The successful implementation and evaluation of the QLA-DSR protocol contributes to developing efficient routing protocols for mobile ad hoc networks. Three simulation scenarios, the impact of varying network density, varying Numbers of Communicating Nodes, and varying Mobility Speeds of the Mobile Nodes are implemented, and performance matrices are evaluated. The finding of the simulation results of QLA-DSR when compared to DSR and AODV shows an average end-to-end delay improvement of 3.56 % and 4.54% respectively with the tradeoff routing overhead.*



*Keywords: MANETs, Latency, Queue length, DSR, QLA-DSR, QoS.*

# CHAPTER ONE

# 1. Introduction

## 1.1. Background of the Study

Wireless communication systems are growing fields that are utilized in different situations where mobility is essential and wires are not practical. Broadly Wireless networks are classified into two groups: wireless networks with infrastructure and wireless networks without infrastructure. A wireless network with infrastructure is a collection of mobile nodes, base stations, controllers, and core network and access points (Rukmani & Ganesan, 2013). In infrastructure-less wireless networks, only mobile nodes exist. In wireless ad hoc networks, nodes can communicate with each other inside as well as outside the network (Bahuguna, Mandoria, & Pranavi, 2013). Frequently changing topology is one feature of mobile ad hoc networks. The fast speed of mobile nodes in ad-hoc networks is one reason why the performance of networks during communication through various routing protocols has been degraded up to a certain limit. To achieve better performance, confusion has been created over which routing protocols, and at what data rates should be used so that maximum throughput can be achieved without losing data packets (Sarao & Sharma, 2022). Different routing protocols perform differently depending upon the nature of the network, the type of network scenario used, and the time for which data communication must be performed. It isn’t easy to declare that a particular routing protocol can perform well in all network conditions.

MANET network was developed for short-term data and communication exchange in places with poor communication infrastructure, like battlegrounds, areas where people need emergency medical care, and areas where people need to evacuate after an earthquake (Agidi, 2023).

In MANET every node can communicate with each other and move randomly in any direction in the network. In MANET every host acts as a router to forward the packet to another node and they act as a host also to send and receive packets.

The inherently dynamic and ad-hoc nature of MANETs makes them suitable for many applications, including emergency response, battlefield communications, sensor networks, emergency medical care, and earthquake and disaster recovery scenarios. It is also used for commercial applications, conferences, car networks, military, personal networking, airports, hospitals, etc.

Mobile Ad-Hoc Network shares common characteristics of wireless networks, such as dynamic network topology, decentralized operations, limited energy, multi-hop communication, and autonomous terminals (Hekmat, 2006). Figure 1.1 below (Bruzgiene, Narbutaite, & Adomkus, 2017) shows the diagram of Mobile the Ad hoc network.

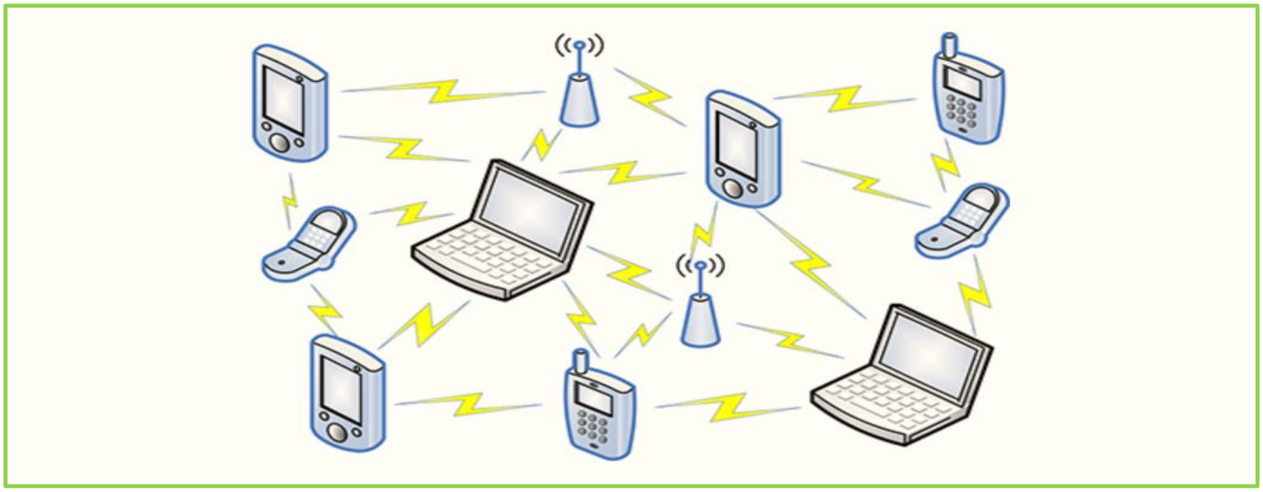


Figure1:1 Mobile Ad hoc network.

The routing protocols for MANETs can be mainly classified into three main categories (Hinds, Ngulube, Zhu, & Al-Aqrabi, 2013). Reactive Routing, Proactive Routing, and Hybrid Routing.

On-demand route discovery mechanism in the reactive routing protocol is utilized, only when a route is needed by the source node (Dugaev, Zinov, Siemens, & Shuvalov, 2015).

Based on source routing, the DSR protocol is an on-demand routing mechanism in which each data packet header in source routing contains the whole list of intermediary nodes packets and this packet must pass through on the way to the destination. Every node is supposed to keep track of the originating route it learned in a route cache. The two stages of DSR to locate and maintain a route are route discovery and maintenance.

Dynamic Source Routing has gained popularity due to its efficacy in MANETs. However, when selecting the next-hop node, it ignores the queue length of the nodes, which leads to packet drops, high delay, congestion, and other problems that reduce network performance.

This study, Queue Length Aware Dynamic Source Routing Protocol (QLA-DSR) is conceived to address these specific challenges. It seeks to give nodes real-time knowledge about surrounding nodes' congestion states which can make sensible next-hop node selections based on queue lengths before packet forwarding. This is achieved by incorporating queue length awareness into the routing decision-making process.

## 1.2. Motivation of the Study

One of the most interesting applications of MANETs in today's communication technology is providing delay-sensitive traffic, which has seen a significant development in use across several domains. Due to this growing technology, several unique characteristics and difficulties arise, including dynamic topology changes, resource limitations, routing overhead, energy consumption, and end-to-end delays. Delay-sensitive applications are one open research field where QoS is not fully addressed. The shortest path is the metric used in most existing MANET routing protocols such as AODV, DSDV, DSR, TORA, etc. (Khader et al., 2013) and is not designed considering QoS support.

The shortcomings of the current MANET routing protocol regarding route selection to ensure quality of service are demonstrated in the next two examples.

**Scenario 1:** Upon reaching the destination after an equivalent number of hops

Figure 1.2 (Temesgen, Yitayal, & Berehan, 2019) illustrates Paths are chosen by the least hop count cost criteria in the current DSR routing algorithm. This indicates that the path chosen for optimization has the fewest hops. The destination node will randomly select the optimum path because both paths have the same amount of hops. However, this approach doesn't address additional QoS measures relevant to route selection.

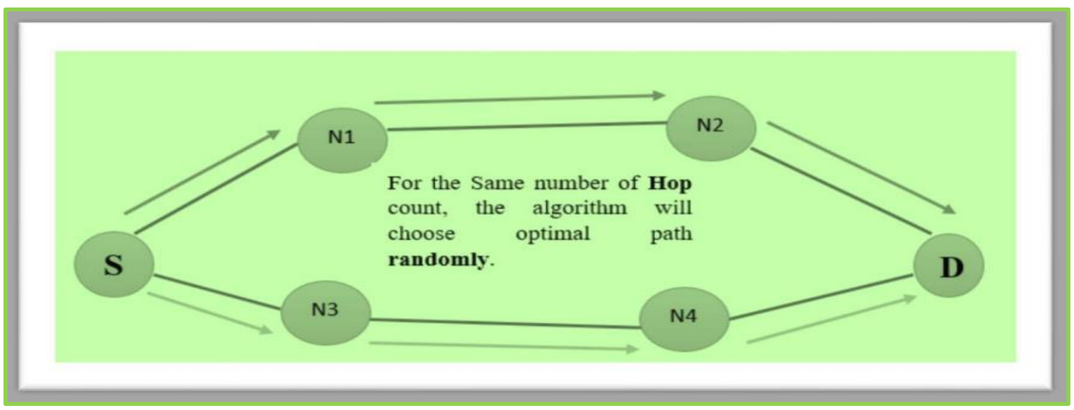



Figure: 1.2 Path Selection Based on Hop Count

**Scenario 2:** When a destination is reached after a different number of hops

In Figure 1.3 (Temesgen et al., 2019) below, Let's assume there are two routes from the source reaching the destination, the destination will choose the first one since it has the fewest hops that is three for the first path, and not select the second path with four hops. Unfortunately, in this case, the chosen approach ignores latency and other QoS criteria. Multimedia real-time applications require prompt communication since they are delay sensitive. Therefore, for MANET routing protocols, especially for delay-sensitive applications, taking queue length into account while choosing a route is an important consideration.

The problem statement proposed is motivated by the previously discussed scenarios.

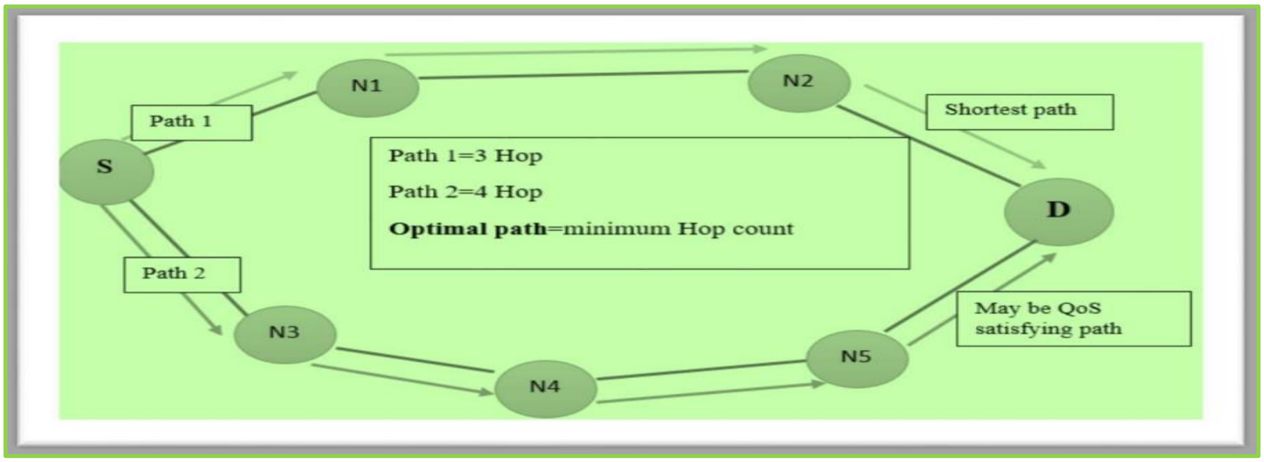



Figure: 1.3 Path Selections with Different Hop Counts

## 1.3. Statement of the Problem

Quality of service support in MANETs has grown more important as real-time applications have increased. However, existing routing protocols face challenges adapting to dynamic topology changes, unpredictable environments, and effectively managing congestion. DSR is one of the existing reactive routing protocols designed to search for only the shortest path with small hop counts without considering real-time network conditions such as congestion at intermediate nodes which affects quality of service performances leading to higher end-to-end delay. The shortest paths may not always provide the best quality performance, particularly while the nodes along these paths are congested. In a dynamic network, where node mobility and traffic load change frequently, DSR lacks mechanisms to adjust routes based on real-time network state, especially concerning delays. Several researchers have proposed to improve the quality of service in MANET routing protocols by considering various parameters. Even if several routing algorithms are developed, achieving data transmission with minimal latency is still a significant challenge and an open research area.

One basic challenge in MANETs is the queue management issue, where nodes maintain and prioritize packets in their buffers to avoid congestion and latency and optimize network performance. To enhance the performance of current DSR routing protocols, this study develops a queue-length-aware DSR routing algorithm that considers node queue length during the route construction process. The latency limitation is one of the QoS requirements that the study mainly focuses on. The goal is to improve the delay performance of the network with a good packet delivery ratio while reducing the overall congestion and packet drops.

## 1.4. Research Questions

This study presents the Queue length aware routing algorithm for MANETs aiming to overcome the limitation addressed in the literature review. We address the following research questions

- How does the QLA-DSR protocol in mobile ad hoc networks address latency and improve network performance?
- What queue management algorithm is used to reduce latency in DSR using the QLA routing algorithm?
- How does the QLA-DSR protocol improve the routing efficiency in delay-sensitive applications compared to existing on-demand routing protocols?

## 1.5. Objectives

### 1.5.1. General Objective

The general objective of this study is to enhance mobile ad hoc networks performance using a dynamic source routing protocol that is aware of the queue length of the node.

### 1.5.2. Specific Objective

- To assess and examine the current mobile ad hoc network routing protocols.
- To improve the performance of the current routing protocols concerning queue length awareness.

- To design and practice a dynamic source routing protocol that uses queue length awareness.
- To compare the proposed protocol's performance with traditional protocols and evaluate its performance regarding latency through simulations.
- To validate the efficiency of the suggested protocol and offer knowledge for the next studies on MANETs.

## 1.6. Significant of the Study

The protocol takes queue lengths into account when making routing decisions, which results in decreased packet loss, lower latency, and more effective use of network resources. By dynamically selecting routes based on queue lengths, the protocol also contributes to energy efficiency by reducing the number of unnecessary retransmissions.

This study QLA-DSR routing protocol has significance in many sectors such as military, universities, commercial, conferencing, education, and emergency services like disaster. And it can also serve as a benchmark for network planners, designers, developers, and researchers working with MANETs.

## 1.7. Expected Outcome of the Study

- Improve Network Performance: The protocol can choose routes dynamically with shorter queue lengths that minimize latency and packet loss. This leads to improved available bandwidth usage with less latency.
- Energy Efficiency: The battery lives of Mobile devices increased by reducing node energy consumption since retransmissions due to congestion are reduced.
- Increased Availability and Reliability: Due to packet loss reduction and packet routes over shorter links, it offers a more available network.
- Better Utilization of Network Resources: The QLA-DSR protocol optimizes the use of network resources by avoiding busy or high-queued nodes.

## 1.8. Scopes and Limitations of the Study

### 1.8.1. Scopes of the Study

- The study involves simulating the QLA-DSR protocol by NS3 collecting and analyzing performance metrics results.
- This study focuses on comparing the performance of QLA-DSR with existing reactive protocols, AODV and DSR.
- Other protocol features and enhancements for very larger networks are outside the scope of this document.

### 1.8.2. Limitations of the Study

- The study's scope is restricted to examining the delay sensitivity of AODV and DSR, two reactive MANET routing protocols.
- The study does not include security issues like authentication, encryption, and privacy.
- The protocol does not guarantee optimal routing at all times but attempts to provide efficient routing based on the current queue lengths of nodes.

## 1.9. Operational Definitions

Some terms that are defined in this study include:

- Congestion Management**:** This strategy involves identifying and alleviating congestion in the network.
- Delay: The average time a packet takes from its source to its destination.
- Flooding: Within this study, flooding refers to the situation in which a mobile node receives a new packet and either retransmits it exactly once, or any newly received packet by a node retransmits it exactly once.
- Node Mobility: This is the movement of nodes inside the MANETs.
- Network Simulator 3 (NS-3): This is a simulating tool used to simulate and test the protocol performance of the real-world network by writing scripts in C++ or Python.

- Queue Length**:** Refers to the buffered packets available on each node at a specific moment.
- Route Discovery: is a process used to find a path from the source node to the destination node in a network.
- Routing Protocol: A standard that determines the path between nodes of networks.

## **1.10.** Organization of the Study

We cover in chapter one, the background of the study, problem statement, research questions, objectives, significance of the study, outcomes of the study, scope, and limitations. Detailed information on routing protocols in MANETs and their application domain is covered in chapter two. The study approach, methodology, and simulation tool are explained in chapter three. The architecture, design, flow chart diagram, and implementation of the suggested queue length-aware dynamic source routing protocol are covered in chapter four. The results of simulations are discussed and explained in chapter five. In chapter six, findings and future works are discussed.

CHAPTER TWO

# 2. Literature Reviews and Related Works

This chapter includes an overview of wireless networks, application areas, characteristics, protocols, and routing methods. It also introduces QoS in MANET and the current quality of service provisioning in the routing protocols and finally related works are described.

## 2.1. Wireless Network

Networks of computing devices or nodes connected by wireless media and possessing computational capacity are called wireless networks. With or without the aid of a wireless infrastructure, these wireless nodes can interact with one another with flexibility. Computer networks are divided into two main categories, wired and wireless. Wireless has two more subclasses, Infrastructure-based and Infrastructure-less. MANET is a kind of infrastructure-less wireless communication network formed by wireless self-configuring nodes without necessarily having a fixed infrastructure (Macker & Corson, 2002). Routing strategies represent one of the main study topics in MANETs. Mobility, overhead, scalability, security protocols, Internet gateway discovery, and quality support are among the challenges associated with MANET routing. QoS routing is a significant topic of study History of Mobile Ad Hoc Network (A. Boukerche, Câmara, Loureiro, & Figueiredo, 2008).

## 2.2. History of Mobile Ad Hoc Network

The Mobile Ad Hoc Network (MANET) was first developed as a packet radio network and was funded in the early 1970s by the Defense Advanced Research Projects Agency (DARPA). Many wireless terminals that could connect on the battlefields were part of a project they worked on. It is interesting to note that these early packet radio systems preexisted the Internet, and indeed were part of the motivation of the original Internet Protocol suite (Malwe, Taneja, & Biswas, 2017). An independent set of nodes that travel and change network connectivity is called a mobile ad hoc network (Simaremare, Abouaissa, Sari, & Lorenz, 2015). Unlike cellular wireless networks, mobile area networks (MANETs) lack fixed or static infrastructure, as well as centralized management. Because of the flexibility that a MANET provides, the network can be built anywhere, at any time, as long as two or more nodes are connected to communicate with one another, either directly when they are in radio range of one another or via intermediate mobile nodes (Conti & Giordano, 2014).

## 2.3. Overview of Mobile Ad Hoc Network

A non-concentrated multi-hop distant network, or MANET, is a group of distributed, autonomous mobile nodes that form a temporary network. Since nodes in this network can move around freely and randomly without forecasting, MANETs have a dynamic topology (Conti & Giordano, 2014). There isn't a centralized administration role in MANETs. Since it is a decentralized architecture, all nodes work together to forward data from the source to the destination. Such characteristics enable MANETs to be used in different applications, including military communications, health care, and disaster situations. (Perkins, 2001), & (Kannhavong, Nakayama, Nemoto, Kato, & Jamalipour, 2007).

The network layer is responsible for the routing mechanism in MANET which is the process of determining the route for transferring packets from source to destination. Currently, several routing protocols are used in MANETs and are classified into three main groups depending on their topology: proactive, reactive, and hybrid routing protocols.

The proactive routing scheme is where each mobile node stores the collected information about the active routes in regularly updated routing tables (Prasath & Sreemathy, 2019). Reactive routing works by having any sender node begin searching for and identifying the path only when it has to send a data packet to a particular sink (Conti & Giordano, 2014). The hybrid routing method is based on integrating proactive and reactive approaches (Prasath & Sreemathy, 2019).

