Deanship of Graduate Studies

Al-Quds University

Reinforcement Authentication Model based on DYMO Routing Protocol for MANET (RAD)

Mohammad Rajeh Ali Ayyad

M.Sc. Thesis

Jerusalem-Palestine

1441-2020

Reinforcement Authentication Model based on DYMO Routing Protocol for MANET (RAD)

Prepared by:

Mohammad Rajeh Ali Ayyad

B.Sc. Computer Engineer, AlQuds University, Palestine

Supervisor: Dr. Rushdi Hamamreh

A thesis submitted to Faculty of Engineering, Al-Quds University in Partial fulfillment of the requirements for the degree of Master of Electronic and Computer Engineering.

1441 - 2020

Al-Quds University

Deanship of Graduate Studies

Electronic and Computer Engineering

Thesis Approval

Reinforcement Authentication Model based on DYMO Routing Protocol for MANET (RAD)

Prepared by: Mohammad Rajeh Ali Ayyad

Registration No. 21610032

Supervisor: Dr. Rushdi Hamamreh

Master thesis submitted and accepted. Date: June 2, 2020

The names and signatures of the examining committee members are as follows:

1- Head of Committee: Dr. Rushdi Hamamreh Signature:

2- Internal Examiner: Dr. Nidal Kafri Signature:

3- External Examiner: Dr. Mohammed Abu Taha Signature:

Jerusalem – Palestine

1441 – 2020

Dedication

To the sake of Allah, my creator and my master,

To My great teacher and messenger, Mohammad (May Allah bless and grant him),

To My parents, who raised me to be the person I am today,

To My beloved wife "Alaa", for her outstanding and highly appreciated support and patience day and night throughout the time of my study, and most of all for believing in me,

To my friends, family, work and master colleague, who encouraged and supported, I dedicate the research.

Mohammad Rajeh Ali Ayyad

Declaration

I certify that this thesis submitted for degree of Master, is the result of my own research, except where otherwise acknowledged, and that this is study (or any part of the same) has not been submitted for a higher degree to any other university or institution.

Signed: 

Mohammad Rajeh Ali Ayyad

Date: June 2, 2020

Acknowledgment

All Praise to ALLAH, the Almighty (swt), the greatest of all, on whom ultimately we depend for sustenance and guidance. I would like to thank. Almighty Allah for giving me opportunity, determination and strength to do my research.

Peace and blessing of Allah be upon the best of humankind, the Messenger of Allah, Prophet Muhammad (pbuh).

Praise, appreciation and gratitude to the great Muslim scientist Muhammad ibn Musa al-Khwarizmi, whom we use the word “Algorithm” that is derived from his name.

I would like to express my sincere gratitude and appreciation to my supervisor Dr. Rushdi Hamamreh, for the continuous support of my study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

I sincerely appreciate the cooperation and help by the academic staff of the Faculty of Engineering at Al-Quds University.

Many thanks to my Master study colleagues and everyone shared me his feelings, encouragement and support.

Big thanks for all

Abstract

Mobile Ad hoc network (MANET) is a group of mobile nodes that make together a network, this network doesn't have a fixed infrastructure or topology. Such networks can work in very tough zones, like crises zones. If one of the mobile nodes must interact with another node within the same transmission range, they can interact directly with each other; on the other hand, if they are in different transmission ranges, the nodes between them must forward the packet to them. Thus, mobile nodes can be act as routers.

the nodes in MANET that want to communicate with other nodes will use one of the known routing protocol to find the shortest path between sender and receiver, this shortest path depends on hops count and sequence number for the nodes, the main routing protocols that use this algorithm to select the shortest path of routing are AODV, DYMO and others.

Due to the unique structure of MANET, there are many threats that face the packets transmission process, but the known routing protocols haven't the ability to realize the security threatens for packet transmission process between the source node and the destination node, therefore, new model is proposed to improve the security in MANET.

The new model uses encryption and authentication techniques for securing the data, and reinforcement learning for improve the cooperation between nodes in MANET that will increase the performance and the efficiency for the network.

The results of the simulation show the improved effectiveness of the proposed model on the MANET activities, while the performance were increased/decreased due to the overheads of the new model.

Table of Contents

List of Figures

List of Algorithms

List of Tables

Chapter One: Introduction

1.1. Introduction

1.2. Problem Statement

1.3. Threats model

1.4. Related work

1.5. Motivations

1.6. Thesis Contributions

1.7. Research methodology

1.8. Thesis outline

1. Chapter One: Introduction

1.1. Introduction:

This chapter presents an overview regarding this work. It introduces the problem statement, related works, motivation, contribution and methodology of this research.

Networking is connecting nodes such as computers, mobile phones, hubs and switches to allow collaboration between users for these nodes in a wide range for sharing information, by using different technologies such as wired networks (e.g. telephone and TV networks), as well as wireless networks (e.g. wireless local area networks (WLANs), wireless sensor networks, satellite communication networks).

Network technology includes two types of networks: local area network (LAN) and wide area network (WAN). LANs has two computers or more connected together in short distance like home or office. WAN is a large network covers cities, countries or the whole world and it contains many LANs connected together [1] .

There are two ways to connect nodes in LANs and WANs: wired and wireless, in wired networks all nodes connected with each other by using wires, which transmit the data between these nodes. Wireless networks use radio signal to transfer the data between nodes. To avoid collisions between radio signals, sender and receiver nodes must use same frequency channel.

During the past decades, wireless networks suffered from lack of confidentiality and security for the data, but with multiple new technologies in security and encryption and the free movement for devices which connected to wireless network, the use of wireless network increased and covered most homes and offices. Despite that, wired network is considered better than wireless in many features as found in Table1.1, but many technologies can’t use wired networks due to the need of mobility like in mobile devices which became the main trend in 21st century and took the world to new stage of technology.

Table 1.1. Differences between wireless and wired networks [2].

Features	Wired network	Wireless network
Data transmission speed	Higher than wireless network	Lower than wired network
Bandwidth range	High	Low
Cost of infrastructure	Low due to the low cost of cables	Expensive because the cost of access points and wireless routers is high
Installation	Complicated and take time and efforts	Easy to install
mobility	Limited mobility because of physical connection	free mobility
Transmission medium	Copper wires	Radio waves
Noise	Low noise signal	High noise signal
Hubs and switches	Need for hubs and switches	No need for hubs and switches
Security	Good	Weak
Maintenance cost	High	Low
Quality of service (QoS)	Better than wireless	Poor due to delay in connection setup and

1.2. Problem statement:

In MANETs, all nodes have freedom of movement, so there are two main problems facing MANETs, routing and security. Due to the mobility of the nodes, the network's topology will change continuously which will decrease the stability of routes that connect nodes together. In addition, malicious nodes can connect to that network, which will produce several kinds of attacks that disturb the network. These reasons call for proposed protocol that can reduce the problems of routing and security.

1.3. Threat Model

- A Malicious node may be any node along the route which attack in data-plane by misdirecting packets
- An active attacker may violate the detection protocol by modifying, dropping or misdirecting probe packets to remain undetected.
- The malicious router may execute other attacks such as:

  Black hole attack & Man in the middle.

1.4. Related works:

Most important process in MANET is create path between source (S) and destination (D) to transmit the packets, this path must be the shortest to decrease the transmission time between S and D. In addition, this path must be secure to prevent any attack from malicious nodes, that shortest path which include security techniques will be the best path. To decrease the time for finding the best path, many researchers use some authentication techniques to create authenticated nodes, here are some examples of that:

The authors Hwan-Seok Yang and Seung-Jae Yoo proposed an authentication technique to provide secure communication by increasing the reliability of the nodes. They use cluster structure for authentication technique, they made a certificate authority by using the proposed techniques and cluster head, it will manage authentication information of member nodes. They confirmed the performance of the proposed technique by experiments [15].

The research by Neha Sharma, Ambrish Gangal showed an effective node authentication structure for MANET that can readily provide a means for the malicious node to be detected. The primary alternatives in the research are use of TTP (trusted third party), public-private key pair and authenticating malicious nodes [16].

Utpal Kumar Verma , Sushil Kumar and Ditipriya Sinha suggest secure mechanism for authentication of nodes in the MANET, they proposed an authentication protocol based on digital signature with hash function to create a certificate that can exchanged between nodes [17].

The author Jaydip Sen showed key exchanged protocol between nodes in MANET, this protocol based on multipath communication. After made the simulation, the results showed the effectiveness of that protocol even in network that have large number of malicious nodes [18].

The research by S. Neelavathy Pari ; Sabarish Jayapal ; Sridharan Duraisamy use a system with a trust model and SHA-1 key encryption. This system tries to detect the malicious nodes in MANET. The experiences for the nodes in network help to build a trust value. By using specific hashing techniques, which called SHA-1 the efficiency of trust, system is enhanced [19].

B. Lu and U.W. Pooch proposed an authentication protocol by using one-way hash chain to provide effective authentication for communications between any two nodes in MANETs. They also made an analysis for the security properties and performance. The results for that analysis are the protocol incurs low overhead penalty and achieves a tradeoff between security and performance [20].

The research by Yogesh P. Singare ; Manish Tembhurkar proposed efficient initial access authentication protocol, it uses roundtrip messages to distribute the key between nodes. The main idea in this protocol to provide a secure path between nodes to pass messages in safe way [21].

Nitnaware and Thakur suggest new strategy to detect and prevent black hole attack in DYMO, This research work attempts to develop a mitigation algorithm to avoid and prevent genuine nodes from malicious attack [22].

1.5. Motivations:

In the last few years, MANET has been gaining a great deal of interest due to its unique topology that does not have fixed infrastructure or centralized unit. It has many applications as mentioned earlier, however the main obstacle in MANET application is to secure the paths that packets will pass through with the least possible time, because the basic nature of MANET makes it unprotected against malicious attacks compared to traditional wired network. In this work, we aim to build a new protocol that creates authenticated nodes that is considered reliable in dealing with packets transmitted in that path by using hashing techniques.

Moreover, we are aiming to use reinforcement techniques to motivate nodes to act in a manner that coordinates with the network without being suspected as a malicious node.

1.6. Thesis Contributions:

The goal of this work is to develop a new model to improve the security based on DYMO protocol. In this thesis, a new approach has been proposed and the results of thesis have proved that there is an improvement in performance according to performance measures.

- We have designed and developed RAD protocol by adding security techniques for Dynamic MANET on Demand (DYMO) protocol; these techniques include MD5 hashing, encryption using Diffeie-Helman algorithm for key management and reinforcement learning phase which is realize using rewards and punishments concepts based on machine learning approach.
- We check the performance of proposed approach by using the simulation and comparing results with DYMO.

1.7. Research methodology:

In this thesis, the research depends on studying previous works regarding routing protocols, and checking their performance and the security for those protocols. Then, comparing these results with the new proposed protocol for the same nodes number. We have used the network simulator (NS-2), as it is an open source software and many researches for MANET routing were implemented using this software. Besides, it provides online support and documentation and its free software.

1.8. Thesis outline:

This thesis is organized as follows: next Chapter Two introduces a background MANET and its routing protocols. In Chapter Three we find a literature review about authentication that use hashing techniques, then review the encryption techniques.

Chapter Four will present the proposed protocol that will improve one of MANET's routing protocols by increasing the security using a combination of hashing, encryption and reinforcement methods. In chapter 5, we discuss and analyze the obtained experimental results. Finally, a conclusion and proposed future work in Chapter 6.

Chapter Two: Background, MANET and its routing protocols

2. Chapter Two: Background, MANET and its routing protocols

2.1. Introduction:

The primary objective for MANET is to respond to the difficulties of the dynamically evolving topology, and create a proper and effective communication route between any two nodes with minimum cost tracking and the least bandwidth use. The issue with the development of routing protocols is not easy, because the environment of ad hoc networks present new difficulties that are not exist in traditional network. A series of routing protocol were created to resolve this issue, and the amount of these protocols continues to increase daily [23].

2.2. Wireless network types:

Wireless networks split into two main sections depending on the way that nodes are connected to each other. First one section is infrastructure based and second one is ad-hoc network.

2.2.1. Infrastructure based networks:

Most wireless network function in infrastructure based mode, in this type of networks all devices connected to single access point or base station which is usually the router, even if two nodes are close to each other they will not be able to communicate directly on their own, instead, they must communicate indirectly using the access point, sender can send the packets to access point and it will resend these packets to receiver. Infrastructure based network is good in construction permanent network, because access points have high power radio waves, so it can cover a wide area [3].

2.2.2. Ad-hoc network:

The Cambridge dictionary defines Ad-hoc as: “made or happening only for particular purpose or need, not planned before it happens”. Ad-hoc network also known as peer to peer network, this type of network doesn’t have an access point or base station to connect nodes together; it relays on the concept of decentralization, instead, nodes in ad-hoc network communicate directly to each other using radio waves. Ad-hoc network includes three types of network: wireless sensor networks (WSNs), wireless mesh networks

(WMNs) and mobile ad-hoc networks (MANETs) [4]. Figure 2.1 shows the difference between infrastructures based network and ad-hoc network.

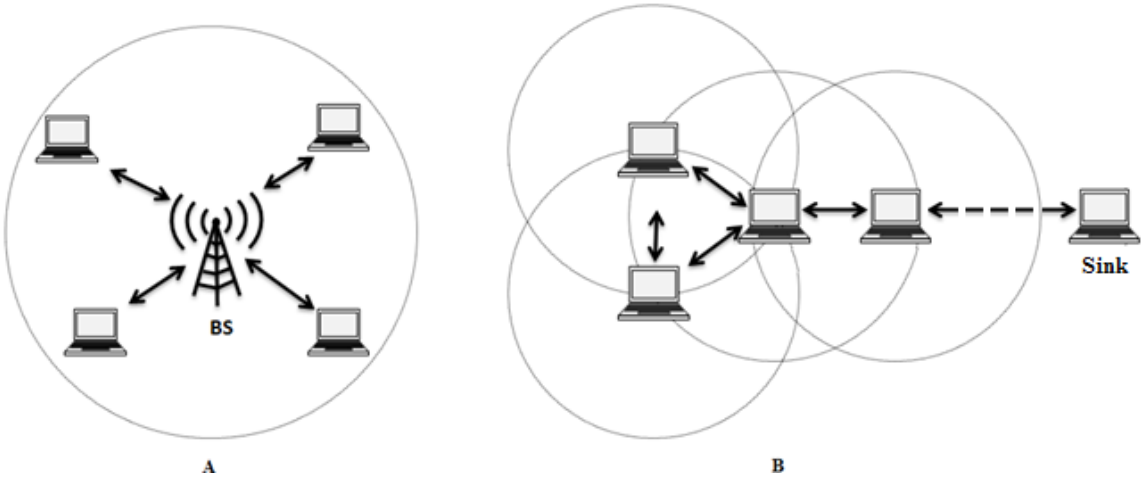

Figure 2.1. a) Infrastructure based network, b) Ad-hoc network

MANET is a collection of mobile nodes; that construct an impermanent network without need of the help of centralized unit as in ordinary networks, and it have a node that connected to a sink and power source. These nodes doesn't have a wide range of transmission, so every node look for support from the neighboring nodes in transmitting packets, in this case all nodes acts as transmitter and receiver, in this way, if any two nodes want to communicate together and they have been in same range, they can communicate directly, else, they need a node in the middle which will act as a router between these nodes to help them in communication [5]. Figure 2.2 shows the direct and indirect communication between nodes. Due to nature of MANETs, this network type is appropriate in case of absence of static structure.

Figure 2.2. a) Direct communication, b) Indirect communication

2.2.3. Features of MANET:

Most important features of MANET can describe as follows:

2.2.3.1. Autonomous terminal:

Mobile nodes in MANET are autonomous nodes; each node can act as client or router, in other words, beside that node can act as sender or receiver, it can also perform as mediator between two other nodes to help in communication between two nodes, which found in wide range.

2.2.3.2. Distributed operation:

Due to decentralization in MANET, the control of the operation is distributed for all nodes; each node in MANET must have the ability to sending and receiving data, path discovery and secure the discovered path.

2.2.3.3. Dynamic network topology:

Nodes in MANET are moving rapidly, so MANET does not have fixed topology and the connections between nodes change in different periods, this mean that nodes will be disconnected from neighboring nodes and search for new connection to other nodes in the new position [6].

2.2.3.4. Fluctuating link capacity:

Wireless connection nature has high bit error rate, one path can used in multiple sessions, and this will cause a noise, fading and interference in the channel used in communication between nodes and it will provide less bandwidth compared with wired network [7].

2.2.3.5. Light weight terminals:

Nodes in MANET usually have limited power, small memory size and small CPU processing ability, devices with these characteristics need a special optimized algorithm to deal with that limited resources.

2.2.4. Security Requirements in MANETs:

MANET security is considered a crucial requirement to protect the data exchanged throughout the nodes. In order for a MANET to be secured we should maintain security

during different stages including linking, routing, data transmission, and data forwarding stages. Numerous security requirements are applied providing different security solutions including:

a. Confidentiality: It implies that data get to nodes that have been approved to access it.
b. Integrity: it gives security of message packets while it passes between the nodes in the route.
c. Availability: A node must be able to gives all its responsibilities without take care of the security condition in that node.
d. Non repudiation: The sender cannot deny that it sent the message, and receiver cannot deny that it received the message.
e. Authentication: It is important to prove that all participant in transmission process are real and not malicious by detecting their identities.
f. Authorization: It is a process to determine the permissions for nodes to access the network resources.
g. Anonymity: The identity for the node must be private, it is not allowed for any node to distribute the other nodes identities.

2.2.5. Challenges in MANETs:

During the last few years, MANET was an important sector. Virtually all aspects of the network have been examined at distinct issue levels one manner or the other. However, no final decision or agreement is reached on any of the issues. There have instead been more issues. The subjects to be addressed are:

a) Device discovery: To identify and report on the existence of newly moved nodes, it is necessary to make a repeatedly update to facilitate the best route discovery process.

b) Bandwidth optimization: Wireless network have much less capacity than wired network. The capacity is always used optimally by routing protocols in wireless networks by maintaining the overhead as small as necessary. The limited range of transmissions also restricts the maintenance of topological data by routing protocols. Maintaining the topological data at all nodes in MANETS, includes greater overhead monitoring that in turn leads to more waste of bandwidth [8].

c) Limited resources: MANET nodes depend on battery power, which is a small finite resource, also storage capacity is limited.

d) **Scalability:** Scalability can widely be described, as whether even in the existence of many nodes, the network is capable of providing an appropriate service.

e) **Limited physical security:** Mobility involves increased security risk, such as the architecture of the peer-to-peer network, it is accessible for legal network users and for malicious attackers. The attacks of eavesdropping, spoofing and denial of service should be taken into account [9].

f) **Infrastructure-less and self-operated:** MANET's self-healing function requires that any nodes that move out of their reach must be replaced by other near nodes.

g) **Poor Transmission Quality:** this is an important problem in wireless network, it happens due to existence of several error sources that result in degradation of the received signal.

h) **Network configuration:** The whole MANET architecture is dynamic, which will be the reason for repeatedly connection and disconnection of the variable links.

2.2.6. Classification of attacks in MANET:

Attacks in MANET can be classified in two main categories: active attacks and passive attacks. This classification is based on their properties and attack goals. Figure 2.3 shows examples for MANET attacks [10].

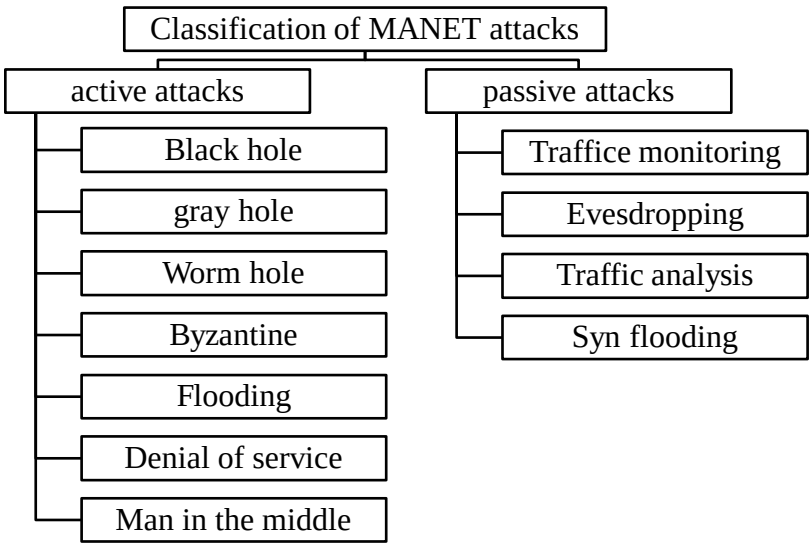

Figure 2.3. Classification of MANET attacks

Active attacks: it is involving modification of data in packets that passing through the malicious node, or even drop some packets. Here there are some examples of active attacks:

1) Black hole: In this attack, a malicious node acts like a black hole and drop all the packets that pass through it. When this malicious node acts as connecting node in a network path, then it will separate this network into two disconnected networks.
2) Gray hole: it is a kind of black hole attack, but in this attack, the packets will drop selectively, not in random manner [11].
3) Warm hole: malicious node makes a fake path that is shorter than the original path, it also can contain two nodes with tunnel between them. Malicious node will capture the packets from one location in the network, and then it sends them to other node that distributes the packets locally.
4) Byzantine: it is include creating routing loops, send the packets through long paths or dropping packets.
5) Flooding: malicious nodes will send false packets into network; this will consume the bandwidth and the processing resources. This attack has serious effects on MANET due to the limited bandwidth and resources like battery power.
6) Denial of service (DoS): main goal for this attack is complete disruption of routing information; this will fail the whole process of MANET. This attack includes flooding the network to prevent the network traffic and disrupting the connections between nodes.
7) Man in the middle: the malicious node takes its place between two nodes, and then it snoops the data transmitted between them. In some cases, the malicious node impersonates the source to send packets for the destination, or impersonates the destination to replay to the source [12].

Passive attacks: this type of attacks does not make any modification in packets, it just snoops the data that transmitted in the network. It is hard to detect this attack since the whole process in network does not affect. Here there are some examples of passive attacks:

1) Traffic monitoring: this type developed to collect data about the nodes in network and detect their functionality to prepare a specific way to attack that network, this type also used to attack satellites, WLANs and cellular networks.

2) Eavesdropping: malicious node shares the same radio channel with the original receiver node, then it read the messages, also it can send some fake messages to that receiver.
3) Traffic analysis: this attack used to collect data about nodes in network and how much data was transmitted in that network.
4) Syn flooding: it is type of DoS, the malicious nodes make many connection requests to other nodes, and this will fill the bandwidth and disturb the network [13].

2.2.7. MANETs applications:

MANET has many characteristics make it a flexible network that contains nodes with free mobility, so it easy to any node to joins or leaves this network. In addition, it is good for applications that used in environment that does not have a fixed infrastructure. MANET have many applications in real life, here is some of these applications:

1) Military applications: in battlefields, there is no fixed infrastructural network or centralized unit that can manage the communications between soldiers and military vehicles, so one of the solutions is MANET, soldiers can communicate with each other and exchange the military orders and plans with the commander in main headquarter [14].
2) Rescue operations: in disasters, like floods, fires and earthquakes, most of systems "including the communications system" will fail, MANET will be the technology that helps rescue teams to communicate together to manage the relief efforts.
3) Business work: whenever any emergency meeting held outside the headquarter, information can exchange by using MANET.
4) Classrooms and conferences: MANET provides an easy way to connect participants in classrooms or conferences to share information and multimedia.
5) Personal area network: it is a short range rang MANET, used to connect small group of nodes like laptops via Bluetooth.
6) Commercial applications: smart transportation uses MANET to manage vehicles movement; this will help to prevent the collisions and traffic congestion. In addition, it can use in smart taxi system which help passengers to find a nearby taxi.

2.3. Routing algorithms:

Routing consists of many processes depending on different rules (protocols) and steps (algorithm), the main task of routing is giving information needed by routing algorithm to determine decisions about the path which must be selected.

Routing algorithms have to main categories: adaptive and non-adaptive [24].

2.3.1. Adaptive routing algorithm:

Also known as dynamic routing algorithm takes the routing decisions depending on topology and traffic for the network by using parameters like hop count, distance and estimated transit time.

Adaptive routing algorithm has three main categories:

2.3.1.1. Centralized algorithm: also called global routing algorithm that depends on calculating the least cost path by using complete information about the global network, link state is an example of this algorithm.

2.3.1.2. Isolation algorithm: it uses local information to determine the path between source and destination.

2.3.1.3. Distributed algorithm: also called decentralized algorithm because it computes the path between source and destination in distributed way, in this algorithm, each node has information about its neighbor nodes only. Distance vector is an example of this algorithm.

2.3.2. Non-adaptive routing algorithms:

It is also known as static routing algorithm; the routing data is transferred to the routers when the network is booted [25].

Non-adaptive routing algorithms have two main categories:

a) Flooding: in this case, the nodes will broadcast every incoming packet for all links.

b) Random walks: the node will send the incoming packets to one of its neighbor in random manner.

2.4. Categories of routing protocols:

The main issue in routing protocols is that nodes will join and disjoin the network continuously and in different areas, to solve this problem there are numerus types of routing protocols, these protocols can divide into three categories according to the way they work: Proactive, Reactive and Hybrid routing protocols. Routing protocols are created to handle a number of nodes with limited resources.

Given the growing number of mobile nodes, it is necessary to decrease the routing message overhead. All the routing information is saved in routing table, the size of this table must be smaller as much as possible in order to prevent affecting the control packet transmitted over the network [26].

Figure 2.4 shows and summarizes the categories "with some examples" that mentioned before which classified on how and when to reach it, but both choose the shortest path to the destination.

