Performance Analysis of Routing Protocols of Mobile Ad hoc Networks (MANETs)

By

CJUO AanishMumtaz

CSM Salman Khan

GC NabeelMasood

CQMS Qaiser Lohani

Supervisor: Maj. Dr. M Faisal Khan

Co Supervisor: Lec. Ubaidullah Khalid

Submitted to the Faculty of Electrical Engineering National University of Sciences and Technology, Rawalpindi in partial fulfillment for the requirements of a B.E Degree in Telecommunication Engineering

JULY 2010

CERTIFICATE

Certified that the contents and form of project report, entitled ***“Performance Analysis of Routing Protocols of MobileAdhoc Networks (MANETs)”*** submitted by 1) CJUO Aanish Mumtaz 2) CSM Salman Khan 3) GC NabeelMasood and 4) CQMS QaiserLohani, have been found satisfactory for the requirement of the degree.

Co -Supervisor: **____________________**

Lec.Obaidullah

Supervisor: _____________________

Maj.Dr. M.Faisal Khan

ABSTRACT

Mobile Ad-hoc Networks (MANETs) are future wireless networks consisting entirely of mobile devices that communicate on the move without any fixed infrastructure. They use multi-hop peer-to-peer routing instead of static network infrastructure to provide network connectivity. These are wireless infrastructure-less networks in which the devices can move and communicate randomly without any backbone. Devices in these networks will both generate user traffic as well as application traffic to carry out network control and routing protocols. MANETs have applications in rapidly deployed and dynamic military and civilian systems. Our aim is to compare various protocols of MANETs in different scenarios and select a suitable protocol basing on the performance parameters of each protocol in these scenarios.

DECLARATION

We declare that this thesis entitled "Performance Analysis of Routing Protocols of MobileAdhoc Networks (MANETs)"is the result of our own work except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

DEDICATION

All our efforts are dedicated to our beloved parents and teachers who have beena constant source of encouragement for us throughout our lives. May AllahAlmighty bless them with long lives and always provide us their loving andthorough guidance.

ACKNOWLEDGEMENTS

First, we praise Allah Almighty, who gave us the strength and courage to undertake this project and to complete it in a timely and beneficent manner.

We wish to express our gratitude to our supervisor; Dr. Faisal Khan and Co Supervisor Lec. Ubaidullah from the Faculty of Electrical Engineering, Military College of Signals, National University of Sciences and Technology, Rawalpindi, for their continuous, untiring support and supervision during the course of the project. It is due to their brilliant ideas and expertise that led this project to its successful outcome. Not to forget their efforts in providing us with a nice environment to work on our project. Their effort to provide us with the necessary equipment and his persistent guidance is certainly commendable.

Many thanks to Kashif Saleem from Faculty of Electrical Engineering, University Technology Malaysia (UTM), Skudai, Johor. MALAYSIA, for supporting us during the course of the project.

I would also like to acknowledge the struggle of our parents for our well-being and Education. They have done enormous and endless efforts to make us standout. We would designate our success to them.

Last but not the least, we express our gratitude for Dr. Faisal Bashir of CS Department, Military College of Signals, for their helpful pieces of advice on Ubuntu and MANETs.

TABLE OF CONTENTS

List of Figures

Chapter 1-Introduction

1.1 Mobile AD hoc Networks (MANET)

A Mobile Ad hoc Network (MANET) [1] is a kind of wireless self-configuring network of mobile routers (and associated hosts) connected by wireless links the union of which forms an arbitrary topology. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology changes rapidly and unpredictably. Such a network operates in a standalone fashion.

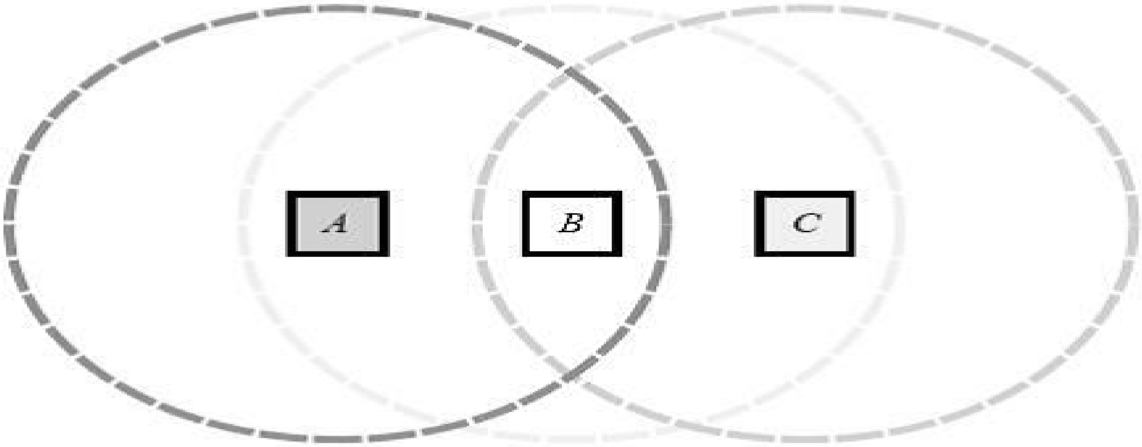


Figure 1.1-Three devices each within the range of one another [3]