## 2.4. Features of MANET

In a MANET, nodes cooperate as independent terminals that can work as routers or hosts, which makes it challenging to distinguish between switches and endpoints. Multi-hop routing allows data packets to be sent from a source to a destination using intermediate nodes. Because of the dynamic nature of the network architecture, mobile nodes can build and change routing patterns as they move, creating their networks in the process. Furthermore, varying link capacity is a limitation because the bandwidth is restricted to directed networks. After all, shared communication routes are sensitive to noise, fading, and interference across several sessions. Paths frequently cross several heterogeneous wireless networks. Last, MANET nodes are usually portable, light devices with limited CPU processing capacity, little memory capacity, and low power storage.

## 2.5. Applications Area of MANET

Ad hoc networking is becoming more significant due to its growing number of widely used applications, which coincide with the rise in portable devices and the advancement of wireless communication. Without infrastructure or fixable networks, ad hoc networking may be used anywhere, at any time, and it makes it simple to add and remove devices from the network while still enabling devices to stay connected to it. Figure: 2.1 (Khan, Olanrewaju, Anwar, Najeeb, & Yaacob, 2018) shows application areas of MANET.


Figure: 2.1 Applications of MANET

Mobile Ad Hoc Networks (MANETs) are used in different areas of application, like military operations, where soldiers can communicate in the field, in disaster-stricken areas; MANETs facilitate communication when traditional infrastructures of communications are damaged. Additionally, they are employed in sensor networks to monitor environmental characteristics through the ad hoc deployment of sensors that measure factors like temperature and contamination levels. Collaborative robotics also benefits from MANETs, enabling multiple robots to communicate and collaborate without centralized control. MANET is used in social networks, like conferences and concerts, to allow participants to connect and share information. They can also be used in wildlife tracking, where mobile sensors or tags on animals form networks to transmit data on animal behavior, migration patterns, and habitat usage of animals.

## 2.6. MANETs Protocol Stack

The protocol stack for MANETs may incorporate extra protocols and procedures in addition to the Internet Protocol (IP) suite to handle the difficulties of mobile and dynamic ad hoc nature,

The protocol stack of MANET in Figure 2.2 (Temesgen et al., 2019) is like the TCP/IP suite but with some differences; the main distinction between these two protocols stacks lies within the network layer. Ad hoc routing is handled by the network layer successively and is split into network and ad hoc routing.

Five layers of the OSI model are implemented in the MANET network architecture model. The presentation, session, and application layers are combined into one by the application layer. The most crucial layer is the network layer, which houses the multicast and unicast routing protocols. The physical layer manages the transmission and reception of the raw bit streams of radio frequency via the physical media. It contains guidelines for the physical attributes of the wireless communication channel. Error detection, framing, and addressing of transmitted data are managed by the data link layer. It guarantees point-to-point communication amongst adjacent nodes and incorporates IEEE 802.11 (Wi-Fi) to enable wireless communication. In MANETs, the network layer controls packet routing from the source to the destination across the network, and specific routing protocols are employed because of the network architecture's frequent and dynamic changes. The transport layer provides an end-to-end connection and reliable delivery services between the source and the destination. Therefore, the transport layer is known as the end-to-end layer. This layer uses Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). Since TCP depends on a stable and reliable network; it may have issues with MANETs. The connections and sessions between nodes are managed by the session layer. In actual use, this layer and the transport layer are frequently merged. It is responsible for Presentation Layer handling Encryption, compression, and data format translation. Due to the dynamic and sometimes unsafe nature of MANETs, encryption and security measures may be especially crucial. An Application Layer contains network applications, including those specific to MANETs like file sharing, video streaming, and other distributed and cooperative applications.

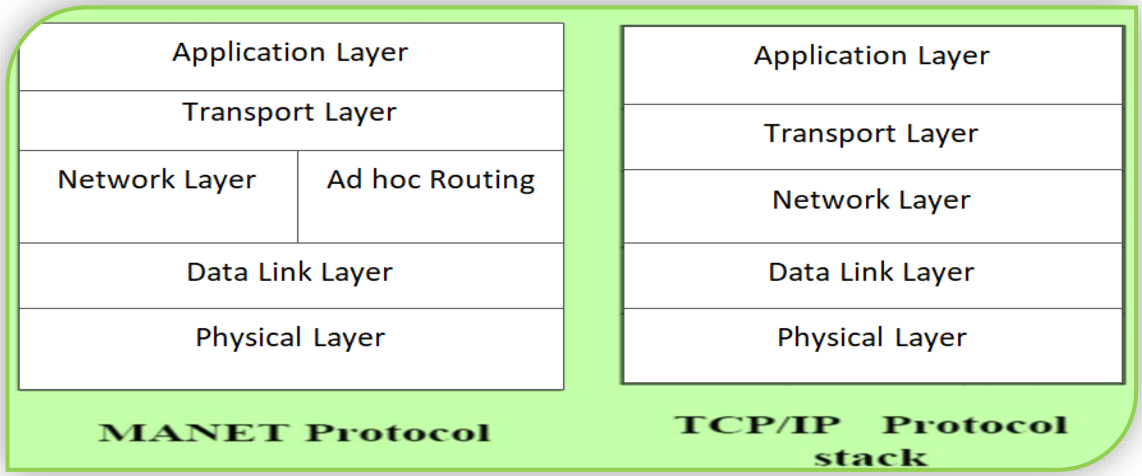



Figure: 2.2 Protocol Stacks of MANET

## 2.7. MANETs Technology

Hardware for wireless networking requires fundamental innovation in data transport and radio frequency. IEEE 802.11 standard describes every facet of radio frequency for ad hoc networks based on 802.11 technologies. The most widely used versions of the standard are IEEE 802.11 b, a, n, and g. Moreover, two more technologies that can be used in ad hoc communications are Bluetooth and HiperLAN2 (Kannhavong et al., 2007; Vasiliou Alexandros, 2006)

## 2.8. MANETs Architecture

The dynamic and self-configuring nature of MANETs allows nodes to communicate with each other without infrastructure. The MANET architecture arranges nodes, functional elements, and communication protocols to facilitate effective and efficient communication inside the network. The architecture of MANET is divided into three main parts and shown in Figure 2.3 below (Mirza, 2018)

- Networking: In the MANET architecture, most of the fundamental functionalities of networking protocols are redesign to adapt a self-configuring, dynamic, and unstable peer-to-peer communication environment. The primary objective of networking protocols is to utilize one-hop transmission services facilitated by the underlying technologies to progress towards end-to-end reliable services from the sender to the receiver. To establish end-to-end communication, the sender must ascertain the receiver within the network. The primary role of a location service is to dynamically associate the address of the receiver device with its current location in the network.

- Enabling Technologies: Depending on the coverage area, enabling technologies are further separated. One of these is the BAN (Body Area Network), which connects wearable computers within a one to two-meter range. In contrast, WLAN (Wireless Local Area Network) links a building or collection of buildings with a communication range of 100 to 500 meters. PAN (Personal Area Network) connects mobile phones and stationary devices with a communication range of up to 10 meters.
- Middleware & Application: Wireless technologies such as WiFi, Bluetooth, IEEE 802.11, and WiMAX play a significant role in promoting the implementation of ad hoc technology and novel ad hoc networking applications. This is especially true in specialized domains like emergency services, disaster recovery, and environment monitoring. Modern mobile ad hoc frameworks rely on individual applications to handle all necessary services rather than integrating middleware.

The key components of MANET's architecture are shown in Figure 2.3 below (Mirza, 2018)

- Nodes: Nodes are end-user devices such as computers, cellphones, tablets, sensors, and other wirelessly enabled devices.
- Network Topology: Because of node mobility, MANETs feature a dynamic and changing topology. Nodes can be created, connected, and disconnected as they move throughout the network. A graph can show the topology, with nodes serving as the vertices and links serving as the communication channels.
- Routing Protocols: Routing protocols are crucial because they are the route through which data can be transmitted between nodes.
- Network Layer: The network layer is responsible for addressing, routing, and forwarding data packets inside the MANET. It works with routing protocols to ensure efficiency.
- Link Layer: Responsible for controlling communication between surrounding nodes. It contains protocols for link formation, fault detection, and rectification. IEEE 802.11 (Wi-Fi) is a commonly used link layer protocol in wireless networks.

- Transport Layer: Provides end-to-end communication and reliability of MANETs. Mobility Management: Since it deals with node mobility inside the network, it is a crucial component of MANET architecture. It covers methods for monitoring node movement, maintaining connectivity when nodes move, and updating routing data.
- Security Mechanisms: MANETs are vulnerable to various threats, and security mechanisms are an integral part of the architecture. This includes authentication, encryption, and intrusion detection mechanisms to protect the network from malicious.
- QoS (Quality of Service): Quality of Service standards, which guarantee that specific performance requirements, such as latency, bandwidth, and dependability, are satisfied for different kinds of networked applications, may be supported by MANETs.
- Application Layer: Network applications that utilize the MANET architecture are found in the application layer. File sharing, chat, teamwork, and apps made especially for mobile ad hoc scenarios are a few examples of these Apps. Figure 2.3 (Mirza, 2018) shows Architecture of MANET.

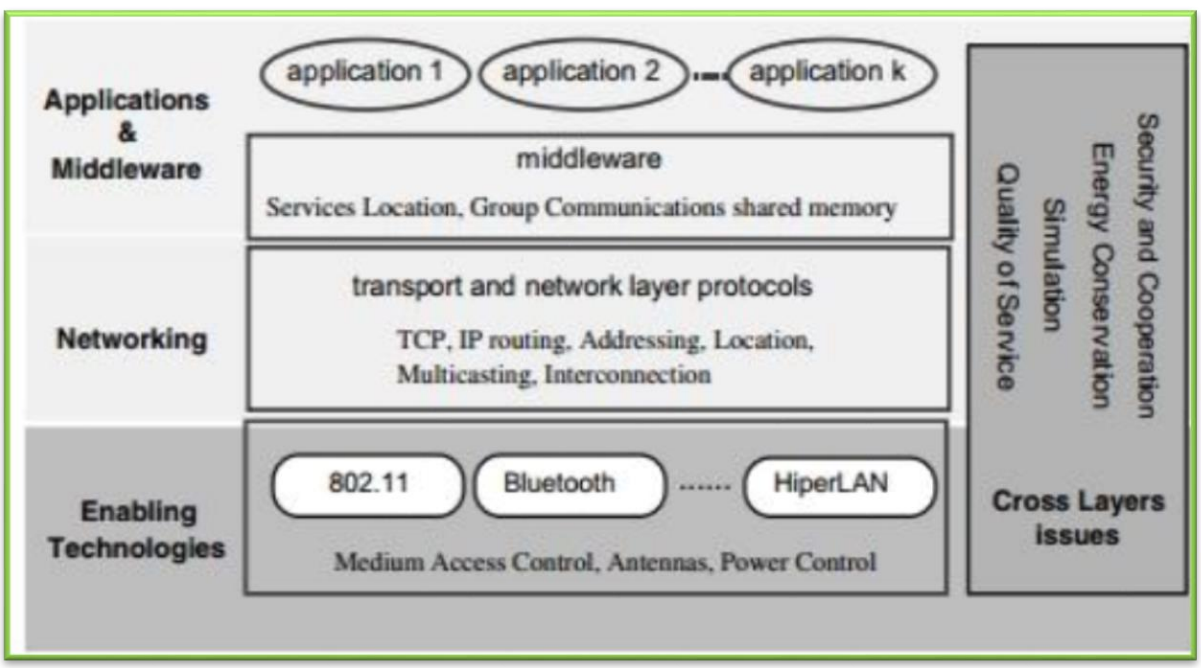



Figure: 2.3 Architecture of MANET

## 2.9. MANET Characteristics and Complexities

Here are several characteristics and complexities unique to MANET.

Mobile Ad Hoc Networks (MANETs) are characterized by several unique features that determine how they function and operate. Since they are infrastructure-less and autonomous, they are set up quickly without the need for centralized management and established infrastructure. Nodes communicate with one another and adjust to topology changes and mobility regularly. Additionally, to extend the network's lifespan, optimal resource usage is required due to limited resources like battery power and computing capacity. The mobility of nodes, which can join and exit the network, leads to constantly changing network topologies. Bandwidth optimization is essential, as wireless links typically offer lower capacity than wired connections. MANETs also utilize multi-hop routing, establishing multiple paths between source and destination nodes to enhance data transmission, reliability and facilitate load balancing. Finally, the distributed operation characteristic of MANETs requires that routing protocols function in a decentralized, ensuring adaptability and resilience in network communication (Kannhavong et al., 2007; Malwe et al., 2017)

Mobility Challenges that frequent topology changes require adaptive routing protocols. So in MANETs node mobility introduces challenges in maintaining stable and efficient communication paths. Security Concerns are another complex issue that MANETs are susceptible to attacks, data tampering, and eavesdropping, among other security threats. Due to topology change and scarce resources, ensuring QoS including satisfying latency, bandwidth, and reliability requirements is difficult that is called the Quality-of-Service issue, and finally, Cross-Layer Interactions are usually necessary for MANET performance optimization. Although difficult, coordinating operations between several protocol stack layers is important for effective network operation.

## 2.10. MANET Advantages and Disadvantages

### 2.10.1.Advantages of Mobile Ad hoc Network

There are many reasons to use ad hoc rather than infrastructure. Infrastructure-less and lower cost are advantages of MANET. Using a service from an infrastructure can be expensive for specific applications, but MANET can easily deploy. Decentralized and robust is another advantage if one node leaves the network or is not working, you can still have connectivity to other nodes and can use these nodes to multi-hop your message to the destination nodes. Spontaneous infrastructure malfunction of network infrastructure is sometimes not avoidable and difficult to repair or replace the malfunctioning infrastructure in a short time, establishing an ad hoc is a good deal in such a situation. Mobility is another advantage of MANET in military networks, emergency/rescue operations, and others (Sampada G. D. & Taha, 2015).

### 2.10.2. Disadvantages of Mobile Ad hoc Network:

Dynamic topology and node mobility pose challenges for routing protocols in MANETs. Maintaining efficient and stable routes becomes complex, leading to potential delays and packet losses. Providing different quality service levels in a constantly changing environment is a challenge. In addition to communication within an ad hoc network, internetworking between MANET and fixed networks is expected in many cases. The coexistence of routing protocols in such a mobile device challenges harmonious mobility management. MANETs are vulnerable to security threats, including eavesdropping, data tampering, and denial-of-service attacks. For most lightweight mobile terminals, the communication functions should be optimized for less power consumption. Mobility, security, and resource limits are among the issues that must be addressed to design efficient protocols for MANETs (Sampada G. D. & Taha, 2015)

## 2.11. Classification of Routing Protocols in MANET

Routing protocol is a set of rules that govern the path of message packets from source to destination in a network. Proactive, reactive, and hybrid are the three basic categories into which MANET routing protocols are classified. The method used to gather route information is the primary distinction between the proactive and reactive categories. Reactive protocol, also known as on-demand routing protocol, gathers routing information only when needed through route discovery (Bisoyi, Sahu, & Raman, 2010). On the other hand, proactive or table-driven protocols constantly propagate routing information.

Hybrid routing protocol combines the advantages of proactive and reactive routing protocols (Lakhtaria, 2010). First, some proactively prospected routes are constructed, and then reactive flooding is used to meet the demand from further activated nodes.

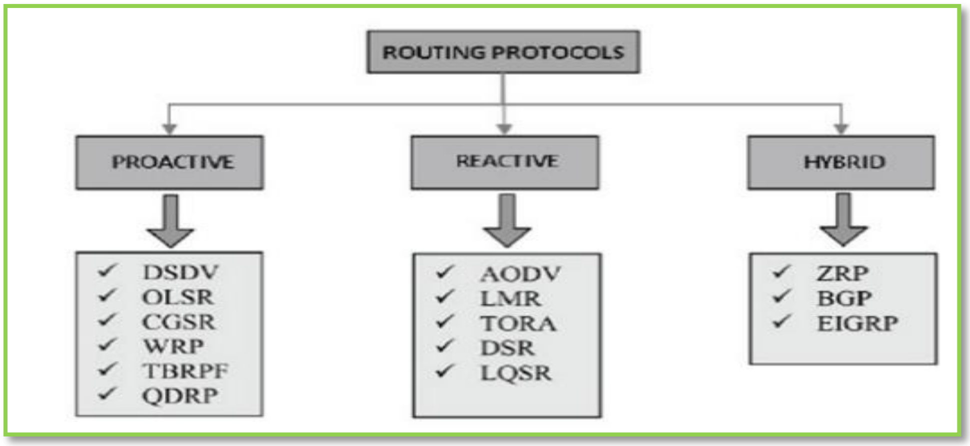



Figure: 2.4 Routing Protocols in MANET

In Figure 2.4 above (Mirza, 2018)depending on the mobility, number of nodes involved, node density, underlying link layer technology, general features of the environment, and applications being supported, each of these three fundamental types has pros and cons and can be used in ad hoc networks.

### 2.11.1. Proactive Routing Protocols

Before forwarding, proactive routing methods ascertain the topology of the network. Every node constructs a routing table that it can use to determine a path to a destination. Routing protocols are based on the periodic exchange of routing information between the various nodes. When there's no traffic to route, it's interesting to keep the network quiet in areas with limited power and bandwidth.

❖ Destination-Sequenced Distance Vector Routing (DSDV)

The conventional Routing Information Protocol (RIP) is modified to provide Destination Sequenced Distance Vector Routing (DSDV) for particularly designated systems directing. Each course table part of the standard RIP has an additional trait, the grouping number. Every route table entry at every node in this method has an additional header field called the sequence number (A. F. M. Boukerche et al., 2011). Every node maintains a routing table, which it uses to send packets to other node stations around the network. In the routing table, list all the accessible destinations along with the number of hops needed to get there. Every mobile station in the network must continuously promote its unique routing table to every neighbor to comply with

DSDV. Make sure that every node in the network can find its neighbors; regular advertisements are necessary because the items in the table can change extremely quickly. Each node will broadcast data that includes the destination address, the number of hops needed to reach the destination, the new sequence number originally stamped by the destination, and the node's new sequence number.

According to surveys on routing protocols, networks with fewer nodes and less mobility exhibit excellent performance levels for DSDV. Routing tables get larger, and more bandwidth is needed to update them as the number of nodes in the network increases. DSDV's primary flaw is that it will result in excessive overhead.

- Optimized Link State Routing Protocol (OLSR)

The Optimized Link State Routing Protocol (OLSR) is a routing protocol created especially for mobile ad hoc networks, while it can also be utilized on other wireless ad hoc networks. OLSR uses hello and topology control (TC) messages as a proactive routing protocol to locate and distribute link state information throughout the mobile ad hoc network. Individual nodes use this topological information to determine each node's next hop destination using the shortest hop forwarding pathways.

### 2.11.2. Reactive Routing Protocols

Reactive routing protocol creates routes on demand rather than maintaining them. It finds a route by flooding the network with Route Request packets (Manickavelu & Vaidyanathan, 2014). In the context of high mobility and network density, these routing methods perform better in packet delivery ratio, end-to-end delay, and routing overhead (Johnson David B.and Maltz, 1996). In contrast to other routing protocols, they require comparatively less storage, and the routes are only available when needed.

