RECEIVER-ORIENTED ROUTING PROTOCOL FOR MOBILE AD HOC NETWORKS

By

Abdulaziz Yahya Yahya Al-nahari

Supervisor

Dr. Saleh Hosni Al-Sharaeh

This Thesis was submitted in Partial Fulfillment of the Requirements for the Master's Degree in Computer Science.

Faculty of Graduate Studies

The University of Jordan

May, 2009

COMMITEE DECISION

This Thesis (Receiver-Oriented Routing Protocol for Mobile Ad Hoc Networks) was Successfully Defended and Approved on May 4, 2009.

Examination Committee	**Signature**
Dr. Saleh Hosni Al-Sharaeh, (Supervisor) Assoc. Prof. of Distributed Computing and Wireless Networks	---------------------------------
Dr. Mohammad Sulieman Qatawneh, (Member) Assoc. Prof. of Network and Parallel Processing	---------------------------------
Dr. Basel Ali Mahafzah, (Member) Assist. Prof of Distributed and Parallel Computing	---------------------------------
Dr. Ayad Mohammad Salhieh, (Member) Assist. Prof of Wireless Communications and Sensor Networks (Jordan University of Science and Technology)	---------------------------------

DEDICATION

I like to dedicate this work to my parents and family who gave me the endless support and faith that makes me reach this point.

ACKNOWLEDGMENT

I would like to thank my thesis supervisor, Dr. Saleh Al-Sharaeh for his great encouragement, supports, and advices. And also to Dr. Wesam Al Mobaideen for his valuable continued support during the work of this thesis. Special thanks with all appreciation and respect to all my teachers, and the discussion committee for their insightful comments.

To my family, specially my parents and my brother Ibrahim who supported me during my study. To all my brothers and my sisters.

Last and not least, to the Yemeni Ministry of Higher Education for endless support that make it easier for me to reach this level in my life, and to my friends who stood beside me, especially Khaled Hushaidan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACK	ACKnowledgement
ACO	Ant Colony Optimization
AntHocNet	Ant Hoc Network
AODV	Ad hoc On-demand Distance Vector
CSI	Channel State Information
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CTS	Clear-To-Send
DCF	Distributed Coordinate System
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
GloMoSim	Global Mobile Information System SIMulator
IARP	Intrazone Routing Protocol
IERP	Interzone Routing Protocol
IETF	Internet Engineering Task Force
MAC	Medium Access Control
MANET	Mobile Ad hoc Wireless Network
MACAW	Multiple Access with Collision Avoidance for Wireless
MPR	MultiPoint Relay
MRL	Message Retransmission List
OLSR	Optimized Link State Routing
PARSEC	PARallel Simulation Environment for Complex systems
RWP	Random Waypoint Model

RERR	Route Error
RICA	Receiver-Initiated Approach for Channel-Adaptive On-Demand Routing
RORP	Reciever-Oriented Routing Protocol
RREP	Route Reply
RREQ	Route Request
RTrack	Route Track Packet
RTS	Request-To-Send
RUPD	Route UPDate
WiFi	Wireless-Fidelity
WRP	Wireless Routing Protocol
ZRP	Zone Routing Protocol

RECEIVER-ORIENTED ROUTING PROTOCOL FOR MOBILE AD HOC NETWORKS

By

Abdulaziz Yahya Yahya Al-nahari

Supervisor

Dr. Saleh Hosni Al-Sharaeh

ABSTRACT

Mobile ad hoc network (MANET) is a self-organized and dynamic deployment infrastructureless wireless network where the mobile nodes form in an ad hoc manner a communication topology. In MANETs, each node plays as a router besides being as a host. Each node is capable on relaying the data to the appropriate next node. The data is moving in multihop among the nodes until it reaches its destination. The nature of the wireless network beside the ad hoc manner and the mobility of the nodes build a barrier on front of the sender of the data to find a suitable available path for the destination. The sender is facing many challenges consequence from the mobile ad hoc networks like the dynamic changing in network topology, the limited resources of bandwidth, power, etc. For these problems and challenges, different algorithms for different layers have been proposed to adapt the MANETs behavior.

We investigate in this thesis on routing algorithms for mobile ad hoc networks. We propose a new hybrid routing algorithm a Receiver-Oriented Routing Protocol (RORP) that combines both reactive and proactive algorithms advantages, in order to increase the overall wireless network performance.

We implement and evaluate the tests of the proposed RORP algorithm which is based on the Global Mobile Information System SIMulator (GloMoSim) network simulation. The test of RORP goes through different simulation experiments such as mobility and node density. We show the results of different performance metrics for real time applications (using the constant bit ration as a transmission type), such as the average end-to-end delay. The average end-to-end delay is significantly improved where the decrease of the delay that achieved is more than 150% compared to Ad hoc On-demand Distance Vector (AODV) and Wireless Routing Protocol (WRP) routing algorithms. In addition, the results show that the delivery ratio and control overhead are improved as well.

Chapter 1

Introduction

1.1 Ad hoc Wireless Networks

With the increasing use of wireless communication along with the amazing fast growth of the wireless devices (ex. handheld, sensors, mobile phones, wireless laptops, etc.); it becomes one of the most important research fields in recent years [Pathan and Hong, 2009]; [Conti M., 2003]. As a result, different wireless technologies and standards have been grew and developed like, Wireless-Fidelity (WiFi) [IEEE 802.11, 2009], Bluetooth [IEEE 802.15, 2009], and Wireless Metropolitan Area Networks, also known as WiMax [IEEE 802.16, 2009].

Wireless networks are either infrastructure or infrastrucureless networks. In infrastructure networks, the mobile nodes are communicating via a centralized fixed infrastructure or base station that manages the communications. The nodes are directly communicated with the base station in one-hop distance, and the base station communicates with the other nodes either directly or via other base stations. On the other hand, the infrastructureless networks are multihop networks where no need for base stations to manage the communication between the nodes. The infrastructureless wireless networks are also referred as ad hoc wireless networks. The ad hoc networks can be in different topologies, mesh networks, sensor networks, or mobile ad hoc networks [Siva Ram and Manoj, 2004].

Mobile ad hoc wireless network (MANET) represent complex distributed systems where the mobile nodes are freely and dynamically self organized into arbitrary network topology Figure 1.1. These networks are connected and installed without pre-existing administrative base station [Jayakumar and Gopinath, 2007]. This kind of network is useful in such applications that do not have enough of time to install or find resources to configure such as military applications, emergency and rescue applications, conferences and meetings [Siva Ram and Manoj, 2004].

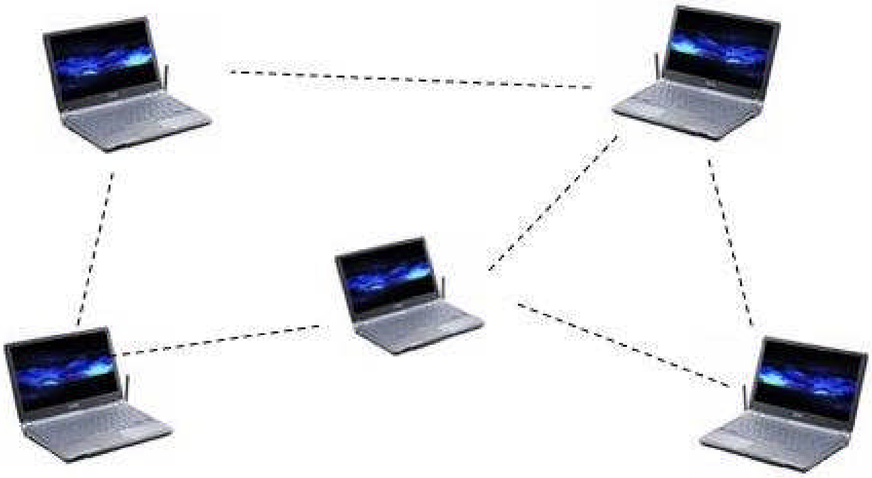


Figure 1.1: Mobile ad hoc Wireless Network

To facilitate the communication within the network, a routing protocol must be used efficiently to find a suitable route between the nodes in order to successfully deliver data. Many routing protocols for MANET have been proposed. These protocols are facing many challenges from the behavior of MANETs. The absence of the fixed infrastructure makes every node plays as a router besides being a host node to manage the flow of the data packets comes in and out of it, and to discover and maintain routes in the network. With these facts of MANET, routing algorithms start seeing the lights trying to adapt to these challenges. That makes a wide area for the researchers to evaluate these protocols or do improvements that raise the performance of the algorithms [Pathan and Hong, 2009]. There are several problems that the routing algorithms face. The routing algorithm first duty is to find or have a path available between the source of the data packets, and the destination. Keeping the path with the dynamic nature of MANETs is not an easy task due to dynamic and unstable topology caused by the dynamic movement of the nodes.

The connectivity between two nodes may change while it is in use and one of the nodes is involved in the route as an intermediate node. In addition, the limitation in resources such as transmission power, energy and battery life time, and bandwidth are also an important issue that must be taken into consideration when designing a routing algorithm for MANETs. The network size is also dynamic. It can be increased by adding new nodes, or decreased when some nodes may turn off or nodes leave the network. Thus, scalability is an important issue that should be considered.

Routing algorithms can be classified into two major categories: proactive routing algorithms and reactive routing algorithms. The proactive routing algorithms are table-driven routing algorithms. The information of the network is propagated in the network in order to update the topology information. The reactive routing algorithms are on-demand algorithms. The information about any route in the network is retrieved when demanded by proceeding two processing phases, the discovery phase, by finding the route, and the maintenance phase, when trying to keep the route or replace it with better one. Both types have advantages and disadvantages. Thus, merging between the two algorithms to take the advantages and benefits of them in one algorithm is preferred in some cases [Pathan and Hong, 2009]. This kind is called hybrid routing algorithm.

In this thesis, we propose the Receiver-Oriented Routing Protocol (RORP), a routing algorithm that comes under the hybrid routing algorithm category for MANETs. This algorithm takes the advantages of the proactive algorithm and reactive algorithm and enhances the performance of the routing for MANETs. We employ the reactive algorithm in finding the route between two nodes, the source and destination (sometimes referred as sender/receiver nodes), to establish a connection session between them, and the proactive in order to do the maintenance of this connection session between the two nodes.

1.2 Thesis Contribution

The aiming of this thesis is the mobile ad hoc wireless networks. We have studied the challenges and performance metrics behind designing the routing algorithms. In addition, this thesis proposes a new routing algorithm (RORP) that works with MANETs in an efficient way by combining the reactive and proactive routing. We compare the performance of this algorithm with some of the state-of-arts routing algorithms using different situations that affects the efficiency of the routing algorithms. The main performance metrics that we used to evaluate RORP algorithm are packet delivery ratio, average end-to-end delay, throughput, and control overhead. Testing these metrics under the RORP algorithms shows that, the average end-to-end delay is improved significantly along with the control overhead, whereas the delivery ratio and throughput have improved quietly with some other routing algorithms

The RORP routing algorithm has been implemented and tested using the Global Mobile Information System simulator (GloMoSim) [Bajaj L. et al., 1999]. Glomosim is a modular library designed for parallel algorithms of wireless networks. It is a scalable simulation library where built using the Parallel Simulation Environment for Complex systems (PARSEC) environment.

1.3 Thesis Outline

This thesis is organized in the following chapters: Chapter 1 is the introduction where we introduce the MANET and the related routing protocols, then state the problem issues that should be considered when designing a routing algorithm. Finally, we state the contribution of the thesis. In Chapter 2, we study the routing protocols in MANETs and discuss different issues, designing and performance issues, for the routing in MANETs. After that, we briefly describe the importance of the MAC protocols in supporting the routing protocols. Then, we study the different categories of routing protocols and investigate on some of the state-of-are routing protocols and other routing protocols where the receiver is playing a major role in the routing algorithms. In Chapter 3, we introduce the proposed routing algorithm (RORP), illustrate the theory of the algorithm and discuss the different stages the algorithm is follow. In Chapter 4, we show the simulation results and discussions, and comparing it with certain known routing algorithms in the literature. Finally, Chapter 5 includes the conclusion of this work and the future works.

Chapter 2

Routing in Mobile Ad hoc Wireless Networks

In this thesis, we are proposing a routing algorithm for mobile ad hoc networks. In the following, we have focused on the mobile ad hoc networks, and issues behind designing a routing algorithm that adapts to the characteristics of the ad hoc networks. Then, we describe the different types of routing algorithms and give different algorithms for the different types.

2.1 Mobile Ad hoc Networks

Mobile Ad hoc networks (MANETs) can be defined as a collection of routers (nodes) connected to each other via the wireless link and dynamically forming a network for temporal time without the use of any infrastructure or centralized administration. The network in ad hoc mode deployed with no preplanning needed and the changes in the network can be done with minimal extra work. The movement of the routers are not restricted that makes the topology of the ad hoc networks change rapidly and unpredictably. Such a network can be used in a standalone fashion or connected the larger internet Multihop, mobility, large network size, bandwidth and battery constraints make the design of suitable routing protocols a major challenge [Stefano et al., 2004].

While the topology is essentially dynamic, it can be increased by adding new wireless routers, reduced by removing nodes, or changed in a continuous way if some of the nodes are mobile. Such dynamic behavior involves an important new challenge in networking technology, while in wired networks, changes happen infrequently. In addition, MANETs rely exclusively on wireless links. This means that data transmission is less reliable, and that there is less available bandwidth. MANETs are usually highly decentralized, lacking hierarchy or central controller.

The nodes of MANETs often have limited resources, memory, processing power, or battery energy. The last important challenge is the scalability where the MANETs are expected to grow to large size.

The MANETs are easy to deploy in fast and simple ways, it just needs at least two mobile nodes. This kind of network is useful in such applications that do not have enough of time to install or find resources to configure such as military applications, emergency and rescue applications, conferences and meetings, and also with sensor networks [Siva Ram and Manoj, 2004].

Therefore, addressing these problems, the routing algorithm must be designed to adapt to these challenges in a way to avoid falling in the consequences from these issues.

2.2 General Issues for Designing Mobile ad hoc Wireless Networks Routing

In this section, we study the different major challenges that a routing protocol faces and discuss the performance issues for MANET routing protocols. Then, we study the impact of the MAC protocols on the routing protocols especially the hidden and exposed problem and the algorithm that is used to reduce these problems.

2.2.1 Issues in Designing a Routing Protocol

The routing protocols as we discussed earlier, are facing different challenges for providing a path between two nodes and relay information between them. Here we study the major challenges as stated by Carson and Macker [1999].

2.2.1.1 Node Mobility

The network topology is dictated by the placement of the nodes. Then, if one node changes its place, that means a new network topology. The routing protocol aims to deliver data from end node into another end node through multihops called intermediate nodes. Movement of one node will cause losing the path, therefore periodic path break. The speed of the node while moving and also how much this node stays in one place before moving. This technique is usually simulated using a model called the Random Waypoint Model (RWP). The RWP mobility model is a widely utilized model to create mobility scenarios in mobile ad hoc network research [Tavli and Heinzelman, 2006]. The node by this model chooses a random destination location and moves to it in a straight line with random speed. Then, the node will wait for a periodic time (pause time) then choose a new location and so on. Mobility model does not produce uniform node distribution in the network. Instead, the node density at the center is higher than the node density at the other parts of the network [Bettstetter et al., 2003]. This is the most popular model in mobility behavior that is used when designing a multihop routing protocol.

2.2.1.2 Bandwidth Constraint

The bandwidth constraint and also the availability of network capacity are limited in wireless networks. On the other hand, wired network can have maximum capacity just by using fiber optics, as an example. As illustrated in the IEEE 802 Standard for physical layer specification [IEEE Standard 802.11, 2007], there are limited throughputs for the channel. The high throughput provided is 54Mbps (IEEE 802.11a and IEEE 802.11g). This limited capacity results in less data rates that can be offered through the wireless channel. In addition, when the network topology is changing with the varying time, the information about the topology is flooded as a results it consumes the bandwidth, because it involves control overhead. For that, more bandwidth is wastage.

2.2.1.3 Resource Constraint

There are two important resources that are limited for the nodes in MANETs which are battery life and processing power. Increasing the power processing will decrease the battery life and therefore the node will be less functioning. Either the node will be dead or the limitation of its reachability to other neighbor nodes will be restricted. This constraint is too important while it is defining how the node is available inside the network. Thus, routing protocols in MANETs must be optimally managing these constraints.

2.2.1.4 Shared Radio Channel

While the nodes share the same wireless link channel that involves a challenge in the congestion of the channel among the nodes. Also the collision of the data and control packets presented with the transmission in the ad hoc networks. This problem is known as the hidden and exposed station problems. The network layer along with the MAC layer should cooperate to find solutions for these problems and find different paths that can reduce the congestion of the channel.

2.2.2 Issues in the Performance of MANET Routing Protocols

This sub-section will illustrate some the performance metrics and characteristics that are used on judging of a routing protocol and measure its suitability and performance [Carson and Macker, 1999]. There are two metric types: qualitative and quantitative metrics. Some desirable qualitative metrics are:

- It must be fully distributed.

- Loop-freedom and free from stale routes.
- Adaptive to the frequent change in the topology.
- Demand-based operations: by letting the routing algorithm sends traffic packets just when needed. This can utilize the energy and bandwidth resource more efficient, but increases the delay.
- Proactive-based operations: this is desirable in cases where the latency is unaccepted that occurs with demand-based operations.
- Uni-directional link support.

The quantitative metrics that can be used to estimate the performance of any routing protocol:

- End-to-end throughput and delay: these two external measures are important to show the effectiveness of the routing protocol (how the protocol is doing its job).
- Route acquisition time: this metrics used for on-demand algorithms which measures the time needed to establish route when requested.
- Packet delivery ratio: this measure is an internal measure which tests the efficiency of the routing protocol. It measures the bit efficiency of delivering data through the network. It is computed as the average number of data bits transmitted to the average number of data bits delivered.
- Control packets: this measure is measured either by the number of packets transmitted, or by the number of control bits transmitted to data packet delivered. It is

an internal measurement that measures the bit efficiency of the protocol in expanding control overhead to delivered data.

The routing protocols should be designed with consideration to the previous issues that stated by the IETF MANET working group [Carson and Macker, 1999].

2.2.3 MAC Protocols for Mobile ad hoc Wireless Networks

Medium Access Control protocols play an important role in the performance of the MANETs which define how each mobile unit can share the limited wireless bandwidth resource in an efficient manner. When the sender needs to send data to a receiver, but this receiver is in a far place where direct communication between them is unavailable. In this case, the sender should send this data by multi-hop fashion and therefore the sender needs to access the medium. This accessing should be controlled in order to manage the medium and the transmission to avoid any interference or errors during the communication [IEEE Standard 802.11, 2007].

In wired networks, to avoid the interference of data packets in the channel, a popular algorithm *Carrier Sense Multiple Access with collision Detection* (CSMA/CD) was used. The nodes in the wired network, by using this scheme, sense the medium if either idle, so the node can send data via the wireless medium; or it is busy, so the nodes waited for certain time then do the sense again. And if a collision occurs, a broadcast error message is flooded to inform the shared nodes about it. But this scenario can not be used with wireless networks. The important problems that happen with wireless networks are the *hidden station problems* and *exposed station problems* as depicted Figure 2.1.