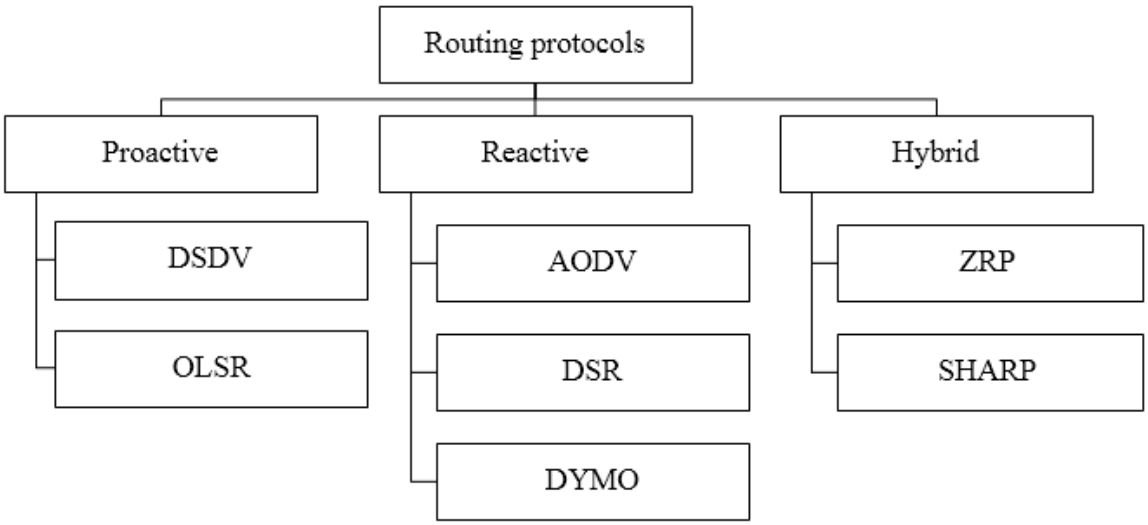

Figure 2.4. Categories of routing protocols

2.4.1. Proactive routing protocols (table driven):

Proactive routing protocols also called table driven protocols. They are based on distance vector and link state protocols that attempt to make every node maintains one or more list called routing tables. These lists are periodically modified. When any update occurs in the

networks topology, the node broadcasts a message to whole network. This message contains the information for the updates that happened in the network. Nevertheless, since up-to-date data is retained, it has higher overhead costs, as a consequence; network performance may be affected, but the real information is provided on network availability [27].

Proactive routing protocols provide the paths for each node in MANET in advance, so whenever the path is needed it will be ready for use, this will decrease the average delay per packet.

Destination-sequenced distance-vector protocols (DSDV) is proactive routing protocol that uses distance vector routing algorithm. While Optimized link state routing protocols (OLSR) are use link-state routing algorithm.

2.4.2. Reactive routing protocols (On demand):

These protocols also called on demand protocols and they are bandwidth-efficient protocols. Only if needed by the destination node will this routing protocol create routes. When a node needs path to destination, a route discovery system is implemented within the network. This task will achieve once a path has been identified or all possible route have been tested.

The way that these protocols work making it an efficient way to minimize the use of bandwidth by reducing the links that created for same routes compared with proactive routing protocols.

These protocols include two main processes: route discovery and route maintenance.

a) **Route discovery process:** In this process, every node prepares a list for direct neighboring nodes and their link cost which called routing table, this list is created by sending route request packet (RREQ) for the neighboring nodes, the nodes that receive that packets will send back a route replay packet (RREP). By measuring the time spend in sending and receiving that packets and dividing it by two, the cost for that link will calculated.

   Route discovery process will create more than one route from source to destination, and then the node will choose the best route to use in sending data [28].

Main advantage of this process is that all nodes will share in create the path through their routing tables; this will reduce the overhead for the future route discovery processes.

b) **Route maintenance process:** This process makes modifications in routing tables for the nodes when it detects a break in any surrounding links. This node sends route error packets (RERR) to the nodes, when other nodes receive that packet, they will start a new route discovery to avoid that break link [29].

2.4.3. Hybrid routing protocols:

Proactive and reactive protocols have advantages and disadvantages; Hybrid routing protocols combine the advantages of both.

Such protocols have ability for adaptation and respond to the area and location of the source and destination. This will split the network into different zones and then observes the location of source and destination.

Proactive routing protocols are used for data exchange if the source and destination are found in same zone, else, if source and destination did not found in same zone, then the reactive protocols are used [30].

Examples for these protocols are Sharp Hybrid Adaptive Routing Protocol (SHARP) and Zone Routing Protocol (ZRP)

2.5. Desirable Properties of Routing Protocols:

2.5.1. Distributed operation: This is one of the important properties to MANET, it proves that MANET networks work in distributed way due to its nature, so any nodes can connect to that network or leave it.

2.5.2. Loop –freedom: Generally, this property is attractive. This applies to stopping packets from spinning randomly around the network. To increasing the performance impact of the problem, solutions such as Time to Live (TTL) values may be used. A more organized approach or a more advanced one is likely to produce better overall results [31].

2.5.3. Demand based operation: By using on demand operations, MANET networks use the bandwidth and energy in an intelligent manner, which will make that networks one of the most efficient networks despite of the delay in route discovery process.

2.5.4. Proactive operation: If the bandwidth and energy capacities are sufficient for proactive operation, then no need for using on demand operation, this will decrease the time for route discovery.

2.5.5. Security: Due to the unique MANET infrastructure, routing protocol exposed to many attacks more than wired network, so the security issue is important in choosing the routing protocols that run the network.

2.5.6. Sleep period operation: as mentioned before, the nodes in MANET have limited resources, so some nodes may need to stop transmitting data for specific time period, routing protocols should handle this status by give that nodes a "sleep" period by avoid dealing with that nodes for a while [31].

2.6. MANETs main routing protocols:

This part will discuss the main routing protocols with examples, and it will highlight the pros and cons for each protocol through a comparison between them.

Main protocols that will discuss are DSDV, DSR, AODV, DYMO and ZRP.

2.6.1. Destination-sequenced distance-vector protocol (DSDV):

2.6.1.1. Description:

In this protocol, each node in the network has a routing table that includes all routes to any destination and the number of hops in each route. Any updates happen in the network will cause a broadcast for new routing tables.

By using sequence number to tag each node, DSDV guarantee loop freedom in the network. Each sequence number determines the freshness of any route, so the highest sequence number shows the latest route to destination.

To decrease the number of broadcasting the updates, DSDV use two ways to define the update messages: full and incremental dump. In case of full dump, the message will include all information about routes, but in incremental dump, the message includes just the changed information since last dump [32].

2.6.1.2. DSDV algorithm:

1. If the new route has the highest sequence number, the source will choose that address and discard the old routes.

2. In case that the new sequence number is equal to another one in other route, DSDV will choose the route with lowest cost.

3. When DSDV choose the new route, all metric for the chosen route will incremented.

4. This process will still be running until all nodes are updated, but if there Is any duplicated packets, the packet with lowest metrics will be chosen.

Figure 2.5, shows a network includes 9 nodes, the continuous lines shows that the node found in the neighboring nodes ranges, the dashed lines shows the outrange nodes for node 3, so node 3 has no information about node 9 [33].

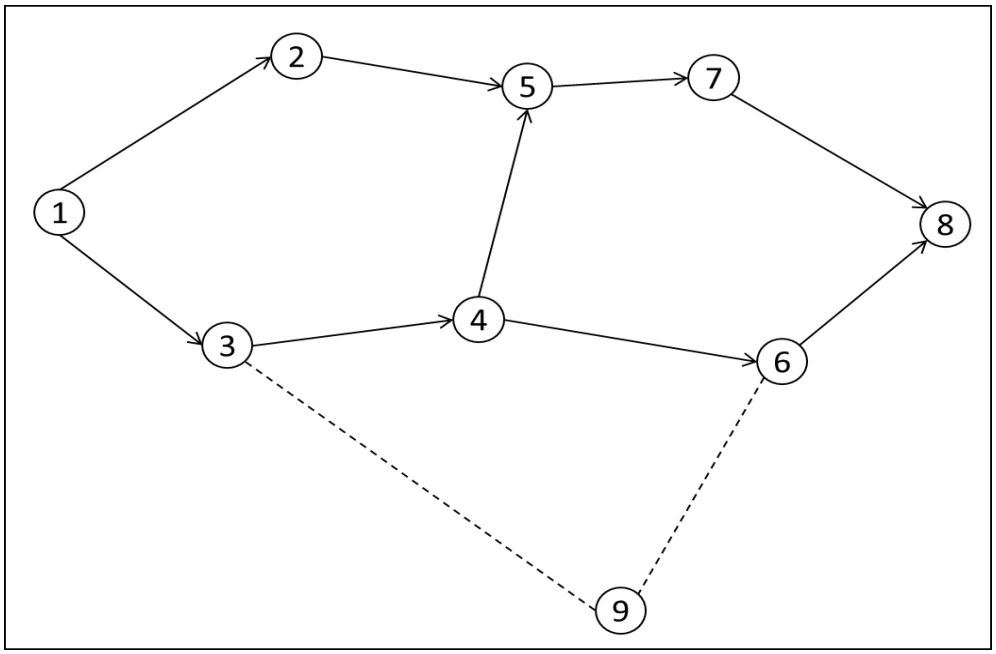

Figure 2.5. MANET with 9 nodes

As an example, Table 2.1 shows the routing table for node 3, nodes 1 and 4 are the neighbor nodes.

Table 2.1. Routing table for node 3

Destination	Next hop	Metrics	Distance sequence number
1	1	1	123
2	1	2	516
3	0	0	212
4	4	1	168
5	4	2	372
6	4	2	434
7	4	3	355
8	4	3	412
9	4	∞	298

2.6.1.3. DSDV advantage:

In this protocol, the routes will be ready for use, so there is no need for wasting time in route discovery process [33].

2.6.1.4. DSDV disadvantage:

The packet overhead increases while the number of nodes increase, because the routing table will include a hug number of records for all nodes, also it will increase the traffic in network and waste the bandwidth due to broadcasting the updated tables every time that any node connect or disconnect to the network, so DSDV is suitable for small networks [33].

2.6.2. Optimized Link State Routing Protocol (OLSR):

2.6.2.1. Description:

Optimized Link Protocol is one of the table-driven routing protocols. It is a development of the Open Shortest Path First (OSPF) protocol to meet the requirement for mobile ad-hoc networks. It uses the link state packet mechanism, which determines a specific number of nodes to be a Multipoint Relay (MPRs); these nodes (MPRs) can reserve the privilege to flood messages in the system. The system is executed when every node on the network sends a Hello Message intermittently to the neighboring nodes and the neighboring hubs

must do not forward the message to other nodes and afterward the node chooses a lot of nearby nodes to turn into a Multipoint Relay (MPRs). Intermittently a (MPRs) flood the network with control Topology (TC) messages so as to distinguish each node and it`s (MPR), this component relies upon the choice of set of nodes from the primary hop that can cover every one of the nodes in the subsequent hop [34].

On node 7, the MPR set is characterized as node 7, can just contact node 2 and 4 by node 5 and consequently node 5 is added to the MPR set and similarly node 8 is added to the MPR set for a similar reason, so node 7 contact all the nodes through the MPR set, utilizing the MPR set network traffic was decreased from eight transmissions utilizing flooding for sending in light of the fact that every node broadcasting information to three transmissions utilizing the MPR set, since node 5 and 8, are the main ones answerable for transmitting information for node 7.

In contrast with the traditional link state routing protocol, two extra enhancements were accomplished, rather than every node broadcasting messages, which demonstrate the condition of each link; just the MPR nodes communicate their link state. In this manner, the primary improvement is in lessening the measure of overhead since the quantity of control packets is little; the subsequent improvement is that the size of the control packets has been diminished.

2.6.2.2. features of OLSR:

1) It is additionally thought about the benefits of proactive protocols. Paths are constantly accessible when required, not at all, like on-demand protocols, and OLSR is a proactive protocol that has the upsides of proactive protocols.

2) OLSR is appropriate for high-density networks since it does not need nodes to flood control messages since it has a set of MPR.

3) The data is periodically transmitted to recognize any adjustments in network topology.

4) OLSR needs time to make another path or rediscover a messed up path.

5) Because the nodes of the sort MPR periodically broadcast packets, this implies overhead on the network.

2.6.3. Dynamic source routing (DSR):

2.6.3.1. Description:

The Dynamic Source Routing Protocol is one of the on-demand or reactive routing protocols, it uses an algorithm called source routing, this routing protocol tries to find a path to destination node only when needed. The packet header has a list of all intermediate nodes that included in the path to the destination node, the main difference between this protocol and the table driven protocols is that every intermediate node will forward that packet to its next hop, which listed in the header, and there is no need to return to its routing table. In addition, if there are any updates in the network, there is no need to broadcast that updates, which will save the bandwidth for the network and battery power for nodes [35].

Like any reactive protocol, DSR consists of two processes: Route Discovery and Route Maintenance, these processes are partially different in each reactive protocol.

2.6.3.2. DSR route discovery process:

In this process, the source node starts checking its cache for a path to the destination node, if no path is found, the source node begins to discover a path by broadcasting RREQ packets to the neighboring nodes, when these nodes receive RREQ, they check some conditions:

1. Is this RREQ reached the node before from different nodes?

2. Is the TTL (Time to Live) counter is greater than zero?

3. Is this node is the destination of that packet?

If all conditions are approved, each node will add its unique address to RREQ, then rebroadcast it to all neighboring nodes until it reaches the destination node, when destination node receives RREQ it will create a RREP packet then send it to source node using the chosen route [36].

Figure 2.7 shows the path discovery process, node 1 is the source node and it add itself to RREQ, the neighboring nodes also do the same thing, so each node adds itself to the RREQ list until it reach to node 8 which will be the destination node.

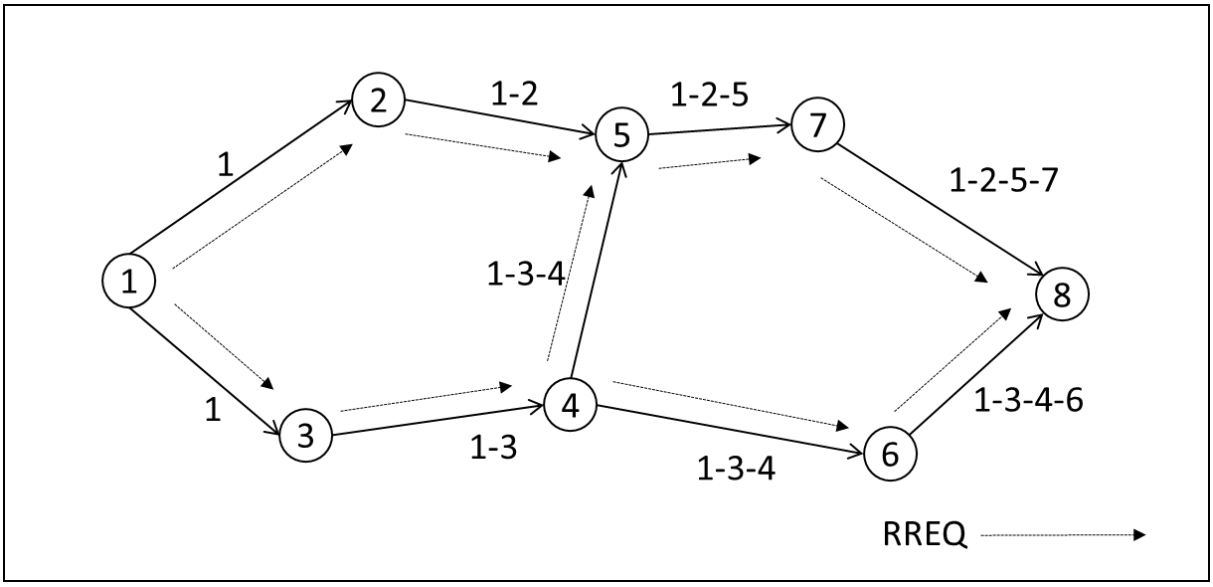

Figure 2.6. RREQ in DSR

A unique number will be added to each RREQ, when node 1 sends a new RREQ, this number will be incremented, and so this unique number and source node will identify any RREQ.

RREQ will arrive to node 8, which will be the destination node, then this node will send RREP to node 1, as shown in Figure 2.8, when node 1 receives RREP, it will save this path to node 8 in its route cache.

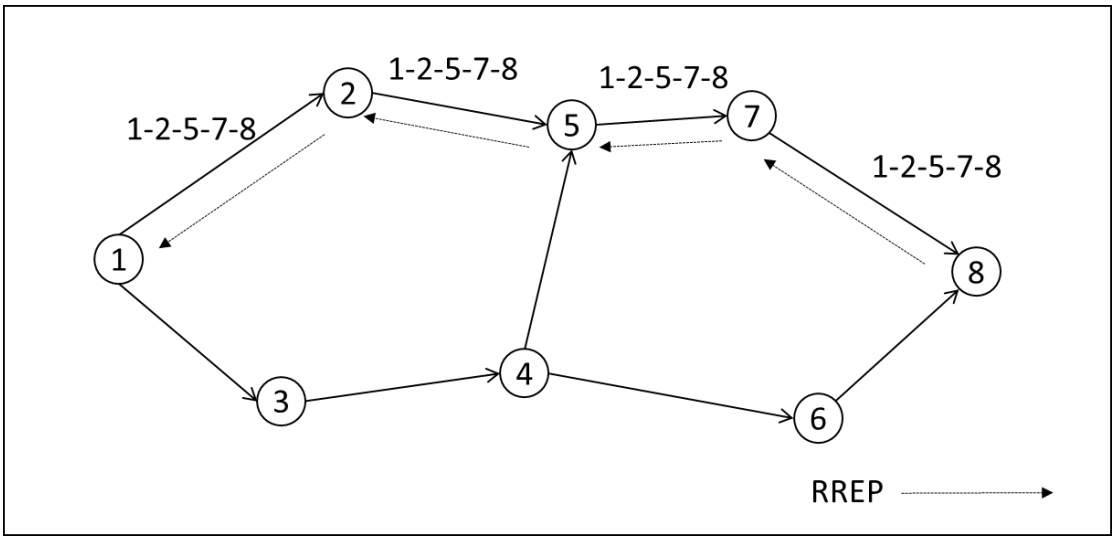

Figure 2.7. RREP in DSR

2.6.3.3. DSR route maintenance process:

When sending a packet by using any route, each node transmitting the packet is liable for affirming that information can stream over the connection from that node to the following node. For example, in Figure 2.8 node 1 want to send a packet for node 8 by using a route that contains the intermediate nodes 2,5 and 7.

For this situation, node 1 is liable for the connection from 1 to 2, node 2 is liable for the connection from 2 to 5, node 5 is liable for the connection from 5 to 7, and node 7 is liable for the connection from 7 to 8.

An acknowledgement can give affirmation that a connection is ready to transmitting the data.

After an acknowledgement has been gotten from neighbor, a node may decide not to require acknowledgements from that neighbor for a short time, except if the network interface connecting a node to that neighbor consistently gets an acknowledgement because of unicast traffic [37].

The acknowledgement request must be retransmitted up to a maximum number of times, after the acknowledgement request has been retransmitted for the maximum number of times, if no acknowledgement has been gotten, at that point the sender considers that connection as a "broken". It should expel this connection from its Route Cache and should return a route error “RERR” to every node that has sent a packet directed over that connection since an acknowledgement was last gotten.

for example, in Figure 2.9, if node 7 does not receive an acknowledgement from node 8 after maximum number of requests, it should return a RERR contains the node address that detected the error and the nodes that the packet was not reached to node 1 by using the same path, just as whatever other node that may have utilized the connection from node 7 to node 8 since last time that node 7 got an acknowledgement from node 8. node 1 at that point expels this messed up connect from its cache then it starts a new route discovery process.

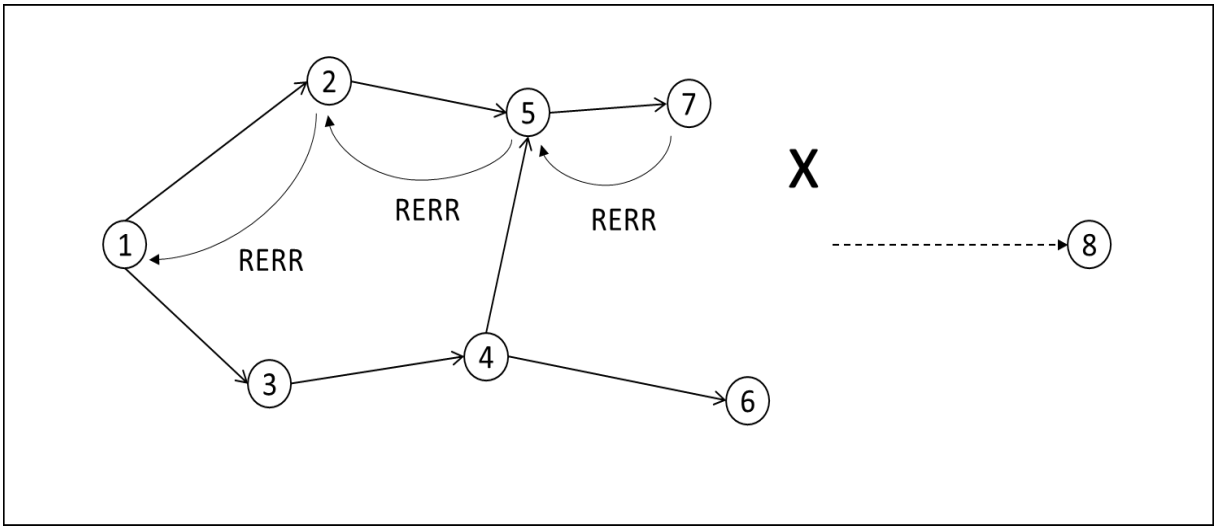

Figure 2.8. RRER in DSR

2.6.3.4. DSR Features:

This protocol can get familiar with the paths by contemplating all the received packets and putting away this data in the routing cache, particularly in the event that it is in the indiscriminate mode, which is where all traffic is heard on the network, without considering the addresses of the link layer. Notwithstanding the advantage of getting to data, there is an issue with data security in light of the fact that any assailant can listen to data and in this manner; there must be applications that encrypt the data before sending it.

Nodes have the features of the source routing, for example, no compelling reason to use Hello Messages every time, this decreases the overhead and improve the transmission capacity and the bandwidth, particularly when the motion of the nodes are slow or when there is an expansion in the number of packets, these features help the nodes to decrease the battery consuming.

The accompanying model shows these features and furthermore outlines a portion of the flaws that may have risen.

If there are three nodes X, Y and Z, where the path from X to Z by means of Y, this prompts X that learns the path to Z as X learns the path to Y, and then again Y learns the path to X and the path Z, and furthermore Z learn path to X and Y. This technique is great to learn network topology and lessens overhead in the network, however the inconvenience of this strategy is that the size of the RREQ packets increments as the movement through the intermediate nodes to get to the destination node [37].

2.6.4. Ad-hoc on-demand distance vector (AODV):

2.6.4.1. Description:

The Ad hoc On-Demand Distance Vector (AODV) protocol gives dynamic, self-beginning and multihop routing between nodes wishing to build up a network. AODV enables mobile nodes to get paths rapidly for new destinations, also, does not expect nodes to keep up paths to destinations that are not in active state.

AODV enables mobile nodes to react to link breakages and changes in network in a convenient way. The process of AODV is loop free, and by keeping away from the Bellman-Ford " counting to infinity " issue offers snappy combination when the ad hoc network topology changes (ordinarily, when a node moves in the network). At the point when link break, AODV makes the influenced set of nodes be notified so that they can delete the path that use the lost link [38].

One distinctive component of AODV is its utilization of destination sequence number for each path. The destination sequence number is made by the destination to be incorporated alongside any path information it sends to requesting nodes. Using destination sequence numbers guarantees loop freedom and is easy to program. Given the decision between two paths to a destination, a mentioning node is required to choose the one with the highest sequence number.

To find routes, the AODV routing protocol uses a reactive approach and to identify the most recent path it uses a proactive approach. That is, it uses the route discovery process similar to DSR to find routes and to compute fresh routes it uses destination sequence numbers [38].

2.6.4.2. AODV route discovery:

In this stage, RREQ packets are transmitted by the source node in a manner like DSR. The parts of the RREQ packet incorporate fields, for example, the source identifier (SId), the destination identifier (DId), and the source sequence number (SSeq), the destination sequence number (DSeq), the broadcast identifier (BId), and Time to live (TTL) [39].

At the point when a RREQ packet is gotten by an intermediate node, it could either forward the RREQ packet or create a Route Reply (RREP) packet if there is an accessible path to the destination in its cache.

To confirm if a specific RREQ has just been gotten to keep away from duplicates, the (SId, BId) pair is used. While transmitting a RREQ packets, each intermediate node gives the past node`s address and its BId. A timer related with each entry is likewise kept up by the node trying to erase a RREQ packet in the event that the replay has not been gotten before it expires.

At the point when a node gets a RREP packet, the data of the past node is save in it as to forward the packet to it as the following node of the destination. This assumes a job of a "forward pointer" to the destination node. By doing it, every node has just the following node data; while in the source routing, all the intermediate nodes on the path towards the destination are saved [39].

Figure 2.10 shows a case of route discovery process in AODV. Assume that node 1 desires to send a packet to node 8 however; it has not an accessible path in its cache. It at that point starts a route discovery process by sends RREQ packets to all its neighboring nodes (2 and 3).

