Sudan University of Science and Technology

College of Graduate Studies

Performance Evaluation of Routing Protocols in Ad Hoc Networks

تقيم أداء برتوكولات التوجيه في الشبكات اللاسلكية العشوائية

A thesis submitted in partial fulfillment of the requirements the degree of M.Sc. in computer and networks engineering

By:

Islam Ahmed Algadi.

Supervisor:

Dr. Hisham Ahmed.

May 2017

DEDICATION

To Mom and Dad

and all of my friends

without whom none of

my success would be possible

ACKNOWLEDGEMENTS

I express my gratitude towards The Almighty God for His blessings upon me. I owe my profound gratitude to my thesis supervisor Dr. Hisham Ahmed for his valuable guidance, supervision and persistent encouragement. Due to the approach adopted by him in handling my thesis and the way he gave me freedom to think about different things, I was able to do the constructive thesis. By working under him I have gained priceless knowledge as to how to go about doing an effective research. It is extremely hard to find words that express my gratitude to my parents, my sibling and my friends for their invaluable help over this year. I wish them all good luck in their future plans. They gave me courage and strength whenever I needed it and supported me in every possible way throughout these years.

Islam Algadi.

مستخلص البحث

الشبكات اللاسلكية العشوائية المتنقلة هي نوع من الشبكات اللاسلكية التي تتمتع بالادارة الذاتية للعقد المتنقلة المتصلة بروابط لاسلكية . العقد المتنقلة تكون حرة في التنقل بطريقة عشوائية . هناك بعض التحديات التي تواجه مصممي البروتوكولات و مطوري الشبكات . تشمل هذه التحديات التوجيه , الخدمة و تغير الهيكلية الكثير. في هذه الاطروحه أُخذت مشكلة التوجيه في الاعتبار و ستركز على ثلاثة برتوكولات معروفة : برتوكول توجيه الموجهات عند الطلب العكسي ، برتوكول توجيه المسافة الحركي عند الطلب و برتوكول التوجيه بحالة الارتباط المحسن. هذه البرتوكولات يتم تنفيذها باستخدام محاكي الشبكات - 2.35 . تقارن المحاكاة بروتوكولات التوجيه باستخدام بروتوكول التحكم فى الارسال وبروتوكول مخطط بيانات المستخدم مع مقاييس أداء مختلفة .

نتائج المحاكاة باستخدام بروتوكول التحكم فى الارسال تشير ان زيادة العقد تؤدي لتدهور الانتاجية الى 470.6 كيلوبت/ثانية من 682 كيلوبت/ثانية في برتوكول توجيه الموجهات عند الطلب العكسي . برتوكول توجيه المسافة الحركي عند الطلب يمتلك انتاجية عالية 685.93 كيلوبت/ثانية مع 50 عقدة. في حين ان زمن التأخير من نهاية الى نهاية يقل كلما زاد حجم الشبكة الى 50 عقدة. برتوكول توجيه المسافة الحركي عند الطلب لديه زمن تاخير منخفض مقارنة بالآخرين . برتوكول توجيه المسافة الحركي عند الطلب لديه أعلى نسبة وصول الحزم. هي 99.11 % . مع وبروتوكول مخطط بيانات, الانتاجية 479.9 كيلوبت/ثانية في برتوكول توجيه الموجهات عند الطلب العكسي, 153.78 كيلوبت /ثانية في برتوكول التوجيه بحالة الارتباط المحسن و 119.63 كيلو بت/ثانية في برتوكول توجيه المسافة الحركي عند الطلب مع 50 عقدة . تاخير الارسال في بروتوكول مخطط بيانات في حالة 10 عقد هو 0.652 ثانية و 1.2 ثانية لـ 50 عقدة في برتوكول التوجيه بحالة الارتباط المحسن. التأخير في برتوكول التوجيه بحالة الارتباط المحسن يزيد عندما يزداد عدد العقد مقارنة بـالآخرين . نسبة وصول الحزم يقل عند زيادة حجم الشبكة الى 30.29% في برتوكول التوجيه بحالة الارتباط المحسن و الى 23.56 % في برتوكول توجيه المسافة الحركي عند الطلب بينما نسبة توصيل الحزم في برتوكول توجيه الموجهات عند الطلب العكسي هي 98% مع 10 عقد و 94.55 % مع 50 عقدة.

ABSTRACT

Mobile ad hoc Networks (MANETs) are a type of wireless ad hoc network which is a self-arranging network of mobile nodes connected by wireless links. The mobile nodes are free to move randomly. There are some challenges that protocols designers and networks developers are faced with. These challenges include routing, service and frequently topology changes. In this thesis the problem of routing is considered and it will focus on three well-known protocols: Revers Ad-hoc On Distance Vector (R-AODV), Destination Sequenced Distance Vector (DSDV) and Optimized Link State Routing (OLSR) Protocol. The routing protocols are implemented using ns-2.35. The simulation compares the routing protocols with using transmission control protocol (TCP) and user datagram protocol (UDP) with different performance metrics.

Simulation results of TCP traffic indicate increasing nodes deteriorates throughput which goes to 470.6 kb\s from 682 kb\s in R-AODV. DSDV has high throughput of 685.93 kb\s with 50 nodes. While end to end delay (EED) decreases as network size is increased to 50 nodes. DSDV has low EED compared to other. DSDV has highest packet delivery ratio (PDR) 99.11 % among all other protocols. With UDP traffic, the throughput is 479.9 kb/s in R-AODV, 153.78 kb\s in OLSR and 119.63 kb\s in DSDV with 50 nodes. R-AODV has highest value. UDP transmission delay in case of 10 nodes is 0.652 second and 1.65 second of 50 nodes in OLSR. The delay OLSR increases when increasing number of nodes as compared to other. The PDR decreases as network size is increasing to 30.29 % and in OLSR to 23.56 % in DSDV while PDR of R-AODV is 98% with 10 nodes and 94.55 % with 50 nodes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ABR	Associativity-Based Routing
AODV	Ad-hoc on Distance Vector
ARPAM	Aeronautical Mobile Ad-hoc Networks
CBR	Constant Bit Rate
CGSR	Cluster Gateway Switch Routing Protocol
CPU	Central Processing Unit
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DAG	Directed Acyclic Graph
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
EED	End to End Delay
FTP	File Transfer Protocol
GloMoSim	Global Mobile Information Systems Simulation
LAN	Local Area Network
MAC	Media Access Control
MANET	Mobile Ad-Hoc Networks
MPR	Multipoint Relays
NAM	Network Animator
NetSim	Network Simulation
NS2	Network Simulator 2
OLSR	Optimized Link State Routing
OMNET++	Operation and Maintenance New Equipment Training
OORP	Order One MANET Routing Protocol
OPNET	Optimized Network Engineering Tool
OTCL	Object oriented Tool Command Language
PDR	Packet Delivery Ratio

R-AODV	Revers Ad-hoc On Distance Vector
RERR	Route Error
RREP	Route Reply
RREQ	Route Request
R-RREQ	Reverse Route Request
SSA	Signal Stability-Based Adaptive Routing Protocol
TC	Topology Control
TCP	Transmission Control Protocol
TORA	Temporally Ordered Routing Algorithm
UDP	User Datagram Protocol
VM	Virtual Machine
WAN	Wide Area Network
WIFI	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WRP	Wireless Routing Protocol
ZRP	Zone Routing Protocol

CHAPTER ONE

Introduction

Chapter One

Introduction

1.1 Overview

A Mobile Ad-hoc Network (MANET) is a multi-hop temporary autonomous system where the mobile users communicate with each other through wireless links, without any pre-established infrastructure. The decentralized network structure may vary rapidly and unpredictably over time because the nodes in the network are mobile and can be connected dynamically in an arbitrary manner. In recent years, the interest in ad hoc networks is at their high because of the availability of wireless communication devices. Routing in this type of networks can be implemented by many routing protocols that can be categorized under different criteria. The most general distinction of MANET routing protocols is proactive and reactive, with hybrid protocols spanning between these two categories. Some of the most popular protocols examined in previous studies are Dynamic Source Routing (DSR), Ad-hoc On-demand Distance Vector (AODV) and Temporally-Ordered Routing Algorithm (TORA), which belong to the reactive or on-demand category and Optimized Link State Routing (OLSR), Destination-Sequenced Distance-Vector (DSDV) and Wireless Routing Protocol (WRP), which belong to the proactive or table-driven category [1].

There have been made several performance evaluation studies that examine the performance and operation of these protocols, comparing them in terms of various metrics. Routing in the ad-hoc network becomes a more challenging task. Therefore it becomes recent research area in MANETs, Basically ad-hoc is a multi-hop wireless networks have been proposed for nomadic computing applications, with the advance of wireless communication low cost and powerful transceiver are widely used in the mobile application. The key requirements in all the above applications are reliable data transfer and congestion control, features that are generally supported by Transmission Control Protocol (TCP) . Unfortunately, TCP performs on wireless in a much less predictable way than on wired protocols. In this study we evaluate and compare the performance of CBR (Constant Bit Rate) over UDP (User Datagram Protocol) and FTP (File Transfer Protocol) over TCP traffic models using Reverse Ad hoc on Demand Distance Vector (R-AODV), Optimized Link State Routing (OLSR) and Destination-Sequenced Distance-Vector (DSDV), in a variable number of nodes to bring out their relative advantages.

1.2 Problem Statement

In recent applications, the trend towards MANET environment is scalability and dynamic mobility. It is difficult to design the routing protocols to overcome scalability and mobility. In MANET, the increasing number of mobile nodes under dynamic mobility leads to attract high control traffic overhead that affects the performance of routing protocol. It also needs high battery life and storage utilization, but it is extremely limited in energy and resource constraint environment. In this work, the performance evaluation is carried out in order to determine the best routing protocol which takes performance metrics under the different sizes of network. The simulated results produced in this study are useful to obtain the in-depth solution about the performance of routing protocol and guidelines to develop the effective routing protocol in the future. This thesis compares performance analysis of R-AODV, OLSR and DSDV Routing Protocols for MANETS using NS-2 based on performance metrics such as throughput, packet delivery ratio and end to end delay.

1.3 Proposed Solution

Through this thesis is found that how TCP and UDP will react under different network conditions. The network performance of different protocols varies under different parameters. In order to achieve this, FTP and CBR traffic conditions are used. End to end delay, throughput and packet delivery ratio are used with different numbers of nodes. This analysis is done to check the quality of service provided by routing protocols under different traffic conditions.

1.4 Aims and Objectives

The aim of this thesis is performance evaluation of R-AODV, OLSR and DSDV protocols in Ad -Hoc Networks. The outcome of this study is in the form of quantitative results of the efficiency of the routing protocols with performance metrics. These results can be used as the baseline for selecting routing protocols in a variety of situations.

The objectives of this thesis are:

- To implement different network scenarios using the NS2 simulator for different routing protocols.
- To analyze and compare the performance of TCP/FTP and UDP/CBR traffic in Reverse-AODV, OLSR and DSDV routing protocol generally implemented in a mobile ad hoc environment with different performance metrics.
- To understand their internal mechanism of working and suggest in high stressful situations which one is preferred among them.

1.5 Methodology

The thesis is based on the implementation and experiment in a simulation environment. Network Simulator 2 is chosen as a simulation environment. Specifically, the NS2 developer will be used to create experiment scenarios. NS2 has several already implemented routing protocols such as AODV, DSDV, DSR and TORA but OLSR protocol is not available as a part of NS-2.35. Other party software is taken that is developed by university of Murcia, Spain [2] called UM-OLSR, which is an implementation of OLSR protocol for NS-2 simulator. Um-olsr-1.0.tgz is used for patching & installing OLSR protocol in NS-2.35. Scenarios are generated by TCP and UDP traffics with varying the numbers of nodes. To simulate any network on NS2, the network parameters for simulation are assigned. This is done by configuring the simulator with the simulation parameters namely, the type of traffic pattern, protocol used, number of nodes, mobility model, simulation time etc. Each run of the simulator accepts a scenario file as input, which describes the position and motion of each node and the sequence of packets originating from each node. The detailed trace files created by each run are stored on disk, and analyzed using a script-routine (written in awk script), that counts the number of packets successfully delivered and the length of the paths taken by the packets, as well as additional information about the internal functioning of each protocol. The NS2 simulator gives two files as output; NAM (Network Animator) generates NAM file, which is used for graphical visualization and other file called trace file is used for calculating the results.