Hidden station problem occurs when there are nodes that not reachable by the sender nodes (hidden from the senders), where those nodes on the other hand, are reachable to the receivers

[Siva Ram and Manoj, 2004]. In this case, those nodes will check the medium as idle then may start sending to the receiver nodes hence, a collision at the receiver side occurs.

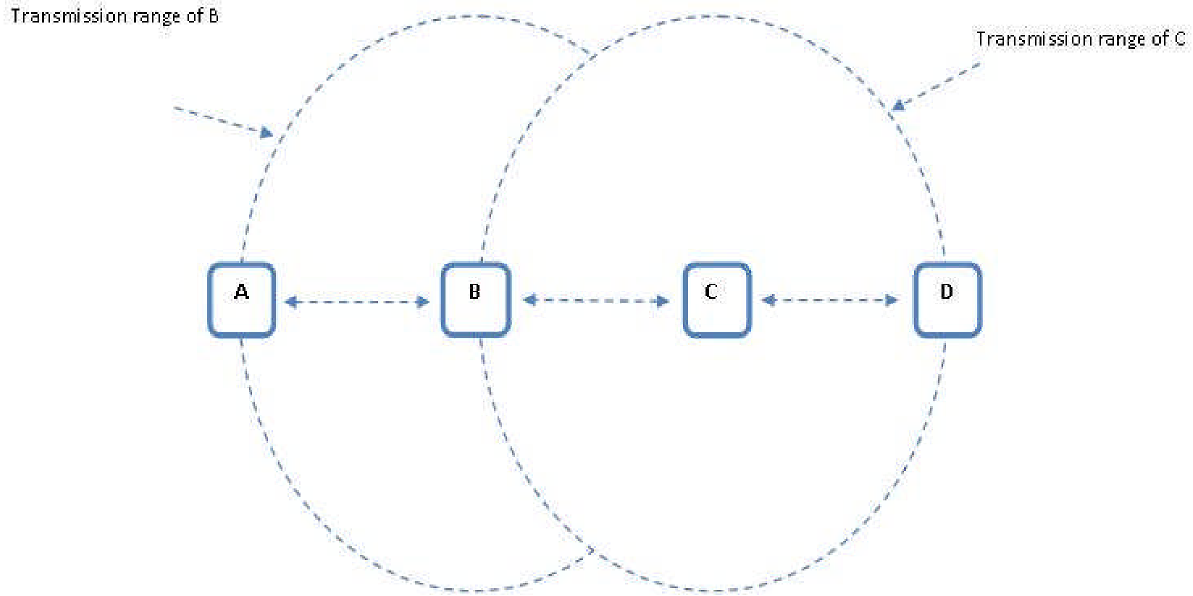


Figure 2.1: Hidden and Exposed Station Problems

In the Figure 2.1, let us assume that node C is sending data to node B. node A is in the range of node B and not belongs to the range of C. So, node A is hidden from node C and can send data to B at anytime. The Exposed Station problem is when there are nodes belong to a sender range and are frozen from sending data to other nodes while the channel is busy. So that, those nodes are exposed to senders but if they sends to other nodes will not cause interference to the data transmitted to the sender node. This problem is wasting the bandwidth due to the underutilization of the channel [Tavli and Heinzelman, 2006]. As example in Figure 2.1, if we assumed that station B is sending to station A, and station C wants to send data to D but should wait till the medium sensed from B side is free. These problems must be avoided to decrease the collisions in the medium with the absence of the collision detection in wireless networks.

IEEE 802.11 MAC sub-layer [IEEE Standard, 2007] defines two coordination functions: the Distributed Coordination Function (DCF), and the Point Coordination Function (PCF). DCF provides distributed channel access based on the *Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)*, while PCF provides contention-free centralized channel access control through *polling*. The IEEE 802.11 DCF MAC algorithm is a contention based single channel protocol. It uses both carrier sensing and extra control packets to limit the number of collisions. The basic working of IEEE 802.11 DCF is as follows. When a node wants to start a transmission, it listens to check whether the wireless medium is free. Next, it sends a request-to-send message (RTS) to the node it wants to transmit to, in order to request the start of a conversation. If this node is free to receive, it sends a clear-to-send message (CTS) back. The requesting node can then start the transmission of data, and if this is successful, the receiving node concludes the communication by sending out an acknowledgement message (ACK). Between each of the steps of this process, there are small waiting times to avoid collisions. If the process goes wrong at any time, it needs to be restarted completely, after a randomly chosen backoff interval. With each failed transmission attempt, a window from which this backoff interval is chosen is doubled. This mechanism is known as MACAW [Bharghavan et al., 1994]. It helps to decrease the interference in hidden and exposed station problems.

2.3 Routing Protocols in Mobile ad hoc Wireless Networks

The ad hoc wireless network is a set of nodes (each node considered as a router) that communicate to each other directly without the needing to a coordinator (e.g. access point) to manage this connection. The network topology of this network is a dynamic topology take a temporal shape that changed rapidly with the movement of the nodes. Besides that, using the space as the link for the communication rather than the wired, make the connection constrained by the limited bandwidth and resource [Siva Ram and Manoj, 2004]. Meeting these challenges is needed to be considering finding a path between two nodes with achieving high performance in terms of delivery ratio, delay, and control overhead when designing a routing protocol. Many studies have addressed these problems. The designing of routings was at first as improvement to the routing algorithm used with wired. The wired routing protocols are not suitable with the wireless networks for the problems that we have addressed earlier. The wired networks are fixed networks beside the stable link capacity that, on the other hand, is changing with the wireless nodes as soon as the power is getting low, noises, fading, and interference while the nodes are using a shared channel. The routing algorithms for wireless then must be adapted to hold these issues in their accounts. Corson and Macker [1999], from the Internet Engineering Task Force (IETF), address these problems and put rules for designing a routing algorithm for the mobile ad hoc networks and the performance issues for judging of these protocols.

In the coming sub-sections, we describe the classifications of routing protocols and the differences between them. Then, we briefly illustrate some of the state-of-art and important routing algorithms, and then the connected routing protocols to our proposed routing algorithm.

2.3.1 Proactive, Reactive, and Hybrid Routing Algorithms

The routing algorithms for mobile ad hoc network are classified according to several criteria [Jayakumar and Gopinath, 2007]. The classification will be either depends on the routing information updates strategy, the routing topology, or how to utilize a specific resource [Siva Ram and Manoj, 2004]. Here, we address the first criteria which considers about how to update the routing information. This type is classified into three types: proactive, reactive, and hybrid routing protocols.

Proactive routing protocol called also table-driven routing protocol. All the nodes of the network try to maintain consistent up-to-date routing information about the existing topology. To achieve that, the routing information need to be updated upon changes in the topology occur. Therefore, the nodes flood these changes to each reachable node. Most proactive routing protocols have inherited properties from algorithms used in wired networks [Sarkar et al., 2008]. But these algorithms adapted to be working with the dynamic behavior of the wireless network as discussed above. The nodes in the mobile ad hoc network proactively send the current state of the network. In addition the routes between the nodes are maintained even there is no need to send data. Therefore, the control overhead will be high. When any node needs to send data, the route is existed, as the last updates reached to this node, and no need to do a setup to the route before sending the data. This overhead on the other hand, will make drawback effect on the throughput and packet delivery ratio. Protocols in this kind like: Destination Sequenced Distance Vector (DSDV) [Perkins and Bhagwat, 1994], Wireless Routing Protocol (WRP) [Murthy et al., 1996], and Optimized Link State Routing (OLSR) [Clauses and Jacquet, 2003].

Reactive routing protocols used primarily to decrease the overhead in the proactive routings and the effects caused by it. The source nodes under this category search for routes to the destination just when needed to send data packets. For that, reactive routing protocols called on-demand routing protocols. Algorithms under this category follow two main processes: route discovery process and route maintenance process. The first process is by flooding the route request control messages searching for the destination. These controls will initiated when there are data needed to be sent. Upon reaching the destination, the destination will reply by unicast packet back to the source following the track the request packet came from. When the reply arrives to the source, the discovery phase is finished and the source is ready to send the data. After that, the maintenance process will start by checking the connectivity of every node with the neighbors, when losing the connectivity with downstream node (next hop toward the destination); the node will initiate a route error packet to the upstream node (toward the source) until reaching the source. The source will then, depends on the algorithm, repeat the route discovery in order to find a new available route. Protocols of this kind like: Ad hoc On-Demand Distance Vector (AODV) Routing [Perkins et al., 2003], and Dynamic Source Routing Protocol (DSR) [Johnson et al., 2007].

Jayakumar and Gopinath [2007] summarize the advantages and disadvantages of the two categories. Proactive routing protocol advantages are as follow: the routing information is always readily available whenever there are data need to be sent. In addition, all changes in the network are taken into account, so that the new routing opportunities can be exploited and back up paths can be provided when primary paths fails. On the other hand, the reactive protocol advantages stated in the reducing in control messages overhead. The disadvantages of both algorithms are that, proactive routing protocols are become quite inefficient when a lot of changes need to be tracked when the topology is high dynamic or the network size become larger [Stefano et al., 2004]. The reactive algorithms major disadvantage is the delay that occurs when the source is waiting till the discovery process completed.

Some studies have compared and evaluated the performance of both proactive and reactive routing algorithms, such as: [Mbarushimana and Shahrabi, 2007]; [Ashwini and Fujinoki, 2005]; [Clausen et al., 2002]; [Das b et al., 2000]; [Broch et al., 1998]; and [Das a et al., 1999].

Hybrid routing algorithms come to take the advantages in both reactive and proactive routing algorithms. The first routing protocol of this category is the zone routing protocol (ZRP) [Haas and Pearlman, b, 1999] where the topology divided into zones. The reactive process is done to connect nodes inside the zone with other nodes outside the zone. The proactive is concerning to make the connectivity of the nodes inside the nodes by the periodic routing information.

2.3.2 The State-of-arts Routing Protocols

In this section, we describe the most representative routing algorithms for mobile ad hoc wireless networks. For reactive routing algorithms we describe two algorithms: AODV and DSR, and three proactive algorithms, DSDV, WRP, and OLSR, and one hybrid routing algorithm, ZRP.

2.3.2.1 Ad hoc On-demand Distance Vector (AODV)

AODV [Perkins et al., 2003] is a reactive, multihop routing protocol. It is an on-demand routing where the algorithm starts when it is required by a node to send data to a target node, and continue processing until all the data sent. In addition, AODV uses the sequence numbers of both source (sender) and destination (receiver) with the route information. The destination sequence number, issued by the destination when a requesting route has arrived, is used to identify the recent available path from any node along to this destination. ADOV allows the intermediate nodes that are involved in the connection between the source and destination to react in two ways. First, it is by holding the information of the next-hop that the message will move to. The next-hop is either in a forward path, to the destination, and the reverse path, to the source when needed to be taken by the reply message from the destination. The second reaction is by informing the source with the changes in the connection so a link failure may occur and the connection between the source and destination is not available anymore, so the intermediate sends a message to the source.

When we go to the algorithm details, the AODV goes through two phases. The startup phase and the maintenance phase. In both phases, three main control messages are playing the role for making the processes to successfully send the data to the destination and to keep the session stays available. These messages are Route Request (RREQ), Route Reply (RREP), and Route Error (RERR) messages.

When a node has data and does not have any information about where and how to send this data, the node will flood a RREQ message asking about a route to the destination. The other nodes will hear this request as well. To avoid the looping every broadcast has a number helping the nodes to verify if this message is not known before so the request will be dealt by every node once. Now the message with intermediate node, if this request has an answer, the intermediate will reply back to the source, the sender, RREP message as unicast, or the request will be flooded again but here by the intermediate node with the same broadcast number, and recording in its routing table the node Id that the request came from to know in the future how to reach the source of this request. This process will continue until either one intermediate has the answer or the request reaches the destination. Upon this, the destination will issue the RREP as a unicast again hop by hop toward the source. When the reply arrives to the source node, the source will start sending the data hop by hop toward the destination.

During the communication, the link between the two nodes is vulnerable to a failure caused by the node movements or power conservation that effect on the connection makes the session between the source and destination to be lost. When a node is under this problem, it will forward a RERR message to the source telling that this route is not available anymore. When the source gets this message, it will do again a startup process and flood a RREQ message looking for a new path available toward the destination. When the destination or one of the intermediate nodes have valid information, a route reply will send back to the affected source and a new session is established. As reactive algorithm, the major disadvantage of this algorithm is the delay that occurs while maintaining the route by exploring a new route.

2.3.2.2 Dynamic Source Routing (DSR) Protocol

DSR [Johnson et al., 2007] is a reactive routing algorithm that designed in 1996 to adapts to the changes on the routes and topology dynamically by the movement of nodes. It needs, compared to proactive algorithms, a little overhead during the changes. The algorithm looks for route just when no valid route available to send the data to the destination. The source has a cache of different valid routes to the destination that offers many options when the active route becomes not valid so the next available route from the cache will be used. The extensive cache then increases the availability of routing information.

When a node has data to be forwarded to a specific destination, first it will check inside its cache for any routing information for this destination. If no information available, a Route Request message (RREQ) will be flooded to all the neighbors. The neighbors will forward this request and adds its name to the routing cache. This cache will be used by the source to decide the path when sending the data, and also the intermediate nodes will avoid looping when reading the nodes in the cache if not came before from this source. When the destination hear the requesting message, a route reply message will be sent either back from the path where the request came from or using, if exist, a path available in its routing cache to that source. At this point the session will be established and the source when the RREP arrives has in its cache an available route to the destination and will start sending the data. DSR is source routing, that means the path that the data will go through will not hop by hop like we explained with AODV. Rather, the source will adds the complete path to the data packet so the source has the control of the way of the packet from its leaving from it until it arrives to the destination. One point here is that every node but the destination is sending the RREQ message. To avoid looping or duplicate message requests that may available by different ways, the intermediate nodes use the source sequence number to distinguish between the requests come from the same source. When an error happens with an intermediate node that is active in the session, a Route Error message (RERR) will be sent back to the sender. The sender will search in the cache for valid route to the destination or if not available will start a route discovery again. This cache can play a bad role during the communication session by holding an information for a route to a destination that is not valid anymore but still saved in the cache. When the sender node receives the RERR and uses this path, the data will be lost beside a desirable overhead will be resume the bandwidth and in addition, the other nodes can also use this path when they read it with the data and have wrong information.

Disadvantage of this protocol is that the intermediate nodes do not locally maintain the route like in AODV. The intermediate nodes discover an error when they did not reach the next-hop that in the cache. At this moment the node will send a route error to the sender where upon receiving it the sender will reinitiate the route discovery process. In Addition, the delay in this protocol is higher than the in table-driven protocols (link state protocols) [Siva Ram and Manoj, 2004].

2.3.2.3 Wireless Routing Protocol (WRP)

WRP [Murthy et al., 1996] is a proactive unicast routing protocol, addresses the count-to-infinity and routing loops problems in Bellman-Ford algorithm used by distance vector routing algorithm, and proposed to adapt to the changes in the network with fast convergence. Every node in the WRP algorithm is setting up four tables:

- *Distance table*: that shows a picture that the neighbors of a node have for the network. It contains a list of the distances to every reachable destination by each neighbor with the predecessor node to that destination.
- *Routing table*: contains fresh information for the available paths through the neighbors to reach the destinations. For each destination, the table holds its address, the shortest path (with shortest distance), the predecessor for this path to it, and a tag that state this path with either correct, error, or null path which means not known yet
- *The link-cost table*: holds the cost of the link between the node and its neighbors and the number of successfully update, and correct messages so that errors can be detected if an update message did not arrive since the last one reported in the table. The cost could be the hop count, the latency, or the throughput of that link to the neighbor.

- *The Message Retransmission List (MRL) table*: records the update messages that did not acknowledge by the neighbor, so that this message will retransmit to the neighbor again and wait for the acknowledgment.

Every node when receiving the update message will update the new distance information that comes from that neighbor and check for the distance of the other neighbors to decide if there is a shorter or better path and therefore, to send the update message to other neighbors in order to update the routing tables. The node also should send an acknowledgment to the neighbor for this message. In addition, the nodes will periodically, if there are no changes needed to be sent, send a hello message to inform the neighbor that the connection still alive between them and by that, it ensures the connectivity between them. From the distance table, if the periodic successful messages (ACK or hello messages) countered since the last one a periodic gab, then this link will be lost and the node should inform its neighbors about it with distance equal *infinity,* that means it is not available anymore again, and the node will see through its distance table if there is another way could be used to reach this destination rather than the broken one.

2.3.2.4 Destination Sequenced Distance Vector (DSDV) Protocol

DSDV is one of the few first routing algorithms in mobile ad hoc networks which published in 1994 [Perkins and Bhagwat, 1994]. It is an adaptation of the distance vector routing like the Routing Information Protocol (RIP). DSDV is a proactive hop-by-hop distance vector routing protocol, requiring each node to broadcast routing updates periodically. To understand this protocol, we first illustrate the distance vector mechanism. Every node i keeps set of distances for each destination x reached through the neighbor j $\{d^{x}_{ij}\}$ where in this set is the distance to the destination x from the source i through its next hop j. the value of the distance can be the number of hops. This distance estimated from the information arrived by the neighbor nodes where each node estimate the distance to every outgoing links, update their tables and forward them to other neighbors. The above description of distance vector is the classical Distributed Bellman-Ford algorithm where we mentioned in WRP algorithm [Sarkar et al., 2008], [Perkins and Bhagwat, 1994]. With DSDV, the algorithm uses the destination sequence number to solve the problems of routing loops and count-to-infinity that will be a result from applying the distance vector algorithms in wireless networks.

The data needed to be sent to a destination *x* will go through the neighbor *j* that will lead to the destination with shortest distance from *i* to *x*. To keep the set (table) of distance updated, each node must periodically recalculate the current outgoing distances and send the updated distance table, if a significant change has observed, to its neighbors. The table updates are of two types [Siva Ram and Manoj, 2004]: incremental updates and full dumps. Full updates are transmissions of a node’s entire routing table. Because the size of these updates scales with the size of the network, these updates are performed relatively infrequently. To reduce processing overhead and bandwidth consumption, incremental updates are transmitted more frequently. Incremental updates include only those routing table entries that have changed size since the last full update [Stefano et al, 2004]. The destinations when sending update tables, their sequence number is incremented to be greater than the previously known one. All these updates help the nodes to have correct estimations of the costs for all paths and therefore a valid view of the network topology.

2.3.2.5 Optimized Link State Routing (OLSR) Protocol

The Optimized Link State Routing (OLSR) [Clauses and Jacquet, 2003] protocol is a proactive link-state routing protocol. OLSR is an optimization for the classical link state protocols which optimized for the mobile ad hoc networks. The key behind the OLSR is the use of the Multipoint Relay node (MPR). This node is selected by the neighbors. Every node will select a subset from its neighbors as MPRs. These MPRs are the nodes who are able to retransmit the packets to other nodes in two hops away, whereas other neighbors not MPRs to the node will receive messages but will not retransmit them farther as in Figure 2.2 [Stefano et al., 2004]. The MPR node has a list of nodes that selected it and those nodes are in one-hops far from it. When a node sends a message packet to its MPR, the MPR will resend this message to all the nodes in the lists [Jayakumar and Gopinath, 2007]. This technique significantly reduces the control overhead that occurs by proactive protocol when retransmitting the updated routes to all the nodes in the networks. MPR nodes have two main responsibilities: for forwarding control traffic, and when declaring the link sate information in the network where the shortest path routes to all destinations are known through the MPR nodes. Theses information will be announced periodically to the MPRs selectors as control messages. The route between source and destination can be defined as a set of hops through the MPRs.