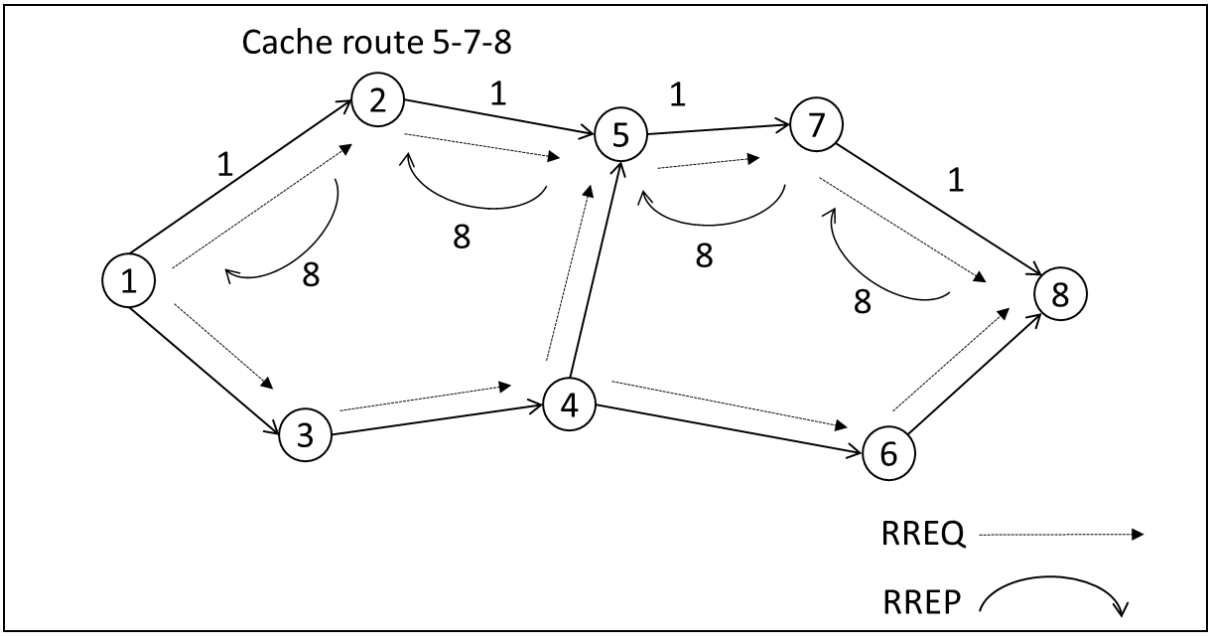

Figure 2.9. Route discovery process

All the SId, DId, SSeq, DSeq, BId, and TTL fields are embedded in the RREQ packet. When RREQ packet delivered to nodes 2 and 3, these nodes quickly scan their separate route caches for a path. For the situation where no path is accessible, they forward the RREQ to their neighbors; generally, an examination is made between the destination sequence number (DSeq) in the RREQ packet and the DSeq in its relating entry in the

route cache. It sends to the source node with a RREP packet comprising of the path to the destination for the situation the DSeq in the RREQ packet is higher. In Figure 2.7, node 2 gets a path to 8 in its cache in same way that we mentioned before for node 1, and its DSeq is more great when compared with that in the RREQ packet. Thus, it sends a RREP back to the source node1. By doing this, node 1 has just saved the path 1-2-5-7-8. A RREP is additionally sent back by the destination node to the source. The intermediate nodes on the path from source to destination update their routing tables with the most recent DSeq in the RREP packet.

2.6.4.3. AODV route maintenance:

The job of this process, that is called route maintenance, is to correct the routes when there are some changes in network topology. The nodes every period of time send a Hello Message to the neighbors' nodes to find the connections between these nodes and confirm the right routes.

On the supposition that the node moved or one of the neighbors out of the scope of transmission, it implies the connections was broken and the Hello Message not reach to the neighbors, another suspicion that there is data come to the intermediate node and this node doesn't keep up the way to the destination node, the node accepts that the route has been broken and there is a break of the connection and in this manner the node sends notice of broken link to the affected nodes.

The node creates and sends RERR message to the affected nodes expressing all destinations nodes are out of reach, where the message contains all invalid destinations, when (RERR) message reaches at the affected nodes, the node makes a comparison between its routing table and the destination in the RERR message, after that it delete the broken link routing table and proceeds with the process until the (RERR) message reaches every single affected node. After the destination is empty from the tables, the source node builds a (RREQ) message to find another route or the node that sent the link error performs find [40].

2.6.4.4. AODV features:

1. The number of control messages is low in AODV because the uses of “on demand” feature.
2. When a new path is required, a delay time will occur due to “on demand" feature.
3. In dynamic network, AODV work in effective way.
4. The number of packets in AODV is high because the use of Hello Messages periodically to identify the neighboring nodes.
5. By using the sequence number, AODV find a solution for counting to infinity problem, and it protect the network from loops.
6. When there is a failure in any link, one RERR sent to the neighboring nodes containing a list of nodes that were affected.

2.6.5. Dynamic MANET on demand (DYMO):

2.6.5.1. Description:

(DYMO) is a Dynamic MANET dependent on the demand, it is also defined to as successor of AODV or ADOVv2, it is a combination between DSR’s characteristics and AODV’s attributes, also, it is type of on demand (reactive) protocols.

DYMO protocol attempting to give a compelling and straightforward protocol, unlike AODV protocol, there is no need for unnecessary HELLO messages; process is purely based on sequence numbers assigned to all the packets. It is a reactive routing protocol that computes unicast routes on demand or when required. It employs sequence numbers to ensure loop freedom [41].

Like all other reactive routing protocols, there are two processes should be utilized to discover paths, which are route discovery and route maintenance.

2.6.5.2. DYMO route discovery:

The DYMO route discovery is very similar to that of AODV except for the path accumulation feature. The nodes in DYMO protocol have routing tables which contain: the address of destination node, the sequence number of destination node, the next hop (the address for neighboring nodes that is part of the path) and the number of hops to reach the destination node (hop count) [42].

In this way, if the source node needs to send data to destination node, the source node looks for a path in its routing table, and if there is no path to the destination node, a RREQ message is produced and broadcast in the network. In the event that the RREQ message arrives at an intermediate node, the node adds its own address to the message "path accumulation feature" and increase the hop count by one. At the point when RREQ message arrives to the destination node, a RREP message is created and sent to the node that sent the RREQ message. Figure 2.12 shows the routing discovery process.

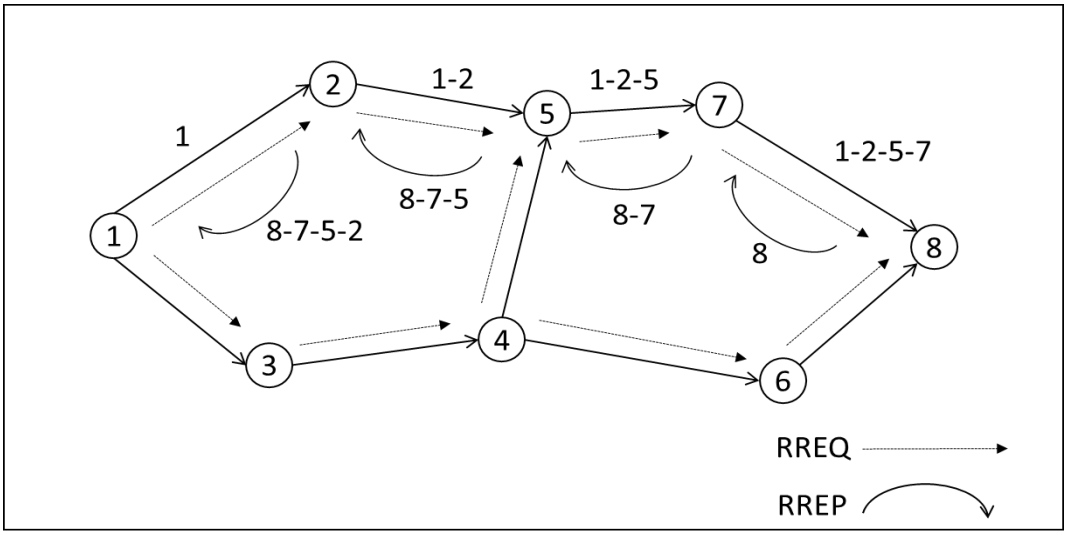

Figure 2.10. Route discovery process for DYMO and path accumulation

In Figure 2.12 node1 needs to contact node 8, so node 1 create RREQ message containing the address for node 1 and a Sequence Number for node 1, afterward increasing the hop count by one, then add the address of node 8 and hop count to reach for node 8 and its sequence number.

Along these lines, the RREQ message has data about the node 8. The message is then broadcasting to the network, with the goal that every node sends a RREQ message just once relying upon the sequence number. Every node that forwards the RREQ message adds itself to the message, for example, node address, sequence number and gateway.

Right now the message is broadcasting through the network, node 1 is thinking about getting a RREP message from the node 8. This period of time is called wait time. In the event that the wait time is finished, node 1 resumes making another RREQ message and returns the process once more.

During the broadcast, node 2,5 and 7 are added to a RREQ message. These nodes procedure the passages in the message to refresh their private information as follows. if

node 1 is absent in the routing table for every node, the address of node 1 is included with the address that the next hop Is the node from which the message received. What's more, if there is data about node 1, the values in the RREQ message are compared with the value in the routing table.

If the data in the RREQ message is new, the data is updated. The data of the nodes added to the RREQ message is likewise updated similarly. Before the intermediate nodes forward the RREQ message, it adds one to the hop count afterward makes a reverse path to node 1. At the point when the message arrives at node 8, the node forms the packet and updates its routing table.

When node 8 receives the RREQ message and update the routing table data, a RREP message containing data about the node 8 and data about node 1, for example, the two nodes addresses, sequence numbers, and gateway, and afterward are sent. The procedure is used to send RREP message similarly as the RREQ message, where the intermediate nodes are added to RREQ and the information for their routing table is updated up arrival of RREQ message to the node 1 as appeared in Figure 2.12 [42].

2.6.5.3. DYMO route maintenance:

After the discovery process and beginning the procedure of communication between nodes, the network monitoring is start to discover any changes in network topology to make sure that all paths is accessible, or find new paths. Route maintenance is very important to check and repair the links between nodes in the network. For this reason, the nodes always check every one of the connections related with these nodes then update the validity date of every path [43].

This data is included to a specific field of the routing table. If the node discovers an inactive path, the node will generate a RERR message, after that, it checks the routing tables for the destinations, which include the infected node. The affected destinations are added to a list to make notifications for all affected nodes that the paths are not available, RERR message will include the address for the infected nodes and the sequence number for that nodes and the list. At the point when the node creates a RERR message, it checks the routing table for every destination dependent on the infected node, at that point the node broadcasts RERR message to the neighboring nodes, that get this message then compare the list in RERR message with their routing table, every destination node on the

list and the similar one in the routing table is compared with each other, If there is a similarity to the destinations, so the next hop will be the affected node and the Sequence Number of the node in RERR message list is larger than or equivalent to the Sequence Number in the routing tables, the destination entries will erased.

After that the nodes will re-broadcast the RERR message if a few entries stay in the routing table. When the routing table gets empty, RERR message is dropped when it arrives at the nodes that do not use the broken link.

The explanations behind breaking the connection is the point at which the neighbor node moves out of the scope of the transmitter node or if the node moved itself or some other motivations to forbid the connection, also it is conceivable to make a RERR message if there is no update of the time entry which notify that the path is expired, also the reason for the Sequence Number of the node inside the RERR message is to ensure that the information is up to date and that no old information is being transmitted, so the route error message is broadcast only once [43].

The broadcasting of RERR message proceeds until it arrives at its destination, and in request to rediscover the path. In the event that there is a need to send information, the node generates a RREQ message for the destination and then broadcast the message to discover the Path again [43].

DYMO have an important property, which is energy efficiency, nodes in MANET have limited power source, so if the node has small amount of energy it cannot share in the route discovery process, in this case this node cannot forward the received RREQ messages.

Another important property in that DYMO routing table is generally less memory squandered than AODV, even with the "Accumulate the path" feature. Furthermore, when network size increased and the network have high mobility, the overhead for the protocol will decrease.

2.6.5.4. DYMO features:

DYMO works efficiently if the network is large and include high mobility nodes, DYMO routing table consumed less than AODV protocol, and in small size network and low mobility it doesn’t work perfectly will due to many control messages that didn't have any work which cause additional overhead [44].

DYMO accumulation path did not increase the overhead especially when there an increase in network size and the mobility.

DYMO shows execution corruption at exceptionally low traffic, and the overhead of routing outstrips actual traffic. In spite of the way, that DYMO functions admirably when traffic is routed from one part of the network to another.

2.7. Comparison between routing protocols:

Many studies made a comparison between routing protocols regarding the performance of the protocols. The results for the simulation show that reactive protocols have better performance than proactive protocols according to [45, 46].

In [46] the simulation result packet delivery ratio compared to the number of nodes, Clearly, AODV performance is better than DSDV and DSR when there is an increasing in the number of nodes as shown in Figure 2.14.

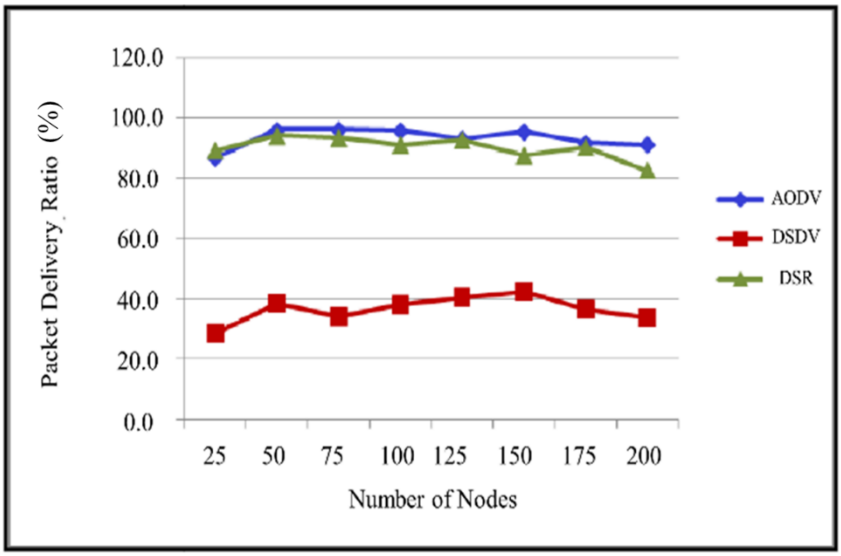

Figure 2.11. Comparison between AODV, DSDV and DSR using Packet delivery ratio Vs. number of nodes [46]

In addition, that AODV gives the throughput superior to DSDV and that it surpasses DSR as shown in Figure 2.15

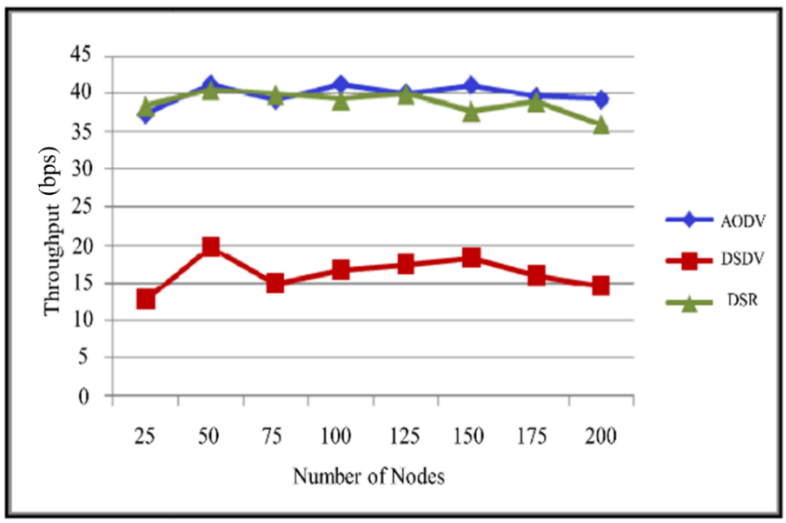

Figure 2.12. Comparison between AODV, DSDV and DSR using Throughput Vs. number of nodes [46]

In [47], it compares the performance of "on demand" protocols which are AODV, DSR and DYMO in several network situations, every one of the three protocols function admirably when the density of nodes is low, the mobility of nodes is low and there is small number of links. When the density is increased, AODV and DYMO will perform better than DSR as shown in Figure 2.16.

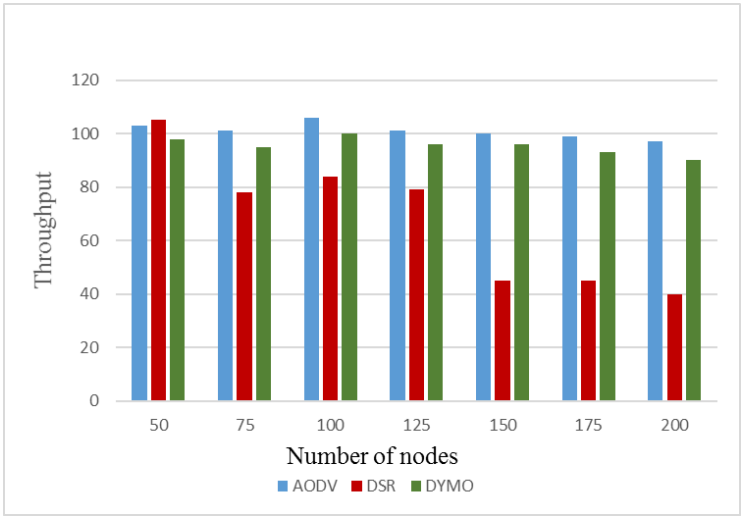

Figure 2.13. Performance comparison against increasing nodes density [47]

In [48] the simulation results show the packet delivery ratio compared to the node mobility m/s, in this comparison DYMO and AODV is almost the same as shown in Figure 2.17,

but in the results of end-to-end delay compared to the node mobility m/s DYMO performs better than AODV.

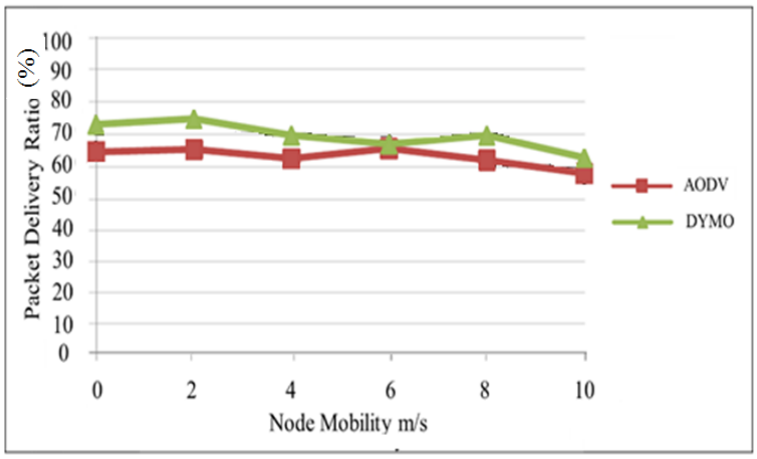

Figure 2.14. Comparison between AODV and DYMO using Packet delivery ratio Vs. Node Mobility [48]

The comparison made in network contains situation like different nodes speeds and different levels of maliciousness as in Figure 2.18

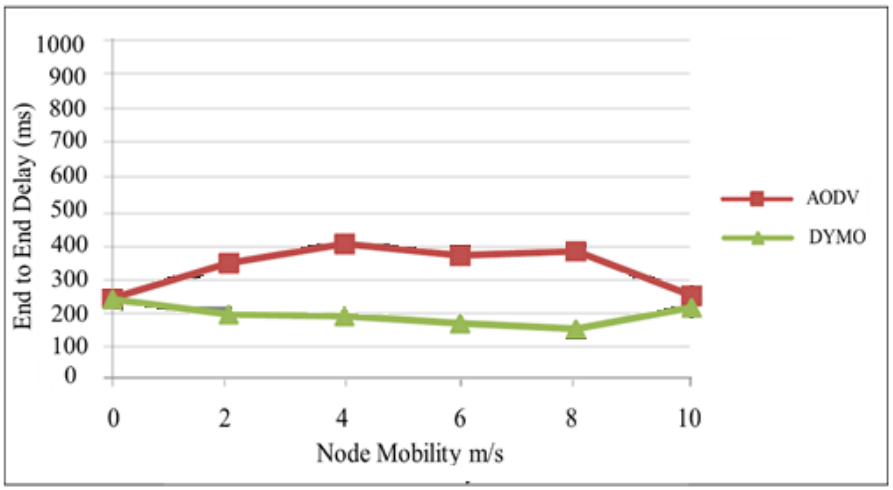

Figure 2.15. Comparison between AODV and DYMO using End to End Delay Vs. Node mobility [48]

When compare all the routing protocol together, DYMO give the best packet delivery ratio as in Figure 2.19 [64].

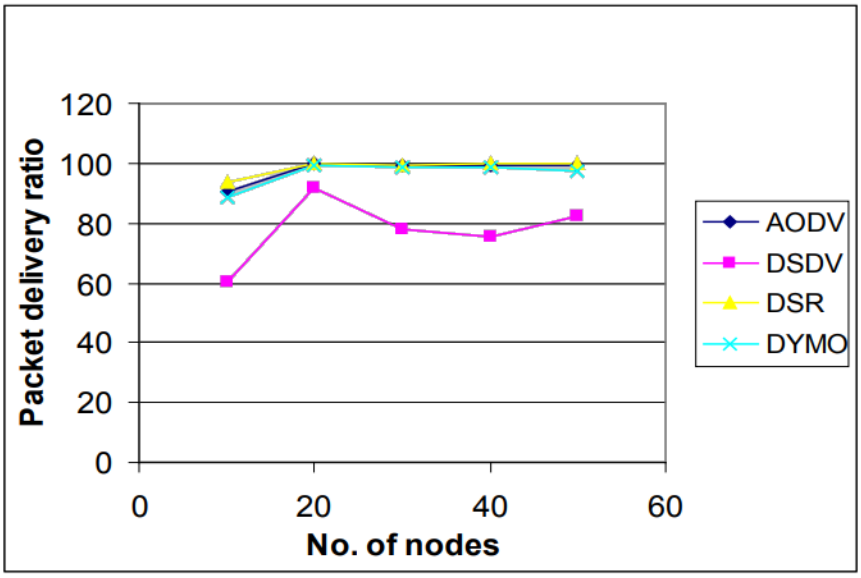

Figure 2.16. Comparison between AODV, DSDV, DSR and DYMO depends on PDR [64]

At the end, Table 2.2 shows a comparison between AODV, DSR, DSDV and DYMO Depends on the mentioned parameters.

Table 2.2. Comparison between AODV, DSDV, DSR and DYMO

Parameters	DSDV	AODV	DSR	DYMO
Protocol Type	Table driven	On demand	On demand	On demand
Average end to end delay	Less	High	High	High
Routing overhead	Less	High	High	Less
Power consumption	High	Less	Less	Less
Quality of service	Poor	Good	Good	Good

2.8. Summary:

Recently, communication means became more sophisticated and available in every field, particularly in wireless devices like laptops and mobile phones. With the expanding utilization of these devices rose the need to set up systems for these sorts of devices, so

there is a need to create new high performance protocols that protect the data in the network from any attack.

This chapter discussed the routing protocols for MANET like reactive and proactive protocols, it mentioned the properties for each protocol and made a comparison between the main protocols in each type, because of that comparison, DYMO appeared that it has the best performance and efficiency.

Chapter Three: Cryptography and Authentication techniques

3. Chapter Three: Authentication and Cryptography techniques

3.1. Introduction:

Nowadays, most of the data stored in digital format that can attack easily, so this data must be protected. Cryptography that includes encryption and decryption provides the protection for the data by using encryption to protect the content, and authentication techniques to provide the authenticity and integrity.

3.2. Authentication techniques:

Authentication is the process of determining whether someone or something is, in fact, who or what it declares itself to be. Authentication techniques give access control for network by checking if a node's credentials match that one in routing table for the neighboring nodes.

Many authentication techniques for networks are available like Certificate, Digital signature and hashing.

3.2.1. Digital signature:

A digital signature is a collection of mathematical techniques use by nodes in network, when nodes exchange any data; they need to keep the authenticity and integrity for the data, so the need for digital signature is appeared.

Digital signature scheme contains three phases: a key generation algorithm, signing algorithm and signature verification algorithm [49].

Key generation algorithm: Digital signatures use to insure that the data was sent by a specific node, so a key generation algorithm select a private key to be the unique key for that node, the private key is selected from random set of possible private keys. The algorithm outputs the private key and a corresponding public key.

Signing algorithm: The process of creating digital signature start with create a one-way hash of the data that need to sign, the signing algorithm use the private key that generated

from the first step to encrypt the hash value, so the encrypted hash value will be the digital signature, the digital signature will be added to the data then sent to the verifier [49].

Signature verification algorithm: When the verifier receives the digital signature and the data, it uses the signature verification algorithm to process the digital signature and the public key (verification key) and generates a value, then it uses the same hash function to process the received data to generate a hash value, at the end, the value that calculated from the verification algorithm and the hash value are compared. If they are equal, the digital signature is valid, else it is invalid [49].

3.2.2. Digital certificate:

Digital certificates are the credentials that facilitate the verification of identities between nodes, which proves sender's identity to the receiver and receiver's identity to the sender, it is issued by a trusted third party that called Certificate Authority (CA). The CA creates an encrypted digital certificate that includes the sender node public key [50].

Digital certificate includes:

1. Certification holders name.
2. A serial number that needs to identify a certificate, the individual or the entity identified by the certificate
3. Expiration dates.
4. Copy of certificate holder's public key
5. Digital Signature of the certificate issuing authority.

Digital certificate is sent with the digital signature and the message

3.2.3. Hashing techniques:

A hashing algorithm is mathematical algorithm that transforms data of arbitrary size to a hash of a fixed size. It was created to be a single direction function, no way to invert this function so it uses to improve the security for packets during the path of network. Hence, the message is intended for a particular recipient only and the packets will be secured against attacking.

Hashing characteristics:

1. Hash function should calculate the value in fast way to improve the efficiency for that function.
2. It should be impossible to regenerate the data from the hash value (one-way function).
3. Each message must have its own hash value to avoid duplication.
4. Any small changes in data will change the hash value and it will produce a completely different hash value, which called avalanche effect.

There are many techniques to generate the hash value; the most important techniques are Message Digest 5 (MD5), Secure Hash Algorithm 1 (SHA-1) and The RIPE Message Digest 160 (RIPEMD-160) Algorithm.

3.2.3.1. Message Digest 5 (MD5):

The MD5 is a traditional model where security is supported to the detriment of speed. It is created by Ron Rivest who is the "R" in the RSA (Rivest-Shamir-Adleman) public key encryption algorithm.

3.2.3.1.1. MD5 hashing process:

The process of MD5 is to transform a message with variable length to fixed length message equal to 128-bit.

The original message M is divided into 512-bit blocks (16 words that have 32-bit), the length of message must be divisible by 512, so the algorithm uses padding to meet that condition. Padding is done by adding single “1” bit followed by the amount of “0” bits to produce a message have length in bits congruent to 448 modulo 512 [51] Figure 3.1 show the length of message after padding.