- Ad Hoc on Demand Distance Vector Routing Protocol (AODV)

Another reactive routing technique that creates routes on demand is called AODV. To store active routes, it keeps routing tables updated at nodes. The on-demand method is used by AODV to locate and identify a specific route. When a node needs to send data, AODV finds the path to the destination through route discovery, which uses control messages like route request (RREQ) and route reply (RREP). Neighbor nodes in AODV store their next-hop neighbor's route information. As a result, AODV can determine the safest route and shortest distance. A source node sends out a route request message to all of its nearby neighbors to find a path. The route request packet is then forwarded by neighbors to other neighbors. This procedure keeps going till the goal is accomplished. The route information will be deleted after the data packet has been transferred. Only when a node needs to send or receive data, does AODV find and identify a route. The source node will get a route error (RERR) message in the event of transmission failure, or link failure, instructing it to select a different path. The primary benefit of AODV is that it finds and identifies a route based on demand. If the source sequence number is old and the intermediate nodes have a higher sequence number unrelated to the most recent destination sequence number, then AODV confronts a severe disadvantage since the intermediate nodes may forward to unreliable routes.

When a loop-free, current route is needed for QoS routing, AODV is the right choice. If a source node has routes to the destination in its routing table, it will transmit packets to the destination to create a communication session. If not, it broadcasts a route request message to start the path-finding process.

AODV, an on-demand routing technology, tracks nearby nodes by sending periodic Hello Message broadcasts, and this periodic propagation results in a network overhead. When necessary, an AODV routing path to a specific packet is found. Interactive applications may perform worse because of this initial search delay. In the same way, the path's quality is unknown until the call is set. It can only be found when establishing the path. Furthermore, all intermediary nodes in an active session check the path quality, even at the expense of extra overhead and latency (Nassir, Refoufi, & Harrag, 2019).

### ❖ Destination Source Routing (DSR)

DSR is a source routing protocol that the entire route information is contained in the packet header. To store previously found routes depends on route caches at nodes. When using DSR, the route pathways are found when a packet is sent from the source to an ad hoc network destination node (A. F. M. Boukerche et al., 2011). At every stage of the network, DSR doesn't require any regular updates. Source route information is carried in the packet header of a data packet and route Cache holds these routes. When improved QoS characteristics, such as packet delivery, overhead, and reliability, need to be considered for packet routing, DSR can be used in MANETs to offer soft QoS assurances in various MANET applications. With the least network overhead possible, the protocol gives a dependable path for packet transfer. Host caches are repeatedly updated as a result, leading to erratic responses. The propagation of all query messages that flood the network cannot be stopped by early queries. Therefore, control and message packets grow in size along with the network. Route discovery and route maintenance are the two mechanisms that make up DSR. In the route discovery stage, a route request (RREQ) packet mainly consists of a unique ID, source address, and destination address (Chatterjee & Das, 2015). After receiving the RREQ, each intermediate participating node will determine whether or not it knows the destination. If not, the node will rebroadcast after adding its address and keep trying until the destination node receives the RREQ. A route reply (RREP) packet will then be sent back to the source by this target node. This packet contains route information that was retrieved from the RREQ and cached by the source to meet the communication requirement (Chatterjee & Das, 2015; Zhang, iu, Liu, Zhang, & Cui, 2018). Figure: 2.5 describes the route discovery phase, where N1 and N7 are the source and the destination respectively. To identify the route, two RREQs with unique different IDs will be sent, and two routes will be discovered. In this case, route 1 will be selected since it has the minimum hop count (Almazok & Bilgehan, 2020).

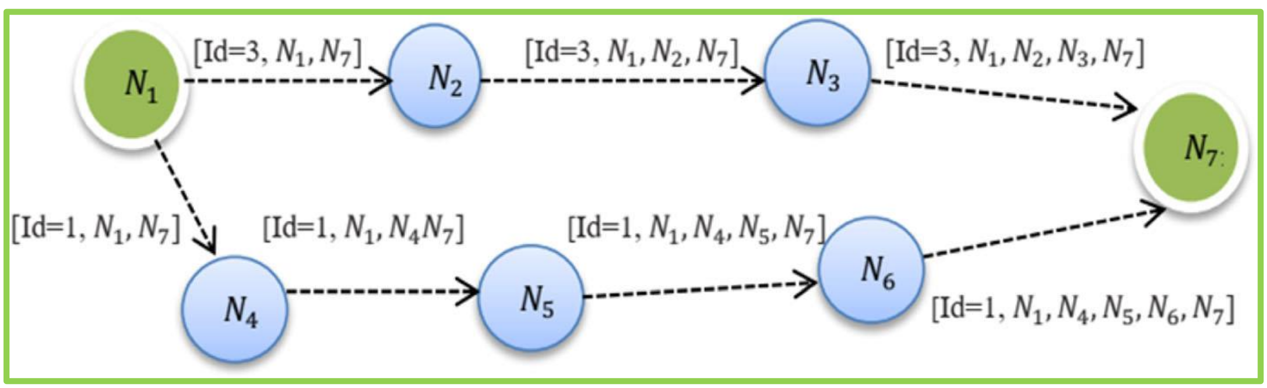



Figure: 2.5 Route Discovery using RREQ Packets

Route maintenance is a method used by DSR to locate alternate routes in case the present path fails to continue transmitting packets. At this point, the source gets a route error (RERR) packet and expunges the old route from its cache to begin the process of finding a new one (Abd, Alsaqour, & Abdelhaq, 2017). Figure: 2.6 describes the route maintenance mechanism. It shows that when the link between node N3 and destination node N7 is disconnected, the intermediate node N3 sends a RERR message to the source node N1. Figure 2.6 (Almazok & Bilgehan, 2020)shows Route Maintenance using RERR Packets .

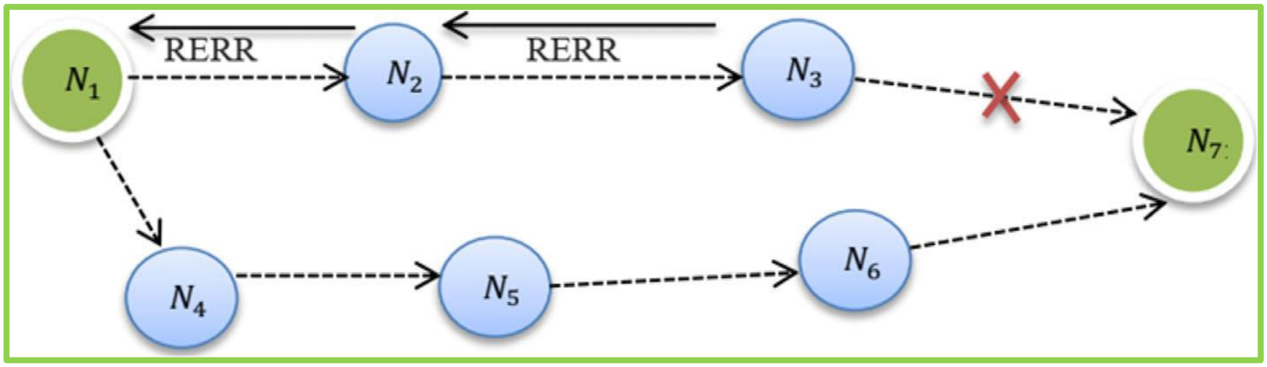



Figure: 2.6 Route Maintenance using RERR Packets

### 2.11.3. Hybrid Routing Protocols

The advantages of proactive and reactive routing strategies are combined in these protocol types. The primary drawback of hybrid routing protocols is that they require more memory and power since the nodes with high-level topological information retain more routing information. The Zone Routing Protocol (ZRP) is example of hybrid routing protocol that creates zones in the topology and uses various routing protocols within and between the zones following the advantages and disadvantages of each protocol (Clausen, Dearlove, Jacquet, & Herberg, 2014).

## 2.12. Qos Issue in MANETs

The increased popularity of MANETs has led to a rise in real-time multimedia applications. Many multimedia applications have special QoS requirements, such as the lowest possible packet loss probability, accessible bandwidth, tolerated jitter, and minimum end-to-end delay. A set of services a network ought to provide is known as quality of service. The specific properties of MANETs make it difficult to provide Quality of Service. While maintaining standard networks might be challenging wireless networks, MANETs, and other networks present additional difficulties because of their characteristics.
(Kalla et al., 2016).

Because different applications have distinct needs, each application has variable requirements for services and factors. Timing is a crucial aspect in multimedia applications (real-time data flow), for instance, where data rate and delay are important factors. QoS parameters can be broadly classified into three groups. Additive refers to the total of the metrics, such as jitter and end-to-end delay, on all connections in the path. For instance, the total delays at each link determine the path's end-to-end delay. Multiplicative represents the product of the metric values on all links over a path. For example, the probability of packet loss p (u, v) for a packet to reach V from U is the product of packet loss probabilities at each intermediate individual link. Lastly, concave takes into account the path's minimal metric value. Concave criteria include those related to bandwidth, security, and end-to-end reliability.

Most of the existing MANET routing protocols are not QoS aware, and because of this various scholar proposed improvement mechanisms for MANET (Vasiliou Alexandros, 2006). However, achieving the necessary level of it for real-time applications continues to be difficult. Figure 2.7 (Temesgen et al., 2019) clearly illustrates how routing affects quality of service support routing protocols. That is, although the latter employs quality of service-satisfying protocols, the former does not display clear video because it uses non-QoS-supporting routing protocols.

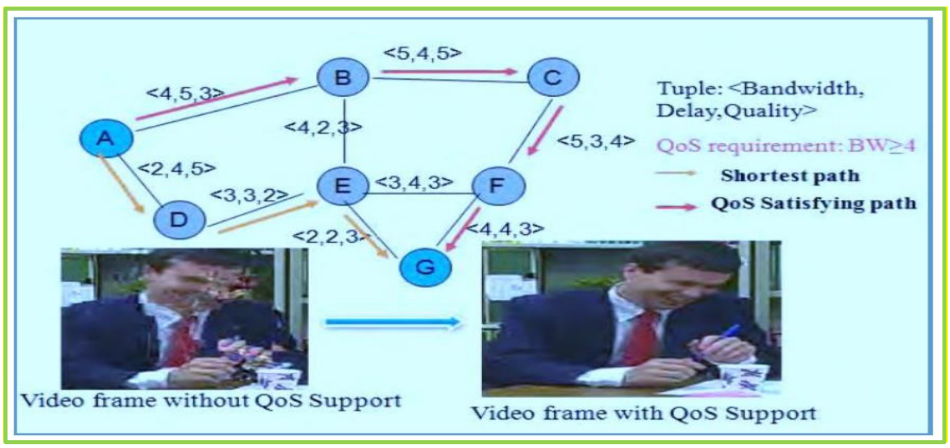



Figure: 2.7 Video Streaming Samples

## 2.13. Related Work

The primary goal is to provide relevant research on the quality of service routing protocol for MANETs in DSR. Wireless communication amongst mobile users has grown in popularity more recently than ever. Multiple research directions are suggested to improve the services offered by MANET apps, quality of service support in Ad-Hoc networks have drawn a lot of attention in the wireless community due to the growing popularity of multimedia applications and the possible commercial use of MANETs. Many of the current MANET routing protocols were created without taking quality metrics into consideration when creating routes, different ways of

improving over DSR routing protocols have been put to support delay-sensitive applications where QoS requirements are essential.
The author (Temesgen et al., 2019) Proposed a QLA-DSR routing protocol based on the nodes queue length approach and performs better in terms of packet delivery ratio, and end-to-end delay with reasonable slight normalized routing overhead increment compared to the AODV and HT-AODV routing protocols but does not experiment on QLA-DSR protocol.

The authors (Almazok & Bilgehan, 2020) proposed a novel and efficient version of DSR approach named MET-MFO DSR. The MET-MFO DSR routing algorithm is based on the applications of a hybrid approach using MET scheduling and MFO scheme. This improved version of the DSR provides an improved routing mechanism by decreasing the energy consumption, which increases the network lifetime, and reduces route failures. It focuses on energy level and minimum energy consumption in addition to hop count but does not consider congested routes by queue length awareness. Dynamic source routing protocol based on path reliability and monitoring repair mechanism model performs data transmission by the best reliability path. The link information is monitored during transmission and broken links are repaired on time to ensure communication stability and reliability and thus improve the data transmission efficiency. DSR-PM protocol reduces overhead, packet loss, delay, and network throughput, and provides better communication performance. Here, only a stable and reliable path is selected but does not consider congestion routes that cause latency (Liang Qing AND Lin, 2021).

The authors (Ma, Lee, Li, & Liu, 2023) proposed the improved DSR protocol that can control congestion using Hello messages to find the congestion status of links. The simulation results show that the DRS-p effectively improves the packet delivery ratio, end-to-end delay, routing overhead, and energy consumption. Even though it improves significantly, the proposed QLA-DSR considers congested routes in addition to the shortest path.

This paper (Johari & Mishra, 2020) has presented the MATLAB Sims-Events approach for building the model to limit the queuing delay. The model gives end-to-end reliable packet delivery in a short time frame and enhances throughput. It uses different processors for different applications, so it uses more resources for queuing to analyze the cellular network in terms of video, audio, and text.

Table: 2.1 Related Works

| No | Title | Author and year of publication | Strength of the study | Limitations of the study |
|---|---|---|---|---|
| 1 | Queue Length Aware Ad Hoc on Demand Distance Vector Routing Protocol for mobile Ad Hoc networks | (Temesgen et al., 2019) | It approaches performs better in terms of packet delivery ratio and end-to-end delay with reasonably slight normalized routing overhead increment compared to the AODV and HT-AODV | It does not experiment on the DSR protocol |
| 2 | A novel dynamic source routing (DSR) protocol based on minimum execution time scheduling and moth flame optimization (MET-MFO) | (Almazok & Bilgehan, 2020) | This improved version of DSR called the (MET-MFO DSR) algorithm provides an improved routing mechanism by decreasing the energy consumption, which increases the network lifetime and reduces route failures. | It focuses on minimum energy consumption in addition to hop count but does not consider congested routes by queue length awareness. |
| 3 | A dynamic source routing protocol based on path reliability and link monitoring repair | (Liang Qing AND Lin, 2021) | The model performs data transmission by filtering the best reliability path. Results show that the DSR-PM protocol effectively reduces parameters such as overhead, packet loss, and delay, improves network throughput, and provides better performance. | Only a stable and reliable path is selected but does not consider congestion routes that cause latency. |

| No | Title | Author and year of publication | Strength of the study | Limitations of the study |
|---|---|---|---|---|
| 4 | Improvement and Implementation of an Adaptive Congestion DSR Routing Algorithm on Chip. | (Ma et al., 2023) | Propose an improved DSR that has congestion control ability by using Hello messages. | Even though it improves significantly, the proposed QLA-DSR will outperform it by considering a congested route. |
| 5 | An Enhanced Queuing Model for Route Optimization in Ad-Hoc Networks Based on Simulation. | (Johari & Mishra, 2020) | Propose the route selection with the least delay while utilizing the specific application with various clients. However, it has similar data, such as text, audio, or video. By queue, it minimizes the end-to-end delay, and the model gives packet delivery in a short time frame. | It uses different processors for different applications, so it uses more resources for queuing. |

Table: 2.2 Parameter used and values

| Parameter | Description | Related Works | | | |
|---|---|---|---|---|---|
| | | MET-MFODSR | DSR-PM | QL-AODV | DSR-p |
| No of node | Total number of nodes in the mobile ad hoc network | 50 | 50 | 10, 20, 30 | 25 |
| Packet size | the total amount of data (in bytes or bits) contained in a packet | 256Bytes | 512Bytes | 512Bytes | 200 - 1000Bytes |
| Total simulation time | Total duration of the simulation | 50s | 100s | 30s | 3600s |
| Simulation area | The physical or virtual space within which nodes operate and interact during a simulation | 1670m × 970m | 2000mx2000m | 1000mx1000m | 100mx100m |
| Node speed | Speed at which the mobile nodes are moving in the network | 5 m/s-20 m/s) | 10m/s-100m/s | 1m/s-10m/s | 15m/s,25m/s |
| Queue Length | Length of the packet queue at a node | 50 | 50 | 50 | 15 |
| Source | | (Almazok& Bilgehan, 2020) | (Liang Qing AND Lin, 2021) | (Temesgen et al., 2019) | (Johari & Mishra, 2020) |
| Dependent variables | | | | | |
| PDR | Ratio of successfully delivered packets to send packets | 90%-100% | 80%-100% | 75%-90% | 60-%100% |
| End-to-End Delay | Time taken for a packet to travel from source to destination | 0-0.4s | 0-0. 35s | 0.1-0.9s | 0.1-0.25s |

## Summary

In summary, although there has been different research done on MANETs using routing protocols, the methods still suffer certain shortcomings, it is not discussed about QoS considering Queue length aware of DSR in MANET. We identify the gaps in the previous work that do not include quality of service in the DSR routing protocol and, hence, in the next chapter, we discuss datasets, methodology, and the tools used for the study and explain the proposed system in detail.

CHAPTER THREE

# 3. Research Methodology

The chapter carefully examines the approach the study needed, and the methodology used to ensure the study yielded reliant MANET based on a routing QLA-DSR. The chapter also includes a data set; tools used and research design. It also justifies the quality of the research and looks at the ethical considerations made when conducting the Research.

The methodology for developing QLA-DSR Protocol for Mobile Ad Hoc Networks is divided into the following steps and the overall methodological procedure for the study is shown below in Figure: 3.1.

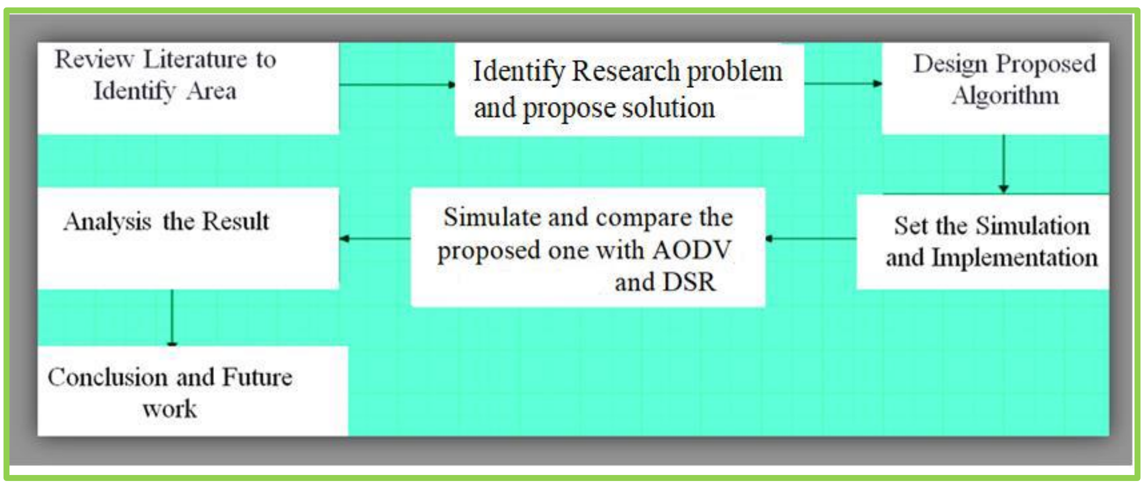



Figure: 3.1 Overall Procedure of the Study

## 3.1. Literature Review

An extensive literature review was conducted to understand the current state of research in mobile ad hoc networks and routing protocols to maintain the limitations of existing routing protocols and suggest how they can be improved. The strategy included looking up journals, articles, and research and conference papers on QoS of DSR.

