IMPROVEMENT OF ROUTING PROTOCOLS FOR UNMANNED FLYING AD HOC NETWORKS (UFANETS) BY USING CROSS LAYER METRICS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

OĞUZ ALKIŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

NOVEMBER 2019

Approval of the thesis:

IMPROVEMENT OF ROUTING PROTOCOLS FOR UNMANNED FLYING AD HOC NETWORKS (UFANETS) BY USING CROSS LAYER METRICS

submitted by **OĞUZ ALKIŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Eng.** ________________

Prof. Dr. İlkay Ulusoy
Supervisor, **Electrical and Electronics Eng., METU** ________________

Examining Committee Members:

Prof. Dr. Ece Güran Schmidt
Electrical and Electronics Engineering, METU ________________

Prof. Dr. İlkay Ulusoy
Electrical and Electronics Eng., METU ________________

Prof. Dr. Emre Aktaş
Electrical and Electronics Engineering, Hacettepe University ________________

Assoc. Prof. Dr. Cüneyt F. Bazlamaçcı
Electrical and Electronics Engineering, METU ________________

Assist. Prof. Dr. Gökhan Muzaffer Güvensen
Electrical and Electronics Engineering, METU ________________

Date: 27.11.2019

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Oğuz Alkış

Signature:

ABSTRACT

IMPROVEMENT OF ROUTING PROTOCOLS FOR UNMANNED FLYING AD HOC NETWORKS (UFANETS) BY USING CROSS LAYER METRICS

Alkış, Oğuz
Master of Science, Electrical and Electronics Engineering
Supervisor: Prof. Dr. İlkay Ulusoy

November 2019, 127 pages

Unmanned Air Vehicles (UAV) have been used to perform different type of missions. Many of these civilian and military missions require information exchange between UAVs, which is performed by using routing protocols for wireless ad hoc networks. An efficient routing protocol for unmanned flying ad hoc networks (UFANETs) plays a critical role for data transmission during various applications. In the literature, there are many studies for mobile ad hoc networks (MANET) routing protocols. Due to high mobility and frequent topology changes, most of these studies are not applicable to flying ad hoc networks. Although there are some studies for UFANETs, none of them use cross layer metrics for routing decision and maintenance. In this thesis, a new routing lifetime decision method is proposed for UFANETs by using cross layer metrics. The proposed method uses signal to noise ratio (SNR), frequency offset due to Doppler Effect and received signal power (RSSI or RSRP) parameters as cross layer metrics. This novel approach is applied to Ad Hoc On Demand Distance Vector (AODV) routing protocol and controls the remaining routing life time of the active routes. Simulations of the new approach and basic AODV protocol are done by using INET Framework of OMNET++ simulation tool. Results show that the proposed method improves the routing lifetime decision performance in terms of end-to-end latency, total number of routing overhead messages and successful data transmission ratio.

Keywords: Ufanets, Ad Hoc routing, Cross Layer Metrics

ÖZ

ÇAPRAZ KATMAN METRIKLERI KULLANARAK İNSANSIZ UÇAN TASARSIZ AĞLARIN (İUTA) YÖNLENDIRME PROTOKOLLERININ İYİLEŞTİRİLMESİ

Alkış, Oğuz
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği
Tez Danışmanı: Prof. Dr. İlkay Ulusoy

Kasım 2019, 127 sayfa

İnsansız Hava Araçları (İHA) sivil ve askeri kullanımlar için farklı türde görevleri yerine getirmek amacıyla kullanılmaktadır. Bu ticari ve askeri görevlerin birçoğu İHA lar arasında bilgi alışverişine ihtiyaç duymaktadır. Bu bilgi alışverişi, kablosuz tasarsız ağlar için yönlendirme protokolleri kullanılarak gerçekleştirilir. İnsansız uçan tasarsız ağlar (İUTA) için verimli bir yönlendirme protokolü, veri iletimi ve çeşitli uygulamalar için kritik bir rol oynar. Literatürde, mobil tasarsız ağların (MTA) yönlendirme protokolleri için birçok çalışma bulunmaktadır. Yüksek mobilite ve sık topoloji değişiklikleri nedeniyle, bu çalışmaların çoğu uçan tasarsız ağlara uygulanamamaktadır. Öte yandan, İUTA'lar için bazı çalışmalar bulunmakla beraber, bu çalışmaların neredeyse tümü, yönlendirme kararı ve idamesi için çapraz katman metriklerini kullanmaz. Bu tezde, İUTA lar için çapraz katman metriklerini kullanarak yönlendirme ömür zamanını belirlemek için yeni bir karar verme yöntemi önerilmektedir. Önerilen yöntem çapraz katman metrikleri olarak sinyal gürültü oranı, dopplere bağlı frekans kaymasını ve alıcı sinyal gücü parametrelerini kullanmaktadır. Bu yeni yaklaşım, Tasarsız İsteğe Bağlı Mesafe Vektörü (TİBMV) yönlendirme protokolüne uygulanmış ve yönlendirme ömür zamanını kontrol etmiştir. Yeni yöntemin ve temel TİBMV protokolünün simülasyonları, OMNET ++ simülasyon aracının INET çerçeve yapısı kullanılarak gerçekleştirilmiştir. Sonuçlar, önerilen yöntemin uçtan uca gecikme, yönlendirme amacıyla fazladan kullanılan mesajların toplam sayısı ve başarılı veri gönderim oranı yönünden yönlendirme ömür zamanına karar verme performansını iyileştirdiğini göstermektedir.

Anahtar Kelimeler: İuta, Tasarsız Yönlendirme, Çapraz Katman Metrikleri

To My Family

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation and thanks to my supervisor, Prof. Dr. İlkay Ulusoy. I would like to thank her for valuable guidance throughout this research. Since I had to suspend this study several times, her patience was a big chance for me. It was a great pleasure to work with her.

A special thanks to my wife, who encouraged me during this study. It was impossible to concentrate on this research without her support. I would like to apologize my three children, since I had to work hard, I could not spend enough time with them.

I would also like to thank my colleague, MSc. Erem Erdem. He encouraged me to return to university and study this thesis.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1. Overview

Unmanned Aerial Vehicle (UAV), also known as “Drone”, is a type of aircraft that operates without a human pilot on board. UAVs can be either operated remotely by pilot at another station (remotely piloted aircraft - RPA) or controlled autonomously (autonomous drones) based on onboard computers. As technology develops, the trend in flying drones tends to be autonomous instead of manually controlled.

Simple taxonomies can be used to categorize UAVs, for instance, in terms of the type of flight (autonomous or remotely controlled), their size (large or small), the type of wings, or their communication capabilities. Regarding the types of wings, there are two main categories: fixed-wing UAVs (FW-UAVs) and rotary-wing UAVs (RW-UAVs). FW-UAVs present longer flight times, higher flight speeds, and have a better aerodynamic design, whereas RW-UAVs are able to perform vertical take-off and landing (VTOL), exhibit greater stability (can control yaw, pitch, roll, and throttle), and have the capacity to hover over static points.

Unmanned Air Vehicles (UAV) have enjoyed growing importance during the last two decades. UAV Systems today are used to perform a multitude of missions both military and civilian. UAVs have been used for military applications during last 3 decades and typical UAV military missions are Reconnaissance, Targeting and Fire Control, Attack, Suppression of Enemy Air Defense (SEAD), Communication Intelligence (COMINT) and Electronic Intelligence (ELINT), Jamming and etc.

On the other side, commercial entities and governments have come to realize that UAVs have multiple uses, which include:

- Aerial photography for journalism and film
- Express shipping and delivery
- Gathering information or supplying essentials for disaster management
- Thermal sensor drones for search and rescue operations
- Aerial Intelligence for Fire Fighting
- Geographic mapping of inaccessible terrain and locations
- Building safety inspections
- Precision crop monitoring
- Unmanned cargo transport
- Law enforcement and border control surveillance
- Storm tracking and forecasting hurricanes and tornadoes
- Aerial Surveillance & Nuclear Detection for Nuclear Plants
- Wireless sensor network applications
- Critical Infrastructure Protection

Many of these commercial and military missions require exchange of information between the UAVs, manned aircrafts and ground based systems. This information exchange is performed by a wireless communication. Compared to a single UAV, multi-UAV systems are much more efficient with faster multitasking ability, longer network lifetime, and higher scalability. However, they also bring many challenging problems due to the unique characteristics of UAVs (e.g., high mobility and dense or sparse deployment). One of the most important basic problems is the cooperative communication between UAVs. In order to efficiently transfer packets, a swarm of UAVs communicate and collaborate with each other to self-organize into a network, called a UAV flying ad hoc network (UFANET). Mobile Ad Hoc Network (MANET) is a collection of mobile nodes that communicate without relying on any pre-existing infrastructure. In a Mobile Ad hoc Network (MANET), mobile nodes move around arbitrarily, nodes may join and leave at any time, and the resulting topology is constantly changing. Routing in a MANET is challenging because of the dynamic topology and the lack of an existing fixed infrastructure. UFANET is a special case of mobile ad hoc networks (MANETs) that are characterized by a high degree of mobility and frequent topology changes. On the other hand, FANET can also be classified as a subset of Vehicular Ad Hoc Network (VANET), which is also a subgroup of MANET. The relationship between these types of ad hoc networks is illustrated in Figure 1.1.

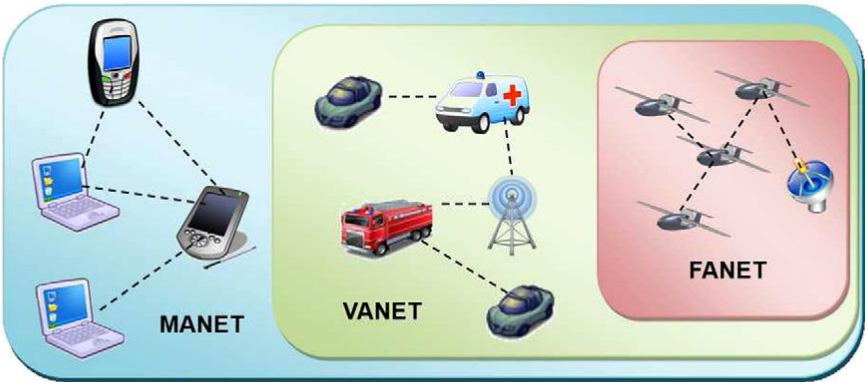

Figure 1.1. MANET, VANET and UFANET [1]

As a research area of interest, UFANET has common characteristics with these networks, and it also has several unique design challenges (*Table 1.1*):

- Mobility degree of FANET nodes is much higher than the mobility degree of MANET or VANET nodes. While typical MANETs are mobile nodes such as mobile phones, laptops and VANET nodes are vehicles such as cars, bikes, FANET nodes fly in the sky.
- Due to the high mobility of FANET nodes, the topology changes more frequently than the network topology of a typical MANET or VANET.
- Differences between FANET and the other ad hoc network operating environments affect the radio propagation characteristics. MANET and VANET nodes are remarkably close to the ground, and in many cases, there is no line-of-sight between the sender and the receiver. Therefore, radio signals are mostly affected by the geographical structure of the terrain. However, FANET nodes can be far away from the ground and in most of the cases, there is a line-of-sight between UAVs.

- Distances between FANET nodes are much longer than in the MANETs and VANETs. In order to establish communication links between UAVs, the communication range must also be longer than in the MANETs and VANETs.

Table 1.1 *Comparison of MANET, VANET and UFANET*

Ad Hoc Network Types / Characteristics	MANET	VANET	UFANET
Mobility Model	Random	Steady – Manhattan mobility models	Usually predetermined, but special mobility models for independent multi-UAV systems
Node Mobility	Lower	Low	High for fixed-wing Medium for rotary-wing
Node Speed	Lower (6 km/h)	Medium - High (20 -100 km/h)	Medium - High (50 - 100 km/h)
Topology Change	Slow	Average	High
Wireless channel for radio propagation	Close to ground, LoS communication is not available for most of the cases.	Close to ground, LoS communication is not available for most of the cases.	Line of sight (LoS) communication is available for most of the cases
Energy Constraints	Medium	Low	Medium for fixed wing High for rotary-wing

The Open Systems Interconnection (OSI) reference model proposed by the International Organization for Standardization (ISO) consists of seven protocol stacks in layers which are ordered from layer 1 (the lowest) to layer 7 (the highest). These seven layers are (from lower to higher) the physical layer, data link layer, network layer, transport layer, session layer, presentation layer, and the application layer. OSI model was primarily created for wired networks. This layered model also is used for wireless networks, but modified with different applications. Like MANET, UFANET is also a wireless network and does not need any infrastructure and preplanning for network establishment. UFANET protocol stack consists of five layers: physical layer, data link layer, network layer, transport layer and application layer. The lower four layers have the same name but the fifth layer in the UFANET model is equivalent to the combined session, presentation and application layers of the OSI model. The other main difference between these protocol stacks lies in the network layer. Mobile nodes, which can be host or router in UFANETs, use an ad hoc routing protocol to route packets. Protocol stack of each layered model is given in *Table 1.2*.

Table 1.2 *Protocol Stack for UFANET and OSI Model*

UFANETs Layered Model	**OSI Layered Model**
Application Layer	Application Layer
	Presentation Layer
	Session Layer
Transport Layer	Transport Layer
Network Layer	Network Layer
Data Link Layer	Data Link Layer
Physical Layer	Physical Layer

Routing issue is one of the most challenging and interesting research areas in MANETs and UFANETs. Generally, the main function of routing in a network is to detect and maintain the optimal route to send data packets between a source and destination via intermediate node(s).

There are two main types of MANET routing protocols: reactive and proactive (although there are others which don't fit into either category). Reactive or on-demand routing protocols update routing information when there is an immediate demand for it, i.e. one of the nodes wants to send a packet (and there is no working route to the destination). Then, they exchange route discovery messages, and forward the packet. The routes stay the same until there is an error in a packet's forwarding, i.e. the packet cannot be forwarded anymore due to a change in the network topology. Examples of reactive MANET routing protocols include Ad hoc On Demand Distance Vector (AODV) [8], Dynamic Source Routing (DSR) [7], Dynamic MANET On Demand (DYMO) [9] etc.

Proactive or table-driven routing protocols continuously maintain routing information, so the routes in the network are always up to date. This typically involves periodic routing maintenance messages exchanged throughout the network. These types of protocols use more maintenance transmissions than reactive protocols in order to make sure the routing information is always up-to-date (they update it even when there is no change in the network topology). Examples of reactive MANET routing protocols include Destination Sequenced Distance Vector Routing (DSDV) [10], Optimized Link State Routing (OLSR) [11] etc.

Reactive protocols require less overhead than proactive protocols (there are no concerning routing when the routes don't change), but also might react more slowly to changes in the network topology. In the case of proactive protocols, due to the up-to-date nature of routing information, latency is lower than in the case of reactive protocols.

1.2. Problem Statement

An efficient communication or routing protocol between UAVs plays a vital role in data transmission during various practical applications. Over the past few years, there has been a rapidly growing amount of research on routing protocols in ad hoc networks, but they cannot be directly used for UFANETs. It is a challenging task to develop an efficient routing protocol for UFANETs. In order to achieve challenging tasks, UAVs must be able to communicate reliably.

Due to be developed for wired networks, the OSI model defines strict layered protocol design. On the other hand, MANETS, also UFANETs, object to strict layered protocol design because of their dynamic nature, infrastructure-less architecture, limited resources, mobility of nodes, time varying unstable links and topology. The concept of cross-layer design is based on architecture where the layers can exchange information in order to improve the overall network performances. By using physical layer metrics, like signal to noise ratio, Doppler Effect and received power, quality of wireless channel and distance changes between nodes can be predicted and this predicted information can be used for route decision and maintenance. Using the cross-layer design or inter-layer information, a more reliable route can be found and route maintenance can be provided more efficiently by optimizing the routing life time.

In the literature, there are many studies for MANET routing protocols. Due to high mobility nature of UFANETS, most of these studies cannot be applicable to UFANETs. On the other hand, there are a few routing studies for UFANETs, none of which use the cross layer metrics for route decision and maintenance. As a result, due to lack of wireless channel quality information, the proposed routing approaches do not provide enough routing performance for UFANETs. There are only a few studies which use some cross layer metrics for routing protocol for UFANETs. Pros and cons of these studies are mentioned in Chapter 2 of this thesis.

In this thesis, first of all, applicability of flat routing protocols, especially the reactive AODV routing protocol, for UFANETs is investigated and simulated for different network topologies. The topologies have different mobility models and number of UAV nodes. After simulating these topologies, the routing performance metrics, described in [2] are obtained in terms of routing overhead, end-to-end delay and average packet delivery ratio. Then, by using cross layer metrics, the performance improvement techniques for UFANETS routing protocols are proposed and simulated with the same network topologies as the original ones. After observing simulation results, the performance improvement in terms of routing overhead, end-to-end delay and average packet delivery ratio are analyzed and compared to the original approach.

1.3. Evaluation of MANET Routing Protocols: Network Simulation Tools

Different simulators have been developed in the recent years with powerful features that cover different aspects of MANET. Simulators provide an economical way to evaluate newly developed protocols or any algorithms. Otherwise it will be very expensive to test all protocols on real platforms. Simulators are used to calculate the accuracy, throughput, scalability, latency traffic ratio etc. parameters of a protocol.

The most popular network simulation tools are Network Simulator 2 (NS-2) and 3 (NS-3), OMNET++, OPNET, QualNet, GloMoSim and JIST [3]. All these simulators have their own powerful features like NS-2 provides energy model, ns-3 is very good for documentation, OMNET++ has a very rich class library, OPNET is easy to use, GloMoSim is very scalable, QualNet provides animation tools and JIST is a very powerful simulator. But still there are some open issues as some simulators like NS-2 and NS-3 are very difficult to use. GloMoSim, OMNET++ and NS-2 are not good for documentation. Instead of using a user friendly GUI, some simulators uses only command line interface. Commercial version of various simulators is very expensive and some simulators are very difficult to install and not up to date [3].

There are also some application specific simulation tools written in different languages. AODV-Matlab [4] is an example of these types of simulators. This tool is a simulation of the ad-hoc on-demand distance vector (AODV) routing protocol for wireless networks in MATLAB. This tool uses the familiar and accessible environment of MATLAB to create base-level accessible, open-source, real time AODV routing scheme.

Omnet++ is open source and very flexible network simulator. Also, the INET framework of Omnet++ includes many protocols for simulating MANETs, such as OLSR, DSR, DSDV, DYMO, AODV, 802.11 (a,g) and RSTP. The simulation models are implemented in C++. The network topology and model interconnections are described through the OMNET++’s topology description language, NED, which is very easy to use. Another important point is that the UI allows interacting with the system through a graphical interface (Tkenv) or a command line interface (Cmdenv). Because of these advantages, in this thesis, INET framework of Omnet++ is used as simulation tool.

1.4. Thesis Outline

The thesis is organized into the following chapters:

Chapter 1 Introduction: This chapter gives an introduction to flying ad hoc networks for UAVs and challenging issues for routing in UFANETs. Also aim of this thesis is explained in this chapter. Different simulations developed for networks simulation are given and brief explanations for these simulations are mentioned in this chapter.

Chapter 2 Related Work: This chapter provides related work on routing protocols for MANETs and UFANETs. Also routing protocols which use cross layer metrics are investigated and related works are given in this chapter.

Chapter 3 3. Ad Hoc On-Demand Distance Vector Algorithm For UFANETs: This chapter gives technical details for reactive AODV protocol. Simulation of AODV routing protocol for different mobility models and network topologies are also described in this chapter. Then AODV routing performance results are obtained in this chapter and these results will be compared to the results of improved AODV protocol.

Chapter 4 Improved AODV Routing Protocol and Cross Layer Metrics for UFANETs Routing: Layered model of UFANETs and available cross layer metrics for routing performance improvement are described in this chapter. Then the proposed methods, using physical layer metrics for AODV routing performance, are given in this chapter. Simulation of the proposed methods is described and improved AODV routing performance results are obtained in this chapter. Then the AODV protocol simulation results and improved AODV protocol simulation results are compared in this chapter.

Chapter 5 Conclusion and Future Work: The obtained results are summarized and possible future works are mentioned in this last chapter.

CHAPTER 2

RELATED WORK

Existing routing protocols used in MANETs and VANETs cannot be directly applicable for UFANETs, because the routing protocols designed for UAVs shall be adapted to high mobility of UAVs, rapidly changing network topology and brokenly connected communication links. [5]

2.1. Types of Routing Protocols for UFANETs

Basic UFANETs routing protocols typically fall into two broad categories: Reactive and Proactive. Most of these basic protocols are proposed for MANETS. Because of high mobility and frequent topology changes in FANETs, modified versions of these protocols are proposed for FANETs in literature. Initially, summary of these protocols will be mentioned, and then the related works based on these protocols will be explained.

In reactive routing protocols, a source node finds a route to destination by flooding route request packets into the network. Since the process is on-demand, the route discovery imposes some latency on the overall performance of the network. Also, flooding of route requests may cause buffer overflow and network congestion [6].On the other hand reactive protocols causes lower overhead due to the lack of need to maintain routes, so these protocols are energy efficient. Ad hoc On Demand Distance Vector (AODV) [8], Dynamic Source Routing (DSR) [7] and Dynamic MANET On Demand (DYMO) [9] protocols are examples of reactive routing protocols.

Proactive routing protocols periodically maintain routing information, so the routes in the network are always up to date. This typically involves periodic routing maintenance messages exchanged throughout the network. These types of protocols use more maintenance transmissions than reactive protocols in order to make sure the routing information is always up-to-date. Examples of reactive MANET routing protocols are Destination Sequenced Distance Vector Routing (DSDV) [10], Optimized Link State Routing (OLSR) [11].

Reactive protocols require less overhead than proactive protocols, but also might react more slowly to changes in the network topology. In the case of proactive protocols, due to the up-to-date nature of routing information, latency is lower than in the case of reactive protocols. Due to exchanging routing maintenance messages periodically, reactive protocols have more overhead and need more processing sources than reactive protocols.

There are other types of MANET routing protocols, such as Hybrid (both reactive and proactive) and Position-based (proactive with position information) routing.

A hybrid routing protocol is a combination of proactive and reactive routing protocols, which can overcome the problem of high control message overhead in proactive routing and the long end-to-end delay in reactive routing [5]. Hybrid Routing based on Clustering (HRC) [12] and Reactive-Greedy-Reactive Routing Protocol [6] are examples of hybrid routing protocols. Most of these methods are application dependent and do not use cross layer metrics for improvement of routing performance.

Position based protocols uses location of neighboring nodes to find the availability of routes. Greedy Perimeter Stateless Routing (GPSR) [13] and Recovery Strategy for the Greedy Forwarding Failure (RSGFF) are examples of this type of protocols [14]. These protocols use GPS signals for position information. Accuracy of GPS devices depends on the total number of used satellites for signal processing. The extreme atmospheric conditions and unavailability of GPS satellites cause problems and position information can be inaccurate. Also it is possible that GPS signals can be jammed, in this case, GPS devices do not give reliable information.

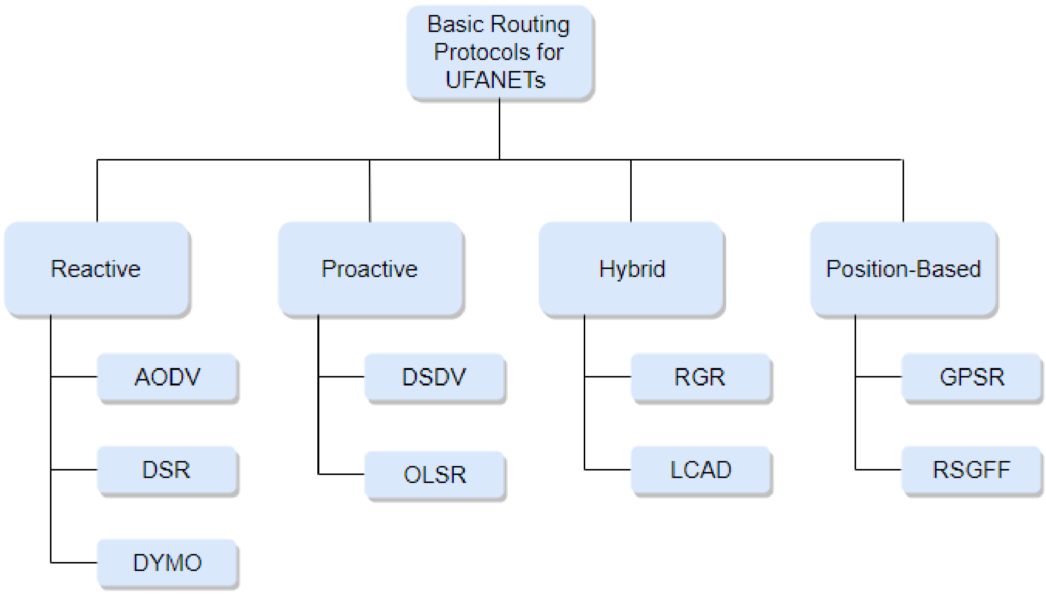

Figure 2.1 Classification of Basic UFANETs Routing Protocols

2.2. Recent Researches for UFANETs Routing

Leono vet al., in [15], compared the AODV and OLSR routing protocols for a sample topology. Although it is stated that the analysis is done for a relaying network, in the simulation, 50 mini-UAVs were moving with RWP mobility model and CBR traffic was generated between randomly selected two nodes with 25 times. In the simulation, IEEE 802.11g standard was used as MAC layer protocol. Simulation results show that AODV outperforms OLSR with respect to packet delivery, throughput and routing overheads. On the other hand, OLSR performs better than AODV with regard to jitter, end to end delay and hop count. Since the AODV is a reactive and OLSR is a proactive routing protocol, these results are expected. This study does not contain any improvement on basic FANET routing protocols, it used AODV and OLSR routing protocols as used in MANETs.

Nayyar [16], worked on a review of FANET routing protocols. In this study, AODV, DSDV, DSR, OLSR, AOMDV and HWMP routing protocols were explained in detail. Then the routing protocols were tested on three performance parameters: Packet

Delivery Ratio, End-to-End Delay and Throughput. NS-2 was used as simulation tool, and the simulation was done as a sample topology with different node velocity cases. In the simulation, IEEE 802.11 standard was used as MAC layer protocol. Since the HWMP is located on layer 2 and it is a hybrid routing protocol (modified version of AODV), it performs better results for PDR and Throughput parameters .DSDV and OLSR are purely proactive protocols, so end to end delay performances of these protocols are better than other protocols. This study compares the known routing protocols in the literature but does not comprise any new proposal for FANET routing protocols.

Hussen et al., in [17], analyzed the performance of different Mobile Ad-hoc Network (MANET) routing protocols (AODV, DSR, GRP and OLSR) for the communication of UAVs. They used Riverbed (OPNET) Modeler as simulation tool, simulations were based on a sample topology (50 mobile nodes and one fixed node) with various data rates supported by IEEE 802.11p (WAVE) standard. Since the OLSR is proactive routing protocol and number of mobile nodes is bigger than 50; simulation results show that the OLSR has highest routing overhead while maintaining the routing table periodically. On the other hand, the OLSR shows the least delay and the highest throughput performance as compared to other protocols. Both Ad-Hoc On Demand Distance Vector (AODV) and the Dynamic Source Routing (DSR) are reactive routing protocols and DSR stores the data as full path for a period of time while AODV stores the data for the next hop only. So, simulation results show that end-to-end delay of DSR is bigger than the end-to-end delay of AODV as expected. Simulation results also show that the DSR has less overhead than AODV, because instead of maintaining a route table for tracking routing information, DSR utilizes a route memory. The route memory allows multiple route entries to be maintained per destination, thereby enabling multipath routing. When one route to a destination breaks, the source can utilize alternate routes from the route memory, if they are available, to prevent another route discovery and able to react quickly to changes in the network. By using the concept of position based routing, the Geographic Routing Protocols (GRP) do not need to set up and to maintain the connections. Nodes do not require to store the routing tables and not maintain the routing tables up to date for transmitting the information. In GRP, different positioning schemes can be used such as GPS, GPRS etc. As expected, results show that GRP has better throughput performance than AODV and DSR, worse overhead performance than AODV and DSR.

In [18], MANET and FANET definitions are given and FANET's application areas are mentioned. Then basic differences between FANET and MANET are also given in this study with respect to mobility, energy consumption and localization. Then, 3D POSITION-BASED ROUTING algorithms are explained and their performance is compared by using NS-2 simulation tool. In the simulation, IEEE 802.11g standard was used as MAC layer protocol. In this study, nodes are considered as static and the effect of mobility on the routing performance is not investigated.

Li et al., in [19], improved the performance of the GPSR (Greedy Perimeter Stateless Routing) algorithm by using "position prediction" and "deciding the beacon signal interval" methods. In this study, initially, the next position information is predicted by using the last 10 position information of the nodes for which the beacon signal is received. The performance of position predicted GPSR and original GPSR was compared by using NS-3 simulation tool. 802.11g standard was used as MAC layer protocol in the simulation. Simulation results show that position predicted GPSR protocol can increase Packet Delivered Ratio (PDR) obviously in moderate beacon interval. Then, repetition interval of the beacon signal, which is used for position estimation, was decided adaptively. The performance of the original GPSR and complete adaptive beacon scheme used GPSR was compared and the experimental results show that the proposed method can achieve relatively accurate position information and improvements in PDR with greatly reduced beacon overhead.