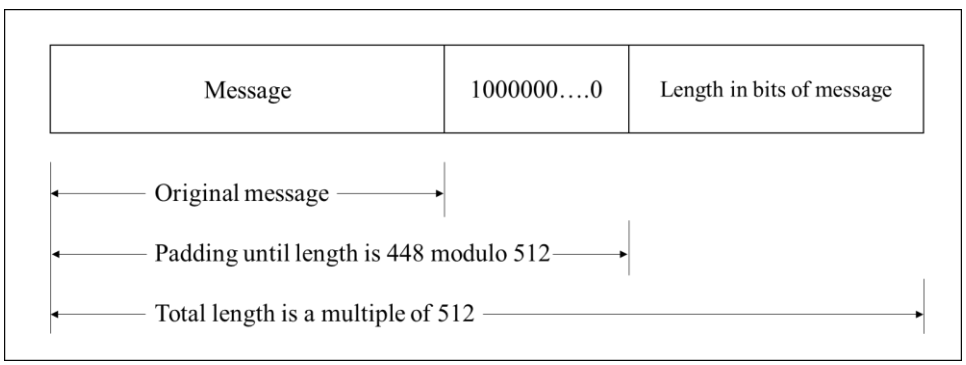

Figure 3.1. Message length after padding

3.2.3.1.2. Security of MD5:

The most important characteristic for MD5 is that each bit of the output is a function of every bit in input, the probability to find two messages that have the same hash result is too small because it requires 2^{64} mathematical operations as Ron Rivest (The authored of the MD2, MD4, MD5 and MD6 cryptographic hash functions) claimed, also he claimed that finding a message given its corresponding message digest will take 2^{128} mathematical operations [52].

3.2.3.1.3. Attacks on MD5:

As mentioned before, MD5 is one-way cryptographic function and there are three main classes of attacks on one-way cryptographic function, which are: pre-image attacks, second pre-image attacks and collision attacks. All types of attacks that target one-way cryptographic functions are distinguished under one of the three classes, because of that, one-way cryptographic function must be resistible against the three classes of attacks.

Here are some of main attacks on one-way cryptography function:

1. Collision attack in MD5 cryptographic hash function:

This attack uses differential cryptanalysis that needs 2^{39} mathematical operations to find a collision, then after many improvements that done by many researchers it was reduced to 2^{29} operations. The main idea for this attack is to find two input strings of a hash function that produce the same hash result [53].

2. Pure Brute-force Attack:

This attack is one that tries all possible words of determined length until it finds the correct one, so the length for hash result must not be too short to make it harder for brute force attack to find the result and to make it too slow.

3.2.3.2. Secure Hash Algorithm 1 (SHA-1):

The Secure Hash Algorithm (SHA) was developed by the National Security Agency (NSA), it is similar to MD5 in the way of divide the message into blocks, but different from MD5 in the way that the message divided because MD5 produce 16 word blocks but SHA-1 produce 80 word blocks.

SHA family includes five types: SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512.

3.2.3.2.1. SHA-1 hashing process:

The process of SHA-1 is to transform a message with maximum variable length of 2^{64}-1 to fixed length message equal to 160-bit.

The original message M is divided into 512-bit blocks (16 words that have 32-bit) using the compression function, the compression function consists of four rounds; each round includes a sequence of twenty steps. A complete SHA-1 round includes eighty steps where a block length of 512 bits is used together with a 160-bit chaining variable to finally produce a 160-bit hash value. The length of message must be divisible by 512, so the algorithm uses padding that mentioned before in MD5 [54].

3.2.3.2.2. Security of SHA-1:

Pre-image resistance: To produce a message from given digest message, it will face a difficulty equal to the order of 2^{160} operation.

Second pre-image resistance: To produce two distinct messages that have same digest message, it will face difficulty equal to the order of 2^{80} operations.

3.2.3.2.3. Attacks on SHA-1:

It is known that SHA-1 produces a 160-bit message digest. SHA-1 is considered secure If one cannot find collision in less than 2^{80} operations. But a group of Chinese cryptographers [55] were able to find collisions in SHA-1 in 2^{69} calculations. This is about 2000 times faster than a brute-force search attack [55].

3.2.3.3. The RIPE Message Digest 160 (RIPEMD-160) Algorithm:

In 1992, RIPEMD was designed by the European RIPE (RACE Integrity Primitives Evaluation) project. It was based on MD4 algorithm and produced a 128-bit hash value, the main difference between RIPEMD and MD4 is RIPEMD's compression function consists of two slightly modified versions of the MD4 compression function, which are executed, in parallel.

3.2.3.3.1. RIPEMD-160 hashing process:

The process of RIPEMD is to transform a message with variable length to fixed length message equal to 160-bit.

The original message M is divided into 512-bit blocks at a time using the compression function, the compression function consists of two parallel trails of a modified version of MD5 algorithm, each trail includes five rounds, each round have a sequence of 16 steps. A complete RIPEMD round includes eighty parallel steps where a block length of 512 bits is used together with a 160-bit chaining variable to finally produce a 160-bit hash value [56].

3.2.3.3.2. Security of RIPEMD-160:

Pre-image resistance: To produce a message from given digest message, it will face a difficulty equal to the order of 2^{160} operation, same as SHA-1.

Second pre-image resistance: To produce two distinct messages that have same digest message, it will face difficulty equal to the order of 2^{80} operations.

3.2.3.3.3. Attacks on RIPEMD-160:

Until now, no successful attacks against RIPEMD-160 [57] is known, but the designers of RIPEMD160 think that soon it will become possible to attack one of the two lines and up to three rounds of the two parallel lines.

3.2.3.4. Comparison between hashing techniques:

The comparison of the standard hash algorithm based on the general properties, including block size, word size, output size, logical operation, and the number of rounds as shown in Table 3.1.

Table 3.1. Comparison between MD5, SHA-1 and RIPEMD-160

Properties	Name of algorithm		
	MD5	SHA-1	RIPEMD-160
Block size	512 bits	512 bits	512 bits
Word size	32 bits	32 bits	32 bits
Output size	128 bits	160 bits	160 bits
Number of steps	64 (4 rounds of 16)	80 (4 rounds of 20)	160 (5 rounds of 16)
Operations	ADD,XOR, AND,OR, NOT, SHIFT	ADD, XOR AND, OR,NOT, ROTATE.	ADD,, ROTATE, XOR,AND, OR,NOT
Speed	High	Low	Low
Number of operations to create the digest message	2^{128}	2^{160}	2^{160}
Complexity	Low	High	High
Security, Number of operations needed by assailant to find the original message	2^{64}	2^{80}	2^{160}

Generally, MD5 is faster than SHA-1 and RIPEMD-160, but RIPEMD-160 provide more security than the other [58], also MD5 generates a digest message with less number of bits in comparison with SHA-1 and RIPEMD-160 which make it better for use in case of the digest message needs to be included in packet header.

3.3. Cryptography techniques:

Cryptography techniques include two main processes: encryption process and key management process.

3.3.1. Encryption techniques:

Encryption is the process of encoding a data in such a way that only authorized members can access it by decrypt that data, to encode the data, the sender and receiver need to share a key, and the key distribution process have two main categories: symmetric (secret) key and asymmetric (public) key.

Symmetric key cryptography depends on sharing the same key between the parties like in Advanced Encryption Standard (AES) algorithm and Data Encryption Standard (DES) algorithm. These algorithms are fast but the problem is that they need to key-exchange process between the sender and receiver only without share it with the entire network.

The AES is used to provide security for sensitive data, and it is based on Substitution and Transposition methods. The AES is used in many password-protected documents and wireless communications such as wireless sensor networks, and in top secret government files for which it was first built. This algorithm takes the input data block of size 128 bit and a variable key size of 128, 192 or 256 bits for 10, 12 or 14 rounds respectively. Each round consists of several processing steps, including the encryption step itself. Similarly, a set of reverse rounds are performed to transform cipher text back into plaintext, the pictorial representation of the AES encryption process to encrypt 128-bit in plaintext to 128-bit in cipher text. When the plaintext size is more than 128-bits, it will be divided into blocks of 128-bit plaintext as shown in Figure 3.2.

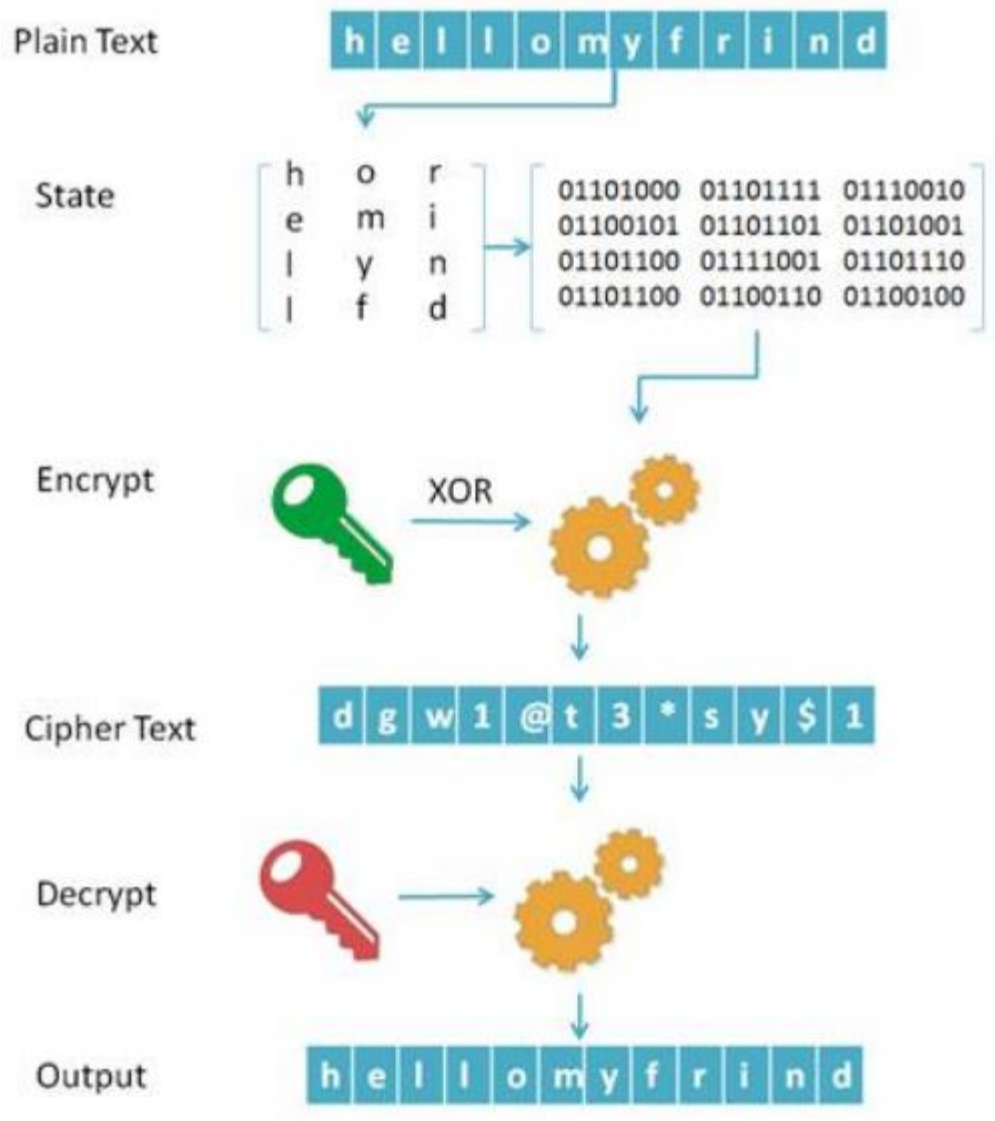

Figure 3.2. AES Encryption / Decryption process example [65]

In such a situation, AES encryption will be done for each block separately. So, the weak part of the algorithm is the secret key; therefore, it should be motivated to do some processing to give more security to this key.

Asymmetric key cryptography allows the sender to broadcast a public key in the network that can used to encrypt the data, but only the receiver can use it for decryption. Diffie-Hellman key exchange is an example of that type of key distribution [59].

The problem in this cryptography that it is much slower than symmetric key cryptography, but it can be solved by use the public key in creating a secret key than can be used for creating symmetric key, this process is used in Diffie-Hellman key exchange.

3.3.2. Key management:

There are many techniques to manage the secret key distribution in encryption process, most of them need trusted third party to distribute that key, MANET doesn't have that third party, so it needs a unique technique that can manage the key distribution in these circumstances which is Diffie-Hellman key exchange.

3.3.2.1. Diffie-Hellman Key exchange:

Diffie-Hellman key exchange is a method of exchanging keys, it allows two parties that did not have prior knowledge of each other to cooperate and create a shared secret key over insecure environment, and then this key can be used to encrypt data using any encryption technique [60].

The main characteristic of Diffie-Hellman is that no need for third party to distribute the secret key, the two parties exchange their public keys over the network, then the Diffie-Hellman algorithm will generate an identical key that is difficult to compute by network elements. The two parties will use this shared secret key to encrypt and decrypt the data between them.

3.3.2.2. Diffie-Hellman Algorithm:

Here is an explanation for the Diffie-Hellman key exchange process in points:

a. Source (S) and Destination (D) agree on Prime Number (P) and Generator of the prime number (G).
b. S randomly chooses private key X_S and D randomly chooses private key X_D.
c. S calculates Y_S, which is equal to G^{Xs} mod P.
d. S sends Y_S to D.
e. D calculates Y_D, which is equal to G^{Xd} mod P.
f. D sends Y_D to S.
g. S computes $K_S = Y_D^{Xs}$ mod P.
h. D computes $K_D = Y_S^{Xd}$ mod P.
i. Then K is the shared secret key.

After this process S uses K to encrypt the data and send it to D, when D receives the encrypted data it can decrypt it using K, Figure 3.3 shows an example for Diffie-Hellman key exchange process [61].

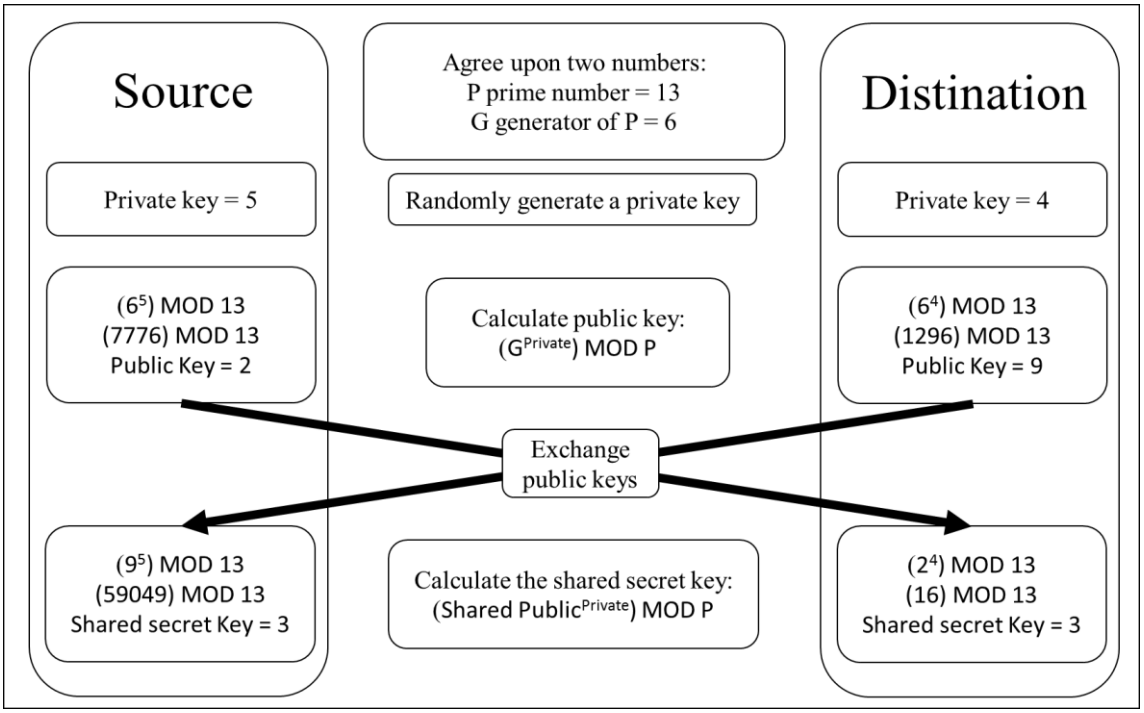

Figure 3.3. Diffie-Hellman key exchange process example

3.4. Reinforcement learning:

Reinforcement learning is an area of machine learning; it is about deciding to take the right action to maximize the rewards in a specific situation. In reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of training dataset, it is bound to learn from its experience (i.e., unsupervised leaning) [62].

As an example, Figure 3.4 shows an agent and reward, if the agent wants to take the rewards it must make its way between many hurdles by finding the best route to reach the reward.

The Figure shows robot, diamond and fire. The robot is the agent, the diamond is the reward and the fire are the obstacles in the route. The robot aims to get the diamond and bypass the fire. By trying all possible routes, the robot will have an experience to get the diamond with the least obstacles. Any right step will give the robot an extra reward. Also any wrong step the robot will lose a reward. At the end, when the robot reaches the diamond, the total reward will be calculated.

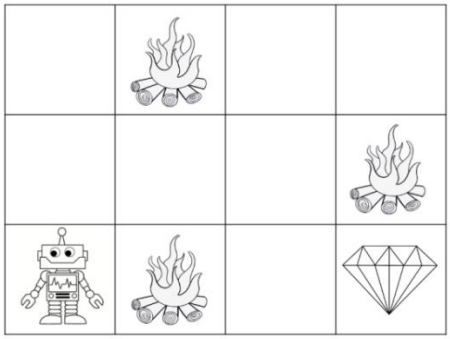

Figure 3.4. Reinforcement learning example

Main points in Reinforcement learning procedure [63]:

- Precondition: the input must be the initial state where the model will begin.
- Post condition: there are many solutions for any problem, which called the output.
- Initial state rewards (R) = 0.
- Training: the training is depending on input, after the model generates the output, the user can make a decision to reward or punish the model depending on the output result.
- Every time the user takes a decision, the model continues to learn.
- The best route will depend on the maximum reward.

Types of Reinforcement: There are two types of Reinforcement:

Positive: positive reinforcement is giving a motivating action to the user when he takes the right decision; this will make this decision more likely to happen in the future.

Negative: negative reinforcement happens when the user takes the wrong decision; the result will be removing some stimulus or may be apply some punishments (penalties) on the use.

3.5. Summary

The unique environment of MANET needs a special technique to secure the data that transacting between the nodes, so as mentioned before, MD5 will be the best choice for hashing data in MANET even though SHA-1 and RIPEMD-160 are more secure, but MD5 gives a moderate security with fast hashing, less bandwidth usage and less memory consumption.

The use of hashing in MANET is to provide the needed authentication between nodes, because most of authentication techniques like digital signature and digital certificate are need to use trusted third party, which is not available in MANET environment.

To replace the trusted third party in key generation process, Diffie-Hellman can generate a private key to use it in encryption process.

Chapter Four: The proposed model for Reinforcement Authentication DYMO Protocol (RAD)

4.1. Introduction

4.2. System Architecture

4.3. Algorithms

4.4. Mathematical Model

4.5. Summary

4. Chapter Four: The proposed model for Reinforcement Authentication DYMO Protocol (RAD)

4.1. Introduction:

MANETs consist of group of mobile nodes that communicate together in special environment that did not have a fixed infrastructure or central server that can manage the communication between the nodes. The mobile nodes are electronic devices like mobile phone or laptops, these devices work with limited memory and finite power source that will create limitations in using these devices.

The connection between source node and destination node will established by cooperation with intermediate nodes that found between them. These intermediate nodes have free mobility and can exit the network in any time, which will cause a break in the path between source and destination. MANET routing protocols solve this problem by finding new paths with new intermediate nodes.

The data will transmit via anonymous intermediate nodes, and it is hard to guarantee the security for this data, so there is a need for a protocol that arranges the transmission process and monitors the behavior for the nodes in MANET.

In this chapter, Reinforcement Authentication DYMO protocol (RAD) will be proposed. Depending on encryption and authentication techniques for securing the data, and reinforcement learning to improve the cooperation between nodes in MANET that will increase the performance and the efficiency for the network.

RAD protocol works in network layer in OSI model, which contains Physical, Data link, Network, Transport, Session, presentation and application layers.

The proposed protocol is based on DYMO protocol for route discovery and maintenance processes, and has two main phases: The Authentication phase, which is carried out by MD5 hashing and Diffie-Hellman algorithm for key management and it will use AES for Encryption process. The Reinforcement learning phase, which is carried out by rewards and punishments principles.

4.2. System Architecture:

Our (RAD) proposed protocol consists of the following:

- **Source node:** Which is the node that wants to send a message to another node (Destination node).
- **Authentication:** In this phase, two main processes are carried out, hashing the MAC address for nodes that will participate in the transmission process using MD5 hashing algorithm, the second process generates and distributes the private key that will be used in AES process for encryption by using Diffie-Hellman key exchange algorithm and this will be found in the sink node.
- **DYMO protocol:** Is the utilized protocol to find the route that the message will pass to reach the destination node.
- **RRER:** The route request error will be the trigger to start the route maintenance process and apply the punishment for the malicious node.
- **Reinforcement learning:** In this phase, two main actions are (carried out) performed, punishment for the malicious and selfish node by exclude it to revocation list, and rewards for the nodes that proves expected good behavior for many cycles.
- **Destination node:** Which is the node that will receive the message from Source node.

Figure 4.1 shows the proposed protocol architecture. The next section in this chapter will introduce the algorithms for the two phases to explain the workflow of the proposed protocol architecture.

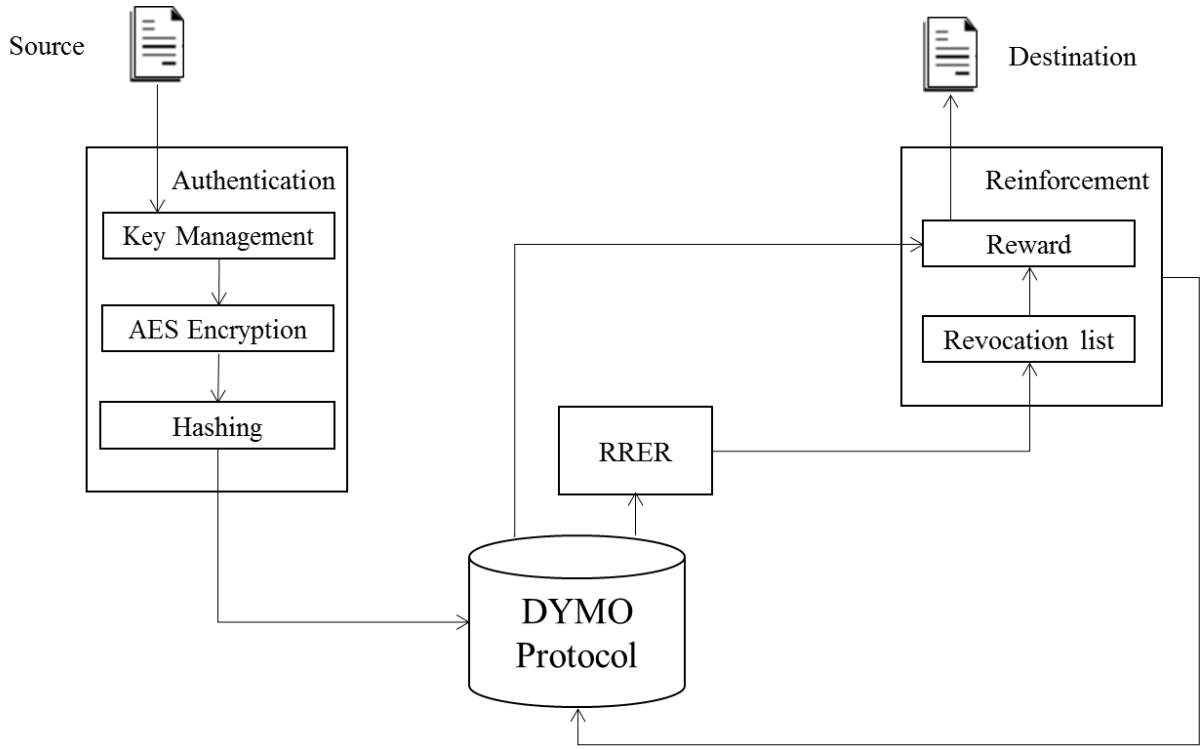

Figure 4.1. The Proposed Protocol Architecture

4.3. Algorithms:

In this section, we will demonstrate the algorithms for our proposed protocol. Our proposed protocol (RAD) has two main phases: Authentication phase, and reinforcement learning phase.

4.3.1. The Authentication phase:

It has nine sequential steps to create safe transmission process. The steps of this phase are as follows:

Step 1: Source S needs to send a message to Destination D, S generate Prime Number P and Generator of the prime number G, and then S Broadcasts P and G in the network.

Step 2: S generates private key X_S, and calculates Y_S, which is equal to G^{Xs} mod P, and then attaches Y_S to header of RREQ packet and sends it to D.

Step 3: The route discovery process in DYMO protocol will start to determine the best route to D by broadcasting RREQ to the network.

Step 4: D generates private key X_D, and calculates Y_D, which is equal to G^{Xd} mod P, and then attaches Y_D to RREP packet and sends it back to S using the discovered route.

Step 5: S calculates encryption key K_S that will be 128-bit size, which is equal Y_D^{Xs} mod P.

Step 6: AES divide the packet into blocks with 128-bit size and an XOR that block with K_S to get the encrypted message.

Step 7: MD5 hashing algorithm will use MAC Address for S and D to generate hash value to check the authentication for end-to-end transmission, and generate another hash value by using the MAC Address for intermediate nodes to check the authentication for node-to-node transmission.

In Source node, MD5 will make a summation for S and D MAC addresses and generate a hash value, this process will repeat in D, the hash values that generated in S and D will compared, if any changes appear between the two hash values, D will define as malicious and will be added to revocation list, this list contains the nodes that showed odd behaviors, this list will be found in the node itself. Same process will apply between node to node in the route, but with the of the IP address.

We use MAC Address in hashing between S and D, because it is fixed and unique for each node

Step 8: The hash values will attach to header of the encrypted message and send it to the discovered route.

Step 9: Every time the message reaches for one of the intermediate nodes, the hash value for this node will compared with the hash value for previous node, if any change is detected, then RRER will generated and the second phase will start and path discovery process will start again.

Step 10: When D receives the message, the hash value between S and D will compared to make sure that D is authenticated, then D will calculate K_D, which is equal Y_S^{Xd} mod P, and uses it to decrypts the message using AES in case that $K_D = K_S$, else the node will define as malicious node.

Figure 4.2 shows a flowchart explaining the steps of the Authentication phase.