## 3.2. Protocol Design

Based on the observation from the literature review, a dynamic source routing protocol that considers queue length awareness was designed.

There are many methods to build protocols for different types of research projects. These methods include employing simulation, emulation, real test beds, analytical models, and emulation to measure the behavior and performance of protocols in wireless networks generally. Building actual test beds for any predetermined scenario is typically costly, even impossible when considering variables such as testing space, mobility, density, and traffic patterns. Furthermore, most of measurements are labor-intensive and non-repeatable. Consequently, simulations are required to avoid these issues. So, we prefer simulation easily to test the existing and our designed QLA-DSR protocol.

Here are some common types of datasets used in MANET research:

- Real-World Traces: Researchers use traces collected from real-world MANET deployments that can provide insights into the dynamic behavior of nodes, mobility, and communication patterns. However, obtaining real-world traces for MANETs might be challenging, and privacy concerns limit the availability of such datasets.
- Synthetic Datasets: Synthetic datasets are generated using simulation tools to model different network scenarios. Researchers can control various parameters, such as node mobility, network density, and traffic patterns.
- Community Datasets: Some research communities or projects may provide datasets for MANET simulations. These datasets can vary widely in content and application, they typically serve the public good by providing access to valuable data for various users, including researchers, developers, policymakers, and community members.
- Modified Existing Datasets: Researchers take existing datasets and incorporate QoS-related parameters, for example, enhancing existing datasets with information about delay, throughput, or packet loss to assess QoS performance.

We use synthetic datasets for our simulations; typically referring to datasets that are generated through simulation models or algorithms rather than being collected from real-world scenarios, they allow us to control various parameters and conditions to study the behavior of protocols under specific scenarios and include information about node mobility, network topology, traffic patterns, and other relevant factors. Synthetic datasets are generated by using the simulation tool NS3, detailed simulation parameters are listed in Table 3.1, and performance matrices (end-to end-delay, and packet delivery ratio) were selected and after running simulation data and results are analyzed to evaluate network performance.

## 3.3. Simulation Tools

Network simulators for MANETs come in different forms in the communication industry, such as NS-2, NS-3, OMNET++, SWAN, OPNET, QUALNET, J-SIM, and GLOMOSIM. However, because there are so many differences in operating systems, hardware specs, programming software requirements, and scalability, choosing the right simulator for a task can be difficult. Different aspects are considered while modeling a MANET environment with one of these network simulators. Therefore, it is essential to choose a suitable network simulator and determine which one will offer the best performance and applicability of the network simulator for carrying out and assessing the proposed task. Here are a few simulation tools that researchers commonly use.

- NS-3 (Network Simulator 3): It is open-source, extensible, supports various networking protocol models, has a large active community, and supports C++ and Python.
- OMNeT++ (Objective Modular Network Testbed in C++): It is modular and extensible, supports discrete event simulation but a steeper learning curve for some users, and may require additional expertise for complex simulations.
- QualNet: It is a commercial simulator with extensive features and a graphical user interface (GUI) for easier simulation setup Proprietary software (license fees) may not be as extensible or customizable as open-source alternatives.

- OPNET (now part of Riverbed): It is a commercial simulation tool and GUI-based interface for simulation design, but Proprietary software (license fees) and Limited extensibility compared to open-source tools.
- J-Sim (Java-based Simulation Environment): It is Java-based, platform-independent, and supports discrete event simulation but is less popular than some other tools and may have fewer built-in models compared to larger simulators.

Table: 3.1 Simulator Comparisons

| **Simulator** | **Qualnet** | **OMNET++** | **NS3** | **Opnet** | **J-sim** |
|---|---|---|---|---|---|
| Language supported/interface | Parse C++ | C++ | C++/python | C++/Java | Java |
| Licensed | commercial | Open source | Open source | commercial | Open source |
| GUI support | Excellent | Good | Poor | Excellent | Good |
| popularity | Less popular | More popular | Most popular | popular | Not popular |
| Granularity | Finer | Medium | finest | fine | fine |
| Platform | Linux | Linux, Mac-OS, Unix | Unix, Mac-OS Window | Window, Linux, and Solaris | Math lab |

We have chosen NS3 for implementation and evaluation based on several factors, including its compatibility with various networks, the variety of modules it offers, its support for debugging and tracing, its active development community, its popularity, its adaptability, and its ability to create dynamic topologies. Moreover, NS3 is open-source software, freely available under the GNU GPLv2 for research, development, and use, and it supports different ad hoc network protocols. It offers realistic and detailed simulation environments, is easy to use and debug, and fulfills all the requirements of the simulation workflow, from simulation configuration to trace collection and analysis and comprehensive documentation. Finally, it is just an event-driven simulation tool that has shown to be useful in researching the dynamic nature of communication networks.

Generally, major NS-3 components are shown below in Figure: 3.2

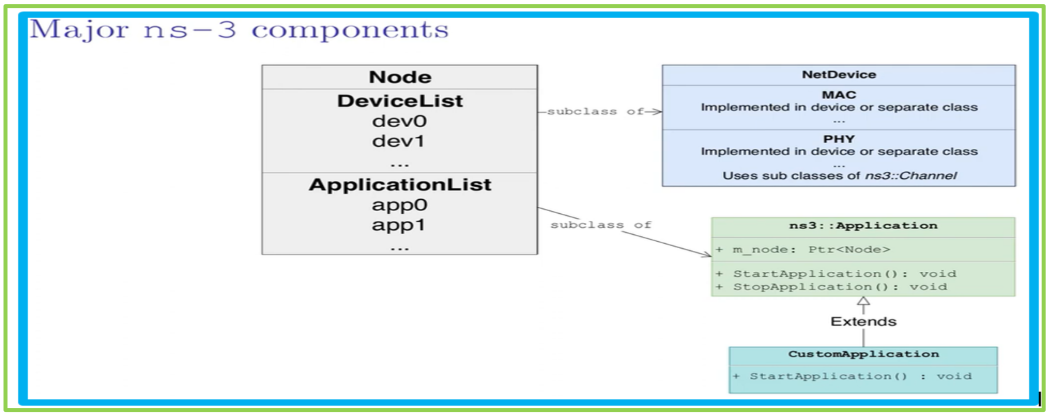



Figure: 3.2 Major NS-3 Components

The simulation parameters and value used for comparing the performance matrices using NS3 are listed in Table: 3.2 below.

Table: 3.2 Simulation Parameters

| Parameter | Value |
|---|---|
| Oprating system and simulation | Ubuntu,Ns3(v3.28.1) |
| Simulation Area | 1000mx1000m |
| Radio Propagation | TwoRay Ground |
| Channel Type | Wireless Channel |
| Traffic Model | CBR |
| Routing Protocol | AODV,DSR,QLA-DSR |
| Mobility Model | Random Waypoint |
| Number of Nodes | 20,30,50 |
| Simulation Time | 30sec |
| Mac Type | IEEE 802.11b |
| Packet Size | 512 bytes |
| Antenna | Omnidirectional |
| Interface Queue | 50 |

## 3.4. Comparison with Existing Protocols

The performance of the QLA-DSR is compared with that of existing protocols such as AODV and DSR. The evaluation process includes selecting appropriate performance metrics, conducting simulations, and comparing the results with baseline protocols like Dynamic Source Routing (DSR) and Ad-hoc On-Demand Distance Vector (AODV). A set of performance metrics will be defined to evaluate the proposed protocol. The metrics include End-to-End Delay, and Packet Delivery Ratio.

- Average End-to-End Delay (AEED): The duration of a message's journey from the source node to the destination node, including any time spent in queues, is known as the End-to-End Delay. It covers all potential delays brought on by propagation and transfer times, interface queuing, transmission, and retransmission delays at the MAC, buffering during route discovery latency, and more. The better an application performs, the smaller its end-to-end delays.

$$\text{Average End} - \text{to} - \text{End Delay} = \frac{\sum_{i=1}^{n}(\text{Time Packet i Received} - \text{Time Packet i Sent})}{\text{Total number of packet recived}} \qquad \text{Eq. (3.1)}$$

**Where:**

***Time Packet i received***: The time at which packet *i* is received at the destination.

***Time Packet i sent***: The time at which packet *i* is sent from the source.

***n***: Total number of packets received by the destination.

***Total Number of Packets received***: The total count of packets received by the destination.

- Packet Delivery Ratio (PDR): The percentage of delivered packets to all packets sent. It is stated as a percentage. As a result, a routing strategy with a higher packet delivery ratio also has a lower packet loss rate. PDR is a crucial indicator since it assesses a network's dependability, and a high PDR is desired for real-time applications.

$$\textbf{PDR} = \frac{\text{Total Number of Data Packets Received by Destination} \times 100}{\text{Total Number of Data Packt Sent by Source}} \qquad \text{Eq. (3.2)}$$

**Where:**

***Total Number of Data Packets Received by Destination***: This is the count of packets that successfully arrive at their intended destination.

***Total Number of Data Packets Sent by Source***: The count of packets originally transmitted from the source node.

## 3.5. Validation and Analysis

The Performance metrics results obtained from the simulations are analyzed and validated to determine the effectiveness of the QLA-DSR. Based on the analysis, suggestions for further improvements to be made.

## 3.6. Conclusion and Future Works

The project concludes with a summary of the results and a discussion of the contributions that the suggested protocol makes to the field of mobile ad hoc networks. The simulation results are illustrated with a graph (a Gnu plot is used).

# CHAPTER FOUR

# 4. Design Proposed QLA-DSR

As mentioned in the previous sections the existing MANETs routing algorithms are not efficient enough for QoS demanding applications (Balhara, Prakash, & Tripathi, 2013). Hence, a mechanism for improving the services of MANET applications is important. In this study, we suggested a DSR-based method for improving quality by taking node queue length into account when building routes to lower latency and increase quality for applications that need QoS. As a result, it provides a detailed workflow explanation, flowcharts, pseudo code, and architecture for the suggested QLA-DSR routing technique.

## 4.1. QLA-DSR Architecture

Here we describe and present the general architecture of our proposed QLA-DSR algorithm in this part to demonstrate the high-level functions of the proposed technique. When many packets are processed and queued in the buffer, a packet is received by an intermediate mobile node in a MANET, stored in its buffers, and then forwarded, if available, to an output link. To retrieve each node's buffer information from the link layer and utilize it for the network layer's route discovery and selection process there must be cross-layer communication between the network layer and data link layer. The modified part of the architecture is indicated to the right side with a white color in Figure 4.1 below.

The modified proposed QLA-DSR protocol, focusing on queue length-based path selection to enhance routing efficiency, has different component parts.

The routing agent is the core component that listens to neighbor’s information and maintains a route cache with a port classifier responsible for categorizing packets based on source and destination ports. The protocol evaluates the intermediate nodes queue size and selects a path that is least congested using queue Size-Based Path Selection, after the optimal path is determined; the packets are forwarded along that route, ensuring efficient and timely delivery by forwarding packets.

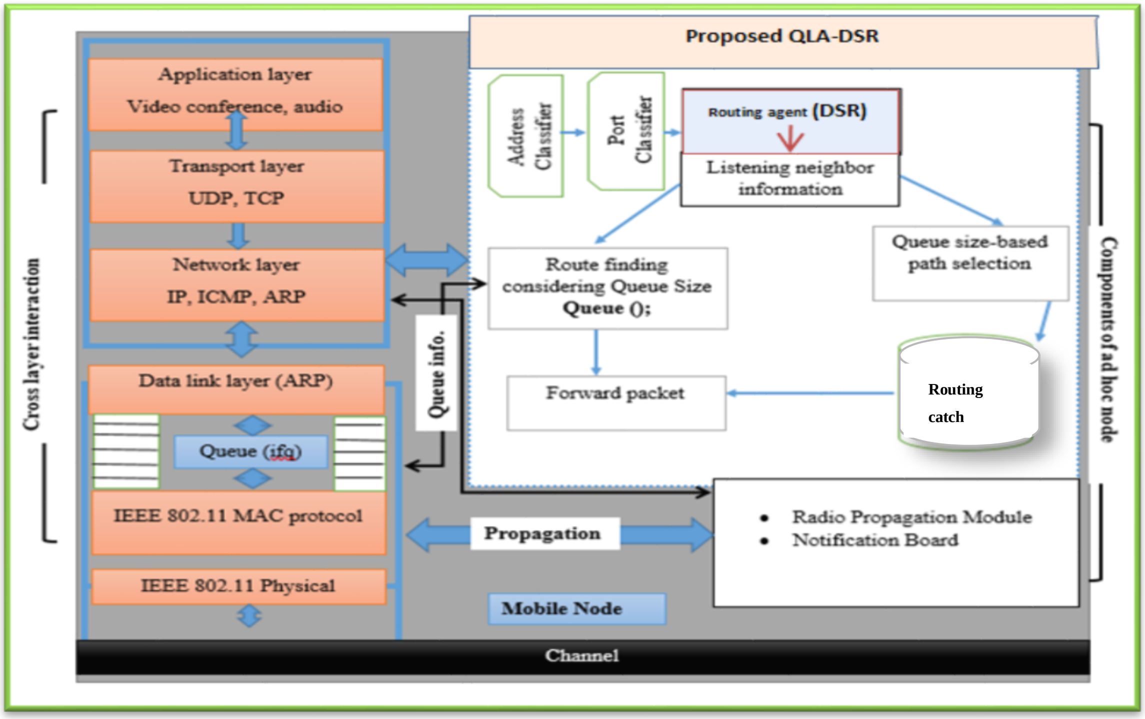



Figure: 4.1 QLA-DSR Architecture

## 4.2. Formula for Queue Length in QLA-DSR

The queue length is typically calculated as the number of packets (or data units) waiting in the buffer at any given time. In QLA-DSR, the queue length is evaluated dynamically for each node as part of the route discovery process.

Let's break down the formula and the key variables involved:

❖ Queue Length at Node (Qn)

The queue length at a node n is defined as:

$$Qn = \sum_{i=0}^{N} Pi \qquad \text{Eq. (4.1)}$$

Where: Qn is the total queue length at node n (in terms of the number of packets or bytes).

Pi is the size of the i-th packet in the queue (usually in bytes or bits).

N is the total number of packets in the queue at node n.

- Total Queue Length for a Path (Qpath)

In QLA-DSR, the total queue length for a given path is the sum of the queue lengths of all intermediate nodes along the route:

$$\text{Qpath} = \sum_{n=1}^{M} Qn \qquad \text{Eq. (4.2)}$$

Where:

QPath is the total queue length along the route (path).

Qn is the queue length at the n-th intermediate node.

M is the total number of nodes along the path (excluding the source and destination).

- Average Queue Length for Path (Qavg**)**

To compare different routes, QLA-DSR might calculate the **average queue length** along a path, which is given by:

$$Q\text{avg} = \frac{\text{Qpath}}{\text{M}} \qquad \text{Eq. (4:3)}$$

Where:

Qavg is the average queue length for the path.

Qpath is the total queue length for the path.

M is the number of intermediate nodes.

- Threshold Queue Length

Some QLA-DSR implementations introduce a threshold queue length Qth, which represents the maximum allowable queue length at any node before it is considered congested. If the queue length of a node exceeds this threshold, the route may be discarded or avoided:

$Qn > Qth \Longrightarrow$ Avoid node in path selection

Where Qth is a predefined threshold based on network conditions or protocol settings.

- Queue Length Based Routing Metric

To make routing decisions, a routing metric based on queue length can be defined. The goal is to minimize the total queue length of the selected route. The routing metric could be expressed as:

$$MR=\alpha\cdot QR+\beta\cdot \text{Other Metrics} \qquad \text{Eq. (4.4)}$$

Where:

MR is the routing metric for route R

$\alpha$ is a weight assigned to the queue length factor,

QR is the total queue length of the route,

$\beta$ is the weight for other metrics (such as hop count, link stability, etc.).

By optimizing MR, QLA-DSR selects routes that not only have fewer hops but also lower queue lengths, thereby reducing congestion and queuing delays.

Nodes may implement dynamic queue management to adjust their queue sizes based on traffic conditions. For example, during high congestion periods, the node might reduce its queue size to prioritize delay-sensitive traffic.

When multiple routes are available from source to destination, the QLA-DSR protocol chooses the route with the minimum total queue length:

Ropt = min (QR)

Ropt is the route with the optimal queue length among all discovered routes.

By minimizing QR, QLA-DSR ensures that packets are routed through less congested nodes, reducing the likelihood of packet drops, queuing delays, and overall congestion.

In QLA-DSR, queue length plays a critical role during route discovery and route selection. By incorporating queue length into the routing decision process, QLA-DSR prioritizes routes with less congestion, resulting in lower delays, higher packet delivery ratios, and more efficient use of the network. The key steps include: Monitoring the queue length at each node, Incorporating queue length into the route discovery process and selecting routes based on the total queue length along the path.

This theoretical framework ensures that the network dynamically adapts to traffic conditions and avoids congested routes, leading to better overall performance in mobile ad hoc networks.

The formula MR=α·QR+β·Other Metrics combines queue length and other routing metrics to select the optimal route in the Queue Length Aware Dynamic Source Routing (QLA-DSR) protocol. This example will demonstrate how this metric might work in a practical scenario, where RMR (the total routing metric) is a weighted sum of the queue length and another routing metric, such as hop count.

The parameters α and β in the formula MR=α·QR+β·Other Metrics are not fixed. Instead, they can be adjusted based on the specific goals and performance requirements of the network. Here's how they can be assigned and adjusted: By adjusting the weights α and β, the protocol can prioritize either queue length or other routing factors (such as hop count) depending on the network requirements.

They can be assigned and adjusted: Based on Network Requirements (Delay-sensitive networks or Energy-efficient networks), Tuning via Simulation (Simulation tools (e.g., NS-3) can be used to experiment with different values of $\alpha$ and $\beta$. We can run simulations using different values and analyze performance metrics such as end to end delay, and packet delivery ratio (PDR) to identify the optimal balance for our specific network scenario.

Recommended Values for Delay-Sensitive Applications: $\alpha$: 0.7 to 0.9 this emphasizes the importance of queue length in selecting a path with lower latency and $\beta$: 0.1 to 0.3 this allows some consideration of other factors like hop count, but the main priority remains minimizing delay.

- ❖ Integration with Route Discovery

During route discovery in QLA-DSR, each node appends its current queue length to the Route Request (RREQ) packet. The source node evaluates the queue lengths from the nodes that have responded with Route Reply (RREP) messages and selects the route that minimizes queue length, among other metrics such as hop count. The main idea is to avoid congested nodes by using queue length as a factor in routing decisions, ensuring that paths with shorter queues (and thus less congestion and delay) are favored for transmitting packets. In Queue Length Aware Dynamic Source Routing (QLA-DSR), the queue length is primarily used during the route discovery process, not during packet transmission. The goal of incorporating queue length in route discovery is to ensure that the selected route has minimal congestion, thereby reducing potential delays and improving overall network performance.