1.6 Thesis outlines

The thesis includes five chapters, chapter one provides introduction of it, the problem statement and objectives while chapter two covers background study of Ad-hoc routing protocols and highlights some of its threats and literature review.

In chapter three the methodology section, where the framework of the simulator, routing metric and simulation environment are defined while chapter four presents the implementation and performance evaluation results. And chapter five includes the conclusion and future work.

CHAPTER TWO

Background and Related Works

Chapter Two

Background and Related Works

This chapter describes the key concepts of ad hoc routing protocols. It describes the classifications in general, select three and give details about them that we have chosen to simulate and analyze. Reverse-AODV, OLSR and DSDV are considered. And it provides an overview of the latest trends of research going in the field of MANET.

2.1 Background

An ad hoc wireless network is a self-maintaining network and all the mobile nodes are interconnected in an arbitrary manner. Hence, the routing in ad hoc networks differs from fixed line protocols in that optimum routing is not the most important requirement for ad hoc routing. Features like rapid route convergence and high reactivity are deemed more important.

2.1.1 Ad Hoc Routing Protocols

The function of ad hoc routing protocol is to control the node decisions when routing packets between devices in MANET. When a node joins or tries to join the network it does not know about the network topology. By announcing its presence or by listening from the neighbor nodes it discovers the topology. In a network route discovery process depends on the routing protocol implementation.

2.1.2 Classification of Ad-hoc Routing Protocols

For wireless ad hoc networks, several routing protocols have been designed and all these protocols are classified under two major fields of protocols [3] called reactive or proactive. An ad hoc routing protocol with the combination of these two is called a hybrid protocol. The approaches involve a trade-off between the amount of overhead required to maintain routes between node pairs (possibly pairs that will never communicate), and the latency involved in discovering new routes as needed [4].

2.1.2.1 Proactive Protocols

Proactive protocols, also known as table-driven protocols, involve attempting to maintain routes between nodes in the network at all times, including when the routes are not currently being used. Updates to the individual links within the networks are propagated to all nodes or a relevant subset of nodes, in the network such that all nodes in the network eventually share a consistent view of the state of the network.

The advantage of this approach is that there is little or no latency involved when a node wishes to begin communicating with an arbitrary node that it has not yet been in communication with [5]. The disadvantage is that the control message overhead of maintaining all routes within the network can rapidly overwhelm the capacity of the network in very large networks, or situations of high mobility. Examples of pro-active protocols include the Destination Sequenced Distance Vector (DSDV), Optimized Link State Routing (OLSR), Wireless Routing Protocol (WRP) and Cluster Gateway Switch Routing Protocol (CGSR) [5]. WRP uses update message transmission to neighbor nodes. If node has update should send acknowledgements. CGSR is also a proactive protocol. In this protocol the nodes are separated into interrelated group of nodes. In these groups, one of the nodes elected as cluster head to achieve distributed mechanism [6].

2.1.2.2 Reactive Protocols

Reactive protocols, also known as on-demand protocols, involve searching for routes to other nodes only as they are needed. A route discovery process is invoked when a node wishes to communicate with another node for which it has no route table entry [5]. When a route is discovered, it is maintained only for as long as it is needed by a route maintenance process. Inactive routes are purged at regular intervals. Reactive protocols have the advantage of being more scalable than table-driven protocols. They require less control traffic to maintain routes that are not in use than in table-driven methods. The disadvantage of these methods is that an additional latency is incurred in order to discover a route to a node for which there is no entry in the route table. Dynamic Source Routing (DSR), the Ad-hoc On-demand Distance Vector Routing (AODV) protocol, Associativity-Based Routing (ABR) and Signal Stability-Based Adaptive Routing Protocol (SSA) are examples of on-demand protocols. SSA is a reactive protocol to obtain the more stable routes in ad hoc network [6]. This protocol performs a route discovery process by signal strength and location stability. In ABR, a route is discovered by the degree of association stability of nodes. In the network, to announce each node has to periodically generate beacon.

2.1.2.3 Hybrid Protocols

There exists another class of ad-hoc routing protocols, such as the Zone Routing Protocol (ZRP) [5], which employs a combination of proactive and reactive methods. The Zone Routing Protocols maintains groups of nodes in which routing between members within a zone is via proactive methods, and routing between different groups of nodes is via reactive methods. Temporarily Ordered Routing Algorithm (TORA) is a reactive routing protocol with some proactive enhancements where a link between nodes is established creating a Directed Acyclic Graph (DAG) of the route from the source node to the destination In this protocol, direction of the link between two nodes determined by height parameter. Ad-hoc Routing Protocol for Aeronautical Mobile Ad-hoc Networks (ARPAM) [7] is primarily an on demand and distance-vector protocol which shares the features of the popular AODV protocol.

The Order One MANET Routing Protocol (OORP) [7] is an hybrid routing protocol which has been designed to operate in wireless mesh networks thanks to its capability to enable nodes communicating by digital radio to cooperate and can handle both highly dynamic and large networks

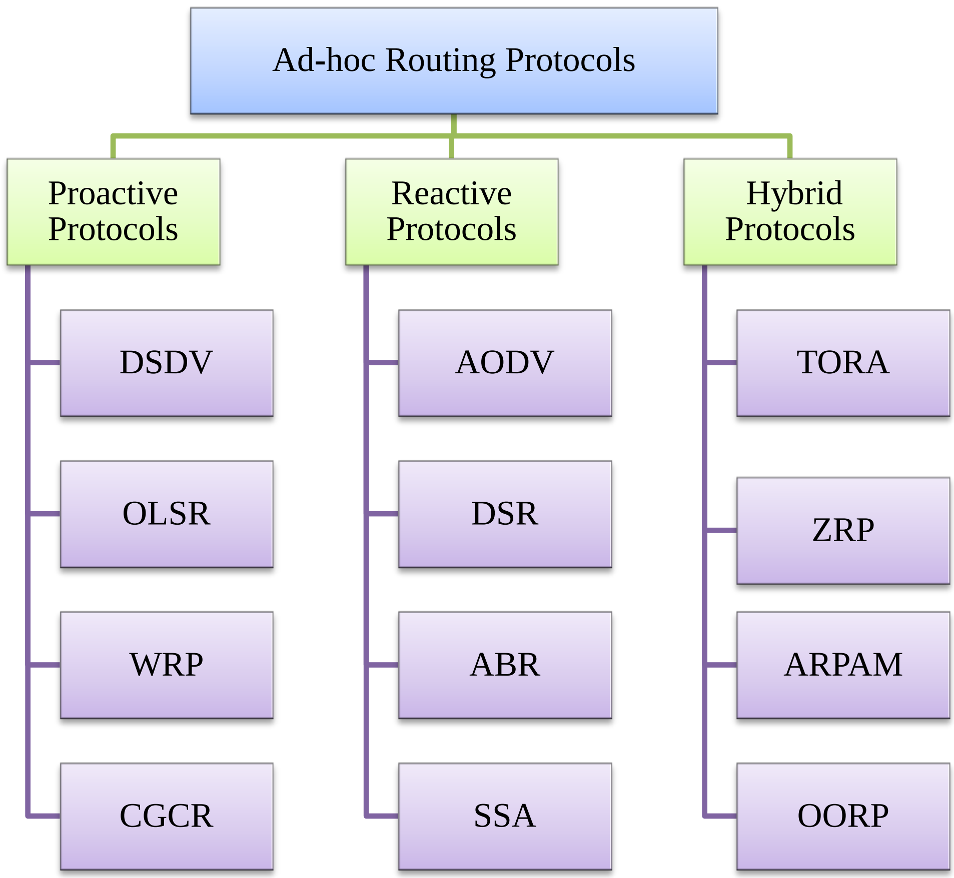

Figure 2.1 Ad-hoc Routing Protocols

2.1.3 Reverse – Ad hoc on Demand Distance Vector Protocol (R-AODV)

Analyzing previous protocols, we can say that most of the on-demand routing protocols, except multipath routing, uses single route reply along the first reverse path to establish routing path. As we mentioned before, in high mobility, pre-decided reverse path can be disconnected and route reply message from destination to source can be missed. In this case, source node needs to retransmit route request message. The R-AODV protocol discovers routes on-demand using a reverse route discovery procedure. During route discovery procedure source node and destination node plays the same role from the point of sending control messages. Thus, after receiving route request (RREQ) message, destination node floods reverse request (R-RREQ), to find source node. When the source node receives an R-RREQ message, data packet transmission is started immediately. Since R-AODV is reactive routing protocol, no permanent routes are stored in nodes. The source node initiates route discovery procedure by broadcasting. The RREQ message contains following information: message type, source address, destination address, broadcast ID, hop count, source sequence number, destination sequence number, request time (timestamp) [4].

Whenever the source node issues a new RREQ, the broadcast ID is incremented by one. Thus, the source and destination addresses, together with the broadcast ID, uniquely identify this RREQ packet. The source node broadcasts the RREQ to all nodes within its transmission range. These neighboring nodes will then pass on the RREQ to other nodes in the same manner. As the RREQ is broadcasted in the whole network, some nodes may receive several copies of the same RREQ. When an intermediate node receives a RREQ, the node checks if already received a RREQ with the same broadcast id and source address. The node cashes broadcast id and source address for first time and drops redundant RREQ messages. The procedure is the same with the RREQ of AODV. When the destination node receives first route request message, it generates so called reverse request (R-RREQ) message and broadcasts it to neighbor nodes within transmission range like the RREQ of source node does. In figure 2-2, R-RREQ message contains the following information: reply source id, reply destination id, reply broadcast id, hop count, destination sequence number, reply time (timestamp) [4]. When broadcasted R-RREQ message arrives to the intermediate node, it will check for redundancy. If it already received the same message, the message is dropped, otherwise forwards to next nodes.

<table>
<tr><td>Type</td><td>Reserved</td><td>Hop Count</td></tr>
<tr><td colspan="3">Broadcast ID</td></tr>
<tr><td colspan="3">Destination IP Address</td></tr>
<tr><td colspan="3">Destination Sequence Number</td></tr>
<tr><td colspan="3">Source IP address</td></tr>
<tr><td colspan="3">Request Time</td></tr>
</table>

Figure 2.2 Reverse RREQ packet format [4]

Furthermore, node stores or updates following information of routing table:

- Destination Node Address
- Source Node Address
- Hops up to destination
- Destination Sequence Number
- Route expiration time and next hop to the destination node.

And whenever the original source node receives first R-RREQ message it starts packet transmission and late arrived R-RREQs are saved for future use. The alternative paths can be used when the primary path fails communications.

In figure 2-3, destination does not unicast reply along pre-decided shortest reverse path D -> 3-> 2 ->1->S. Rather, it floods R-RREQ to find source node S. And forwarding path to the destination is built through this R-RREQ. Following paths might be built:

S->4-> 5 -> 6-> D, S->11->10 ->9->8 ->7 ->D, and etc. Node S can choose best one of these paths and start forwarding data packet. So route replay (RREP) delivery fails problem on AODV does not occur in this case, even though node 1 moves from transmission range.