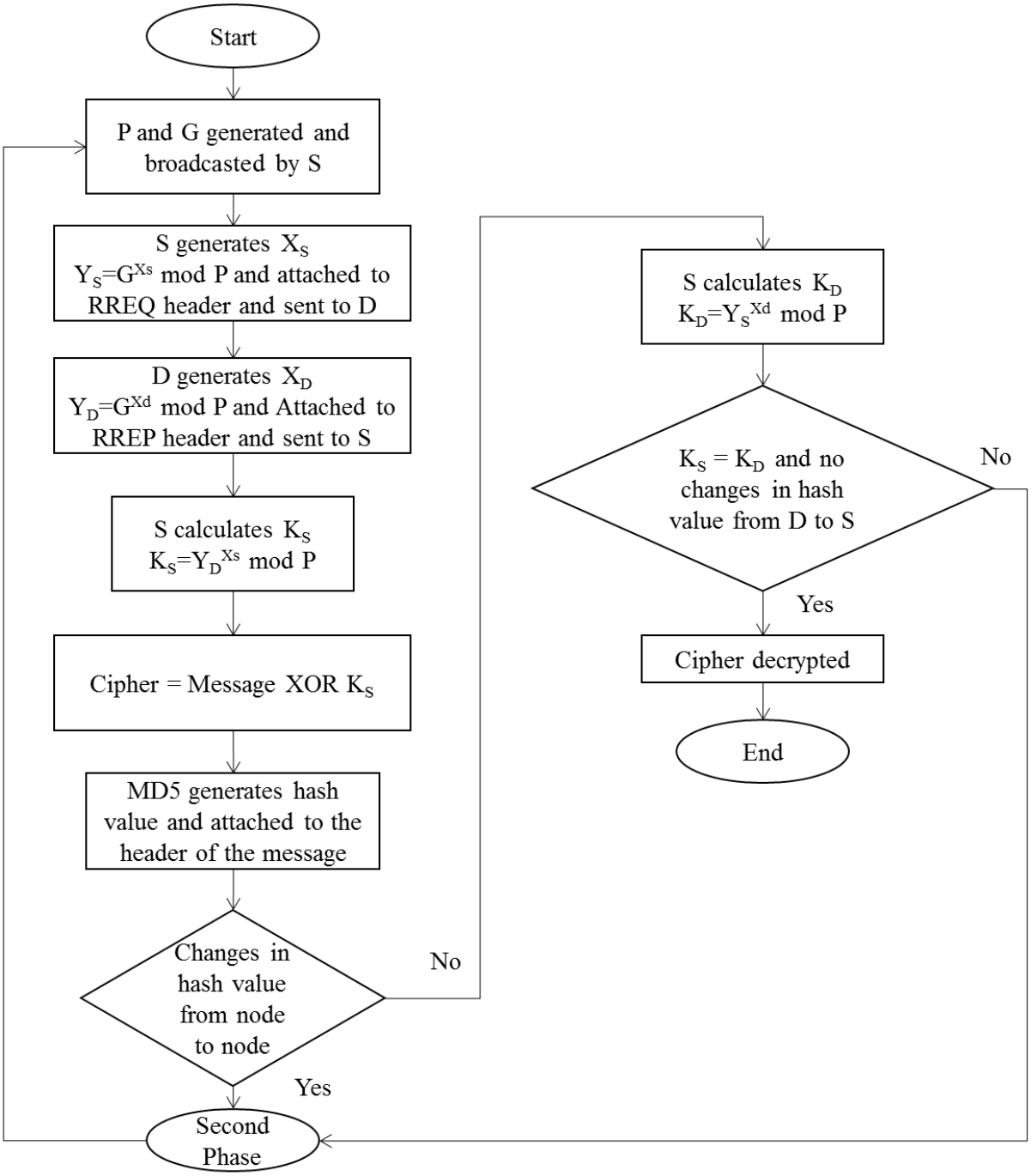

Figure 4.2. Flowchart for the Authentication Phase

Now, we will show the pseudocode for Authentication phase that summarize the process in algorithm 4.1.

Algorithm 4.1. Authentication phase algorithm

Procedure begin

S select (P,G,X_S)

For node S do

$Y_S = G^{Xs} \bmod P$

//S broadcast P,G

//RREQ include X_S

While (Message from S){

If (no route exits to D)

Broadcast RREQ

End If

If (RREQ is received by intermediate node)

Update routing table

Forward the RREQ

End If

If (RREQ is received by D)

D select (X_D)

For node D do

$Y_D = G^{Xd} \bmod P$

Create RREP include Y_D

End If

If (RREP is received by intermediate node)

Update routing table

Forward the RREP

End If

If (RREP is received by S)

Route is ready

End If

}

End While

For node S do{

$K_S = Y_D^{Xs} \bmod P$

H_{SD} = MD5 (S,D MAC Address)

}

For Intermediate Nodes N do {

H_N = MD5 (N,N+1 IP Address)

N++

}

While N != D

//Message M encrypted using K_S by AES encryption and generate Cipher C

C = M XOR K_S

//M header include H_{SD},H_N

IF (H_N = H_{N+1})

N Forward C to N+1

Else {

RREQ is created

Reinforcement learning (N+1)

}

D receive C

For node D do

K_D = Y_S^{Xd} mod P

H_{DS} = MD5 (S,D MAC Address)

IF (K_S = K_D && H_{SD}= H_{DS})

C Decrypted by AES encryption and generate M

Else {

RREQ is created

Reinforcement learning (D)

}

Procedure End

4.3.2. The Reinforcement learning phase:

It has six sequential steps to deals with malicious nodes. The steps of this phase are as follows:

Step 0: R is the rank number for the malicious node, when any node enters the revocation list, R for this node will be equal 0.

Step 1: When any node generates RRER, the next node will be recognized as malicious node, and then it will be added to revocation list.

Step 2: The malicious node will exclude from the previous route, but it has chance to participate in new route.

Step 3: every time the malicious node gives a good behavior in another route, R will be incremented.

Step 4: We test performance of protocol when R equal 1,3 and 5, then we find that R=3 will be the best value the cause less delay and more throughput as shown in Figures 5.1 and 5.2, so If R = 3, then the malicious node will exclude from the revocation list and will be recognized as normal node.

Figure 4.3 shows a flowchart explaining the steps of the reinforcement learning phase.

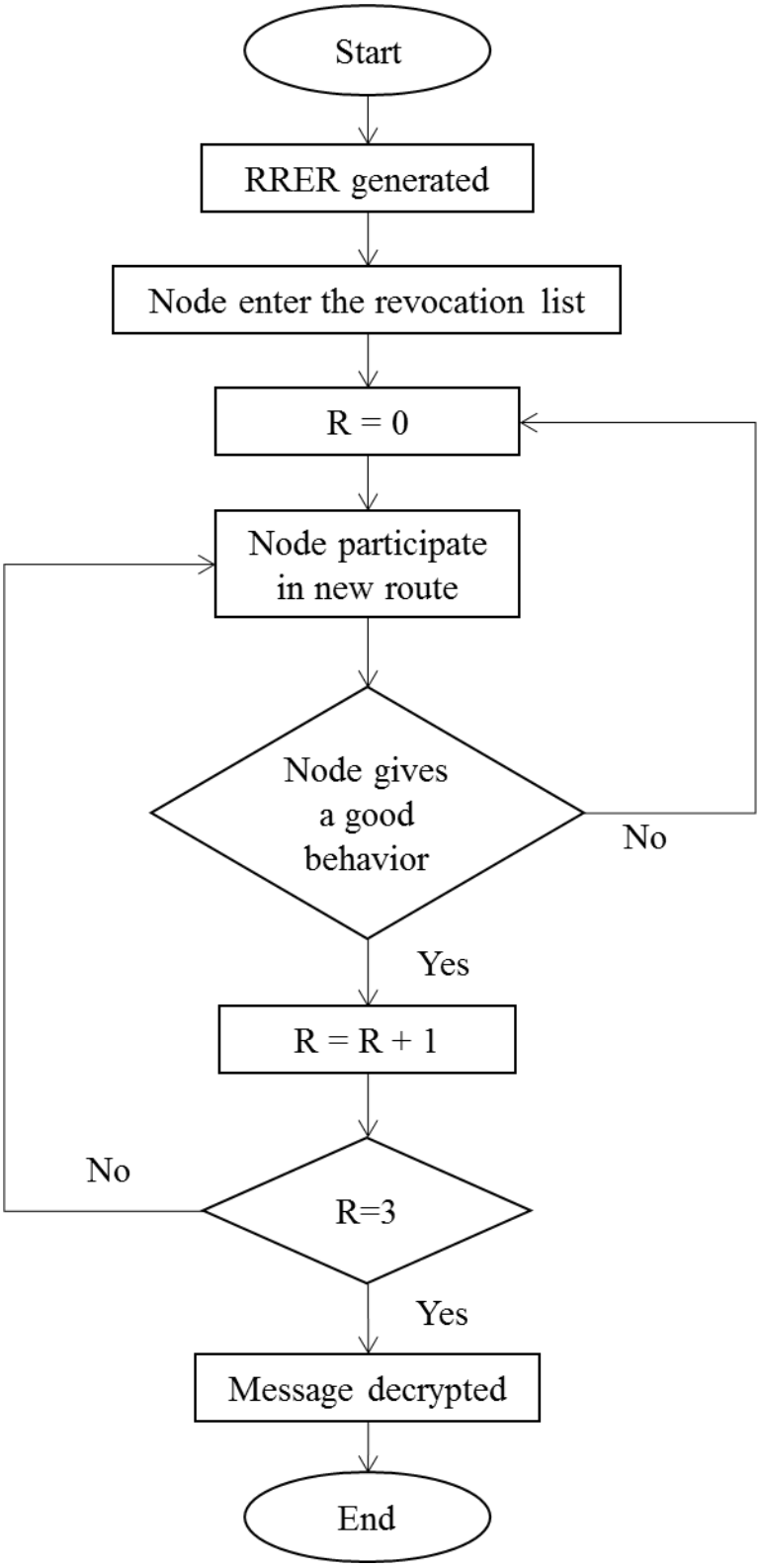

Figure 4.3. Flowchart for reinforcement learning phase

Now, we will show the pseudocode for Reinforcement learning phase that summarize the process in algorithm 4.2

Algorithm 4.2. Reinforcement learning phase algorithm

Procedure begin

RRER created for N

Malicious Node = N_M

Revocation List = RL

//RL include N_M

For N_M do

Rank R = 0

For Authentication (N_M) do{

If Authentication (N_M) = Good Behavior

R = R+1

Else If Authentication (N_M) = Malicious Behavior

R = 0

}

While R = 3

If (R=3)

N_M=N

\\RL exclude N_M

Procedure End

4.3.3. Sequence diagram for proposed model:

The UML will be used to illustrate the interactions between the two phases for the model. Figure 4.4 presents the sequence diagram model to show the interactions of the proposed model.

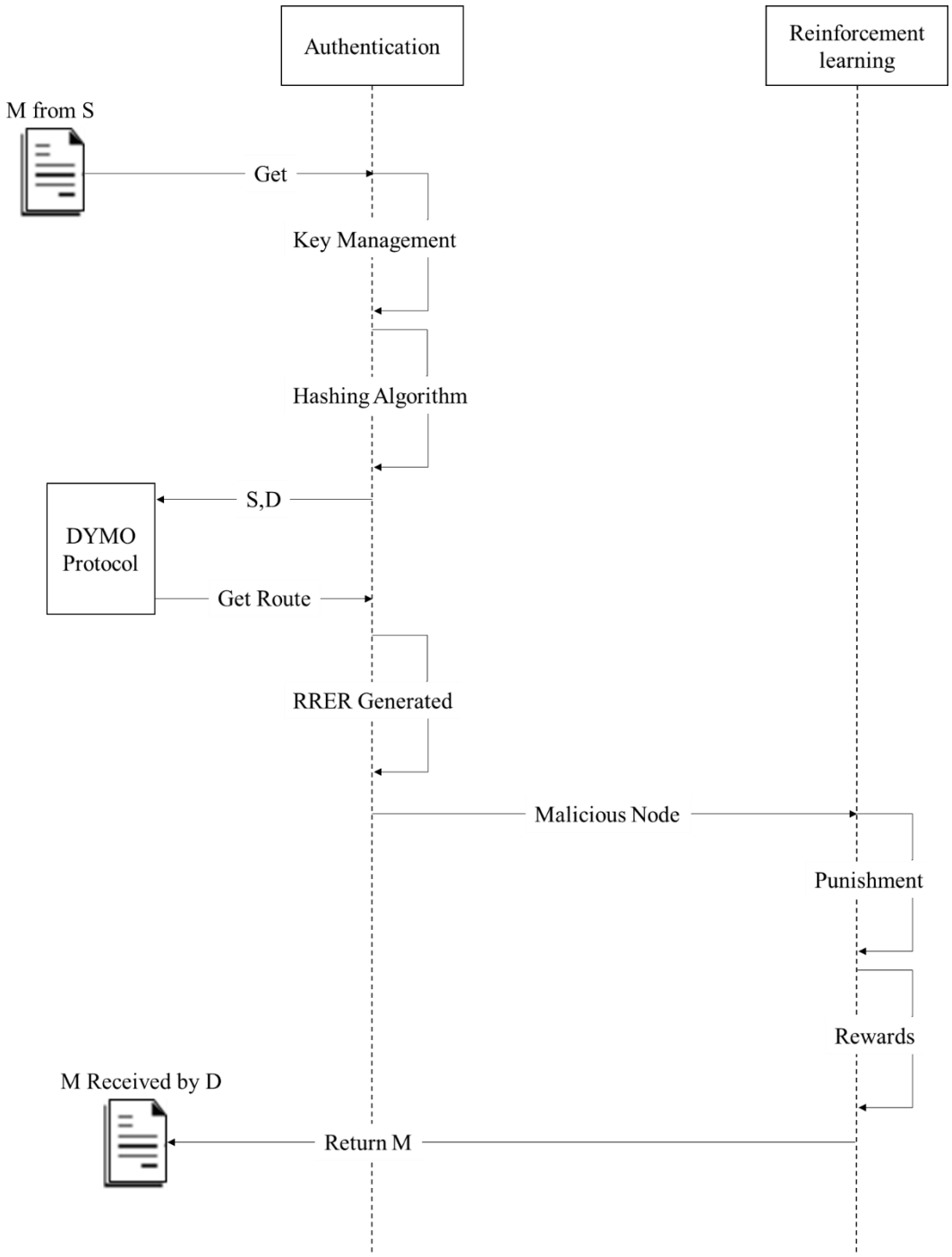

Figure 4.4. The System UML Sequence Diagram

4.4. Mathematical Model for reinforcement learning:

In this section, we will present the mathematical model of our proposed protocol; we will use this model to calculate the rewards and the maximum utility that node can get from using the reinforcement learning in DYMO protocol.

The reward for taking action a in state st is $R(st,a)$:

$$R(st,a) = \sum_{st=0}^{st'} P(st'|st,a)r(st,a,st') \quad (4.1)$$

Where:

- ST is set of states of paths.
- A is set of actions a (like selection of the path in DYMO).
- $P(st'/st,a)$ is the probability of transmission from st to st' given action a (success in transfer data to neighbor nodes in the DYMO path)

The maximum utility for taking action a in state st is $U(st)$:

$$U(st) = max(R(st,a) + \sum_{st=0}^{st'} T(st,a,st')U(st')) \quad (4.2)$$

Where:

- $T(st,a,st') = \sum_{st=0}^{st'} P(st'|st,a)$ (4.3)

4.5. Summary:

In this chapter, the new RAD protocol is presented. Starting with an introduction, which gives a clear idea about the protocol. Then in second section, the architecture model is illustrated.

In the third section, the algorithms and flowcharts are introduced to explaining processing flow, in addition, by using UML we represent the interactions between the phases in our proposed protocol. After that the mathematical equations that is used in reinforcement learning phase is provided.

Our protocol was represented by different methods to understand and simplify the overview of this proposed protocol. And also to prepare the proper implementation to be done in simulation chapter which is the next chapter.

Chapter Five: Simulation and results

5.1. Introduction

5.2. Simulation tool

5.3. Simulation environment and parameters

5.4. Performance metrics

5.5. Simulation result

5.6. Statistical analysis

5.7. Security analysis

5.8. RAD performance vs previous protocols

5.9. Summary

5. Chapter Five: Simulation and results

5.1. Introduction:

In this chapter we will present the simulation for the proposed protocol, this model is implemented from DYMO protocol, so we will make a comparison between the proposed protocol and DYMO protocol depending on performance metrics to check the improvement that happen on DYMO protocol.

To create the simulation, we need a simulation tool, there are many types of software that concern with wireless network and ad hoc networks simulation, the simulation tools give the ability to examine the new routing protocols such as NS-2, NS-3 and OPNET++, and then we get results to make a comparison with the original routing protocol.

5.2. Simulation tool:

In this thesis, we will create the simulation by using a software called “the Network simulator version two (NS-2)”, it was developed in 1989.

NS-2 is an open source event-driven simulation tool that is useful for studying the dynamic environment of networks such as Ad-hoc networks, and the traditional networks like the wired networks.

NS-2 is developed using C++ and Object-oriented extension of Tool Command Language (OTcl) as a front end, and it includes package for standardized existing protocols, this package is free and non-commercial. It is primarily Unix based but it can work with Windows and Linux.

5.3. Simulation environment and parameters:

In this thesis, we run NS-2 in windows 10 using VMware Workstation, we used the version ns2.35 on Ubuntu 12.04 LTS 64 bit operating system to simulated our new proposed protocol and compare it with DYMO protocols, all that work on computer have intel core i5 – 7200 CPU @ 2.5 GHz, the installed memory is 8 GB.

All scenarios were applied on an area simulating 3000 m x1000 m based on Table 5.1 where Mobility model is Random way point and simulation time 900s.

When the simulation begun, one node was selected as source node and another one was the Destination node. During the simulation, the selected nodes were not change.

The simulation was applied to RAD protocol depending on change the number of nodes from 10 to 100 nodes without change the speed that equal 30 m/s, and was applied when the number of nodes was 100 and the speed changed from 5-30 m/s, at the end we applied the simulation when the number of nodes is 100, the speed is 30 m/s and the time start from 0s to 900s. The timing will start from 0, and the message will generate randomly by DYMO protocol library in NS2 using Poisson Distribution.

Table 5.1. Simulation Parameters

Parameters	Value
Simulation Tool	NS-2.35
Operating System	Ubuntu 12.04
Channel type	Wireless channel
No. of Nodes	10,20,30,40,50,60,70,80,90,100 nodes
Antenna model	Omni directional
Interface queue size	50 packets
Transmission range	250m
Speed	5,10,15,20,25,30 m/s
Simulation time	900s
Mobility model	Random way point
Examined protocol	DYMO, RAD
Simulation area	3000m*1000m
Bandwidth	2 Mbps

5.4. Performance metrics:

In this simulation, we used three parameters to test the performance of the proposed protocol by made a comparison between the RAD protocol simulation results and the DYMO protocol simulation results.

a) Packet Delivery Ratio (PDR): It represents the ratio of the number of packets received successfully in destination node to the number of packets sent by source node per unit of time.

PDR = $\frac{\sum_{i=1}^{N} Ri}{\sum_{i=1}^{N} Si}$

R: Received data packets

S: Sent data packets

b) End to End Delay: It represent the average time needed to deliver data packets from source node to destination node including all delays in route.

E2ED = $\frac{\sum_{i=1}^{N} (TRi - TSi)}{\sum_{i=1}^{N} Ri}$

TR: Receiving time

TS: Sending time

R: Received data packets

c) Throughput: Is the average number of bits that successfully delivered per unit of time.

TH = $\frac{\sum_{i=1}^{N} Ri}{\sum_{i=1}^{N} TRi}$

TR: Receiving time

R: Received data packets

5.5. Simulation result:

In this part, we applied the previous environment parameters and get the results, we will show these results and discuss them.

The results compared the behavior for DYMO protocol and RAD protocol in three scenarios: performance metrics with network size (number of nodes), nodes speed and with the simulation times.

5.5.1. Results of performance metrics when value of R (Rank) equal 1,3 and 5 Vs Network size:

We applied the parameters in a network includes variable number of nodes start from 10 nodes until 100 nodes that moves in speed of 30 m/s, as we mentioned in part 4.4.2 (The Reinforcement learning phase) the malicious nodes will stay in the revocation list for R= 3 cycles, which is the best value to get less delay as shown in Table 5.2.

Table 5.2. End to end delay (ms) for variable value of R

Number of nodes	Value of R		
	1	3	5
10	141.32	136.4	139.45
20	146.31	143.25	146.76
30	157.24	151.37	154.97
40	170.33	166.49	166.11
50	163.7	163.25	166.98
60	188.15	192.81	190.05
70	199.95	192.32	195.59
80	209.27	200.23	204.8
90	211.18	207.72	209.1
100	211.74	211.43	217.43

When R=1, the number of malicious nodes in the network will increase, then the delay will increase. When R = 5, the number of malicious nodes in the network will decrease, but the delay will increase because the number of available nodes in network will decrease, then the route discovery process will take more time to find a path. Also if R>5 the number of available nodes will decrease, so we stop the test on R=5 As shown in Figure 5.1 R=3 is the best value to get the minimum delay.

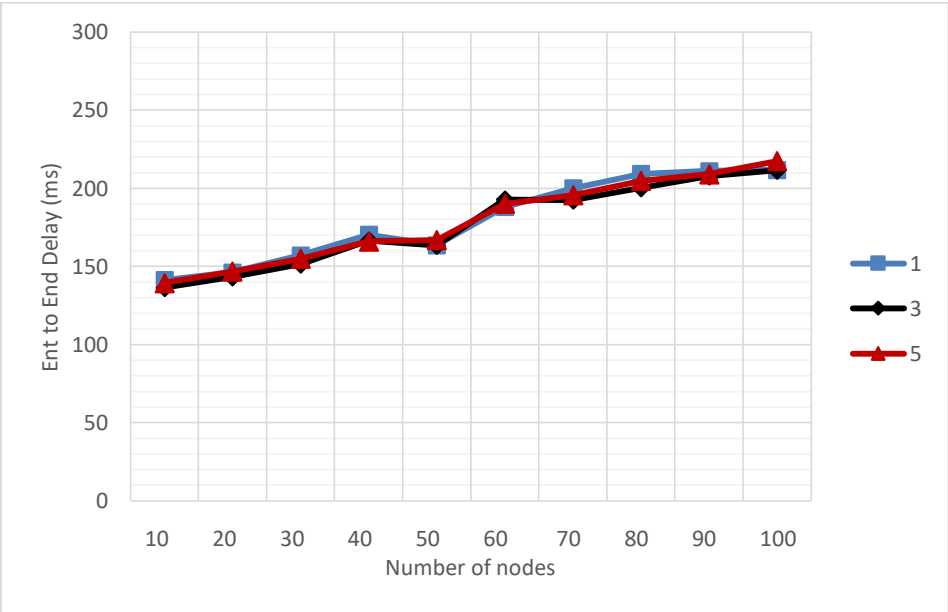

Figure 5.1. End to end delay (ms) for variable value of R

We will compare the throughput for R= 1,2 and 3 to find the best value of R to get the highest throughput as shown in Table 5.3.

Table 5.3. Throughput (Kbps) for variable value of R

Number of nodes	Value of R		
	1	3	5
10	579.21	624.47	584.73
20	568.30	609.44	570.42
30	568.08	613.53	579.31
40	540.98	593.98	565.70
50	542.29	607.27	598.35
60	562.20	596.38	557.98
70	553.36	586.83	558.89
80	494.08	533.61	528.20
90	458.68	506.17	482.07
100	578.21	624.47	584.73

When R=1, the number of malicious nodes in the network will increase, then the throughput will decrease. When R = 5, the number of malicious nodes in the network will decrease, but the throughput will decrease because the number of available nodes in network will decrease then the rout discovery process will not be effective. As shown in Figure 5.2 R=3 is the best value to get the maximum throughput.

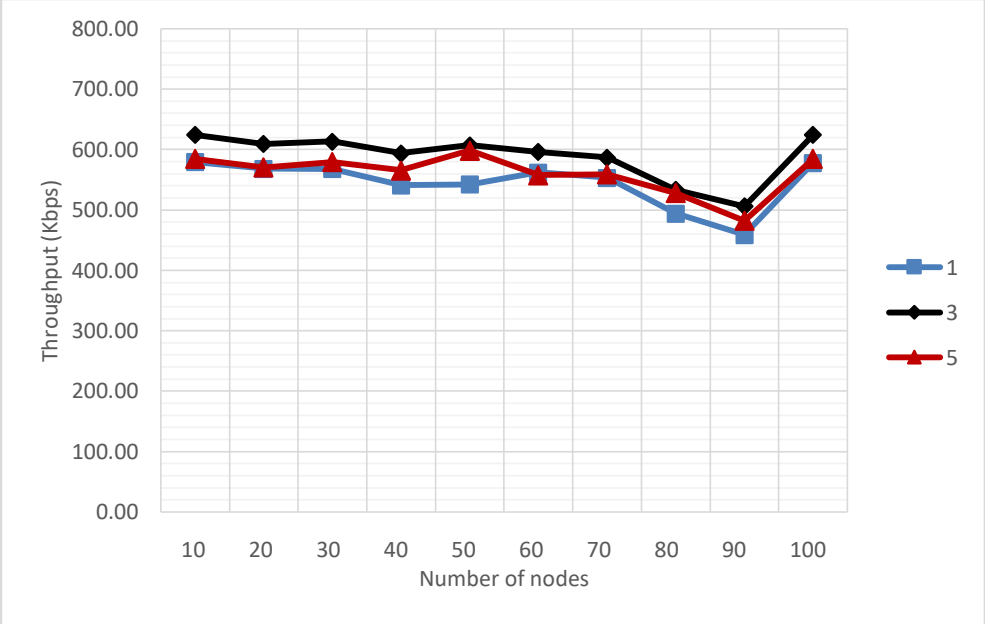

Figure 5.2. Throughput (Kbps) for variable value of R

To calculate the utilization (%) gained from R = 1,3 and 5, we calculate the average throughput for the three values of R and divided it by bandwidth that equal 2Mbps as mentioned in simulation parameters because NS2 use simple form of IEEE 802.11 standard.

U_{R1} = Average throughput for R1 / Bandwidth

= (544.539 k / 2M) * 100 %

= 27.22 %

U_{R3} = 29.48 %

U_{R5} = 28.05 %

Figure 5.3 shows that the maximum utilization gained when R = 3, so in all scenarios we will use R = 3.