## 4.3. QLA-DSR Algorithm

As mentioned in the related work section, we have seen that the existing DSR routing algorithm is not very successful in considering node queue size during the route construction process as route selection cost metrics. The proposed algorithm of this study on the selected DSR MANETs routing protocol is a queue length-aware improved routing algorithm for both route discovery and route selection mechanisms. Routes between sources and destinations are established using the minimal hop count criterion by the basic Existing DSR routing algorithm. To meet QoS criteria, choosing the route with the fewest hops is typically not the best option. In this study, to reduce latency for delay-sensitive applications, we examine a node queue length-based route selection method rather than using an established DSR routing algorithm, additionally, we use a Max-Min-Path based queue length maintenance through RREQ packet; this guarantees a packet won't be lost due to unexpected buffer overflow at the intermediate node during packet transmission. We have added a field to the routing table header fields to keep the node queue length at a specific time; the suggested approach will first get the queue length at each node and the queue length field of the route request (RREQ) packet updated with the minimum of these two values. If the node is not the destination or does not have a route to it, then it rebroadcasts the RREQ packet with the modified queue length information.

Every intermediate node will transmit RREP to the source after receiving an RREQ packet, if it has a route to a destination, send it. If not, the intermediate node updates the RREQ Queue length field with the current available queue length that it has calculated. Until the request reaches the destination node, this rebroadcast process will be repeated. After deciding on the optimal course of action, the destination node will send a route replay packet (RREP) to the source node, which transfers the data to the destination node via the choice path. After receiving the primary RREQ, the destination node could need some time to create a lot of routes. After selecting the route with the shortest queue length value, it sends back an RREP packet. As soon as the source node receives the RREP packet, then it will start sending information packets over this route to the destination node.

## 4.4. High-level Overview of the Algorithm's Design:

- Node Initialization: Every node in the MANET initializes its routing tables and queue length which contains information about its neighbors and their corresponding queue lengths. Initial observations are used to update the queue length information, or it will be set to a default value.
- Queue Length Exchange: Nodes periodically share queue-length information with their nearby nodes to maintain an up-to-date view of the queue lengths. The frequency of queue length exchange is based on network dynamics and requirements.
- Route Discovery and Route Reply: The Route discovery process broadcasts a route request (RREQ) packet containing the destination address and its queue length information that can be initiated when the source node needs to transmit a packet. Then neighboring nodes update their routing tables and compare their

queue lengths with the incoming RREQ's queue length after receiving the RREQ packet. If a node's queue length is smaller than the RREQ packet, then it becomes a candidate for the next hop in the route. The Node with the smallest queue length responds to a route reply (RREP) packet that contains its queue length information.

- Source Route Maintenance: The source node caches the discovered route and uses it for packet transmissions to the destination. If a node in the cache has high queue length, then the source node reinitiates the route discovery process to find another route. The frequency of route maintenance can be adjusted to balance network overhead and responsiveness to changing situations.
- Data Packet Forwarding: The source node uses source routing information to wrap data packets once it has a valid path to the destination. After receiving the packet, each intermediate node uses the source-routing instructions to choose which hop to send it to next. To prevent congestion, the node's queue management policies, and queue length also play a role in the decision to transmit a packet.
- Queue Length Updates: Intermediate nodes update their queue lengths on the routing table and queue length exchange the process based on packet transmission. This allows for dynamic adjustment of routing decisions based on real-time queue length measurements.

## 4.5. Flow Chart of QLA-DSR Algorithm

The process of creating a route request (RREQ) packet and processing it at intermediate nodes while taking node queue length into account during route establishment is shown in the flowchart in Figure 4.2. To ensure the RREQ maintains the queue length for each node, the source node initiates the route discovery whenever it has packet to transmit to the destination. This means the queue length value in RREQ and the queue length value in intermediate nodes are considered, and the RREQ packet replaces its current queue length value by the minimum of these two values.

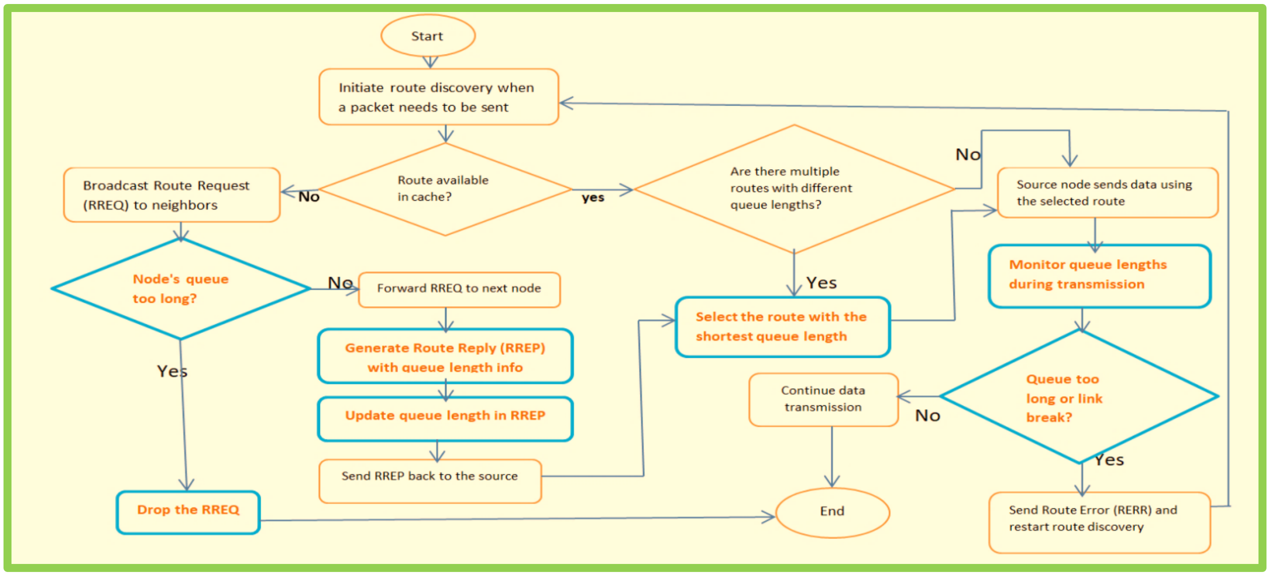



Figure: 4.2 Flow Chart of QLA-DSR Algorithm

## 4.6. Implementation of QLA-DSR

### 4.6.1. Algorithm Modification

Classes and methods of the existing DSR protocol algorithm in NS3 have been modified. In NS-3, dsr has dsr.cc and dsr.h files that implement the DSR routing protocol. Dsr.h has two helpers (dsr-helper.h and dsr-main-helper.h). In the same manner dsr.cc has two. cc (dsr-helper.cc and dsr-main-helper.cc). Functions are arranged according to their assigned tasks in the routing activity. To implement the suggested QLA-DSR algorithm, some of the existing DSR C++ modules and files are updated in this work. First, a new folder is created with the name qla having all files of dsr. Then we modified a helper class under dsr.cc/.h, which existed under, ns-all-in-one- 3.28.1/ns-3.28.1/src/dsr/helper folder and stored the route ns-all-in-one- 3.28.1/ns-3.28.1/src/qla/helper folder to get the queue length of each node at the link layer. Here are the modified helper classes' file locations and sample source codes. The File location of the modified helper is at Home/ns-all-in-one- 3.28.1/ns-3.28.1/src/qla/helper.


Figure: 4.3 Modified helper Locations

Then, dsr-fs-header.h/.cc, dsr-routing.h/.cc, dsr-rreq-table.h/.cc and others located under ns-all-in-one-3.28.1/ns-3.28.1/src/dsr/model folders are modified to work with the additional header fields at the route entry header format and a new code qla-routing-protcol.h/cc is created under ns-all-in-one-3.28.1/ns-3.28.1/src/qla/model folders.

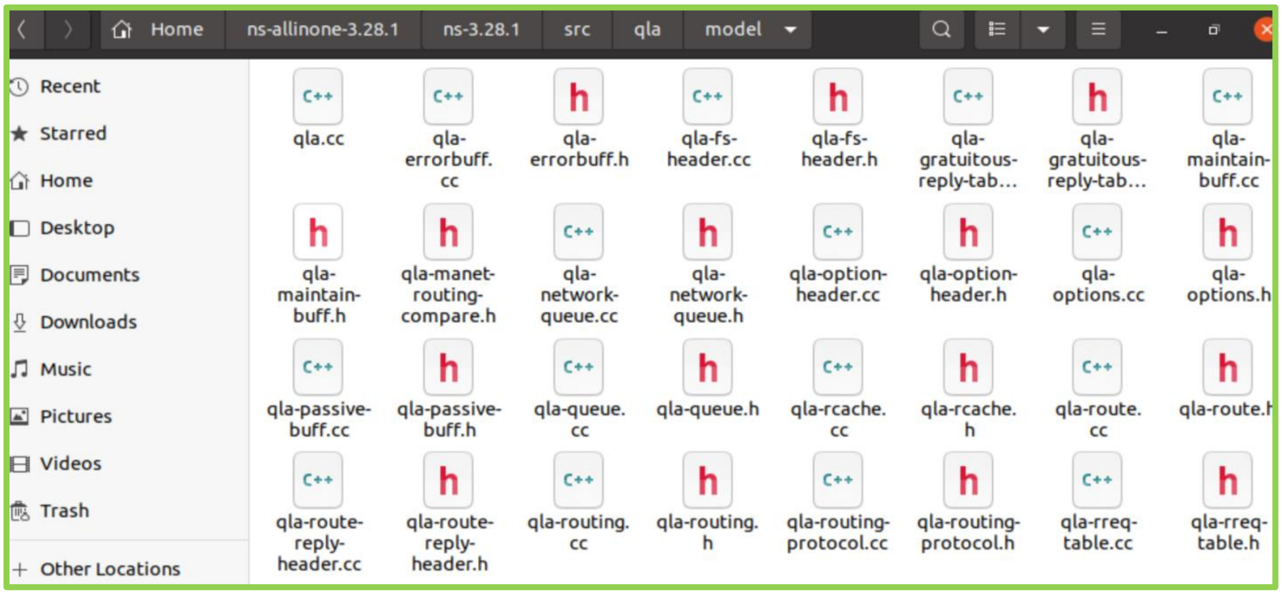



Figure: 4.4 Modified qla Modules Location

We have changed a few functions in the dsr-routing.cc/.h file, including queue length, receive request, send request, receive replay, and send reply, to take into account node buffer information when doing route discovery and selection processes in addition to the algorithm's previous header files.

Lastly, the dsr.cc is changed to qla.cc to access the queue length of each node by calling the function that holds the queue length of each node from the link layer and takes node buffer information into account during the route discovery and selection process.


Figure: 4.5 Modified qla.cc Location

### 4.6.2. Modification of RREQ at Intermediate Nodes

The RREQ packet is modified at each intermediate node based on the queue length, this information is used by the destination or intermediate nodes with valid routes to decide on the best path based on queue lengths and update the queue length value, helping to avoid congested route and improving overall network performance for delay-sensitive applications.

```
Modification of RREQ at Intermediate Nodes

Notations:
 S: Source Node
 D: Destination Node
 I: Intermediate Node
 QLEN: Queue Length at node I
 RREQ: Route Request
 RREP: Route Reply
 RREQ_QLEN: Queue Length field in the RREQ packet

Algorithm:
FOR (Receive_RREQ(I, RREQ))
   IF (I has a valid route to D) THEN
        Generate_RREP(I, S);
        RETURN;
       ELSE
            IF (QLEN(I) > RREQ_QLEN) THEN
             RREQ_QLEN = QLEN(I);
         ENDIF
        IF (RREQ is not duplicated) THEN
             Rebroadcast_RREQ(I, RREQ);
          ELSE
             Discard_RREQ(I, RREQ);
         ENDIF
     ENDIF
END FOR
```

Figure: 4.6 Modification of RREQ at Intermediate Node

### 4.6.3. Modification of RREP at Destination Node

This algorithm proves that the RREP packet carries the maximum queue length encountered along the path which allows the source node to make decisions on the optimal path for transmitting data, minimizing delay in the network.

**Modification of RREP at destination Node**

***Notation:***

***S***: *Source Node*

***D***: *Destination Node*

***I***: *Intermediate Node*

***QLEN***: *Queue Length*

***RREQ***: *Route Request*

***RREP***: *Route Reply*

***SN***: *Sequence Number*

***Input:*** ***N***: *Number of available paths*

***Process:***

***1. FOR*** *(every received RREQ at the destination node* ***D****)*

***2. IF*** *(received sequence number > stored sequence number)*

***3. THEN*** *Select new route*

***4. ELSE IF*** *(received sequence number == stored sequence number)* ***THEN***

*5. Compute cost value based on the queue length for each received RREQ*

***6.*** ***Pn*** *= Min (Max (QLEN (I))) for each path*

*7. Create reverse path (Prepare RREP)*

*8. Send RREP with the minimum value of* ***Pn*** *to the source node* ***S***

***9. ELSE***

*10. Discard the RREQ ()*

***11. END IF***

***12. END FOR***

Figure: 4.7 Modification of RREP at Destination Node

CHAPTER FIVE

# 5. Evaluation and Result Discussion

In this chapter, the performance evaluation scenarios for the suggested QLA-DSR routing protocol in comparison to AODV and DSR are discussed in detail. By altering the manner of route that can be built in the DSR routing algorithm, this study offers a QoS enhancement strategy over the current DSR routing protocol.

## 5.1. Simulation Scenario and Model

The environment consists of a 1000 m x 1000 m simulated area with a maximum of 50 nodes to evaluate and contrast the proposed QLA-DSR, AODV, and DSR routing protocols under various mobility speed situations. The characteristics that need to be assessed are average end-to-end delay, and packet delivery ratio. There are three different node combinations, 20, 30, and 50, for sparse, medium, and dense networks, respectively. A drop tail priority queue is used with a queue size of 50 packets for each node. Omnidirectional antenna is used for the noses to listen signals in all directions. The two-ray ground propagation model is also used as a signal propagation model as it uses both a direct path and a ground reflection path.

Three settings are made up of simulation, each of them used to evaluate how a unique network operating situation affects the QLA-DSR routing protocol's performance. First, the effect of network density is investigated by changing the number of mobile nodes. The second simulation scenario, varying the number of source-destination pairings for a fixed number of nodes positioned on the region, and the effect of load density on the performance of the proposed QLA-DSR routing protocol is examined. In the third simulation scenario, the speed of a fixed number of mobile nodes is varied to determine the impact of node mobility on the performance of the routing protocol.

## 5.2. The Mobility Model

Each simulation's movement scenario files are identified by speed and pause duration. The movement patterns developed for varying speeds of 2 m/s, 5 m/s, and 10 m/s are used. Mobile nodes in realistic scenarios move and create traffic randomly, making up MANETs. We use the random waypoint mobility model to construct the mobility model, since according to many research studies; random waypoint is the most popular mobility model in MANETs (Balhara et al., 2013; Pramanik, Choudhury, Choudhury, Arif, & Mehedi, 2015). In Figure 5.1 below, a mobile node begins at its current location in the simulation region and travels uniformly between the minimum and maximum speeds to a randomly chosen destination. The movable node moves in a freshly chosen direction and speed after pausing briefly upon arriving at its destination. Various mobility speed variations have also been examined using the analogies of a man walking, a bicycle, and a car, respectively, at 2 m/s, 5 m/s, and 10 m/s.

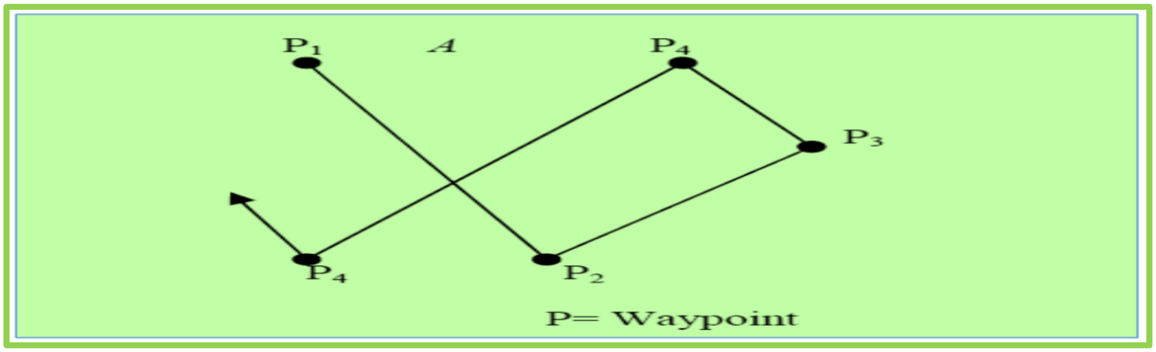



Figure: 5.1 Mobility Model of Random Waypoint

## 5.3. The Traffic Models

To simulate traffic, we use the ns-3 network simulator and the Net Anim network animator. The network topology, packet flow, and queue lengths at each node all are displayed in the visualization. With a traffic rate of four packets per second and a packet size of 512 bytes, various size connections were built between the nodes in the network during the simulation, rising in line with the number of nodes taken into consideration. The nodes have the On-Off traffic model, and the simulation time is thirty seconds. Net Anim is used to visualize the network traffic shown in Figure: 5.2 below.