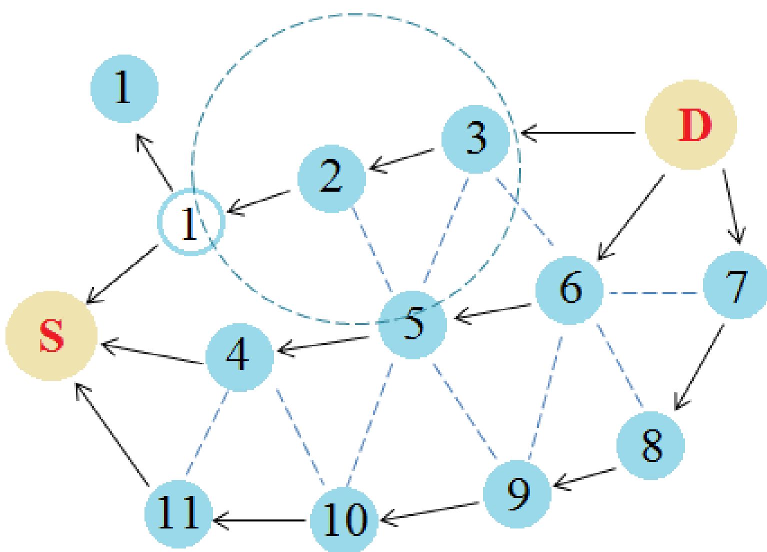

Figure 2.3 R-RREQ from Destination to Source Node

2.1.3.1 Route Discovery and Maintenance

Whenever the control packets are received, the source node selects the better path to update, that is the first node compares sequence numbers, and if it is higher sequence number meaning it indicates recent routes. If it has same sequence numbers, then a next number of route hops up to the destination are compared, usually routing path with lesser hops is selected. As the wireless communication channel quality is time changing the best path differs after some time. The advice from the medium access layer can be utilized to distinguish the availability of the connection. On the off chance that disappointment happens closer to the destination node, RRER control message received nodes can attempt neighborhood repair, generally the nodes forward RRER control message until it comes to the source node [4]. The source node can choose option route or trigger another route disclosure strategy.

2.1.4 Destination Sequenced Distance Vector Protocol (DSDV)

Perkins et. al. [8] proposed destination sequence distance vector routing protocol based on the traditional Bellman Ford algorithm [8] with some improvements to prevent count to infinity problem. Each node maintains routing table having entries corresponding to all other nodes in the network. Each node maintains a set of distances to reach the destination via its neighbors and chooses the neighbor as next hop having a minimum distance for packet delivery to that destination. It is a proactive protocol so the nodes periodically transmit their routing tables to their immediate neighbors or whenever the change in topology occurs. While sending an update message, a node has to increment its sequence number. Whenever a node receives a broadcasted routing message from its neighborhood, it compares received message's sequence number and hop count fields with the corresponding value stored in its routing table and updates its routing table depending on larger sequence number and smaller hop count by re-computing the distances. DSDV responds to RERR messages by invalidating all routes in their routing table containing broken link. These routes are immediately assigned an infinite metric and an incremented sequence number [8].

The advantages of this protocol are route discovery latency is very low as a route is always available, generates loop-free paths and count to infinity problem of distance vector routing is also removed. And extra traffic can be avoided using incremental update strategy than sending full updates. While disadvantages [8] are wastage of bandwidth and a large amount of network overhead while transmitting periodic route update messages. So, DSDV does not scale well in large and dense networks and it doesn't support multipath routing since the single path to the destination is maintained.

2.1.5 Optimized Link State Routing Protocol (OLSR)

OLSR is a proactive routing protocol, so the routes are always immediately available when needed. OLSR is an optimization version of a pure link state protocol [8]. So the topological changes cause the flooding of the topological information to all available hosts in the network. To reduce the possible overhead in the network protocol uses Multipoint Relays (MPR) [8]. MPRs have selected nodes which forward broadcast messages during the flooding process. MPRs provide the shortest path to a destination by declaring and exchanging the link information periodically for their MPR's selectors. By doing so, the nodes maintain the network topology information. The MPR is used to reduce the number of nodes that broadcasts the routing information throughout the network. To forward data traffic, a node selects its one hop symmetric neighbors, referred to as MPR set that covers all nodes that are two hops away. The MPR set is calculated from information about the node's symmetric one hop and two hop neighbors. This information in turn is extracted from HELLO messages.

Similar to the MPR set, a MPR Selectors set is maintained at each node. A MPR Selector set is the set of neighbors that have chosen the node as a MPR. Upon receiving a packet, a node checks its MPR Selector set to see if the sender has chosen the node as a MPR. If yes, the packet is forwarded, otherwise the packet is processed and discarded. This technique substantially reduces the message overhead as compared to a classical flooding mechanism (where every node retransmits each message received). The MPR set is calculated from information about the node's symmetric one hop and two hop neighbors. This information in turn is extracted from HELLO messages Hello messages are interchanged at 0.5 second and Topology Control (TC) messages at 2 second interval. TC messages are flooded using the MPR optimization. This is done on a regular interval [8], but TC messages are also generated immediately when changes are detected in the MPR selector set. OLSR uses two kinds of the control messages: Hello and Topology Control.

HELLO messages are used for finding the information about the link status and the host's neighbors [8]. With the HELLO message the MPR Selector set is constructed which describes which neighbors have chosen this host to act as MPR and from this information the host can calculate its own set of MPRs. The HELLO messages are sent only one hop away but the TC messages are broadcasted throughout the entire network. TC messages are used for broadcasting information about own advertised neighbors which includes at least the MPR Selector list. The TC messages are broadcasted periodically and only the MPR hosts can forward the TC messages [9].

The advantages of this protocol are due to OLSR is distributed protocol so no central administration to handle the routing process, the link is reliable since the update messages are sent periodically, OLSR works well with for large and high density networks as optimization is done by using MPRs and routes are always available so no route discovery delays for finding a route. While disadvantages are
OLSR periodically sends the updated topology information throughout the entire network, it requires a reasonably large amount of bandwidth and CPU power for computing optimal routing paths in the network and in case of packet loss in the network, some nodes that are not part of MPR set may start retransmitting the packets.

2.2 Related Works

Several performance evaluations of MANET routing protocols using TCP and UDP traffic have been done by considering various parameters such as mobility, network load and pause time. In [10] Bindeshwar S. and Pramod K. Mishra compared DSDV, AODV and DSR by useing three traffic generators namely exponential, Pareto and CBR (Constant Bit Rate) over an ad hoc network and analyze the behavior of routing protocols. DSDV is showing better performance than AODV and DSR. In addition, after analyzing all three protocols it can be observed that there are optimal values of packet size and offered load for which value of throughput and PDR values are optimal, after that their values are decreased or become constant.

Santosh and Umesh [11] analyzed AODV. They evaluate the performance of WiMAX over WiFi through Network simulator NS-2. The analysis of TCP results is better than CBR Results. Network Traffic Load and delays were measured. By using WiMAX technology effective data transmission average end to end delay and network traffic load very low.

In [12] DSR, AODV and OLSR are compared. Dimitra , Anastasios used non-specific application traffic and FTP traffic at the same time. OLSR has the best performance of all three protocols in terms of PDR and AEED. In addition, Pravanjan and Upena [1] analyzed AODV and R-AODV. The performance of R-AODV in terms of packet delivery fraction, average end-to-end delay and average energy consumption completely dominates AODV at a cost of higher control overhead. It also shows that R-AODV uses lesser number of hops and shortest path to route the data packets.

In [13] AODV and R-AODV are analyzed. In RAODV they change route replay packet configuration of AODV and named it RRREQ. These packets should be transmitted to the destination node for building multiple routes. According to the simulation results, this algorithm is better than another version of AODV algorithm.

In [14] AODV and R-AODV were compared by Pravanjan Das. The results show that R-AODV completely out performs AODV for larger network size and low density networks with lower network mobility. AODV performs significantly well for higher network densities as compared to R-AODV.

Ritika Sharma, Kamlesh Gupta [15] analyzed AODV. This paper compares the two traffic scenarios that are TCP/FTP and UDP/ CBR. Throughput: AODV provides better efficiency with TCP/FTP than UDP/CBR. Packet Delivery Ratio (PDR): Although the PDR of UDP/CBR has greater maximum and minimum values than TCP/FTP, the latter offers almost a constant trend, whereas, the former offers highly varying (rising and falling trends), TCP/FTP is more reliable than UDP/CBR. Average End to End Delay: The UDP/CBR offers lesser, average end to end delay, than TCP/FTP.

Ramprasad and Vinay Somani [9]compared OLSR and DSDV. The comparison is done on the basis of parameters like PDF, Throughput, end-to-end delay and normalized routing overhead by taking the pause time 0, 40% and 100% of simulation time. It is clear that in less stressful environment (Low traffic load and mobility) DSDV gives better throughput and PDF value compared to OLSR. But at high traffic load the performance of DSDV degrades with increases in pause time. Also, DSDV suffers from large delay and normalized routing overhead compared to OLSR. The following table 2.1 shows the summary of related works.

Table 2.1 Summary of related works

No	Authors	Paper Title and Year	Protocols	Tool	Overview and Results
10	Bindeshwar S. Kushwaha,Pramod K. Mishra	Different Traffic Patterns Over Ad Hoc Network Routing Protocols, 2016	DSDV, AODV, DSR	NS2	-They use three traffic generators namely exponential, Pareto and CBR (Constant Bit Rate) over an ad hoc network and analyze the behavior of routing protocols. -DSDV is showing better performance than AODV and DSR.
11	Santosh Kumar Sharma, Umesh Barahdiy	Performance Analysis of Different Traffic Sources TCP and CBR in AODV MANET, 2016	AODV	NS2	-They evaluated and analyze the performance of WiMAX over WiFi through Network simulator NS-2. -The analysis of TCP results are better compare than CBR Results.
12	Dimitra Kampita, Anastasios A. Economi	Simulation study of MANET routing protocols under FTP traffic, 2014	DSR, AODV, OLSR	OMNeT++	-They used non-specific application traffic and FTP traffic at the same time. -OLSR has the best performance of all three protocols in terms of PDR and AEED.
1	Pravanjan Das, Upena D. Dalal	A Comparative Analysis of AODV and R-AODV Routing Protocols in MANETS, 2013	AODV, R-AODV	NS2	-The performance of R-AODV in terms of packet delivery fraction, average end-to-end delay and average energy consumption completely dominates AODV at a cost of higher control overhead. It also shows that R-AODV uses lesser number of hops and shortest path to route the data packets.
13	Sujata Wasudeorao Wankhade P. R. Deshmukh	Comparison of AODV and RAODV Routing Protocols in Mobile Ad Hoc Networks, 2013	AODV, R-AODV	NS2	In RAODV we changed route replay packet configuration of AODV and named it RRREQ. These packets should be transmitted to destination node for building multiple routes. According to the simulation results, this algorithm is better than other version of AODV algorithm.

14	Pravanjan Das	Comparison of AODV and R-AODV Routing Protocols by varying Network Mobility, Network Area and Network Density, 2013	AODV, R-AODV	NS2	-The results show that R-AODV completely Out performs AODV for larger network size and low density networks with lower network mobility. -AODV performs significantly well for higher network densities as compared to R-AODV.
15	Ritika Sharma, Kamlesh Gupta	Comparison based Performance Analysis of UDP/CBR and TCP/FTP Traffic under AODV Routing Protocol in MANET, 2012	AODV	NS2	-This paper compare the two traffic scenarios that are TCP/FTP and UDP/ CBR -Throughput: AODV provides better efficiency with TCP/FTP than UDP/CBR. -Packet Delivery Ratio (PDR): Although the PDR of UDP/CBR has greater maximum and minimum values than TCP/FTP, the latter offers almost a constant trend, whereas, the former offers highly varying (rising and falling trends), TCP/FTP is more reliable than UDP/CBR. -Average End to End Delay: The UDP/CBR offers lesser, average end to end delay, than TCP/FTP.
9	Ramprasad, Vinay Somani	Comparative Analysis of DSDV and OLSR Routing Protocols in MANET at Different Traffic Load, 2011	OLSR, DSDV	NS2	-The comparison is done on PDF, Throughput, end-to-end delay and normalized routing overhead by taking the pause time 0, 40% and 100% of simulation time. -With low traffic load and mobility, DSDV gives better throughput and PDF value compared to OLSR. -With high traffic load the performance of DSDV degrades with increases in pause time. Also, DSDV suffers from large delay and normalized routing overhead compared to OLSR

2.3 Summary

This chapter explores the Classification of Ad-hoc Routing Protocols. The most general distinction of MANET routing protocols is proactive and reactive, with hybrid protocols spanning between these two categories. Some of the most popular protocols are DSR, AODV and TORA, which belong to the reactive or on-demand category and OLSR, DSDV and WRP, which belong to the proactive or table-driven category. Furthermore, the chapter includes several performance evaluation studies that examine the performance and operation of these protocols, comparing them in terms of various metrics.

CHAPTER THREE

Simulation Setup

Chapter Three

Simulation Setups

3.1 Introduction

This section covers techniques, tools, performance metrics which are chosen for evaluating the performance of protocols. The importance of performance evaluation and simulation are also described in this section.