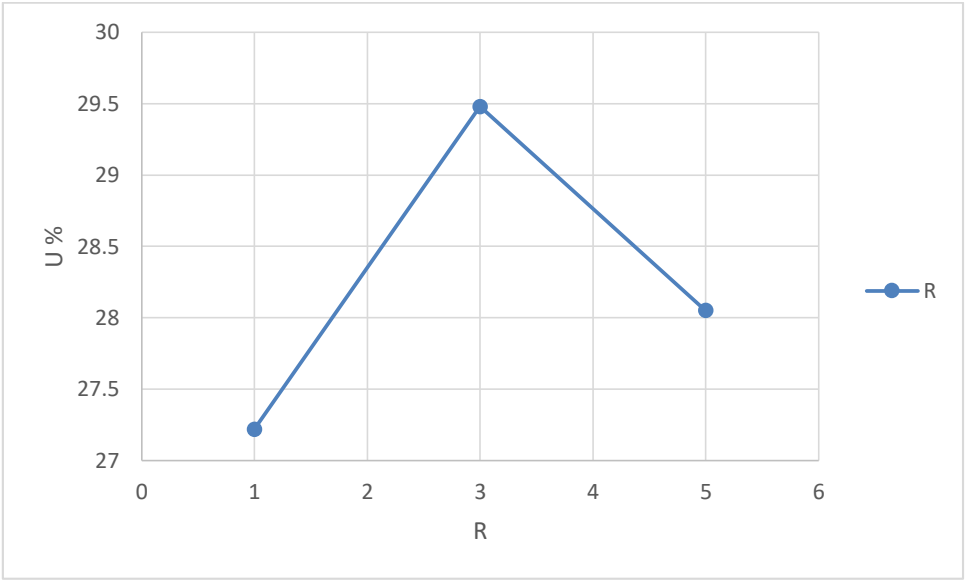

Figure 5.3. Utilization for R= 1,2 and 3

5.5.2. Scenario 1: Results of performance metrics Vs Network size:

We applied the parameters in a network includes variable number of nodes start from 10 nodes until 100 nodes that moves in speed of 30 m/s.

Packet delivery ratios (PDRs) at different numbers of nodes are shown in Table 5.4, Figure 5.4 shows PDR for DYMO and RAD protocols.

Table 5.4. Packet Delivery Ratio (%) at speed of 30m/s and different numbers of nodes

Number of nodes	Protocol	
	DYMO	RAD
10	95.43	93.47
20	93.58	93.54
30	90.72	93.14
40	90.23	92.74
50	88.06	92.11
60	81.43	88.79
70	75.95	84.18
80	74.71	82.05
90	71.03	80.71
100	70.2	78.55

PDR for RAD is better than DYMO when the number of nodes increased over 20 nodes (the size of network increased), but at small number of nodes (less than 20 nodes) DYMO showed slightly better results than RAD.

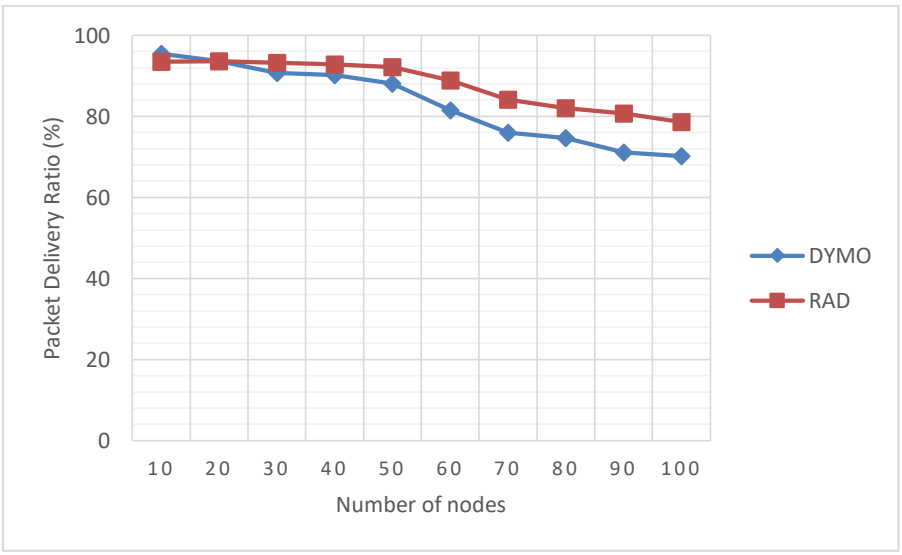

Figure 5.4. Packet Delivery Ratio (%) at speed of 30m/s by different numbers of nodes

Throughput at different numbers of nodes are shown in Table 5.5, Figure 5.5 shows throughput for DYMO and RAD protocols.

Table 5.5. Throughput (Kbps) at speed of 30m/s and different numbers of nodes

Number of nodes	Protocol	
	DYMO	RAD
10	611.45	624.47
20	608.4	609.44
30	613.97	613.53
40	624.59	593.98
50	602.37	607.27
60	583.78	596.38
70	527.13	586.83
80	499.37	533.61
90	463.61	506.17
100	378.18	492.58

Throughput for RAD is better than DYMO when the number of nodes increased over 50 nodes, when number of nodes less than 50 nodes DYMO and RAD protocols showed almost identical results.

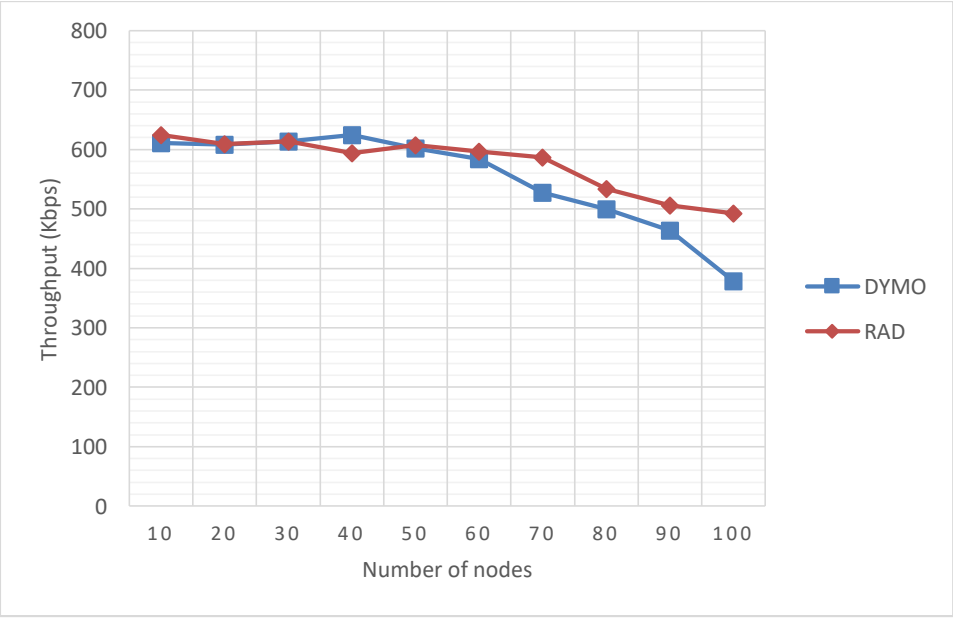

Figure 5.5. Throughput (Kbps) at speed of 30m/s by different numbers of nodes

End to end delay at different numbers of nodes are shown in Table 5.6, Figure 5.6 shows End to End delay for DYMO and RAD protocols.

Table 5.6. End to end delay (ms) at speed of 30m/s and different numbers of nodes

Number of nodes	Protocol	
	DYMO	RAD
10	123.33	136.4
20	130.43	143.25
30	142.36	151.37
40	126.93	166.49
50	131.36	163.25
60	139.5	192.81
70	128.45	192.32
80	122.67	200.23
90	141.44	207.72
100	148.58	211.43

DYMO is better than RAD with less delay, this was as a result of cryptography process in RAD, but also the delay increased in DYMO because when the number of nodes increased, the probability of malicious nodes appearance increased, then DYMO protocol needs many route maintenance processes which consumed time.

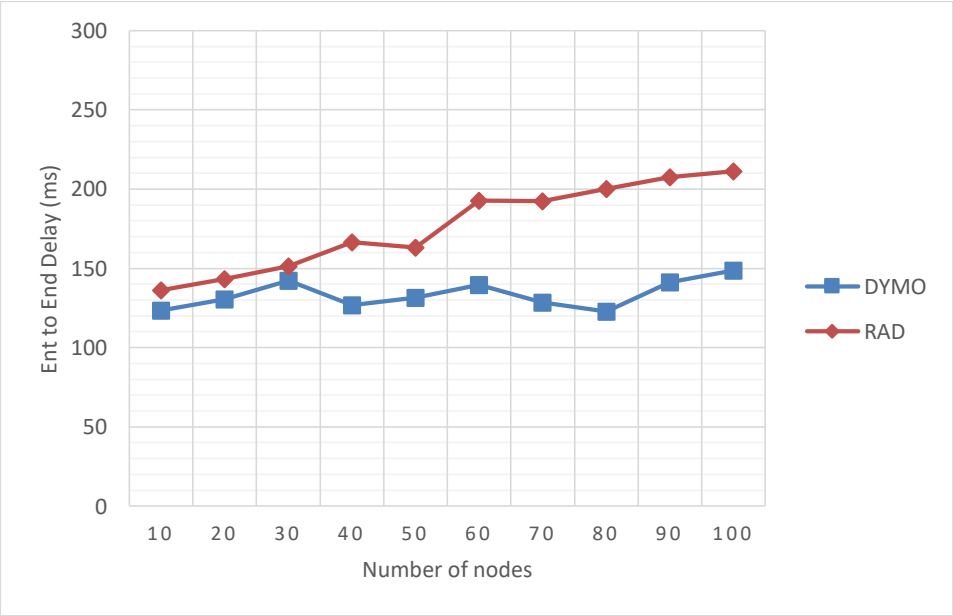

Figure 5.6. End to end delay (ms) at speed of 30m/s and different numbers of nodes

5.5.3. Scenario 2: Results of performance metrics Vs Nodes speed:

We applied the parameters in a network includes 100 nodes have different speeds start from 5 m/s until 30 m/s.

Table 5.7. Packet Delivery Ratio (%) at 100 nodes and different speeds

Speed (m/s)	Protocol	
	DYMO	RAD
5	83.46	84.61
10	73.03	78.18
15	77.75	76.27
20	65.2	70.35
25	38.29	65.43
30	69.29	76.96

PDRs at different speeds are shown in Table 5.7, Figure 5.7 shows PDR for DYMO and RAD protocols, where PDR for RAD is better than DYMO at speeds over 20 m/s, at speeds slower than 20 the differences in PDRs is slightly small.

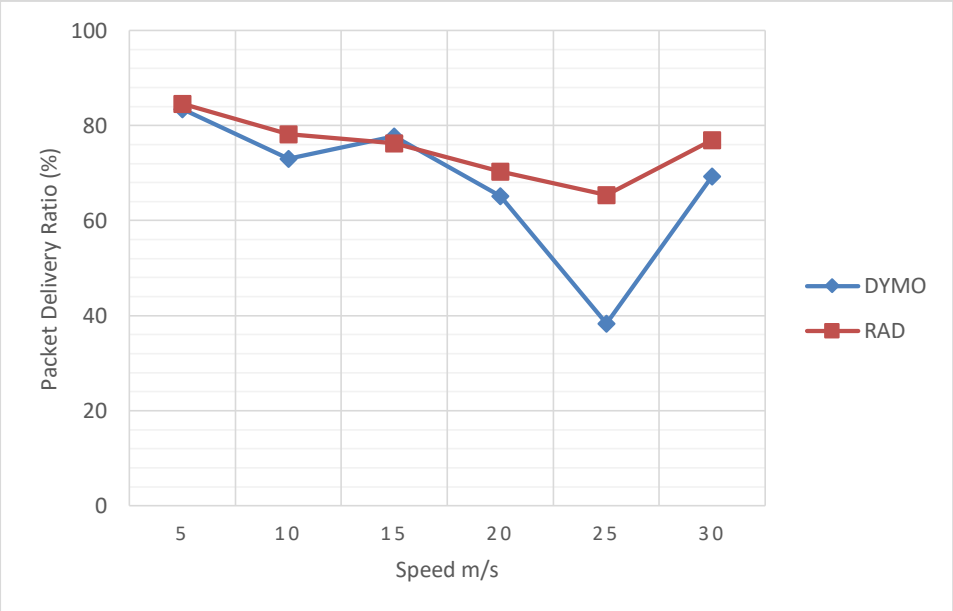

Figure 5.7. Packet Delivery Ratio (%) at 100 nodes and different speeds

Throughput at different speeds are shown in Table 5.8, Figure 5.8 shows throughput for DYMO and RAD protocols.

Table 5.8. Throughput (Kbps) at 100 nodes and different speeds

Speed (m/s)	Protocol	
	DYMO	RAD
5	351.4	381.34
10	318.67	398.12
15	394.36	394.6
20	300.28	355.25
25	199.42	319.41
30	376.2	488.48

Throughput for RAD is better than DYMO, when the speed increased over 25 m/s the throughput increased for DYMO and RAD, but RAD still have the best throughput.

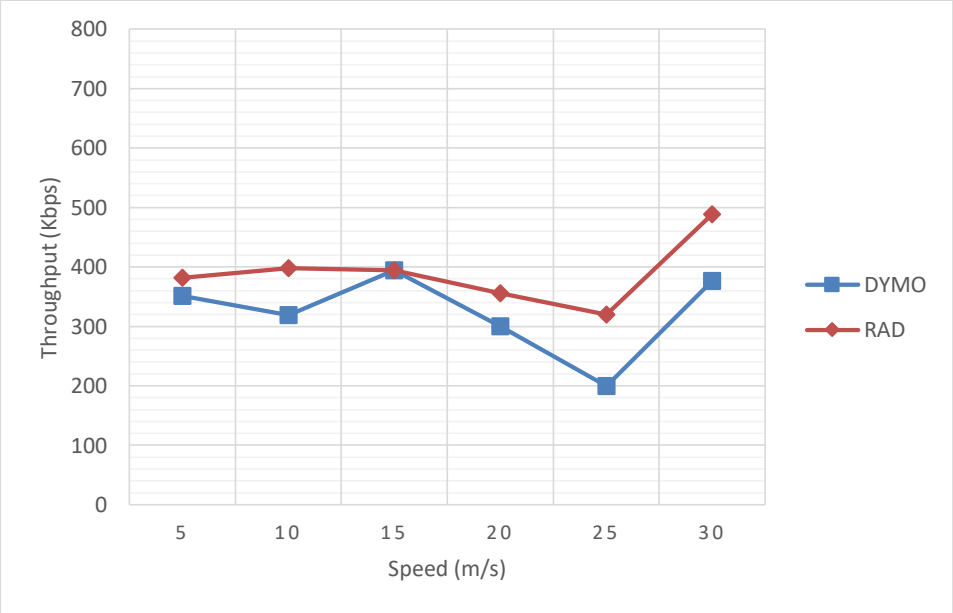

Figure 5.8. Throughput (Kbps) at 100 nodes and different speeds

End to end delay at different speeds are shown in Table 5.9, Figure 5.9 shows End to End delay for DYMO and RAD protocols.

Table 5.9. End to end delay (ms) at 100 nodes and different speeds

Speed (m/s)	Protocol	
	DYMO	RAD
5	24.86	49.73
10	56.82	72.27
15	81.54	91.4
20	122.14	108.91
25	144.2	148.36
30	147.06	208.64

DYMO is better than RAD with less delay, as we mentioned before this was as a result of cryptography process in RAD, between the speeds 15 and 25 m/s RAD gives closed results to DYMO

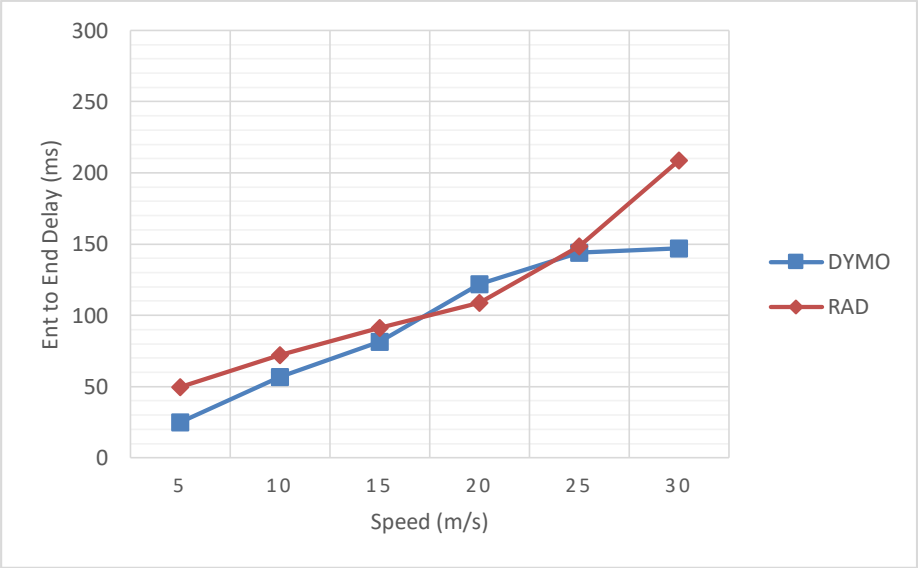

Figure 5.9. End to end delay (ms) at 100 nodes and different speeds

5.5.4. Scenario 3: Results of performance metrics Vs Simulation time:

We applied the parameters in a network includes 100 nodes moved at speed 30 m/s, we notice the behavior for the nodes during the simulation time.

Table 5.10. Packet Delivery Ratio (%) Vs Simulation time at speed of 30 m/s and 100 nodes

Simulation time (s)	Protocol	
	DYMO	RAD
100	55.52	47.44
200	60.09	55.68
300	63.91	57.52
400	63.28	63.96
500	64.09	65.68
600	69.62	75.14
700	72.33	70.76
800	70.95	73.49
900	68.39	75.78

PDR at different periods of time are shown in Table 5.10, Figure 5.10 shows PDR for DYMO and RAD protocols, at the beginning of simulation, DYMO gives better results than RAD, after 400 seconds RAD start to gives better results, this improvement in the results for RAD protocol was mainly due to reinforcement learning which improves the nodes behavior and excludes the malicious nodes with time.

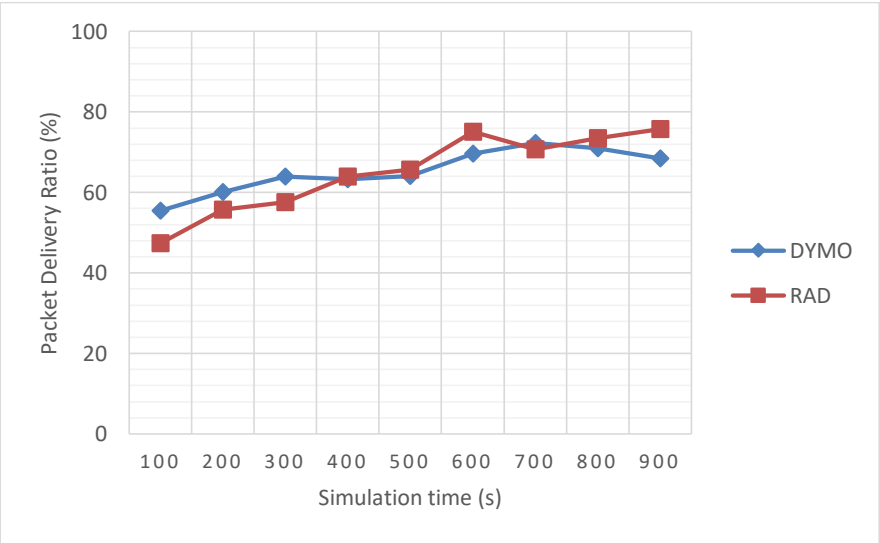

Figure 5.10. Packet Delivery Ratio (%) Vs Simulation time at speed of 30 m/s and 100 nodes

throughput at different periods of time are shown in Table 5.11, Figure 5.11 shows throughput for DYMO and RAD protocols.

Table 5.11. Throughput (Kpbs) Vs Simulation time at speed of 30 m/s and 100 nodes

Simulation time (s)	Protocol	
	DYMO	RAD
100	207.6	177.2
200	240.45	218.4
300	339.55	287.6
400	346.4	299.8
500	379.45	328.4
600	348.1	375.7
700	351.65	353.8
800	354.75	407.45
900	372.52	488.52

At the beginning of simulation, DYMO gives better results than RAD, after 600 seconds RAD start to gives better results than DYMO due to reinforcement learning.

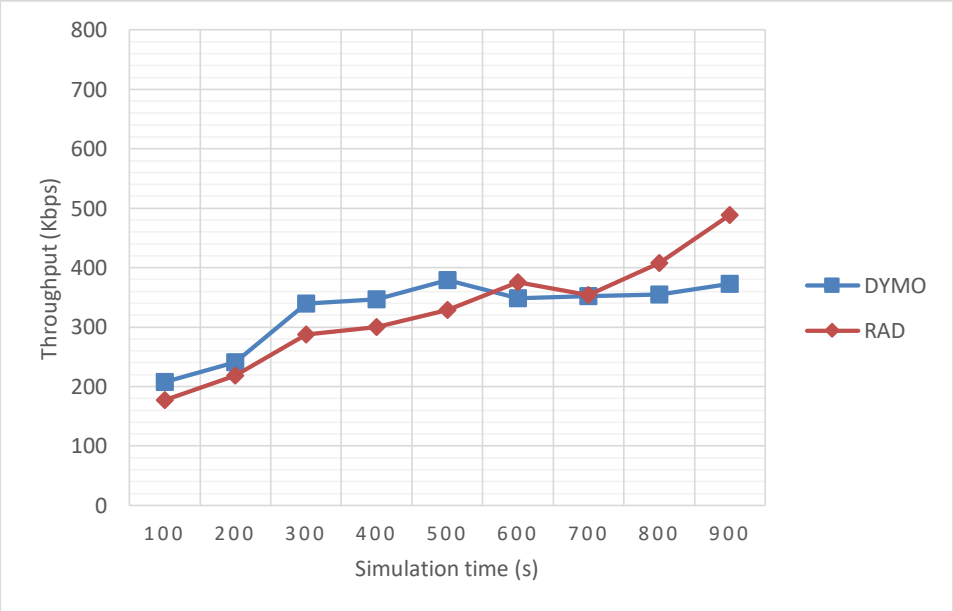

Figure 5.11. Throughput (Kpbs) Vs Simulation time at speed of 30 m/s and 100 nodes

End to end delay at different periods of time are shown in Table 5.12, Figure 5.12 shows End to end delay for DYMO and RAD protocols.

Table 5.12. End to end delay (ms) Vs Simulation time at speed of 30 m/s and 100 nodes

Simulation time (s)	Protocol	
	DYMO	RAD
100	83.8	270.95425
200	100.225	235.277
300	149.775	249.8405
400	153.2	228.386
500	169.725	229.367
600	154.05	221.254
700	155.825	215.236
800	157.375	217.838
900	145.4	209.33

DYMO always have less delay time than RAD, but the delay time for RAD decreased with time and increased for DYMO.

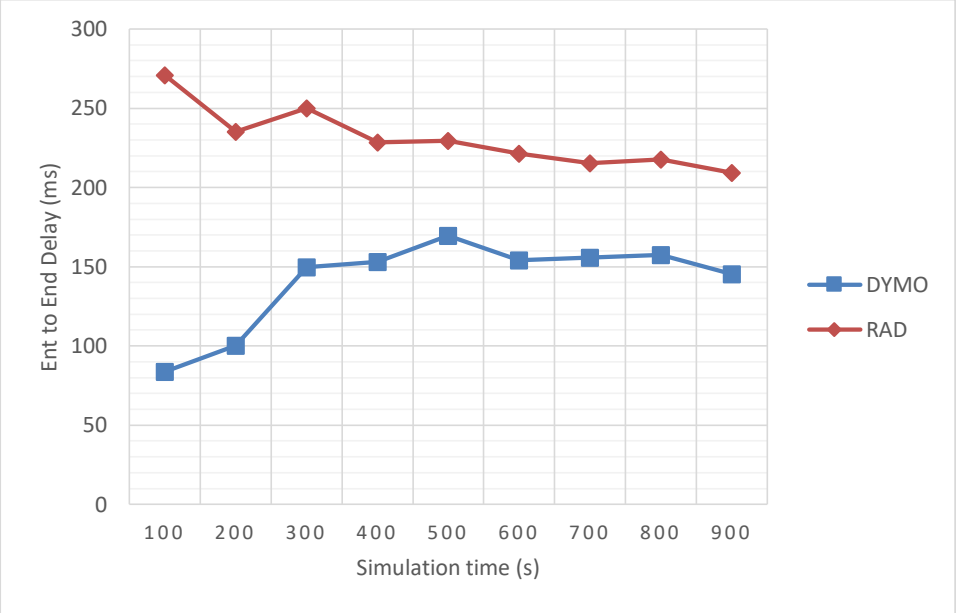

Figure 5.12. End to end delay (ms) Vs Simulation time at speed of 30 m/s and 100 nodes

5.6. Statistical analysis:

In this section, we will calculate Relative Standard Deviation (RSD) to determine the data precision for the two protocols, then calculate the utility gained from using RAD, at the end we calculate the overhead that resulted from using RAD protocol.

5.6.1. RSD: In this part, we will show the statistical analysis for the DYMO and RAD protocols, based on the calculated Average ($\bar{X}$), RSD.

$$\bar{x} = \frac{x_1 + x_2 + x_3 + x_4 + \dots}{n}$$

$$\text{Standard deviation } S = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + (x_3 - \bar{x})^2 + \dots}{n - 1}}$$

$$RSD = 100S/\bar{x}$$

The following Table 5.13 illustrates the statistical analysis results for DYMO and RAD protocols taking into consideration different points of comparison between the two protocols.

Table 5.13. $\bar{X}$ and RSD for DYMO and RAD for the previous three scenarios

Scenario		$\bar{\bar{X}}$		RSD	
		DYMO	RAD	DYMO	RAD
1	Packet Delivery ratio	67.83667	75.3	23.3211	8.820829
	Throughput	323.3883	389.5333	21.66131	14.54463
	End to End delay	96.10333	113.2183	47.33946	50.8039
2	Packet Delivery ratio	83.134	87.928	11.56368	6.77521
	Throughput	551.285	576.426	14.90693	8.247283
	End to End delay	133.505	176.527	6.3041	15.67964
3	Packet Delivery ratio	65.35333	65.05	8.377405	15.14994
	Throughput	326.7189	326.3189	18.40466	29.10755
	End to End delay	141.0417	230.8314	19.29147	8.340346

Based on the showed statistical analysis results, we can conclude that the average packet delivery ratio and throughput for RAD protocol is higher than that of DYMO protocol, in the case of end to end delay it is also higher in RAD protocol due to the encryption processes, however this delay will be decreased with time as a result of reinforcement learning process.