Zhang et al., in [20], proposed a new mobility model named as Particle Swarm Mobility Model (PSSM) for UFANETs. The mobility models for aerial vehicles can be classified into two categories: traditional MANET models adapted to aerial ad hoc networks and new models developed for aerial ad hoc networks. Since the movement of UAV nodes obeys some kinematic and dynamic constraints, the pure random mobility models are not suited to UFANETs. The mobility model should take group motion into account for swarm UAV missions. During the flying session, the multiple UAVs should maintain group form. The proposed mobility model was compared with other mobility models (RWP, RPGM and Manhattan) by using open source Bonn Motion mobility tool. The simulation results show that the proposed mobility model (PSMM) can achieve good performance in terms of group form behavior (temporal correlation, spatial correlation) and path availability metrics.

Performances of AODV, OLSR and DSDV were compared in [21]. In this study, simulations were done by different simulation schemes (various number of nodes, various velocity of nodes) and NS-3 was used as simulation tool. In the simulation, IEEE 802.11n standard was used as MAC layer protocol. The main goal of the simulation was to simulate the repetitive sending of images from the nodes to the sink. In the paper, the sink node was not described clearly but it was assumed that the fixed station at the center of simulation area was the sink node. The metrics used for comparison are throughput, delivery rate, average delay, average jitter and mean hops. Since the routing overhead is not a comparison metric, simulation results show that proactive protocols OLSR and DSDV have better performance than AODV. This study also compares the known routing protocols in the literature but does not comprise any new proposal for FANET applications.

The studies mentioned above are the most recent routing protocol researches for UFANETs and all of them do not use link awareness information for routing decision improvement. The routing decision for the most of above mentioned protocols are done with respect to hop count between source and destination nodes. Because of the dynamic property of UFANETs, metrics based on hop count may lead to some misjudgments, since the link may be easily broken or may be seriously interfered, which will worsen the network performance. So, it is obvious that the hop count metric is less efficient in UFANETs. Instead of using hop count, it is possible to use cross layer, especially physical layer metrics, given in *Table 2.1*, for route decision.

Table 2.1 *Usable Cross Layer Metrics for Routing Decision*

Usable Cross Layer Metrics for routing decision		
Physical Layer	**Data Link Layer (MAC Layer)**	**Network Layer**
Bit Error Rate Channel Rate (physical throughput) Received power Reference Signal Received Power (RSRP) Signal to Noise Ratio (SNR) Frequency offset (Doppler Effect)	MAC frame error rate Queuing information Retransmission count Inter-arrival time Packet train size Service time	Hop count

2.3. Cross Layer Metrics usage for MANETs

In the literature there are many researches about cross layer metrics usage for routing decision, and most of these researches are interested in MANETs and VANETs.

Elshaikh et al., in [22], compared the reactive protocols AODV, DSR and DYMO in the first part of their study. In the second part of the study, they proposed a cross layer approach for Dynamic Sequence Distance Vector (DSDV) routing protocol for MANETs. They used a SNR-based routing metric instead of traditional hop count metric. Then, comparison of the proposed and traditional DSDV was done in terms of throughput, packet delivery ratio (PDR) and end-to-end delay. OMNET++ was used as simulation tool, but simulations were done with ten fix nodes. In the simulation, IEEE 802.11 standard was used as MAC layer protocol. Results show that, the SNR-based metric gets higher throughput and packet delivery ratio than the traditional hop count. Also the new metric achieves a smaller end-to-end delay than the traditional hop count metric. Since the simulations were done without mobility, this study does not converge to real scenarios. In the study, also, there is no information about how they used the SNR-based metric in routing decision.

Alnajjar et al., in [23], proposed a mechanism that allows the network layer to adjust its routing protocol dynamically based on SNR and Received Power (RP) along the end-to-end routing path for each source to destination link. Since the DSR protocol is a common multipath reactive routing protocol for MANETs, they applied this mechanism to DSR protocol. The proposed mechanism adds new fields to Route Reply (RREP) message of DSR protocol; these new fields are used for SNR and RP information of the communication link. Each intermediate node updates these fields. If the SNR and/or RP values obtained by intermediate node are worse than the values in the RREP message, then intermediate nodes changes these values with worse ones and forwards RREP message. The proposed method was implemented in OPNET simulation tool, and simulation results show that the proposed method achieved better performance than traditional DSR protocol in terms of delivery rate, delay and throughput. The proposed method presents better performance for multipath routing protocols, but this method does not introduce any improvement for in case of single path between source and destination. Additionally, since this method adds extra fields to RREP message format, overall routing overhead possibly increases. But simulation results do not have any information about routing overhead performance of proposed method.

Ramachandran et al., [24], used received power as a metric in cross layer design r for energy conservation, unidirectional link rejection and reliable route formation in MANETs. The proposed method, CLAODV, forwards the route request packet (RREQ) to destination node if the received power of RREQ message is bigger than some predefined threshold value. If the received power of RREQ message is not bigger than threshold value but it is bigger than the valid previous received power from the source of the RREQ message, than it forwards the RREQ message to next node. They compared the proposed method with AODV and Node Transition Probability (NTP) based routing protocols. GloMoSim was used as simulation tool. In the simulation, nodes used the distributed co-ordination function (DCF) of IEEE 802.11 WLAN standard with RTS/CTS extension and provide link layer failure notification to AODV routing protocol. There were different number of nodes and their velocities were between 0 – 25 m/s. Simulation results show that the proposed method improves the energy conservation, rejects unidirectional links, and reduces the routing overhead. But decision of the threshold value was not explained in this study. Simulations were done for a maximum 25 m/s velocity, on the other hand most of UFANETs velocities are bigger than this value. It is obvious that this study does not involve UFANETs scenarios.

Nityananda et al., in [25], proposed a Route Stability based QoS Routing (RSQR) protocol in MANETs which is an extension of cross layer routing with throughput and delay constraints. In order to guarantee the suitable data path for longer duration in MANET, they proposed a model for measuring the link stability and route stability depending on received signal power. They have proposed a route stability model which considers node mobility and signal power for computing the probability of link failure rather than using probability distribution of link lifetimes. In the proposed method, when a node receives RREQ packet it measures the received signal power and also records that in Neighbor Information Table (NIT). It also compares current received signal power with previous received signal power. The protocol compares received signal power with two thresholds Thr1 and Thr2. If the received signal power is greater than Thr1 then the link is considered as “stable”. If the received signal power is less than Thr2 then the link is considered as less stable. Using current and previous received signal power and two Thresholds Thr1 and Thr2 link stability is determined. Some additional fields in route request/ reply packets is taken into consideration so that the route stability information can be used to choose a route with increased stability when compared to all possible routes among existing source destination pair. The use of this route stability model in the proposed method significantly reduces the number of route recoveries required during data transmission. In this study, they implemented the proposed protocol and compared it with Ad hoc QoS on-demand routing with a set of simulations. Simulations were done with NS-2 simulation tool. Simulation results show that the proposed protocol achieves performance improvements in terms of control overhead, average end-to-end delay and packet delivery ratio especially in highly mobile scenarios. On the other hand, since the proposed protocol adds additional fields to RREQ and RREP packets, the routing overhead increases in case of enough received signal powers. Since the reference signal received power (RSRP) uses digital signal power instead of whole analogue bandwidth, it is less affected from interference signal than the analogue received power. Instead of using analog received power, received power in digital domain, like RSRP in LTE networks, might be used as routing metric.

Boumetjout et al., in [26], introduced a new routing protocol AodvPw which uses the received signal power information to enhance the stability of ad hoc network. In the proposed method, the received signal power information is used to compute path loss of the wireless channel. This information is added to RREP message field. This method is not much different than the previous methods which use received power.

Gu et al., in [27], proposed Minimum Interference Routing (MIR) method. This method chooses the links that have long connectivity duration, and then it builds the least interfered route based on a new routing metric which takes interference and link connectivity duration together into account. They used NS-2 as simulation tool, and compared the performance MIR in terms of node density with the traditional AODV routing protocol by using end-to-end delay, packet loss ratio, throughput and routing overhead metrics. In the simulation, maximum velocity of nodes was 20m/s and RWP mobility model was used. Simulation results show that MIR can improve the network performance. In this study, it is assumed that relative velocity between two nodes is known by each node. And this information is used for calculation of link connectivity duration. In a real scenario, in order to use this method, the velocity (magnitude and direction information) of each node shall be shared with neighbor nodes. This sharing will cause routing overhead, this study does not give any information regarding to this extra overhead. On the other hand, nodes velocities and used mobility model are not exactly suitable for UFANETs.

Shehri et al., [28], proposed a multipath routing protocol using both Received Power (RP) and SNR metrics for tactical ground MANETs. This protocol decides threshold values both RP and SNR. According to the proposed method, reserved or unused 11-bit field is used and one extra byte is added for reporting link quality (LQ). The LQ field contains information on SNR and RP threshold values. The multiple RREP packets are sent back to source node over the reverse path. Unlike the case of AODV, every RREP is considered by the source in order to discover multiple paths in route discovery. The route with the highest quality is then selected as the primary path. If the primary path fails, the second discovered path is activated and so on. NS-2 was used as simulation tool to compare the proposed method with AODV, H-AODV and AOMDV. The performance comparisons were done by using throughput, PDR and end-to-end delay performance metrics. Simulation results show that the proposed protocol outperforms the other routing protocols. Adding extra byte to RREQ message causes extra overhead for this method, but this study does not investigate the overhead performance of proposed method. In real scenarios, threshold values will be decided with some margins, so it is possible that communication will be available although when RP and SNR values are lower than thresholds. If this condition is occurred in a real scenario, the proposed method adds extra delay when RP and SNR values are lower than thresholds. In a real tactical MANET scenario, latency metric is critical for time-sensitive and mission critical applications. So this study does not investigate the latency estimation of real scenario and overhead performance metric of proposed method. Since the scenarios used in this study can be used in real ground tactical MANET and maximum speed of nodes is 20 m/s, this study does not give enough information for UFANETs network.

In [37], a weight based clustering scheme is proposed by using trust, density, mobility and energy metrics. This study provides a cluster based hierarchical network architecture. Mobility metric is calculated for each neighbor node by using doppler shift and received power values; and weighted and used for cluster head election. In the study, there is no detailed information about how doppler shift and received power parameters are obtained. As a result, the application challenges of doppler shift and received power usage are not detailed in this work. Also, simulations are done with limited network topologies and simulations results are compared with other cluster based schemes in terms of average number of CHs, average number of CH changes, total number of re-affiliations, clusters stability, and total overhead.

The first usage of doppler shift for routing decision is introduced in [31]. In this study, Sakhaee et al. proposed a new handoff mechanism for In Flight Internet Access. Each commercial airplane calculates the doppler shift values of received signals from each candidate satellite, then chooses one of the satellite which has a minimum doppler shift value. This satellite will be used for internet access until reaching a link cost threshold value. If the link cost value exceeds the threshold value, then a handoff procedure is processed for a new satellite connection. Simulation results shows that usage of doppler shift value decreases the total number of handoffs. In this study, there is not enough information for simulation cases, especially the velocity of the airplanes and frequency of wireless communication. Since, these values are used for doppler shift calculation, these parameters should be clarified. Generally the satellite communication is in Ku Band (12-18 GHz) and typical cruise speed of passenger aircraft is approximately between 820 – 900 km/h, it is possible to measure significant doppler shifts with traditional local oscillators. On the other hand, usage of doppler shift for wifi signals (2.4 GHz or 5 GHz) and commercial UAVs (0 – 300 km/h) should be explained and details doppler shift measurement should be given.

Then, in [32] and [33], Sakhaee et al. introduced QoS Multipath Doppler Routing algortihm for aeronautical ad hoc networks. In these studies, velocity of airplanes is assumed 840 km/h and wireless communication range is assumed between 200 – 600 km ranges. These velocity and wireless communication ranges are not applicable to UFANETs. On the other hand, since doppler shift values of nodes are added to routing messages, the overhead ratio is increased with high node densities. Simulation results show that proposed method has less total number of handoffs. But, other performance metrics (latency, overhead ratio) are not investigated in these studies. It is obvious that the latency and overhead ratio would be worse than the traditional ones. In [107], the pseudo linear (predefined) mobility is used for commercial aircrafts. Since, UAVs can be used for search and tracking missions, this approximation is not suitable for UFANETs.

In [34], Sakhaee et al. also added Link Expiration Time (LET) estimation to their algorithm. In this study, LET is calculated by using doppler shift and measured power of received signals. Simulation results show that the proposed method improves the total number of overhead messages. But usage of beacon messages are not mentioned in this study. On the other hand, latency and packet transmission ratio performance metrics are not investigated. Also, aircrafts moves in a predefined direction with linear fashion, which is not suitable for UFANETs applications. Since, the proposed method is compared with reactive routing protocol DSR, beacon message (HELLO) usage is not explained.

In [35], link expiration time (LET) is predicted by using received power and GPS (velocity, heading, position) information. Then this LET value is used for route selection for source and destination nodes. The proposed method, named as Mobility-adaptive Routing (MAR), is simulated by using different number of nodes (40 to 80 nodes) and mobility cases (RWP mobility with 1 m/s to 50 m/s speeds) and compared with traditional DSR protocol. Simulation results show that the proposed method improves the packet delivery ratio, overhead ratio and latency. Since the analog received power is used for route selection, the proposed algorithm does not give enough performance at the existence of interference signal. Also, it uses GPS information for route selection, it is known that accuracy of GPS devices depends on the total number of used satellites for signal processing. The extreme atmospheric conditions and unavailability of GPS satellites cause problems and position information can be inaccurate. Also it is possible that GPS signals can be jammed, in this case, GPS devices do not give reliable information. As a result, performance of the proposed method can be degraded with these conditions.

In [36], Zhou et al. propose a method which uses Doppler Shift and Traffic Load metrics for aeronautical ad hoc networks routing. This method is named as nodes mobility and traffic load aware routing (NTAR). Performance NTAR is evaluated by simulations and compared with the basic AODV routing algorithm. During the simulations, RWP mobility model is used with different speeds (100 m/s to 800 m/s). Simulation results show that NTAR improves the packet delivery rate, but latency of NTAR is worse than the basic AODV. In NTAR, uses Doppler Shift and Traffic Load metrics are used for only route establishment, route maintenance is done as in the basic routing algorithm (AODV). As a result, the maximum improvement rate of pack delivery rate is only %3. Since the performance of NTAR is not investigated for speeds less than 100 m/s, applicability of this method for UFANETs is unknown. On the other hand one of the basic performance metric, overhead ratio, is not investigated for NTAR.

As we mentioned in Chapter-1, UFANET is a special case of mobile ad hoc networks (MANETs and VANETs) that are characterized by a high degree of mobility and frequent topology changes. The above mentioned researches using cross layer metrics routing protocol do not give enough information about cross layer routing for UFANETs.

CHAPTER 3

AODV ROUTING ALGORITHM FOR UFANETS

The Ad hoc On-Demand Distance Vector (AODV) routing protocol is chosen as basic routing protocol for this study. This protocol is one of the most popular and efficient routing protocol which uses dynamic and multi-hop routing between mobile nodes. The AODV routing protocol defines neighbor discovery, route discovery, route establishment and error detection processes to establish and maintain a mobile ad hoc network. AODV protocol is a flat routing protocol and does not need any central infrastructure system to handle the routing process. AODV routing protocol establishes a loop-free and self-starting MANET which can be scaled to a large number of mobile nodes.

In 2018, the AODV routing protocol is awarded as SIGMOBILE Test-of-Time Paper Award. The committee of the award notes that this protocol is the most influential ad hoc routing protocol to date by using these words: "This algorithm proposes a novel and suitable solution for the operation of these dynamic and unstable networks. Its major impact on the industry and related standards demonstrate the practical importance of this work. Additionally, the protocol is a "must-teach" in academic curricula related to mobile networking." [https://www.sigmobile.org/grav/awards/test-of-time-paper]. (SIGMOBILE is non-profit organization and international association of scientists, researchers, educators, industry scientists and developers. This organization recognizes accomplishments in the field of mobile computing and communications through awards. The SIGMOBILE Test-of-Time awards recognize papers that have had a sustained and significant impact in the SIGMOBILE community over at least a decade. The award recognizes that a paper's influence is often not fully apparent at the time of publication, and it can be best judged with the perspective of time)

The Ad hoc On-Demand Distance Vector (AODV) routing protocol uses dynamic, reactive (self-starting), multi-hop routing between mobile nodes trying to establish and maintain an ad hoc network. By using this protocol, mobile nodes can establish routes for new destinations. Since AODV is a reactive protocol, it does not maintain routes to destinations that are not in active communication. AODV routing protocol overcomes the Bellman-Ford "counting to infinity" problem and establishes a loop-free network. In this protocol, it is possible to use a beacon message (Hello Message) for maintaining the established ad hoc network. When network topology changes and link breaks occur, all possibly affected nodes are notified, so invalid routes are not used for data transmission [8].

3.1. AODV Basics

AODV maintains a routing table with the next hop for target destinations. Routes time out after a while if not used (i.e. no packets are sent on them). Route table of AODV contains following fields for each entry:

- Destination IP Address
- Destination Sequence Number
- Valid Destination Sequence Number flag
- Other state and routing flags (e.g., valid, invalid, repairable, being repaired)
- Network Interface
- Hop Count (number of hops needed to reach destination)
- Next Hop
- List of Precursors
- Lifetime (expiration or deletion time of the route)

Unlike other protocols, AODV only maintains information about the next hop in the route for destination, not the entire routing list. This saves memory and decreases computational overhead for route maintenance.

AODV uses the following routing message types:

- Route Request (RREQ) is used for route establishment.
- Route Reply (RREP) is used for route establishment.
- Route Error (RERR) is used for route maintenance.
- Beacon Message (HELLO) is used for route maintenance.

3.1.1. RREQ Message

When a node has some data to transmit to destination node, it checks its route table. If there is an active route to this destination then it transmits data directly. If there is not an active route for the destination then it broadcasts RREQ message. AODV uses sequence number property for avoiding problems (such as "counting to infinity") associated with classical distance vector protocols. Every time a node sends a new message, it uses a new sequence number which increases monotonically. A valid destination route must have a sequence number at least as great as that contained in the RREQ message. RREQ message format is given in the Figure 3.1 and definition of each field in RREQ message is given in the *Table 3.1*. [8]

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |J|R|G|D|U|   Reserved          |   Hop Count   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            RREQ ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination IP Address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Destination Sequence Number                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Originator IP Address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Originator Sequence Number                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3.1 RREQ Message format

Table 3.1 *RREQ Message Fields*

Field Name	**Definition**
Type	1
J	Join flag; reserved for multicast
R	Repair flag; reserved for multicast
G	Gratuitous RREP flag; indicates whether a gratuitous RREP should be unicast to the node specified in the Destination IP Address field.
D	Destination only flag; indicates only the destination may respond to this RREQ
U	Unknown sequence number; indicates the destination sequence number is unknown
Reserved	Sent as 0; ignored on reception
Hop Count	The number of hops from the Originator IP Address to the node handling the request.
RREQ ID	A sequence number uniquely identifying the particular RREQ when taken in conjunction with the originating node's IP address. This RREQ ID is incremented each time source node sends a RREQ message
Destination IP Address	The IP address of the destination for which a route is desired.
Destination Sequence Number	The latest sequence number received in the past by the originator for any route towards the destination. If the source node doesn't know it, this field is zero.
Originator IP Address	The IP address of the node which originated the Route Request.
Originator Sequence Number	The current sequence number to be used in the route entry pointing towards the originator of the route request.

AODV uses <RREQ ID, Originator IP Address> pair for handling unnecessary RREQ flooding. This pair is a unique identifier for the RREQ. Each node stores these pairs for all the recent RREQs it has received. If an intermediate node receives RREQ from another node, it first checks whether it has received this pair information before. If it has received this RREQ message, then it discards this RREQ message. Otherwise, it broadcasts the RREQ message and sets up a reverse path entry in its route table for the source (originator) node. As a RREQ propagates through the network, intermediate nodes use it to update their routing tables (in the direction of the source node: reverse path)

3.1.2. RREP Message

When a RREQ reaches a destination node or an intermediate node which has an active route to the destination node, this node sends RREP message back to the source node. While RREQ message is a broadcast message, RREP message is a unicast message. As the RREP propagates back to the source node, intermediate nodes update their routing tables (in the direction of the destination node: forward path). RREP message format is given in the Figure 3.2 and definition of each field in RREP message is given in the Table 3.2. [8]

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |R|A|    Reserved     |Prefix Sz|   Hop Count   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Destination IP address                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Destination Sequence Number                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Originator IP address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Lifetime                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3.2 RREP Message Format

Table 3.2. RREP Message Fields

Field Name	**Definition**
Type	2
R	Repair flag; used for multicast
A	The ’A’ bit is used when the link over which the RREP message is sent may be unreliable or unidirectional. When the RREP message contains the ’A’ bit set, the receiver of the RREP is expected to return a RREP-ACK message
Reserved	Sent as 0; ignored on reception
Prefix Size	If nonzero, the 5-bit Prefix Size specifies that the indicated next hop may be used for any nodes with the same routing prefix (as defined by the Prefix Size) as the requested destination.
Hop Count	The number of hops from the Originator IP Address to the Destination IP Address. For multicast route requests this indicates the number of hops to the multicast tree member sending the RREP.
Destination IP Address	The IP address of the destination for which a route is supplied.
Destination Sequence Number	The destination sequence number associated to the route.
Originator IP Address	The IP address of the node which originated the RREQ for which the route is supplied.
Lifetime	The time in milliseconds for which nodes receiving the RREP consider the route to be valid.

3.1.3. RRER Message

In active routes, all nodes follow the link status of next hope. When a node detects a link break in an active route, a RERR message is used to inform the related nodes. In order to enable this reporting mechanism, each node keeps a "precursor list", containing the IP address for each its neighbors that are likely to use it as a next hop towards each destination.

A node initiates processing for a RERR message in three cases:

(i) if it detects a link break for the next hop of an active route in its routing table, or

(ii) if it gets a data packet destined to a node for which it does not have an active route, or

(iii) if it receives a RERR from a neighbor for one or more active routes.

The node, decided to initiate RERR message processing, lists all unreachable destination nodes for each case. Then, the node checks its route table for each unreachable destination node. If there is at least one route entry for each unreachable destination node and precursor list of this entry is not empty, then the node decides to send a RERR message. If the precursor list of this entry contains only one neighbor, the RERR is a unicast message and sends to that neighbor. If there are many precursors, the RERR is a broadcast message. If broadcast is not suitable for network, then it is possible to unicast the RERR iteratively for all precursors. RRER message format is given in the Figure 3.3 and definition of each field in RRER message is given in the Table 3.3. [8]

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |N|          Reserved           |   DestCount   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Unreachable Destination IP Address (1)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Unreachable Destination Sequence Number (1)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
|  Additional Unreachable Destination IP Addresses (if needed)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Additional Unreachable Destination Sequence Numbers (if needed)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3.3 RERR Message Format

Table 3.3 RERR Message Field

Field Name	**Definition**
Type	3
N	No delete flag; set when a node has performed a local repair of a link, and upstream nodes should not delete the route.
Reserved	Sent as 0; ignored on reception
DestCount	The number of unreachable destinations included in the message; MUST be at least 1.
Unreachable Destination IP Address	The IP address of the destination that has become unreachable due to a link break.
Unreachable Destination Sequence Number	The sequence number in the route table entry for the destination listed in the previous Unreachable Destination IP Address field.

3.1.4. Beacon (HELLO) Message

In an AODV network, a node can send its connectivity information via broadcasting a beacon (HELLO) message. HELLO messages are used to determine local connectivity. Normally, if a node is not a part of any active route, it does not send this beacon message. After being a part of an active route, it sends this message periodically (every HELLO_INTERVAL in milliseconds). HELLO message format is same as RREP message with Time-to-Leave (TTL) = 1. The other message fields of HELLO message are given in the Table 3.4. [8]

Table 3.4 HELLO Message Fields

Field Name	Definition
Destination IP Address	The node's own IP address.
Destination Sequence Number	The node's latest sequence number.
Hop Count	0
Lifetime	ALLOWED_HELLO_LOSS * HELLO_INTERVAL in milliseconds

When a node receive a HELLO message, if exists, it updates related entry of its route table or adds a new entry by using Lifetime field of HELLO message. If a route already exists, then the Lifetime for the route should be increased, if necessary, to be at least ALLOWED_HELLO_LOSS * HELLO_INTERVAL. Routes that are created by hello messages and not used by any other active routes will have empty precursor lists and would not trigger a RERR message if the neighbor moves away and a neighbor timeout occurs. If a node does not receive any packets (HELLO messages or otherwise) from a neighbor for more than ALLOWED_HELLO_LOSS * HELLO_INTERVAL, the node will assume that the link to this neighbor is currently lost.

Route establishment of the AODV protocol [AODV_ 2003] is processed by using RREQ and RREP messages. After establishing routes, route maintenance is processed by using RERR and HELLO messages.

3.1.5. Route Establishment of AODV Routing Protocol

In the originator (source) node, when there is a new data packet waiting for transmission, algorithm checks whether there is an entry in the Route Table for destination node or not. If there is an entry, then the source node directly transmits the data packet to next hop. If there is not an entry, then it starts to establish the route for destination node. First it broadcasts the RREQ packet, and waits for RREP in NET_TRAVERSAL_TIME milliseconds. If the source node receives a RREP message within the NET_TRAVERSAL_TIME, then it transmits the data packet to next hop. If a route is not received within NET_TRAVERSAL_TIME milliseconds, the node tries again to discover a route by broadcasting another RREQ, up to a maximum of RREQ_RETRIES. In order to reduce congestion in the network, the algorithm utilizes a binary exponential backoff for each retry of RREQ message. For each additional attempt, the waiting time for the RREP is multiplied by 2, so that the time conforms to binary exponential backoff. If a route discovery has been attempted RREQ_RETRIES without receiving any RREP, all data packets for the corresponding destination node is dropped.

The originating node uses an expanding ring search technique for preventing unnecessary network-wide dissemination of RREQs. In this technique, the originating node initially uses a TTL =TTL_START in the RREQ packet IP header and sets the timeout for receiving a RREP to RING_TRAVERSAL_TIME milliseconds. RING_TRAVERSAL_TIME is calculated as described in section 3.2.1. The TTL_VALUE used in calculating RING_TRAVERSAL_TIME is set equal to the value of the TTL field in the IP header. If the RREQ times out without a corresponding RREP, the originator broadcasts the RREQ again with the TTL incremented by TTL_INCREMENT. This continues until the TTL set in the RREQ reaches TTL_THRESHOLD, beyond which a TTL = NET_DIAMETER is used for each attempt. Once TTL = NET_DIAMETER, the timeout for waiting for the RREP is set to NET_TRAVERSAL_TIME. When it is desired to have all retries traverse the entire ad hoc network, this can be achieved by configuring TTL_START and TTL_INCREMENT both to be the same value as NET_DIAMETER.

The flowchart of the above mentioned process is given in the Figure 3.4.