3.2 Importance of Performance Evaluation and Simulation

In a computer system performance is a key factor. All the software and hardware design go through the performance tests again and again before implementation. Integration of computer system in almost every walk of life demands a reliable computer network system. It is therefore considers necessary for all computer professionals, researchers and system engineers to acquire basic knowledge of performance evaluating technique. Performance can be evaluated via measurement, modeling and simulation [16]. The simulation technique is suitable for testing models especially in research areas and educational centers. Potential advantages of the simulation are, it saves time, cost and provides detail results and a good understanding of event's occurrence.

3.3 Network simulator

There are many simulators such as OPNET, NetSim, GloMoSim, NS3, OMNET++ and NS2 etc [17]. NS2 is used for simulation due to it is free, open source , support different types of networks such as wired Network, wireless ad-hoc mode, wireless managed mode and wired cum wireless [18] . Also NS-2 comes closer to reality than other simulators, NS-2 has the rich collection of models than others simulators and A good simulation design, good results can be achieved with NS-2.

NS2 is quite difficult to use for first time user but once user gets to know the simulator it becomes fairly easy. NS2 is a discrete event simulator developed at UC Berkeley and written in C++ and Object oriented Tool Command Language (OTCL) [19]. Primarily, NS2 was useful for simulating LAN (Local Area Network) and WAN (Wide Area Network) only. Multi-hop wireless network simulation support is provided by the Monarch Research Group [19] at Carnegie-Mellon University. For wireless simulation, it contains physical, data link and medium access control layer. The Distributed Coordination Function (DCF) of IEEE 802.11 for wireless LANs is used as MAC (Media Access Control) layer protocol. For transmitting data packets, an Unslotted Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is used. Radio model is similar to the commercial radio interface, Lucent's wave LAN. Wave LAN has a share-media radio with a nominal bit rate of 2 Mb/s and a nominal radio range of 250 m [19].

NS2 interprets OTCL scripts defined by user. A user describes various network components in OTCL such as libraries and scheduler objects which are then simulated by the main NS2 program written in C++. Figure 3.1 shows the architecture of NS2. The acceptance of NS2 in research and education sector is because of its free distribution and open source. NS2 is being developed and contributed by researchers and developers over the time. It is suitable for comparing different protocols, traffics and developing new protocols.

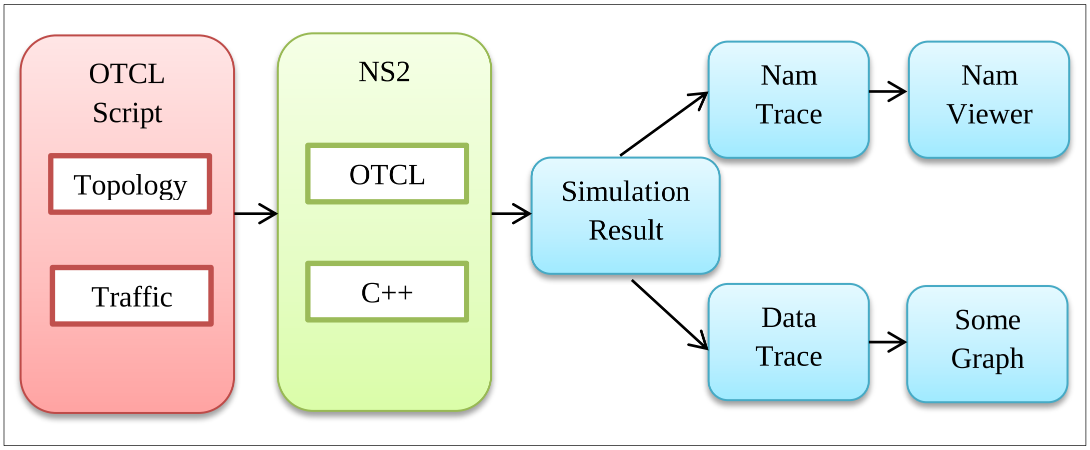

Figure 3.1 Architecture of NS2

3.4 Performance Metrics

In the evaluation of routing protocols different performance metrics are used. They show different characteristics of the whole network performance. In this performance comparison, we evaluate the packet delivery ratio, throughput and end to end delay of selected protocols in order to study the effects on the whole network.

3.4.1 Packet Delivery Fraction

It is the fraction of a number of packets received at the destination to the number of packets sent from the source [8].

3.4.2 Average End-to-End Delay

It is the time interval between sending the packet by the source node and receiving it at the destination node, [8] which includes buffering of data packets during route discovery, queuing at the interface queue and retransmission delays at the MAC.

3.4.3 Throughput

It is the rate of successfully transmitted data packets per second in the network during the simulation.

3.5 Evaluation Technique

The simulation software NS-2 .35 has been used for performance assessment of R-AODV, DSDV and OLSR based on various performance metrics. NS-2.35 is an open source network simulator that is widely used for networking research. Performance evaluation of different routing protocol is done on NS2 which is installed on virtual machine (VM) under the Linux platform (Ubuntu 14.04). The simulation environment consists of an area of 800m x 800m, where randomly 10 to 50 mobile nodes are placed. A source and a destination are selected randomly. Data sources generate data according to CBR and FTP traffic pattern. Source and destination pairs are spread randomly over the network. By observing the performance of the network under mobility it can be test the stability of design. The simulation parameters are shown in Table 3-1.

Table 3-1 Simulations Parameters

Parameters	**Values**
Traffic Agent Type	FTP / CBR
Data Type	TCP / UDP
MAC	802.11
Channel	Wireless
Network Size	800m x 800 m
Routing Protocol	R-AODV, DSDV and OLSR
Number of nodes	10, 20, 30, 40, 50
Simulation time	50 seconds

In NS2, the steps for getting trace and NAM files after the simulation are as follows:

i) Writing of the program in OTCL. OTCL is used to write the program for generate a network, network environment, and trajectory of mobile nodes.

ii) Run the **.tcl** file on the terminal under the Linux mint platform.

iii) NS2 trace analyzer is use to analyses trace file obtained during simulation and according to trace file generate the respective graphs.

3.6 Summary

Performance evaluation of different routing protocol is done on NS2 by considering different scenario. NS2 is a free simulator which provides the facility to set up network topology, configure and optimize the parameter according to the need of the application. The metrics to measure and compare the performance of the protocols are throughput, end to end delay and packet to delivery ratio.

CHAPTER FOUR

Results and Discussion

Chapter Four

Results and Discussion

4.1 Introduction

The following tables show the observations taken for the various configurations, and their effect on the three performance metrics for TCP/FTP, and UDP/CBR separately for R-AODV, OLSR and DSDV. The results are provided through graphs plotted as Performance metrics vs. numbers of nodes. Then they are compared between TCP/FTP and UDP/CBR for each protocol by using 30 nodes.

4.2 TCP/FTP Traffic

The following table 4.1 specifies the values of parameters used for TCP traffic.

Table 4.1 observations for varying number of nodes for TCP traffic

	Throughput (kb/s)			**End to End Delay (second)**			**Packet Delivery Ratio (%)**		
No. of nodes	R-AODV	OLSR	DSDV	R-AODV	OLSR	DSDV	R-AODV	OLSR	DSDV
10	682.0	593.2	658.93	0.5514	0.285	0.1798	98.203	97.90	99.093
20	527.5	313.0	703.58	0.2637	0.230	0.1226	97.415	98.24	97.613
30	685.1	396.3	444.19	0.4301	0.178	0.1200	98.309	95.53	98.511
40	666.2	276.2	690.58	0.5191	0.192	0.1328	97.979	95.10	95.089
50	470.6	627.8	685.11	0.4787	0.145	0.1459	97.867	98.77	99.110

4.2.1 Throughput for FTP Traffic

Figure 4.1 shows the response of throughput expressed in kb/s against the number of nodes for the three protocols taken from table 4.1.

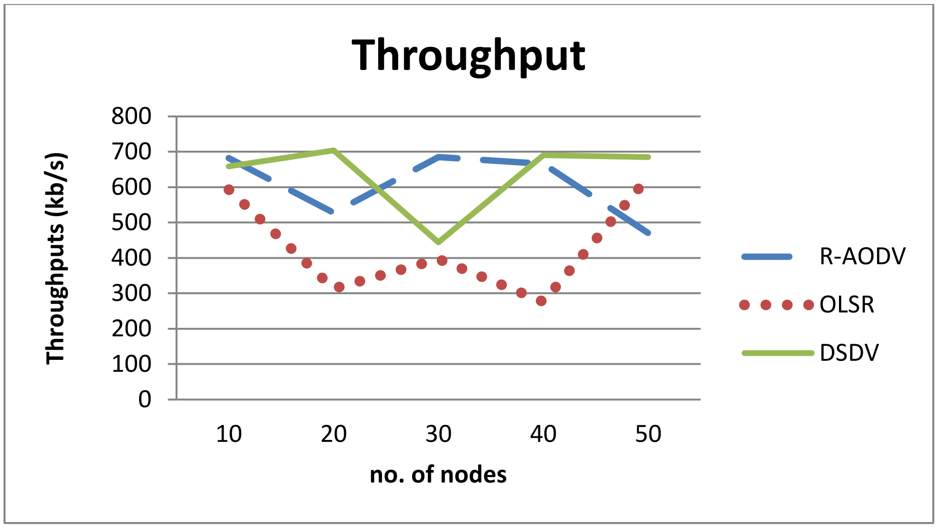

Figure 4.1 throughput vs. number of nodes for TCP traffic

Throughout is directly related to the packet drops. Packet drops typically happens because of network congestion or for lack of route. Figure 4.1 depicts the variation in throughput by increasing number of nodes. On an average throughput decreases as network density increases due to congestion and collision in the networks. The throughput of R- AODV and OLSR are decreased until the number of nodes is 20 after that the throughput is increased, at the same time the throughput of DSDV is started increase reverse others. It is the highest one among them.

4.2.2 End to End Delay for FTP Traffic

Based on the observations of table 4.1, the response of end to end delay in second against varying number of nodes is shown in figure 4.2

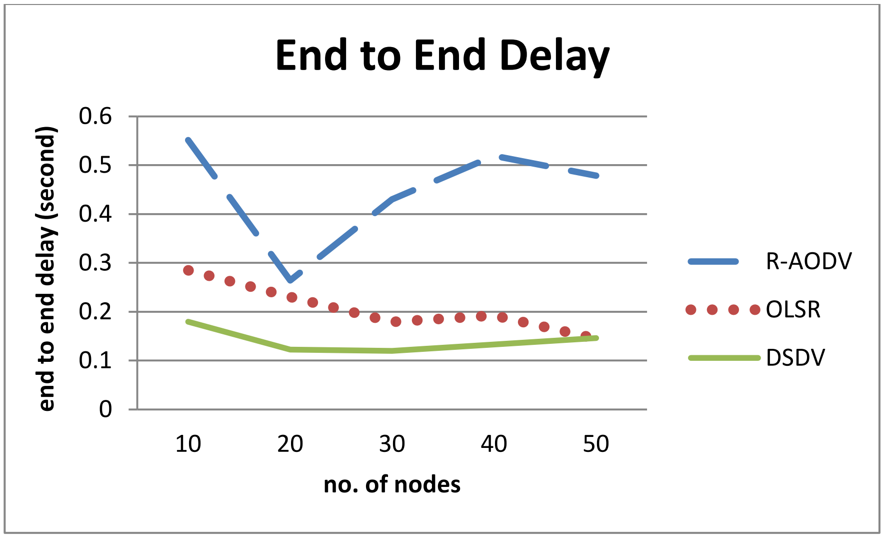

Figure 4.2 end to end delay vs. number of nodes for FTP traffic

Refer to figure 4.2 DSDV has low end to end delay compared to OLSR and R-AODV in the simulation scenarios. Due to in DSDV protocol, routes to every destination were always available and up-to-date. The R-AODV achieves high end-to-end delay due to its hop-by-hop routing methodology

4.2.3 Packet Delivery Ratio for FTP Traffic

Based on the observations of table 4.1, the response of packet delivery ratio against varying number of nodes is shown in figure 4.3