Regarding RSD values, we can conclude that the RSD values is lower in RAD protocol than that of DYMO protocol in most scenarios, this means that the data precision is higher in RAD protocol.

5.6.2. Utilization: To calculate the utility difference (Udif) obtained by using RAD, we can calculate utility for DYMO (Ud) and utility for RAD (Ur), then find the difference between them.

U = average throughput (Avg) / Bandwidth (B)

For scenario 1:

$Ud = (Avg_d / B) * 100\% = (323.3883 K / 2M) *100\% = 16.16\%$

$Ur = (Avg_r / B) * 100\% = (389.5333 K / 2M) *100\% = 19.47\%$

$Udif = Ur – Ud = 3.31\%$

The utility increased by 3.31 % when using RAD in scenario 1.

For scenario 2:

$Ud = (Avg_d / B) * 100\% = (551.285 K / 2M) *100\% = 27.56\%$

$Ur = (Avg_r / B) * 100\% = (576.426 K / 2M) *100\% = 28.82\%$

$Udif = Ur – Ud = 1.26\%$

The utility increased by 1.26 % when using RAD in scenario 2.

For scenario 3:

$Ud = (Avg_d / B) * 100\% = (326.7189 K / 2M) *100\% = 16.33\%$

$Ur = (Avg_r / B) * 100\% = (326.3189 K / 2M) *100\% = 16.31\%$

Udif = Ur – Ud = - 0.02%

The utility decreased by 0.02 % when using RAD in scenario 3.

5.6.3. Tradeoff between security and performance: to calculate the overhead that resulted from using RAD protocol, the extra average delay that RAD protocol added to network must be determined.

For scenario 1: Extra delay = RAD delay – DYMO delay

= 113.2183 – 96.1033

= 17.115 ms

For scenario 2: Extra delay = 43.022 ms

For scenario 3: Extra delay = 89.789 ms

The results have shown that we can get a significant increase in the encryption strength at a very small overhead for our RAD protocol compare with basic protocol DYMO, we believe this small overhead which was incurred by added time of encryption, decryption and authentication is worth the effort to increase the security level in MANET. As a result of this extra delay and operations, the power consumption will increase for the nodes.

5.7. Security analysis:

MANET have a unique topology that make it exposed to many attacks that we mentioned before in chapter one, RAD protocol solve some of these attacks like man in the middle and black hole attack.

5.7.1. Black hole attack: RAD protocol try to produce a network with authenticated nodes by using hashing, each node in this network try to act in good way that keep it away from blocked in the revocation list.

In black hole attack the malicious node receives a packet and doesn't resend it to the next node in the route, this will break the route and generate a route maintenance process which will increase the end to end delay, also it will consume

the bandwidth because many RRERs and RREQs will exchanged between nodes this will decrease the throughput for the network.

As shown in the result, RAD protocol increases the average throughput, but the increasing in the end to end delay is occurred due to the encryption process, this process solves the man in the middle attack.

5.7.2. Man in the middle attack: in this attack the malicious node tries to snoops the packets in the route, but when using the encryption process, it is impossible for the malicious nodes to decrypt the cipher packet, to prove that, here is a calculation for the time needed to decrypt the cipher packet:

We consider that the malicious node will use the brute force to decrypt the cipher packet, in this process the malicious node checks all possible keys to find the correct one.

Just consider the following:

- possible number of key combinations for AES – Diffie Hellman = 11.56×10^{76} [66].
- Supercomputer: K Computer
- Speed: 10.51 Pentaflops10.51×10^{15} Flops [Flops = Floating point operations per second]
- Flops required per combination check = 1000 (very optimistic but just assume for now)
- Combination checks per second = $(10.51 \times 10^{15}) / 1000 = 10.51 \times 10^{12}$
- Seconds in a Year = 365 x 24 x 60 x 60 = 31536000
- No. of Years to crack AES - Diffie Hellman with 128-bit Key = $(11.56 \times 10^{76}) / [10.51 \times 10^{12}) \times 31536000]$

$= (1.099 \times 10^{64})/31536000$

$= 3.484 \times 10^{56}$ years

5.8. RAD performance vs previous protocols

In this section we will compare RAD performance results with other protocol that tries to improve DYMO performance.

In “Black hole attack detection and prevention strategy in DYMO for MANET” [22], it compares the normal DYMO with 2 other protocols, black hole detection and black hole prevention.

Figure 5.13 normalizes RAD throughput in scenario one and compare it with the normalized result in that research.

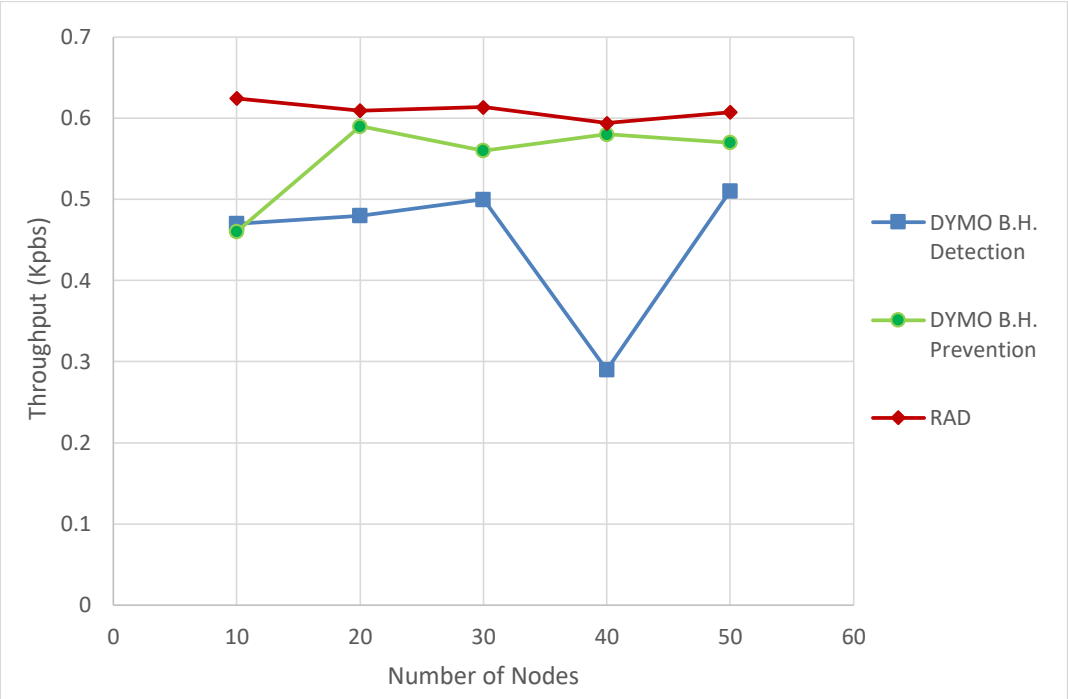

Figure 5.13. RAD compared with DYMO black hole detection and prevention protocols

The figure shows that RAD protocol improves the average throughput and gives better results than the DYMO black hole detection and prevention protocol.

5.9. Summary:

In this chapter, we designed an approach to improve the performance of the DYMO protocol, we applied the performance metrics to check the reliability for our protocol, and to make sure that the new protocol gives a better performance in MANET than DYMO.

Based on the trace files, we concluded that the new protocol shows an improvement performance of MANET, in the three scenarios, the throughput and packet delivery ratio for the new protocol is better than DYMO protocol. This happen as a result of improving

the security for the packets transmission between nodes. On the other hand, the delay is increased due to the authentication and encryption processes.

Authentication, encryption and decryption processes need a determined period of time, which increase the end to end delay. However, we increased the security of the data by relatively little increasing the delay in delivery.

Chapter Six: Conclusion and future work

6.1. Thesis conclusion

6.2. Future work

Chapter Six: Conclusion and future work:

6.1. Thesis Conclusion:

The goal of this work is to develop a new model based on DYMO protocol where a modification was proposed to route discovery and route maintenance processes. In route discovery process we made an authentication process between the nodes by using MD5 hashing algorithm, then we used reinforcement learning to improve the route maintenance process based on machine learning approach. At the end we used Diffie-Hellman key management to exchange the secret key to encrypt and decrypt the data between Source S and Destination D.

When we tested the proposed protocol, the results show improvement in the performance of MANETs, despite the little increased in the end to end delay in comparison with DYMO protocol. This is due to the overheads in authentication and encryption processes.

6.2. Future work:

In this thesis, we applied the new approaches to improve the performance and security for DYMO protocol, we can apply the same approach with other protocols in MANET.

In this thesis, we focused on security consideration for the data, the delay that happened in transmitting the data has not been addressed. Therefore, our proposed protocol can be developed to decrease that delay by using RAD protocol in the large networks. This is because the delay that happened due to cryptography process can be less than the delay in DYMO that happened due the repeating the route maintenance process to overcome the malicious nodes in the route. Thus we can use RAD protocol in large network that have more than 100 nodes to get better performance.

References

1. Anna malai, S., *INTRODUCTION TO NETWORKING.* 2012.
2. Kavitha, R., *Smart Home Systems Using Wireless Sensor Network –A Comparative Analysis.* 2012.
3. Imran, E., et al., *Implementing Wireless Infrastructure Network with Efficient Security*. 2007.
4. Bang, A., *Wireless Ad-Hoc Networks: Types, Applications, Security Goals*. 2015.
5. Gupta, A., P. Verma, and R. Sambyal, *An Overview of MANET: Features, Challenges and Applications*. 2018.
6. Naghshegar, A., A. Darehshoorzadeh, and A. Dana, *Dynamic Topology Control Scheme in MANETs for AODV Routing*. 2008.
7. Kumar, A., L. Jacob, and A.L. Ananda, *SCTP vs TCP : performance comparison in MANETs*. 2004. 431-432.
8. Bang, A. and P. Ramteke, *MANET: History,Challenges And Applications.* 2013.
9. Kumar, S. and S. Kumar, *Study of MANET: Characteristics, Challenges, Application, Routing Protocol and Security Attacks.* International Journal of R & D in Engineering Science and Management (ISSN 2393-865X), 2015. **2**: p. 266-274.
10. Amara korba, A., M. Nafaa, and S. Ghanemi, *Analysis of security attacks in AODV*. Vol. 0. 2014. 752-756.
11. Dhende, S., et al., *A survey on black hole attack in mobile ad hoc networks*. 2018. 1-7.
12. Sowah, R., G. Mills, and K. Koumadi, *Detection and Prevention of Man-in-the-Middle Spoofing Attacks in MANETs Using Predictive Techniques in Artificial Neural Networks (ANN).* Journal of Computer Networks and Communications, 2019. **2019**: p. 1-14.
13. Bogdanoski, M., T. Shuminoski, and A. Risteski, *Analysis of the SYN flood DoS attack.* International Journal of Computer Network and Information Security, 2013. **5**: p. 1-11.
14. Guarnera, M., et al., *MANET: possible applications with PDA in wireless imaging environment*. 2002. 2394-2398 vol.5.
15. Yang, H.S. and S.J. Yoo, *Authentication Techniques for Improving the Reliability of the Nodes in the MANET*. 2014 International Conference on IT Convergence and Security, ICITCS 2014, 2015.
16. Sharma, N. and A. Gangal, *MOBILE NODE AUTHENTICATION IN MANET USING ENHANCED CLUSTER BASED AUCRES ALGORITHM.* Far East Journal of Electronics and Communications, 2016: p. 1-12.
17. Verma, U., S. Kumar, and D. Sinha, *A secure and efficient certificate based authentication protocol for MANET*. 2016. 1-7.
18. Sen, J., *A Robust and Efficient Node Authentication Protocol for Mobile Ad HocNetworks*. 2010.
19. Subu, N., S. Jayapal, and D. Sridharan, *A trust system in manet with secure key authentication mechanism*. 2012. 261-265.
20. Lu, B. and U. Pooch, *A Lightweight Authentication Protocol for Mobile Ad Hoc Networks*. Vol. 11. 2005. 546-551.
21. Tembhurkar, M. and Y. Singare, *Design of an Efficient Initial Access Authentication over MANET*. 2015.
22. D. Nitnaware and A. Thakur, *"Black hole attack detection and prevention strategy in DYMO for MANET,"* 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN), Noida, 2016.

23. Gupta, A., H. Sadawarti, and A. Verma, *Review of Various Routing Protocols for MANETs.* International Journal of Information and Electronics Engineering, 2011. **1**: p. 251-259.
24. Ferdaus, J. and R. Salihi, *Routing: Internet Routing Protocols and Algorithms*. 2015.
25. Chiu, J.-C., C.-L. Chen, and H.-W. Chiu, *The Adaptive Routing Algorithm for Linked-List Wireless Network with Wormhole Mechanism*. 2016. 120-125.
26. Fihri, M., C. Badr, and A. Ezzati, *Comparative study of routing protocols in MANET*. 2014. 149-153.
27. Venkatesan, T.P., P. Rajakumar, and A. Pitchaikkannu, *Overview of Proactive Routing Protocols in MANET*. 2014. 173-177.
28. Johnson, D. and D. Maltz, *Truly seamless wireless and mobile host networking. Protocols for adaptive wireless and mobile networking.* Personal Communications, IEEE, 1996. **3**: p. 34-42.
29. Khatri, P., et al., *Performance Study of Ad-Hoc Reactive Routing Protocols.* Journal of Computer Science, 2010. **6**.
30. Al-Dhief, F., et al., *MANET Routing Protocols Evaluation: AODV, DSR and DSDV Perspective.* MATEC Web of Conferences, 2018. **150**: p. 06024.
31. Naski, C. and S. Naski, *Performance of Ad Hoc Routing Protocols: Characteristics and.* 2004.
32. Mahdipour, E., A. Rahmani, and E. Aminian, *Performance Evaluation of Destination-Sequenced Distance-Vector (DSDV) Routing Protocol.* Proceedings - 2009 International Conference on Future Networks, ICFN 2009, 2009.
33. Manjunath, M. and M. D H, *Spatial DSDV (S-DSDV) routing algorithm for mobile ad hoc network.* Proceedings of 2014 International Conference on Contemporary Computing and Informatics, IC3I 2014, 2015: p. 625-629.
34. Clausen, T.H., et al., *Optimized link state routing protocol (OLSR).* 2003.
35. Alaparthi, S., et al., *Dynamic Source Routing Protocol—A Comparative Analysis with AODV and DYMO in ZigBeebased Wireless Personal Area Network*. 2019. 1042-1046.
36. Kumar, V. and T. Romila, *IMPLEMENTATION OF DYNAMIC SOURCE ROUTING (DSR) IN MOBILE AD HOC NETWORK (MANET).* International Journal of Research in Engineering and Technology, 2013. **02**: p. 339-345.
37. Rajneesh, M., R. Gujral, and A. Kapil, *An Efficient Searching and an Optimized Cache Coherence handling Scheme on DSR Routing Protocol for MANETS.* International Journal of Computer Science Issues, 2011. **8**.
38. Liu, S., Y. Yang, and W. Wang, *Research of AODV Routing Protocol for Ad Hoc Networks1*. AASRI Procedia, 2013. **5**: p. 21–31.
39. Abdelshafy, M. and P. King, *AODV Routing Protocol Performance Analysis under MANET Attacks.* International Journal for Information Security Research (IJISR), 2014. **Volume 3**: p. 418-426.
40. Rastogi, P., *AODV Routing Protocol for MANET - A Review.* International Journal of Recent Advancement in Engineering & Research, 2017. **2**: p. 5.
41. Hamamreh, R. and O. Salah, *An Intelligent Routing Protocol Based on DYMO for MANET.* International Journal of Digital Information and Wireless Communications, 2018. **8**.
42. Sommer, C. and F. Dressler. *The DYMO Routing Protocol in VANET Scenarios*. in *2007 IEEE 66th Vehicular Technology Conference*. 2007.
43. Gupta, A., H. Sadawarti, and A. Verma, *Implementation of DYMO Routing Protocol.* International Journal of Information Technology, Modeling and Computing, 2013. **1**.
44. Quan-xing, M. and X. Lei, *DYMO routing protocol research and simulation based on NS2.* Proc. of ICCASM, 2010. **14**.

45. Aujla, G., *Comprehensive Evaluation of AODV, DSR, GRP, OLSR and TORA Routing Protocols with varying number of nodes and traffic applications over MANETs.* IOSR Journal of Computer Engineering, 2013. **9**: p. 54-61.
46. Aggarwal, A., S. Gandhi, and N. Chaubey, *Performance Analysis of AODV, DSDV and DSR in MANETs.* International Journal of Distributed and Parallel systems, 2014. **2**.
47. Elboukhari, M., M. Azizi, and A. Azizi, *Performance Comparison of Routing Protocols in Mobile Ad Hoc Networks.* International Journal of UbiComp, 2015. **6**: p. 1-11.
48. Sudhir Agrawal, S.J., Sanjeev Sharma and Roopam Gupta. , *Mobility based Performance Analysis of AODV and DYMO under Varying Degree of Node Misbehavior.* International Journal of Computer Applications 29(7), September 2011. : p. 36-41.
49. Kaur, R. and A. Kaur, *Digital Signature*. 2012. 295-301.
50. Reddy, A.R., *A Review of digital Certificates.* IETE Golden Jubilee Compendium: Evolution and perspective – Electronics, Telecommunications, IT and Broadcasting, 2003: p. pp.97-105.
51. Shakya, A. and N. Karna, *Enhancing MD5 hash algorithm using symmetric key encryption*. 2019. 18-22.
52. Kioon, M., Z. Wang, and S. Das, *Security Analysis of MD5 Algorithm in Password Storage.* Applied Mechanics and Materials, 2013. **347-350**.
53. Biham, E., R. Chen, and A. Joux, *Cryptanalysis of SHA-0 and Reduced SHA-1*. Journal of Cryptology, 2014. **28**: p. 110-160.
54. Toma, D., et al., *Design of a proven correct SHA circuit*. 2004. 31-34.
55. Wang, X., Y. Yin, and H. Yu, *Finding Collisions in the Full SHA-1*. Vol. 3621. 2005. 17-36.
56. Michail, H., et al., *Authentication with RIPEMD-160 and Other Alternatives: A Hardware Design Perspective*. 2010.
57. Sobti, R. and G. Ganesan, *Cryptographic Hash Functions: A Review.* International Journal of Computer Science Issues, ISSN (Online): 1694-0814, 2012. **Vol 9**: p. 461-479.
58. Kumar, S. and E.P. Gupta, *A Comparative Analysis of SHA and MD5 Algorithm.* International Journal of Computer Science and Information Technologies, 2014. **5**: p. 4492-4495.
59. Munir, M., *Cryptography*. 2005.
60. Ferguson, N., B. Schneier, and T. Kohno, *Diffie-Hellman*. 2015. p. 181-193.
61. Sehgal, P., et al., *Modification of Diffie-Hellman Algorithm to Provide More Secure Key Exchange.* International Journal of Engineering and Technology, 2013. **5**: p. 2498-2501.
62. Szepesvári, C., *Algorithms for Reinforcement Learning*. Vol. 4. 2010.
63. *https://www.includehelp.com/ml-ai/main-points-of-reinforcement-learning-in-artificial-intelligence.aspx.*

64. Gupta, Anuj & Kaur, Jatinder & Kaur, Sandeep. *(2011). COMPARISON OF DYMO, AODV, DSR AND DSDV MANET ROUTING PROTOCOLS OVER VARYING TRAFFIC.* International Journal of Research in Engineering & Applied Science. 1. 71-83.

65. NoorAldeen Jabali, "*Extending AES with DH Key-Exchange to Enhance VoIP Encryption in Mobile Networks*", 2018

66. Daniel J. Bernstein,(2005). " *Understanding brute force* ".The author was supported by the National Science Foundation under grant CCR– 9983950.

67. Hamamreh, Rushdi & Bawatna, Mohamed. (2014). *Protocol for Dynamic Avoiding End-to-End Congestion in MANETs.* 10.5923/j.jwnc.20140403.01.

Appendices

Appendix A - Published Paper

RAD: Reinforcement Authentication DYMO Protocol for MANET

Accepted and presented in International Conference on Promising Electronic Technologies 2019 (ICPET), and published on IEEE Xplore digital library.

https://ieeexplore.ieee.org/document/8925310

RAD: Reinforcement Authentication DYMO Protocol for MANET

Rushdi A. Hamamreh
Department of computer engineering
Al-Quds University
Jerusalem, Palestine
rushdi@staff.alquds.edu

Mohammad Ayyad
Department of computer engineering
Al-Quds University
Jerusalem, Palestine
rushdi@staff.alquds.edu

Mohammad Jamoos
Department of computer science
Al-Quds University
Jerusalem, Palestine
jamoos@staff.alquds.edu

***Abstract*: Mobile ad hoc network (MANET) does not have fixed infrastructure or centralized server which manages the connections between the nodes. Rather, the nodes in MANET move randomly. Thus, it is risky to exchange data between nodes because there is a high possibility of having malicious node in the path. In this paper, we will describe a new authentication technique using message digest 5 (MD5), hashing for dynamic MANET on demand protocol (DYMO) based on reinforcement learning. In addition, we will describe an encryption technique that can be used without the need for a third party to distribute a secret key. After implementing the suggested model, results showed a remarkable enhancement in securing the path by increasing the packet delivery ratio and average throughput. On the other hand, there was an increase in end to end delay due to time spent in cryptographic operations.**

Index terms: MANET, DYMO, authentication, encryption, reinforcement.

I. INTRODUCTION

A Mobile Ad Hoc Network (MANET) is a collection of mobile nodes which interacts together via wireless connections and change their location dynamically. When two nodes (Source (S) and Destination (D)) want to communicate with each other, MANET tries to create a path between the specified nodes. As shown in figure 1, this path includes many other nodes (intermediate Nodes) which transmit the message between S and D.

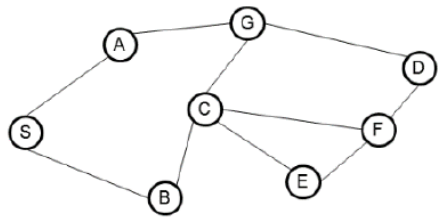

Figure1: MANET nodes

There are two main protocols used to specify the path between S and D, which are described as Table Driven and On Demand.

A. Table Driven or Proactive Protocols:

In this type of protocols, routes are created only when required [1], following is an example for this protocol.

Destination-Sequenced Distance-Vector (DSDV):
In this protocol, routing discovery process does not need much time, so it is classified as fast way to transmit packets between S and D. Routing table in DSDV updates frequently, this will consume the limited resources like battery power and the bandwidth, even when the network is inactive. If any changes happen in the network infrastructure, the routing table should change accordingly before the network rearrange. For that reason, DSDV is not considered appropriate for networks with high mobility environment or large-scale networks [2].

B. On Demand or Reactive Protocols:

In this protocol, each node creates one or more tables that include routing information to all the remaining nodes in the network. Following are examples of this protocol:

1. Dynamic source routing (DSR):
All types of reactive protocols have the same main features: route discovery and route maintenance. Routing discovery process: S broadcasts routing request (RREQ) to neighbouring nodes. When RREQ arrives the neighbouring node, the node retransmit it until it reaches D. D will send routing replay (RREP) message to answer the request. Any route discovery process may produce many routes to D. the longer the packet moves through the route. packet header size grows due to source routing as shown figure 2.
Route maintenance mechanism starts immediately by the middle node when an interruption occurs in the next hop that is linked to the destination node, in this case S node sends a route error (RERR) messages, after receiving RERR, S node start searching for alternative route or starts a new route discovery process [3].

978-1-7281-2337-0/19/$31.00 ©2019 IEEE
DOI 10.1109/ICPET.2019.00032

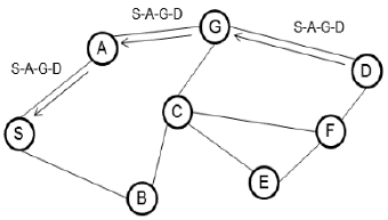

Figure 2: DSR protocol

2. **Ad hoc On-Demand Distance Vector (AODV):** AODV tries to enhance DSR by including routing tables in the nodes, as shown in figure 3, in this way data packets do not have to contain routes. AODV uses small messages defined as HELLO messages to determine local connectivity, which will shorten the time required to respond to routing requests, and activate updates if required. Sequence numbers are assigned to routes and routing table entries to take place of old cached routing entries [4].

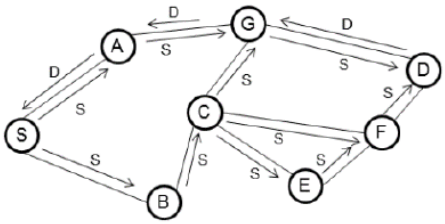

Figure 3: AODV protocol

3. **Dynamic MANET on demand (DYMO):** DYMO is a simpler version of AODV, but with path accumulation feature as shown in figure 4. Path accumulation reduces the number of RREQs and makes the route maintenance easier [5].

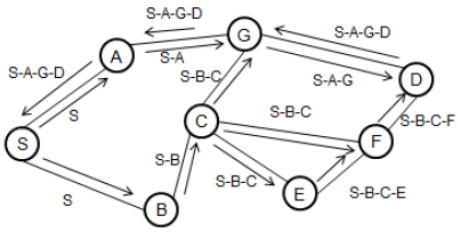

Figure 4: path accumulation in DYMO

II. PROBLEM STATEMENT

In this paper, we try to describe a way to guarantee that the message will be securely sent from S to D without being attacked by any nodes in the path. Moreover, we will explain how to recover the path between S and D in fast and efficient way if any failure occurred while transferring the data.
These problems are found in an environment that has properties such as: nodes have finite power, limited bandwidth and no centralization media [6].

III. OBJECTIVES

The main objective is to make a fast authentication between intermediate nodes and between S and D by minimizing the delay that occurs when nodes start contacting each other. Another one is securing the message between S and D by using a specific method for encryption. The last objective is to find a new way for the authentication between the nodes by using hashing algorithm.