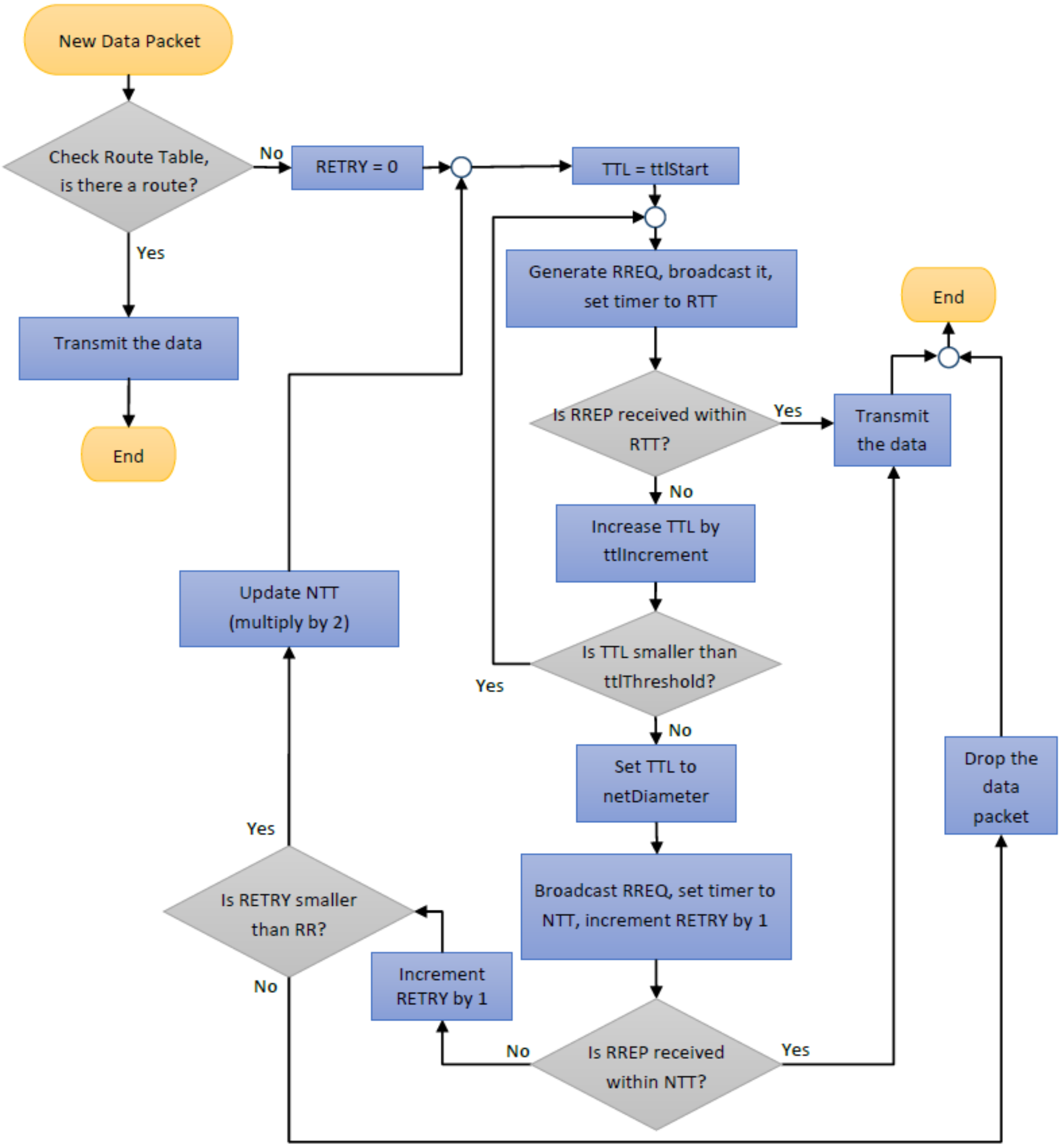

Figure 3.4 Flowchart of New Data Transmission at Source Node

When a node receives RREQ message in the network, it checks the <RREQ ID, Originator IP Address> pair of the received message. If this RREQ has been already processed, then it ignores this RREQ message. If this RREQ is a new message, this node sets or updates the reverse path. The lifetime of the reverse path is calculated by using size of network and other parameters. The current node can use the reverse route to forward data packets in the same way as for any other route in the routing table. After setting or updating the reverse path, the current node checks whether the destination node address in the RREQ message is its own address or not. If it is itself the destination node or it has a fresh active route to the destination, then it replies this RREQ by generating RREP message. If, the destination sequence number in the node’s existing route table entry for the destination is valid and greater than or equal to the Destination Sequence Number of the RREQ (comparison using signed 32-bit arithmetic), and the "destination only"(’D’) flag is NOT set, then the node decide that this active route is fresh and can be used for routing. If this node decides not to generate RREP for receiving RREQ message, it checks the TTL value in the IP header of RREQ message. If it is larger than 1, then the node updates (decrease TTL by one, increase Hop Count field by one) and broadcasts the RREQ message. The flowchart of this process is given in the Figure 3.5.

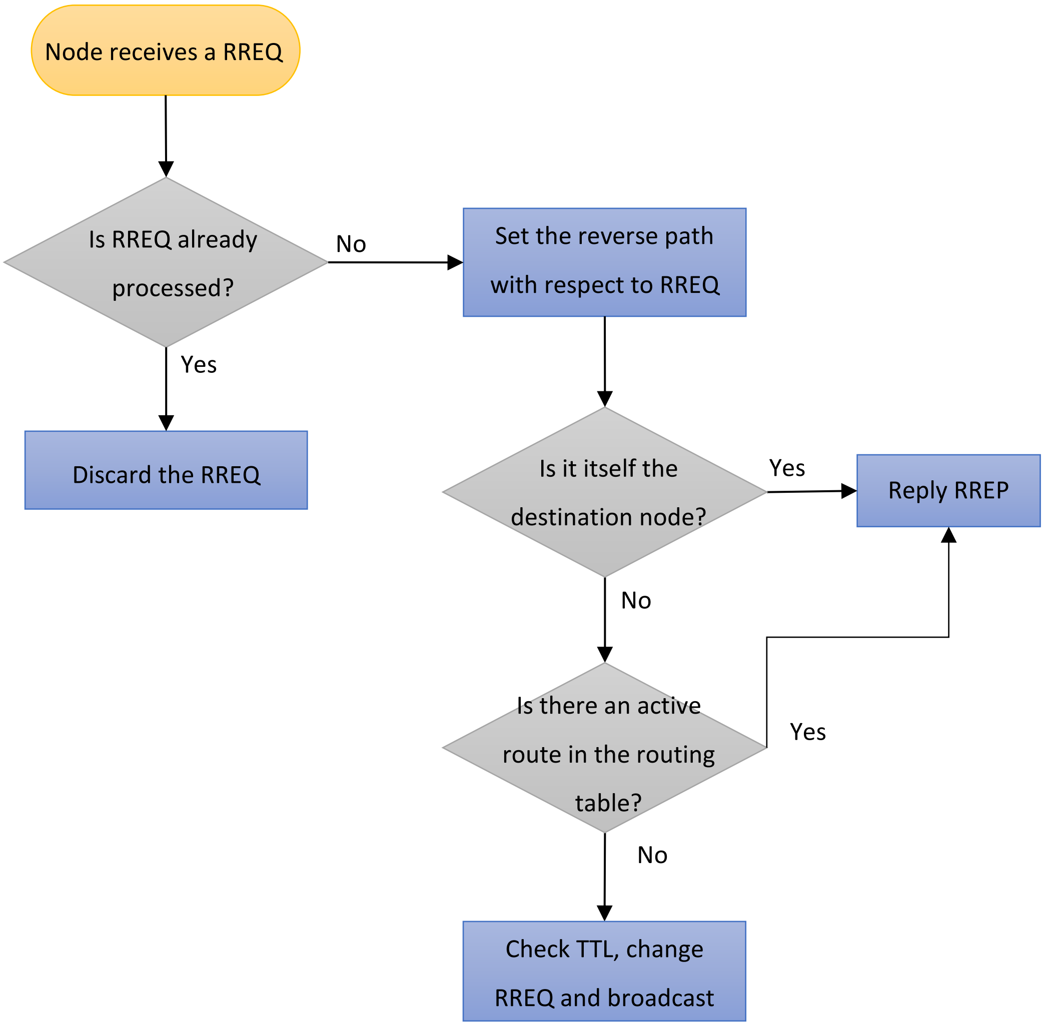

Figure 3.5 Flowchart of Receiving RREQ Message Process

If a node receives a RREP message, it sets or updates the forward path and checks whether the originator (source) node address in the RREP message is its own address or not. If it is itself the originator node, it starts to transmit the data packet. If not, it updates the next hop information in the RREP message and unicast the RREP by using reverse path. The flowchart of this process is given in the Figure 3.6.

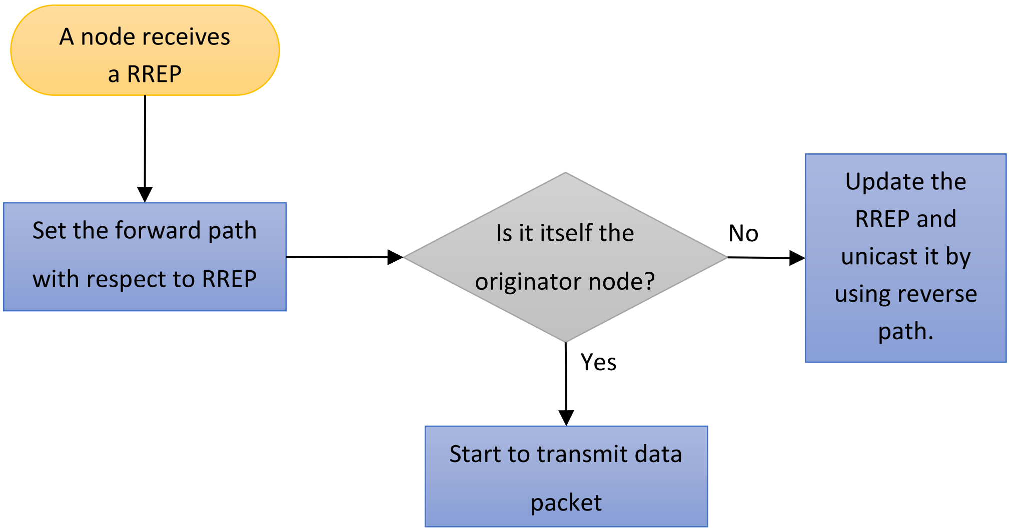

Figure 3.6 Flowchart of Receiving RREP Message Process

3.1.6. AODV Route Establishment Example

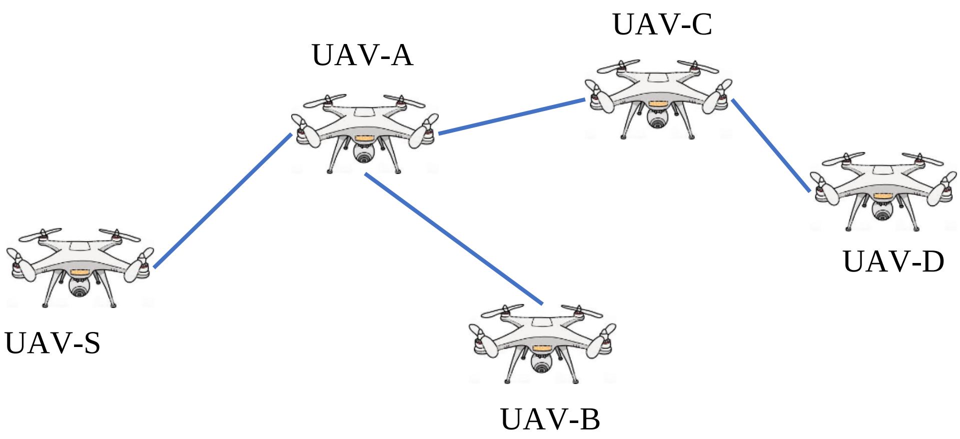

Figure 3.7 AODV Route Establishment Example

There is an example of UAV network topology in the *Figure 3.7*. In this topology, blue lines show that there is a wireless data link between UAV nodes and all UAV nodes use AODV routing protocol for route establishment. When the source node UAV-S needs to send data packets to the destination node UAV-D, it checks its routing table and sees that there is no entry for routing information. Then UAV-S starts to establish a route to UAV-D:

- UAV-S generates RREQ packet and broadcast it to its neighbors (UAV-A)

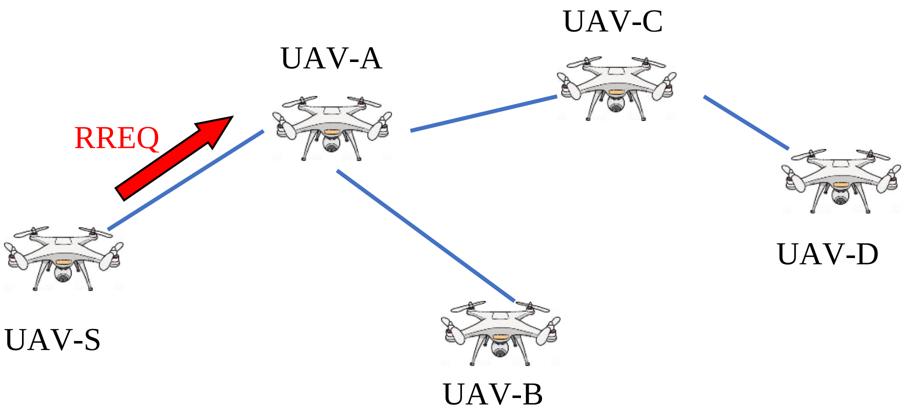

Figure 3.8 Generation and Transmission of RREQ by UAV-S

- UAV-A receives the RREQ and makes a reverse route entry for UAV-S (Destination node = UAV-S, Next hop = UAV-S, hop count = 1)
- UAV-A checks the destination node field of received RREQ message and sees that the destination node is UAV-D.
- UAV-A checks its routing table and sees that there is no active route for UAV-D, then it decreases the TTL field of RREQ by one and rebroadcasts RREQ message to its neighbors (UAV-S, UAV-B, UAV-C).

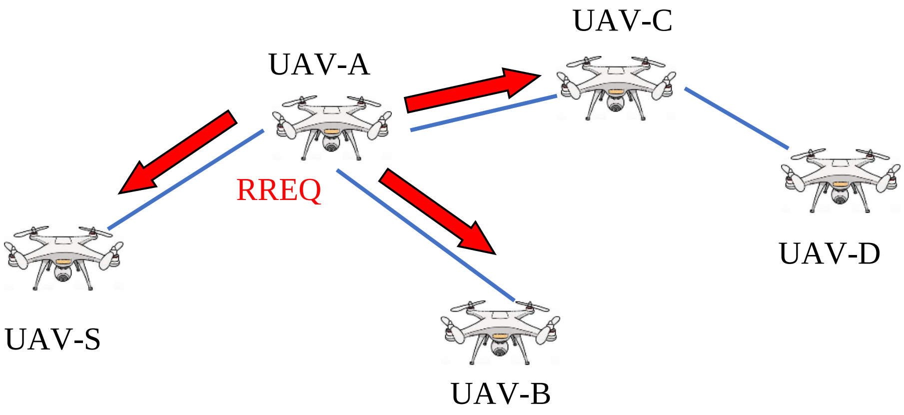

Figure 3.9 Generation and Transmission of RREQ by UAV-A

- UAV-S receives the RREQ packet and sees that this RREQ has already been processed. Then UAV-S ignores this RREQ and adds this new route (Destination node = UAV-A, Next hop = UAV-A, hop count = 1) to its routing table.
- UAV-B receives the RREQ and makes a reverse route entry for UAV-S (Destination node = UAV-S, Next hop = UAV-A, hop count = 2)
- UAV-B checks the destination node field of received RREQ message and sees that the destination node is UAV-D.
- UAV-B checks its routing table and sees that there is no active route for UAV-D, then it decreases the TTL field of RREQ by one and rebroadcasts RREQ message to its neighbors (UAV-A).

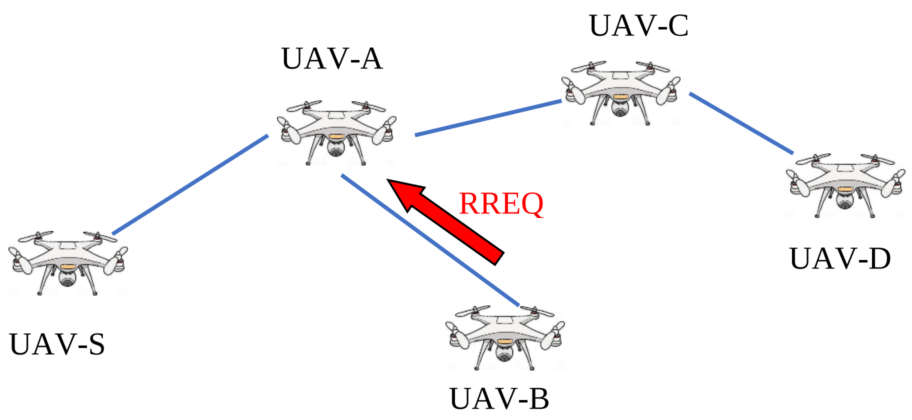

Figure 3.10 Generation and Transmission of RREQ by UAV-B

- UAV-A receives the RREQ packet and sees that this RREQ has already been processed. Then UAV-A ignores this RREQ and adds this new route (Destination node = UAV-B, Next hop = UAV-B, hop count = 1) to its routing table.
- UAV-C receives the RREQ packet from UAV-A (*Figure 3.9*) and makes a reverse entry for UAV-S (Destination node = UAV-S, Next hop = UAV-A, hop count = 2)
- UAV-C checks the destination node field of received RREQ message and sees that the destination node is UAV-D.
- UAV-C checks its routing table and sees that there is no active route for UAV-D, then it decreases the TTL field of RREQ by one and rebroadcasts RREQ message to its neighbors (UAV-A, UAV-D)

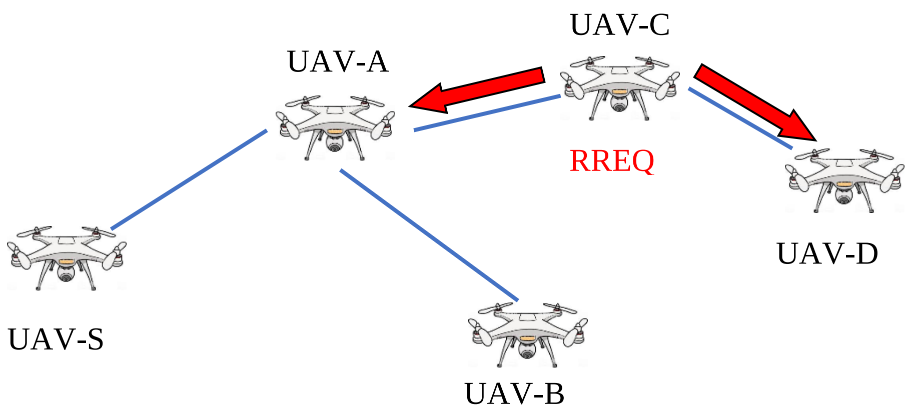

Figure 3.11 Generation and Transmission of RREQ by UAV-C

- UAV-A receives the RREQ packet and sees that this RREQ has already been processed. Then UAV-A ignores this RREQ and adds this new route (Destination node = UAV-C, Next hop = UAV-C, hop count = 1) to its routing table.
- UAV-D receives the RREQ packet and makes a reverse entry for UAV-S (Destination node = UAV-S, Next hop = UAV-C, hop count = 3)
- UAV-D checks the destination node field of received RREQ message and sees that the destination node is itself and the originator node is UAV-S. Then UAV-D generates RREP message and unicasts it to UAV-C by using reverse entry. If it does not know the physical (MAC) address of the UAV-C, it first sends an ARP packet to resolve the MAC address of UAV-C. After resolving the MAC address of UAV-C, then it sends RREP packet.

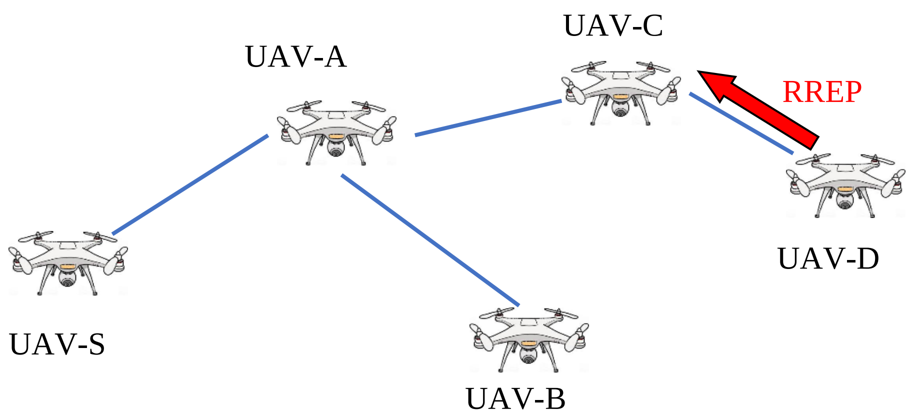

Figure 3.12 Generation and Transmission of RREP by UAV-D

- UAV-C receives RREP message from UAV-D and makes a forward route entry for UAV-D (Destination node = UAV-D, Next hop = UAV-D, hop count = 1).
- UAV-C checks the originator node field of received RREP message and sees that the originator node is UAV-S. Then it updates the RREP and unicasts the RREP message to the UAV-A by using reverse path entry. If it does not know the physical (MAC) address of the UAV-A, it first sends an ARP packet to resolve the MAC address of UAV-A. After resolving the MAC address of UAV-C, then it sends RREP packet.

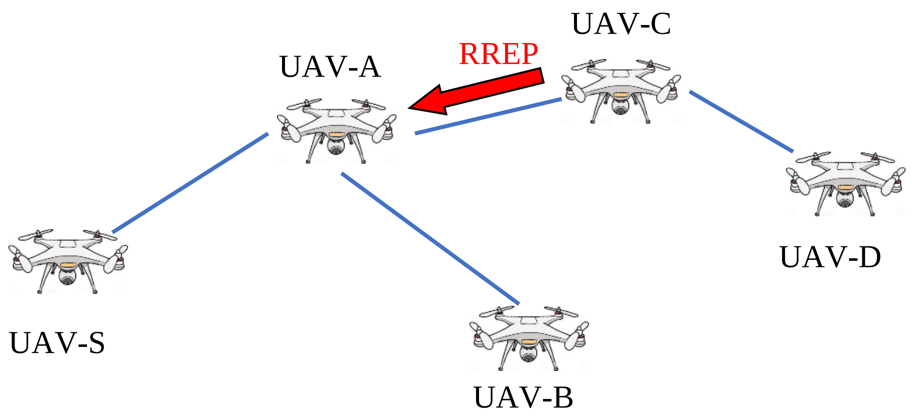

Figure 3.13 Generation and Transmission of RREP by UAV-C

- UAV-A receives RREP message from UAV-C and makes a forward route entry for UAV-D (Destination node = UAV-D, Next hop = UAV-C, hop count = 2).
- UAV-A checks the originator node field of received RREP message and sees that the originator node is UAV-S. Then it updates the RREP and unicasts the RREP message to the UAV-S by using reverse path entry. If it does not know the physical (MAC) address of the UAV-S, it first sends an ARP packet to resolve the MAC address of UAV-S. After resolving the MAC address of UAV-S, then it sends RREP packet.

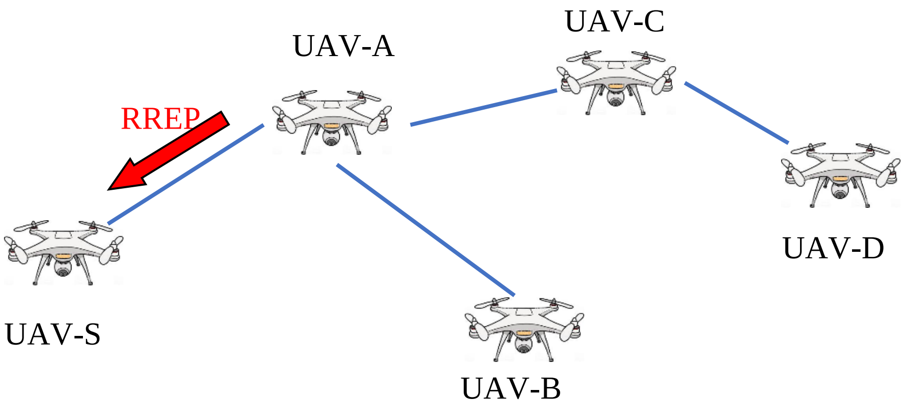

Figure 3.14 Generation and Transmission of RREP by UAV-C

- UAV-S receives RREP message from UAV-A and makes a forward route entry for UAV-D (Destination node = UAV-D, Next hop = UAV-A, hop count = 3).
- UAV-S checks the originator node field of received RREP message and sees that the originator node is itself. Then it sends the data packet to UAV-A by using forward route entry.
- UAV-A receives the data packet from UAV-S and checks the destination field of it. It sees that the destination node is UAV-D, then it forwards the data packet to UAV-C by using forward route entry.

- UAV-C receives the data packet from UAV-A and checks the destination field of it. It sees that the destination node is UAV-D, then it forwards the data packet to UAV-D by using forward route entry.
- UAV-D receives the data packet from UAV-C and checks the destination field of it. It sees that the destination node is itself and processes the data packet.

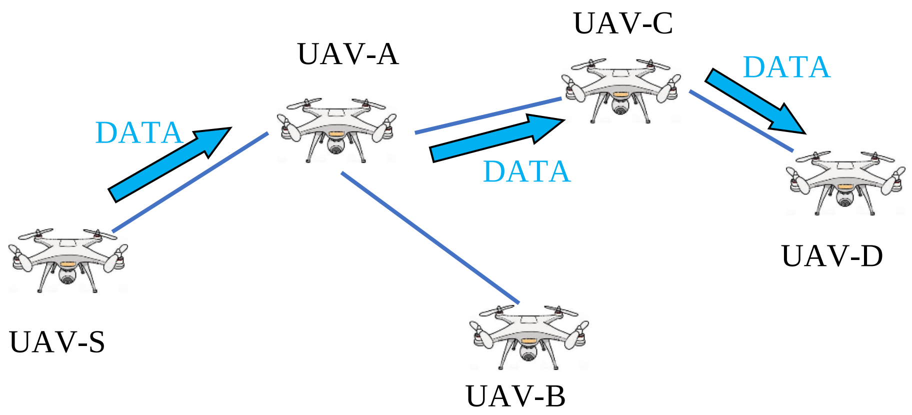

Figure 3.15 Transmission of Data from Source to Destination

3.2. Simulation of AODV Routing Protocol with Omnet++

Network simulations can be done with different tools. One of them is Omnet++.

The OMNET++ is an Integrated Development Environment (IDE) based on Eclipse platform. OMNeT++ adds functionality for generating and configuring models (NED and INI files), performing batch executions and analyzing the simulation results; while Eclipse provides C++ editing, SVN/GIT integration and other optional features (UML modeling, bug-tracker integration, database access, etc.) by using various open-source and commercial plug-ins.

INET Framework is an open-source library for the OMNeT++ simulation environment. It provides protocols, agents and other models for researchers and students working with communication networks. INET is especially useful when designing and validating new protocols, or exploring new or exotic scenarios. INET supports a wide class of communication networks, including wired, wireless, mobile, ad hoc and sensor networks. It contains models for the Internet stack (TCP, UDP, IPv4, IPv6, OSPF, BGP, etc.), link layer protocols (Ethernet, PPP, IEEE 802.11, various sensor MAC protocols, etc), refined support for the wireless physical layer, MANET routing protocols, DiffServ, MPLS with LDP and RSVP-TE signaling, several application models, and many other protocols and components. It also provides support for node mobility, advanced visualization, network emulation and more.

Several other simulation frameworks take INET as a base, and extend it into specific directions, such as vehicular networks, overlay/peer-to-peer networks, or LTE

INET benefits from the infrastructure provided by OMNeT++. Beyond making use of the services provided by the OMNeT++ simulation kernel and library (component model, parameterization, result recording, etc.), this also means that models may be developed, assembled, parameterized, run, and their results evaluted from the comfort of the OMNeT++ Simulation IDE, or from the command line.