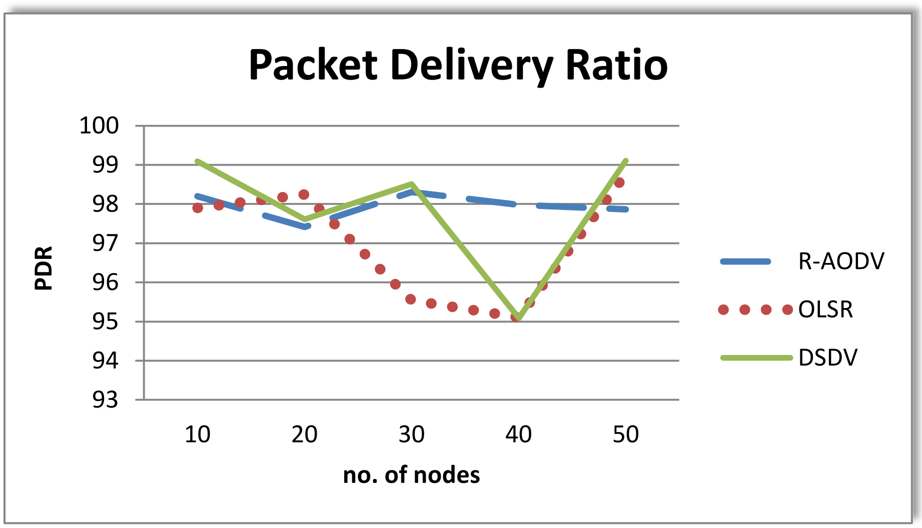

Figure 4.3 packet delivery ratio vs. number of nodes for FTP traffic

Refer to figure 4.3, PDR decreases with increasing number of nodes as congestion in network increases resulting in more dropped packets due to collisions When the number of nodes is between 40 to 50, the growth in PDR has increased (in the case of DSDV and OLSR) while the PDR of R-AODV maintains semi-fixed values that change with various number of nodes, but less than the previous protocols. DSDV has high PDR. Highest PDR value indicates the good performance.

The comparison between related works and this study is provided in table 4.2. It includes the network scenario, metrics and protocols are used with TCP traffic.

Table 4.2 comparison between related works and this study

	Reference [12]			**Reference [9]**				**This study**		
Network Scenario	Varying number of nodes			Low traffic load and mobility		High traffic load and mobility		Varying number of nodes		
Protocols /Metrics	AODV	OLSR	DSR	DSDV	OLSR	DSDV	OLSR	R-AODV	OLSR	DSDV
Throughput	-	-	-	High	Low	Low	High	Medium	Low	High
EED	Medium	Low	High	-	-	-	-	High	Medium	Low
PDR	Medium	High	Low	Hig4h	Low	Low	High	Medium	Low	High

From table 4.2 it is clear in [12] varying number of nodes and TCP traffic are used. AODV, OLSR and DSR are compered. PDF of DSR is low and the end to end delay is low while PDF of OLSR is high and end to end delay is low. The performance of AODV is among them. That means OLSR is better. Comparative analysis is done for DSDV and OLSR [9]. With Low traffic load and mobility, DSDV gives better throughput and PDR value compared to OLSR. But at high traffic is opposite. In this study when combined among R-AODV, OLSR and

DSDV with varying number of nodes, DSDV gives good performance in large network size (50 nodes) while R-AODV is good in small network size.

4.3 UDP/CBR Traffic

The following table 4.3 specifies the values of parameters used for UDP traffic.

Table 4.3 observations for varying number of nodes of UDP traffic

	Throughput (kb/s)			End to End Delay (second)			Packet Delivery Ratio (%)		
No. of nodes	R-AODV	OLSR	DSDV	R-AODV	OLSR	DSDV	R-AODV	OLSR	DSDV
10	497.6	290.08	233.8	0.0541	0.6521	0.0376	98.0	57.134	46.059
20	334.2	272.75	282.4	0.0482	0.5182	0.0317	65.834	53.735	55.639
30	409.6	178.05	131.6	0.0389	0.7894	0.0890	80.696	35.066	25.936
40	290.5	167.29	157.7	0.2071	1.3694	0.0254	57.19	32.958	31.074
50	479.9	153.78	119.6	0.4797	1.6573	0.3249	94.554	30.296	23.561

4.3.1 Throughput for CBR Traffic

The following Figure 4.4 shows the response of throughput expressed in kb/s against number of nodes for the three protocols obtained by table 4.4

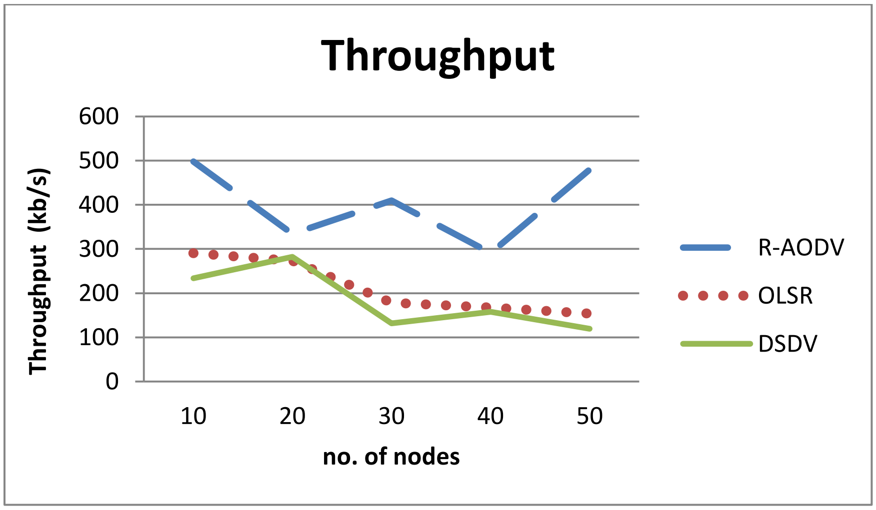

Figure 4.4 throughput vs. number of nodes for CBR traffic

It can be seen that at figure 4.4, R-AODV has more throughput as compared to OLSR and DSDV. Throughput in case of DSDV and OLSR decreases with increasing number of nodes because DSDV and OLSR routing protocols are table driven protocol and require more control overhead to maintain the route to every other node. DSDV and OLSR works efficiently under small scale networks. Since, it consumes less bandwidth owing to the less frequent broadcasting of update packets. Here R-AODV routing protocol showing best throughput with increasing number of node because in R-AODV routing protocol, routing table is established at every node, so there is no need to carry entire route information along with data packet that will decrease the control overhead.

4.3.2 End to End Delay CBR Traffic

Based on the observations of table 4.2, the response of end to end delay in second against varying number of nodes is shown in Figure 4.5

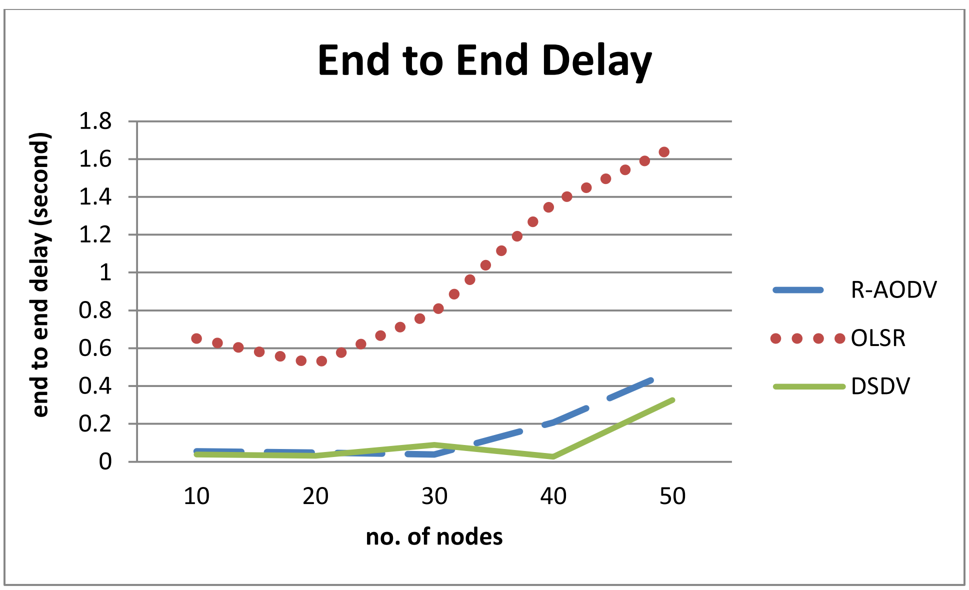

Figure 4.5 end to end delay vs. number of nodes for CBR traffic

Refer to figures 4.5, DSDV has low end to end delay compared to R-AODV and OLSR in the simulation scenario because updated route to the intended node is always available whenever any node wishes to send the data to any other node when number of nodes increases, R-AODV and DSDV take less time to deliver the packets to the destination than OLSR and it is still stable. So, the delay of OLSR increases when increasing number of nodes. For the large network, the route discovery process consumes more time to find the short hop count path to the destination. It causes the link failure often and it leads to the repeated route recovery process therefore it introduces a large delay in the network. The increased mobility causes more routing packet generation to find the fresh route. If the valid route is known under the route discovery process, data packets are forwarded to the destination; otherwise, data packets are buffered until the route is discovered, which makes delay in the data transmission.

4.3.3 Packet Delivery Ratio CBR Traffic

Based on the observations of table 4.2, the response of packet delivery ratio against varying number of nodes is shown in Figure 4.6

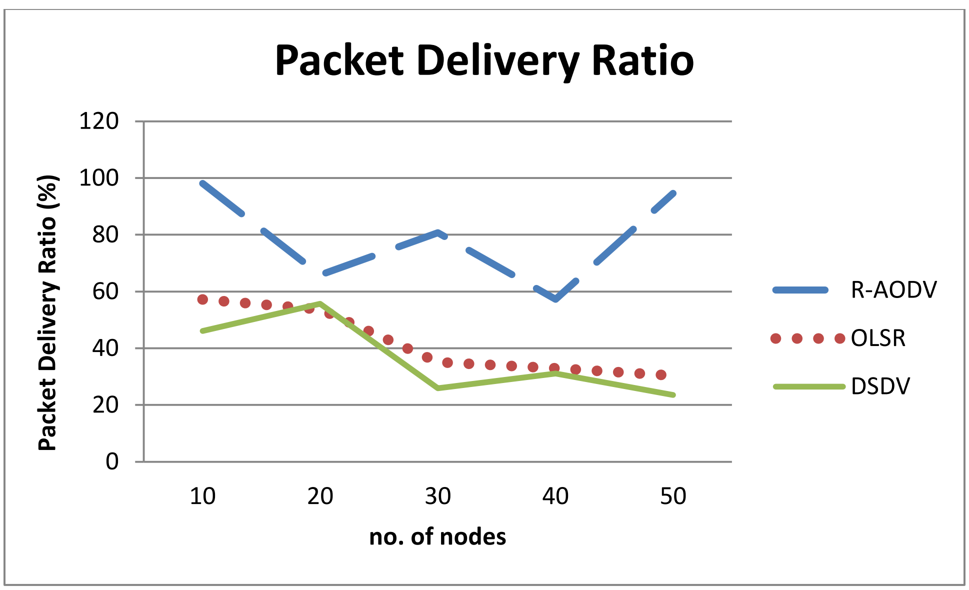

Figure 4.6 packet delivery ratio vs. number of nodes for CBR traffic

The packet delivery ratio of OLSR and DSDV protocol degrades, when the number of node increases. This is because; it is difficult to maintain the routing information under a large scale network. R-AODV has high PDR, because R-

AODV develop multipath in one route discovery process that means the chance of dropping decreases.

The comparison between related works and this study is provided in table 4.4. It includes the network scenario, metrics and protocols are used with UDP traffic.