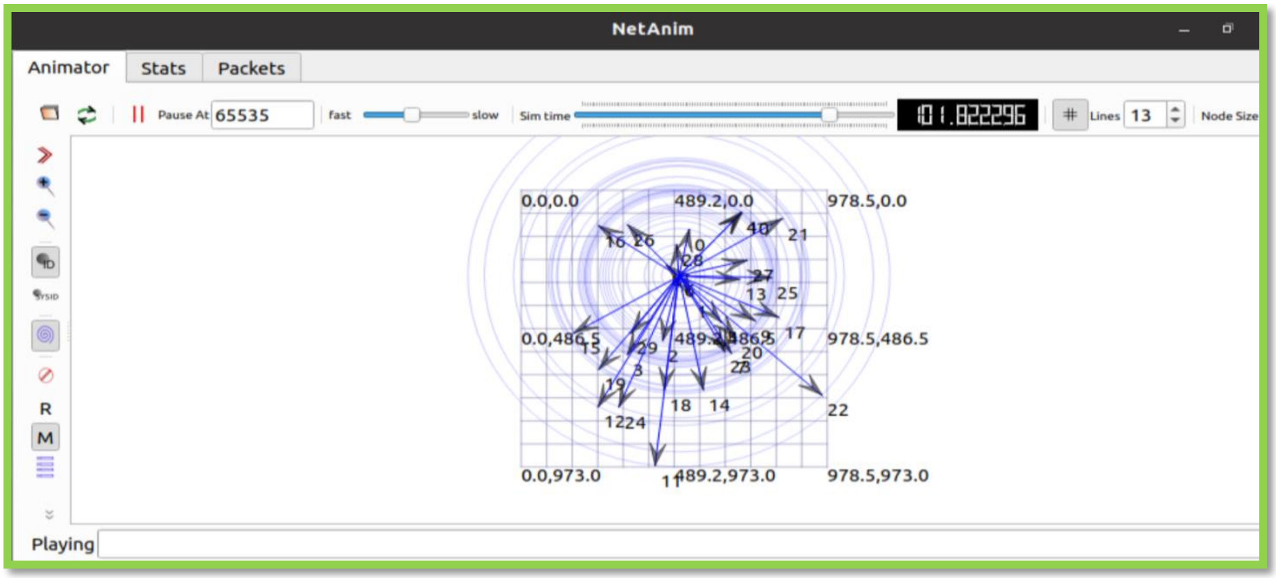
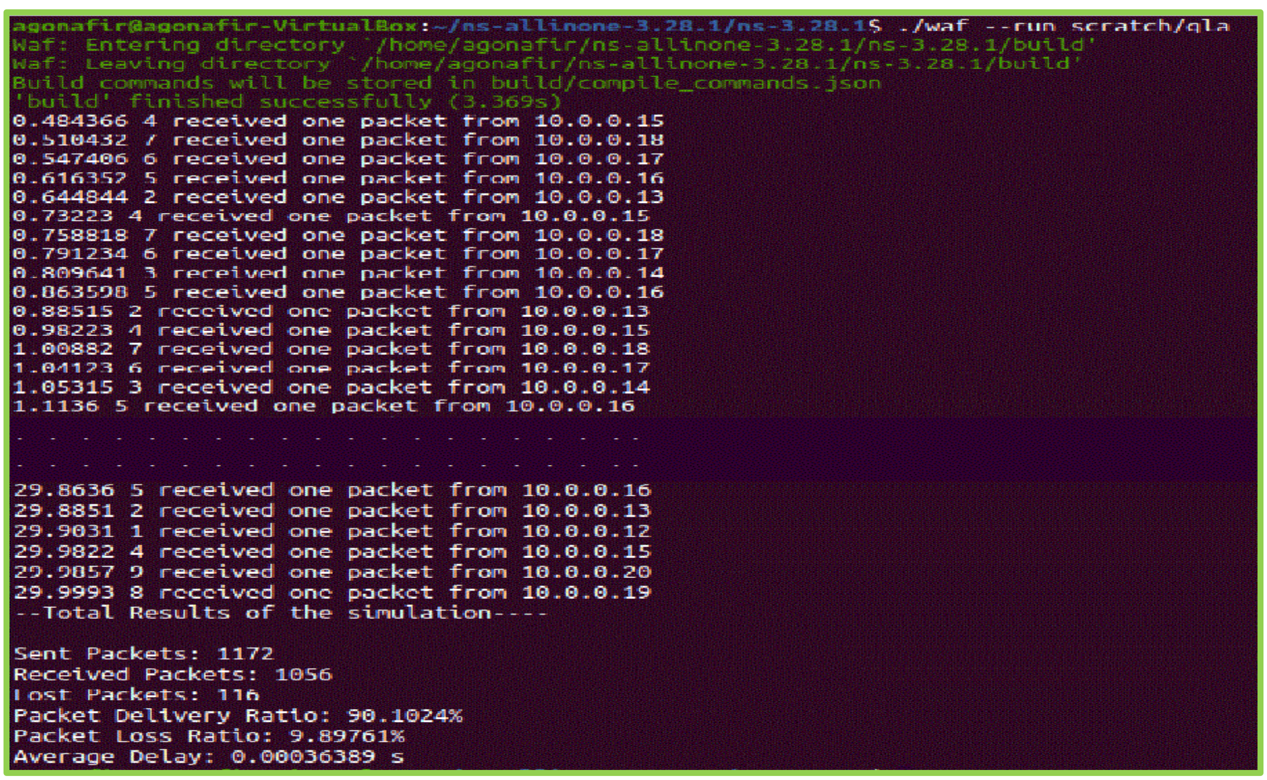



Figure: 5.2 QLA-DSR Simulated Networks a 30 Node Topology Using Net Anim

For example the sample of qla.cc file for QLA-DSR protocol run and the simulation result displayed in Figure: 5.3 shown below.


```
agonafir@agonafir-VirtualBox:~/ns-allinone-3.28.1/ns-3.28.1$ ./waf --run scratch/qla
Waf: Entering directory `/home/agonafir/ns-allinone-3.28.1/ns-3.28.1/build'
Waf: Leaving directory `/home/agonafir/ns-allinone-3.28.1/ns-3.28.1/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (3.369s)
0.484366 4 received one packet from 10.0.0.15
0.510432 7 received one packet from 10.0.0.18
0.547406 6 received one packet from 10.0.0.17
0.616352 5 received one packet from 10.0.0.16
0.644844 2 received one packet from 10.0.0.13
0.73223 4 received one packet from 10.0.0.15
0.758818 7 received one packet from 10.0.0.18
0.791234 6 received one packet from 10.0.0.17
0.809641 3 received one packet from 10.0.0.14
0.863598 5 received one packet from 10.0.0.16
0.88515 2 received one packet from 10.0.0.13
0.98223 4 received one packet from 10.0.0.15
1.00882 7 received one packet from 10.0.0.18
1.04123 6 received one packet from 10.0.0.17
1.05315 3 received one packet from 10.0.0.14
1.1136 5 received one packet from 10.0.0.16
 . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . . . . .
29.8636 5 received one packet from 10.0.0.16
29.8851 2 received one packet from 10.0.0.13
29.9031 1 received one packet from 10.0.0.12
29.9822 4 received one packet from 10.0.0.15
29.9857 9 received one packet from 10.0.0.20
29.9993 8 received one packet from 10.0.0.19
--Total Results of the simulation----

Sent Packets: 1172
Received Packets: 1056
Lost Packets: 116
Packet Delivery Ratio: 90.1024%
Packet Loss Ratio: 9.89761%
Average Delay: 0.00036389 s
```

Figure: 5.3 Sample Simulation Output

## 5.4. Performance Metrics

End-to-End Delay and Packet delivery ratio are used to evaluate the Performance of QLA-DSR routing protocol against the existing DSR and AODV.

## 5.5. Result Analysis and Discussions

To evaluate the performance, the simulation was done six times iteratively for three scenarios i.e. for 20, 30, and 50 nodes two times (Impact of Network Density), two times (for the number of source-destination pairings) for a fixed number of nodes by varying no of sink 2, 5 and 10 and finally two times (by varying nodes speed) for a fixed number of mobile nodes and speed of 2m/s, 5m/s, and 10m/s. Nine performance results were achieved and analysis and evaluation were discussed.

Table: 5.1. Parameter used and value range for QLA-DSR

| Parameter | Description | Value Range | source |
|---|---|---|---|
| No of node | Total number of nodes in the mobile ad hoc network (MANET) | 20, 30, 50 | (Temesgen et al., 2019) and configured in NS-3 simulation |
| Packet size | The total amount of data (in bytes or bits) contained in a packet | 512Bytes | configured in NS-3 simulation and (Liang Qing AND Lin, 2021) |
| Total simulation time | Total duration of the simulation | 30s | (Temesgen et al., 2019) |
| Simulation area | The physical or virtual space within which nodes operate and interact during a simulation | 1000mx1000m | (Temesgen et al., 2019) |
| Node speed | Speed at which the mobile nodes are moving in the network | 2 m/s, 5m/s, 10m/s | (Almazok& Bilgehan, 2020) and (Temesgen et al., 2019) |
| Queue Length | Length of the packet queue at a node | 50 | (Almazok& Bilgehan, 2020) and (Temesgen et al., 2019) |
| Dependent variables | | | |
| PDR | Ratio of successfully delivered packets to send packets | 80%-100% | (Temesgen et al., 2019) and derived from simulation results |
| End-to-End Delay | Time taken for a packet to travel from source to destination | 0-0.4s | (Temesgen et al., 2019) and derived from simulation results |

## 5.6. Impact of Network Density

Here we describe how the three routing protocols AODV, DSR, and QLA-DSR perform when there is a different network density. The routing protocols in this scenario are simulated by changing the number of nodes. The simulated network has 20, 30, and 50 nodes, which have the same speed (10 m/s) and destination (sink = 10), given sparse, medium, and dense networks, respectively.

- ❖ Packet Delivery Ratio

The packet delivery ratio versus network density is displayed in Figure 5.4 below. For every routing system, the percentage of packets delivered drops when the network density is increased. This is because channel contention and packet collisions in dense networks create excessively redundant retransmissions of control packets (such as RREQ and RREP packets), which lowers the bandwidth used for packet transmission. Figure: 5.4 illustrate that, when compared to DSR, QLA-DSR has the highest packet delivery ratio but less than AODV which has a high packet delivery ratio even as the network size increases. This is especially true at initial and when network density is high, as QLA-DSR chooses a route that is not crowded while taking node queue length into account.

Reducing the chance of link breakdown and the amount of missed data packets due to node buffer overflow at intermediate nodes is necessary to increase the packet delivery ratio.

As seen by the diagram, QLA-DSR outperforms the conventional DSR routing protocol with a small no of nodes, and the number of nodes rises, in the simulation, we attain an average growth of 1.48% packet delivery ratio than DSR and 0.22% than AODV.

Generally, when the number of nodes increases, the packet delivery ratio for DSR, QLA-DSR, and AODV tends to decrease. However, QLA-DSR has a higher PDR than standard DSR and also at some point above performs better than AODV due to its queue length awareness of AODV may retain a relatively higher PDR compared to both QLA-DSR and standard DSR because of its on-demand routing capabilities but after some point due to higher routing overhead and more frequent route discoveries, PDR for AODV decrease than QLA-DSR.

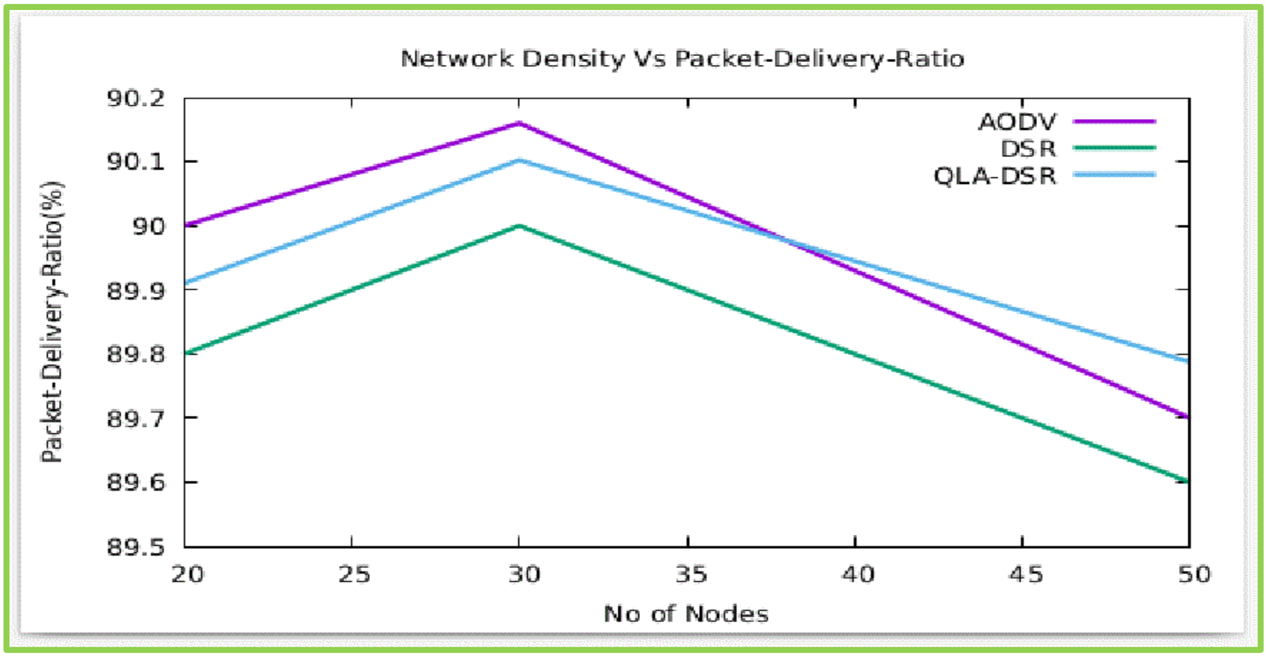



Figure: 5.4 Network Densities vs. Packet Delivery Ratio

❖ Average End-to-End Delay

Figure: 5.7 illustrates the end-to-end delay of routing protocol increases when the network density increases. This is because dense networks generate and send a greater number of routing packets, increasing packet collisions, neighbor node interference, and channel contention that prolong the time it takes to reach the destination.

When the same number of hops is required to reach the destination, AODV and DSR routing protocols take packet transmission time into account. However, they do not consider the queue state of each node.

QLA-DSR outperforms AODV and DSR routing protocols as the number of nodes increases, since, during the route selection process, QLA-DSR considers each intermediate node's buffer status in addition to the minimum hop count. Compared to DSR, AODV experiences more delays when there are fewer nodes. This occurs in AODV's sporadic activity (exchanging HELLO messages) and does not use a cache to store the routes. Generally, it achieves an end-to-end delay improvement of -5.79% on average compared to DSR and -4.4% to AODV.

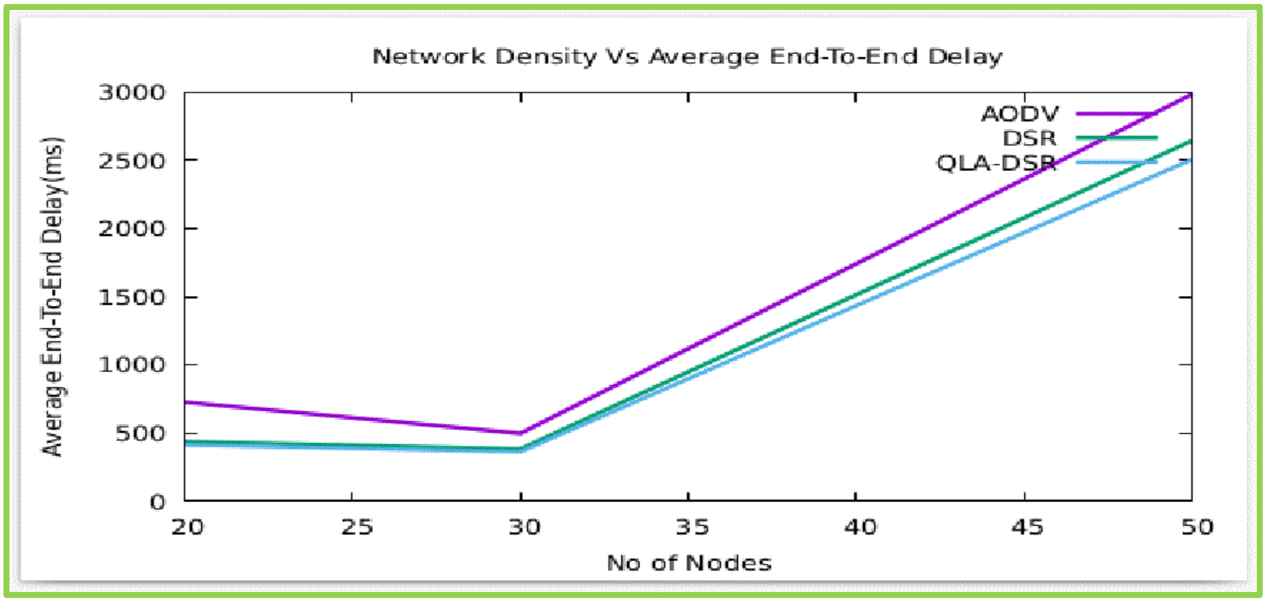



Figure: 5.5 Network Densities vs. Average End-to-End Delay

## 5.7. Impact of Communicating Nodes

To test the performance of the QLA-DSR, DSR, and AODV routing algorithms, another setup that modifies network load in terms of different communication flows (i.e., source-destination pairs) is tested. Several source-destinations have been conducted for a fixed number of nodes, mobility speed, and simulation time.

For every connection, a random source-destination pair is selected. For our study, we take the number of sinks as 2, 5, and 10, and see what happens on the performance metrics if the number of sinks varies.

❖ Average End-to-End Delay

For QLA-DSR, AODV, and conventional DSR routing protocols, end-to-end packet delays increase in conjunction with the number of source-destination pairs. This is because as source-destination pairs grow, so does the number of packets that flow through the network. Due to this, more packets are created and distributed throughout the network, increasing contention, inter-node interference, and packet delays at the queue of intermediate nodes, so, End-to-End Delay increases.

Figure 5.8 illustrates how QLA-DSR performs better than others achieving an average End-to-End Delay. This is because QLA-DSR chooses the best way with the shortest queue length by considering the node queue length during path selection, which extends beyond the hop count. QLA-DSR reduces an average end-to-end delay by -2.1% compared to DSR and -3.78% compared to AODV.

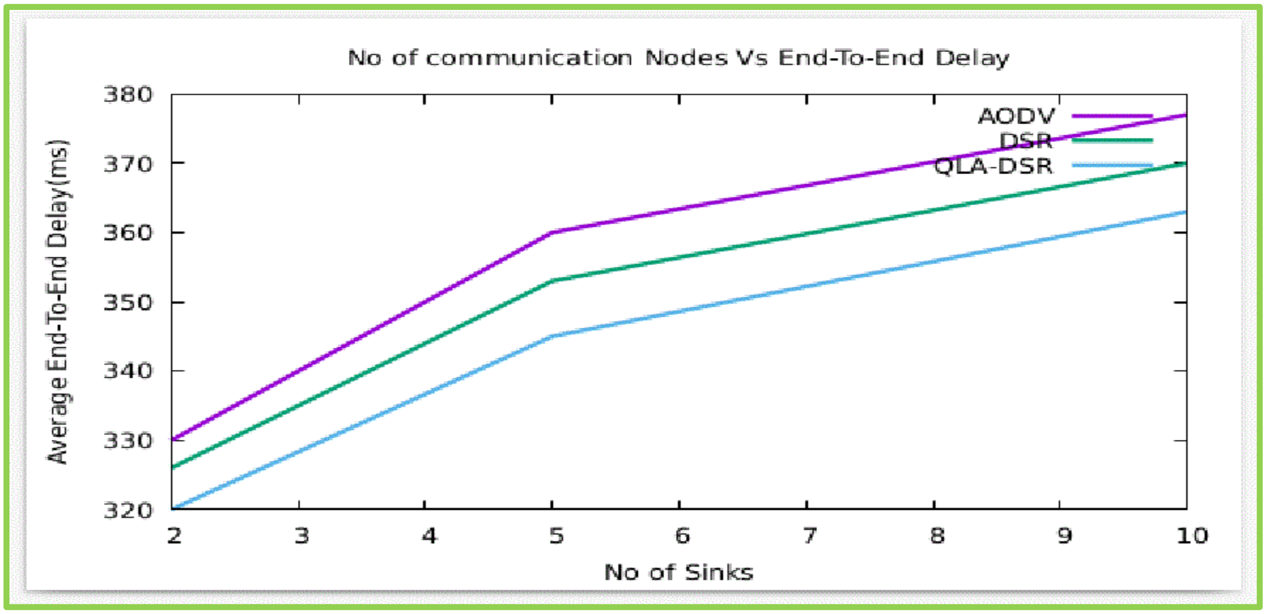



Figure: 5.6 No of Communication Nodes vs. Average End-to-End Delay

❖ Packet Delivery Ratio

As network traffic grows, three routing systems become less effective. This is because additional nodes join networks when source-destination flows rise, increasing network congestion and contention rates. This results in a reduction of the available bandwidth for actual data transmission, leading to a degradation of the delivery ratio. Figure 5.11 illustrates the improvement in the packet delivery ratio of QLA-DSR over the DSR and AODV routing protocol while having different source-to-destination connections. This is due to QLA-DSR choosing a route that considers node queue size and minimum hop count.

Generally, QLA-DSR outperforms the existing DSR and AODV routing protocols by an average of 1.38% and 0.21% respectively.