3.2.1. Used OMNET++ Parameters and application details

For AODV routing protocol [8] in INET framework, there are some parameters for initial settings. Definitions of these parameters are given below.

askGratuitousRREP: This parameter is used for Gratuitous flag “G” set in RREQ message. If the RREQ has the “G” flag set, and the intermediate node returns a RREP to the originating node, it must also unicast a gratuitous RREP to the destination node. In INET framework, default value of this parameter is false (0).This value has not been changed during the simulations.

```
Bool askGratuitousRREP = default(false); // see RFC 3561: 6.6.3
```

useHelloMessages: This parameter is used for Hello message usage for AODV routing protocol. If the value of this parameter is false (0), then the simulation does not use Hello message property of AODV. If it is set to true (1), then nodes offer connectivity information by broadcasting local Hello messages. In INET framework, default value of this parameter is false (0). This value has been changed during the simulations.

```
Bool useHelloMessages = default(false); // see RFC 3561: 6.9
```

useLocalRepair: This parameter is used for Local Repair usage for AODV routing protocol. When a link break in an active route occurs, the node upstream of that break may choose to repair the link locally if the destination was no farther than MAX_REPAIR_TTL hops away. If the value of this parameter is false (0), then the simulation does not use Local Repair property of AODV. If it is set to true (1), then nodes offer connectivity information by broadcasting local Hello messages. In INET framework, default value of this parameter is false (0). This value has not been changed during the simulations.

```
bool useLocalRepair = default(false); // see RFC 3561: 6.12
```

destinationOnlyFlag: This parameter is used for destination only flag "D" set in RREQ message. Normally, when an intermediate nodes receives a RREQ message, it checks whether there is a fresh active route in the routing table for destination node or not. If the value of this parameter is true (1), then only the destination node is allowed to respond the specified RREQ, any intermediate node cannot respond the RREQ message although it has a fresh active route for destination node. In INET framework, default value of this parameter is false (0). This value has not been changed during the simulations.

```
bool destinationOnlyFlag = default(false); // see RFC 3561: 5.1
```

helloInterval: This parameter is used for determining the period of Hello message. If **useHelloMessages** parameter is set to true (1), then the node broadcasts Hello message every **helloInterval** seconds if it is necessary. In INET framework, default value of this parameter is 1 second. This value has been changed during the simulations.

```
double helloInterval@unit(s) = default(1s); // every helloInterval
seconds a node broadcasts Hello messages
```

allowedHelloLoss: This parameter is used for specifying the value of the lifetime field of Hello message. The lifetime of Hello message is calculated by multiplying this parameter with **helloInterval**. In INET framework, default value of this parameter is 2. This value has been changed during the simulations.

```
int allowedHelloLoss = default(2); // allowedHelloLoss * helloInterval is the lifetime value for Hello messages
```

activeRouteTimeout: This parameter defines how long a route is kept in the routing table after the last transmission of a packet on this route. In INET framework, default value of this parameter is 3 seconds. This value has been changed during the simulations. (Note: In Aodv.cc file, createRoute and update ValidRouteLifeTime functions are used for adding and updating the route table entry. In these functions, 4*activeRouteTimeoutvalue is used as lifetime of route table entries. Then it is changed to 1*activeRouteTimeoutvalue)

```
double activeRouteTimeout@unit(s) = default(3s); // the timeout value for cached routes. If Hello messages are used, then the ACTIVE_ROUTE_TIMEOUT parameter value MUST be more than the value (ALLOWED_HELLO_LOSS * HELLO_INTERVAL).
```

netDiameter: This parameter specifies the maximum possible number of hops between two nodes in the network. In INET framework, default value of this parameter is 35. This value has been changed during the simulations.

```
int netDiameter = default(35); // the maximum possible number of hops between two nodes in the network
```

nodeTraversalTime: This parameter is a conservative estimate of the average one hop traversal time for packets and should include queuing delays, interrupt processing times and transfer times. In INET framework, default value of this parameter is 40 milliseconds. This value has not been changed during the simulations.

```
double nodeTraversalTime@unit(s) = default(0.04s); // an estimation of the average one-hop traversal time
```

rerrRatelimit: This parameter specifies the maximum number of RERR messages that the AODV routing protocol may originate in 1 second. In INET framework, default value of this parameter is 10. This value has not been changed during the simulations.

```
int rerrRatelimit = default(10); // maximum number of RERR messages that the AODV may originate in 1s.
```

rreqRetries: This parameter specifies the number of times that AODV routing protocol will repeat an expanded ring search for a destination if no Route Reply Packet is received within the specified amount of time. In INET framework, default value of this parameter is 2. This value has not been changed during the simulations.

```
int rreqRetries = default(2); // specifies the number of times AODV will repeat an expanded ring search for a destination
```

rreqRatelimit: This parameter specifies the maximum number of RREQ messages that the AODV routing protocol may originate in 1second. Since the protocol uses expanding ring search and tries to prevent unnecessary dissemination of RREQ messages, number of originated RREQ messages varies. In INET framework, default value of this parameter is 10. This value has not been changed during the simulations.

```
int rreqRatelimit = default(10); // maximum number of RREQ messages that the AODV may originate in 1s.
```

ttlStart: This parameter specifies the TTL value of IP header when initiating a route request. In INET framework, default value of this parameter is 2. This value has been changed during the simulations.

```
int ttlStart = default(2); // specifies the TTL value when initiating a route request
```

ttlIncrement: This parameter specifies the value by which the TTL value of IP header will be incremented each time a RREQ is retransmitted. In INET framework, default value of this parameter is 2. This value has been changed during the simulations.

```
int ttlIncrement = default(2); // specifies the value by which the TTL will be incremented each time a RREQ is retransmitted
```

ttlThreshold: This parameter specifies the maximum value of TTL over which NET_DIAMETER value will be used to broadcast any RREQ. In INET framework, default value of this parameter is 7. This value has been changed during the simulations.

```
int ttlThreshold = default(7); // the maximum value of TTL over which NET_DIAMETER value will be used to broadcast any RREQ
```

myRouteTimeout: This parameter specifies the value of the lifetime field that a destination node places in RREP messages. In INET framework, the value of this parameter is 2 *activeRouteTimeout seconds. This value has been changed during the simulations. (Note: In Aodv.cc file, 2*myRouteTimeoutvalue is used as lifetime field of RREP message. Then it is changed to 1*myRouteTimeout value.)

```
double myRouteTimeout@unit(s) = default(2 * activeRouteTimeout); // the value of the lifetime field that a destination node places in RREPs
```

deletePeriod: This parameter specifies the time after which an expired route is deleted.In INET framework, an expired route is deleted after 5 multiplied by the greater of activeRouteTimeout and helloInterval. Since the activeRouteTimeout value has been changed during the simulations, this value has also been changed during the simulations.

```
double deletePeriod@unit(s) = default(5 * max(activeRouteTimeout, helloInterval)); // the time after which an expired route is deleted
```

blacklistTimeout: This parameter specifies the time after which a blacklisted node is removed from the blacklist. To prevent the processing of RREQ packets received from unidirectional links, when a node detects that the transmission of RREP message has been failed (e.g. absence of RREP ACK), it remembers the next-hop of the failed RREP in a "blacklist" set. A node ignores all RREQs received from any node in its blacklist set. Nodes are removed from the blacklist set after a blacklistTimeoutperiod. Since this period shall be set to the upper limit of the time that it takes to process all RREQ retry attempts, this value is set to rreqRetries multiplied by netTraversalTime, this value has not been changed during the simulations.

```
double blacklistTimeout@unit(s) = default(rreqRetries * netTraversalTime); // the time after which a blacklisted node is removed from the blacklist
```

netTraversalTime: This parameter specifies an estimation of the traversal time for the complete network. If a RREP of a RREQ message is not received within netTraversalTime, the originator node tries again to discover a route by broadcasting another RREQ until retry of RREQ is smaller than rreqRetries number. In INET framework, the value of this parameter is equal to the 2 * nodeTraversalTime * netDiameter. Since the netDiameter parameter depends on the size of networks and can be changed for each simulation topology, this value has also been changed during the simulations.

```
double netTraversalTime@unit(s) = default(2 * nodeTraversalTime * netDiameter); // an estimation of the traversal time for the complete network
```

pathDiscoveryTime: This parameter specifies the buffer timeout for each broadcasted RREQ message. Before broadcasting the RREQ, the node buffers the RREQ ID and the Originator IP address of the received RREQ for pathDiscoveryTime. In this way, when the node receives the packet again from its neighbors, it will not reprocess and forward the packet. In INET framework, the value of this parameter is equal to 2 * netTraversalTime. Since the netTraversalTime parameter depends on the size of networks and can be changed for each simulation topology, this value has also been changed during the simulations.

```
double pathDiscoveryTime@unit(s) = default(2 * netTraversalTime); // buffer timeout for each broadcasted RREQ message
```

ringTraversalTime: This parameter specifies the timeout of receiving RREP for expanding ring search technique. If an originator node does not receive a RREP of a RREQ, the originator node tries again to broadcast RREQ with new TTL which is not greater than `ttlThreshold`. Since the nodeTraversalTime has not been changed during the simulations, also this value has not been changed during the simulations.

```
double ringTraversalTime@unit(s) = default(2 * nodeTraversalTime * (ttl + timeoutBuffer)); // an estimation of the traversal time for the complete network
```

Used OMNET++ parameters and application details are given below.

Mobility Models

In the Random Waypoint (RWP) mobility model the nodes move in line pieces. A random destination position and a random speed is chosen for each line piece. When the node reaches the destination, it waits for the time, which can also be defined as a variable. After this time the algorithm calculates a new random position and random speed which can be uniformly distributed between minimum and maximum speeds. The node then travels toward the newly chosen destination at the selected speed. Upon arrival, the mobile node pauses for a specified time period before starting the process again.

Gauss-Markov (GM) mobility model uses the Gauss-Markov mobility model that involves random elements when describing the motion. It has an alpha parameter which can run from 0 (totally random motion) to 1 (deterministic linear motion), with the default value of 0.5. The random variable has a mean of 0, and its variance can be set by the variance parameter. The margin parameter adds a margin to the boundaries of the constraint area, so that the mobility bounces back before reaching it. The mobility module is set to totally random motion, with a variance of 0.5.

Linear Mobility (LM) model uses linear movements with a constant speed and angle of movement. Angle of movement only changes when the mobile node reaches the edge of the movement area. When the node reaches the edge, then it reflects off its movement with the same angle.

Wireless Interface (Physical Layer and MAC Layer)

AckingWirelessInterface is a wireless interface that generates simplicity for scenarios where physical and mac layer effects can be completely ignored. In this thesis, the main goal of simulations is testing the basic functionality AODV and IAODV, as a result AckingWirelessInterface is chosen for simulations. AckingWirelessInterface contains a unit disk radio (UnitDiskRadio) and a negligible MAC protocol (AckingMac).

Unit disk radio model provides a very simple but fast and predictable physical layer behavior. In this model, transmissions are described with a few distance based parameters: communication range, interference range, and detection range. Whether the reception is successful or not, depends on the distance between the transmitter and the receiver. The most important parameter of UnitDiskRadio model is the transmission range. When a radio transmits a packet, all other radios within transmission range are able to receive the packet correctly.

AckingMac implements a negligible MAC protocol that has packet encapsulation and decapsulation, but no real medium access procedure. Packets are simply transmitted on the wireless channel as soon as the transmitter becomes idle. There is no carrier sense, collision avoidance, or collison detection. AckingMac also provides an optional out-of-band acknowledgement mechanism (using C++ function calls, not actual wirelessly sent frames), which is turned on by default. There is no retransmission: if the acknowledgement does not arrive after the first transmission, the MAC gives up and counts the packet as failed transmission.

3.2.2. AODV Simulation with Stationary Nodes

For simplicity and understanding the performance of different settings of initial parameters, the first simulation for AODV routing protocol is done with 4 stationary UAV nodes and distance between each pair of UAVs is lower than the communication range of wireless channel.

3.2.2.1. Stationary Case-1

In this case, period of transmitting the application data is greater than the life time of active routes. So, the originator node always start a new route request for each application data transmission attempt. Simulation parameters for Omnet++ are given in *Table 3.5*.

Table 3.5 *Simulation Parameters for Stationary Case*

SIMULATION PARAMETER	VALUE
MOBILITY MODEL	Stationary
SIZE	800 x 800 m
RADIO TYPE	UnitDiskRadio
COMMUNICATION RANGE	250 m
WLAN TYPE	Acking Wireless Interface
WLAN BIT RATE	2 Mbps
NUMBER OF NODES	4
UAV[0] POSITION (NODE-1)	x: 200 m, y: 300 m
UAV[1] POSITION (NODE-2)	x: 350 m, y: 300 m
UAV[2] POSITION (NODE-3)	x: 500 m, y: 300 m
UAV[3] POSITION (NODE-4)	x: 650 m, y: 300 m
APPLICATION TYPE	Ping Request from UAV[0] to UAV[3]
PING START TIME	`uniform(1s,5s)`
PING PERIOD TIME	10 s
SIMULATION TIME	1000 s

In this simulation, communication range is 250 m and all nodes are stationary then status of wireless communication between each pair is give in *Table 3.6*.

Table 3.6 *Wireless Connectivity Matrix between Stationary Nodes*

Node Name	**UAV[0]**	**UAV[1]**	**UAV[2]**	**UAV[3]**
UAV[0]	N/A	OK	Not OK	Not OK
UAV[1]	OK	N/A	OK	Not OK
UAV[2]	Not OK	OK	N/A	OK
UAV[3]	Not OK	Not OK	OK	N/A

AODV routing protocol parameters are given in *Table 3.7*.

Table 3.7 *AODV Parameters used for Simulation*

AODV PARAMETER	**VALUE**
HELLO MESSAGE USAGE	Not Used
DESTINATIONONLYFLAG	Not used
ACTIVEROUTETIMEOUT	3 s
MY ROUTE TIMEOUT	2 * 3 = 6 s
DELETEPERIOD	5 * 3 = 15 s
NETDIAMETER	4
NODETRAVERSALTIME	0.04 s
NET TRAVERSAL TIME	2 * 0.04 * 4 = 0.32 s
PATHDISCOVERYTIME	2 * 0.32 = 0.64 s
RERRRATELIMIT	10
RREQRETRIES	2
BLACK LIST TIMEOUT	2 * 0.32 = 0.64 s
TTL START	4
TTLINCREMENT	4
TTLTHRESHOLD	4

Simulation screen of Omnet++ with above settings is given in Figure 3.16.

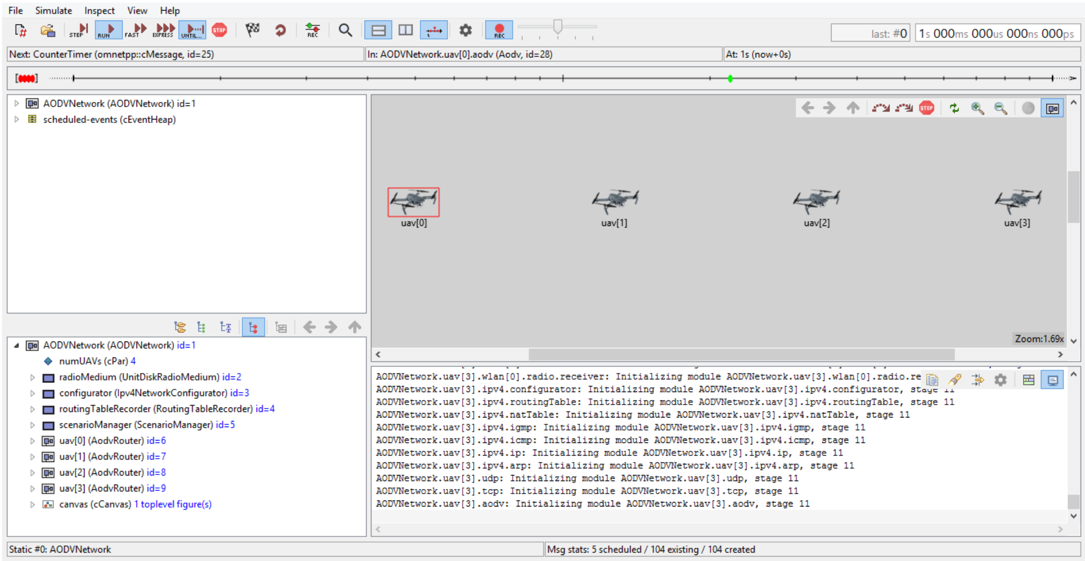

Figure 3.16 AODV simulation screen view of stationary model in Omnet++

3.2.2.1.1. Simulation Results

Since the simulation time is 1000 seconds and ping interval is 10 seconds, then total number of ping request is 100. Statistics of ping request are given in *Table 3.8*.

Table 3.8 *Data Statistics of the Stationary Case-1*

Number of Ping Request Sent	100
Number of Ping Request Received	100
Loss Rate of Ping Requests	% 0
Minimum Round-Trip Time (RTT)	10.0061 ms
Average Round-Trip Time	17.4338 ms
Maximum Round-Trip Time	24.266 ms
Standard Deviation of RTT	3.13077 ms
Variance of RTT	9.8e-6

AODV message statistics are given in *Table 3.9*.

Table 3.9 *AODV Message Statistics of the Stationary Case-1*

Module Name	Total sent RREP	Total sent RREQ
AODVNetwork.uav[0].aodv	0	100
AODVNetwork.uav[1].aodv	100	100
AODVNetwork.uav[2].aodv	100	100
AODVNetwork.uav[3].aodv	100	0
Total (600)	300	300

Since all UAVs are connected each other like daisy chain, loss rate of ping request is zero. Also, it can be seen that total number of overhead messages (RREP and RREQ) is 600 and same as the number of application message transmissions. So, overhead ratio is % 50 for number of packet transmissions.

3.2.2.2. Stationary Case-2

In this case, period of transmitting the application data is lower than the life time of active routes. So, the originator node only starts one route request for first data transmission attempt. Since there is an active route entry, rest of the data transmission is done without route request process.

All simulation parameters are same as with Case-1 except the Ping Period Time and the Simulation Time parameters. Ping Period Time is set to 2 s which is lower than the active route timeout and Simulation Time is set to 200 s in order to reach 100 ping requests.

3.2.2.2.1. Simulation Results

Since the simulation time is 1000 seconds and ping interval is 2 seconds, then total number of ping request is 100. Statistics of ping request are given in *Table 3.10*.

Table 3.10 *Data Statistics of the Stationary Case-2*

Number of Ping Request Sent	100
Number of Ping Request Received	100
Loss Rate of Ping Requests	% 0
Minimum Round-Trip Time (RTT)	2.787 ms
Average Round-Trip Time	3.00396ms
Maximum Round-Trip Time	24.266 ms
Standard Deviation of RTT	2.15873 ms
Variance of RTT	4.7e-6

AODV message statistics are given in *Table 3.11*.

Table 3.11 *AODV Message Statistics of the Stationary Case-2*

Module Name	**Total sent RREP**	**Total sent RREQ**
AODVNetwork.uav[0].aodv	0	1
AODVNetwork.uav[1].aodv	1	1
AODVNetwork.uav[2].aodv	1	1
AODVNetwork.uav[3].aodv	1	0
Total (6)	3	3

The loss rate of ping request is same as the Stationary Case-1. Since the period of transmitting the application data is lower than the life time of active routes, route decision for each node is done for first ping request and all routes are active during the rest of the simulation. As a result, overhead ratio of packet transmission is % 1 (6/200) and very low compared to Stationary Case-1. Also, round-trip time statistics are affected positively. Average of round trip time is 3 ms for Stationary Case-2 and

17.4 ms for Stationary Case-2. It is obvious that average latency of first case is much higher than the average latency of the second case.

When the results of both stationary cases are compared, although AODV routing protocol parameters and topology of the network do not change, changes in the application properties affect the simulation results too much. So, it is possible that if the AODV routing protocol is improved, the simulation results of the first case can be improved and same as the results of second case. Since, all nodes are stationary and distance between each pair of nodes do not increase during the simulation, if this situation is known by each node, these nodes can decide that wireless communication quality is always same for this situation and do not inactivate the active routes. How this situation can be known and improvement of the AODV routing protocols explained in Chapter 4.

3.2.3. AODV Simulation with Mobile Nodes

Nowadays, UAVs are used for commercial and military applications. The usage of UAVs is explained in an operational concept (OPCON) document for both applications. This OPCON document explains all steps of operation, and UAVs communication requirements are given in this document. In some parts of the operation, specific UAVs are moving in a group manner or independently. An example of operation concept is given below in detail.

Since the disaster areas are located in difficult terrain conditions, it is impossible to reach and monitor these zones by using traditional transportation vehicles. So, in most cases, UAVs are frequently used for disaster area monitoring. It is assumed that, after a natural disaster, there is a 1 km square disaster area (1 x 1 km) and it is requested to investigate the disaster area with UAVs. In this operation concept, the mission consists of two parts. In the first half of the mission, UAVs are flying on the disaster area and if any UAV gets useful information, it sends this info to leader UAV as soon as possible. In the second half of the mission, UAVs are divided into four groups and disaster area also divided into four areas with same sizes (each one is 250 x 250 m). Each group members sends useful information to one of the other group member via its group leader.

Simulation details of these two consecutive scenarios are explained below.

3.2.3.1. Simulation of the first part of the mission

In this case, it is assumed that there are 20 UAVs flying on the disaster area with linear mobility model. Initial positions and moving angles of UAVs are chosen randomly. All UAVs are moving with same speed, and if any UAV reaches the edge of the area, it reflects its movement with the same movement angle.

Two simulations are done for this case. In the first simulation, the period of transmitting the application data is greater than the life time of active routes. So, the originator node always start a new route request for each application data transmission attempt. In the second simulation, the period of transmitting the application data is lower than the life time of active routes. So, if there is not an active route entry in the routing table, the originator node starts route request. Otherwise, the originator node uses active route information for data transmitting. The simulation parameters for Omnet++ are given in Table 3.12 and used AODV routing protocol parameters for the simulation are given in Table 3.13

Table 3.12 Simulation Parameters for Random Mobility Model

SIMULATION PARAMETER	VALUE
MOBILITY MODEL	Linear Mobility
SIZE	1000 x 1000 m
RADIO TYPE	UnitDiskRadio
COMMUNICATION RANGE	250 m
WLAN TYPE	AckingWirelessInterface
WLAN BIT RATE	2 Mbps
NUMBER OF UAVS	20
SPEED OF UAVS	50 mps
INITIAL POSITION OF UAVS	Randomly
APPLICATION TYPE	Ping Request from UAV[1] to UAV[0] (useful data simulation) UAV[0] is the leader UAV.
PING START TIME	`uniform(0s,1s)`
PING PERIOD TIME	First simulation: 10 s Second simulation: 2s
SIMULATION TIME	First simulation: 1000 s Second simulation: 200 s

Table 3.13 *AODV Parameters used for Simulation*

AODV PARAMETER	**VALUE**
HELLO MESSAGE USAGE	Not used
DESTINATIONONLYFLAG	Not used
ACTIVEROUTETIMEOUT	3 s
MY ROUTE TIMEOUT	2 * 3 = 6 s
DELETEPERIOD	5 * 3 = 15 s
NETDIAMETER	19
NODETRAVERSALTIME	0.04 s
NET TRAVERSAL TIME	2 * 0.04 * 19 = 1.52 s
PATHDISCOVERYTIME	2 * 1.52 = 3.04 s
RERRRATELIMIT	10
RREQRETRIES	2
BLACK LIST TIMEOUT	2 * 1.52 = 3.04 s
TTL START	2
TTLINCREMENT	2
TTLTHRESHOLD	7

Simulation screen of Omnet++ with above settings is given in Figure 3.17.

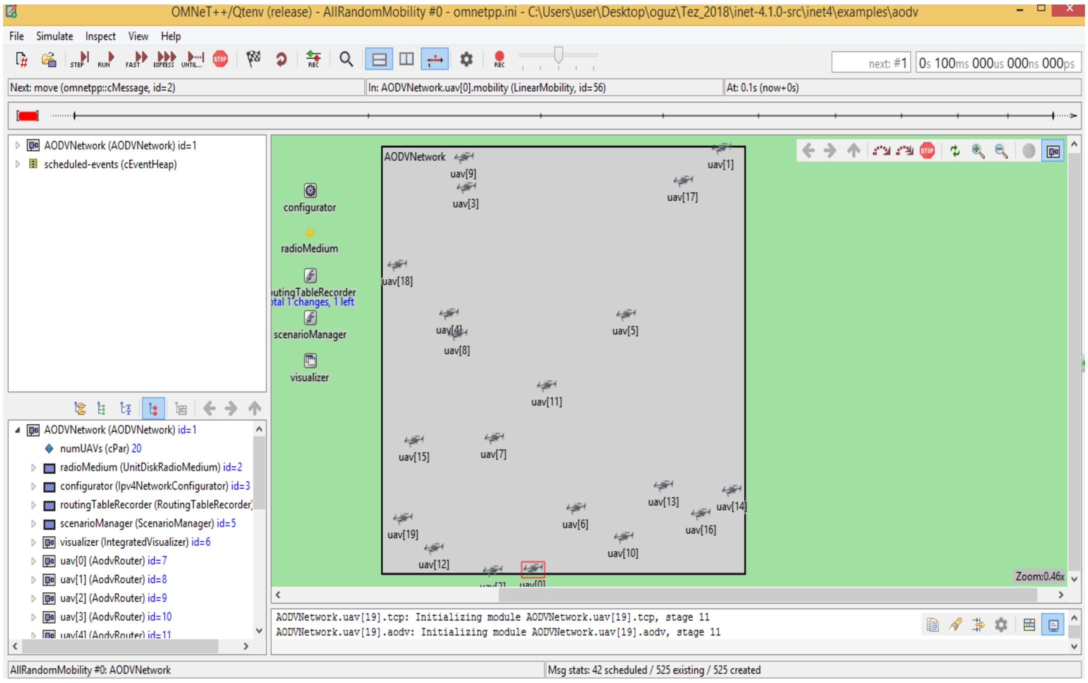

Figure 3.17 AODV simulation screen view of linear mobility in Omnet++

3.2.3.1.1. Simulation Results of Linear Mobility with 10 s ping interval

Since the simulation time is 1000 seconds and ping interval is 10 seconds, then total number of ping request is 100. Simulation results are given in Table 3.14 and Table 3.15. These results are used in Chapter 4 to compare the original and improved AODV routing algorithm protocols.

Table 3.14 *Data Statistics of the Linear Mobility Case-1*

Number of Ping Request Sent	100
Number of Ping Request Received	56
Loss Rate of Ping Requests	% 44
Minimum Round-Trip Time (RTT)	2.09 ms
Average Round-Trip Time	1450.82ms
Maximum Round-Trip Time	4831.81ms
Standard Deviation of RTT	1584.97ms
Variance of RTT	2.5121

Table 3.15 *AODV Message Statistics of the Linear Mobility Case-1*

Module	Total sent RREP	Total sent RREQ	Total sent RRER
AODVNetwork.uav[0].aodv	89	10	16
AODVNetwork.uav[1].aodv	5	353	12
AODVNetwork.uav[2].aodv	9	119	1
AODVNetwork.uav[3].aodv	3	95	0
AODVNetwork.uav[4].aodv	9	83	2
AODVNetwork.uav[5].aodv	12	95	0
AODVNetwork.uav[6].aodv	4	125	0
AODVNetwork.uav[7].aodv	16	100	3
AODVNetwork .uav[8].aodv	10	105	1
AODVNetwork.uav[9].aodv	7	103	2
AODVNetwork.uav[10].aodv	11	117	1
AODVNetwork.uav[11].aodv	14	102	1
AODVNetwork.uav[12].aodv	9	105	0
AODVNetwork.uav[13].aodv	14	89	2
AODVNetwork.uav[14].aodv	8	113	1
AODVNetwork.uav[15].aodv	11	117	0
AODVNetwork.uav[16].aodv	13	94	1
AODVNetwork.uav[17].aodv	7	100	2
AODVNetwork.uav[18].aodv	13	102	2
AODVNetwork.uav[19].aodv	14	95	0
Total (2547)	278	2222	47

3.2.3.1.2. Simulation Results of Linear Mobility with 2 s ping interval

Since the simulation time is 200 seconds and ping interval is 2 seconds, then total number of ping request is 100. Simulation results are given in Table 3.16 and Table 3.17. These results are used in Chapter 4 to compare the original and improved AODV routing algorithm protocols.

Table 3.16 *Data Statistics of the Linear Mobility Case-2*

Number of Ping Request Sent	100
Number of Ping Request Received	41
Loss Rate of Ping Requests	% 59
Minimum Round-Trip Time (RTT)	0.92ms
Average Round-Trip Time	975.808ms
Maximum Round-Trip Time	4935.82ms
Standard Deviation of RTT	1351.03ms
Variance of RTT	1.82527

Table 3.17 *AODV Message Statistics of the Linear Mobility Case-2*

Module	Total sent RREP	Total sent RREQ	Total sent RRER
AODVNetwork.uav[0].aodv	26	6	3
AODVNetwork.uav[1].aodv	2	128	19
AODVNetwork.uav[2].aodv	8	20	3
AODVNetwork.uav[3].aodv	8	37	3
AODVNetwork.uav[4].aodv	4	36	0
AODVNetwork.uav[5].aodv	1	32	0
AODVNetwork.uav[6].aodv	0	46	0
AODVNetwork.uav[7].aodv	6	44	3
AODVNetwork.uav[8].aodv	4	28	2
AODVNetwork.uav[9].aodv	1	32	0
AODVNetwork.uav[10].aodv	1	44	1
AODVNetwork.uav[11].aodv	2	37	1
AODVNetwork.uav[12].aodv	3	37	0
AODVNetwork.uav[13].aodv	6	24	1
AODVNetwork.uav[14].aodv	7	43	3
AODVNetwork.uav[15].aodv	6	41	2
AODVNetwork.uav[16].aodv	6	34	3
AODVNetwork.uav[17].aodv	1	38	0
AODVNetwork.uav[18].aodv	3	32	3
AODVNetwork.uav[19].aodv	1	43	0
Total (925)	96	782	47

3.2.3.2. Simulation of the second part of the mission

In this case, it is assumed that there are 4 group of UAVs and each group has 5 UAVs. The simulation area is also divided into 4 areas with the same size. Each group is flying on one of the divided areas. Two simulations are done for this case. In the first simulation, the period of transmitting the application data is greater than the life time of active routes. So, the originator node always starts a new route request for each application data transmission attempt. In the second simulation, the period of transmitting the application data is lower than the life time of active routes. So, the originator node only starts one route request for first data transmission attempt. Since there is an active route entry, rest of the data transmission is done without route request process.

The simulation parameters for Omnet++ are given in *Table 3.18*.