Table 4.4 comparison between related works and this study

	Reference [10]			**Reference [1]**		**Reference [13]**		**This study**		
Network Scenario	Offered Load			Varying speed		Varying number of nodes		Varying number of nodes		
Protocols /Metrics	AODV	DSDV	DSR	AODV	R-AODV	AODV	R-AODV	R-AODV	OLSR	DSDV
Throughput	Medium	High	Low	-	-	-	-	High	Medium	Low
EED	-	-	-	High	Low	High	Low	Medium	High	Low
PDR	Medium	High	Low	Low	High	Low	High	High	Medium	Low

In [10] [1] [13], CBR traffic is used with various network scenarios. It is clear from table 4. 4 in [10] different offered Loads are applied with AODV, DSDV and DSR. DSDV has high performance and DSR is low. In [1] [13], the network scenario is varying speed and number of nodes. AODV and R-AODV are compered and the last is better in throughput and end to end delay. This study combines between R-AODV and DSDV beside OLSR. R-AODV gives the best performance.

4.4 TCP/FTP and UDP/CBR Traffic

The following table 4-5 specifies the parameters used for TCP and UDP traffics by using 30 nodes for R-AODV, OLSR and DSDV.

Table 4-5 observations for varying data types with 30 nodes

	Throughput (kb/s)			**End to End Delay (second)**			**Packet Delivery Ratio (%)**		
Data Type	R-AODV	OLSR	DSDV	R-AODV	OLSR	DSDV	R-AODV	OLSR	DSDV
UDP	409.6	178.05	131.67	0.0389	0.7894	0.0890	80.696	35.066	25.93
TCP	685.1	396.36	444.19	0.4301	0.1789	0.1200	98.309	95.536	98.51

4.4.1 Throughput

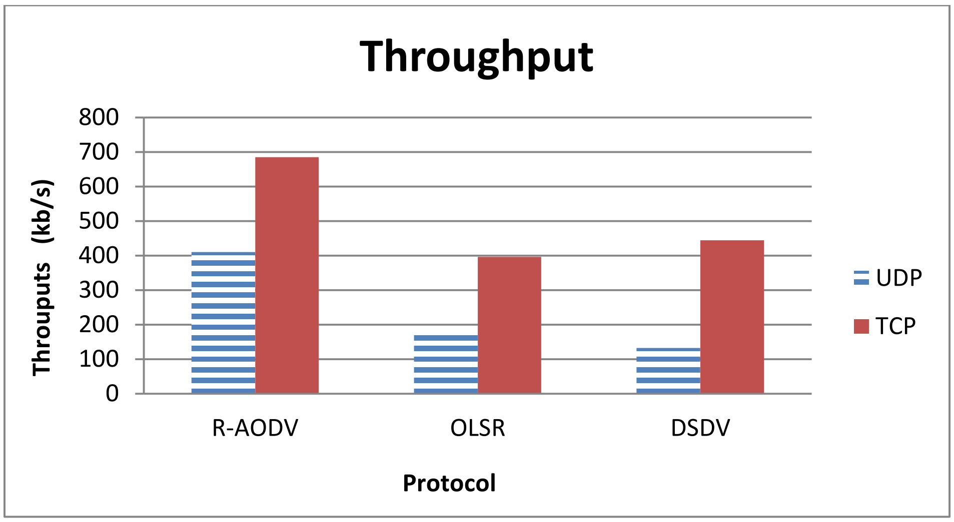

Figure 4.7 throughput with using TCP and UDP

From figure 4.7, Out of the two traffic types i.e. TCP/FTP and UDP/CBR, the TCP provides far better performance than the UDP. This proves that the network working with R-AODV, OLSR and DSDV provide better efficiency with TCP/FTP than UDP/CBR.

4.4.2 End to End Delay

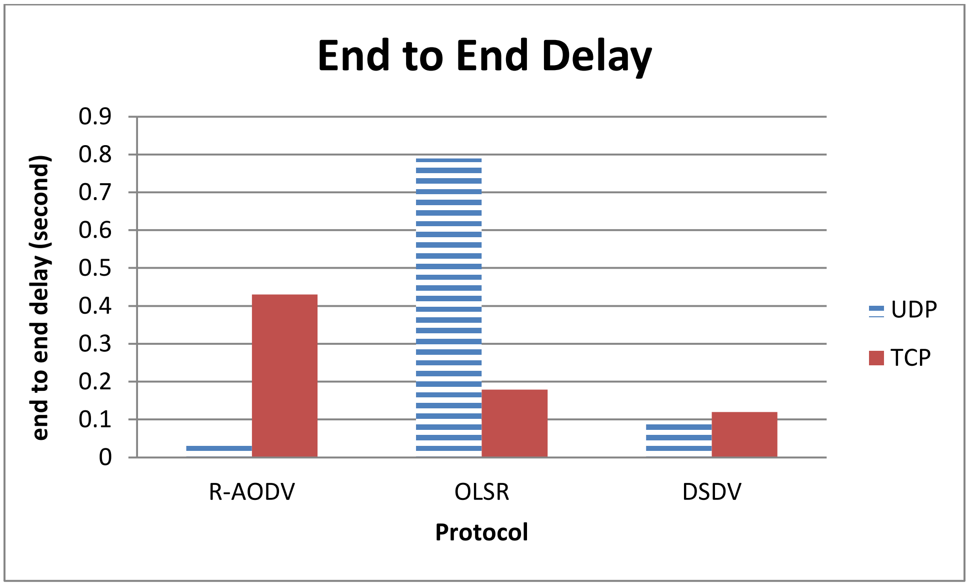

Figure 4.8 end to end delay with using TCP and UDP

The UDP/CBR offers lesser, end to end delay, than TCP/FTP, but as an exception in OLSR the end to end delay increases with UDP traffic.

4.4.3 Packet Delivery Ratio

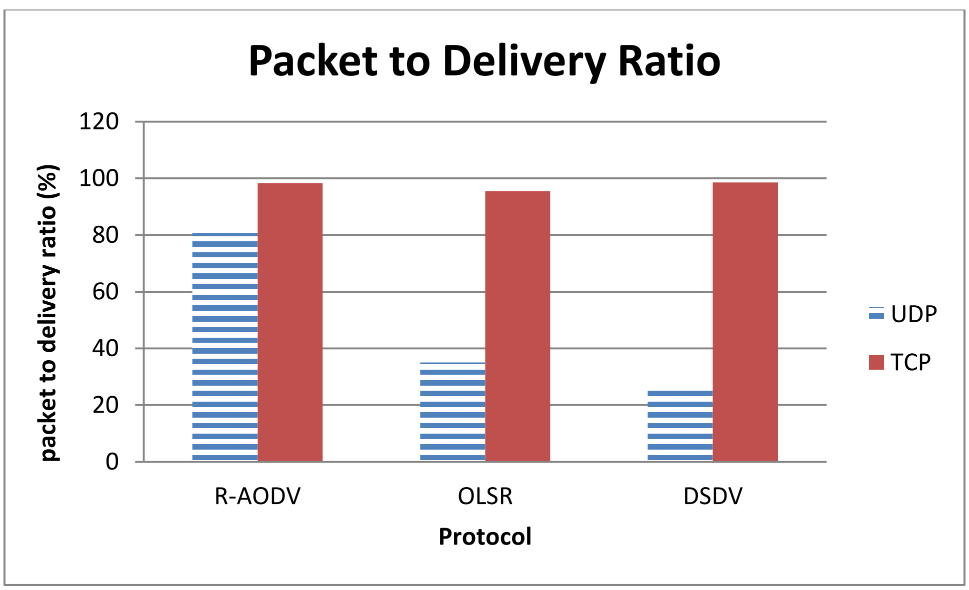

Figure 4.9 packet delivery ratio with using TCP and UDP

With TCP traffic all protocols have high PDR as compared to UDP traffic. Therefore, TCP/FTP is more reliable than UDP/CBR. Due to the TCP protocol is used there is a "guaranteed delivery”.

4.5 Summary

From the previous results is clear that DSDV has got higher performance in throughput, end to end delay and also in packet delivery ratio with TCP traffic. With UDP traffic the R-AODV has high performance in throughput and packet delivery ratio. While the DSDV and R-AODV have got less end to end delay beyond to them OLSR performance is better than the rest.

CHAPTER FIVE

Conclusion and Future Work

Chapter Five
Conclusion and Future Work

5.1 Conclusion

In the present scenario the performance of MANET routing protocols is examined with respect to the following parameters namely throughput, end-to-end delay and packet delivery ratio. DSDV and OLSR protocols come under proactive whereas R-AODV comes under reactive protocols. Every individual protocol has got its own advantages and disadvantages and performed well at their peer level, but for the purpose of efficiency when they are compared using the tool NS2 with the help of TCL scripts.

The various conclusions drawn from various experiments, observations, and analysis done in the thesis are as follows: Throughput: for TCP traffic, the network working with DSDV provides better efficiency when network size increases to 50 nodes. The throughput of DSDV is 685.11 kb/s while 626.86 kb/s in OLSR and 470.6 in R-AODV. For UDP traffic, R-AODV has more throughput as compared to OLSR and DSDV in different network sizes. Throughput in case of DSDV decreases from 233.87 kb/s to 119.63 kb/s with increasing number of nodes. Also in OLSR, throughput goes down from 290.08 kb/s to 153.68 kb/s. Packet Delivery Ratio (PDR): Although the PDR of DSDV has greater values than R-AODV and OLSR. It is around 99% when using TCP traffic. In UDP traffic, R-AODV has high PDR. PDR of all these protocols degrades with increasing size of network from 10 to 50 nodes except R-AODV increases with 50 nodes from 57.19 % with 40 nodes to 94.55 % with 50 nodes. Average End to End Delay: With TCP and UDP traffics, The DSDV offers lesser end to end delay than OLSR and R-AODV.

It can also be concluded from the simulation results that the efficiency of R-AODV and DSDV is better than OLSR. Generally the R-AODV, OLSR and DSDV work well for TCP traffic as compared to UDP type.

5.2 Future Work

A future study could be conducted on comparison the performance of these three protocols when the traffic generator is other than FTP and CBR like TELNET and HTTP because these traffic generators are the representatives of the traffic in the real scenario and expanding the study towards hybrid routing protocols, considering more metrics and more complex scenarios.

BIBLIOGRAPHY

[1] Pravanjan Das, Upena D. Dalal, "A Comparative Analysis of AODV and R-AODV Routing Protocols in MANETS," International Journal of Emerging Technology and Advanced Engineering, vol. 72, no. 21, June 2013.

[2] Francisco J. Ros. (2015, Jan) ―um-olsr-1.0.tgz‖, University of Murcia. [Online]. http://masimum.inf.um.es/ fjrm/development/um-olsr/ and http://sourceforge.net/ projects/um-olsr/, 2006

[3] Manish Singh Chaudhary, Varsha Singh, "Simulation and Study of AODV Routing Protocol under CBR and TCP Traffic Source," International Journal of Future Computer and Communication, vol. 3, no. 2, April 2014.

[4] Humaira Nishat, Vamsi Krishna K, D.Srinivasa Rao, Shakeel Ahmed, "Performance Evaluation of On Demand Routing Protocols AODV and Modified AODV (R-AODV) in MANETS," International Journal of Distributed and Parallel Systems (IJDPS), vol. 2, no. 1, Jan 2011.

[5] A. Zahary, A. Ayesh, "Analytical study to detect threshold number of efficient routes in multipath AODV extensions," in proceedings of International Conference of Computer Engineering & Systems, ICCES, 2007.

[6] S. Geetha, C. Karpagam, "Routing protocols in wireless ad-hoc networks: an overview," International Journal of Scientific Engineering and Applied Science (IJSEAS), vol. 2, no. 2, pp. 455-459, Feb 2016.

[7] Gatete Marcel,N. Vetrivelan, "Performance evaluation of four hybrid routing protocols for low and high density MANETs," International Journal of Computer Science & Engineering Technology (IJCSET), vol. 6, no. 4, pp. 183-188, Apr 2015.

[8] A.K. Verma, Tarunpreet Bhatia, "QoS Comparison of MANET Routing Protocols," I.J. Computer Network and Information Security, vol. 7, no. 9, pp.

64-73, August 2015.

[9] Vinay Somani, Ramprasad Kumawat, "Comparative Analysis of DSDV and OLSR Routing Protocols in MANET at Different Traffic Load," in International Conference on Computer Communication and Networks CSI-COMNET, 2011.