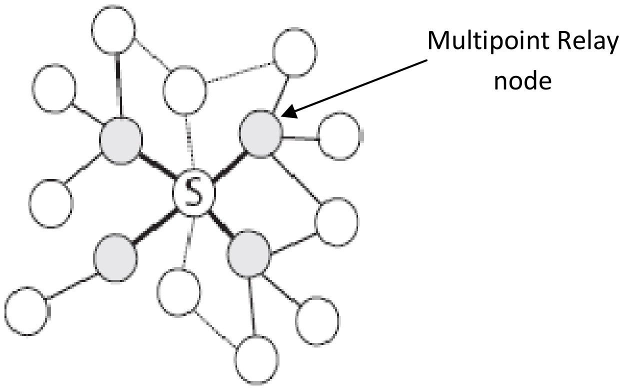


Figure 2.2: OLSR Multipoint Relay

Mbarushimana and Shahrabi [2007] have compared the OLSR, AODV and DSR algorithms. Depends on their simulation tests and environments, the results show that OLSR outperforms AODV and DSR in terms of delivery ratio and average end-to-end delay. The aim of their study is to compare reactive and proactive protocols. So, they conclude that proactive is performing better than reactive routing protocols.

2.3.2.6 Zone Routing Protocol (ZRP)

ZRP [Haas and Pearlman b, 1999] is the first hybrid routing algorithm that combines the best features in the reactive and proactive routing protocols. Each node has its own zone where the node become in the center of the zone and have a radius zone equal to the number of hops from the node as in Figure 2.3 [Stefano et al., 2004]. The nodes of a zone are categorized under two groups:

- Peripheral nodes: the nodes where the minimum distance (number of hops) to the central node is exactly the zone radius (i.e. the nodes at the border zone)
- Interior nodes: the nodes where the minimum distance is less than the zone radius.

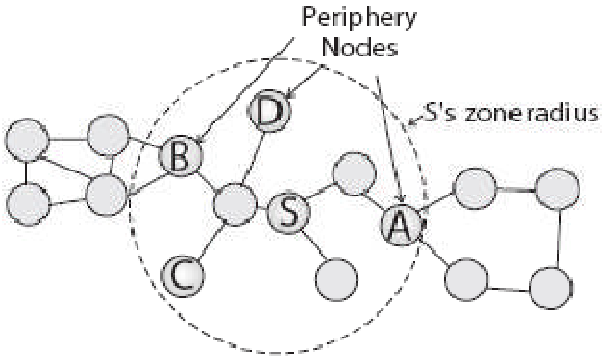


Figure 2.3: ZRP Zone Radius

The proactive part in the ZRP protocol concerns on up-to-date the information for each node about the nodes that comes under its zone. This is the responsibility of the Intrazone Routing Protocol (IARP) which is a locally proactive component belongs to the family of proactive, link state routing protocols [Stefano et al., 2004]. The changes in the nodes that belong to the node zone are periodically send a Hello beacon for the connectivity and the information is updated in the neighbor table. If after certain time from the last beacon no farther messages arrived, the node will be removed. These messages triggered to the IARP by the MAC layer using the Neighbor Discovery Protocol. By this strategy, the node can know if there is a new neighbor enters to its zone or a failure with existing one occurs. The reactive part is concerned with dealing with nodes come outer the zone using the Interzone Routing Protocol (IERP) which is a globally reactive routing protocol. The Peripheral nodes play the important role by treated as gateway nodes. When a node needs to send data to nodes not belongs to the zone, IERP bordercasts a query message to the peripheral nodes. Each peripheral node has also its own zone. If the destination belongs to its zone, then the reply by this peripheral will be directly sent back to the sender node. Otherwise, the peripheral node will bordercast the query message request again to its peripheral nodes. This query will just treated by nodes that not already received this query so the overhead is decreased. The process will continue until the destination found. Then, the route reply packet will be sent back to the sender using the hops the request came from. The use of zones in reactive process reduces the overhead that can be resulted by other reactive routing algorithms. Haas and Pearlman [a, 2001] stated that ZRP performance reduces the messaging and processing overhead by using the query control mechanism. In addition, ZRP reduced the overhead of flooding the RREQ mechanism in on-demand routing protocols and the periodic routing information flooded by the table-driven routing protocols [Siva Ram and Manoj, 2004]. The indicating of the zone radius is playing a major role in the efficiency of the ZRP. If the radius is equal 1, then the algorithm will be a pure reactive protocol, where the entire query message will be flooded to the whole peripheral nodes which will be here like ordinary neighbor nodes. This zone radius therefore, will draw the limits of using a proactive mechanism in the protocol.

2.3.3 Additional Related Routing Algorithms

In this section, we describe some routing algorithms that the destination plays a major role in finding and choosing the routes between the peers. Here we described three routing algorithms: two hybrid routing algorithms, AntHocNet and RICA, and one reactive routing R-AODV.

2.3.3.1 AntHocNet Routing Protocol

AntHocNet routing protocol [Di Caro et al., 2005], is a hop-by-hop, multipath, hybrid routing algorithm combines the advantages of both reactive and proactive protocols. This algorithm is based on specific self-organized behavior of ant colonies, the shortest path discovery, and the ant colony optimization framework (ACO) [Dorigo and Stützle, 2004]. The session setup between the source and destination starts reactively when the node needs to send data. This setup process starts by sending the *reactive forward ants* to find different paths. Each ant maintains a list of the nodes that has visited in order to avoid cyclic problem. Each node maintains a pheromone table which contains the cost of the routes that reach to the destination. This table is calculated by the *backward ants* where the destination sends it back hop-by-hop reverse the list of the nodes that available with the ants for every forward ant received. When the source receives these *backward ants* from different neighbors, it will decide who will be the next hop for the data from the pheromone table with best cost (number of hops and time delay). The intermediate nodes will do the same process with the data locally where the next hop can changed during the different received backward ants.

The proactive process starts by sending the proactive *forward ants* with the data to maintain the routes. These ants will be sent after *n* data packet sent by the source. If there exist in the intermediate nodes a next hop to the destination, then this forward ant will be sent as a unicast message otherwise it will broadcasted to explore new nodes. And the destination upon receiving these forward ants again, it will send back the backward ants. Each intermediate will therefore updates the pheromone tables.

The link failure is detected in AntHocNet protocol by using the periodic *Hello* message between the neighbors to insure the connectivity. If a next hop node in the pheromone table lost the connection with an intermediate node. Then this intermediate will change the pheromone table and inform the other neighbors about this change if there is no other path exists in its pheromone table reachable to the destination (i.e can be used rather than the lost one in the future).

Di Caro, et al. [2005] compared this algorithm with the ADOV in terms of average end-to-end-delay and deliver ratio. They show that the AntHocNet outperforms the AODV especially in more mobile and larger networks. In addition, Sivajothi and Naganathan [2007] also make a comparison for the video transmission and show how the AntHocNet outperforms the AODV in different metrics. The disadvantage of AntHocNet can be stated in the overhead of the controls especially with the diffusion of the ants during the proactive phase when the forward ants should be broadcasted again finding new routes to the destination where no previous pheromone table exist in the intermediate.

2.3.3.2 A Receiver-Initiated Approach for Channel-Adaptive On-Demand Routing (RICA)

RICA [Lin et al., 2002] is a hybrid algorithm designed to state the channel condition, where the routes between the source and destination change depends on it. Every terminal records the current state of the channel with its neighbors. The receiver is the responsible for initiating a control called *channel state information* (CSI) toward the source. This control is initiated upon receiving a route request along with the route reply message from the sender like in AODV route discovery process. The intermediate nodes measure the CSI (the throughput between the pair of nodes which then converted as number of hops) from the route request that comes from the sender and the route reply that comes from the receiver. This is the reactive process. After the setup of the session started, the destination periodically broadcast the CSI checking packet control to update the link state; the sender will collect the different CSI from different routes and will take the route with the best channel state information. The CSI can be measured as number of hops where the shortest path is the route that will be used by the sender. The sender when deciding, it will send a route update packet to the next hop (RUPD) informing it that it is ready for sending data packet The next hop will activate the link and send it to the next hop to it toward the destination until the path is activated and ready for data. For maintenance, the RICA algorithm nodes periodically send an ACK packet to test the connectivity between the nodes. If an error occurs, the node will send route error packet to the sender. The sender if no other routes are available by the CSI packets, the route discovery is restarted again.

2.3.3.3 Reverse AODV Routing Protocol

R-AODV [Kim et al, 2006] is an enhancement protocol to the AODV routing protocol. The aim of this algorithm is to avoid the lost of the reply packet to the source in high change in the network topology. The RREP will be not sent as unicast, rather the destination will broadcast the reply packet to the source and this control called reverse route request (R-RREQ). The delay that may occurs by the waiting for the reply by the source and may be increases due to the timeout for waiting and re-broadcasting the RREQ packet again will be reduced. The sender therefore will receive multiple routes by the different replies, R-REQ, and chose the best route among them to deliver the data. The first one received to the source will be used and the other will be saved for farther used.

The maintenance of the routes, like AODV algorithm, is due to the notification from the MAC layer about the connectivity with the neighbors. If the link with the downstream node is lost, then a RERR packet is sent to the upstream node till reaching the source node.

Chapter 3

Receiver-Oriented Routing Protocol for Mobile ad hoc networks

Receiver-Oriented Routing Protocol (RORP) is a hybrid routing algorithm in the sense of taking information or elements from both reactive and proactive algorithms. The reactive part in the algorithm is to setup the session between the sender and receiver, whereas the proactive part is the maintenance of this route. The algorithm is aiming to make it a destination (receiver) oriented where the receiver will be the manger of making the session (the route) with the source (sender) to be alive and available as much time as possible. It is also taking into consideration that this link is vulnerable to be broken from the intermediate side or even the sender and the receiver while moving or being down.

In this chapter, first we give a general overview about the RORP routing algorithm and the terms that used to describe it. Then, we discuss in details the components of the algorithm.

3.1 Overview

RORP is a hybrid routing algorithm that combines between the reactive and proactive routing algorithms. Like reactive routing algorithms (such as: AODV, DSR), the algorithm take the first phases which is, the startup phase or as known the discovery process. In AODV, the discovery process starts when the sender has data to be sent and there is no available active route to the receiver, where the sender floods Route Request (RREQ) to the destination and stay waiting for a reply. The reply may come from the intermediates or from the receiver itself. The RORP is in this phase is alike with AODV. The sender after sending the request will wait for a reply or in our strategy to a Route Track Packet, if already available. When the RREQ arrives to the receiver, a Route Reply (RREP) is sent back as a unicast to the sender through the same way where the RREQ came from. And after the session between the sender and receiver established and the sender started to send the data, the route is like a temporary route while the nodes (sender, receiver, and intermediates) are not in static mode and in anytime one may change its place. That means the topology will change too. Therefore, the route will be broke. For that, we make the receiver to help to decrease the occurrence of this problem. Here the receiver will start taking the charge of sending a packet broadcasted to the sender telling the sender to use a fresh route proactively. Here is the proactive part where, (unlike the reactive algorithms where no action proceeds before a problem occur) the receiver keeps sending every Ttime a packet to the sender upon the RREQ arrived. This packet called Route Track Packet (RTrack). When the sender receives the RTrack, it will check if there is a data buffered to be sent and send it through that way. Every RTrack has a broadcast number. If the new coming RTrack to the sender is holding the same broadcast number, then this packet will be dropped. This is because the sender will take just the first new arrival which indicated that the route to the destination through it is better than the others.

The intermediate nodes are playing a role in the algorithm in the sense that when a RTrack arrived, the node will relay that route as a broadcast to its neighbors if this packet was the first time to arrive and has a fresh packet comes from the receiver. Otherwise, this packet will be dropped. The intermediate nodes maintain a routing table describing the new updates and saving the node address that the RTrack came from, so it will be used later, for sending the data toward the destination.

In our proposed idea, we did not take into account the use of Route Error packet like that being used in the maintenance phase as other routing algorithms. When an error occurs due to losing a neighbor connectivity, the only procedure is to re-activate, if exists, the route to the destination in the routing table. Therefore, if data came from the sender, this data will be dropped. On the other hand, if an RTrack came from the receiver it will act like a new RTrack and will be relayed to other neighbors.

Data packet here is transmitted by the sender to the next neighbor where the link is active. The decision of taking the next step is decided locally by each node the packet arrives to. The data take a hop-by-hop like in AODV and because of the periodicity of RTrack, it may choose different routes.

3.2 Justification

To justify and explain our proposed idea, we made a scenario like in Figure 3.1. Node 3 wants to send data to node 1. In the scenario, all the nodes are static but node 1. The simulation time was 800 seconds and node 1 throughout the simulation period, it moves as shown in Figure 3.1 at the times of 100, 200, and 300, and 400 seconds. Node 3 will send a data packet every 2 seconds (light network load), with 375m transmission range.

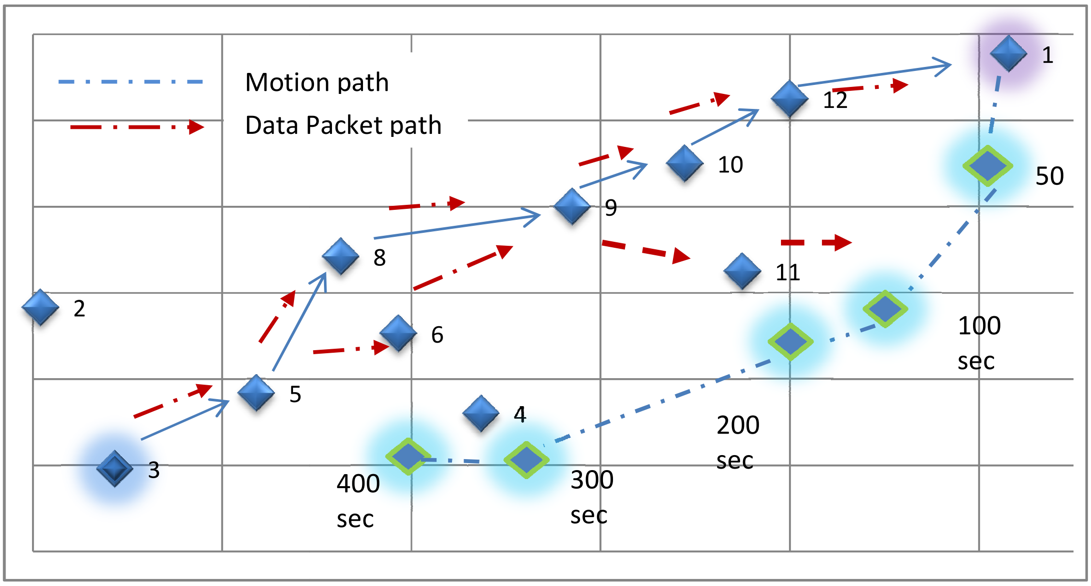


Figure 3.1: A Static Topology Where Node 3 Wants To Contact With Node 1

In this scenario, we are looking to find if the RORP routing algorithm is selecting the appropriate path or not, besides giving a simple illustration about how the algorithm works. And then we compared it with the AODV routing algorithm. After running the simulation (GloMoSim), we find that from time 0 to time 45 seconds, the data that was sent from node 3, was waving between the two paths (3→5→8→9→12→1 and 3→5→6→9→12→1). After that, from 54 till 160 seconds, the path switched from node 12 while node 1 entered the range of node 11. The new path took either 3→5→8→9→11→1 or 3→5→6→9→11→1. While a 170 seconds the path started to be shorter and the data took the path 3→5→8→9→1. After that, at 220 seconds, the path became shorter while node 1 entered the boundary of both node 4 and node 6; and the path since that time took either 3→5→6→1 or 3→5→4→1 until the time became 275 seconds. Due to the movement of node 1, it became under the node 5's range and the new path took is 3→5→1 till the end of the simulation. On the other hand, running the same scenario with AODV gave us just the following: from time 0 till 81 seconds, the data flow took the path 3→5→6→9→12→1. After that, till 275seconds, it took the path 3→5→6→9→11→1. Then, from 275 seconds to the end of the simulation time, it took the path 3→5→1 where node 1 became under the range of node 5 directly as we saw with RORP experiment. Comparing RORP with AODV here, we can notice that with RORP, as soon as the node 1 has changed its place and being under the range of new nodes, the new selected routes become shorter, such as from 170 till 270 seconds. This is because, periodically the receiver sends the RTrack to the sender to inform it about the updated available path. Whereas with AODV, the path was almost the same during the simulation time, but changes just with the movement of node 1 to a new location, which causes losing in the connectivity with the current neighbor. Therefore, a new route discovery is established by the sender. The average end-to-end delay was between RORP and

AODV equal 0.01008 and 0.01087 second respectively. The decreasing in the delay is about 7.84%. Another thing and as we see, RORP sometimes in a period of time was waving the decision between the intermediate nodes, like with nodes (6 and 8), (12 and 11), and also (6 and 4), which indicates that, there is a balancing between the load on the nodes depending on which route the RTrack will be received to the sender through one of them.

3.3 Description of our Proposed Approach RORP

In this section, we go deeply to the RORP routing algorithm elements and description. We describe the packets structure. Then we go through the routing processes by describing the reactive setup processes and after that the proactive process, and discuss the role and conditions where the control packets to be sent or received, and the actions that will be taking by each and every node.

3.3.1 Data Structure in RORP

In this sub-section, first we discuss the structure of the control packets that initiated by both the sender and the receiver. Then, we illustrate the tables that maintained by each node during the session processes between the sender and receiver.

3.3.1.1 Route Request Packet

The Route Request packet (RREQ) is initiated by the sender when there is a data needed to be sent. This packet contains the parameters: the broadcast number, the source address, the source sequence number, the destination address, destination sequence number, and the hop count. The broadcast number is the number of the last request initiated to find the receiver (destination) and it is important to avoid the looping such that the node sends the same packet more than once. The sequence number of the source is used to indicate the latest update for the sender while trying to send to the receiver. The sequence number of the receiver is to let the intermediate nodes to have knowledge about how long the sender known the destination. In another word, to find out if the sender has fresh information about the receiver compared to the information that they may have or they updating what they are holding.

3.3.1.2 Route Reply Packet

The Route Reply packet (RREP) is initiated by the receiver upon the request for route arrived. This packet sent hop-by-hop to the sender as a unicast through the path the RREQ came from. The RREP packet in the RORP routing algorithm contains the parameters: the source address, the destination address, the destination sequence number, and the packet life time. The life time is used so that the intermediate nodes can predict till when this new route will be valid.