Table 3.18 *Simulation Parameters for Group Mobility Model*

SIMULATION PARAMETER	**VALUE**
MOBILITY MODEL	Group Mobility (Linear and Attached Mobility Models)
SIZE	1000 x 1000 m
RADIO TYPE	UnitDiskRadio
COMMUNICATION RANGE	250 m
WLAN TYPE	AckingWirelessInterface
WLAN BIT RATE	2 Mbps
NUMBER OF UAVS	20
SPEED OF UAVS	50 mps
INITIAL POSITION OF GROUP-1	UAV[0] → x= 250 m, y= 250 m UAV[1], UAV[2], UAV[3] and UAV[4] are attached to UAV[0] (leader of Group-1)
CONSTRAINT AREA OF GROUP-1	x_{min}= 0 m and x_{max}= 500 m y_{min}= 0 m and y_{max}= 500 m
INITIAL POSITION OF GROUP-2	UAV[5] → x= 750 m, y= 250 m UAV[6], UAV[7], UAV[8] and UAV[9] are attached to UAV[5] (leader of Group-2)
CONSTRAINT AREA OF GROUP-2	x_{min}= 500 m and x_{max}= 1000 m y_{min}= 0 m and y_{max}= 500 m

Table 3.18 (Cont'd)

INITIAL POSITION OF GROUP-3	UAV[10] → x= 250 m, y= 750 m UAV[11], UAV[12], UAV[13] and UAV[14] are attached to UAV[10] (leader of Group-3)
CONSTRAINT AREA OF GROUP-3	x_{min}= 0 m and x_{max}= 500 m y_{min}= 500 m and y_{max}= 1000 m
INITIAL POSITION OF GROUP-4	UAV[15] → x= 750 m, y= 750 m UAV[16], UAV[17], UAV[18] and UAV[19] are attached to UAV[15] (leader of Group-4)
CONSTRAINT AREA OF GROUP-4	x_{min}= 500 m and x_{max}= 1000 m y_{min}= 500 m and y_{max}= 1000 m
APPLICATION TYPE	Ping Request from one group member to another group member, which is out of communication range. Data transfer is done by over group leaders. (ex: Group-1 ping request pairs: • UAV[1] and UAV[4] sends ping request to each other via UAV[0] • UAV[2] and UAV[3] sends ping request to each other via UAV[0]
PING START TIME FOR EACH GROUP	First UAV →`uniform(0s,1s)` Second UAV →`uniform(1s,2s)` Third UAV →`uniform(2s,3s)` Fourth UAV →`uniform(3s,4s)`

Table 3.18 (Cont'd)

PING PERIOD TIME	First simulation: 10 s Second simulation: 2 s
SIMULATION TIME	First simulation: 1000 s Second simulation: 200 s

Used AODV routing protocol parameters for the simulation are given in *Table 3.19*.

Table 3.19 *AODV Parameters used for Simulation*

AODV PARAMETER	**VALUE**
HELLO MESSAGE USAGE	Not used
DESTINATIONONLYFLAG	Not used
ACTIVEROUTETIMEOUT	3 s
MY ROUTE TIMEOUT	2 * 3 = 6 s
DELETEPERIOD	5 * 3 = 15 s
NETDIAMETER	10
NODETRAVERSALTIME	0.04 s
NET TRAVERSAL TIME	2 * 0.04 * 10 = 0.8 s
PATHDISCOVERYTIME	2 * 0.8= 1.6 s
RERRRATELIMIT	10
RREQRETRIES	2
BLACK LIST TIMEOUT	2 * 0.8 = 1.6 s
TTL START	1
TTLINCREMENT	4
TTLTHRESHOLD	10

The simulation screen of the Omnet++ is given in Figure 3.18

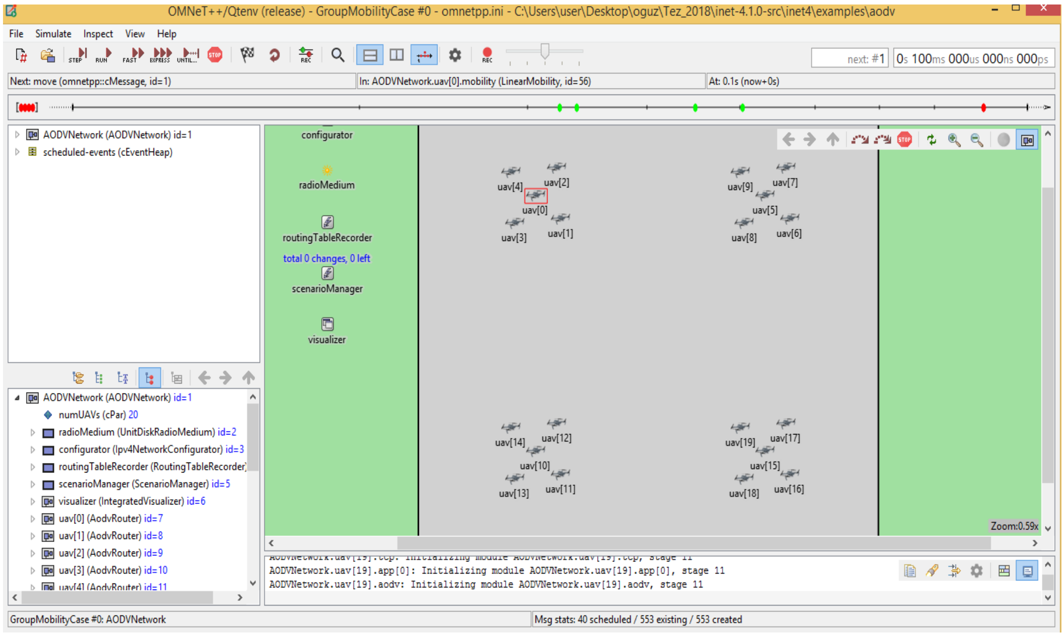

Figure 3.18 AODV simulation screen view of group mobility in Omnet++

3.2.3.2.1. Simulation Results of Group Mobility with 10 s ping interval

Simulation results are given in Table 3.2.2.14 and Table 3.2.2.15. These results are used in Chapter 4 to compare the original and improved AODV routing algorithm protocols.

Table 3.20 *Data Statistics of the Group Mobility Case-1*

UAV No	Loss Rate of Ping Requests	Minimum RTT (ms)	Average RTT (ms)	Maximum RTT (ms)
UAV[1]	%22	243.70	393.10	2253.48
UAV[2]	%23	244.97	378.43	1610.56
UAV[3]	%7	1.86	61.73	656.22
UAV[4]	%17	1.86	398.44	1611.49
UAV[6]	%16	244.77	451.90	2820.24
UAV[7]	%18	244.20	463.29	1857.13
UAV[8]	%3	1.86	104.00	921.33
UAV[9]	%12	1.86	120.99	1609.24

Table 3.20 (Cont'd)

UAV[11]	%30	244.95	426.10	1304.18
UAV[12]	%19	245.81	432.65	1304.76
UAV[13]	%5	1.86	100.92	2573.51
UAV[14]	%11	1.86	202.59	1856.17
UAV[16]	%24	244.35	505.69	2815.07
UAV[17]	%23	245.86	445.63	1842.25
UAV[18]	%4	1.86	109.84	896.81
UAV[19]	%8	1.86	142.66	1396.06
Average	**%15**	**123.34**	**296.12**	**1708.03**

Table 3.21 *AODV Message Statistics of the Group Mobility Case-1*

Module	Total sent RREP	Total sent RREQ	Total sent RRER
AODVNetwork.uav[0].aodv	170	427	32
AODVNetwork.uav[1].aodv	183	448	62
AODVNetwork.uav[2].aodv	112	535	40
AODVNetwork.uav[3].aodv	232	325	56
AODVNetwork.uav[4].aodv	207	409	60
AODVNetwork.uav[5].aodv	127	372	19
AODVNetwork.uav[6].aodv	112	445	40
AODVNetwork.uav[7].aodv	105	455	32
AODVNetwork.uav[8].aodv	195	277	46
AODVNetwork.uav[9].aodv	219	275	60
AODVNetwork.uav[10].aodv	145	404	19
AODVNetwork.uav[11].aodv	134	468	48
AODVNetwork.uav[12].aodv	112	484	40
AODVNetwork.uav[13].aodv	242	313	65
AODVNetwork.uav[14].aodv	211	329	63
AODVNetwork.uav[15].aodv	130	384	16
AODVNetwork.uav[16].aodv	126	461	53
AODVNetwork.uav[17].aodv	106	488	46
AODVNetwork.uav[18].aodv	220	265	55
AODVNetwork.uav[19].aodv	222	289	60
Total (12075)	**3310**	**7853**	**912**

3.2.3.2.2. Simulation Results of Group Mobility with 2 s ping interval

Simulation results are given in Table 3.22 and Table 3.23. These results are used in Chapter 4 to compare the original and improved AODV routing algorithm protocols.

Table 3.22 *Data Statistics of the Group Mobility Case-2*

UAV No	Loss Rate of Ping Requests	Minimum RTT (ms)	Average RTT (ms)	Maximum RTT (ms)
UAV[1]	% 0	0.928472	0.959013	3.98262
UAV[2]	% 0	0.928472	0.98045	6.12626
UAV[3]	% 0	0.928472	0.986703	6.6934
UAV[4]	% 0	0.928472	0.944356	2.50106
UAV[6]	% 0	0.928472	0.963724	4.45374
UAV[7]	% 0	0.928472	0.96384	4.4653
UAV[8]	% 0	0.928472	0.968645	4.90564
UAV[9]	% 0	0.928472	0.972692	5.30627
UAV[11]	% 0	0.928472	0.988024	6.88373
UAV[12]	% 0	0.928472	0.966286	4.70989
UAV[13]	% 0	0.928472	0.959794	4.02934
UAV[14]	% 0	0.928472	0.956996	3.7524
UAV[16]	% 0	0.928472	0.953474	3.4287
UAV[17]	% 0	0.928472	0.979427	6.02404
UAV[18]	% 0	0.928472	0.982182	6.24581
UAV[19]	% 0	0.928472	0.943544	2.4206
Average	**% 0**	0.928472	0.966821	4.74555

Table 3.23 *AODV Message Statistics of the Group Mobility Case-2*

Module	**Total sent RREP**	**Total sent RREQ**	**Total sent RRER**
AODVNetwork.uav[0].aodv	4	0	0
AODVNetwork.uav[1].aodv	3	1	0
AODVNetwork.uav[2].aodv	2	1	0
AODVNetwork.uav[3].aodv	1	1	0
AODVNetwork.uav[4].aodv	0	1	0
AODVNetwork.uav[5].aodv	4	0	0
AODVNetwork.uav[6].aodv	3	1	0
AODVNetwork.uav[7].aodv	2	1	0
AODVNetwork.uav[8].aodv	1	1	0
AODVNetwork.uav[9].aodv	0	1	0
AODVNetwork.uav[10].aodv	4	0	0
AODVNetwork.uav[11].aodv	3	1	0
AODVNetwork.uav[12].aodv	2	1	1
AODVNetwork.uav[13].aodv	1	1	0
AODVNetwork.uav[14].aodv	0	1	0
AODVNetwork.uav[15].aodv	4	0	0
AODVNetwork.uav[16].aodv	3	1	0
AODVNetwork.uav[17].aodv	2	1	1
AODVNetwork.uav[18].aodv	1	1	0
AODVNetwork.uav[19].aodv	0	1	0
Total (58)	**40**	**16**	**2**

CHAPTER 4

IMPROVED AODV (IAODV) ROUTING ALGORITHM

As mentioned in Chapter 3, if it is possible to detect the relative velocity between the flying UAV nodes, AODV routing algorithm can be improved. In a wireless communication, the quality of wireless channel can be predicted by using different cross layer metrics. The most important part of these metrics belong to physical layer of the network. By using these metrics and some other techniques, the relative velocity between nodes can also be predicted. In this chapter, first, the properties of the layered structure of wireless communication and usable metrics of physical layer are explained. Then a generic Routing Life Time Decision Method (RLTDM) for UFANETs routing algorithms is described and application of this method to AODV protocol (IAODV) is given. Finally, the IAODV routing algorithm is simulated by using Omnet++ simulation and simulation results are compared with the simulation results which are given in Chapter 3.

4.1. Layered Structure of Wireless Communication for UFANETs

The architecture of a wireless network defines the protocols and components in order to meet application requirements. One of the most popular standard for defining the network architecture is the seven-layer Open System Interconnect (OSI) Reference Model, developed by the International Standards Organization (ISO). This OSI model was first developed for wired networks. Later, the popularity of the wireless networks was increased with the technological developments and this OSI model has also been used for wireless networks with some modifications. The OSI model states a complete set of network functions which are grouped into seven layers and ordered from layer 1 (the lowest) to layer 7 (the highest). These seven layers are (from lower to higher) the physical layer, data link layer, network layer, transport layer, session layer, presentation layer, and the application layer. On the other hand, as mentioned in Chapter-1, the protocol stack of UFANET consists of five layers: physical layer, data link layer, network layer, transport layer and application layer. The lower four layers have the same name but the fifth layer in the UFANET model is equivalent to the combined session, presentation and application layers of the OSI model. The other main difference between these protocols stacks lies in the network layer. Mobile nodes, which can be host or router in UFANETs, use an ad hoc routing protocol to route packets. Layered structure of UFANETs and protocol examples are given in Table 4.1.

Table 4.1 *UFANETs Layers and Protocol Examples*

UFANETs Layered Model	**Protocol Examples**
Application Layer	File Transfer Protocol (FTP)
Transport Layer	Transmission Control Protocol (TCP) User Datagram Protocol (UDP) Sequenced Packet Exchange (SPX)
Network Layer	Internet Protocol (IP) Address Resolution Protocol (ARP) Ad Hoc Routing Protocols
Data Link Layer	Various IEEE 802.11 WiFi MAC Layer (802.11a/b/g/n/ac/ax/p)
Physical Layer	Various IEEE 802.11 WiFiPHY Layer (802.11a/b/g/n/ac/ax/p)

One of the most famous wireless ad hoc networks is defined in IEEE 802.11 standard. This standard describes the architectural view, emphasizing the separation of the system into two major parts: the MAC of the data link layer (DLL) and the PHY. These layers are intended to correspond closely to the lowest layers of the ISO/IEC basic reference model of Open Systems Interconnection (OSI) (ISO/IEC 7498- 1: 1994). In this standard wireless ad hoc network is defined as Independent Basic Service Set (IBSS) and layered structure of this mode is given in Figure 4.1. This mode of operation is possible when IEEE 802.11 wireless stations (STAs) are able to communicate directly. Because this type of IEEE 802.11 LAN is often formed without preplanning, this type of operation is often referred to as an *ad hoc network.*

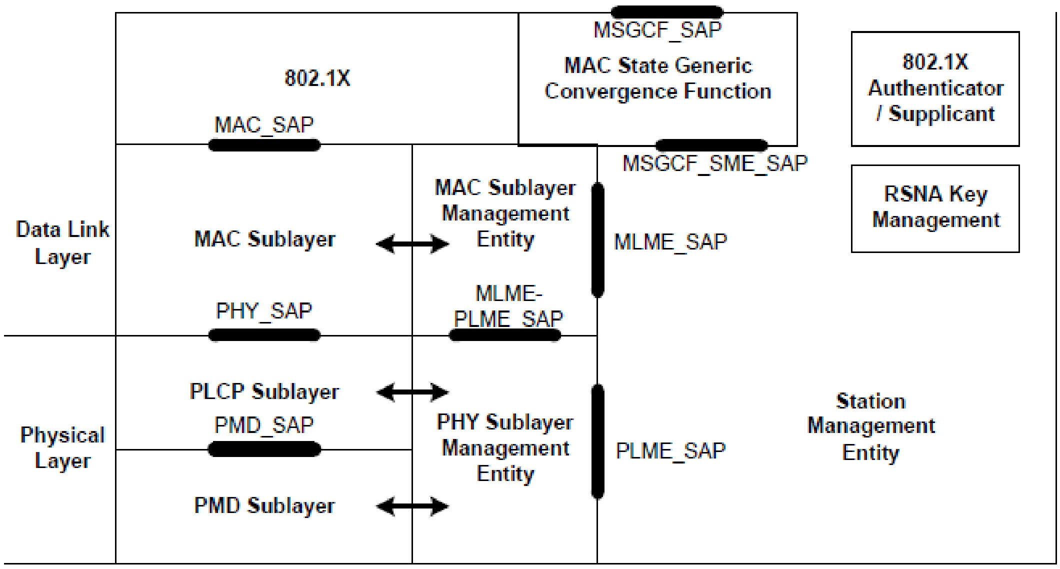

Figure 4.1 Layered Structure of IEEE 802.11 ad hoc networking

In this layered structure, it can be seen that there are interfaces between layers and management entities in order to share information between the layers. This type of information exchange is known as cross layer information sharing between the layers of system model. As mentioned previously, the quality of the wireless channel can be predicted by using these cross layer metrics. And most of them are Physical Layer (PHY) metrics and definition of these metrics is given in Chapter 4.2.

4.2. Physical Layer Metrics (useable for routing decision)

4.2.1. Signal to Noise Ratio (SNR)

In information theory, maximum capacity of the communication channel depends on the communication channel bandwidth and signal to noise ratio. This relation is given in Shannon–Hartley theorem:

$$C = B * \log_2(1 + \frac{S}{N})$$

Where C is the channel capacity in bits per second (a theoretical upper bound on bit rate), B is the bandwidth of the channel in hertz, S/N is the signal to noise ratio (SNR, expressed as a linear power ratio), S is the average received signal power over the bandwidth (measured in watts), N is the average power of the noise and interference over the bandwidth (measured in watts). It is obvious that if the SNR of a received signal can be estimated, then it is possible to know transmission quality of the communication channel. A higher SNR value means that the received signal strength is stronger enough than the noise levels, which allows higher data rates. On the other site, a lower SNR means that wireless device can operate at lower data rates, which decreases throughput.

SNR is given in logarithmic decibels (dB) format in most applications:

$$SNR_{dB} = 10 * \log_{10}\left(\frac{S}{N}\right) = 10 * \log_{10}(S) - 10 * \log_{10}(N) = S_{dB} - N_{dB}$$

Where S_{dB} is the average received power in decibel format (dbW) and N_{dB} is the average noise power in decibel format (dBW). Sometimes miliwatts (mW) is used as unit of the received power and noise power. In this case, dBm unit is used in decibel format:

$$S_{dBm} = 30 + S_{dBW}$$

$$N_{dBm} = 30 + N_{dBW}$$

SNR is illustrated in the Figure 4.2. For example, if the measured average power of signal is -70 dBm (-40 dBW) and noise floor is -85 dBm (-55 dBW) then

$$SNR_{dB} = S_{dBm} - N_{dBm} = -70 - (-85) = 15\ dB$$

In linear scale, the SNR is equal to

$$SNR = 10^{\frac{SNR_{dB}}{10}} = 10^{1.5} = 31{,}6$$

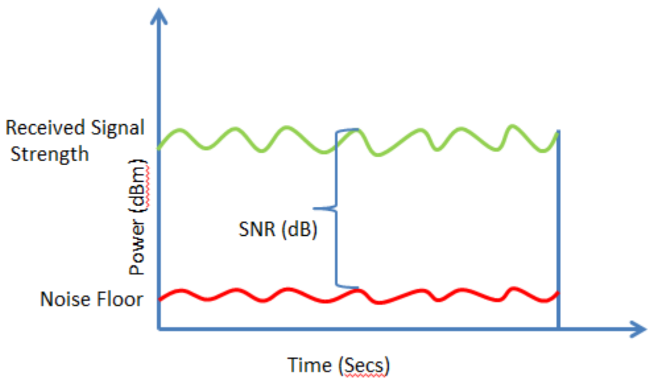

Figure 4.2 SNR definition in dB scale

The theoretical SNR requirement of BPSK modulation scheme is given in *Figure 4.3*. It can be seen that when the SNR is equal to the 12 dB then the bit error rate (BER) of receiver at physical layer is 10^{-8}.

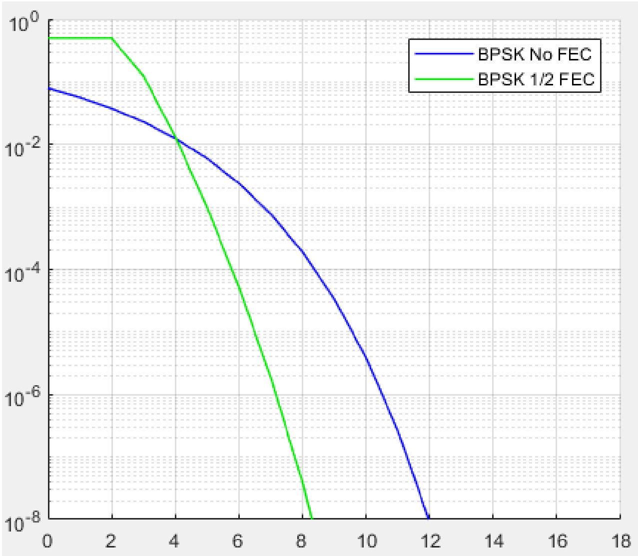

Figure 4.3 SNR vs BER for BPSK modulation scheme

In [29], the relationship between data rate and SNR of 802.11 a/g standards is given. This relationship is summarized in Table 4.2. It can be seen that the minimum required SNR for 6 Mbps data rate is 5 dB. In wireless communication, fade margin can also be added to this value. The overall results are given in Table 4.2.

Table 4.2 *Data Rate vs SNR for 802.11 a/g standards*

802.11a/g Data Rate (Mbps)	Modulation and Coding	Minimum SNR (dB)	Fade Margin (dB)	Required Link SNR (dB)
6	BPSK 1/2	5	9	14
9	BPSK 3/4	6	9	15
12	QPSK 1/2	7	9	16
18	QPSK 3/4	9	9	18
24	16QAM 1/2	13	9	22
36	16QAM 3/4	17	9	26
48	64QAM 2/3	20	9	29
54	64QAM 3/4	22	9	31

4.2.2. Doppler Shift (Frequency offset due to mobility)

The doppler shift of an electromagnetic wave is proportional to the frequency of the electromagnetic wave and the relative velocity between transmitter and receiver. The equation of this relation is given in below:

$$f_d = \frac{v_r * f_c}{c} * cos\alpha$$

Where f_d is doppler shift in Hz, v_r is relative speed between transmitter and receiver in m/s, f_c is carrier frequency of electromagnetic wave in Hz and $\alpha \epsilon [0, \pi]$ is the angle of the relative velocity vector. When the transmitter and the receiver approaches each other, then the doppler shift value is positive and if they recedes each other, then the doppler shift value is negative. It is obvious that if the relative velocity between transmitter and receiver is zero, then the doppler shift is also zero.

Frequency offset (FO) of a received signal can be measured and calculated at the receiver. This frequency offset occurs when the local oscillator (LO) signal for down-conversion in the receiver does not synchronize with the carrier signal of the received signal. Two important factors cause this frequency offset: frequency mismatch at the transmitter and the receiver local oscillators and doppler shift effect.

In order to calculate the doppler shift and relative velocity between transmitter and receiver, the frequency mismatch of transmitter and receiver LOs shall be known or this value shall be more smaller than the doppler shift. LOs of the transmitter and receiver can be calibrated and calibration results can be written in a lookup table at transmitter and receiver memories. After this calibration process, the frequency mismatch value of the transmitter and the receiver local oscillators can be known during the wireless communication. If this calibration process cannot be applied before the wireless communication, precise LOs shall be used at the transmitter and the receiver.

The 802.11 g standard uses 2.4 GHz carrier frequency. If the speed of nodes in UFANET network is 50 m/s, then the maximum absolute value of doppler shift is

$$f_d = \frac{(2 * 50) * 2.4 * 10^9}{3 * 10^8} = 800\ Hz$$

The absolute value of this doppler shift is between 0 and 800 Hz. If both transmitter and receiver uses rubidium oscillators as LO, then the frequency ambiguity of LO can be calculated by using long term stability and temperature drift specifications of LO. An example of Rubidium LO is given in [30]. Specifications of this LO are

- Frequency offset over temperature range = +/- $1*10^{-10}$
- Long term stability = $5*10^{-10}$ (per year)

The carrier frequency of electromagnetic wave is 2.4 GHz, then frequency ambiguity at the transmitter and the receiver is

$$f_a = 2.4 * 10^9 * (2 * 10^{-10} + 5 * 10^{-10}) \cong 2\ Hz$$

The total frequency mismatch ambiguity at the received signal is

$$f_m = 2 * f_a = 2 * 2 = 4\ Hz$$

This frequency mismatch ambiguity (4 Hz) means that the relative velocity lower than 0.5 m/s cannot be measured or calculated at the receiver. Since the nodes speed of UFANET is 50 m/s and 50 m/s >> 0.5 m/s, it is possible to measure the relative velocity. If the calculated relative velocity is smaller than k*0.5 m/s (where k is an integer, used for a control parameter), then there is an ambiguity, so the relative velocity information cannot be used for any decision or calculation.

4.2.3. Received Signal Strength Indicator (RSSI) and Received Signal Reference Power (RSRP)

RSSI is the power of received signal at the receiver input. This value is measured at analog part of the receiver chain.

RSRP is measured at digital part of the receiver by using reference pilot signals. Then this value is scaled by using pilot signals bandwidth and total bandwidth of received signal. There are 4 pilot carriers and 48 data carriers in an OFDM symbol of 802.11 a/g wifi signal. Since the digital measurements are done by using pilot signals, then the total bandwidth power ($RSRP_T$) can be calculated by using the total of the pilot signals power ($RSRP_A$).

$$RSRP_T = RSRP_A * 13$$

Since the RSRP measurement uses reference pilot signals power, it is more robust to jamming or interfering signal existence. In case of a single tone jammer or interfering signal, the measured RSSI at the analog part of the receiver can give wrong information about the received signal power. On the other hand, RSRP measurement uses reference pilot signals power, it is more reliable than the RSSI measurement in case of jammer signal existence (It is assumed that carrier frequency of jammer signal differs from carrier frequencies of pilot signals). Assume that there is a single tone jammer signal at -40 dBm level and there is a target signal at -70 dBm. In this case, the result of RSSI measurement is about -40 dBm, and the result of RSRP measurement is about -70 dBm.

Normally, the transmitted RF signal is exposed to free space path loss (FSPL). The power of received signal at receiver is calculated with the formula given below. By using this formula and measured RSSI (or RSRP) value, the estimated distance between two nodes can be calculated.

$$RSSI\ (or\ RSRP) = P_{Tx} + G_{Tx} - FSPL + G_{Rx}$$

Where P_{Tx} is the transmitted power (in dBm), G_{Tx} is the transmitter antenna gain (in dB), G_{Rx} is the receiver antenna gain (in dB) and FSPL is the free space path loss between transmitter and receiver.

Figure 4.4 Wireless Communication

The free space path loss (FSPL) is calculated by using carrier frequency of electromagnetic wave and distance between transmitter and receiver.

$$FSPL = 32{,}44 + 20 * log_{10}(d) + 20 * log_{10}(f_c)$$

Where f_c is the carrier frequency (MHz) and d (Km) is the distance between Tx and Rx

By using these formulas and measured RSSI (or RSRP) value, the estimated distance between two nodes can be calculated.

4.3. A new Routing Life Time Decision Method (RLTDM)

This RLTDM uses physical (PHY) layer parameters of beacon signal (i.e Hello packet in AODV or independent beacon source) for routing life time decision. The used PHY layer parameters are

- Measured Signal to Noise Ratio (MSNR): SNR of the beacon signal is calculated at PHY layer. The modulation and coding scheme used at PHY layer works well at desired SNR levels.
- Doppler shift: Relative velocity of one moving node to another moving node can be calculated from Doppler shift.
- Received power: RSSI (or RSRP) is the power of received signal at the receiver input. This value can be measured at PHY layer.

The flowchart of the RLTDM is given in Figure 4.5. Routing life time can be optimized by using this method. In the first stage of the algorithm, the measured SNR (MSNR) value is checked by using targeted SNR (TSNR). It is assumed that the most reliable metric for link quality is SNR. So, there is ±3 dB threshold value for SNR measurement. If the MSNR value is 3 dB lower than target SNR then it is assumed that there is no reliable link between two nodes. If the MSNR value is bigger than TSNR-3 and lower than TSNR+3, the algorithm continues to countdown the remaining routing life time (RRLT). If the MSNR value is bigger than TSNR+3, it is assumed that there is a strong signal for wireless communication. Then if the calculated relative velocity (CRV) by using doppler shift is available and equal to or lower than zero, it can be said that two nodes are not moving away from each other. In this condition, the algorithm sets the RRLT to maximum routing life time (MRLT). MRLT is a configurable parameter can be used as activeRouteTimeout parameter of the AODV routing algorithm. If the CRV value is bigger than zero it is assumed that two nodes are moving from each other, then the estimated link life time (ELLT) is calculated by using RSSI/RSRP and link margin values. After the calculation, the algorithm sets RRLT to ELLT. Normally the RSRP value is used during the condition checks and calculations, if RSRP value is not available then the RSSI value can be used. If CRV is not available, present and previous values of RSSI or RSRP are compared, if present value is greater than the previous value, it can be said that two nodes are not moving away from each other. In this condition, the algorithm sets the RRLT to maximum routing life time (MRLT). If RSSI/RSRP value is decreased, it is assumed that two nodes are moving from each other, then the algorithm continues to countdown the remaining routing life time (RRLT).