[10] Pramod Kumar, Mishra Bindeshwar, Singh Kushwaha, "Different Traffic Patterns Over Ad Hoc Network Routing Protocols," International Journal of Computer Applications, vol. 138, no. 11, March 2016.

[11] Umesh Barahdiya, Santosh Kumar Sharma, "Performance Analysis of Different Traffic sources TCP and CBR in AODV MANET," International Journal, vol. 1, no. 1, pp. 178-183, 2016.

[12] Anastasios A. Economidesa, Dimitra Kampitakia, "Simulation study of MANET routing protocols under FTP traffic," in Conference on Electronics, Telecommunications and Computers – CETC, 2013, pp. 231-238.

[13] P. R. Deshmukh, Sujata Wasudeorao Wankhade, "Comparison of AODV and RAODV Routing Protocols in Mobile Ad Hoc Networks," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 1, January 2013.

[14] Pravanjan Das, "Comparison of AODV and R-AODV Routing Protocols by varying Network Mobility, Network Area and Network Density," International Journal of Emerging Technology and Advanced Engineering, vol. 3, no. 8, August 2013.

[15] Kamlesh Gupta ,Ritika Sharma, "Comparison based Performance Analysis of UDP/CBR and TCP/FTP Traffic under AODV Routing Protocol in MANET," International Journal of Computer Applications, vol. 56, no. 15, October 2012.

[16] Azizol A, Ahmed F Nor. S, "Performance Evaluation of AODV, DSDV &

DSR Routing Protocol in Grid Environment," IJCSNS International Journal of Computer Science and Network Security, vol. 9, no. 7, July 2009.

[17] Sardar M. Bilal, Mazliza Othman, Attaur Rehman Khan, "A Performance Comparison of Network Simulators for Wireless Networks," July 2013.

[18] Mangesh M. Ghonge, Parag D. Thakare, P. M. Jawandhiya Suraj G. Gupta, "Open-Source Network Simulation Tools: An Overview," International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 2, no. 4, April 2013.

[19] (2010) Network Simulator 2. [Online]. www.isi.edu/nsnam/ns

Appendix I

R-AODV TCL File

```
set opt(chan)       Channel/WirelessChannel      ;# channel type
set opt(prop)       Propagation/TwoRayGround     ;# radio-propagation model
set opt(netif)      Phy/WirelessPhy              ;# network interface type
set opt(mac)        Mac/802_11                   ;# MAC type
set opt(ifq)        Queue/DropTail/PriQueue      ;# interface queue type
set opt(ll)         LL                           ;# link layer type
set opt(ant)        Antenna/OmniAntenna          ;# antenna model
set opt(ifqlen)     50                           ;# max packet in ifq
set opt(nn)         10                           ;# number of mobile nodes change 10-50
set opt(rp)         AODV                         ;# routing protocol
set opt(x)          800                          ;# X dimension of topography
set opt(y)          800                          ;# Y dimension of topography

set ns_ [new Simulator]
set tracefd [open raodv.tr w]
$ns_ trace-all $tracefd
set namtrace [open raodv.nam w]
$ns_ namtrace-all-wireless $namtrace  $opt(x) $opt(y)
set topo [new Topography]
$topo load_flatgrid $opt(x) $opt(y)
create-god $opt(nn)
set chan_1_ [new $opt(chan)]

$ns_ node-config -adhocRouting $opt(rp) \
                 -llType $opt(ll) \
                 -macType $opt(mac) \
                 -ifqType $opt(ifq) \
                 -ifqLen $opt(ifqlen) \
                 -antType $opt(ant) \
                 -propType $opt(prop) \
                 -phyType $opt(netif) \
                 -topoInstance $topo \
                 -agentTrace ON \
                 -routerTrace ON \
                 -macTrace ON \
                 -movementTrace ON \
                 -channel $chan_1_

set Server1 [$ns_ node]
set Server2 [$ns_ node]
set n2 [$ns_ node]
set n3 [$ns_ node]
```

```
set n4 [$ns_ node]
set n5 [$ns_ node]
set n6 [$ns_ node]
set n7 [$ns_ node]
set n8 [$ns_ node]
set n9 [$ns_ node]
set opt(seed) 0.1
set a [ns-random $opt(seed)]
set i 0
while {$i < 5} {incr i}

$Server1 set X_ 513.0
$Server1 set Y_ 517.0
$Server1 set Z_ 0.0
$Server2 set X_ 445.0
$Server2 set Y_ 474.0
$Server2 set Z_ 0.0
$n2 set X_ 36.0
$n2 set Y_ 529.0
$n2 set Z_ 0.0
$n3 set X_ 143.0
$n3 set Y_ 666.0
$n3 set Z_ 0.0
$n4 set X_ 201.0
$n4 set Y_ 552.0
$n4 set Z_ 0.0
$n5 set X_ 147.0
$n5 set Y_ 403.0
$n5 set Z_ 0.0
$n6 set X_ 230.0
$n6 set Y_ 291.0
$n6 set Z_ 0.0
$n7 set X_ 295.0
$n7 set Y_ 419.0
$n7 set Z_ 0.0
$n8 set X_ 363.0
$n8 set Y_ 335.0
$n8 set Z_ 0.0
$n9 set X_ 334.0
$n9 set Y_ 647.0
$n9 set Z_ 0.0
$ns_ at 0.75 "$n2 setdest 379.0 349.0 20.0"
$ns_ at 0.75 "$n3 setdest 556.0 302.0 20.0"
$ns_ at 0.20 "$n4 setdest 309.0 211.0 20.0"
$ns_ at 1.25 "$n5 setdest 179.0 333.0 20.0"
```

```
$ns_ at 0.75 "$n6 setdest 139.0 63.0 20.0"
$ns_ at 0.75 "$n7 setdest 320.0 27.0 20.0"
$ns_ at 1.50 "$n8 setdest 505.0 124.0 20.0"
$ns_ at 1.25 "$n9 setdest 274.0 487.0 20.0"

$ns_ initial_node_pos $Server1 125
$ns_ initial_node_pos $Server2 125
$ns_ initial_node_pos $n2 70
$ns_ initial_node_pos $n3 70
$ns_ initial_node_pos $n4 40
$ns_ initial_node_pos $n5 70
$ns_ initial_node_pos $n6 70
$ns_ initial_node_pos $n7 70
$ns_ initial_node_pos $n8 70
$ns_ initial_node_pos $n9 70

#Set 5 TCP connections
set tcp [new Agent/TCP]
set sink [new Agent/TCPSink]
$ns_ attach-agent $Server1 $tcp
$ns_ attach-agent $n5 $sink
$ns_ connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns_ at 10.0 "$ftp start"
set tcp [new Agent/TCP]
set sink [new Agent/TCPSink]
$ns_ attach-agent $n8 $tcp
$ns_ attach-agent $n4 $sink
$ns_ connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns_ at 10.0 "$ftp start"
set tcp [new Agent/TCP]
set sink [new Agent/TCPSink]
$ns_ attach-agent $Server2 $tcp
$ns_ attach-agent $n2 $sink
$ns_ connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns_ at 10.0 "$ftp start"
set tcp [new Agent/TCP]
set sink [new Agent/TCPSink]
$ns_ attach-agent $n4 $tcp
$ns_ attach-agent $n7 $sink
```

```
$ns_ connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns_ at 10.0 "$ftp start"
set tcp [new Agent/TCP]
set sink [new Agent/TCPSink]
$ns_ attach-agent $Server2 $tcp
$ns_ attach-agent $n3 $sink
$ns_ connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns_ at 10.0 "$ftp start"

# Set 5 UDP connections
set udp [new Agent/UDP]
$ns_ attach-agent $Server2 $udp
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
set null [new Agent/Null]
$ns_ attach-agent $n5 $null
$ns_ connect $udp $null
$cbr set packetSize_ 512
$cbr set rate_ 0.1Mb
$cbr set random_ false
$ns_ at 10.0 "$cbr start"
set udp [new Agent/UDP]
$ns_ attach-agent $Server1 $udp
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
set null [new Agent/Null]
$ns_ attach-agent $n3 $null
$ns_ connect $udp $null
$cbr set packetSize_ 512
$cbr set rate_ 0.1Mb
$cbr set random_ false
$ns_ at 10.0 "$cbr start"
set udp [new Agent/UDP]
$ns_ attach-agent $n5 $udp
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
set null [new Agent/Null]
$ns_ attach-agent $n9 $null
$ns_ connect $udp $null
$cbr set packetSize_ 512
$cbr set rate_ 0.1Mb
```

```
$cbr set random_ false
$ns_ at 10.0 "$cbr start"
set udp [new Agent/UDP]
$ns_ attach-agent $n7 $udp
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
set null [new Agent/Null]
$ns_ attach-agent $n9 $null
$ns_ connect $udp $null
$cbr set packetSize_ 512
$cbr set rate_ 0.1Mb
$cbr set random_ false
$ns_ at 10.0 "$cbr start"
set udp [new Agent/UDP]
$ns_ attach-agent $n3 $udp
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
set null [new Agent/Null]
$ns_ attach-agent $Server1 $null
$ns_ connect $udp $null
$cbr set packetSize_ 512
$cbr set rate_ 0.1M
$cbr set random_ false
$ns_ at 10.0 "$cbr start"

$ns_ at 0.0 "$ns_ trace-annotate \"mobile node movements\""
$ns_ at 4.1 "$ns_ trace-annotate \"node2 cache the data fro server\""
$ns_ at 4.59 "$ns_ trace-annotate \"packet loss at node27\""
$ns_ at 4.71 "$ns_ trace-annotate \"node1 cache the data\""

proc stop {} { global ns_ tracefd
                $ns_ flush-trace
                close $tracefd
                exec nam aodv.nam &
                exit 0}
puts "Starting Simulation........"
    $ns_ at 50.0 "stop"
    $ns_ run
```