3.3.1.3 Route Track Packet

Route Track packet (RTrack) is also initiated by the receiver, but not when the RREQ arrives, it will be sent after small time back as a broadcast toward the sender. This packet contains the parameters like in RREQ packet: the broadcast number, the source address, the source sequence number, the destination address, and the hop count. The parameters here like the ones used with the RREQ, but in reverse mode. Where the receiver here will be the source of the packet and the sender will be the destination. More details about this will be later. (See sub-section 4.3.3).

3.3.1.4 Algorithm Tables

Every node in the network has to maintain different tables. The routing table is holding the information about the destination and how to reach it through that node. It contains the next hop toward this destination beside the sequence number of this destination and the life time for this route to be valid. This life time is frequently updated upon receiving the RTrack from the receiver, or if an error for that node occur like losing the connection with the next hop to the receiver. This table is also playing two roles. The first one is as mentioned above to store the information when a connection from the sender to the receiver is applied, and also the information in a backward for the searching for the sender from the receiver when the latter sends the RTrack to that destination. Unlike AODV, in our RORP proposed algorithm, we did not take the action of deleting information in the routing table in the receiver nodes, because the receiver needs the last sender information periodically to initiate the RTrack control toward the sender. The senders and intermediate nodes will delete, on the other hand, unwanted information when expired.

The neighbors table is just holding the information about the neighbors that control packets came from so can predict the current topology of the network. This information is changing whenever disconnection and failure occurs with this neighbor or a new neighbor is involved in the session and joined the network.

In addition to the previous two tables, there is a table must be maintained by the sender which is the *sent* table. This table is holding the information of the sender this time and the broadcast number of the packet that sent. The packets are either when sending the RREQ from the sender side or the RTrack when initiated by the receiver side. The role that this table is taking in RORP will be discussed more later on. The last table maintained by the nodes is the *seen* table. The *seen* table is created by the nodes to check if the RREQ has been seen before and already got processed by the node or not yet. It contains the RREQ originator (i.e. the sender), or the RTrack originator (i.e. the receiver), and the broadcast number of the packet. This table is necessary to avoid duplicating of the request, therefore avoiding the looping problem.

3.3.2 Reactive Process

The reactive process starts whenever data arrives by an application in the sender side (source: of this data packet) exists to be sent to the receiver (destination: the target where that data must goes to), and there are no routes yet to this destination available to start sending the data. Even if the node knew any route before to the destination, but seems this route is not valid anymore, because of the movement of the sender or one of its next one hop neighbors, or even because the validity of this route to the receiver was known by other sessions but has reached the time to be invalid again. In this case, the sender buffers the data, initiates a RREQ packet, and floods it to all the nodes that belong to its range. The sender will wait until a RREP received from the receiver. The sender also keeps the information about this request in the sent table. This will be kept in the table until a RREP received. If after a certain time no reply noticed where the request still stored in the table, then the sender will re-send a new RREQ packet and update the information in the sent table.

Upon receiving the RREQ by the intermediate nodes, the node will first check whether this packet is the first time to be seen or not. That is done by reading broadcast number of it from the *seen* table. If already this packet has arrived, then it will be ignored and discarded. Whereas, if this packet is a new packet, then it will be saved in the *seen* table by adding the sender address and the packet broadcast number. This information will not stay in the *seen* table too long. After a certain time, this information will be removed. The node will check if there exists any information stored in the routing table about the requested node (receiver). If information was stored from former sessions, the information will be compared with the information that came from the sender. As we mentioned before, the RREQ contains the sequence number of the destination. If compared with the number in the routing table and was newer, then the node will update the information and re-broadcast this RREQ packet to its neighbors. If this was the first time for this packet, then it will be added to the node’s routing table as well and the neighbor information that this packet came from so it will be used if a reply came later targeting the sender. Then the intermediate node will re-broadcast again the packet to other neighbors that come in its range .The other nodes will follow the same procedure with this packet until the request reaches its destination (the receiver).

At the receiver, upon the RREQ arrival, the receiver besides adding the information that came from the sender, it will initiate the route reply packet (RREP) and send it backward to the sender. Other RREQ packets with the same broadcast number will be discarded. This packet, RREP packet, will follow the same path where the request came from hop-by-hop as a unicast message. The intermediate node that sent the request to the receiver will see this reply message, and upon that, the information about the receiver will be set in the routing table and adding the next hop that the reply came from in order to be followed when sending the data later (in this case it will be to the receiver). Now, this route through this next hop to the receiver will be available and ready to be used for forwarding data (i.e. the link is active).

Upon receiving the RREP packet by the sender, the sender will first delete from the sent table, the information stored of the receiver, so that no need to re-send a request again. After that, the sender will start sending the buffered data packets to the receiver.

This reactive process is similar to most of the reactive routing algorithm in setting up the session between the sender and the receiver in order to send data in mobile ad hoc wireless networks.

3.3.3 Proactive Process

After the first connection established, the maintenance process starts to update the routing information also to discover new routes between the sender and the receiver. This process started after the RREP packet sent to the sender. When the receiver initiated the RREP packet, it waited for certain time and started the proactive process steps. All the controls of the maintenance are done by the receiver. The receiver initiates the Route Track Packet (RTrack) to find the available routes to the sender. The packet contains information about the last known sequence number of the sender which it came with the request, and the broadcast number of this packet This will be used to address the target (the sender) and avoid the duplication of sending this packet. Therefore, the algorithm avoids the looping problem. In addition, RTrack contains the receiver address and sequence number. Who will receive this packet will treat the receiver as a sender whom looking for a destination. The broadcast number of the RTrack will be added in the *seen* table, so this packet will be avoided the duplication form the receiver side. Then the receiver broadcast the RTrack to its neighbors in its transmission range. Finally, the receiver sets a timer and waits Ttime seconds to send another route track packet (This Ttime by experiments quantified to be about 4 seconds).

Upon receiving the RTrack by the intermediate nodes, they will check how new the packet is. If this packet is not the first time to arrive, by checking its broadcast number, then it will drop. Otherwise, the intermediate will re-broadcast the RTrack to its neighbors and updates the information in the routing table. Furthermore, it will activate the route to the destination that will be used when the data is transmitted by the sender. The information about the receiver address and the broadcast number will be kept in the *seen* table too.

All the intermediate nodes will treat this RTrack as the sender asking about the receiver. Like the RREQ packet but with some different actions. It actually combined between both the RREQ packet in the way of looking for a destination, and the RREP in just activating the route to the receiver, so it will be used when sending the data. But in the RORP algorithm, the addresses in the RTrack of the sender and receiver are exchanged. Where the intermediate nodes know the receiver as the sender of this packet and the sender here is the destination that the sender of the packet is looking for.

When the RTrack arrives to the sender, the sender will update the information in its routing table about the receiver. One important thing in the routing table, beside the destination sequence number, there is a flag telling that wither or not the RTrack has arrived or not. This flag, is called the track flag, has an important role when the application wants to send data to a destination. The procedure in this case is done by the sender to initiate a RREQ packet, if no route information available yet to the receiver. The sending of the RREQ packet will happen if there is no RTrack has arrived by the receiver. This flag will be set If a problem occurs like broken link with the next neighbor, this flag will be reset and then the sender is able to send a new RREQ packet After setting the track flag, the sender will check if there exist any buffered data waiting to be sent. Finally, the sender will send these data toward the receiver following the path where the RTrack came from.

The receiver will continue updating the sender with the last new route connecting each of them. As we see, the receiver is playing the major role in the proactive process and is responsible here for achieving the maintenance that may needed during the session.

3.3.4 Data Packet Forwarding

The sender will send the data packets either when the RREP arrives or the arrival of the RTrack too. The sender will send the data to the one hop neighbor toward the receiver. Here the sender does not have any information about the nodes that the data will follow; every node already knows who the next neighbor toward the receiver is. Because of the combination of both processes, reactive and proactive, the route between the sender and receiver is updated and explored periodically, and be available when needed.

When the data arrived to an intermediate node, it will check first the validity of the route to the receiver if still accessible. If there was no problem, the intermediate will transmit the data to the already known next hop. This process will continue until the data reached the destination.

3.3.5 Link Failure

The link failure in RORP algorithm is just needed to notify the node, the sender/ receiver or intermediate nodes, that the next hop is not available anymore. At this situation the node will reactivate and delete the information from the routing table. The notification of the failure is announced by the MAC layer. IEEE 802.11 protocol waits from the node that received data from it to acknowledge back. If this acknowledgment did not receive, that means an error to the link occurs and the MAC layer notifies the network layer about this failure. The RORP algorithm does not relay a notification to the other nodes. The node that detect the error will ignore any data came to be sent toward the destination and will wait to hear from the receiver. While the receiver is sending periodic control packets (RTrack), the nodes will reactivate the available routes to the destination again. For that, the routes are always updated or explored for the sender.

In Chapter 4 we compare the RORP algorithm with other routing algorithm using the Global Mobile Information System SIMulator (GloMoSim) [Bajaj et al., 1999], and show the results and comparisons that evaluate our algorithm compared to the other algorithms.

Chapter 4

Result Evaluation and Discussion

In this chapter, we show the results after doing set of simulation experiments and compare these results with some of the well-known MANET routing algorithms. The experiments are focusing on how the RORP routing algorithm deals with the mobility of the nodes and how it can keep the topology of the network overcome with the fast changes that may occur. We show how the RORP algorithm is more efficient with dynamic network topology.

In this chapter, first we describe the parameters and environment of the experiments that will be set in the simulation. After that, we show the various experiments that we compare RORP algorithm with other well-known algorithms in the literature. Finally, we test the RORP algorithm internally by varying the RORP Ttime value.

4.1 Simulation Environment and Setup

The area of the network we used is 2000X2000 m^2. This area is an open area such that the nodes while moving will not consider anything or barrier can stop the nodes from moving. The number of nodes is 50 nodes move randomly inside this area. The movement of a node is according to the Random Way Point mobility model (RWP) [Tavli and Heinzelman, 2006]. The node will start moving from random location and with random speed in straight line targeting a random location. After reaching this new location, the node will stay there in few seconds fixed as pause time seconds. After that, a new random location with random speed, the node will start moving in straight line toward this location. The node will continue this until the end of the simulation time. The simulation time for the experiments is 15 minutes (900 seconds). The speed was choosing in the experiments to be varied between 0 m/s as minimum and 5, 10, 20, or 30 m/s as maximum speed. The pause time also was varied between 5, 10, 100, 200, and 400 seconds. These values will be explained when talking about the different scenarios later on.

Data traffic is organized to be between 40 of the 50 nodes, where we set 20 sessions. The session starts running at the beginning of the simulation and stay running and sending packets during the simulation time of size 512 bytes until the end of the simulation. To have a high traffic load, the time between each packet in each session have to wait about 0.125 second (traffic rate pause time) which means sending 8 packets per second. These packets will be generated by constant bit rate session (CBR) from in the application layer as a real time application. Table 1 summarizes the important simulation parameters.

Other general parameters we used in the experiments are as follow:

- Physical layer: Two-Ray signal propagation model. This approach is used to model the propagation in open space. It assumes that the signal reaches the receiver over two different paths, by direct ray and reflected ray over the ground [Tavli and Heinzelman, 2006]. Where the Radio Frequency band used is 2.4 GHz in 2Mbps bandwidth, and the transmission range 376.782m for each node.
- MAC layer: the MAC protocol used is IEEE 802.11.
- Network layer: Internet Protocol (IP).
- Routing Protocols: RORP, AODV, or WRP protocols.
- Transport layer: we used the UDP protocol.

The used simulator is the Global Mobile Information System SIMulator (GloMoSim), version 2.03. The simulation has been run on laptop, centrino cor 2 duo, 1.83 GHz of CPU, and 1GB of RAM. The used operating system is Windows XP.

Table 4.1: Simulation Parameters

Simulation Time	15 (minutes)
Propagation Pathloss	Two-Ray
Mobile Speed	0~10 m/sec
Mobile Pause-Time	10 sec
Radio Frequency	2.4 GHz
Radio Bandwidth	2 Mbps
Radio Transmission Power	15.0 dBm
Transmission Range	376 meters
MAC Protocol	IEEE 802.11
Network Protocol	IP
RORP Ttime	4 sec

4.2 Performance Metrics

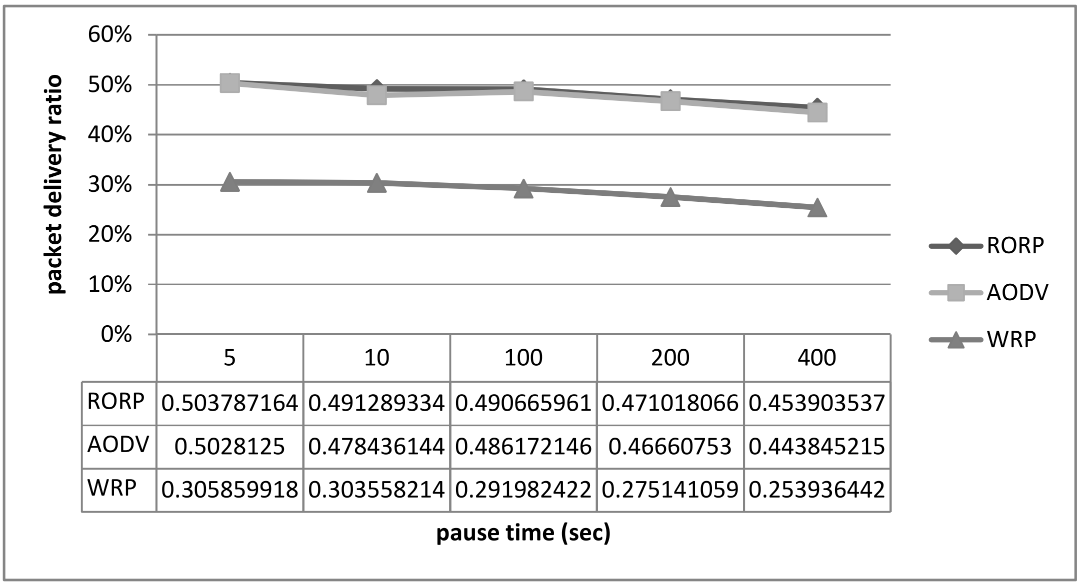
The metrics that we used in this research are as recommended by the IETF MANET in deploying mobile ad hoc routing protocols [Corson et al., 1999]. These metrics, quantitative metrics, can be used to assess the performance of the routing algorithm. The first metric is the packet delivery ratio, computed as the ratio of the total correctly delivered packets, to the total sent packets. This metric is important to measure the performance of the channel due to the changes in the topology. The second metric is the average end-to-end delay. It is the average time from the packet sent by the sender to the arrival of the packet at the receiver. This delay is also including to the queuing time delay of the packet in other nodes. The third metric is the control overhead, which is defined as the total number of control packets issued during the simulation experiment time for the establishment and maintenance of the sessions between the nodes. In addition to these metrics, we included the data throughput which is the total number of bits received over the session time between two nodes. Throughput can be defined also as the average of the successfully delivered data over the network. It is the speed of the data over the communication channel.

4.3 Simulation Experiments and Results

In this section, we show the different scenarios and experiments that are used to compare the RORP routing algorithm to the routing algorithms for MANETs. These scenarios are to obtain the effect on the performance of RORP algorithm. We test the node mobility, by varying the node speed in RWP mobility model and also by changing the pause time of it. This test is very important and it aims to find out how RORP algorithm will perform especially when the topology is in rapid changes. In addition, we test the traffic load by changing the packet transmission rate of the sessions, by making the node sends one packet every 0.065, 0.1, 0.125, 0.25, and 0.5 second, which results an increasing in the traffic (15, 10, 8, 4, and 2 packets/sec). After that, we test the scalability of RORP algorithm under different network size by varying the number of nodes.

4.3.1 Varying the Pause Time.

In this scenario, we change the pause time of the RWP model, from 5 seconds up to 400 seconds and the speed was fixed in every test with 10m/s. The results are depicted in Figures 4.1, 4.2, 4.3 and 4.4 respectively. Here, we discuss the efficiency of the RORP algorithm as compared with AODV, and WRP. The pause time is playing a role in node mobility. Decreasing the pause time will increase the node mobility and as a results a change in the network topology. This cause the node to stay in a location for a few period of time then moves again and so on. The network by then will be very dynamic. On the other hand, by increasing the pause time, the nodes will stay in a location for a longer period of time. That means, the node mobility and the topology will decrease till the increase of the pause time reach to the static networks scenario, and therefore we have a less dynamic networks.



	5	10	100	200	400
RORP	0.503787164	0.491289334	0.490665961	0.471018066	0.453903537
AODV	0.5028125	0.478436144	0.486172146	0.46660753	0.443845215
WRP	0.305859918	0.303558214	0.291982422	0.275141059	0.253936442

Figure 4.1: The Data Packet Delivery Ratio with Different Pause Times

In Figure 4.1, the packet delivery ratio, obtains that RORP algorithm and AODV algorithm at 5 seconds performs the same performance with ratio of 50.3% and 50.2% of the delivered packets respectively. At 10 seconds, a little different in the ratio appears with the RORP sent 49%, where 47% of the packet ratio has been sent by the AODV algorithm with improvement in the RORP performance by 4%. At the end, when the pause time is 400 sec, where the nodes just have the opportunity to move a complete one movement, the RORP still preserve the delivery ratio by 45% and the AODV with 44%. The improvement in performance is 2% of the RORP compared to AODV. RORP algorithm is sending the periodic RTrack every 4 seconds to find the latest link between the sender and receiver as we discussed in the last chapter. Therefore, along with the high traffic transmission rate for sending 8 packets per second, the path is vulnerable to be lost, therefore the sender will send the packets in invalid route until a new update route be known from the RTrack. With AODV, also the same problem occurs due to the high interference of the data packets, the route errors issued by the intermediate nodes, and also the route requests that issued by the senders. Therefore, the data packets might be sent to invalid routes, and then they got dropped. WRP algorithm has the lowest performance in the packet delivery ratio; started with sending 30% at the 5 seconds pause time until it reaches the 400 seconds, with ratio becomes 25%. As average of the different RWP pause times, the improvement of RORP increases nr 1.36% compared to AODV.

As we see, the huge gap between AODV and our RORP algorithm, to WRP algorithm. The improvement in RORP is about 48%. The reason of the low delivery in WRP algorithm is that, with the high mobility, the convergence in sending the updated routing information is lower and delayed due to the changes in the network topology. For that, the packets are sent to a wrong route while the exit information would be stale. The interference that can be occurs between the updated messages and the data packets will affect of reaching these packets to the destination while in addition, a routing loops might occur temporally for losing the update messages. By comparing it with RORP, the improvement of RORP increases by 40.66% on average.