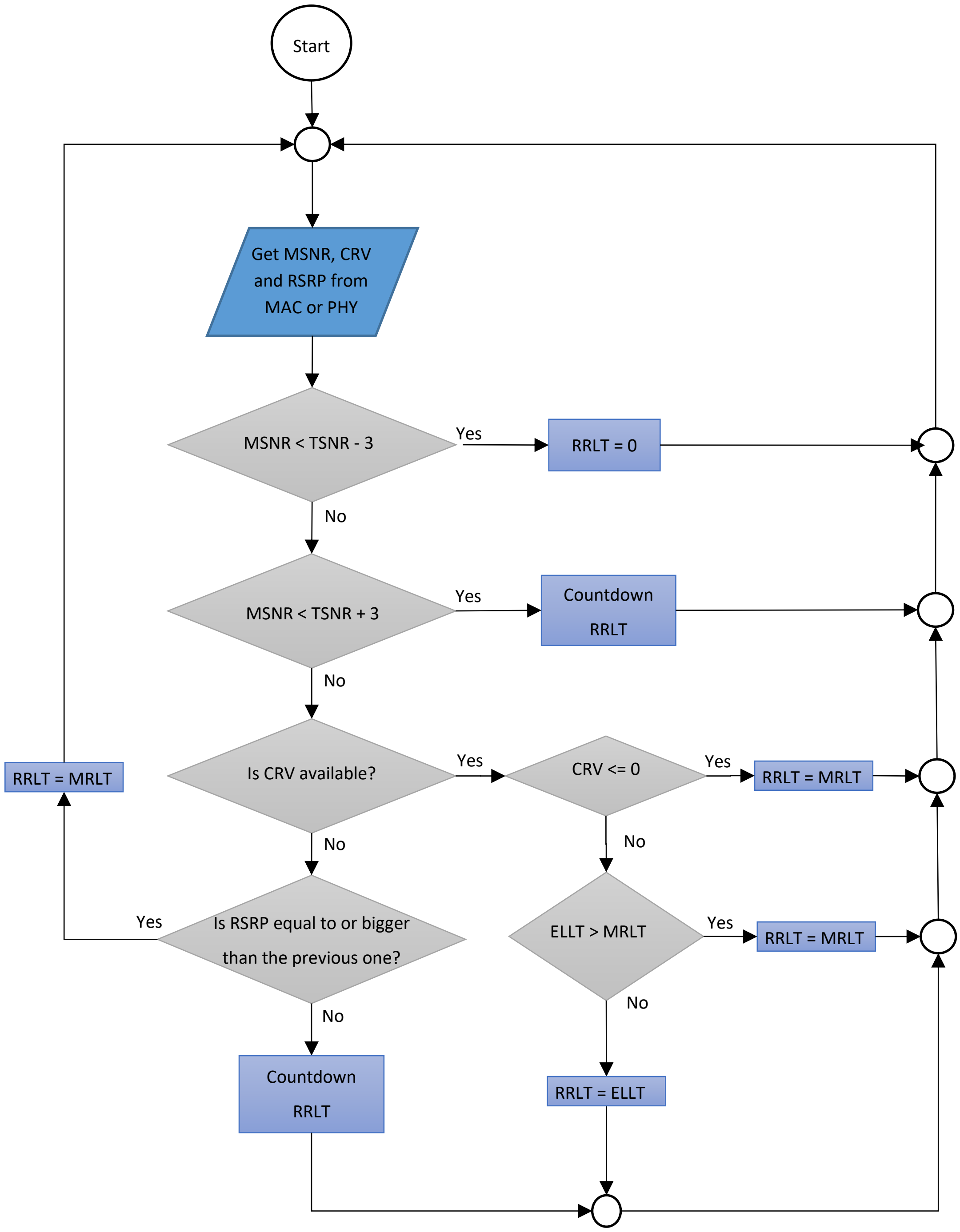

Figure 4.5 Flowchart of RLTDM

Where

- MSNR = Measured SNR (in dB)

- CRV = Calculated relative velocity wrt Doppler shift (in m/s) or RSRP changes
- TSNR = Target SNR for desired data rate (in dB)
- RRLT = Remained route life time (in s)
- MRLT = Maximum route life time (in s)
- ELLT is the estimated link life time, and can be calculated by using Link Margin (LM) and Estimated Distance with Respect to Received power (EDWR).

EDWR can be calculated from measured received power (RSSI or RSRP). By using measured received power, first we can observe the FSPL:

$$\text{FSPL} = RSRP - P_{Tx} - G_{Tx} - G_{Rx} \ \ (in\ dB)$$

After observing FSPL, EDWR (in km) can be calculated:

$$EDWR = 10^{\frac{fspl-32{,}44-20*log_{10}(f)}{20}}$$

LM is the difference between measured SNR and target SNR:

$$\text{LM} = \text{MSNR} - \text{TSNR}$$

Then, ELLT can be calculated by using EDWR, LM and CRV:

$$ELLT = \frac{EDWR * 1000 * \left(10^{\frac{LM}{20}} - 1\right)}{CRV}$$

4.4. AODV Simulation with RLTDM Method

In Chapter 3, the original AODV routing algorithm performances are simulated by using different topologies and applications. The list of this simulations are given below:

- Chapter 3.2.2.1: Stationary Case-1 (ping period 10 s)
- Chapter 3.2.2.2: Stationary Case-2 (ping period 2 s)
- Chapter 3.2.3.1: Simulation of the first part of the mission (Random Mobility)
    - Chapter 3.2.3.1.1: First Simulation (ping period 10 s)
    - Chapter 3.2.3.1.2: Second Simulation (ping period 2 s)
- Chapter 3.2.3.2 Simulation of the second part of the mission (Group Mobility)
    - Chapter 3.2.3.2.1: First Simulation (ping period 10 s)
    - Chapter 3.2.3.2.2: Second Simulation (ping period 2 s)

All of these simulations are executed by using the new RLTDM method with AODV routing algorithm.

4.4.1. Compared Parameters

The original AODV routing algorithm and improved AODV algorithm by using the new RLTDM are compared by using parameters given below:

- Latency: End to end delay of data transmission.
- Total number of routing overhead messages: The AODV algorithm uses some message types for routing establishment (RREQ, RREP) and maintenance (RERR, HELLO). This value is the total amount of sent overhead messages for the same number of data packets transmitted and used as a performance parameter.
- Successful data transmission ratio: During the simulations, specific amount of application data are send to the receiver. But, since there is not available route for all of simulation duration, some of the data packets cannot be delivered to the receiver. The ratio of the number of delivered data packets to all data packets is used as a performance parameter.

4.4.2. Simulations of Stationary cases with the new method:

As mentioned in Chapter 3, although AODV routing protocol parameters and topology of the network do not change, changes in the ping application properties affect the simulation results. By using this new method for stationary cases, it is possible to achieve better results.

The used simulation parameters are the same as those in Chapter 3 (Table 3.5 and Table 3.7). Simulation of Stationary Case-1 (ping period 10 s) was carried out. First, it was assumed that there was an independent beacon source for cross layer metrics used for new method. Results of this assumption is given in *Table 4.3* and *Table 4.4*.

Table 4.3 *Data Statistics of the new method with Stationary Case-1*

Number of Ping Request Sent	100
Number of Ping Request Received	100
Loss Rate of Ping Requests	% 0
Minimum Round-Trip Time (RTT)	2.787 ms
Average Round-Trip Time	3.00179ms
Maximum Round-Trip Time	24.266 ms
Standard Deviation of RTT	2.1479 ms
Variance of RTT	4.6e-6

Table 4.4 *AODV Message Statistics of of the new method with Stationary Case-1*

Module Name	Total sent RREP	Total sent RREQ
AODVNetwork.uav[0].aodv	0	1
AODVNetwork.uav[1].aodv	1	1
AODVNetwork.uav[2].aodv	1	1
AODVNetwork.uav[3].aodv	1	0
Total (6)	3	3

Then, it was assumed that Hello messages of basic AODV routing protocol were used for cross layer metrics calculation and these parameters used for new method for routing decision improvement. Results of this assumption is given in *Table 4.5* and *Table 4.6*.

Table 4.5 *Data Statistics of the new method with Stationary Case-1*

Number of Ping Request Sent	100
Number of Ping Request Received	100
Loss Rate of Ping Requests	% 0
Minimum Round-Trip Time (RTT)	2.787 ms
Average Round-Trip Time	2.95802 ms
Maximum Round-Trip Time	19.8893 ms
Standard Deviation of RTT	1.71023 ms
Variance of RTT	2.9e-6

Table 4.6 *AODV Message Statistics of of the new method with Stationary Case-1*

Module Name	**Total sent RREP**	**Total sent RREQ**	**Total Sent Hello**
AODVNetwork.uav[0].aodv	0	1	570
AODVNetwork.uav[1].aodv	1	1	571
AODVNetwork.uav[2].aodv	1	1	572
AODVNetwork.uav[3].aodv	1	0	571
Total (600)	3	3	2284

Comparison of results for basic and improved AODV routing protocols for stationary cases are given in Figure 4.6, Figure 4.7 and Figure 4.8.

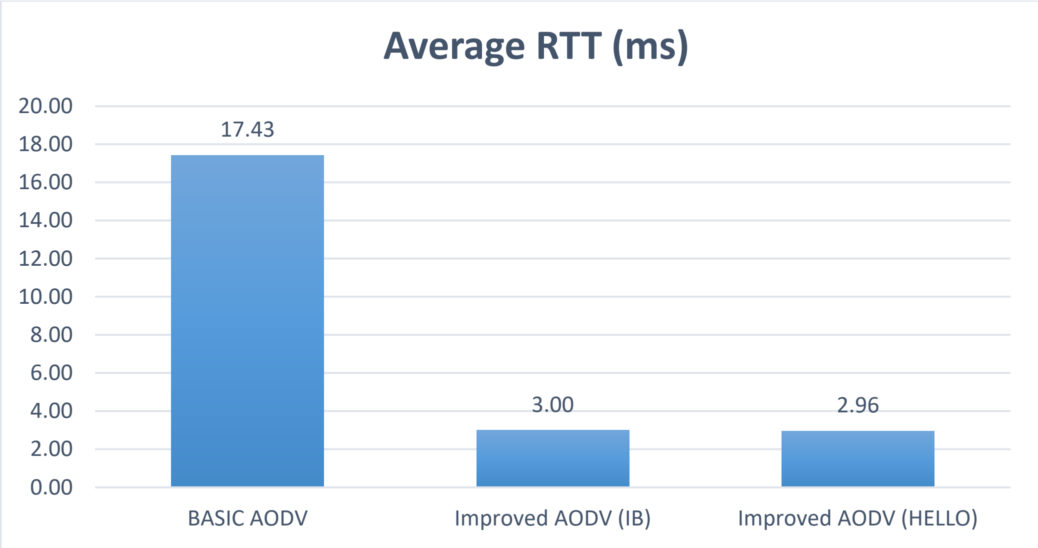

Figure 4.6 RTT Comparison

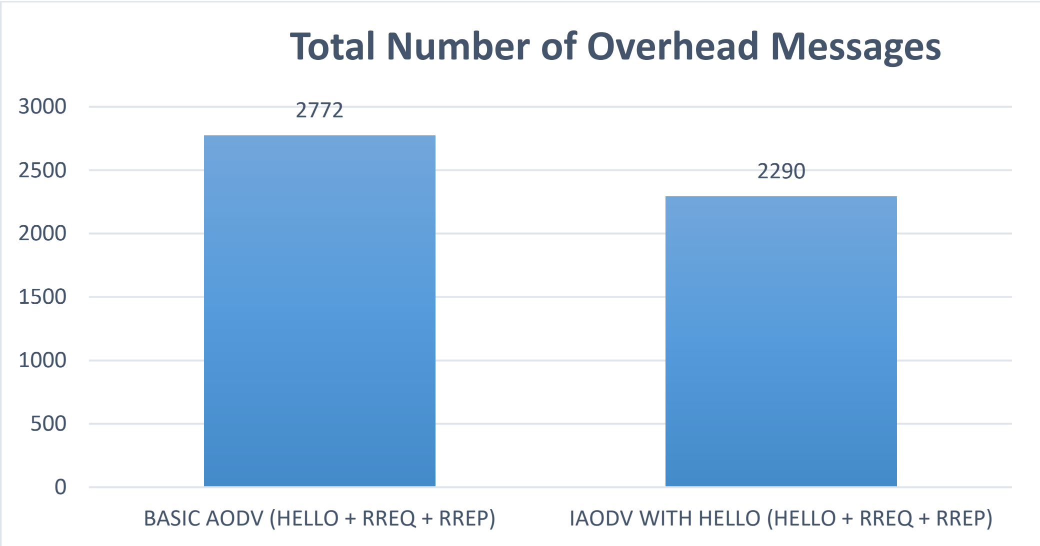

Figure 4.7 Comparison of Total Overhead Messages

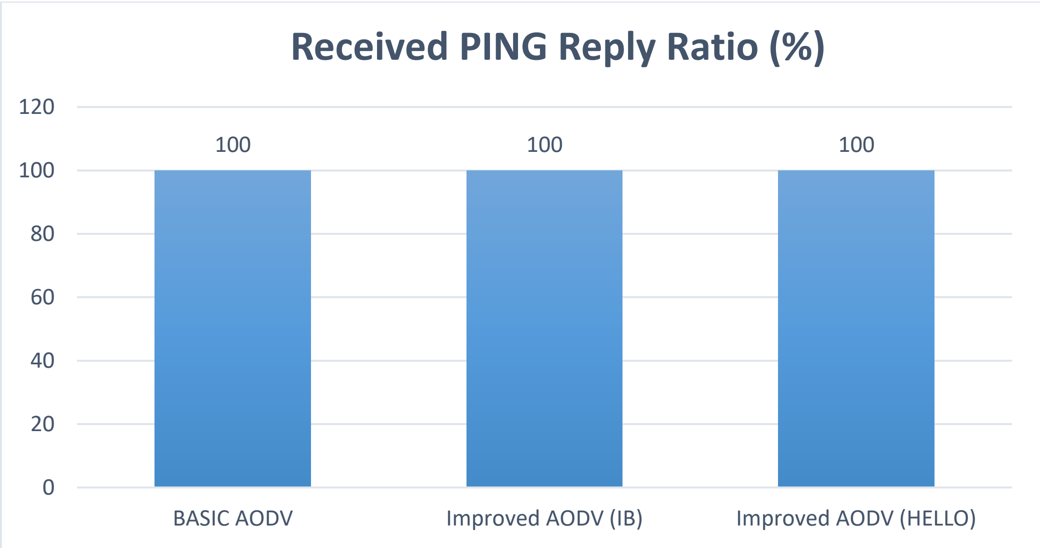

Figure 4.8 Comparison of Received PING Reply Ratio

These results show that the new method for AODV routing protocol performance is improved significantly for Stationary Case-1 (ping period 10 s). Round trip time is latency parameter, RTT comparison results shows that the new method results achieve 3 ms latency, on the other hand the basic one achieves 17 ms latency. Since the latest wireless communication technologies requires lower latencies this improvement can also be useful for new wireless technologies. One of the important performance parameter of routing algorithms is total number of transmitted overhead messages. Basic overhead messages of AODV routing protocols RREQ, RREP and Hello messages. Simulation results show that there is also a significant improvement for this performance metric. The new method (improved AODV) uses 2290 overhead messages, however the basic AODV uses 2772 over head messages. The total decrease ratio in number of overhead messages is about % 17.3. Since all nodes are stationary and wireless communication range is lower than the distance between consecutive nodes, received ping reply ratio is % 100 for all methods.

After completing the simulation of Stationary Case-1 (ping period 10 s), then simulation of Stationary Case-2 (ping period 2 s) was carried out. Simulation results are same as those in Chapter 3 and also same as those in Stationary Case-1 (ping period

2 s). These results show that when the new method is applied to simulation cases, changes in the application properties do not affect the simulation results.

4.4.3. Simulations with Linear Mobility :

As mentioned in Chapter 3, in the first part of the given scenario for mobile nodes, there are 20 UAVs flying on the disaster area with linear mobility model. Two simulations are done for this case. In the first simulation, ping period is 2 s. The other simulation parameters are same as those in Chapter 3. First, it was assumed that there was an independent beacon source for cross layer metrics used for new method. Results of this assumption is given in *Table 4.7* and *Table 4.8*.

Table 4.7 *Data Statistics of the new method with Linear Mobility*

Number of Ping Request Sent	100
Number of Ping Request Received	54
Loss Rate of Ping Requests	% 46
Minimum Round-Trip Time (RTT)	0.928 ms
Average Round-Trip Time	446.751 ms
Maximum Round-Trip Time	3152.11 ms
Standard Deviation of RTT	787.068 ms
Variance of RTT	0.619476

Table 4.8 *AODV Message Statistics of the new method with Linear Mobility*

Module	**Total sent RREP**	**Total sent RREQ**	**Total sent RRER**
AODVNetwork.uav[0].aodv	37	3	5
AODVNetwork.uav[1].aodv	2	101	21
AODVNetwork.uav[2].aodv	2	28	0
AODVNetwork.uav[3].aodv	2	30	2
AODVNetwork.uav[4].aodv	3	34	1
AODVNetwork.uav[5].aodv	4	41	3
AODVNetwork.uav[6].aodv	3	34	1
AODVNetwork.uav[7].aodv	12	44	4
AODVNetwork.uav[8].aodv	2	28	1
AODVNetwork.uav[9].aodv	1	29	0
AODVNetwork.uav[10].aodv	7	36	4
AODVNetwork.uav[11].aodv	2	39	0
AODVNetwork.uav[12].aodv	4	43	1
AODVNetwork.uav[13].aodv	12	30	4
AODVNetwork.uav[14].aodv	4	46	2
AODVNetwork.uav[15].aodv	7	43	4
AODVNetwork.uav[16].aodv	3	44	2
AODVNetwork.uav[17].aodv	3	36	0
AODVNetwork.uav[18].aodv	3	33	3
AODVNetwork.uav[19].aodv	4	43	2
Total (942)	117	765	60

Then, it was assumed that Hello messages of basic AODV routing protocol were used for cross layer metrics calculation and these parameters used for new method for routing decision improvement. Results of this assumption is given in *Table 4.9* and *Table 4.10*.

Table 4.9 *Data Statistics of the new method with Linear Mobility*

Number of Ping Request Sent	100
Number of Ping Request Received	52
Loss Rate of Ping Requests	% 48
Minimum Round-Trip Time (RTT)	0.928 ms
Average Round-Trip Time	402.02 ms
Maximum Round-Trip Time	4804.76 ms
Standard Deviation of RTT	856.054 ms
Variance of RTT	0.732828

Table 4.10 *AODV Message Statistics of the new method with Linear Mobility*

Module	Total sent HELLO	Total sent RREP	Total sent RREQ	Total sent RRER
AODVNetwork.uav[0].aodv	109	27	4	4
AODVNetwork.uav[1].aodv	82	1	103	25
AODVNetwork.uav[2].aodv	105	3	26	2
AODVNetwork.uav[3].aodv	99	4	35	1
AODVNetwork.uav[4].aodv	100	3	35	1
AODVNetwork.uav[5].aodv	100	4	33	2
AODVNetwork.uav[6].aodv	99	3	43	1
AODVNetwork.uav[7].aodv	94	10	44	2
AODVNetwork.uav[8].aodv	105	4	29	2
AODVNetwork.uav[9].aodv	100	6	29	1
AODVNetwork.uav[10].aodv	97	3	38	2
AODVNetwork.uav[11].aodv	101	3	38	0
AODVNetwork.uav[12].aodv	98	9	38	2
AODVNetwork.uav[13].aodv	102	9	22	1
AODVNetwork.uav[14].aodv	96	12	43	4
AODVNetwork.uav[15].aodv	99	3	38	1

Table 4.10 (Cont'd)

AODVNetwork.uav[16].aodv	97	12	40	5
AODVNetwork.uav[17].aodv	99	5	37	0
AODVNetwork.uav[18].aodv	103	6	35	0
AODVNetwork.uav[19].aodv	94	4	42	0
Total(2862)	1979	131	752	56

Comparison of results for basic and improved AODV routing protocols for stationary cases are given in Figure 4.9, Figure 4.10 and Figure 4.11.

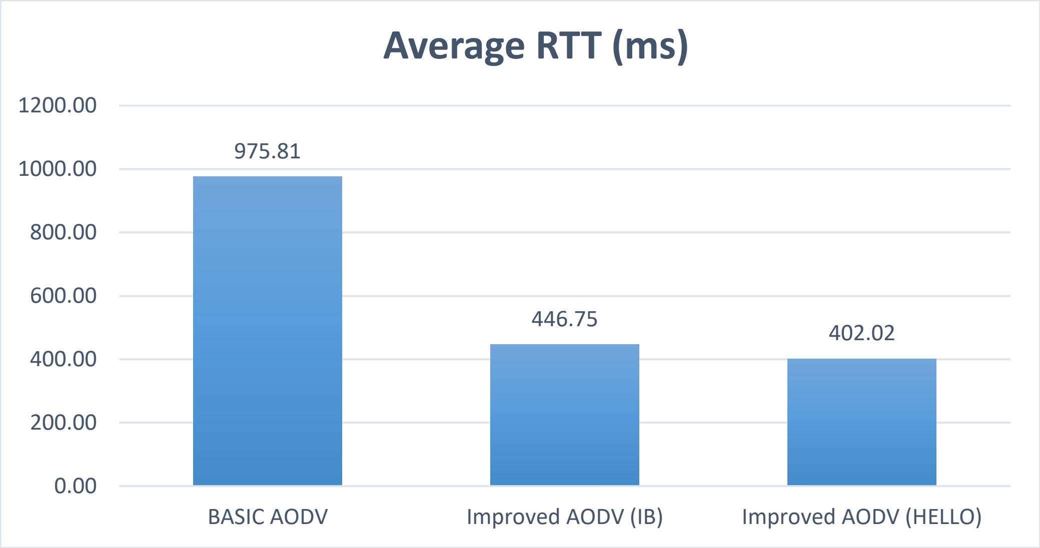

Figure 4.9 RTT Comparison

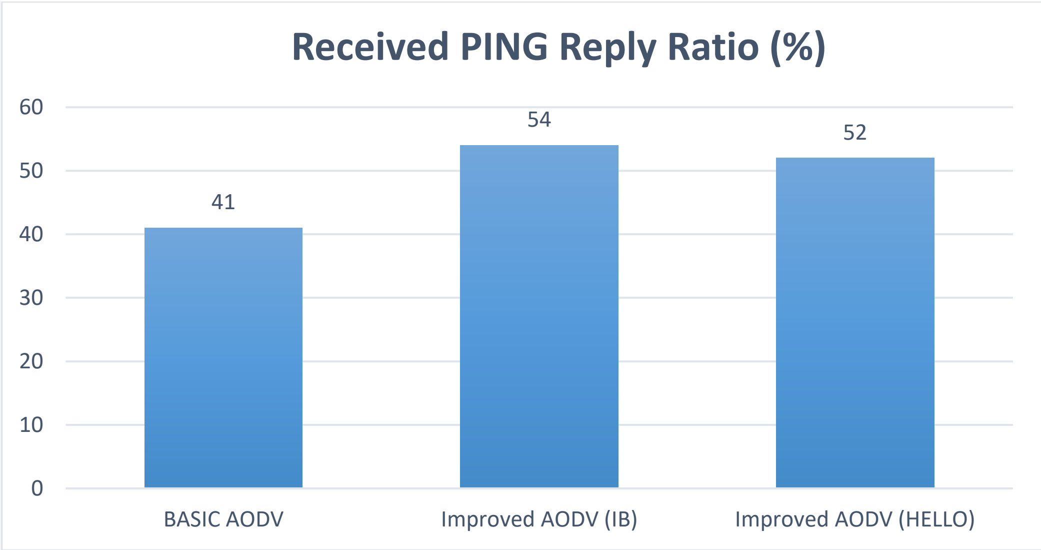

Figure 4.10 Comparison of Received PING Reply Ratio

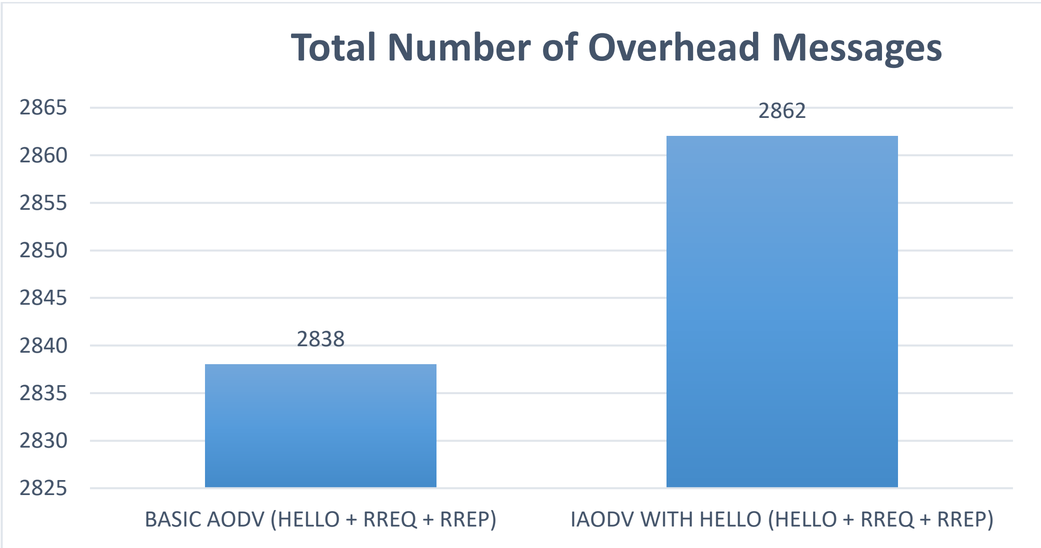

Figure 4.11 Comparison of Total Overhead Messages

These results show that the new method for AODV routing protocol performance is improved significantly for Linear Mobility (ping period 2 s). RTT comparison results shows that the new method results achieve about 400 ms average latency, on the other hand the basic one achieves about 1000 ms average latency. One of the important performance parameter of routing algorithms is received ping reply ratio. Simulation results show that there is also a significant improvement for this performance metric. The new method (improved AODV) achieves about %50 ratio, however the basic AODV achieves about %40 ratio. The total increase ratio in received ping reply ratio is about % 20. The other important parameter is total number of overhead messages of AODV protocol. This performance metric is almost same both basic and new methods.

After completing the simulation of Linear Mobility Case-1 (ping period 10 s), then simulation of Linear Mobility Case-2 (ping period 2 s) was carried out. Simulation results are same as those in Linear Mobility Case-2 (ping period 2 s). These results show that when the new method is applied to simulation cases, routing protocol performance are improved significantly.

4.4.3.1. Simulations with various speeds:

In this simulation cases, performance of the proposed method is compared with the original one for various nodes speeds. Number of nodes is 20, simulations are done for 10, 20, 30, 40 50 mps speeds. Linear Mobility (LM) model is used with random initial positions and directions for each simulation. Results of the simulations are given in Figure 4.12, Figure 4.13 and Figure 4.14.