Appendix II

OLSR Tcl File

```
set opt(chan)        Channel/WirelessChannel     ;# channel type
set opt(prop)        Propagation/TwoRayGround    ;# radio-propagation model
set opt(netif)       Phy/WirelessPhy             ;# network interface type
set opt(mac)         Mac/802_11                  ;# MAC type
set opt(ifq)         Queue/DropTail/PriQueue     ;# interface queue type
set opt(ll)          LL                          ;# link layer type
set opt(ant)         Antenna/OmniAntenna         ;# antenna model
set opt(ifqlen)      50                          ;# max packet in ifq
set opt(nn)          10                          ;# number of mobile nodes change 10-50
set opt(rp)          OLSR                        ;# routing protocol
set opt(cp)          " "                         ;# connection pattern file
set opt(sc)          " "                         ;# node movement file.
set opt(x)           800                         ;# x coordinate of topology
set opt(y)           800                         ;# y coordinate of topology
set opt(seed)        0.0                         ;# seed for random number gen.
set opt(stop)        50                          ;# time to stop simulation

if {$opt(seed) > 0} {
   puts "Seeding Random number generator with $opt(seed)\n"
   ns-random $opt(seed)
}
set ns_ [new Simulator]

Agent/OLSR set use_mac_    true
Agent/OLSR set debug_      false
Agent/OLSR set willingness 3
Agent/OLSR set hello_ival_ 2
Agent/OLSR set tc_ival_    5

set tracefd  [open output.tr w]
set namtrace [open output.nam w]
$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace $opt(x) $opt(y)
set topo [new Topography]
$topo load_flatgrid $opt(x) $opt(y)
create-god $opt(nn)
$ns_ node-config -adhocRouting $opt(adhocRouting) \
             -llType $opt(ll) \
             -macType $opt(mac) \
             -ifqType $opt(ifq) \
             -ifqLen $opt(ifqlen) \
             -antType $opt(ant) \
```

```
                 -propType $opt(prop) \
                 -phyType $opt(netif) \
                 -channelType $opt(chan) \
                 -topoInstance $topo \
                 -wiredRouting OFF \
                 -agentTrace ON \
                 -routerTrace ON \
                 -macTrace OFF

for {set i 0} {$i < $opt(nn)} {incr i} {
    set node_($i) [$ns_ node]        }

$node_(0) set X_ 5.0
$node_(0) set Y_ 5.0
$node_(0) set Z_ 0.0
$node_(1) set X_ 490.0
$node_(1) set Y_ 285.0
$node_(1) set Z_ 0.0
$node_(2) set X_ 150.0
$node_(2) set Y_ 240.0
$node_(2) set Z_ 0.0
$node_(3) set X_ 143.0
$node_(3) set Y_ 666.0
$node_(3) set Z_ 0.0
$node_(4) set X_ 201.0
$node_(4) set Y_ 552.0
$node_(4) set Z_ 0.0
$node_(5) set X_ 147.0
$node_(5) set Y_ 403.0
$node_(5) set Z_ 0.0
$node_(6) set X_ 230.0
$node_(6) set Y_ 291.0
$node_(6) set Z_ 0.0
$node_(7) set X_ 295.0
$node_(7) set Y_ 419.0
$node_(7) set Z_ 0.0
$node_(8) set X_ 363.0
$node_(8) set Y_ 335.0
$node_(8) set Z_ 0.0
$node_(9) set X_ 334.0
$node_(9) set Y_ 647.0
$node_(9) set Z_ 0.0

$ns_ at 1.0 "$node_(0) setdest 250.0 250.0 3.0"
$ns_ at 5.0 "$node_(1) setdest 45.0 285.0 5.0"
```

```
$ns_ at 2.0 "$node_(0) setdest 480.0 300.0 5.0"
$ns_ at 1.25 "$node_(5) setdest 179.0 333.0 20.0"
$ns_ at 0.75 "$node_(6) setdest 139.0 63.0 20.0"
$ns_ at 0.75 "$node_(7) setdest 320.0 27.0 20.0"
$ns_ at 1.50 "$node_(8) setdest 505.0 124.0 20.0"
$ns_ at 1.25 "$node_(9) setdest 274.0 487.0 20.0"

# set 5 UDP connections
set udp [new Agent/UDP]
$ns_ attach-agent $node_(0) $udp
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
set null [new Agent/Null]
$ns_ attach-agent $node_(5) $null
$ns_ connect $udp $null
$cbr set packetSize_ 512
$cbr set rate_ 0.1Mb
$cbr set random_ false
$ns_ at 10.0 "$cbr start"
set udp [new Agent/UDP]
$ns_ attach-agent $node_(1) $udp
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
set null [new Agent/Null]
$ns_ attach-agent $node_(3) $null
$ns_ connect $udp $null
$cbr set packetSize_ 512
$cbr set rate_ 0.1Mb
$cbr set random_ false
$ns_ at 10.0 "$cbr start"
set udp [new Agent/UDP]
$ns_ attach-agent $node_(5) $udp
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
set null [new Agent/Null]
$ns_ attach-agent $node_(2) $null
$ns_ connect $udp $null
$cbr set packetSize_ 512
$cbr set rate_ 0.1Mb
$cbr set random_ false
$ns_ at 10.0 "$cbr start"
set udp [new Agent/UDP]
$ns_ attach-agent $node_(7) $udp
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
```

```
set null [new Agent/Null]
$ns_ attach-agent $node_(9) $null
$ns_ connect $udp $null
$cbr set packetSize_ 512
$cbr set rate_ 0.1Mb
$cbr set random_ false
$ns_ at 10.0 "$cbr start"
set udp [new Agent/UDP]
$ns_ attach-agent $node_(3) $udp
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
set null [new Agent/Null]
$ns_ attach-agent $node_(0) $null
$ns_ connect $udp $null
$cbr set packetSize_ 512
$cbr set rate_ 0.1Mb
$cbr set random_ false
$ns_ at 10.0 "$cbr start"

# Set 5 TCP connections
set tcp [new Agent/TCP]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns_ attach-agent $node_(0) $tcp
$ns_ attach-agent $node_(1) $sink
$ns_ connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns_ at 10.0 "$ftp start"
set tcp [new Agent/TCP]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns_ attach-agent $node_(0) $tcp
$ns_ attach-agent $node_(4) $sink
$ns_ connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns_ at 10.0 "$ftp start"
set tcp [new Agent/TCP]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns_ attach-agent $node_(2) $tcp
$ns_ attach-agent $node_(1) $sink
$ns_ connect $tcp $sink
set ftp [new Application/FTP]
```

```
$ftp attach-agent $tcp
$ns_ at 10.0 "$ftp start"
set tcp [new Agent/TCP]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns_ attach-agent $node_(6) $tcp
$ns_ attach-agent $node_(0) $sink
$ns_ connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns_ at 10.0 "$ftp start"
set tcp [new Agent/TCP]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns_ attach-agent $node_(3) $tcp
$ns_ attach-agent $node_(7) $sink
$ns_ connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns_ at 10.0 "$ftp start"

$ns_ at 10.0 "[$node_(0) agent 255] print_rtable"
$ns_ at 15.0 "[$node_(0) agent 255] print_linkset"
$ns_ at 20.0 "[$node_(0) agent 255] print_nbset"
$ns_ at 25.0 "[$node_(0) agent 255] print_nb2hopset"
$ns_ at 30.0 "[$node_(0) agent 255] print_mprset"
$ns_ at 35.0 "[$node_(0) agent 255] print_mprselset"
$ns_ at 40.0 "[$node_(0) agent 255] print_topologyset"

if { $opt(cp) == "" } {
   puts "*** NOTE: no connection pattern specified."
   set opt(cp) "none"
} else {
   puts "Loading connection pattern..."
   source $opt(cp)
}
if { $opt(sc) == "" } {
   puts "*** NOTE: no scenario file specified."
   set opt(sc) "none"
} else {
   puts "Loading scenario file..."
   source $opt(sc)
   puts "Load complete..."
}
for {set i 0} {$i < $opt(nn)} {incr i} {
```

```
$ns_ initial_node_pos $node_($i) 20}
for {set i 0} {$i < $opt(nn) } {incr i} {
$ns_ at $opt(stop).0 "$node_($i) reset";}
$ns_ at $opt(stop).0002 "puts \"NS EXITING...\" ; $ns_ halt"
$ns_ at $opt(stop).0001 "stop"

proc stop {} {
   global ns_ tracefd namtrace
   $ns_ flush-trace
   close $tracefd
   close $namtrace }
puts "Starting Simulation..."
$ns_ run
```

Appendix III

DSDV TCL File

```
set val(chan)        Channel/WirelessChannel       ;# channel type
set val(prop)        Propagation/TwoRayGround     ;# radio-propagation model
set val(netif)       Phy/WirelessPhy              ;# network interface type
set val(mac)         Mac/802_11                   ;# MAC type
set val(ifq)         Queue/DropTail/PriQueue      ;# interface queue type
set val(ll)          LL                           ;# link layer type
set val(ant)         Antenna/OmniAntenna          ;# antenna model
set val(ifqlen)      50                           ;# max packet in ifq
set val(nn)          10                           ;# number of mobile nodes change 10-50
set val(rp)          DSDV                         ;# routing protocol
set val(x)           800                          ;# X dimension of topography
set val(y)           800                          ;# Y dimension of topography
set val(stop)        50                           ;# time of simulation end

set ns          [new Simulator]
set tracefd     [open dsdv.tr w]
set namtrace    [open dsdv.nam w]
$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
set topo     [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

$ns node-config -adhocRouting $val(rp) \
                -llType $val(ll) \
                -macType $val(mac) \
                -ifqType $val(ifq) \
                -ifqLen $val(ifqlen) \
                -antType $val(ant) \
                -propType $val(prop) \
                -phyType $val(netif) \
                -channelType $val(chan) \
                -topoInstance $topo \
                -agentTrace ON \
                -routerTrace ON \
                -macTrace OFF \
                -movementTrace ON

    for {set i 0} {$i < $val(nn) } { incr i } {
        set node_($i) [$ns node]        }
```

```
$node_(0) set X_ 5.0
$node_(0) set Y_ 5.0
$node_(0) set Z_ 0.0
$node_(1) set X_ 490.0
$node_(1) set Y_ 285.0
$node_(1) set Z_ 0.0
$node_(2) set X_ 150.0
$node_(2) set Y_ 240.0
$node_(2) set Z_ 0.0
$node_(3) set X_ 143.0
$node_(3) set Y_ 666.0
$node_(3) set Z_ 0.0
$node_(4) set X_ 201.0
$node_(4) set Y_ 552.0
$node_(4) set Z_ 0.0
$node_(5) set X_ 147.0
$node_(5) set Y_ 403.0
$node_(5) set Z_ 0.0
$node_(6) set X_ 230.0
$node_(6) set Y_ 291.0
$node_(6) set Z_ 0.0
$node_(7) set X_ 295.0
$node_(7) set Y_ 419.0
$node_(7) set Z_ 0.0
$node_(8) set X_ 363.0
$node_(8) set Y_ 335.0
$node_(8) set Z_ 0.0
$node_(9) set X_ 334.0
$node_(9) set Y_ 647.0
$node_(9) set Z_ 0.0

$ns at 1.0 "$node_(0) setdest 250.0 250.0 3.0"
$ns at 5.0 "$node_(1) setdest 45.0 285.0 5.0"
$ns at 2.0 "$node_(0) setdest 480.0 300.0 5.0"
$ns at 1.25 "$node_(5) setdest 179.0 333.0 20.0"
$ns at 0.75 "$node_(6) setdest 139.0 63.0 20.0"
$ns at 0.75 "$node_(7) setdest 320.0 27.0 20.0"
$ns at 1.50 "$node_(8) setdest 505.0 124.0 20.0"
$ns at 1.25 "$node_(9) setdest 274.0 487.0 20.0"

# Set 5 TCP connections
set tcp [new Agent/TCP]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
```

```
$ns attach-agent $node_(1) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 10.0 "$ftp start"
set tcp [new Agent/TCP]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
$ns attach-agent $nod_(4) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 10.0 "$ftp start"
set tcp [new Agent/TCP]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(2) $tcp
$ns attach-agent $node_(1) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 10.0 "$ftp start"
set tcp [new Agent/TCP]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(6) $tcp
$ns attach-agent $node_(0) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 10.0 "$ftp start"
set tcp [new Agent/TCP]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(3) $tcp
$ns attach-agent $node_(7) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 10.0 "$ftp start"

# Set 5 UDP connections
set udp [new Agent/UDP]
$ns attach-agent $node_(0) $udp
```

```
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
set null [new Agent/Null]
$ns attach-agent $node_(5) $null
$ns connect $udp $null
$cbr set packetSize_ 512
$cbr set rate_ 0.1Mb
$cbr set random_ false
$ns at 10.0 "$cbr start"
set udp [new Agent/UDP]
$ns attach-agent $node_(1) $udp
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
set null [new Agent/Null]
$ns attach-agent $node_(3) $null
$ns connect $udp $null
$cbr set packetSize_ 512
$cbr set rate_ 0.1Mb
$cbr set random_ false
$ns at 10.0 "$cbr start"
set udp [new Agent/UDP]
$ns attach-agent $node_(5) $udp
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
et null [new Agent/Null]
$ns attach-agent $node_(2) $null
$ns connect $udp $null
$cbr set packetSize_ 512
$cbr set rate_ 0.1Mb
$cbr set random_ false
$ns at 10.0 "$cbr start"
set udp [new Agent/UDP]
$ns attach-agent $node_(7) $udp
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
set null [new Agent/Null]
$ns attach-agent $node_(9) $null
$ns connect $udp $null
$cbr set packetSize_ 512
$cbr set rate_ 0.1Mb
$cbr set random_ false
$ns at 10.0 "$cbr start"

set udp [new Agent/UDP]
$ns attach-agent $node_(3) $udp
```

```
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
set null [new Agent/Null]
$ns attach-agent $node_(0) $null
$ns connect $udp $null
$cbr set packetSize_ 512
$cbr set rate_ 0.1Mb
$cbr set random_ false
$ns at 10.0 "$cbr start"
for {set i 0} {$i < $val(nn)} { incr i } {

$ns initial_node_pos $node_($i) 20
}
for {set i 0} {$i < $val(nn) } { incr i } {
   $ns at $val(stop) "$node_($i) reset";}

$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "stop"
$ns at 50.01 "puts \"end simulation\" ; $ns halt"

proc stop {} {
   global ns tracefd namtrace
   $ns flush-trace
   close $tracefd
   close $namtrace
}
$ns run
```