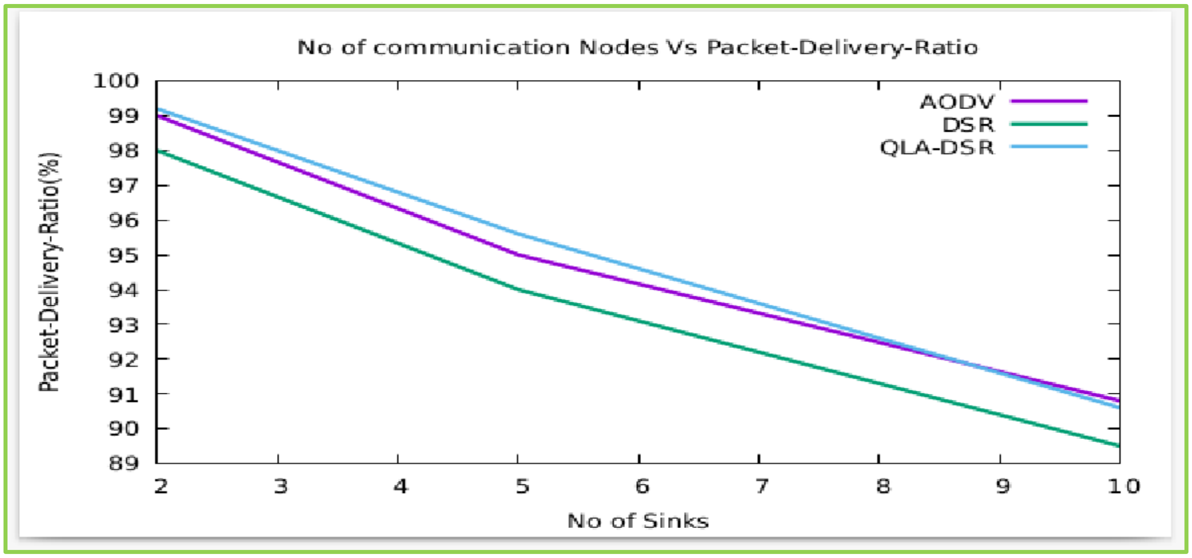



Figure: 5.7 No of Communication Nodes vs. Packet Delivery Ratio

## 5.8. Impact of Mobility Speed

We examine the effects of node mobility on the functionality of three routing protocols, QLA-DSR, AODV, and regular DSR in this simulated environment. With a random waypoint mobility model, the network size in this simulation is 30 nodes and 1000 × 1000 m simulated area. Every movement scenario was examined at three distinct speeds. The network has also been adjusted to operate at speeds of 2 m/sec, 5 m/sec, and 10 m/sec.

❖ Packet Delivery Ratio

When a node's speed rises, the packet delivery ratio decreases because the network becomes more dynamic, and connection failures occur more frequently. High-speed networks have more frequent and unpredictable topology changes, which significantly increase the number of link breakages. Additionally, the broken routes brought on by these frequent topology changes result in more re-requests for route maintenance and discovery, which raises the amount of RREQ packets generated and spread throughout the network. As a result, Figure 5.14 illustrates that there are more packet losses, which lowers the delivery percentage. QLA-DSR outperforms the current AODV and DSR, despite the challenge of efficiently delivering packets in a highly mobile network. Generally, QLA-DSR outperforms the existing DSR by 1.16% in packet delivery ratio and by 1.21% to AODV.

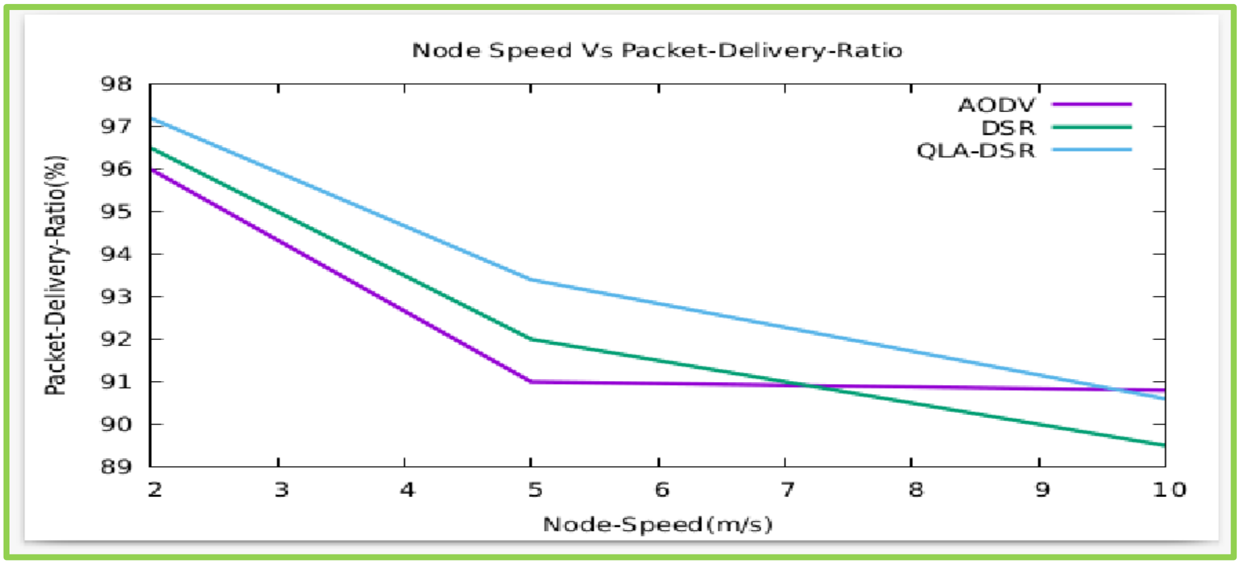



Figure: 5.8 Node Speed vs. Packet Delivery Ratio

- ❖ Average End-to-End Delay

Figure 5.15 shows the average end-to-end delays for QLA-DSR, DSR, and AODV at various speeds. With an increase in the maximum node speed, each of the three protocols experiences an average delay, this is because most of the originating RREQ packets fail to reach their destinations at a relatively fast speed because of the increased likelihood of packet collisions and channel contention brought on by the RREQ packets' excessive redundant retransmissions, this potentially increases the time required for data packets to be transmitted from the source to destination nodes due to the delay of data packets waiting at the source and intermediate nodes' queue. Additionally, broken links might cause stale routes at mobile nodes, which can raise the end-to-end delay of data packets overall.

In comparison to DSR and AODV, QLA-DSR exhibits a notable improvement in average end-to-end delay. This is because, when choosing a route based on the destination node, QLA-DSR takes node buffer size into account. This helps to avoid situations wherein each intermediate node could run out of buffer space before transmitting all the data packets, which increases the likelihood of packet loss. Generally, the average end-to-end delay reduction is obtained at -2.79% of DSR and -5.43 % of AODV.

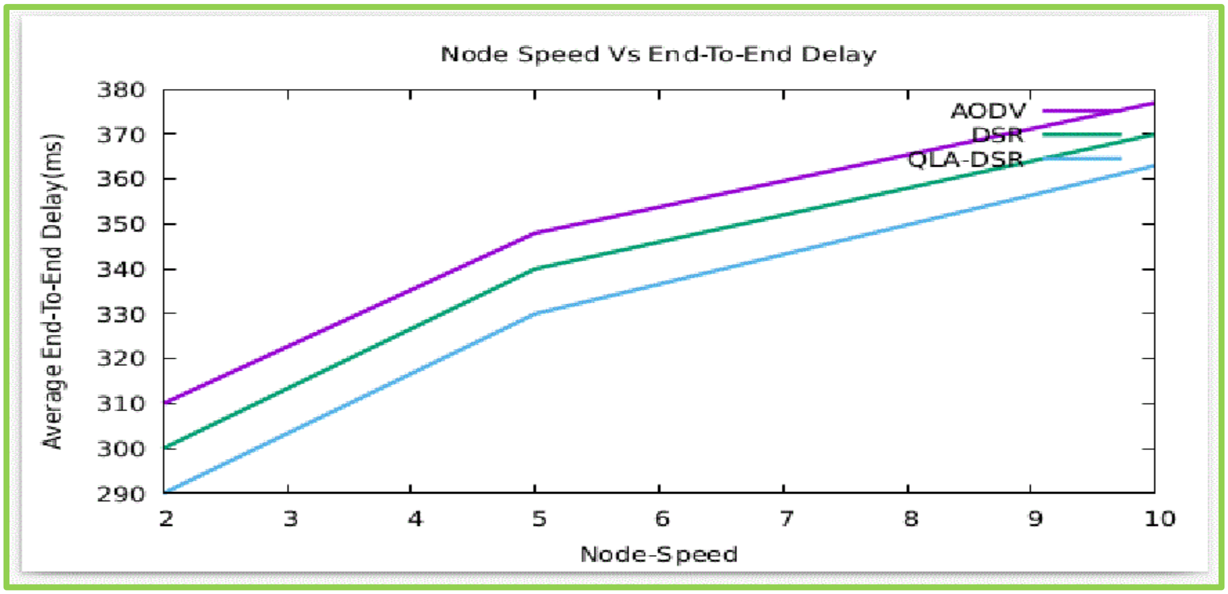



Figure: 5.9 Node Speed vs. Average End-to-End Delay

## 5.9. Answering for the Research Question

**Q1**. How does the QLA-DSR protocol in mobile ad hoc networks address latency and improve network performance?

By considering intermediate node buffer occupancy, QLA-DSR addresses latency during route selection. The protocol avoids the routing traffic when the node is congested, which reduces delays caused by queuing. This approach helps to minimize End-to-End Delay. QLA-DSR improves overall network performance metrics such as packet delivery ratio and throughput by selecting less congested routes.

**Q2**. What queue management algorithm is used to reduce latency in DSR using QLA routing?

The proposed Max-Min-Path (MMP) based queue length maintenance method provides a queue management algorithm in which the smallest queue length is selected during the route discovery process. The RREQ packet updates the information throughout the network. It evaluates the buffer space availability at intermediate nodes to avoid congestion, thereby indirectly managing queuing delays. The decision-making process is based on the real-time queue length information, where nodes with lower buffer occupancy are preferred, helping to reduce packet delay.

**Q3**. How does the QLA-DSR protocol improve routing efficiency in delay-sensitive scenarios compared to existing on-demand routing protocols?

QLA-DSR enhances routing efficiency by integrating real-time queue length data into its decision-making process, allowing it to choose less congested routes. Unlike traditional on-demand routing protocol DSR, which only focuses on factors like hop count or route availability, QLA-DSR considers traffic conditions at intermediate nodes. This leads to more balanced traffic distribution and prevents bottlenecks, making it more suitable for delay-sensitive applications where minimal latency is crucial.

## Summary

We have shown an assessment of the QLA-DSR, AODV, and DSR routing protocols. In the current performance analysis of MANET routing protocols, the most used QoS performance metrics are packet delivery ratio, and average end-to-end delay, and we employed them to quantify the performance of the routing protocols. Several simulation situations have been taken into consideration when doing performance analysis. First, by changing the number of nodes positioned in a specific region, the effect of network density on the routing protocols' performance is examined. Second, by varying the number of source-destination connections (changing the number of sinks), the effect of the loads on the efficiency of the routing protocols is investigated. Lastly, by changing the node speed in the network, the performance analysis of the routing protocols has been observed under changing node mobility.

The simulation findings verify that, with a small but fair increase in average routing overhead, QLA-DSR performs better, especially in average end-to-end delay and packet delivery ratio.

CHAPTER SIX

# 6. Conclusion and Future Works

## 6.1. Conclusions

This study's primary contribution was the creation of a queue length-aware DSR routing protocol, which reduces latency for applications that are sensitive to delays. To improve on the current DSR's overall performance, we propose the QLA-DSR algorithm, which enables the DSR to take into account node queue length during the route selection process.

Designing a Dynamic Source Routing (DSR) protocol that includes the queue length of intermediate nodes for route selection is crucial in delay-sensitive services. The difficulty of minimizing end-to-end delay, particularly in networks with varying traffic volumes, is directly addressed by integrating queue length as a parameter. By avoiding crowded routes and paths that are less prone to experiencing delays from long queue lengths, this algorithm can enhance DSR performance in situations where latency is crucial. As the main target of this study's work was to develop an efficient QoS-aware routing strategy protocol, the researcher has focused on QoS-related metrics, and three evaluation metrics, namely packet delivery ratio, and end-to-end delay, have been selected to evaluate the performance of the algorithm.

The researcher looks at different MANET research projects, article reviews, academic recommendations, and standards, and chooses relevant QoS-related metrics. The metrics listed above have been selected based on these considerations. If mobility, density, traffic patterns, and testing area are taken into consideration, building actual test beds for any predetermined scenario is typically an expensive or even difficult task. To bypass these problems, simulations are needed, and we prefer simulations to test the existing and our designed QLA-DSR protocol. The performance of the DSR, AODV, and QLA-DSR routing protocols has been assessed using the NS 3.28.1 simulator, which was chosen based on the criteria the researcher employed during the analysis. The study considers several evaluation factors while selecting the popular network simulator, including compatibility, flexible dynamic topology design, availability of modules, debugging, tracing support, popularity, and flexibility.

In conclusion, when compared to the AODV and DSR routing protocols, the suggested QLA-DSR routing protocol which is based on the nods queue length route design approach outperforms in terms of packet delivery ratio, and end-to-end delay, improvement of latency was achieved with the expanse of this routing overhead when the number of nodes, source-destination connections, and mobility speed increased. The simulation results of QLA-DSR when compared to DSR and AODV showed an average improvement in all three scenarios of 3.56% and 4.54% in average end-to-end delay, respectively. This confirms that QLA-DSR is better for delay-sensitive applications than AODV and DSR routing protocols.

## 6.2. Future Work

Better performance is possible for QoS-requiring MANET applications, such as multimedia data transfer, thanks to the suggested queue length-aware routing technique. By taking latency constraints into account, it was promised to address the issues with the current routing protocols and enhance their performance in several areas. The outcome demonstrates better End-to-End Delay, Packet Delivery Ratio, and Packet Loss due to QoS support routing. Nevertheless, several QoS metrics must be satisfied due to the rapidly growing usage of multimedia applications over MANETs; so, future research will concentrate on taking into account other QoS metrics to improve the efficiency of QLA-DSR.

The following are summarized future works of the thesis:

- Efficient forwarding probability using the node’s queue length could be used to control flooding route requests (RREQ) through the network.
- Testing the approach in a real-world test bed.
- Testing the approach with Large Scale Network Model and Size Scenarios.
- Include security issues to cooperate with the QLA-DSR Protocol.

# Refrences


Abd, M., Alsaqour, R., & Abdelhaq, M. (2017). Mobile Ad Hoc Network Energy Cost Algorithm Based on Artificial Bee Colony. *Hindawi*, *2017*. doi:10.1155/2017/4519357

Agidi, I. F. (2023). UNIVERSITY OF GHANA, LEGON IMPROVING THE PERFORMANCE AND EFFICIENCY OF DATA ROUTING IN A MOBILE AD -HOC NETWORK (MANET) Master of Science in Computer Science IMPROVING THE PERFORMANCE AND EFFICIENCY OF DATA ROUTING IN A MOBILE AD -HOC NETWORK (MANET).

Almazok, S., & Bilgehan, B. (2020). A novel dynamic source routing (DSR) protocol based on minimum execution time scheduling and moth flame optimization (MET-MFO). *EURASIP Journal on Wireless Communications and Networking*, *2020*, 219. doi:10.1186/s13638-020-01802-5

Bahuguna, R., Mandoria, H., & Pranavi, T. (2013). *Routing Protocols in Mobile Ad-Hoc Network: A Review* (pp. 52–60). doi:10.1007/978-3-642-37949-9_5

Balhara, S., Prakash, A., & Tripathi, R. (2013). QoS Bandwidth Estimation Scheme for Delay Sensitive Applications in MANETs. *Communications and Network*, *05*. doi:10.4236/cn.2013.51001

Bisoyi, S. K., Sahu, S., & Raman, C. V. (2010). *Performance analysis of Dynamic MANET On-demand (DYMO) Routing protocol*. Retrieved from https://api.semanticscholar.org/CorpusID:112420212

Boukerche, A., Câmara, D., Loureiro, A. A. F., & Figueiredo, C. M. S. (2008). Algorithms for Mobile Ad Hoc Networks. In *Algorithms and Protocols for Wireless and Mobile Ad Hoc Networks* (pp. 1–20). John Wiley & Sons, Ltd. doi:https://doi.org/10.1002/9780470396384.ch1

Boukerche, A. F. M., Turgut, B., Aydin, N., Ahmad, M. Z., Bölöni, L., & Turgut, D. (2011). Routing protocols in ad hoc networks: A survey. *Comput. Networks*, *55*, 3032–3080. Retrieved from https://api.semanticscholar.org/CorpusID:12146750

Bruzgiene, R., Narbutaite, L., & Adomkus, T. (2017). MANET Network in Internet of Things System. In J. H. Ortiz & A. P. de la Cruz (Eds.), *Ad Hoc Networks*. Rijeka: IntechOpen. doi:10.5772/66408

Chatterjee, S., & Das, S. (2015). Ant colony optimization based enhanced dynamic source routing algorithm for mobile Ad-hoc network. *Information Sciences*, *295*. doi:10.1016/j.ins.2014.09.039

Clausen, T. H., Dearlove, C., Jacquet, P., & Herberg, U. (2014). RFC7181: The Optimized Link State Routing Protocol version 2. *The Internet Engineering Task Force*. doi:10.17487/RFC7181

Conti, M., & Giordano, S. (2014). Mobile ad hoc networking: milestones, challenges, and new research directions. *IEEE Communications Magazine*, *52*(1), 85–96. doi:10.1109/MCOM.2014.6710069

Dugaev, D., Zinov, S., Siemens, E., & Shuvalov, V. P. (2015). A survey and performance evaluation of ad-hoc multi-hop routing protocols for static outdoor networks. *2015 International Siberian Conference on Control and Communications (SIBCON)*, 1–11. Retrieved from https://api.semanticscholar.org/CorpusID:8357683

Hekmat, R. (2006). *Ad-hoc Networks: Fundamental Properties and Network Topologies*. *Ad-hoc Networks: Fundamental Properties and Network Topologies*. doi:10.1007/1-4020-5166-2

Hinds, A., Ngulube, M., Zhu, S., & Al-Aqrabi, H. (2013). A review of routing protocols for mobile ad-hoc NETworks (MANET). *International Journal of Information and Education Technology*, *3*, 1–5. doi:10.7763/IJIET.2013.V3.223

Johari, L., & Mishra, D. (2020). An Enhanced Queuing Model for Route Optimization in Ad Hoc Networks Based o n Simulation. *International Journal of Innovative Technology and Exploring Engineering*, *9*, 226–231. doi:10.35940/ijitee.L7964.1091220

Johnson David B. and Maltz, D. A. (1996). Dynamic Source Routing in Ad Hoc Wireless Networks. In H. F. Imielinski Tomasz and Korth (Ed.), *Mobile Computing* (pp. 153–181). Boston, MA: Springer US. doi:10.1007/978-0-585-29603-6_5

Kalla, M., Panwar, A., Chakrabarti, P., & Purohit, A. (2016). *QOS AWARE BANDWIDTH CONSTRAINED PRECEDENCE BASED ROUTING PROTOCOL FOR MOBILE AD-HOC NETWORKS*. Retrieved from https://api.semanticscholar.org/CorpusID:212484106

Kannhavong, B., Nakayama, H., Nemoto, Y., Kato, N., & Jamalipour, A. (2007). A survey of routing attacks in mobile ad hoc networks. *IEEE Wireless Communications*, *14*(5), 85–91. doi:10.1109/MWC.2007.4396947

Khader, S., Timcenko, V., Stojanovic, M., Saadawi, T., Shrivastava, L., Bhadauria, S. S., … quot. (2013). *Quality of Service (QoS) Routing in Mobile Ad-hoc Network (MANET) using AODV protocol: Cross-Layer Approach*. Retrieved from https://api.semanticscholar.org/CorpusID:5252538

Khan, B., Olanrewaju, R., Anwar, F., Najeeb, A., & Yaacob, M. (2018). A Survey on MANETs: Architecture, Evolution, Applications, Security Issues and Solutions. *Indonesian Journal of Electrical Engineering and Computer Science*, *12*, 832–842. doi:10.11591/ijeecs.v12.i2.pp832-842

Lakhtaria, K. (2010). Analyzing Zone Routing Protocol in MANET Applying Authentic Parameter.