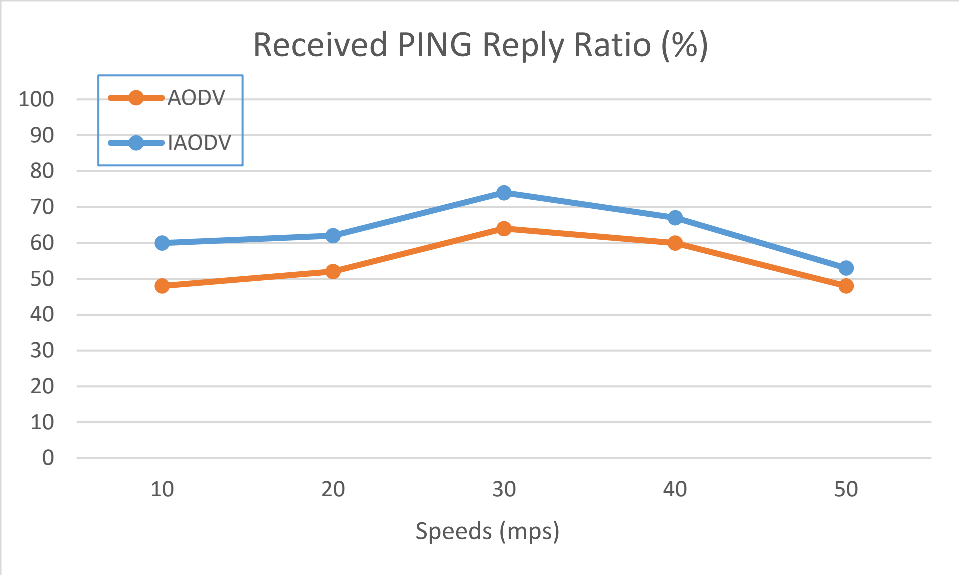

Figure 4.12 Comparison of Received PING Reply Ratio

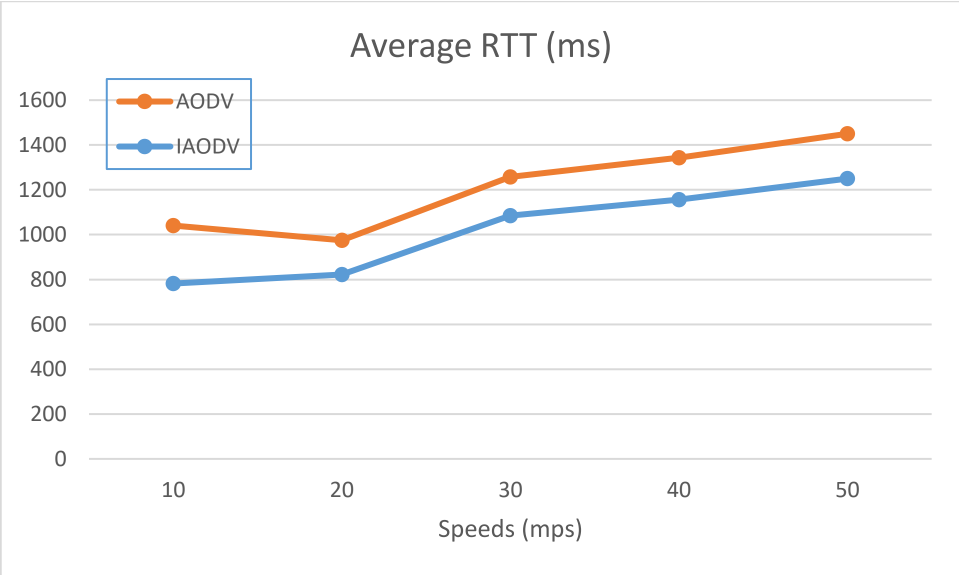

Figure 4.13 RTT Comparison

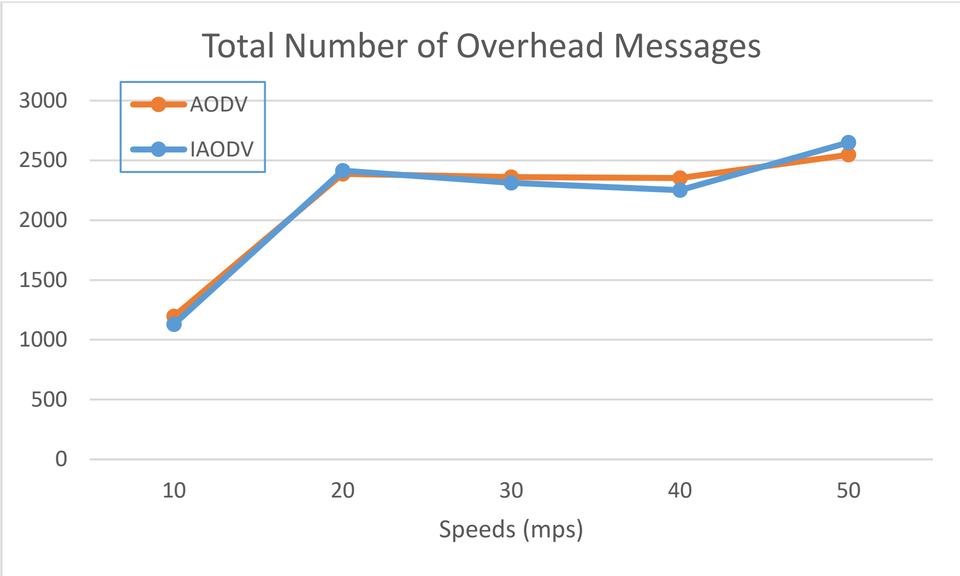

Figure 4.14 Comparison of Total Overhead Messages

Simulation results show that the proposed method improves Received PING Reply Ratio and Latency (RTT) performance metrics significantly for lower speeds (10 and 20 mps). When the nodes speed increases, although performance of the proposed method is degraded, there are also significant performance improvements for Received PING Reply Ratio and Latency performance metrics. The other performance metric result, Total Overhead Messages, is almost same both basic and new methods for each nodes speed.

4.4.4. Simulations with Group Mobility :

As mentioned in Chapter 3, in the second part of the given scenario for mobile nodes, there are 4 group of UAVs and each group has 5 UAVs. The simulation area is also divided into 4 areas with the same size. Each group is flying on one of the divided areas. Two simulations are done for this case. In the first simulation, ping period is 10 s. The other simulation parameters are same as those in Chapter 3. First, it was assumed that there was an independent beacon source for cross layer metrics used for new method. Results of this assumption is given in *Table 4.11* and *Table 4.12*.

Table 4.11 *Data Statistics of the new method with Group Mobility*

UAV No	Loss Rate of Ping Requests	Minimum RTT (ms)	Average RTT (ms)	Maximum RTT (ms)
UAV[1]	%0	1.86	4.36	252.26
UAV[2]	%14	244.10	359.42	2564.18
UAV[3]	%4	1.86	83.39	1693.21
UAV[4]	%0	1.86	1.86	1.86
UAV[6]	%0	1.86	4.38	253.62
UAV[7]	%14	244.71	368.55	1605.82
UAV[8]	%1	1.86	101.85	1602.64
UAV[9]	%0	1.86	1.86	1.86
UAV[11]	%3	1.86	21.46	662.90
UAV[12]	%17	244.91	349.09	1004.58
UAV[13]	%4	1.86	74.71	1245.01
UAV[14]	%3	1.86	39.09	1605.85
UAV[16]	%3	1.86	57.06	893.45
UAV[17]	%16	244.85	418.24	1609.42
UAV[18]	%8	1.86	94.39	1607.59
UAV[19]	%3	1.86	40.26	1246.48
Average	**%6**	**62.55**	**126.25**	**1115.67**

Table 4.12 *AODV Message Statistics of new method with Group Mobility*

Module	Total sent RREP	Total sent RREQ	Total sent RRER
AODVNetwork.uav[0].aodv	50	153	5
AODVNetwork.uav[1].aodv	65	152	11
AODVNetwork.uav[2].aodv	85	139	25
AODVNetwork.uav[3].aodv	71	155	24
AODVNetwork.uav[4].aodv	58	148	17
AODVNetwork.uav[5].aodv	48	151	2
AODVNetwork.uav[6].aodv	56	149	7
AODVNetwork.uav[7].aodv	71	148	21
AODVNetwork.uav[8].aodv	81	139	24
AODVNetwork.uav[9].aodv	56	147	11
AODVNetwork.uav[10].aodv	67	176	7
AODVNetwork.uav[11].aodv	76	177	13
AODVNetwork.uav[12].aodv	83	172	24
AODVNetwork.uav[13].aodv	91	174	30
AODVNetwork.uav[14].aodv	73	174	23
AODVNetwork.uav[15].aodv	81	203	15
AODVNetwork.uav[16].aodv	69	206	19
AODVNetwork.uav[17].aodv	87	213	35
AODVNetwork.uav[18].aodv	113	184	37
AODVNetwork.uav[19].aodv	93	203	21
Total (5208)	**1474**	**3363**	**371**

Then, it was assumed that Hello messages of basic AODV routing protocol were used for cross layer metrics calculation and these parameters used for new method for routing decision improvement. Results of this assumption is given in *Table 4.13* and *Table 4.14*.

Table 4.13 *Data Statistics of the new method with Group Mobility*

UAV No	Loss Rate of Ping Requests	Minimum RTT (ms)	Average RTT (ms)	Maximum RTT (ms)
UAV[1]	%0	1.86	16.90	1248.65
UAV[2]	%5	1.86	193.43	2574.51
UAV[3]	%4	1.86	442.61	3056.08
UAV[4]	%1	1.86	33.80	2818.41
UAV[6]	%1	1.86	17.02	1248.51
UAV[7]	%7	2.84	225.29	2565.44
UAV[8]	%8	1.86	448.50	2523.12
UAV[9]	%1	1.86	29.49	647.81
UAV[11]	%2	1.86	73.14	1614.45
UAV[12]	%8	2.52	175.11	2358.10
UAV[13]	%10	1.86	296.74	1590.19
UAV[14]	%9	1.86	41.06	1611.22
UAV[16]	%0	1.86	22.42	894.22
UAV[17]	%12	2.54	167.19	2253.58
UAV[18]	%6	1.86	228.44	1609.30
UAV[19]	%2	1.86	16.07	646.38
Average	**%5**	**2.00**	**151.70**	**1828.75**

Table 4.14 *AODV Message Statistics of new method with Group Mobility*

Module	Total Sent Hello	Total sent RREP	Total sent RREQ	Total sent RRER
AODVNetwork.uav[0].aodv	529	176	109	4
AODVNetwork.uav[1].aodv	495	173	114	139
AODVNetwork.uav[2].aodv	499	79	203	36
AODVNetwork.uav[3].aodv	496	41	222	41
AODVNetwork.uav[4].aodv	497	175	111	142
AODVNetwork.uav[5].aodv	529	163	130	6
AODVNetwork.uav[6].aodv	500	153	135	117
AODVNetwork.uav[7].aodv	497	102	212	47
AODVNetwork.uav[8].aodv	505	44	200	52
AODVNetwork.uav[9].aodv	488	156	138	123
AODVNetwork.uav[10].aodv	528	182	139	15
AODVNetwork.uav[11].aodv	486	171	155	131
AODVNetwork.uav[12].aodv	497	89	214	58
AODVNetwork.uav[13].aodv	502	73	230	46
AODVNetwork.uav[14].aodv	497	170	146	130
AODVNetwork.uav[15].aodv	530	190	116	5
AODVNetwork.uav[16].aodv	500	179	120	145
AODVNetwork.uav[17].aodv	516	66	204	42
AODVNetwork.uav[18].aodv	495	63	199	47
AODVNetwork.uav[19].aodv	506	178	124	147
Total (17409)	**10092**	**2623**	**3221**	**1473**

Comparison of results for basic and improved AODV routing protocols for stationary cases are given in Figure 4.15, Figure 4.16 and Figure 4.17

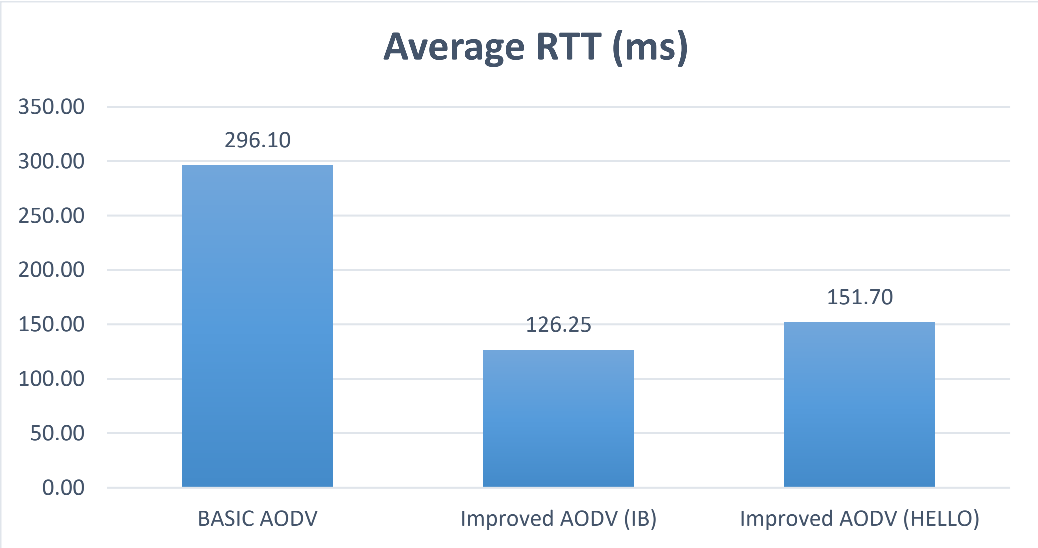

Figure 4.15 RTT Comparison

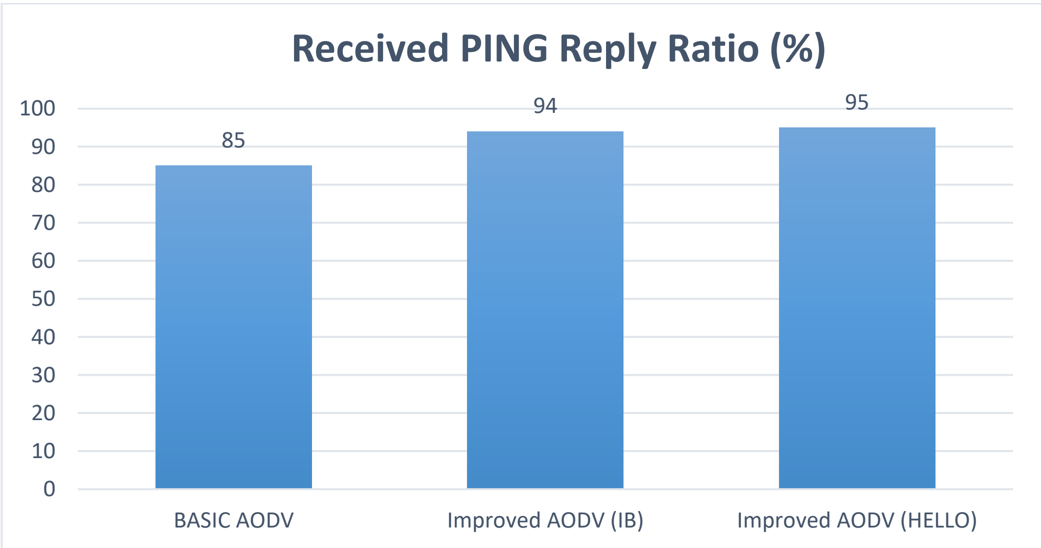

Figure 4.16 Comparison of Received PING Reply Ratio

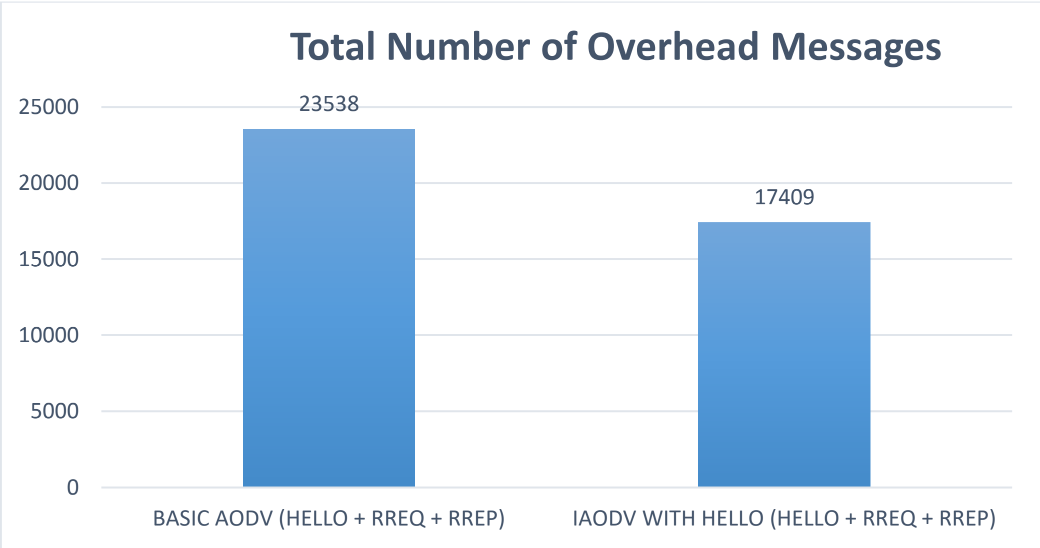

Figure 4.17 Comparison of Total Overhead Messages

These results show that the new method for AODV routing protocol performance is improved significantly for Group Mobility (ping period 10 s). RTT comparison results shows that the new method results achieve about 150 ms average latency, on the other hand the basic one achieves about 300 ms average latency. One of the important performance parameter of routing algorithms is received ping reply ratio. Simulation results show that there is also a significant improvement for this performance metric. The new method (improved AODV) achieves about %95ratio, however the basic AODV achieves about %85 ratio. The total increase ratio in received ping reply ratio is about % 12. The other important parameter is total number of overhead messages of AODV protocol. There is also a significant improvement for this performance metric. The new method (improved AODV) uses 23500 overhead messages, however the basic AODV uses 17400 over head messages. The total decrease ratio in number of overhead messages is about % 26.

After completing the simulation of Group Mobility Case-1 (ping period 10 s), then simulation of Group Mobility Case-2 (ping period 2 s) was carried out. Simulation results are same as those in Group Mobility Case-2 (ping period 2 s). These results show that when the new method is applied to simulation cases, routing protocol performances are improved significantly.

4.4.5. Simulations with Random Waypoint Mobility Model :

In this simulation cases, performance of the proposed method is compared with the original one for Random Waypoint Mobility model with various nodes speeds. Number of nodes is 20, simulations are done for 10, 20, 30, 40 50 mps speeds. Results of the simulations are given in Figure 4.18, Figure 4.19 and Figure 4.20.

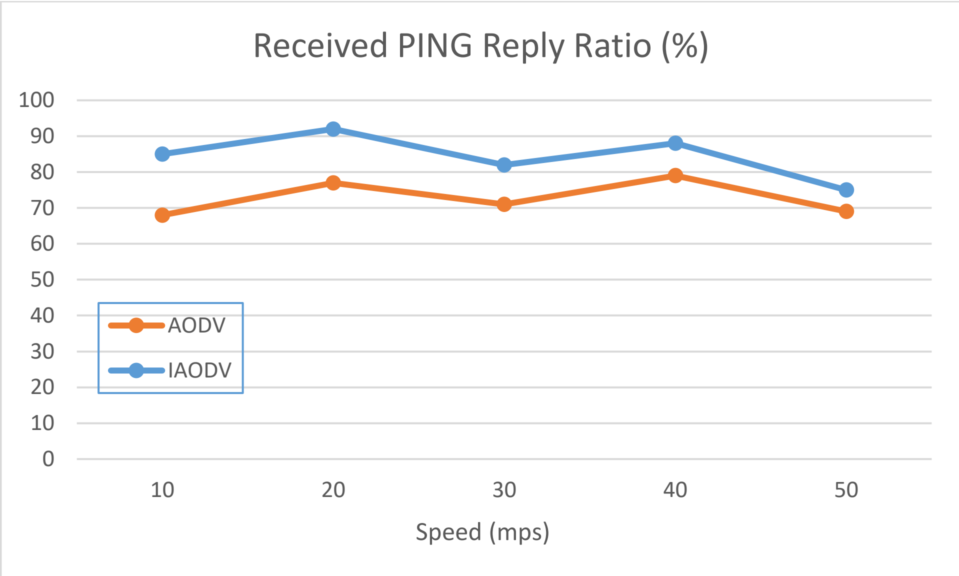

Figure 4.18 Comparison of Received PING Reply Ratio

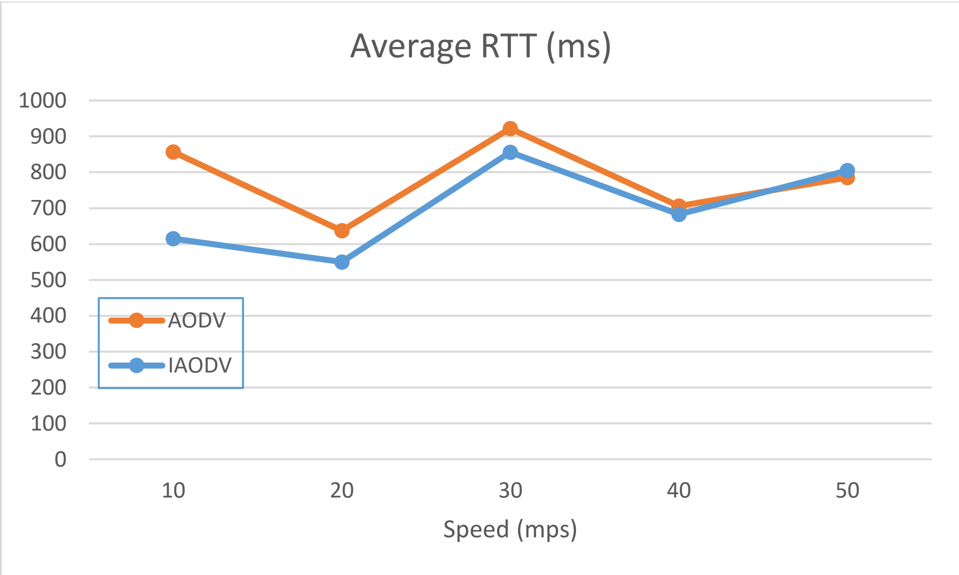

Figure 4.19 RTT Comparison

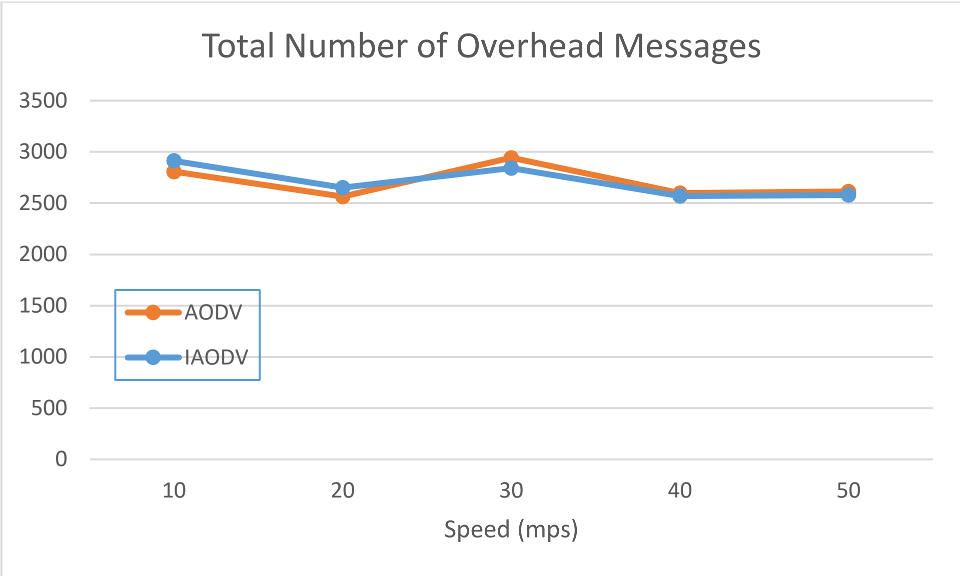

Figure 4.20 Comparison of Total Overhead Messages

Simulation results show that the proposed method improves Received PING Reply Ratio and Latency (RTT) performance metrics significantly for lower speeds (10 and 20 mps). When the nodes speed increases, although performance of the proposed method is degraded, there are also significant performance improvements for Received PING Reply Ratio and Latency performance metrics. The other performance metric result, Total Overhead Messages, is almost same both basic and new methods for each nodes speed.

4.4.6. Simulations with Gauss-MarkovMobility Model :

In this simulation cases, performance of the proposed method is compared with the original one for Gauss-Markov Mobility model with various nodes speeds. Number of nodes is 20, simulations are done for 10, 20, 30, 40 50 mps speeds. Results of the simulations are given in Figure 4.21, Figure 4.22 and Figure 4.23.

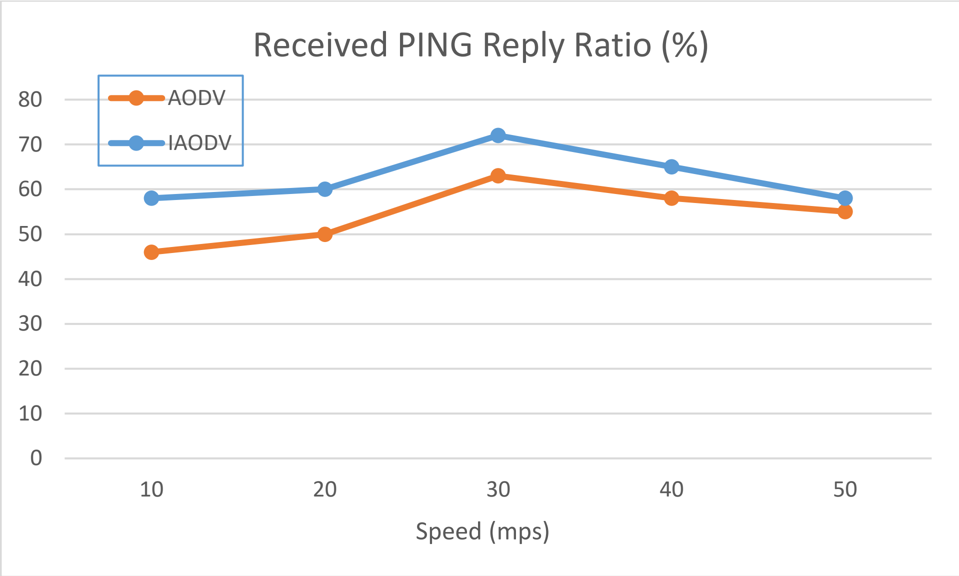

Figure 4.21 Comparison of Received PING Reply Ratio

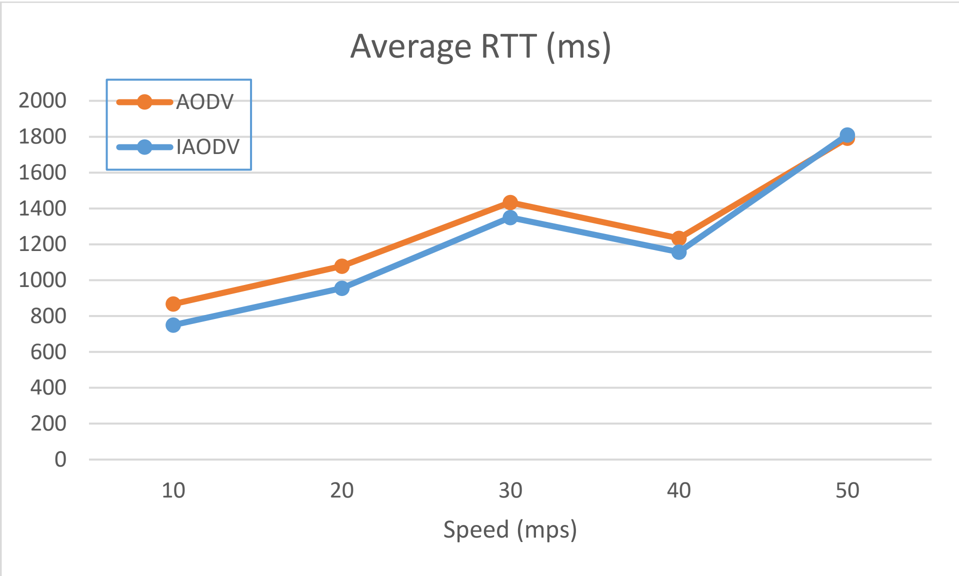

Figure 4.22 RTT Comparison

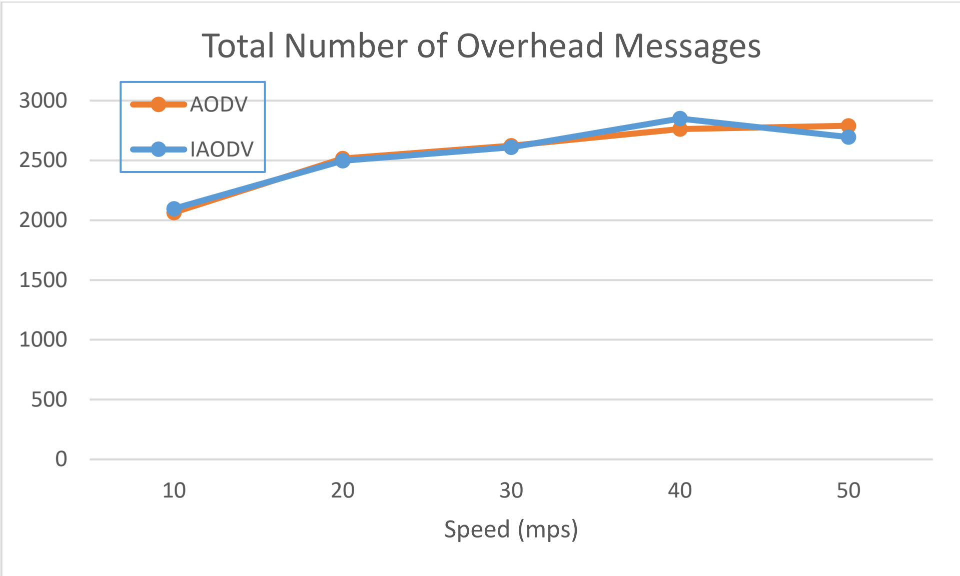

Figure 4.23 Comparison of Total Overhead Messages

Simulation results show that the proposed method improves Received PING Reply Ratio and Latency (RTT) performance metrics significantly for lower speeds (10 and 20 mps). When the nodes speed increases, although performance of the proposed method is degraded, there are also significant performance improvements for Received PING Reply Ratio and Latency performance metrics. The other performance metric result, Total Overhead Messages, is almost same both basic and new methods for each nodes speed.

4.4.7. Simulations for only RSRP usage:

In this simulation case, first performances of the original AODV routing algorithm and the proposed method are investigated for

- Stationary
- Group
- Linear
- Random Waypoint
- Gauss-Markov

mobility models.