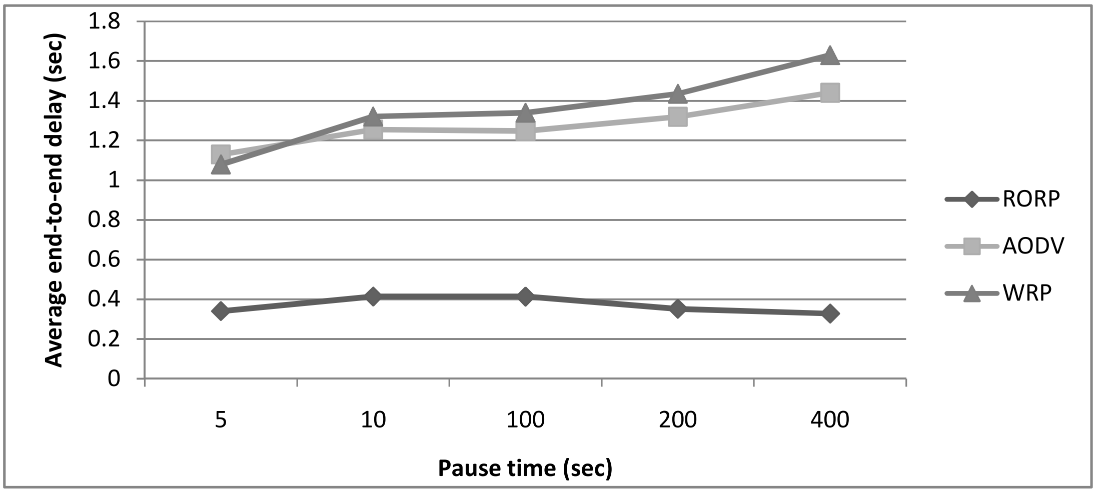


Figure 4.2: Average End-To-End Delay with Different Pause Times

For average end-to-end delay, the RORP algorithm shows the best performance comparing to the other algorithms as depicted in Figure 4.2. RORP algorithm by using the factor of sending the Rtrack message by the receiver periodically, the best available path for the communication between both sender and receiver are guaranteed. In addition, the broadcast of this message via different paths toward the destination, the correct path is the one who will deliver this message to the destination while the others may find a broken link or can encounter longer paths. The improvement in the performance with RORP is by decreasing the delay with 246% and 268.5% compared to AODV and WRP algorithms respectively, on average of the whole experiment results.

The delay in AODV algorithm is caused by the delay in the re-setup of the session between the senders and receivers when an error occurs. In addition, the unicast reply from the receiver is also vulnerable to be lost and not reaching the sender, so that the sender may stay waiting for period of time and resending again the request of a route. WRP algorithm with high mobility is also facing the same problem, where the updated messages for the changes in the network is delayed and could be lost, so the convergence to the changes is facing a significant delay and that caused by the high traffic rate in the communication channel.

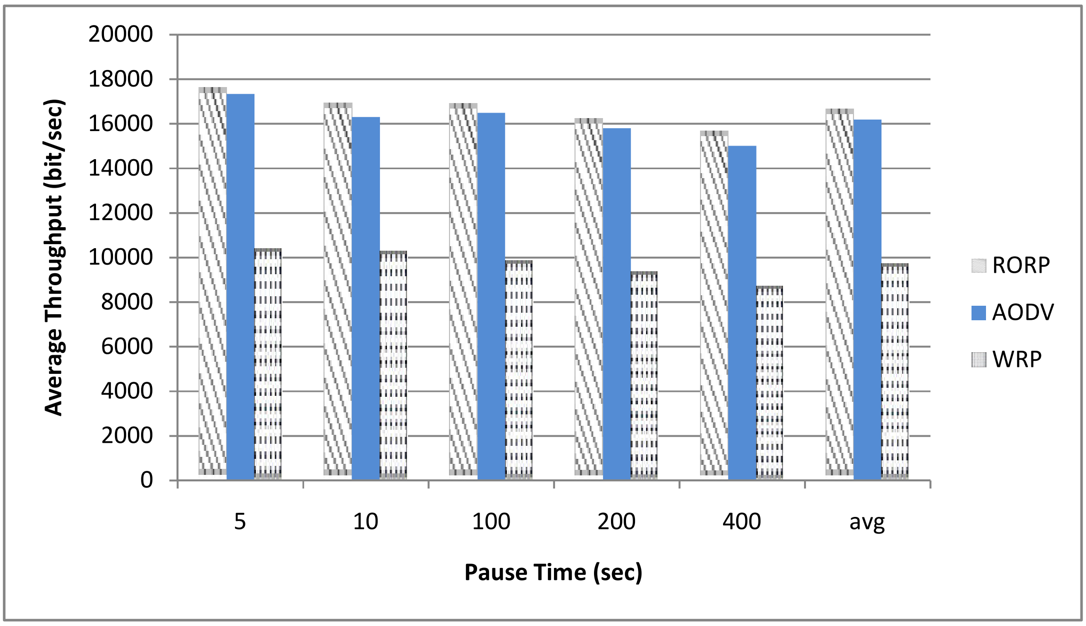


Figure 4.3: Average Throughput with Different Pause Times

For average throughput, RORP algorithm outperformed other algorithms understudy. As we can notice in Figure 4.3, the contest between RORP and AODV algorithm to perform efficiently with mobility changes. At the lowest pause time (highest mobility), the improvement of RORP algorithm is 1.67% increase to the AODV algorithm. While with WRP algorithm, the efficiency is improved by 40.9%. At the highest pause time in this experiment the performance of the RORP improved by 4.3% to the AODV and 44% to the WRP algorithm. The whole average of the throughput shows that RORP improved by 2.9% and 42% when compared to AODV and WRP algorithm respectively.

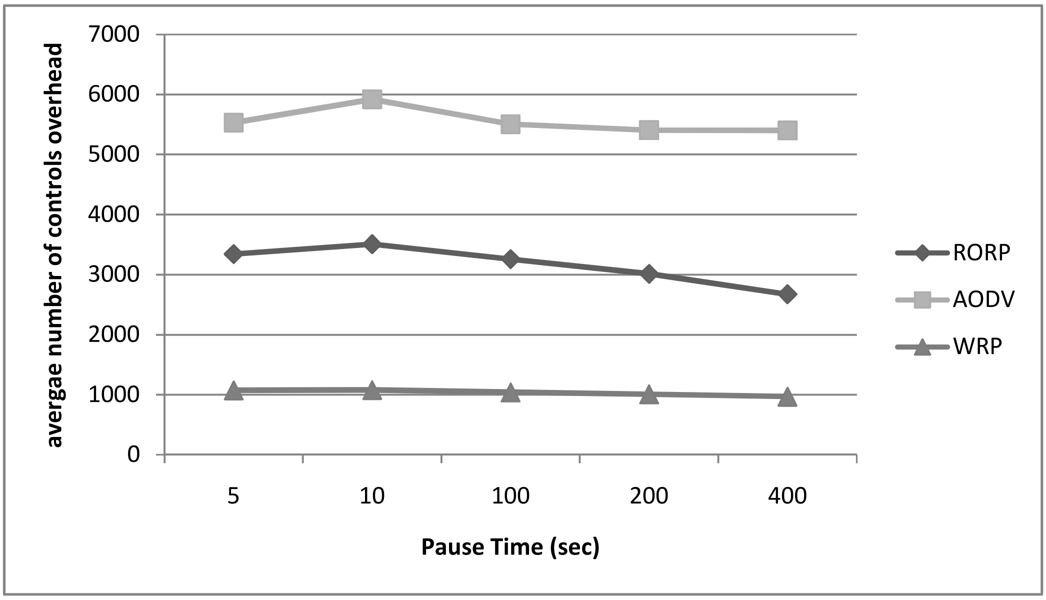
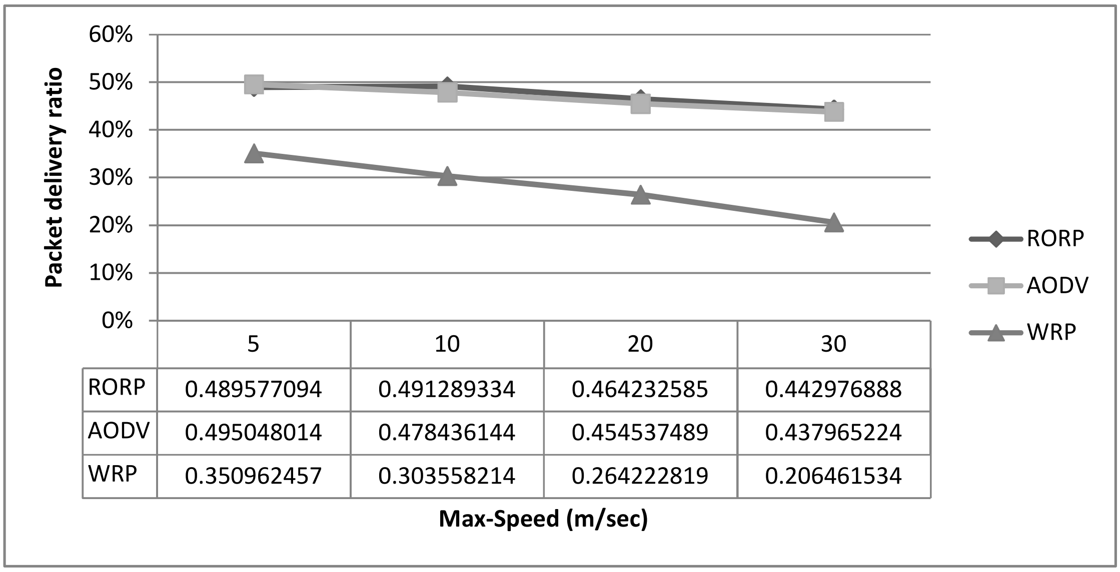


Figure 4.4: Average Number of Controls Overhead with Different Pause Times

Finally, the control overhead of the RORP as shown in Figure 4.4, is better than the AODV algorithm, whereas WRP algorithm outperforms both RORP and AODV algorithms. The control packets for RORP algorithm is affected by the RTrack sent by the receivers periodically as a broadcast. The route request messages are few while no need to re-setup the session in case of a broken and link failure. On contrast, with AODV, the high control overhead is due to the highly packets sent to re-setup the session for the case of link failure notified by the intermediate nodes with the error messages or the movement of the source or the destination. The WRP algorithm is just sending periodically the updated information in case there are new changes or a new node becomes a neighbor, or ACK packets for connectivity purpose of the links. Despite that WRP is a proactive algorithm, the updated packets are sent. Due to the high traffic rate (8 packets per second), these messages may not reach the neighbors even with high mobility where the updates must be propagated. Those neighbors may not be informed and re-propagate these new information. The node that sent these updates will retransmit these updates again after significant time. Hence, the algorithm will encounter a high delay, as we saw previously, and the data might be dropped and not delivered successfully.

4.3.2 Varying the Speed

In this test, we vary the speed of the RWP model. We take the max speeds: 5m/sec, 10m/sec, 20m/sec, and 30m/sec (18Km/H, 36Km/H, 72Km/H, and 108Km/H respectively) where the pause time is 10 seconds to guarantee high mobility. The effect of varying the speed will affect the node mobility. Figure 4.5 to Figure 4.8; show the results of the experiments with this scenario in terms of packet delivery ratio, average end-to-end delay, average throughput, and the overhead in number of controls.



	5	10	20	30
RORP	0.489577094	0.491289334	0.464232585	0.442976888
AODV	0.495048014	0.478436144	0.454537489	0.437965224
WRP	0.350962457	0.303558214	0.264222819	0.206461534

Figure 4.5: The Packet Delivery Ratio after Varying the Speed

For packet delivery ratio (Figure 4.5), the results show that the ratio decreases when the node speed increases with all algorithms. RORP algorithm also gives the good results in the different tests which sent 49% of the data packets and decreases to 44% when the speed reaches its maximum (11% degradation). This is because that, with RORP, the sender nodes are not informed when changes in the network happened like in AODV algorithm where the intermediate node upon finding a link-failure, a RERR packet will inform the senders about an error, so the sender at this case will stop sending through this route. Therefore, the senders will stay sending the data packets even in a wrong way to the receiver. When we compare it with AODV, it shows almost the same performance. At low speed in the scenario (5 m/sec), the AODV sent about 49.5% of data packets. After that, AODV, along with RORP algorithm, its delivery packet ratio decreases and the contest between them at the maximum speed comes for the RORP where the AODV sent 43%. On the other hand, WRP algorithm shows a gap compared to RORP. The improvement of performance of deliver packets compared to WRP algorithm is at the maximum speed 54%.

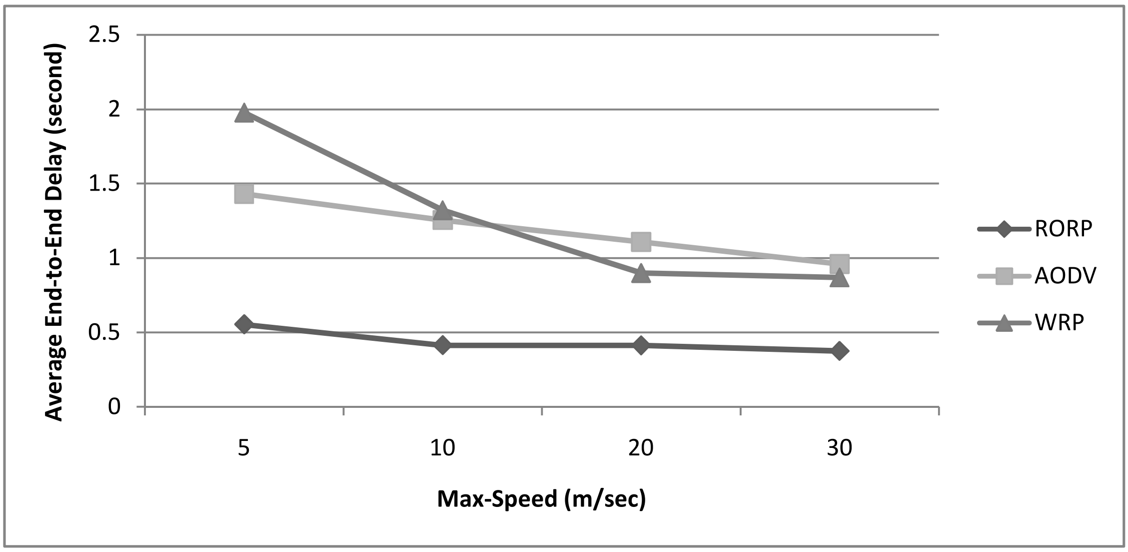


Figure 4.6: Average End-To-End Delay after Varying the Speed

For average delay, Figure 4.6 shows that RORP algorithm is giving again the best efficiency in terms of average end-to-end delay than the AODV and WRP algorithms. The average end-to-end delay of RORP algorithm is more stable where the delay in the different maximum speeds is preserved with close results. At the minimum max speed, the average delay is 0.55 second and decreased until it reaches 0.36 second. Hence, RORP algorithm is showing an improvement when increase the speed. In this case, the improvement is 55%. In contrast, AODV algorithm at the minimum max speed, the average delay is 1.43 second and decreased to reach 0.96 second. The difference between the two delays is 49%. However with WRP algorithm, the different is more which is 126% from 1.97 second decreased to 0.86 second. The efficiency improvement of RORP algorithm as average of the different max speeds compared to AODV and WRP algorithm are 168% and 186% respectively (RORP with 0.438 sec, AODV with 1.188 sec, and WRP with 1.26 sec in the average). The delay becomes less with the increasing in the speed because of the fast convergence for updating the routes. The high speed means the node will reach its location faster than at low speed. Then, while the node in its location (for certain time defined by the RWP pause time), the node can overcome again and explored with the new neighbors, therefore the new route (founded by the different strategies of the studied algorithms) is faster updated.

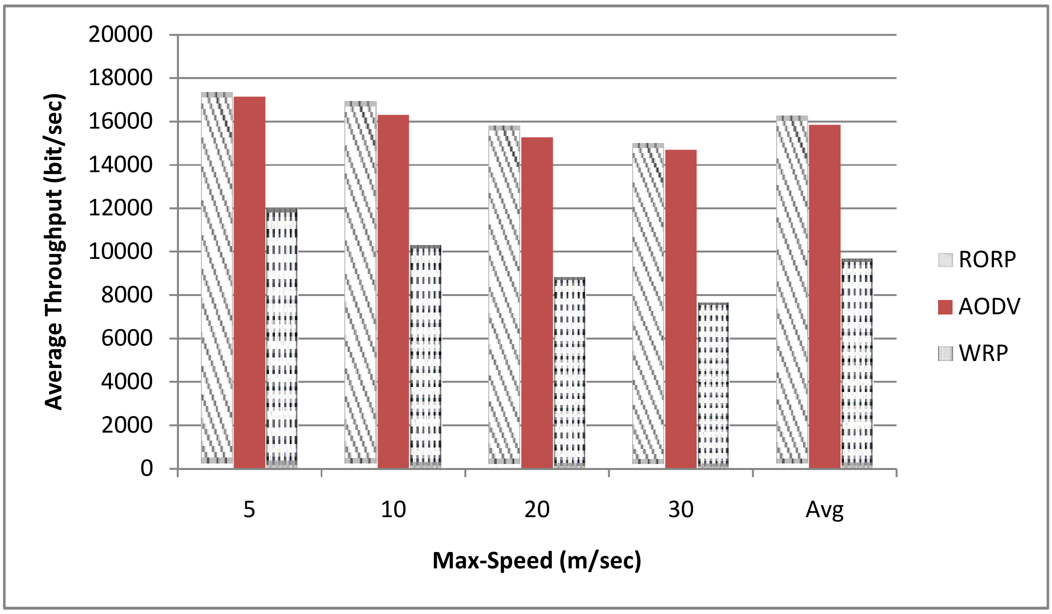


Figure 4.7: Average Throughput in bit/sec after Varying the Speed

For the average throughput, in Figure 4.7, it shows that RORP algorithm still giving the best results than the AODV and WRP algorithms. At the speed of 5 m/sec, the throughput of RORP algorithm 17339.19 bit/sec and the difference between it and the AODV algorithm (with 17147 bit/sec) is 1.108% which is the improvement in the RORP algorithm. With the WRP algorithm (11980 bits/sec), the throughput is low, because of the consumption in the bandwidth. The consumption of bandwidth is due to an increase of the size of the updated information that is passed between the nodes. In addition to that, the delay caused by the propagation of these updates due to mobility is effecting on the throughput. The throughput started from 11980 bit/sec at the low max speed and decreased to 7645.23 bit/sec. As average of the whole throughputs in this scenario, the RORP algorithm (with 16273.55 bit/sec) increased by 2.5% and 40.5% of the AODV (with 15857 bit/sec) and WRP algorithms (with 9688 bit/sec) respectively.