In areas, in which there is little or no communication infra-structure or the existing infrastructure is expensive or inconvenient to use, wireless mobile users can communicate through the formation of an Ad hoc Network. In such a network, each mobile node operates not only as a host but also as a router, forwarding packets for other mobile nodes in the network that may not be within direct wireless transmission range of each other. Each node participates in an ad hoc network in discovering paths through the network to any other node. The idea of ad hoc networking is sometimes also called infrastructure less network, since the mobile nodes in the network dynamically establish routing among themselves to form their own network on the fly.

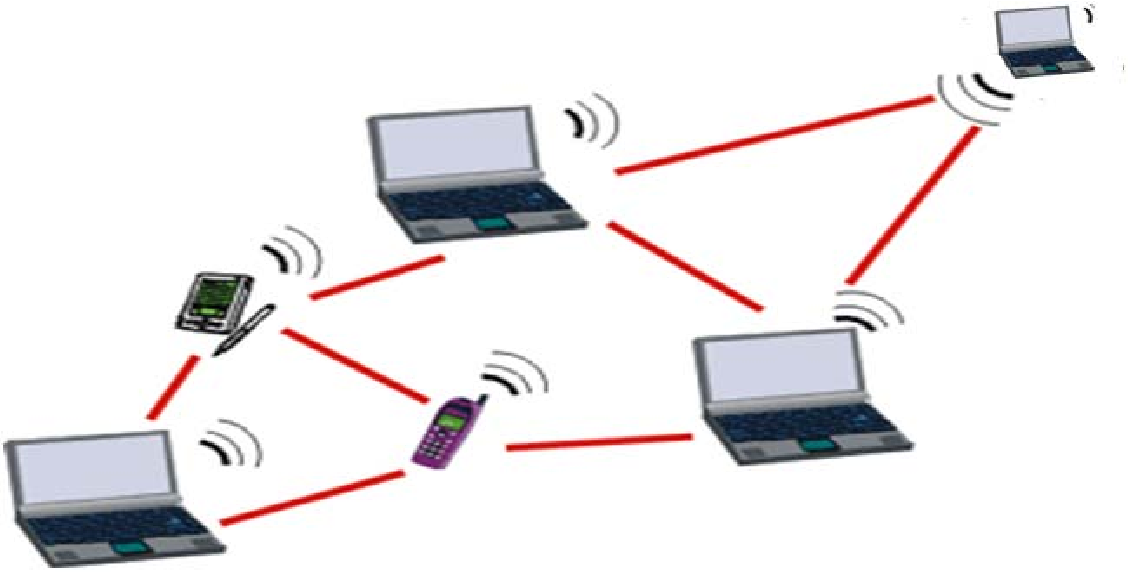


Figure 1.2-A simple Mobile Ad hoc Network [17]

1.2 Difference between MANETs and WSN

1.2.1 Differences

Although WSNs and MANETs seem to be similar, yet there are also fundamental differences.

1. MANETs are considered [2] “close” to humans in the sense that most nodes in the network are devices that are meant to be used by human beings. (e.g. laptop computers, PDAs, mobile radio terminals, etc.). On the contrary, sensor networks do not focus on human interaction but instead focus on interaction with the environment. Indeed, nodes in a sensor network are usually embedded in the environment to sense some phenomenon.

2. WSN can be seen as a special type of MANET, but the protocols, algorithms and design issues of WSN cannot be applied to the MANET unchanged. Terms such as unicast and multicast common in MANETs, are hardly applicable in WSN where we find other forms of routing such as one-to-many, many-to-one, many-to-many, etc.
3. There is a security issue in WSN since all the data is flowing to the sink from all the sensor nodes so the sink can easily be tracked by sensing the network and flow of data. If the sink is down due to any reason, the whole communication gets disrupted. But in MANETs if an intermediate node is down, protocol will find out a new path to the destination.
4. Opposite to ad hoc networks where the IEEE 802.11 radio interface has become a defector standard for communications (and its underlying physical-layer settings), IEEE 802.15.4 standard is an effort to set a radio standard for general-purpose sensor deployments. This radio is aimed at low-power low–range communications devices that may allow for years of battery-life without replacement.
5. The IEEE 802.11 uses Channel Sense Multiple Access (CSMA) with collision avoidance (CA). The Energy Efficient MAC protocol for WSNs (SMAC) is based on a listen/sleep cycle specifically designed for WSN. In SMAC, a sensor node transmits SYNC packets carrying the node’s listen/sleep schedule so that other nodes know exactly when they can communicate with it. SMAC schedules communications without the need for a local or global synchronization entity.
6. MANETs traditionally implement the full TCP/IP protocol stack, meaning MANET nodes will have IP addresses or something similar, support broadcast, unicast and multicast routing, and more important, be fully compatible with UDP/TCP transport protocols. But bringing TCP/IP to wireless sensor networks is a difficult task, however due to following reasons:
    a. First, because of their limited physical size and low cost, sensors are severely constrained in terms of memory and processing power. Traditionally, these constraints have been considered too limiting for a sensor to be able to use the TCP/IP protocols.