Stationary and Group mobility simulations for original AODV algorithm and the proposed method are not done again, simulation results in the Chapter 3 and the Chapter 4 are used for Stationary and Group mobility cases.

If the proposed method given in this chapter is analyzed, it can be seen that the proposed method uses RSRP metric for routing decision when the Doppler Measurement is not available (Look at the ” Figure 4.5 Flowchart of the proposed method”). So, examining the performance of the only RSRP usage is valuable. So, first, simulations are done for only RSRP usage with Stationary and Group mobility models. Then simulations of other mobility models are done for “original AODV”, ”the proposed new method with Doppler and RSRP metrics usage”, and “the proposed method with only RSRP usage” respectively.

During the simulations, it is assumed that there are 20 mobile nodes and speed of each node changes between 10 and 50 mps with uniformly.

Results of the simulations are given in Figure 4.24, Figure 4.25 and Figure 4.26.

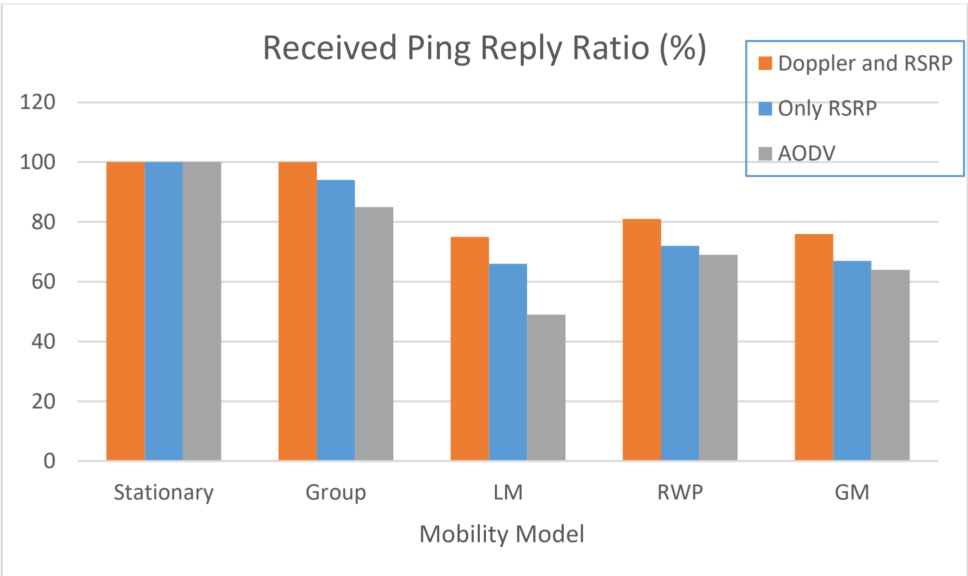

Figure 4.24 Comparison of Received PING Reply Ratio

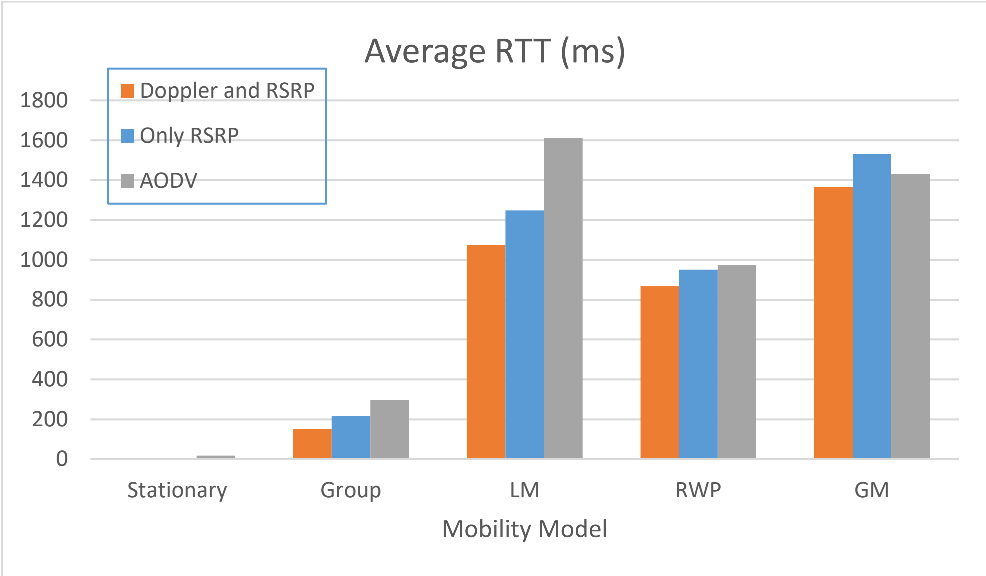

Figure 4.25 RTT Comparison

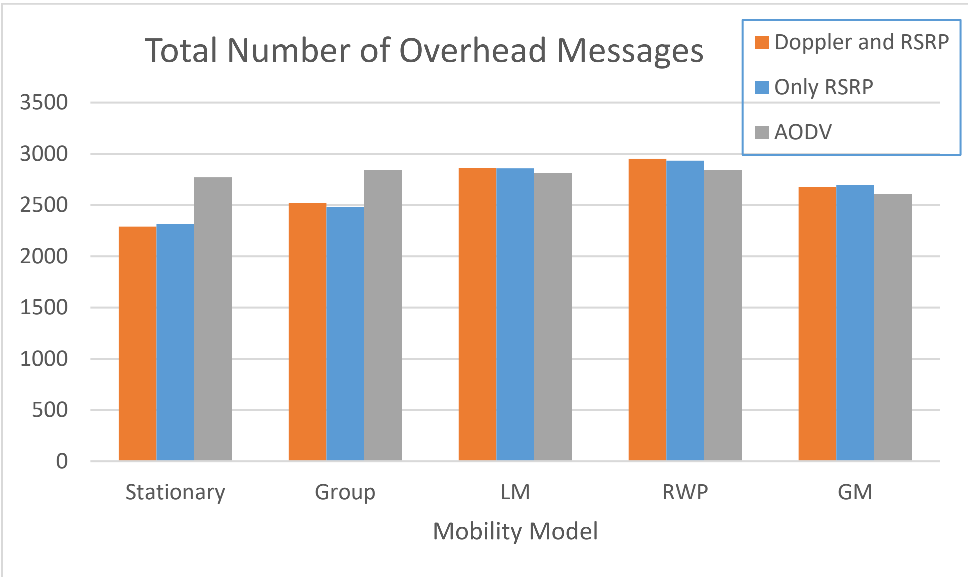

Figure 4.26 Comparison of Total Overhead Messages

Simulation results show that when performances of "Doppler and RSRP usage" and "only RSRP usage" are compared, it can be seen that there are not any significant performance changes for Stationary and Group mobility cases. Also it can be seen that when only the RSRP metric used, Received PING Reply Ratio and Latency (RTT) performance metrics are degraded for other mobility models. The other performance metric result, Total Number of Overhead Messages, is almost same both “Doppler and RSRP usage” and “only RSRP usage” for each mobility model.

4.4.8. Simulations for RSRP and RSSI usage comparison:

It is obvious that if there is no interference and jammer signals, usage of RSSI or RSRP gives same results. So, in these simulation cases, it is assumed that there is an interference or jammer signal, which degrades the performance of the wireless connection. Since the RSSI measurement is done in the analog front end of the receiver, the measured RSSI is increased due to interfence or jammer signal existence. Then the method, which uses RSSI metric, gives wrong decisions, and performance of the routing algorithm is degraded. On the other hand, the proposed method uses digital RSRP value, obtained by pilot signals measurement, and performance of the proposed method does not change significantly.

All simulations are done for “only RSRP usage” and “only RSSI usage” with Linear,vRandpm Waypoint and Gauss-Markov mobility cases. During the simulations, it is assumed that there are 20 mobile nodes and speed of each node changes between 10 and 50 mps with uniformly.

Results of the simulations are given in Figure 4.27, Figure 4.28 and Figure 4.29.

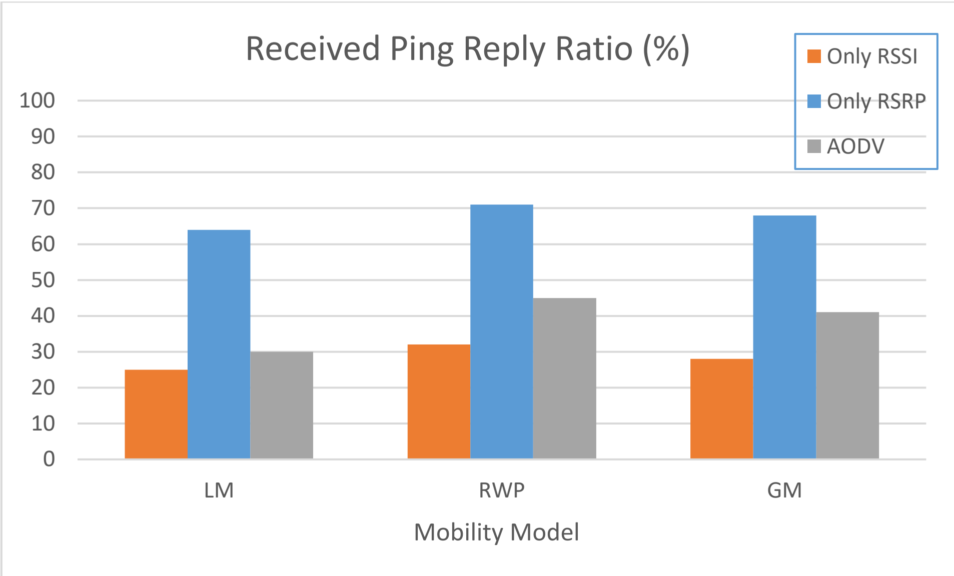

Figure 4.27 Comparison of Received PING Reply Ratio

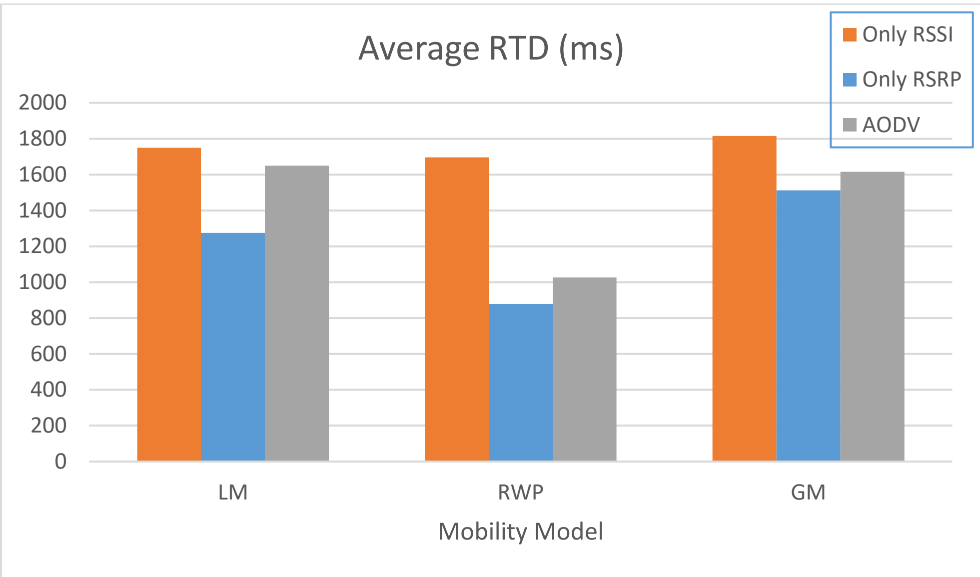

Figure 4.28 RTT Comparison

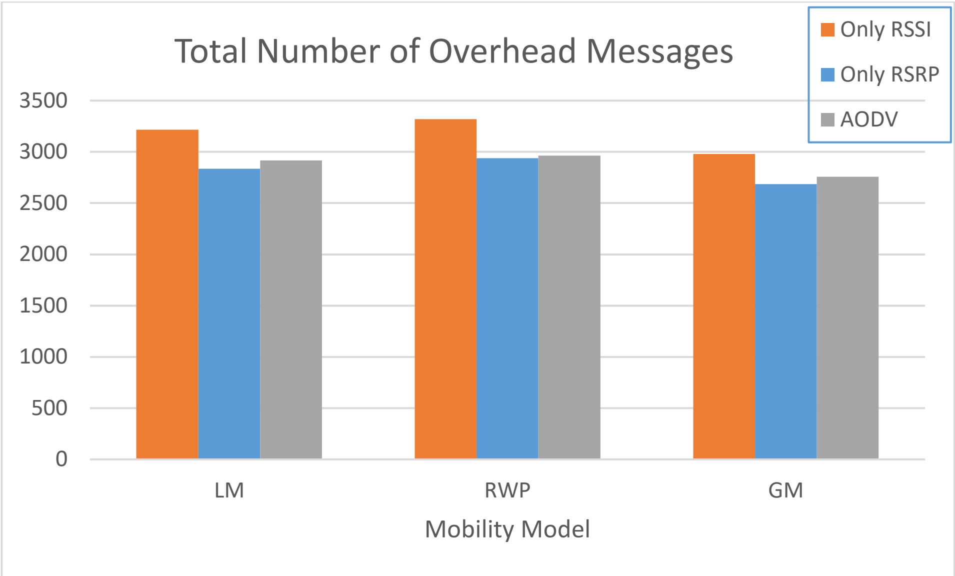

Figure 4.29 Comparison of Total Overhead Messages

Simulations results show that if there is an interference or jammer signal, performance of “only RSSI usage” is worse than the original AODV algorithm. On the other hand, performance of the “only RSRP usage” do not change and is better than the original AODV algorithm during the interference or jammer signal existence.

4.5. Comparison of the proposed method and other literature studies

In this study SNR, Doppler Shift and RSRP cross layer metrics are used for routing performance improvement. As it mentioned in “Chapter 2 Related Work“, when other studies in the literature are investigated, there are some studies which uses only SNR ([22]), SNR and RSSI ([23]), only Doppler Shift ([31]), Doppler Shift and RSSI ([34]), RSSI and GPS information ([35]). It is obvious that RSRP metric is not used in the literature. On the other hand feasibility of Doppler Shift measurement is not investigated in these studies. Comparison of the proposed method and other literature studies are given in Table 4.15 and Table 4.16.

Table 4.15 *Comparison for Used Metrics and Mobilty Cases*

	Cross Layer Metrics	**Performance Metrics**	**Mobility Cases**
IAODV (proposed method)	• SNR • Doppler Shift • RSRP	• Latency • Data Tx Ratio • Overhead Ratio	• Stationary • Group Mobility • Linear (0 – 50 m/s) • RWP (10 - 50 m/s) • GM (10-50 m/s)
AeroRouter ([31])	• Link Cost • Doppler Shift	Total number of handoff	• Predefined linear fashion
QoS Multipath Doppler Routing ([32], [33])	• QoS Cost (delay and attenuation) • Doppler Shift	Total number of handoff	• pseudo linear mobility (840 km/h velocity)
Link Expiration Time (LET) ([34])	• Received Power (RSSI) • Doppler Shift	• Total Number of Control Packets	• Predefined linear fashion (240 – 720 km/h)
MAR ([35])	• Received Power (RSSI) • GPS information	• Latency • Data Tx Ratio • Overhead Ratio	• RWP (1 m/s to 50 m/s speeds)
NTAR ([36])	• Doppler Shift • Traffic Load	• Latency • Data Tx Ratio • Overhead Ratio	• RWP (100 m/s to 800 m/s speeds)
SNR-based DSVD[([22]]	• SNR	• Data Tx Ratio • Latency	• Stationary
SNR/RP based ([23])	• SNR • Received Power (RSSI)	• Data Tx Ratio • Latency • Overhead Ratio	• RWP (3, 6, 9 and 12 m/s)

Table 4.16 Comparison for Advantages and Disadvantages

	Advantages	**Disadvantages**
IAODV (proposed method)	• SNR is used as control parameter • More immune to interference signals (instead of RSSI, RSRP is used) • Less latency and overhead ratios • Improvement for successful data transmission ratio and latency performance metrics • Excellent improvements for stationary and group mobility cases • Significant improvements for LM, RWP and GM mobility models • GPS independent routing	• Processing power requirements shall be investigated for commercial drones • More physical space will be needed for precise oscillators • If AODV Hello messages are not used, it needs another beacon subsystem • Slightly worse overhead performance for LM, RWP and GM mobility models • Route establishment is done as in the traditional one.
AeroRouter ([31])	• Reduced number of handoffs for multipath routing	• No detailed information for link cost • Not suitable for ad hoc networks • No information about nodes velocity and wireless communication frequency • Feasibility of doppler shift measurement not explained
QoS Multipath Doppler Routing ([32], [33])	• Reduced number of handoffs for multipath routing • Qos Cost and Doppler Shift Cost are compared	• Mobility, velocity and wireless communication ranges (200 – 600 km) are not applicable to UFANETs • Other performance metrics (latency, overhead ratio) are not investigated

Table 4.16 (cont'd)

Link Expiration Time (LET) ([34])	• Reduced number of total number of control packets	• Usage of Beacon (Hello) message for Doppler shift and received power calculations are not mentioned • No detailed information for simulation case(e.g. wireless communication range and frequency) • Mobility model is not applicable to UFANETs • Other performance metrics (like latency, successful data transmission ratio) are not investigated. • Feasibility of doppler shift measurement not explained
MAR ([35])	• Improvement for packet delivery ratio, overhead ratio and latency	• Not enough performance at the existence of interference signal • Disadvantages of GPS signal usage • No detailed information for LET calculation • Since analog power measurement (RSSI) is used, it is not immune to interference or jamming signals
SNR-based DSVD ([22]]	• Improvement for packet delivery ratio and latency	• Overhead performance metric is not investigated • Mobility cases are not investigated • Not enough information for SNR usage

Table 4.16 (cont'd)

NTAR ([36])	• Improvement for packet delivery ratio	• Latency performance is worse than the AODV • Performance of overhead ratio is not investigated • Applicability of this method for UFANETs is unknown (nodes speed > 100m/s) • Feasibility of doppler shift measurement not explained • No route maintenance improvement
SNR/RP based ([23])	• Improvement for packet delivery ratio and latency	• Overhead ratio performance is worse than the AODV • No route maintenance improvement • Since analog power measurement (RSSI) is used, it is not immune to interference or jamming signals • Speeds bigger than 12 m/s are not investigated

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this thesis, routing protocols for UFANETs is investigated for cross layer parameters. First, differences between MANETs and UFANETs are explained, then recent researches for UFANETs routing protocols are reviewed. After examining these studies, it has been seen that most of these researches are using traditional routing protocols. There are also a few studies which are using cross layer metrics for routing decisions, but these researches are interested in MANETs, and do not consider differences between MANETs and UFANETs. These studies also use basic cross layer metrics (like SNR, RSSI, etc.) and do not consider new physical layer properties of latest hardware and software technologies (Wi-Fi and 4G). In this study, all implementations and simulations consider the differences between UFANETs and MANETs, and new cross layer metrics (frequency offset – Doppler and RSRP) are used as well as traditional ones (SNR and RSSI).

The most popular on demand ad-hoc routing protocol is AODV, and this routing protocol is chosen for basic UFANETs routing algorithm. First, some operational scenarios are formed, and basic AODV routing protocol performances (successful data transfer ratio, latency of data transmission and total count of AODV protocol messages) are observed by using simulations of these scenarios with INET Framework of OMNET++. From the results, it can be seen that when the application properties (like ping period) change, performance of the basic AODV routing protocol changes dramatically with same routing control parameters. Then, a new routing life time decision method is proposed for UFANETs routing protocols. This novel approach uses new and traditional cross layer metrics of physical layer, and this method is used for improvement of AODV routing protocol performance. Same operational scenarios are used for simulations, all results are also observed by using INET Framework of OMNET++. Results of the proposed approach and the traditional one are compared. This comparison shows that the new besides traditional cross layer parameters can be used for improvement of routing protocol decisions. This method has significant improvements on stationary and group mobility scenarios, and also have little improvement for random mobility scenarios of UFANETs. It can be seen that the proposed approach improves the routing performances in terms of lower latencies, higher successful data transmission ratio and lower total count of AODV protocol messages.

It is obvious that most of the latest communication systems use wireless communication instead of wired one, so the wireless communication ratio increases day by day. As a result, the frequency spectrum of ISM bands is getting more crowded, and wireless systems interfere with each other. On the other hand, it is possible that someone can try to block wireless communication by using jamming signals. In case of existing jamming and interfering signals, the routing decision methods using physical layer can give wrong decisions and performance of these methods can be worse than the basic ones. In this study, SNR of the received digital signal is used as a control parameter. Also RSRP is used as a trusted received power indicator. By using this counter measuring technique, the proposed new method using cross layer metrics is more immune to jammer or interference signals.

The proposed new method is applied to basic AODV routing protocol and gets better performance results. As a feature work, this method can be applied to other routing protocols with some minor modifications. Also physical implementation of the new routing life decision method can be done by using small drones and commercial hardware setups.

REFERENCES

[1] I. Bekmezci, O. K. Sahingoz, and S. Temel, “Flying Ad-Hoc Networks (FANETs): A survey,” *Ad Hoc Networks*, ISSN: 1570-8705, Vol: 11, Issue: 3, Page: 1254-1270, 2013.

[2] Request for Comments (RFC) from Internet Engineering Task Force (IETF), “Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations”, *IETF RFC2501*, January 1999.

[3] Manpreet and J. Malhotra, “A Survey on MANET Simulation Tools,” in *CIPECH 2014*, IEEE, 2014.

[4] S. Miller, “An Accessible, Open-Source, Real Time AODV Simulation in MATLAB”, Department of ECE Missouri University of Science & Technology, 2018, [Online]. Available: https://stewythe1st.github.io/doc/cpe6420.pdf

[5] J. Jiang and G. Han, “Routing Protocols for Unmanned Aerial Vehicles,” *IEEE Communications Magazine January*, Volume: 56 , Issue: 1 , Page: 58 – 63, 2018.

[6] R. Shirani, “Reactive-Greedy-Reactive in Unmanned Aeronautical Ad-hoc Networks: A Combinational Routing Mechanism,” MS Thesis in ECE Carleton University, August 2011.

[7] Request for Comments (RFC) from Internet Engineering Task Force (IETF), “The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4,” *IETF RFC 4728.*

[8] Request for Comments (RFC) from Internet Engineering Task Force (IETF), “Ad Hoc On-demand Distance Vector Routing,” *IETF RFC3561*.

[9] Internet-Draft from Internet Engineering Task Force (IETF), “Dynamic MANET On-demand (DYMO) Routing,” *IEFT Internet-Draft v26.*

[10] C. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", in *ACM SIGCOMM'94 Conference on Communications Architectures*, Protocols and Applications 234-244, 1994.

[11] Request for Comments (RFC) from Internet Engineering Task Force (IETF), “Optimized Link State Routing Protocol (OLSR),” *IETF RFC 3626*.

[12] K. Liu, J. Zhang and T. Zhang, “The Clustering Algorithm of UAV Networking in Near-Space,” *in Proc. ISAPE,* pp. 1550–53, 2008.

[13] B. Karp, “Greedy Perimeter Stateless Routing (GPSR)”, in *IEEE Journal on Selected Areas in Communications,* Vol. 27 pp. 1158-1168, 2000.

[14] J. Biomo, T. Kunz, and M. St-Hilaire, “Routing in UnmannedAerial Ad Hoc Networks: A Recovery Strategy for Greedy Geographic Forwarding Failure”, in *Proc. WCNC,* pp.2236–41, 2014.

[15] A. V. Leonov and G. A. Litvinov, “About Applying AODV and OLSR Routing Protocols to Relaying Network Scenario in FANET with Mini-UAVs”, in *APEIE 2018,* 2-6 Oct. 2018.

[16] A. Nayyar, “Flying Adhoc Network (FANETs): Simulation Based Performance Comparison of Routing Protocols: AODV, DSDV, DSR, OLSR, AOMDV and HWMP,” in *International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD),* 6-7 August 2018.

[17] H. R. Hussen, S. C. Choi, J. H. Park and J. Kim, “Performance Analysis of MANET Routing Protocols for UAV Communications,”in *ICUFN 2018,* 3-6 July 2018.

[18] A. Bujari, C. E. Palazzi, D. Ronzani, “A Comparison of Stateless Position-based Packet Routing Algorithms for FANETs, ” in *IEEE Transactions on Mobile Computing,* Vol. 17, Issue: 11, 2018.

[19] X. Li and J. Huang, "ABPP: An Adaptive Beacon Scheme for Geographic Routing in FANET," in *18th PDCAT Conference*, 2017.

[20] T. Zhang, X. Li and J. Li, "A Particle Swarm Mobility Model for Flying Ad Hoc Networks," in *IEEE Global Communications Conference,* 2017.

[21] A. C. Zucchi and R. M. Silveira,"Performance Analysis of Routing Protocol for Ad Hoc UAV Network," in *Proceedings of the 10th Latin America Networking Conference*, 73-80, 2014.

[22] M. Elshaikh, M. F. M. Fadzil, C. M. Nor, C. Isa , N. Kamel and L. Hasnawi, "SNR-based Dynamic Sequence Distance Vector Routing Protocol Metric for Mobile Ad-Hoc Networks," in *International Conference on Computer & Information Science (ICCIS)*, 12-14 June 2012.

[23] F. Alnajjar and Y. Chen, "SNR/RP Aware Routing Algorithm: Cross-Layer Design for MANETs," in *International Journal of Wireless & Mobile Networks (IJWMN),* Vol 1, No 2, November 2009.

[24] B. R. Miete and S. Shanmugavel, "Received Signal Strength-based Cross-layer Designs for Mobile Ad Hoc Networks", in *IETE Technical Review*, 25:4, 192-200, 2008.

[25] N. Sarma and S. Nandi, "A Route Stability Based Multipath QoS Routing (SMQR) in MANETs," in *Proceedings of the First International Conference on Emerging Trends in Engineering and Technology,* July 16 - 18, 2008.

[26] B. Amel and M. M. Zoulikha, "Routing Technique with Cross-layer Approach in Ad Hoc Network," in *Second International Conference on the Applications of Digital Information and Web Technologies,* IEEE, Vol. 1, issue 9, pp.313-318, 2009.

[27] C. Gu and Q. Zhu, "Cross-Layer Routing Protocol for Mobile Ad Hoc Networks Based On Minimum Interference Duration," in *Proceedings of the 2nd*

International Conference on Computer Science and Electronics Engineering, Vol. 347-350, pp. 2028-2032, 2013.

[28] S. M. Al-Shehri and P. Loskot, “Enhancing Reliability of Tactical MANETs by Improving Routing Decisions,” in *J. Low Power Electron. Appl.*, 2018.

[29] Cisco Wireless Mesh AP, “Design and Deployment Guide, Release 7.3”, [Online].Available:https://www.cisco.com/c/en/us/td/docs/wireless/technology/mesh/7-3/design/guide/Mesh/Mesh_chapter_011.html

[30] Spectra LO, “iSource+™ Low Cost LPFRS Spec” August 2018, [Online]. Available: https://www.orolia.com/sites/default/files/document-files/iSource_LPFRS_Spec_1.pdf

[31] Sakhaee, Ehssan, and Abbas Jamalipour. "Aerouter/spl trade/-a graphical simulation tool for routing in aeronautical systems," in *IEEE Wireless Communications and Networking Conference*, 2005. Vol. 4. IEEE, 2005.

[32] E. Sakhaee, A. Jamalipour and N. Kato, “Aeronautical ad hoc networks” in *IEEE Wireless Communications and Networking Conference (WCNC 2006)*, Vol. 1: Las Vegas, NV, 2006; 246– 251

[33] E. Sakhaee and A. Jamalipour, “Stable clustering and communications in pseudolinear highly mobile ad hoc networks,” in *IEEE Transactions on Vehicular Technology 2008*; 57(6): 3769– 3777

[34] E. Sakhaee, T. Taleb, A. Jamalipour, N. Kato, Y.Nemoto, “A Novel Scheme to reduce control overhead and increase link Duration in highly mobile ad hoc networks,” in *Proceedings Wireless Communications and Networking Conference*, Hong Kong, March 2007, pp. 3972–3977

[35] Hu, Xi, Jinkuan Wang, and Cuirong Wang, "Mobility-adaptive Routing for Stable Transmission in Mobile Ad Hoc Networks," JCM 6.1 (2011): 79-86

[36] Zhou, Jinhua, et al. "A simulation analysis of nodes mobility and traffic load aware routing strategy in aeronautical ad hoc networks," in *Proceedings of 2012 9th International Bhurban Conference on Applied Sciences & Technology (IBCAST)*. IEEE, 2012

[37] A. Bentaleb, S. Harous, S., A. Boubetra, "A Weight Based Clustering Scheme for Mobile Ad hoc Networks", in *The 11 th International Conference on Advances in Mobile Computing & Multimedia (MoMM2013)* , pp. 161-167, 2013