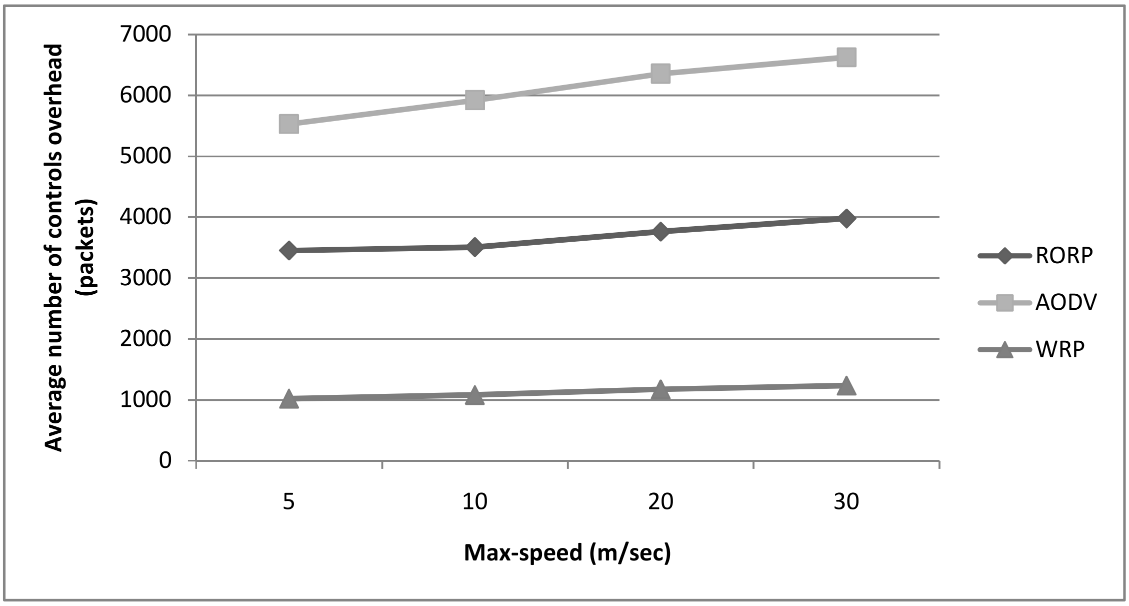


Figure 4.8: Average Number of Control Overhead with Different Node Speeds

Finally, the control overhead in terms of number of packets shows in Figure 4.8 the RORP algorithm comes in the middle between the AODV and the WRP algorithms. We see that WRP algorithm gives the lowest number of average control packets with the increase in mobility; therefore it outperforms RORP and AODV in terms of control overhead. For RORP and AODV algorithms, the average number of control packets increase with the increasing in mobility to the max speed. That is because the fast movement increases the changes in the network topology and losing the connection links. For AODV algorithm, the link failure increases between the nodes and the route error packets, as a result, also increases. That means the re-setup of the routes by the senders will increase with sending the route request packets. These request packets may not be replied and the resending of the packets flooded again. Whereas, RORP algorithm, the same problem of losing the link occurs, but the intermediate nodes will not send the route error packet to the senders and the senders will not react with the failure of the link and waited for the periodic RTracks to become from the receivers. The senders will send the RREQ packets just if there is no RTrack arrived yet, or the previous one has been deleted, because the movement of the sender and the data buffered before a new RTrack activate the link again.

4.3.3 Traffic Transmission Rate Experiments

In this section, we test the effects on the performance under data traffic rate for RORP algorithm. The scenario for this experiment will be as follows: we have 20 sessions; each of them is sending 2, 4, 8, 10, and 15 packets per second of size 512 bytes. That means the transmission throughput will be 8, 16, 32, 40, 61 kb/s respectively in each experiment. In other word, every session will send 1 packet every 0.5, 0.25, 0.125, 0.10 and 0.065 second. By increasing the transmission rate, the network load will be high; therefore, the congestion and interference will increase too.

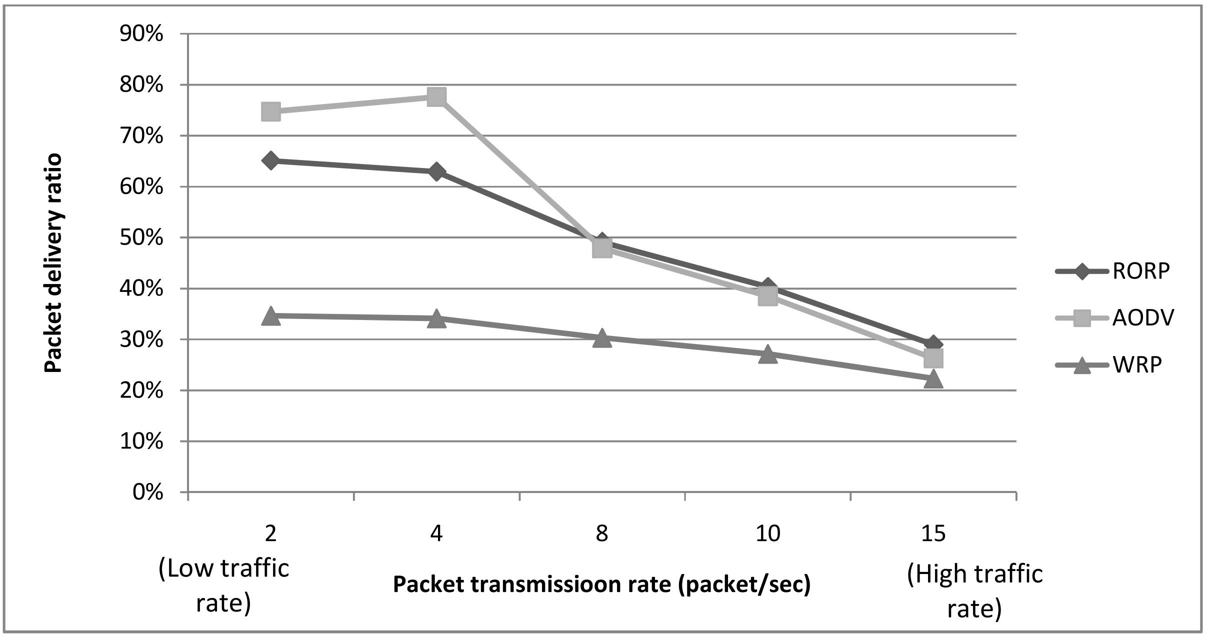


Figure 4.9: Packet Delivery Ratio with Different Packet Transmission Rate

The first metric we study is the delivery ratio as depicted in Figure 4.9. The results show that RORP algorithm gives better performance with high traffic rate, when sending 15 packets per second this ratio increases until the traffic rate becomes 8 packets per second. After that, the AODV algorithm shows a better efficiency in the performance at lower transmission rate. The high drops in the packets at high transmission rate caused by the high interference and channel congestion that lead to packet lost.

The RORP algorithm at low traffic rate successfully delivered 65% of the data packets where at high traffic rate the drops increased and the packets that delivered are about 28%. Whereas the AODV algorithm at low traffic rate successfully delivered 74% and at low rate the ratio decreases to be 26% of delivered packets. The reason of high drops is that RORP algorithm while the convergence is acquired by the periodic RTracks that increases the interference with the high traffic load whereas the increase in drops with AODV algorithm is because the maintenance scheme where the need to re-setup the routes between the sources and destinations acquired by the route error packets and therefore the route request packets that floods the channel with higher interference with the high traffic load. On the other hand, we see that at low traffic rate the periodic packets sent by the receivers (RTrack), make interference and high traffic rate will result in an increase in the packet drops. In AODV algorithm, the need to re-setup the routes decreased and the ratio of packet delivery become higher. While for WRP algorithm, it is less affected by the changes in the traffic rate, but it gives the worst efficiency in terms of delivery ratio. For low traffic load, the ratio of successfully delivered packets is 34% and at high traffic rate, it becomes less by 22%. In WRP algorithm due to the network changes, drop packets increases due to high communication between neighbor nodes, plus the ACK messages that required for link connectivity, and as a result, high interference. For that, the nodes are holding stale information about the network and the packet can move in a wrong direction while the sender thoughts the path to the receiver is correct.

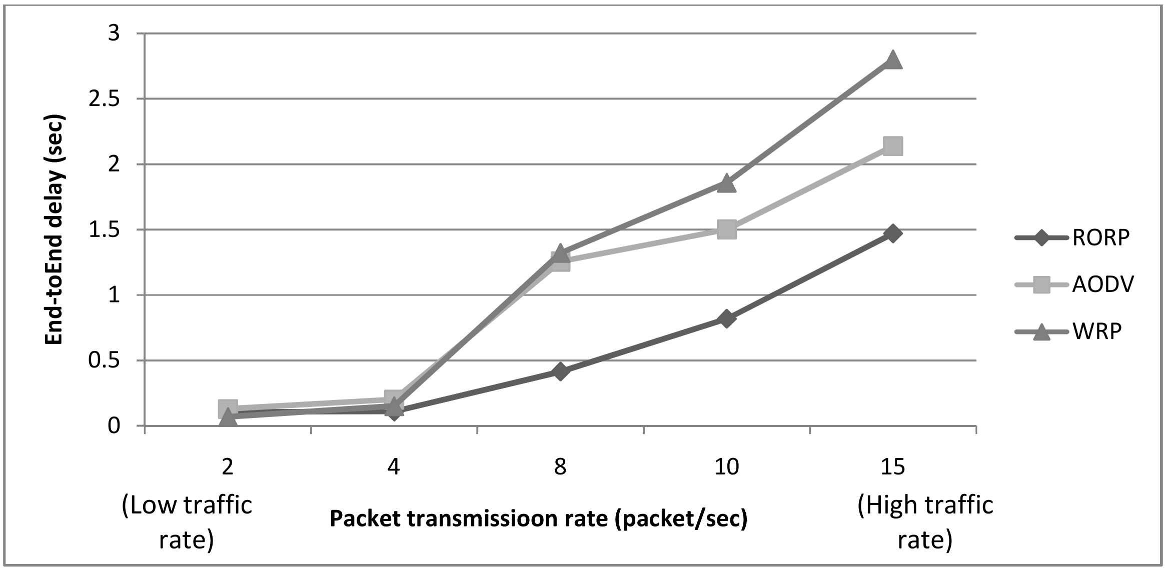


Figure 4.10: Average End-To-End Delay with Different Packet Transmission Rate

For average end-to-end delay, RORP algorithm shows better performance compared to other algorithm as shown in Figure 4.10. The performance efficiency increases as the traffic rate increases. And like in the delivery ratio, we notice that WRP at high traffic rate gives the worst results. The high interference stops the nodes to have clear information about the shortest paths; as a result, the data drops increase and cause further delay. With the decreasing in the traffic rate, the WRP algorithm converges fast at rate of 8 packets per second. And for 4 packets per second, it shows a stable delay performance along with the RORP and AODV algorithms too. The AODV algorithm also is affected in the average end-to-end delay because of the high traffic rate. The effect of low delivery ratio also effects the increasing in the delay. And like the WRP algorithm, at traffic rate of 8 packets per second, we see the fast decreases in the delay from 1.25 seconds to be 0.2 second at rate of 4 packets per second. The RORP algorithm gives the best results and this efficiency appears more when the traffic rate is high. At high traffic rates the delay is 1.4 second (2.136 and 2.8 seconds with AODV and WRP algorithms), where the difference of the RORP to the AODV and WRP algorithm is less by 52% and 100% respectively. Then the delay decreases to converge with other algorithms at rate of 4 packets per second with a delay of 0.108 second (0.2 and 0.15 second for the AODV and WRP algorithms respectively).

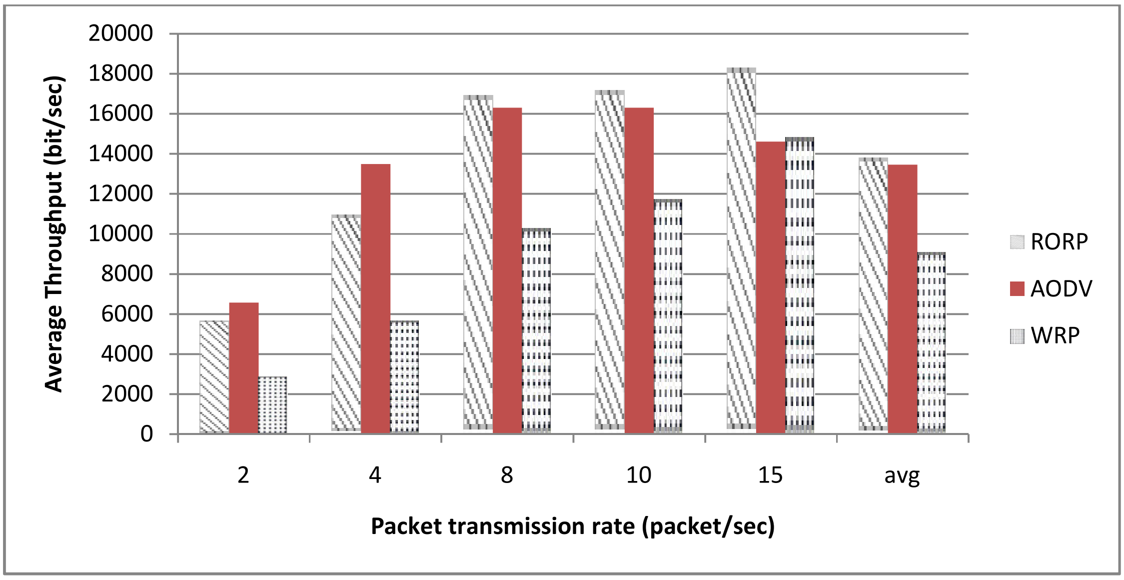


Figure 4.11: Average Throughput with Different Packet Transmission Rate

For throughput (Figure 4.11), RORP shows the high throughput efficiency at high traffic rate until reaches the rate of 4 packets per second, where it becomes lower than the AODV. The WRP algorithm shows the lowest throughput in the 5 experiments. At high traffic rate, the improvement of the RORP compared to AODV and WRP algorithms is 20% and 19% respectively. But, at rate of 4 packets per second the performance decreased compared to AODV by 23%, but it still perform better than the WRP algorithm. The whole average throughput of the 5 experiments, as shown in the Figure 4.11, shows that the RORP algorithm is higher than the AODV and WRP algorithms with an improvement of 2.5% and 34%.

Finally the overhead, as depicted in Figure 4.12, shows that RORP gives good results too in terms of number of control packets. At high traffic rate, the periodic control packets (RTracks) that sent to the senders are less than the control packets that are issued by the AODV.

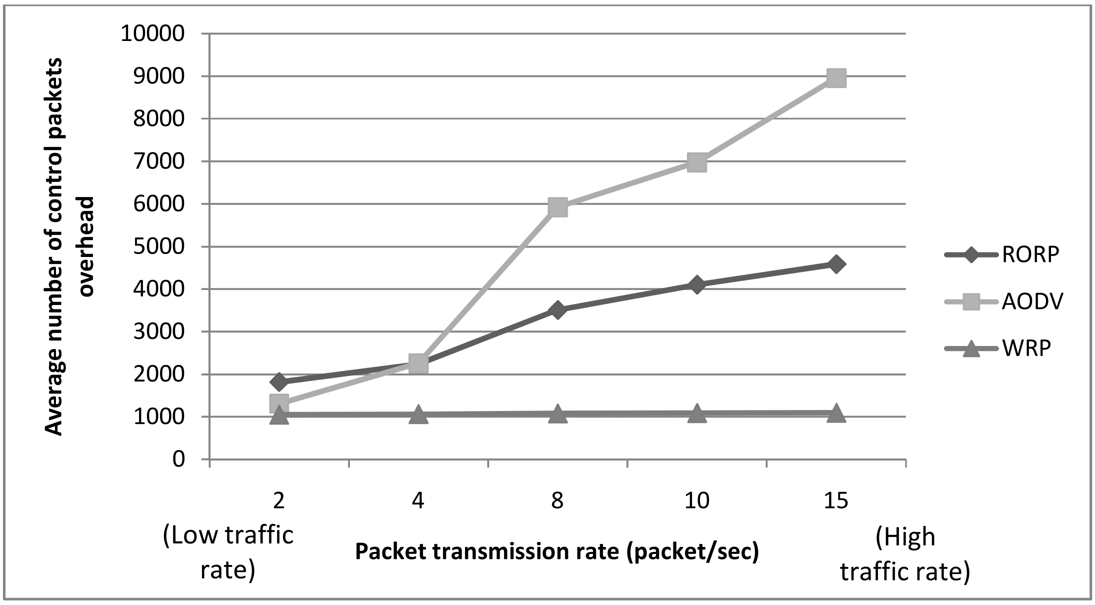
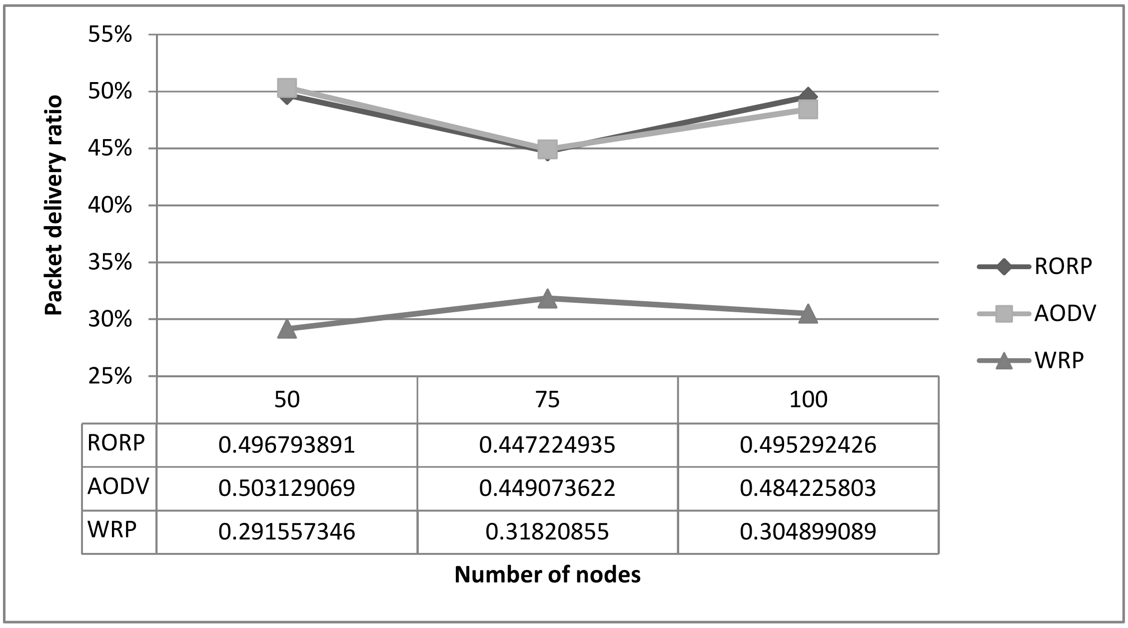


Figure 4.12: Average Number of Control Overhead with Different Packet Transmission Rate

As we see, the number of control packets in RORP is half the number of control packets in AODV. The AODV as the rate increases, the needs to re-setup the route increases too beside the increase in the packet drops. Also, the RREQ packets may be lost and never reach their destinations due to the high traffics, that results in high interference, therefore, increase in packet drops. On the other hand, control overhead is less with the RORP algorithm, where the periodic RTrack is issued by the receivers and the need to maintain the route is just relies on it. Beside that, small number of RREQ controls is issued. These controls increases with the increase in traffic rate due to the increase in the interference. WRP outperforms the other algorithms which issued the least number of control packets during the different rates due to the fact that the nodes will issue these controls periodically despite how the traffic rate is, however the delay and delivery ratio are less when compared to RORP and AODV algorithms, as discussed earlier.