Liang Qing AND Lin, T. A. N. D. W. F. A. N. D. Z. F. A. N. D. X. W. (2021). A dynamic source routing protocol based on path reliability and link monitoring repair. *PLOS ONE*, *16*(5), 1–12. doi:10.1371/journal.pone.0251548

Ma, N., Lee, J., Li, J., & Liu, Y. (2023). Improvement and Implementation of an Adaptive Congestion DSR Routing Algorithm on Chip. Research Square. doi:10.21203/rs.3.rs-1433046/v1

Macker, J. P., & Corson, M. S. (2002). Mobile ad hoc networking and the IETF. *SIGMOBILE Mob. Comput. Commun. Rev.*, *6*(1), 3–4. doi:10.1145/511226.511232

Malwe, S., Taneja, N., & Biswas, G. (2017). Enhancement of DSR and AODV Protocols Using Link Availability Prediction. *Wireless Personal Communications*, *97*. doi:10.1007/s11277-017-4733-y

Manickavelu, D., & Vaidyanathan, R. U. (2014). Particle swarm optimization (PSO)-based node and link lifetime prediction algorithm for route recovery in MANET. *EURASIP Journal on Wireless Communications and Networking*, *2014*(1), 107. doi:10.1186/1687-1499-2014-107

Mirza, S. , & B. S. Z. (2018). Introduction to MANET. International Research Journal of Engineering and Technology.

Nassir, H., Refoufi, A., & Harrag, A. (2019). New NSGA-II-based OLSR self-organized routing protocol for mobile ad hoc networks. *Journal of Ambient Intelligence and Humanized Computing*, *10*, 1–21. doi:10.1007/s12652-018-0947-4

Perkins, C. E. (2001). *Ad hoc networking: an introduction*. Retrieved from https://api.semanticscholar.org/CorpusID:1201094

Pramanik, A., Choudhury, B., Choudhury, T., Arif, W., & Mehedi, J. (2015). *Simulative Study of Random Waypoint Mobility Model for Mobile Ad hoc Networks*.

Prasath, N., & Sreemathy, J. (2019). Optimized dynamic source routing protocol for MANETs. *Cluster Computing*, *22*(5), 12397–12409. doi:10.1007/s10586-017-1638-1

Rukmani, P., & Ganesan, R. (2013). Scheduling Algorithm for Real Time Applications in Mobile ad hoc Network With Opnet Modeler. *Procedia Engineering*, *64*, 94–103. doi:https://doi.org/10.1016/j.proeng.2013.09.080

Sampada G. D. & Taha, A. (2015). Mobile Ad-Hoc Networks Its Advantages and Challenges, Sampada Ganesh Datey, Taha Ansari, International Journal of Electrical and Electronics Research, ISSN 2348-6988 (online), Research Publish Journals. Retrieved from https://www.researchpublish.com/papers/mobile-ad-hoc-networks-its-advantages-and-challenges

Sarao, P., & Sharma, M. (2022). Reactive and Proactive Route Evaluation in MANET. *Journal of Communications*, 143–149. doi:10.12720/jcm.17.2.143-149

Simaremare, H., Abouaissa, A., Sari, R. F., & Lorenz, P. (2015). Security and performance enhancement of AODV routing protocol. *International Journal of Communication Systems*, *28*, 2003–2019. Retrieved from https://api.semanticscholar.org/CorpusID:2212088

Temesgen, E., Yitayal, E., & Berehan, D. (2019). DEPARTMENT OF INFORMATION TECHNOLOGY Queue Length Aware Ad Hoc on Demand Distance Vector Routing Protocol for Mobile Ad Hoc Networks. Retrieved from http://197.156.88.199/bitstream/handle/123456789/334/Eyob_Thesis_Finalized.pdf?sequence=1

Vasiliou Alexandros, E. A. (2006). MANETs for environmental monitoring. doi:10.1109/ITS.2006.4433383

Zhang, D., Liu, S., Liu, X., Zhang, T., & Cui, Y. (2018). Novel dynamic source routing protocol (DSR) based on genetic algorithm-bacterial foraging optimization (GA-BFO). *International Journal of Communication Systems*, *31*(18), e3824. doi:https://doi.org/10.1002/dac.3824

# APPENDIX

## Qla-Dsr source code

```
#include <fstream>
#include <iostream>
#include <cmath>
#include <sstream>
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/mobility-module.h"
#include "ns3/wifi-module.h"
#include "ns3/qla-module.h"
#include "ns3/qla-helper.h"
#include "ns3/qla-main-helper.h"
#include "ns3/applications-module.h"
#include "ns3/flow-monitor-module.h"
#include "ns3/flow-monitor-helper.h"
#include "ns3/qla-fs-header.h"
#include "ns3/qla-maintain-buff.h"
#include "ns3/qla-network-queue.h"
#include "ns3/qla-option-header.h"
#include "ns3/qla-options.h"
#include "ns3/qla-errorbuff.h"
#include "ns3/qla-rcache.h"
#include "ns3/qla-routing.h"
#include "ns3/qla-gratuitous-reply-table.h"
#include "ns3/qla-rreq-table.h"
#include "ns3/qla-passive-buff.h"
#include "ns3/qla-rsendbuff.h"
#include "ns3/config-store-module.h"
#include "ns3/qla-network-queue.h"
#include "ns3/qla-routing-protocol.h"
#include "ns3/qla-queue.h"
#include "ns3/qla-rreq-header.h"
#include "ns3/packet.h"
#include "ns3/netanim-module.h"
#include "ns3/qla-routing-protocol.h"
#include "my_functions.h"
#include "ns3/queue-disc.h"
using namespace ns3;
NS_LOG_COMPONENT_DEFINE ("queue-length-aware");
class QueueLength {
public:
    QueueLength();
    void Run(int nSinks, double txp, double maxQueueLength, std::string
CSVfileName);
    int GetmaxQueueLength(ns3::Ptr<QueueDisc> queueDisc) {
    return queueDisc->GetNPackets(); // Get number of packets directly
}
    std::string CommandSetup(int argc, char **argv);
  double CalculateMetric (double maxQueueLength, double otherMetrics,
double alpha, double beta) {
    return (alpha * maxQueueLength) + (beta * otherMetrics);
```

```
}
double GetRouteDelay(Ipv4Address address) {
    return 0.0; // Placeholder for actual delay calculation
}
Ipv4Address SelectBestRoute(ns3::Ptr<QueueDisc> queueDisc,
std::vector<Ipv4Address> addresses, double alpha, double beta) {
    double bestMetric = std::numeric_limits<double>::max();
    Ipv4Address bestRoute;
    for (const auto& route : addresses) {
        double queueLength = GetmaxQueueLength(queueDisc); // Get the current
queue length
        double delay = GetRouteDelay(route); // Get the delay for the route
        double metric = CalculateMetric(queueLength, delay, alpha, beta);
        if (metric < bestMetric) {
            bestMetric = metric;
            bestRoute = route;
        }
    }
    return bestRoute; // Return the best routeðŸ˜Šï¸□
}
 private:
    Ptr<Socket> SetupPacketReceive(Ipv4Address addr, Ptr<Node> node);
    void ReceivePacket(Ptr<Socket> socket);
    void CalculateAndPrintFinalResults(); // Declaration added here
    void SetmaxQueueLength(uint32_t maxQueueLength);
    uint32_t port;
    uint32_t bytesTotal;
    uint32_t packetsReceived;
    std::string m_CSVfileName;
    int m_nSinks;
    std::string m_protocolName;
    std::string m_queueType;
    double m_txp;
    bool m_traceMobility;
    uint32_t SentPackets;
    uint32_t ReceivedPackets;
    uint32_t LostPackets;
    double totalDelay;
    uint32_t maxQueueLength;
    uint32_t queueSize = 50;
   double alpha = 0.7;
   double beta = 0.3;
    NodeContainer m_nodes;
    Ipv4InterfaceContainer m_interfaces;
    double m_simulationTime;
     };
QueueLength::QueueLength()
    : port(9),
      bytesTotal(0),
      packetsReceived(0),
      m_CSVfileName("queue-length.output.csv"),
      m_traceMobility(false),
      SentPackets(0),
      ReceivedPackets(0),
      LostPackets(0),
      totalDelay(0.0)
      {
```

```
   uint16_t port = 9;
    std::vector<ApplicationContainer> apps;
    std::vector<Ipv4Address> potentialRoutes;
    for (uint32_t i = 1; i < m_nodes.GetN(); ++i) {
        OnOffHelper onOff("ns3::UdpSocketFactory",
InetSocketAddress(m_interfaces.GetAddress(i), port));
        onOff.SetConstantRate(DataRate("64kbps"));
        apps.push_back(onOff.Install(m_nodes.Get(0)));
        apps.back().Start(Seconds(1.0));
        apps.back().Stop(Seconds(m_simulationTime));
    }
}
static inline std::string
PrintReceivedPacket(Ptr<Socket> socket, Ptr<Packet> packet, Address
senderAddress) {
    std::ostringstream oss;
    oss << Simulator::Now().GetSeconds() << " " << socket->GetNode()-
>GetId();
    if (InetSocketAddress::IsMatchingType(senderAddress)) {
        InetSocketAddress addr =
InetSocketAddress::ConvertFrom(senderAddress);
        oss << " received one packet from " << addr.GetIpv4();
    } else {
        oss << " received one packet!";
    }
    return oss.str();
}
void QueueLength::ReceivePacket(Ptr<Socket> socket) {
    Ptr<Packet> packet;
    Address senderAddress;
    while ((packet = socket->RecvFrom(senderAddress))) {
        bytesTotal += packet->GetSize();
        packetsReceived += 1;
        NS_LOG_UNCOND(PrintReceivedPacket(socket, packet, senderAddress));
    }
}
Ptr<Socket> QueueLength::SetupPacketReceive(Ipv4Address addr, Ptr<Node> node)
{
    TypeId tid = TypeId::LookupByName("ns3::UdpSocketFactory");
    Ptr<Socket> sink = Socket::CreateSocket(node, tid);
    InetSocketAddress local = InetSocketAddress(addr, port);
    sink->Bind(local);
    sink->SetRecvCallback(MakeCallback(&QueueLength::ReceivePacket, this));
    return sink;
}
std::string QueueLength::CommandSetup(int argc, char **argv) {
    CommandLine cmd;
    cmd.AddValue("CSVfileName", "The name of the CSV output file name",
m_CSVfileName);
    cmd.AddValue("traceMobility", "Enable mobility tracing",
m_traceMobility);
    cmd.AddValue ("alpha", "Weight for queue length", alpha);
    cmd.AddValue ("beta", "Weight for other metrics (delay)", beta);
    cmd.Parse(argc, argv);
    return m_CSVfileName;
}
int main(int argc, char *argv[]) {
```

```
    QueueLength experiment;
    std::string CSVfileName = experiment.CommandSetup(argc, argv);
    Packet::EnablePrinting();
    // Blank out the last output file and write the column headers
    std::ofstream out(CSVfileName.c_str());
    out << "SimulationSecond," <<
           "ReceiveRate," <<
           "PacketsReceived," <<
           "NumberOfSinks," <<
           "RoutingProtocol," <<
           "TransmissionPower" <<
           std::endl;
    out.close();
    int nSinks = 10;
    double txp = 16;
    uint32_t maxQueueLength = 50;
    experiment.Run(nSinks, txp, maxQueueLength, CSVfileName);
      }
void QueueLength::Run(int nSinks, double txp, double maxQueueLength,
std::string CSVfileName) {
    m_queueType = "ns3::CMUPriQueue";
    Packet::EnablePrinting();
    m_nSinks = nSinks;
    m_txp = txp;
    m_CSVfileName = CSVfileName;
    int nWifis = 50;
    double TotalTime = 30.0;
    std::string rate("2048bps");
    std::string phyMode("DsssRate11Mbps");
    std::string tr_name("queue-length-aware");
    int nodeSpeed = 10; // in m/s
    int nodePause = 0; // in s
    maxQueueLength = maxQueueLength;
    m_protocolName = "protocol";
    Config::SetDefault  ("ns3::OnOffApplication::PacketSize",StringValue
("64"));
    Config::SetDefault("ns3::OnOffApplication::DataRate", StringValue(rate));
    // Set Non-unicastMode rate to unicast mode
    Config::SetDefault("ns3::WifiRemoteStationManager::NonUnicastMode",
StringValue(phyMode));
    // Create adhoc nodes
    NodeContainer adhocNodes;
    adhocNodes.Create(nWifis);
    // Setting up WiFi PHY and channel using helpers
    WifiHelper wifi;
    wifi.SetStandard(WIFI_PHY_STANDARD_80211b);
    YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default();
    YansWifiChannelHelper wifiChannel;
wifiChannel.SetPropagationDelay("ns3::ConstantSpeedPropagationDelayModel");
    wifiChannel.AddPropagationLoss("ns3::FriisPropagationLossModel");
    wifiPhy.SetChannel(wifiChannel.Create());
    // Add a MAC and disable rate control
    WifiMacHelper wifiMac;
    wifi.SetRemoteStationManager("ns3::ConstantRateWifiManager",
                                 "DataMode", StringValue(phyMode),
                                 "ControlMode", StringValue(phyMode));
    wifiPhy.Set("TxPowerStart", DoubleValue(txp));
```

```
    wifiPhy.Set("TxPowerEnd", DoubleValue(txp));
    wifiMac.SetType("ns3::AdhocWifiMac");
    NetDeviceContainer adhocDevices = wifi.Install(wifiPhy, wifiMac,
adhocNodes);
    // Set up mobility
    MobilityHelper mobilityAdhoc;
    int64_t streamIndex = 0; // used to get consistent mobility across
scenarios
    ObjectFactory pos;
    pos.SetTypeId("ns3::RandomRectanglePositionAllocator");
    pos.Set("X",
StringValue("ns3::UniformRandomVariable[Min=0.0|Max=1000.0]"));
    pos.Set("Y",
StringValue("ns3::UniformRandomVariable[Min=0.0|Max=1000.0]"));
    Ptr<PositionAllocator> taPositionAlloc = pos.Create()-
>GetObject<PositionAllocator>();
    streamIndex += taPositionAlloc->AssignStreams(streamIndex);
    std::stringstream ssSpeed;
    ssSpeed << "ns3::UniformRandomVariable[Min=0.0|Max=" << nodeSpeed << "]";
    std::stringstream ssPause;
    ssPause << "ns3::ConstantRandomVariable[Constant=" << nodePause << "]";
    mobilityAdhoc.SetMobilityModel("ns3::RandomWaypointMobilityModel",
                                   "Speed", StringValue(ssSpeed.str()),
                                   "Pause", StringValue(ssPause.str()),
                                   "PositionAllocator",
PointerValue(taPositionAlloc));
    mobilityAdhoc.SetPositionAllocator(taPositionAlloc);
    mobilityAdhoc.Install(adhocNodes);
    streamIndex += mobilityAdhoc.AssignStreams(adhocNodes, streamIndex);
    QlaHelper qla;
    Ipv4ListRoutingHelper list;
    InternetStackHelper internet;
    internet.Install(adhocNodes);
    qla.Set("MaxQueueLength", UintegerValue(maxQueueLength));
    Ipv4AddressHelper addressAdhoc;
    addressAdhoc.SetBase("10.0.0.0", "255.0.0.0");
    Ipv4InterfaceContainer adhocInterfaces;
    adhocInterfaces = addressAdhoc.Assign(adhocDevices);
    // Install traffic generators
    OnOffHelper onoff1("ns3::UdpSocketFactory", Address());
    onoff1.SetAttribute("OnTime",
StringValue("ns3::ConstantRandomVariable[Constant=1]"));
    onoff1.SetAttribute("OffTime",
StringValue("ns3::ConstantRandomVariable[Constant=0]"));
    for (int i = 0; i < nSinks; i++) {
        Ptr<Socket> sink = SetupPacketReceive(adhocInterfaces.GetAddress(i),
adhocNodes.Get(i));
        AddressValue
remoteAddress(InetSocketAddress(adhocInterfaces.GetAddress(i), port));
        onoff1.SetAttribute("Remote", remoteAddress);
        Ptr<UniformRandomVariable> var =
CreateObject<UniformRandomVariable>();
        ApplicationContainer temp = onoff1.Install(adhocNodes.Get(i +
nSinks));
        temp.Start(Seconds(var->GetValue(0.0, 1.0)));
        temp.Stop(Seconds(TotalTime));
    }
```

```
    FlowMonitorHelper flowmon;
    Ptr<FlowMonitor> monitor = flowmon.InstallAll();
    // Run simulation
    wifiPhy.EnablePcapAll(tr_name);
    Simulator::Stop(Seconds(TotalTime));
    AnimationInterface anim("Qla.xml");
    Simulator::Run();
    monitor->CheckForLostPackets();
        Ptr<Ipv4FlowClassifier> classifier =
DynamicCast<Ipv4FlowClassifier>(flowmon.GetClassifier());
    std::map<FlowId, FlowMonitor::FlowStats> stats = monitor->GetFlowStats();
    for (std::map<FlowId, FlowMonitor::FlowStats>::const_iterator i =
stats.begin(); i != stats.end(); ++i) {
             SentPackets += i->second.txPackets;
        ReceivedPackets += i->second.rxPackets;
        LostPackets += i->second.txPackets - i->second.rxPackets;
        totalDelay += i->second.delaySum.GetSeconds();
          }
    CalculateAndPrintFinalResults();
    Simulator::Destroy();
}
void QueueLength::CalculateAndPrintFinalResults() {
    double packetDeliveryRatio = (ReceivedPackets * 100.0) / SentPackets;
    double packetLossRatio = (LostPackets * 100.0) / SentPackets;
    double averageDelay = totalDelay / ReceivedPackets;
 NS_LOG_UNCOND("--Total Results of the simulation----"<<std::endl);
    NS_LOG_UNCOND("Sent Packets: " << SentPackets);
    NS_LOG_UNCOND("Received Packets: " << ReceivedPackets);
    NS_LOG_UNCOND("Lost Packets: " << LostPackets);
    NS_LOG_UNCOND("Packet Delivery Ratio: " << packetDeliveryRatio << "%");
    NS_LOG_UNCOND("Packet Loss Ratio: " << packetLossRatio << "%");
    NS_LOG_UNCOND("Average Delay: " << averageDelay << " s");
  }
```