IV. MANET SECURITY MECHANISIM

Many security mechanisms are available for MANET; these mechanisms are summarized as follows:

Certificate (C): in this mechanism, the distribution for certificates is done by nodes themselves, there is no need for trusted party to manage certificates. Nodes distribute the certificates in an independent manner. Certifications are released with determined time interval. The nodes can update the certificates before they are expired [7].

Tickets (T): Probe packets (PKT) are released with determined number of "tickets". Many routes can be checked to determine whether they are appropriate for transmitting data or not, this checking done by tickets [8].

Digital Signatures (DS): Digital signature scheme includes: a key generation algorithm, signature algorithm and verification algorithm.
As an example for signature:
Source floods signed RREQ, then transmit it with certificates RREQ which include IP address for D.
S sends RREQ = [RREQ|| IP_D || $C[_S]$] DS_S
first node adds its own signature and certificates
A resends RREQ = [[RREQ|| IP_D || $C[_S]]DS_S]DS_A$ || $C[_A]$
Each node checks the previous node signature, then substitute it with its own, then adds a reverse route to S.
B resends RREQ = [[RREQ|| IP_D || $C[_S]]DS_S]DS_B$ || $C[_B]$
D also checks S signature[9].

Cookies session (Cookie): a cookie is a data packet, transmits from the S to D, and then transmits back to the S when D try to read this packet.
Cookies are transmitted from S to D using "Set-Cookie" headers:
Set-Cookie: NAME=VALUE; expires=DATE; path=PATH; domain=DOMAIN_NAME; secure [10].

Security Association (SA): it is aimed to create a secure communication by constructs shared security features between any two nodes in the network.
Many protocols help SA to implement several functions, here is some examples for those protocols: Authentication Headers (AH), Encapsulating Security Payloads (ESP) and Key Management Protocol.
SA features [Cryptographic algorithm || mode (Tunnel or Transport) || traffic encryption key|| parameters for the network data to be passed over the connection (Addresses) [11].

The authentication techniques that mentioned above work in different layers in OSI model. Table 1 show that layers.

Table I: Authentication techniques and OSI layers

Authentication techniques	Layer
Certificates	application
Tickets	application
Digital signatures	application
Cookies session	session
Security association	network
RAD	network

V. HASHING TECHNIQUES

Hashing are assigned values, resulted by using a specific mathematical functions. Also, it is a one-way operation to guarantee the security for data while transmitting it on the route of network. The message is designed for a determine receiver only, and the packets will be secured against tampering. We will introduce three types of Hashing:

Secure Hash Algorithm 1 (SHA-1): it is process of generation a 160-bit message digest (MD). Hence, the MD is included into a Digital Signature Algorithm (DSA) which generates/verifies the signature for the message. Due to the tiny size of MD, signing MD process will give the efficiency for the message, then the hash function will check and verify that message [12].

Secure Hash Algorithm 512 (SHA-512): it creates a digest of 512 bits from a multiple-block message. SHA-512 is a new version of SHA-1. To make SHA-512 reliable, it must have the ability to create the longest hash value that any hash function can creates, which equal to 512-bit. By using long hash value, SHA-512 will be more powerful against any attack than the function which use a small hash value. Also, SHA-512 is describe as a powerful, and high speed hash function [13].

Message digest 5 (MD5): it is a hashing algorithm specialized in processing the data in 512-bit blocks, it works in creates 16 words, each one contains 32bit. This process will produce a 128-bit message digest value. Generating MD5 is faster than SHA-1 [14].

In our model, we will use MD5 hashing because it is fast and consumes less memory. This is shown in figure 5 which describes the total bit length in the three techniques for hashing.

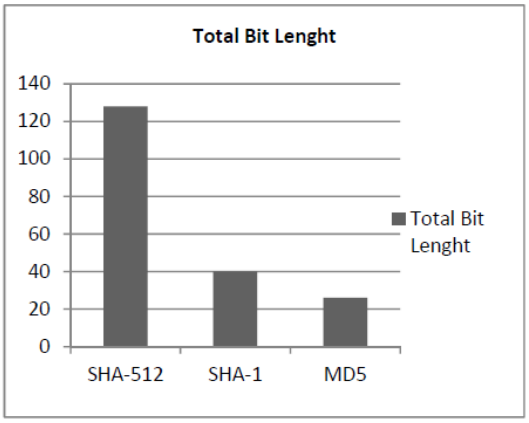

Figure 5: comparison between SHA 512, SHA1 and MD5 based on total bit length.

VI. ENCRYPTION TECHNIQUE – DIFFIE HELLMAN

Diffie-Hellman is a process to create a shared secret key between two nodes, by using a specific technique to share that key through the middle nodes without give those nodes the ability to discover it. It is good way to share a secret key between two nodes without need for third party, because the two nodes share on the key generation process [15].

VII. REINFORCEMENT LEARNING

Reinforcement protocol is a software aims to maximize the utility U(S) by taking the right action in network. When any node in network works in the right way, it will achieve some rewards. If that node takes a wrong action during transmitting the data, it will be sent to revocation list and stay there until the network test this node again in other discovered routes [16].
Here is an algorithm for reinforcement learning:

- ST– set of states of paths
- A– set of actions a (like selection of the path)
- T(st,a,st') = P(st'|st,a)– the probability of transmission from *st* to *st'* given action *a* (success in transfer data to neighbor nodes in the path)
- R(st,a)– the reward for taking action *a* in state *st* [17].

$$R(st,a) = \sum_{st'} P(st' \mid st,a) r(st,a,st')$$

$$R(st,a) = \sum_{st'} T(st,a,st') r(st,a,st')$$

$$U(st) = \max_{a} (R(st,a) + \gamma \sum_{st'} T(st,a,st') U(st'))$$

Here is the function for reinforcement of nodes in a path[18]:

function NODES-REWARD (*node, reward* ← 1)
if node succeed to transfer data? **then**
{
reward ← *reward* +
end
}
else
{
do {
node added to revocation list
}
While (*reward<3*)
end
}

VIII. PROPOSED ALGORITHM

We propose the algorithm in the following steps:
Step 1: The source node S starts the route discovery phase by broadcasting the RREQ packet to all its neighbouring nodes.
Step 2: For authentication between nodes, MD5 algorithm will produce hash value between each intermediate nodes and between S and D.
Step 3: if there are any changes in hash value between two neighboured nodes, these two nodes will be considered as malicious and stored in the black list.
Step 4: the path discovery will continue from the node where the cut was happened, and there will be no need to restart the discovery from S.
Step 5: if any nodes from the black list used in another path for 3 times, we can treat it as normal node and delete it from the black list.
Step 6: The packets are transmitted from source to destination through the discovered path.
Step 7: Source encrypts the message using Deffie-Hellman:

a. S and D agree on p and α
b. S chooses X_S and sends $Y_S = \alpha^{X_S} \bmod q$
c. D chooses X_D and sends $Y_D = \alpha^{X_d} \bmod q$
d. S computes $K_S = Yd^{X_S} \bmod q$
e. D computes $K_D = Yd^{X_d} \bmod q$
f. Then K is the shared secret.

Step 8: Destination decrypts the message using K.
The following flow chart describes our model:

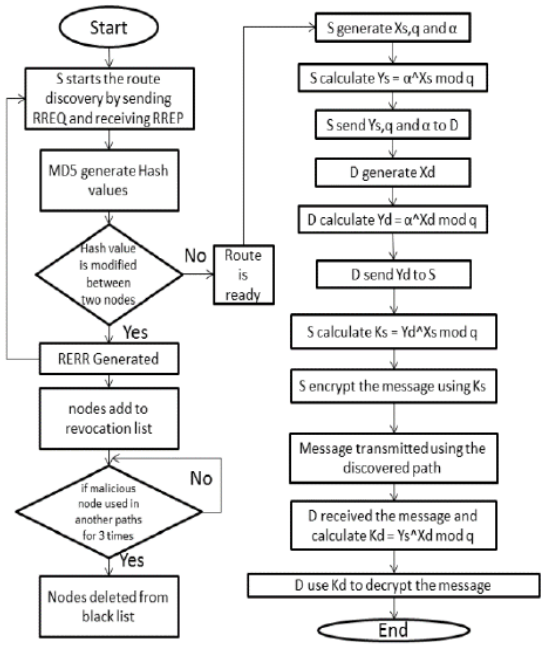

Figure 6: flow chart for proposed algorithm

IX. PERFORMANCE EVALUATION

In this part, we compare the protocol DYMO with the proposed protocol RAD. NS2 will be used for simulation to perform that comparison based on four points. These points are: Throughput, End to End Delay, Packet Delivery ratio and Packet Loss ratio. Table 2 shows the parameters which will be used in NS2 simulation.
These indicators analyse the performance as follows:

1. Throughput: is the amount of bits delivered per unit of time.
2. End to End Delay: is the time needed to deliver the packets from S to D.
3. Packet Delivery ratio: is the ratio of the number of packets successfully received to the number of total packets sent.
4. Packet Loss ratio: is the rate at which information is not sent through the network

Table II: NS2 simulation parameters

Simulation Tool	NS-2.35
Operating System	Ubuntu 12.04
No. of Nodes	10,20,30,40,50
Antenna model	Omni directional
Interface queue size	50 packets
Transmission range	250m
Examined protocol	DYMO, RAD
Simulation area	1100M*1100M

Figure 6 shows the throughput for DYMO and RAD. RAD has better throughput than DYMO because RAD has a guaranteed delivery for packets in unit of time. In DYMO, the rate for dropped packets is higher than RAD which makes negative effect for throughput.

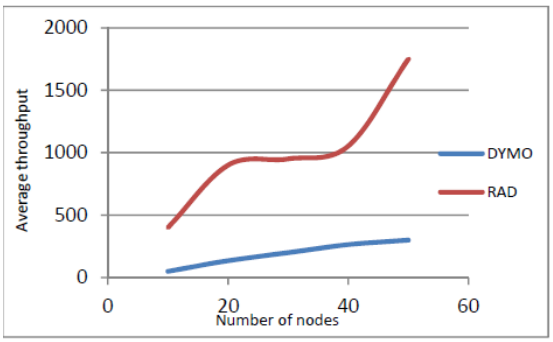

Figure 6: Average throughput

Figure 7 shows that packet delivery ratio for RAD is always better than DYMO. This happens because RAD avoids the paths which include malicious nodes.

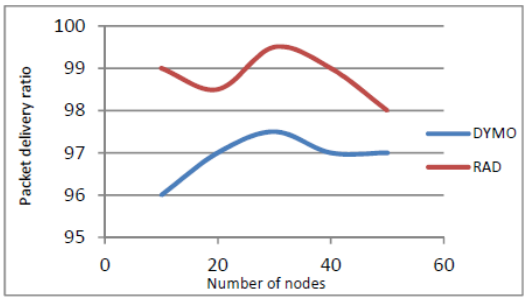

Figure 7: Packet delivery ratio

Figure 8 shows that packet loss ratio for RAD is less than DYMO. This proves that RAD ensures that packets will be transmitted in a secured path.

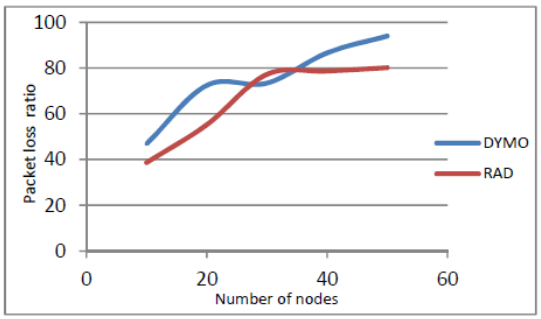

Figure 8: Packet loss ratio

Figure 9 shows that DYMO have less delay than RAD. This happens because RAD needs time to encrypt the packets and deal with hash values between nodes.

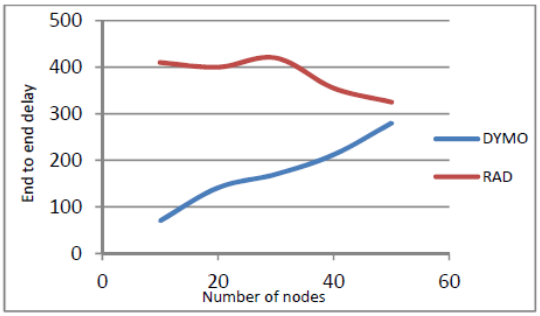

Figure 9: End to end delay

X. CONCLUSIONS

In this paper, we described a new protocol for MANET environment which is called RAD. RAD protocol is based on MD5, Diffie-Hellman and Reinforcement techniques. MD5 for hashing makes the neighboring nodes authenticated. Diffie-Hellman model distributes K_S through public key infrastructure. Reinforcement technique improves the utility of the networks through avoiding path which includes malicious nodes.

REFERENCES

1. Alslaim, M.N., H.A. Alaqel, and S.S. Zaghloul. *A comparative study of MANET routing protocols*. in *The Third International Conference on e-Technologies and Networks for Development (ICeND2014)*. 2014.
2. Khan, K.U.R., et al. *An Efficient DSDV Routing Protocol for Wireless Mobile Ad Hoc Networks and its Performance Comparison*. in *2008 Second UKSIM European Symposium on Computer Modeling and Simulation*. 2008.
3. Lili, P. *Research on performance of on- demand routing protocols in "linear structure" of Ad hoc network*. in *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. 2016.
4. Gupta, A., et al. *Comparison of various routing algorithms for VANETS*. in *2016 International Conference System Modeling & Advancement in Research Trends (SMART)*. 2016.
5. Hamamreh, R. and O. Salah, *An Intelligent Routing Protocol Based on DYMO for MANET*. International Journal of Digital Information and Wireless Communications, 2018. **8**.
6. Dhar, S., *MANET: Applications, Issues, and Challenges for the Future*. IJBDCN, 2005. **1**: p. 66-92.

7. Forne, J., et al., *Certificate status validation in mobile ad hoc networks.* IEEE Wireless Communications, 2009. **16**(1): p. 55-62.
8. Salah, S., et al., *A Model for Incident Tickets Correlation in Network Management.* Journal of Network and Systems Management, 2015. **24**.
9. Prasad, A. and K. Kaushik, *Digital Signatures*. 2019. p. 249-272.
10. Stallings, W., *Cryptography and Network Security Principles and Practices*. 2010.
11. Benin, A., S. Toledo, and E. Tromer. *Secure Association for the Internet of Things*. in *2015 International Workshop on Secure Internet of Things (SIoT)*. 2015.
12. Xiao-hui, C. and D. Jian-zhi. *Design of SHA-1 Algorithm Based on FPGA*. in *2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing*. 2010.
13. Gueron, S., S. Johnson, and J. Walker. *SHA-512/256*. in *2011 Eighth International Conference on Information Technology: New Generations*. 2011.
14. Yong-Xia, Z. and Z. Ge. *MD5 Research*. in *2010 Second International Conference on Multimedia and Information Technology*. 2010.
15. Nan, L. *Research on Diffie-Hellman key exchange protocol*. in *2010 2nd International Conference on Computer Engineering and Technology*. 2010.
16. Liang, X., et al., *A Deep Reinforcement Learning Network for Traffic Light Cycle Control.* IEEE Transactions on Vehicular Technology, 2019. **68**(2): p. 1243-1253.
17. Kiumarsi, B., et al., *Optimal and Autonomous Control Using Reinforcement Learning: A Survey.* IEEE Transactions on Neural Networks and Learning Systems, 2018. **29**(6): p. 2042-2062.
18. Takada, K., H. Iizuka, and M. Yamamoto, *Reinforcement Learning to Create Value and Policy Functions using Minimax Tree Search in Hex.* IEEE Transactions on Games, 2019: p. 1-1.

Appendix B – NS2 functions

Part of the NS2 functions that include generating the secret key using Diffie Hellman, Authentication using MD5:

```
# Initialization (DiffieHellman)
proc dh_encrypt {message publicKey} {
    if {[lindex $publicKey 0] ne "publicKey"} {error "key handling"}
    set toEnc 0
    foreach char [split [encoding convertto utf-8 $message] ""] {
      set toEnc [expr {$toEnc * 256 + [scan $char "%c"]}]
    }
    return [modexp $toEnc $publicKey]
}

proc dh_decrypt {encrypted privateKey} {
    if {[lindex $privateKey 0] ne "privateKey"} {error "key
handling"}
    set toDec [modexp $encrypted $privateKey]
    for {set message ""} {$toDec > 0} {set toDec [expr {$toDec >>
8}]} {
      append message [format "%c" [expr {$toDec & 255}]]
    }
    return [encoding convertfrom utf-8 [string reverse $message]]

proc dh_slow_decrypt {input pkey} {
        upvar $pkey key

        if {[bitsize $key(n)] < [bitsize $input]} {
            error "keysize [bitsize $key(n)] must be greater than
text [bitsize $input]/$input"
        }
        if {[catch {set ptext [powm $input $key(d) $key(n)]}]} {
            puts "dh_slow_decrypt: $input [hex $key(d)] [hex
$key(n)]"
            error "powm error"
        }
        return $ptext
    }

    proc pack_text {ptext keylen} {
        # pack ptext with md5
        while {[string length $ptext] < ($keylen - 16)} {
            append ptext [binary format H* [::md5::md5 $ptext]]
        }

        if {[string length $ptext] < $keylen} {
            set md5 [binary format H* [::md5::md5 $ptext]]
            append ptext [string range $md5 0 [expr $keylen - [string
length $ptext] - 1]]
        }


    # encrypt a packet
    # packet format: [md5][length][payload][padding]
```

```
    proc encrypt_packet {ptext pkey} {
        upvar $pkey key

        set plen [binary format I [string length $ptext]]
        set md5 [binary format H32 [::md5::md5 $ptext]]

        set ptext ${md5}${plen}$ptext

        return [encrypt $ptext key]
    }

    proc decrypt {ctext pkey} {
        upvar $pkey key

        set keylen [bytesize $key(n)]

        binary scan $ctext H* block
        append ptext [hex [dh_decrypt 0x$block key]]

        return [binary format H* $ptext]
    }

    # decrypt a packet
    # packet format: [md5][length][payload][padding]
    proc decrypt_packet {ctext pkey} {
        upvar $pkey key

        set ptext [decrypt $ctext key]
        binary scan $ptext a16I md5 plen
        set ptext [string range $ptext 20 end]
        set ptext [string range $ptext 0 [expr $plen - 1]]

        set md5calc [binary format H* [::md5::md5 $ptext]]

        if {$md5calc != $md5} {
            error "packet checksum failed $md5calc != $md5: $plen /
$ptext"
        }
        return $ptext
    }

    namespace export encrypt* decrypt*
}



#Generating public key and private key for Diffie Hellman
set sNM [RandomInteger 336]
set QNM [expr $sNM*$P]
puts "NM publishes the System parameters
{$l,$cag,$cmg,$q,$P,$e,$HL,$hL,$QNM}"
for {set i 0} {$i<$val(nn)} {incr i} {

}
set ti 0
$ns at $ti "$ns trace-annotate \"NM publishes the System parameters
{$l,$cag,$cmg,$q,$P,$e,$HL,$hL,$QNM} \""
for {set i 0} {$i<$val(nn)} {incr i} {
```

```
      if {$i!=$NM} {
            set null($i) [new Agent/LossMonitor]
            $ns attach-agent $n($i) $null($i)
            set cbr($i) [attach-cbr-traffic $n($NM) $null($i) 10
0.05]
            $ns at $ti "$cbr($i) start"
            $ns at [expr $ti+0.2] "$cbr($i) stop"
            set ti [expr $ti+0.2]
      }
}

#Authentication
proc SRM_Proc {wban_id AP_id} {
      global ti Tdash U v tc T P array names array names n indCv HL
array named Q2 array names s1 array names S2 array names QAP ns
      set k [myRand 3 34]
      set t [myRand 3 34]
      set T [expr $t*$P]
      set Tdash [expr $t*$QAP($AP_id)]
      set tc [$ns now]
      set v [HashFun $tc $T]
      set U [expr ($k*$S2($wban_id))-
($v*$s1($wban_id)*$Q2($wban_id))]
      set session_key [HashFun $v $T]
      set service_request_msg [list $v,d $U, $indCv($wban_id), $HL,
$tc, $Tdash]
      puts "service_request_msg: $service_request_msg"
      $ns at $ti "$ns trace-annotate \"The $wban_id send the service
request:$service_request_msg to AP:$AP_id\""
      set null2 [new Agent/LossMonitor]
      $ns attach-agent $n($AP_id) $null2
      set cbr2 [attach-cbr-traffic $n($wban_id) $null2 10 0.05]
      $ns at $ti "$cbr2 start"
      $ns at [expr $ti+1.0] "$cbr2 stop"
      set ti [expr $ti+1.5]
}
$ns at $ti "SRM_Proc $wsn_node_id $ap_node_id"
set flag 0
#AP checks the validity
proc Verification {wbasid apid} {
      global ns array names n ti flag U v tc T Tdash session_key P
array names sAP array names indCv
      set r [expr [e_proc $U $P]*$indCv($wbasid)]
      set Tv [expr (1/$sAP($apid)*$Tdash)]
      set vv [HashFun $tc $T]
      set session_keyv [HashFun $v $T]
      $ns at $ti "$ns trace-annotate \"The Ap:$apid checks the
validity\""
      $ns at $ti "$ns trace-annotate \"The session key is
$session_keyv\""
      if {$session_keyv==$session_keyv} {
            puts "Valid"
            $ns at $ti "$ns trace-annotate \"Valid\""
      } else {
            puts "Invalid"
            $ns at $ti "$ns trace-annotate \"Invalid\""
            set flag 1
      }
```

```
      set ti [expr $ti+1.0]
      return $session_keyv
}
$ns at $ti "Verification $wsn_node_id $ap_node_id"
set null3 [new Agent/LossMonitor]
proc DataTransmission {} {
      global array names n ns wsn_node_id ap_node_id ti flag null3
      if {$flag==0} {
            $ns at $ti "$n($wsn_node_id) add-mark m2 purple square"
            $ns at $ti "$n($ap_node_id) add-mark m1 purple square"
            $ns attach-agent $n($ap_node_id) $null3
            set cbr3 [attach-cbr-traffic $n($wsn_node_id) $null3 100
0.05]
            $ns at $ti "$cbr3 start"
            $ns at [expr $ti+3.0] "$cbr3 stop"
      } else {
            $ns at $ti "$ns trace-annotate \"$wsn_node_id Quit the
Current Session\""
      }
}
$ns at $ti "record"
$ns at $ti "DataTransmission"
#Record the network parameters
#record the events
set pm 0
set pd 0
set nps 0
set pm 0
set pm1 0
set tim 0.5
proc record {} {
      global null3 ns pd nps pm en pacdelrat paclossrat e2edelay pm1
tim th bestpath array names energy array names path_length array
names aroute
      set now [$ns now]
      set time 0.5
      set pd [$null3 set npkts_]
      set pl [$null3 set nlost_]
      set dly [$null3 set lastPktTime_]
      puts $e2edelay "$tim $dly"
      set nps [expr $pd+$pl]
      set ut [expr $now+$time]
      set thpt [expr $pd/$time]
      puts $th "$tim $thpt"
      if {$nps!=0} {
            set pdr [expr $pd+0.0/$nps+0.0]
            set pm [expr $pm+5]
            puts $pacdelrat "$pm $pdr"
      }
      set t [expr [$ns now]+$time]
      puts $paclossrat "$tim $pl"
      set tim [expr $tim+0.5]
    $ns at [expr $now+$time] "record"
}
```

Appendix C – Acronyms and abbreviations

A

AASR
Authenticated Anonymous Secure Routing 13
AES
Advanced Encryption Standard 48
AODV
Ad-hoc on-demand Distance Vector 29

B

BId
Broadcast Identifier 29

C

CA
Certificate Authority 43
CPU
Central Processing Unit 6

D

DES
Data Encryption Standard 48
DId
Destination Identifier 29
DoS
Denial of Service 9
DSDV
Destination Sequenced Distance Vector 18
DSeq
Destination Sequence number 29
DSR
Dynamic Source Routing 25
DYMO
Dynamic MANET on Demand 12

L

LAN
Local Areea Network 2

M

MANET

N

O

P

Q

R

S

T

W

Z

نموذج تعزيز المصادقة على أساس بروتوكول توجيه DYMO لـ MANET

إعداد الطالب: محمد راجح علي عياد

إشراف: د. رشدي حمامرة

الملخص

شبكة ad hoc المتنقلة (MANET) عبارة عن مجموعة من العقد المتنقلة التي تشكل معاً شبكة، هذه الشبكة لا تمتلك بنية أو طوبولوجية ثابتة. مثل هذه الشبكات ممكن أن تعمل في مناطق صعبة جداً، مثل مناطق الحروب أو المناطق التي لا يمكن للمستخدمين التواصل معاً بشكل مباشر عن طريق ابراج الاتصالات. في حال أرادت إحدى العقد المتحركة أن تتواصل مع عقدة أخرى موجودة في نطاق أرسالها فيمكن أن تتواصل معها مباشرة، ولكن إذا كانت العقدة المستقبلة غير موجودة في نطاق إرسال العقدة المرسلة، فان العقد الموجودة بينهما ستقوم بتمرير الحزمة من عقدة الى أخرى، في هذه الحالة ستتصرف العقد البينية كجهاز توجيه.

ستقوم العقدة التي تريد الاتصال بعقد أخرى باستخدام أحد بروتوكولات التوجيه المعروفة لإيجاد أقصر طريق بينها و بين العقدة المستقبلة، هذا الطريق الأقصر يعتمد على عدد القفزات البينية و رقم تسلسل فريد لكل عقدة، من أهم بروتوكولات التوجيه التي تعتمد على إيجاد الطريق الأقصر هي AODV و DYMO و غيرها.

نظراً لطبيعة MANET المميزة فإن هناك العديد من التهديدات التي تواجهها عملية تبادل الحزم، لكن البروتوكولات المعروفة لا تمتلك القدرة على إستيعاب هذه التهديدات الأمنية بين العقدة المرسلة و العقدة المستقبلة، بسبب ذلك، تم طرح النموذج الجديد لتعزيز الأمان في MANET.

النموذج الجديد يستخدم تقنيات التشفير و المصادقة لتأمين حزم المعلومات، كما يستخدم تقنية تعزيز التعلم لتحسين التعاون بين العقد في MANET والتي ستقوم بزيادة الأداء والفعالية للشبكة.

تظهر نتائج المحاكاة بعد إستخدام مؤشرات الأداء أن النموذج المقترح قدم دوراً كبيراً في تحسين فعالية أنشطة MANET.