4.3.4 Varying the Number of Nodes

In this experiment, we aim to test the scalability of the RORP algorithm. We vary the number of nodes by 50, 75, and 100 nodes and increase the network area size to 2500x2500 m^2 to make the node density constant during the movement of the nodes that concentrate at the center. The nodes less than 50 nodes may result a less connectivity due to the random placement and movement of these nodes in the area. From the experiments, with 50 nodes the 20 sessions between the nodes are done correctly, whereas, with less number of nodes most the sessions never occur. The results with 50 nodes show an acceptable connectivity between the nodes and this connectivity with the increase in nodes start to be lower as states in the following results.



	50	75	100
RORP	0.496793891	0.447224935	0.495292426
AODV	0.503129069	0.449073622	0.484225803
WRP	0.291557346	0.31820855	0.304899089

Figure 4.13: Packet Delivery Ratio with Different Number of Nodes

When studying the results for the delivery ratio, we see that RORP at high number of nodes is able to deliver successfully more packets than AODV and WRP. The difference between RORP and AODV is slightly small but both are better in performance in terms of delivery packet ratio than WRP, as shown in Figure 4.13. The delivery ratio in the three algorithms is low in the three different tests. The RORP delivered 49.6% of packets for the 50 nodes where the AODV has sent 50% of the packets. WRP shows the worst performance with just 29% of the packets successfully sent. RORP and AODV give the same results also when the number of nodes got increased, but the delivery ratios become lower with 44% for each. For the 100 nodes, the performance of both RORP and AODV still low with ratio of delivered packets reached the 49% and 48% respectively. At high node density, the improvement of RORP increases by 2.23% and 38.44% compared to AODV and WRP algorithms respectively. The delivery ratio with 75 nodes shows that, the connectivity becomes less due to the RWP random locations for the nodes; this connectivity becomes stronger with the increasing in the nodes while the movement becomes more strict and concentrated in the center, as a result less in losing the link between the nodes.

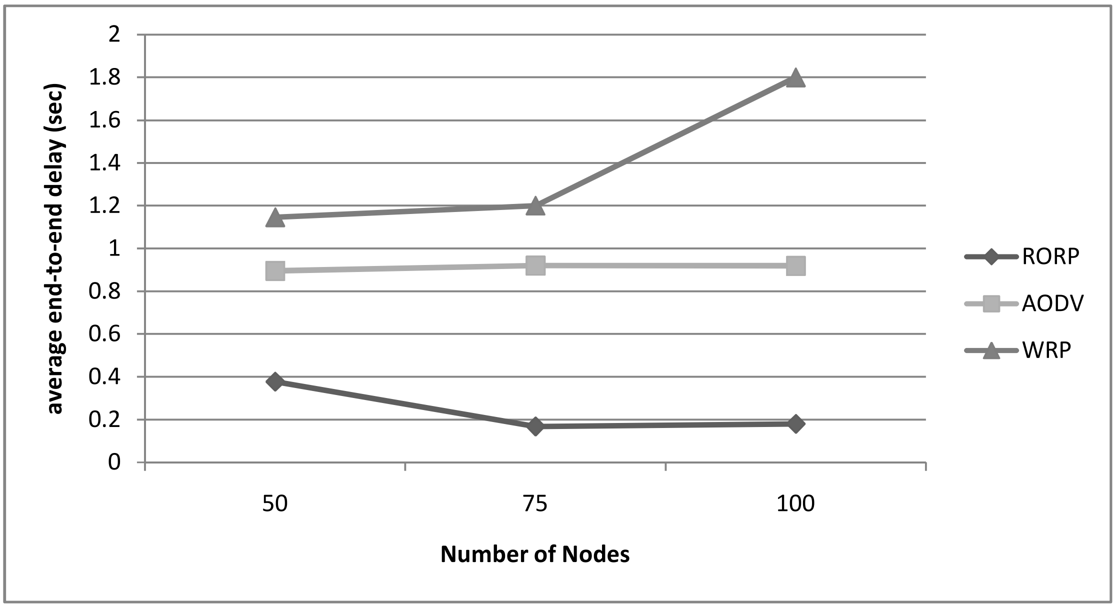
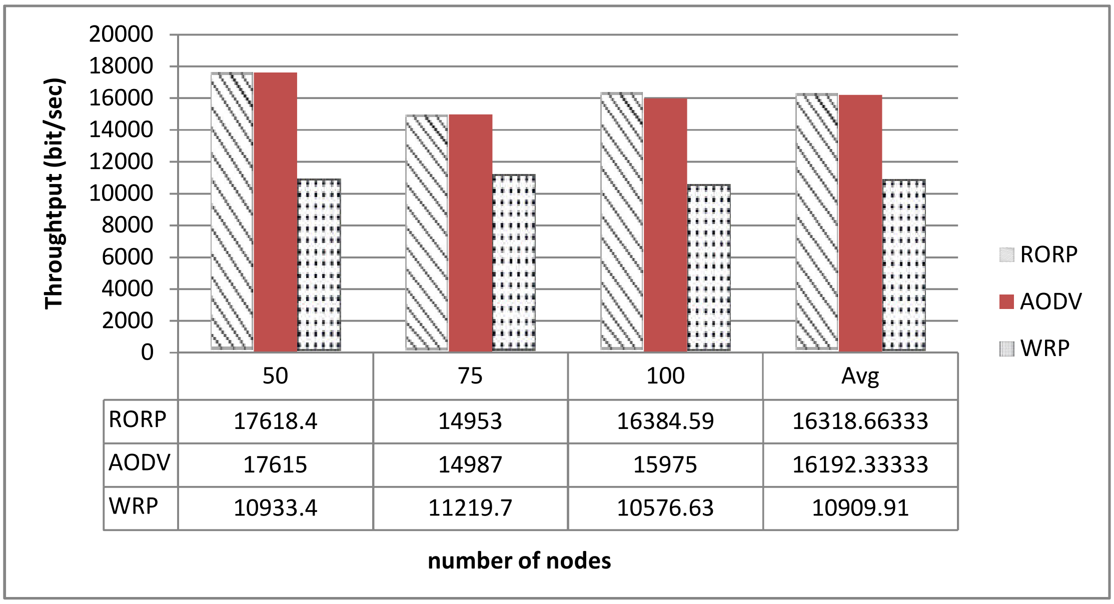


Figure 4.14: Average End-To-End Delay with Different Number of Nodes

For average end-to-end delay, the results show that, RORP is more scalable in terms of average end-to-end delay as shown in Figure 4.14. RORP gives the best performance compared to AODV and WRP. With RORP, The delay decreased as we increase the number of nodes where the connectivity becomes stronger and the best path that acquired due to the maintenance process. The efficiency becomes less than AODV by 137% (0.37 second delay with RORP, 0.89 second with AODV) at low node density of 50 nodes. However, WRP shows at that stage the worst performance, where the delay is 1.15 seconds.

With the increasing in the number of nodes, we find that RORP performance gets better and decreased to be 0.18 second at high nodes density with 100 nodes. The performance increased by 109% (from 0.37 to 0.18 second). The performance of the AODV is stable and the delay did not affected by the increasing in the number of nodes. Where, when the number of nodes is 100, the delay becomes 0.92 second, a bit more than the delay at low node density. WRP encountered an increasing in the delay, when we increase the number of nodes above the 70 nodes. The difference between the delays at 100 nodes to that at 50 nodes is 36% (from 1.14 to reach 1.8 seconds). Comparing RORP with AODV and WRP, the RORP outperforms them with improvement by 410% and 900% respectively.



	50	75	100	Avg
RORP	17618.4	14953	16384.59	16318.66333
AODV	17615	14987	15975	16192.33333
WRP	10933.4	11219.7	10576.63	10909.91

Figure 4.15: Average Throughput with Different the Number of Nodes

Studying the throughput on the other hand, shows similar results with respect to the delivery ratio. The RORP and AODV give similar results as shown in Figure 4.15. WRP is less by 61% of the throughput to the RORP and AODV. As average throughput of the three tests, RORP shows a non notable improvement to the AODV by just 0.77%, where compared to WRP, the improvement is better by 33% on average. Where the good results of the RORP is shown at high node density of 100 nodes where the improvement compared to AODV and WRP algorithm increases by 2.5% and 35.44% respectively.

Finally, the number of control overhead results show that between RORP and AODV a huge gap increase with the increasing in the number of nodes as shown in Figure 4.16. The increasing in the number of nodes results in an increase in the number of control packets in AODV, because of the maintenance and repairing processes. Whereas with RORP, it just needs, periodic control packets to maintain the routes. RORP did not get affected by increasing the number of nodes. The RORP is scalable in terms of control overhead efficiency. Besides that, the number of controls is less than the AODV by 93%, 170% and 190% for the 50, 75, and 100 nodes respectively. This indicates that, AODV is less scalable than RORP and also WRP. WRP shows the best results and the number of controls increases as we increase the number of nodes. The number of controls at high node density (100 nodes) increased by 30% compared to the number of controls at lower node density (50 nodes).

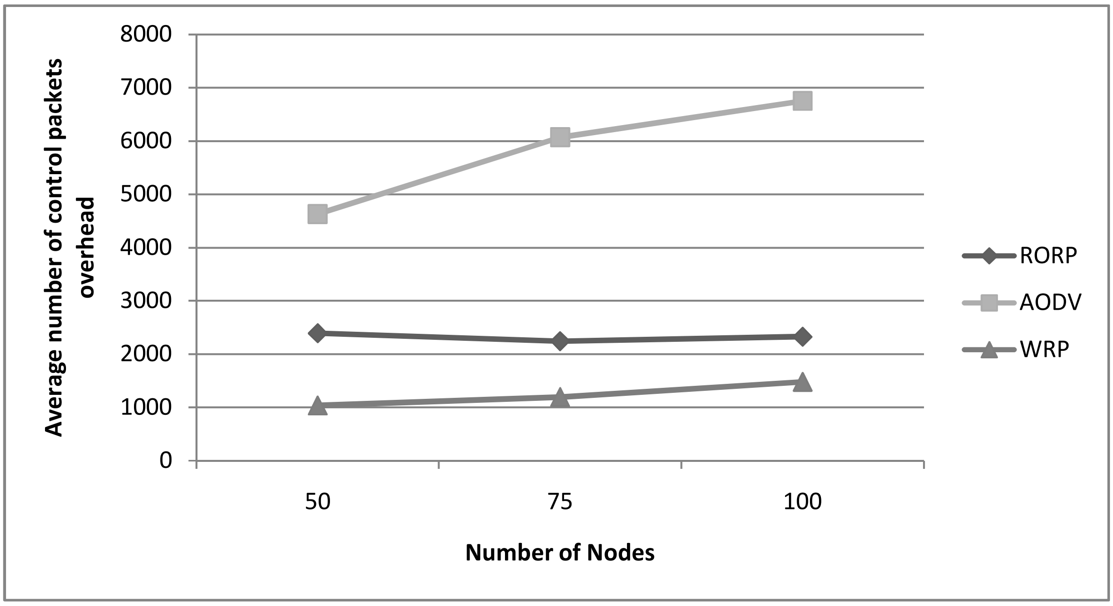


Figure 4.16: Average Number of Control Overhead with Different Number of Nodes

4.4 Internal Experiment Test of The RORP Algorithm

In this section, we do an internal test for the RORP algorithm, where we evaluate the algorithm by varying the RORP Ttime in seconds, which is the time between each RTrack sent by a receiver. In the previous simulation experiments, we have chosen the RORP Ttime to be 4 seconds. In this experiment, we illustrate how the effect of varying RORP Ttime will be on the performance of RORP algorithm. We do the test by varying the max-speeds: 10, 20, and 30 m/sec. the RORP Ttime values will be: 2, 4, and 6 seconds. After repeating the simulation experiments, we have the following results.

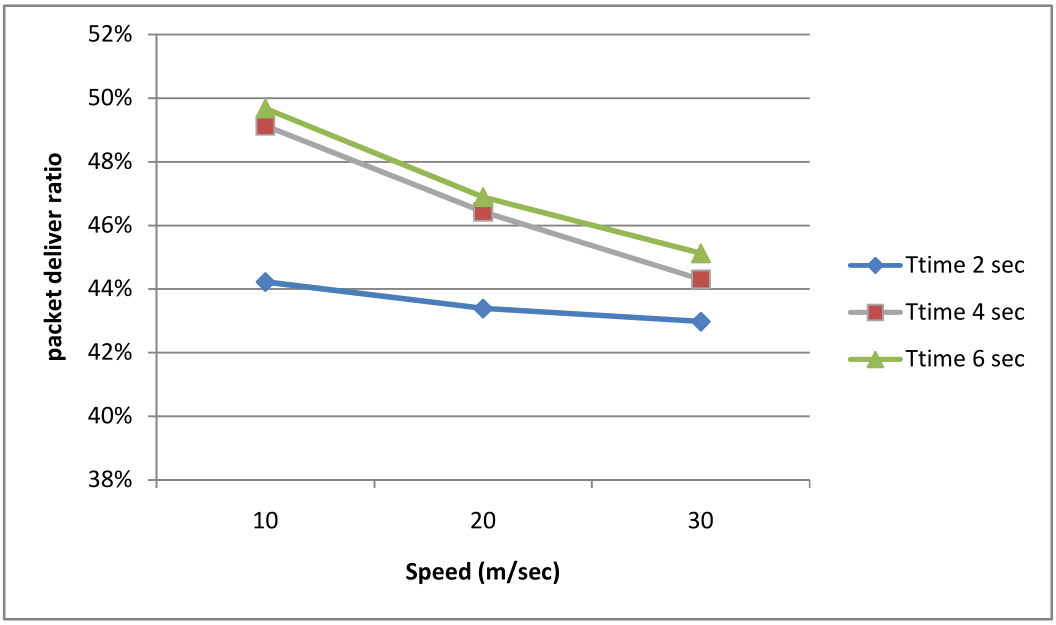


Figure 4.17: The Packet Delivery Ratio with Different RORP Ttimes

For delivery ratio, in Figure 4.17, it is obviously noticed that at 2 seconds of Ttime, the delivery ratio is the worst in performance. The overhead of the RTrack control becomes high, which results an increase in the collision and hence, increases the packet drops. As we can notice that at RORP Ttime 4 and 6 seconds intervals, the performance is converged and gives better results compared to the former Ttime.

For delay, in Figure 4.18, we see that at 2 seconds interval, the RORP is giving high delay compared to the results with 4 and 6 seconds of Ttime interval. The interference that occurs as a result of the high overhead, affects on the performance of the algorithm in terms of delay. We can notice that with the different max-speed tests, and RORP Ttime is 4 seconds, the performance starts to converged and gives better results. These results are kept steady with Ttime equal 6 seconds.

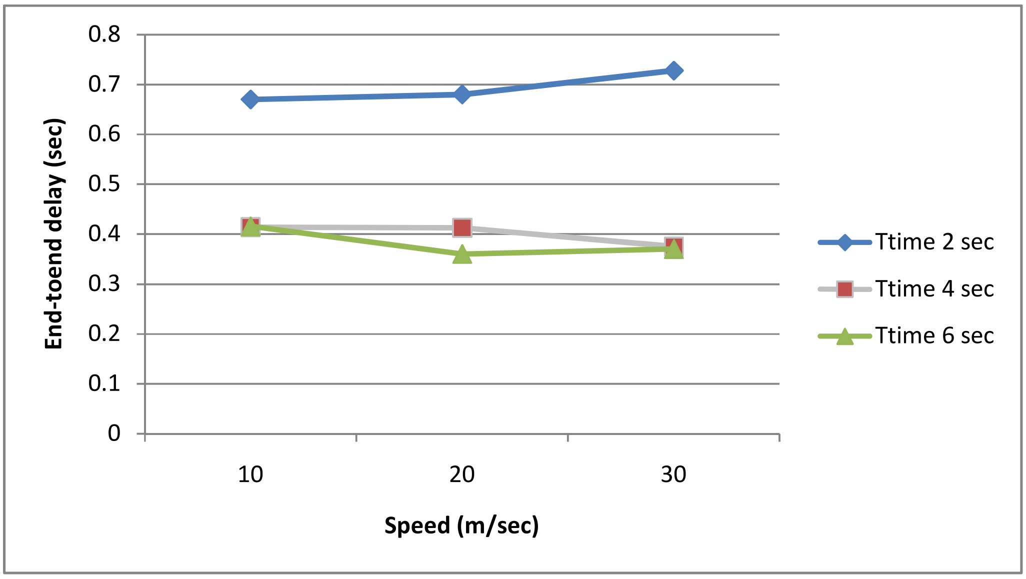


Figure 4.18: Average End-To-End Delay after Varying the RORP Ttime

The control overhead as a result (in Figure 4.19), is high with RORP Time equal 2 seconds. That is because after the proactive process in the RORP algorithm starts, every 2 seconds the receivers will flood the RTrack which increases the overhead in the network. When the RORP Ttime equal 4 seconds, it is obliviously noticed that the RORP algorithm is starting to give better efficiency for different speed. In our simulation scenarios, we chose the RORP Ttime to be 4 seconds in order to achieve better results as we have seen in this section.

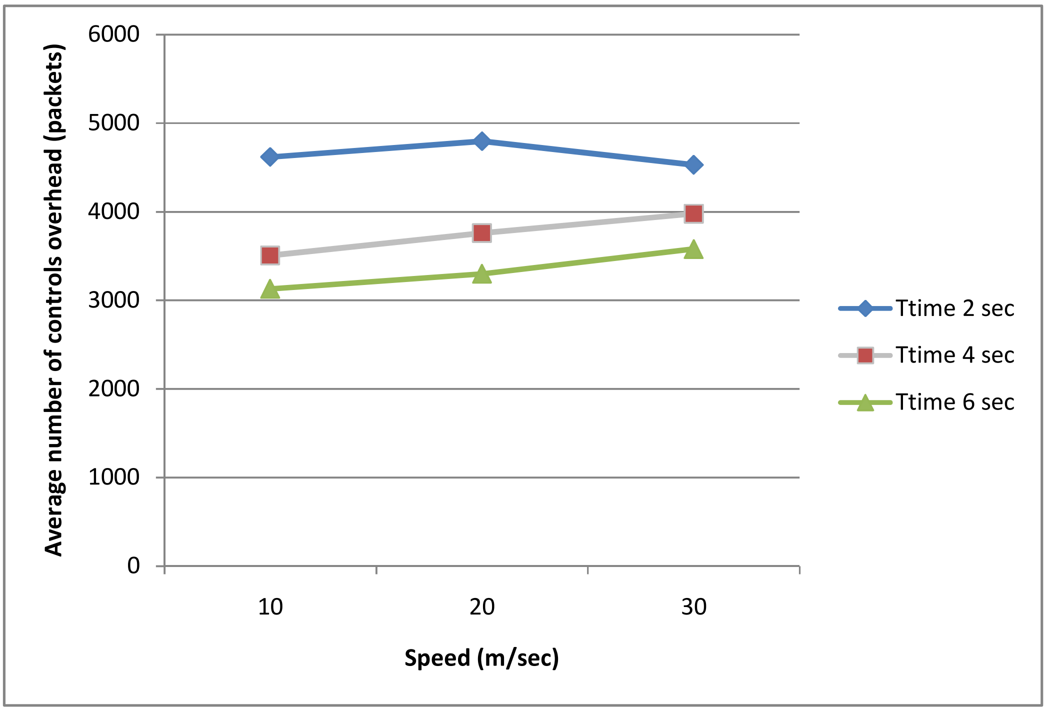


Figure 4.19: Control Overhead After Varying the RORP Time

Chapter 5

Conclusions and Future Research Works

In this thesis, we have proposed and implement a new routing protocol for MANETs called Receiver-Oriented Routing Protocol (RORP). We have addressed the problems that faced when designing a routing algorithm for MANETs. After that, we have discussed the metrics that are needed to evaluate routing algorithms in MANETs and introduced some routing protocols. Then, we have described the RORP algorithm as a new hybrid routing algorithm; we implemented the RORP algorithm, and compared it with some other known routing algorithms by exhaustive simulation scenarios, and finally, we discussed the results. In this chapter, we conclude the thesis and then discuss possible future research works.

5.1 Conclusion

In this thesis, we have developed RORP, a new routing algorithm for mobile ad hoc wireless networks. RORP is a hybrid algorithm where a reactive route setup process is used when demanded at the beginning of each new communication between two different nodes. This process aims to find a path for the data. The proactive routing process then aims to maintain this path throughout the duration of the communication to preserve the path and keep it updated and also find new better paths. This maintenance is done by sending a control packet from the destination (receiver) periodically toward the source (sender).

Our proposed algorithm, RORP, has been evaluated and tested in a free space with various experiments. The results show a significant improvement in the performance of the routing protocols in MANETs. We have compared the results with Ad hoc On-demand Distance Vector

(AODV) and Wireless Routing Protocol (WRP) routing algorithms. For high node mobility, we did two different experiments by varying the pause time and varying the speed. For the former test, RORP shows a notable improvement in the average end-to-end delay where it has decreased by 246% and 268% on average of the whole different pauses tested compared to the AODV and WRP algorithm respectively. On the other hand, the delivery ratio performance improvement is increased by 4% compared to the AODV and by 40.66% compared to WRP. The control overhead shows different results. For control overhead, the performance has increased compared to the AODV, but the WRP was less than both RORP and AODV algorithms. The decreasing of the RORP in overhead was about half compared to the AODV routing algorithm. These improvement results are shown in the Appendix A. For throughput, RORP outperforms AODV and WRP algorithms by 2.9% and 41% respectively on average.

The performance of the RORP also has been tested under different number of nodes. RORP shows a significant improvement in the average end-to-end-delay. At high node density, the RORP decreased the delay by 410% and 900% compared to AODV and WRP algorithms. That shows that RORP is more scalable in terms of delay. The delivery ratio and throughput results at high node density in the experiment shows an improvement in RORP about 2.23% and 38.44% for delivery ratio, and 2.5% and 35.44% for average throughput respectively. The control overhead on the other hand, decreased by 190% compared to AODV, but was higher than WRP by 36.36% at high node density in the experiment. For that, RORP algorithm is scalable routing algorithm.

Finally, we have tested the RORP algorithm by varying the traffic rate where the interval between a data packet and the next one was varied. At highest traffic rate in the experiment, RORP shows better performance in terms of packet delivery ratio, where the improvement increased by 9.3% and 22.9% compared to AODV and WRP algorithms respectively. The average end-to-end delay in addition, is decreased by 45% and 90%. The control overhead like the other experiments shows better results compared to the AODV by decreasing the overhead by approximately half, with 95% less. However, WRP algorithm is less than RORP in terms of overhead where RORP is higher by 76%. More results are shown in Appendix A.

5.2 Recommended Future Researches

RORP algorithm is proposed for multihop unicast routing algorithm for MANETs. This algorithm can be studied to be improved to work with multipath routing algorithms. We have build this algorithm using the GloMoSim simulation tool and tested it with some of the routing algorithms that supported by it such as AODV and WRP. Building this algorithm using different simulators in order to test it with other routing algorithms is also an important task in order to evaluate RORP algorithm more. Also, increase the number of scenarios and environments such varying the number of sessions between nodes and varying the data packet size.

REFERENCES

Ahvar, E., and Fathy, M. (2007). **Performance Evaluation of Routing protocols for High Density Ad Hoc Networks based on QoS by GloMoSim Simulator**. Proceeding of World academy of Science. Engineering and Technology.

Ashwini, K., and Fujinoki, H. (2005). **Study of MANET Routing Protocols by GloMoSim Simulator**. International Journal of Network Management .

Bajaj, L., Takai, M., Ahuja, R., Tagk, Bagorodia, R., and Gerla, M. (1999). **GloMoSim: A Scalable Network Simulation Environment**. University of California.

Bettstetter, C., Resta, G., and Santi, P. (2003). **The node distribution of the random waypoint mobility model for wireless ad hoc network**. IEEE Transactions on Mobile Computing, (pp. 257-269).

Bharghavan, V., Demers, A., Shenker, S., Zhang, L. (1994). **MACAW: a media access protocol for wireless LAN's**. In Proceedings of the ACM Symposium on Communications Architectures and Protocols (SIGCOMM).

Carson, S., and Macker, J. (1999). **Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations**. The Internet Engineering Task Force (IETF), rfc 2501.

Clausen, T., Jacket, P., and Viennot, L. (2002). **Comparative study of Routing Protocols for Mobile Ad Hoc Networks**. The First Annual Mediterranean Ad Hoc Networking Workshop .

Clauses, T., and Jacquet, P. (2003). **Optimized Link-State Routing Protocol (OLSR)**. Retrieved April 2008, from IETF Network Working Group, Experimental RFC 3626: www.ietf.org/rfc/rfc3626.txt.

Conti, M. (2003). **Body, Personal, and Local Ad Hoc Wireless Networks**. In M. Ilyas, The Handbook of Ad hoc Wireless Networks (pp. 13 - 34). CRC Press.

Das, S. R., Castaneda, R., Yan, J., and Sengupta, R., a (1998). **Comparative Performance Evaluationof Routing Protocols for Mobile Ad hoc Networks**. Proceedings of International Conference On Computer communications and Networks (pp. 153-161). ICCCN 1998.

Das, S. R., Perkins, C. E., and Royer, E., b (2000). **Performance comparison of Two On-demand Routing Protocols for Mobile Ad hoc Networks**. INFOCOM2000 .

Di Caro, G., Ducatelle, F., and Gambardela, L. M. (2005). **AntHocNet: an Adaptive Nature-inspired Algorithm for Routing in Mobile Ad Hoc Networks**. European Transaction on Telecommunications (Special Issue on Self-Organization in Mobile Networking) .

Dorigo, M., and Stützle, T. (2004). **Ant Colony Optimization**. Cambridge: MIT Press.

Haas, Z. J., and Pearlman, M. R. a (2001). **The Performance of Query Control Schemes for the Zone Routing Protocol**. ACM/IEEE Transactions on Networking, (pp. 427-438).

Haas, Z. J., and Pearlman, M. R. b (1999). **The Zone Routing Protocol (ZRP) for Ad Hoc Networks**. Internet Draft, IETF Network Working Group.

IEEE Standard 802.11 (2007). **Part11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications**, IEEE Computer Society, -(Revision of IEEE Std 802.11-1999).

IEEE 802.11 LAN/MAN Standards Committee (2009), official website, available from: http://ieee802.org/11/ .

IEEE 802.15 LAN/MAN Standards Committee (2009), official website, available from: http://ieee802.org/15/ .

IEEE 802.16 LAN/MAN Standards Committee (2009), official website, available from: http://ieee802.org/16/ .

Jayakumar, G., and Gopinath, G. (2007). **Ad hoc Mobile Wireless Networks Routing Protocols- A review**. Journal of Computer Science, Science Publication .

Johnson, D., Hu, Y., and Maltz, D. (2007). **The Dynamic Source Routing Protocol (DSR) for Mobile Ad hoc Networks for IPv4**. Retrieved April 2008, from Internet Engineering task force (IETF), Experimntal rfc 4728: www.ietf.org/rfc/rfc4728.txt

Kim, C., Talipov, E., and Ahn, B. (2006). **A Reverse AODV Routing Protocol in Ad Hoc Mobile Networks**. IFIP International Federation for Information Processing. EUC Workshops.

Lin, T. (2004). **Mobile Ad-hoc Networks Routing Protocols: Methodologies and Applications**. Virginia Polytechnic Institute and State University.

Lin, X. H., Kwong, Y. K., and Lau, V. K. (2002). **RICA: A Receiver-Initiated Approach for Channel-Adaptive On-Demand Routing in Ad Hoc Mobile Computing Networks**. Proceedings of the 22nd IEEE International Conference on Distributed Computing Systems.

Mbarushimana, C., and Shahrabi, A. (2007). **Comparative Study of Reactive and Proactive Routing Protocols Performance in Mobile Ad Hoc Networks**. Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops, (pp. 679-684).

Misra, S., Woungang, I., and Misra, S. C. (2009). **Guide to Wireless Ad hoc Networks**. The Computer Communications and Networks. Springer.

Murthy, S., and Garcia-Luna-Aceves, J. J. (1996). **An Efficient Routing Protocol for Wireless Networks**. ACM Mobile Networks and Applications Journal, Special Issue on Routing in Mobile Communication Networks. ACM.

Pandeny, A. K., and Fujinoki, H. (2005). **Study of MANET Routing Protocols by Glomosim Simulator**. International Journal of Network Management .

Pathan, A. K., and Hong, C. S. (2009). **Routing in Mobile Ad Hoc Networks**. In S. Misra, I. Woungang, & S. C. Misra, Guide to Wireless Ad hoc Networks (pp. 59-96). Springer.

Perkins, C. B., and Bhagwat, P. (1994). **Highly Dynamic Destination-Sequence Distance Victor (DSDV) routing for Mobile Ad Hoc Networks**. In Proceeding of the ACM Conference on Communications Architecture, Protocols and Applications.

Perkins, C., Belding-Royer, E. M., and Das, S. (2003). **Ad hoc On-demand Distance vector (AODV) Routing**. Retrieved April 2008, from Internet Engineering Task Force (IETF), Expermintal rfc 3561: www.ietf.org/rfc/rfc3561.txt

Sarkar, S. K., Basavaraju, T. G., and Puttamadappa, C. (2008). **Ad Hoc Mobile Wireless Networks: Principles, Protocols, and Applications**. Auerbach Publications.

Siva Ram, C., and Manoj, B. (2004). **Ad Hoc Wireless Networks: Architecture**. Prentice Hall.

Sivajothi, M., and Naganathan, E. R. (2007). **Video Transmission Performance Evaluation of Ad Hoc Routing Protocols with Emphasis on Ant Colony Optimization Technique**. International Conference on Computational Intelligence and Multimedia Applications .

Stefano, B., Marco, C., Silvia, G., and Ivan, S. (2004). **MOBILE AD HOC NETWORKING**, A JOHN WILEY and SONS, INC., PUBLICATION, IEEE PRESS.

Tavli, B., and Heinzelman, w. (2006). **Mobile Ad Hoc Networks Energy-Efficient Real-Time Data Communications**. Netherlands: Springer.

Appendix A: The RORP Performance Improvement Results

In this section, we show the percentage improvement results from the different experiments discussed in Chapter 4. The improvement tables below show the advantages of the RORP as compared to AODV and WRP protocols.

1. The Percentage Improvement Results from Varying Pause Time Experiments

1.1 The Delivery Packet Ratio

Table 1a: The Improvements of Delivery Packet Ratio of RORP with Varying RWP Pause Time

Pause time (sec)	AODV %	WRP %
5	+0.193467	+39.28787
10	2.616216	38.21193
100	0.91586	40.49263
200	0.936384	41.58588
400	2.21596	44.05498
average	1.360228	40.66041

1.2 The average end-to-end delay

Table 2a: The Improvements of average End-To-End Delay of RORP with Varying RWP Pause Time

Pause time (sec)	AODV %	WRP%
5	-232.353	-217.647
10	-203.335	-219.164
100	-201.499	-223.985
200	-275.213	-308.186
400	-339.56	-397.558
Average	-246.123	-268.567

(-) means the average delay is decreased by

1.3 The Average Number of Control Overhead

Table 3a: The Improvements of Control Overhead of RORP with Varying RWP Pause Time

Pause time (sec)	AODV%	WRP%
5	-65.4099	67.8436
10	-68.8491	69.1952
100	-68.8524	68.04234
200	-79.3298	66.55607
400	-101.869	63.75701

(-) means the control overhead is decreased by

1.4 The Average Throughput

Table 4a: The improvements of Throughput of RORP with varying RWP pause time

Pause time (sec)	AODV	WRP
5	+1.67213	+40.97452
10	3.712496	39.22479
100	2.461131	41.59945
200	2.752135	42.28082
400	4.324821	44.3049
average	2.955691	41.6269

2. The Percentage Improvement Results From Varying Speed

2.1 Packet Delivery Ratio

Table 5a: The Improvements of Delivery Ratio of RORP with Varying RWP Speed

Speed (m/sec)	AODV	WRP
5	-1.11748	28.31314
10	2.616216	38.21193
20	2.088413	43.08396
30	1.13136	53.39226
Average	1.179628	40.75032

2.2 Average End-to-end Delay

Table 6a: The Improvements of Average End-To-End-Delay of RORP with Varying RWP Speed

Speed (m/sec)	AODV	WRP
5	-158.123	-256.859
10	-203.335	-219.164
20	-168.468	-117.838
30	-156	-131.733

(-) means the average delay is decreased by

2.3 Control Overhead

Table 7a: The Improvements of Control Overhead of RORP with Varying RWP Speed

Speed (m/sec)	AODV	WRP
5	-60.2376	70.43907
10	-68.8491	69.1952
20	-69.0157	68.8215
30	-66.5577	68.95988

(-) means the control overhead is decreased by

2.4 Throughput

Table 8a: The Improvements of Throughput of RORP with Varying RWP Speed

Speed (m/sec)	AODV%	WRP%
5	1.108414	30.90796
10	3.712496	39.22479
20	3.360821	44.08734
30	2.086395	49.0856
Average	2.558354	40.4647

3. The Percentage Improvement Results From Varying Node Density

3.1 Packet Delivery Ratio

Table 2a: The Improvements of Packet Deliver Ratio of RORP with Varying Node Density

No. of nodes	AODV%	WRP%
50	-1.27521	41.31221
75	-0.41337	28.84821
100	2.234361	38.44059
Average	0.181927	36.20034

3.2 Average End-to-end Delay

Table 3a: The Improvements of average End-To-End of RORP with Varying Node Density

No. of nodes	AODV%	WRP%
50	-137.401	-203.979
75	-447.619	-614.286
100	-410	-900

(-) means the average delay is decreased by

3.3 Control Overhead

Table 4a: The Improvements of Control Overhead of RORP with Varying Node Density

No. of nodes	AODV%	WRP%
50	-93.7113	56.65272
75	-170.614	46.66904
100	-190.28	36.36129

(-) means the control overhead is decreased by

3.4 Throughput

Table 5a: The Improvements of Throughput of RORP with Varying Node Density

No. of nodes	AODV%	WRP%
50	0.019298	37.94329
75	-0.22738	24.9669
100	2.499849	35.4477
Average	0.774144	33.14459

4. The Percentage Improvement Results from Varying Traffic Transmission Rate

4.1 Packet Delivery Ratio

Table 6a: The Improvements of Packet Delivery Ratio of RORP with Varying Traffic Rate

Traffic rate (p/sec)	AODV%	WRP%
15	9.347991	22.92898
10	4.572626	32.58623
8	2.616216	38.21193
4	-23.3263	45.69912
2	-14.8196	46.7015
Average	-7.49867	39.64677

4.2 Average End-to-end Delay

Table 7a: The improvements of average End-to-End delay of RORP with varying traffic rate

Traffic rate (p/sec)	AODV%	WRP%
15	-45.3061	-90.4762
10	-83.5985	-127.417
8	-203.335	-219.164
4	-84.5018	-38.3764
2	-21.6981	36.79245

(-) means the average delay is decreased by

4.3 Control Overhead

Table 8a: The Improvements of Control Overhead of RORP with Varying Traffic Rate

Traffic rate (p/sec)	AODV%	WRP%
15	-95.3296	76.13706
10	-70.0732	73.52683
8	-68.8491	69.1952
4	-0.73733	52.49799
2	27.94928	42.22712

(-) means the control overhead is decreased by

4.4 Throughput

Table 9a: The Improvements of Throughput of RORP with Varying Traffic Rate

Traffic rate (p/sec)	AODV%	WRP%
15	20.21172	18.98144
10	5.074709	31.63347
8	3.712496	39.22479
4	-23.2715	48.06375
2	-15.9231	49.38944
Average	2.533689	34.20533

بروتوكول المستقبل الموجه لايجاد المسار في شبكات التنقل العشوائية

إعداد

عبدالعزيز يحيى يحيى النهاري

المشرف

الدكتور صالح حسني الشرايعة

الملخّـــــــــص

الشبكات المتنقلة العشوائية (MANET) عبارة عن شبكة لاسلكية تتشكل من ذاتها وبشكل ديناميكي بدون الحاجة للشبكات الذاتية المعدة مسبقا، حيث أن الأجهزة المتحركة تشكل بشكل عشوائي منطقة الاتصال. هذه الأجهزة لها القدرة على التحكم في عملية إرسال واستقبال البيانات و تحديد كيفية تحرك البيانات خلال الشبكة. حيث ان الشبكة تكون محدودة بمدى كل جهاز لاسلكي. في حالة ان يكون الجهاز المراد الوصول إليه ليس ضمن المدى فان البيانات تنتقل بين الأجهزة مكونة جسر بين المرسل لهذه البيانات والمستقبل لها. طبيعة الشبكات اللاسلكية والشبكات المتنقلة عشوائيا المتحركة جعلت انتقال البيانات تواجه صعوبة وتحدي أما المرسل لإيجاد طريق ثابت لإرسال البيانات الى المستقبل. من هذه التحديات تحرك الأجهزة الحر مما يؤدي الى تغير شكل الشبكة بشكل متكرر، كذلك محدودية الموارد كالطاقة الكهربائية والنطاق الترددي. نتيجة لهذه التحديات ظهرت العديد من الدراسات المختلقة الخاصة بتحسين أداء الشبكات العشوائية المتنقلة.

في هذه الرسالة قمنا بدراسة الخوارزميات الخاصة بإيجاد الطرق المختلفة بين المرسل والمستقبل. الخوارزمية الذي اقترحناه (RORP) هو من نوع هجين (hybrid) الذي دمج بين (reactive) و (proactive) بحيث اخذ المميزات الخاصة بالخوارزميات من اجل زيادة كفائية وأداء الشبكات اللاسلكية.

تم بناء وتجربة النظام المقترح باستخدام نظام المحاكاة الشبكية باستخدام بيئات وتجارب مختلفة. تمحورت دراسة النتائج لعدد من الكفاءات مثل التأخير الحاصل للبيانات، وكفائية انتقال البيانات خلال الشبكة. مما يجدر الإشارة إليه أن النظام المقترح استطاع بشكل ملحوظ من تقليل التأخير مقارنة ببعض الأنظمة المعروفة. هذا بالاظافة الى زيادته لفاعلية الشبكة من حيث زيادة إمكانية إرسال كم اكبر من البيانات وتقليل من الضغط على الشبكة وزيادة في سرعة إرسال هذه البيانات.