   b. Second, the harsh communication conditions make TCP/IP perform poorly in terms of both throughput and energy efficiency. Sensor networks may exhibit higher packets losses (2% to 30%) compared to ad hoc networks.

7. In IP-based networks, there is the concept of the network address used to identify nodes and endpoint communication entities (MANETs are usually IP based). Contrary to these node-centric networks, sensor networks are data-centric, since the identifier of individual nodes is not as important as the data gathered by sets of nodes.

1.3 Issues in MANETs

If there are, only two nodes that want to communicate with each other and are located very closely to each other [3], then no specific routing protocols or routing decisions are necessary. Routing protocols come into play only when more than two nodes wish to communicate with each other. In this scenario, critical decisions like the optimal route from source to destination are made since this is very important because often, the mobile nodes operate on some kind of battery power. Thus, it becomes necessary to transfer the data with the minimal delay to waste less power. There may also be some kind of compression involved, which might be provided by the protocol to waste less bandwidth. Further, there is also a need of some type of encryption to protect the data from prying eyes. In addition to this, Quality of Service is also a requirement so that the least packet drop is obtained.

The other factors, which have been considered while choosing a protocol for MANETs, are as follows:

1.3.1 Multicasting: This is the ability to send packets to multiple nodes at once. This is similar to broadcasting except the fact that broadcasting is done to all the nodes in the network. This is important as it takes less time to transfer data to multiple nodes.

1.3.2 Loop Free: A path taken by a packet never transits the same intermediate node twice before it arrives at the destination. To improve the overall, we want the routing protocol to guarantee that the routes supplied are loop-free. This avoids any waste of bandwidth or CPU consumption.

1.3.3 Multiple routes: If one route gets broken due to some disaster, then the data could be sent through some other route. Thus the protocol should allow creating multiple routes.

1.3.4 Distributed Operation: The protocol should not be dependent on a centralized node and have a decentralized architecture.

1.3.5 Reactive: It means that the routes are discovered between a source and destination only when the need arises to send data. Some protocols are reactive while others are proactive which means that the route is discovered to various nodes without waiting for the need.

1.3.6 Unidirectional Link Support: The radio environment can cause the formation of unidirectional links. Utilization of these links and not only the bi-directional links improves the routing protocol performance.

1.3.7 Power Conservation: The nodes in an ad-hoc network can be laptops and thin clients, such as PDAs that are very limited in battery power and therefore use some sort of stand-by mode to save power. It is therefore important that the routing protocol support these sleep-modes.

There are various routing protocols available for MANETs. The most popular ones are DSR, AODV and DSDV.

1.4 Examples

Some examples [4] of the possible uses of ad hoc networking include

- students using laptop computers to participate in an interactive lecture
- business associates sitting in their offices and interacting
- sharing information during a meeting
- soldiers relaying information
- situational awareness on the battlefield
- Emergency disaster relief personnel coordinating efforts after a hurricane or earthquake.

Many different protocols have been proposed to solve the multihop routing problem in ad hoc networks, each based on different assumptions and intuitions. However, little is known about the actual performance of these protocols, and no attempt has previously been made to directly compare them in a realistic manner.

1.5 Simulation Environment

NS is a discrete event simulator developed by the University of California at Berkeley and the VINT project. The simulations have been done in Ns-2. Ns is explained in detail in chapter 4.

Chapter 2 -Protocols

2.1 Protocol

Rules and regulations are always needed for communication:

- In Human communication, rules play an important role, for example, if two persons are speaking at the same time, neither of them will be audible.
- Same is the case for communication devices. They also require a set of rules for efficient communication.
- The set of rules and regulations is called a Protocol.

2.2 Types of protocols

There are two types of routing protocols [5]

- Proactive protocols
- Reactive protocols.

2.2.1 Proactive protocols

Proactive protocols periodically send control packets in the network to update routing tables. But as the network becomes more and more dynamic these control packets increase network congestion.

2.2.2 Reactive protocols

Reactive protocols send control packets for route discovery upon demand from the sender. Tables are not established before communication.

The performance of both proactive protocols as well as reactive protocols degrades when network becomes highly dynamic due to movement of nodes. This results in increment of packet delay and network congestion. Four typical routing protocols of ad hoc networks are DSDV [6] ,AODV[7], DSR[8] and OLSR[9].

2.3 AODV (Reactive)

AODV [10] is essentially a combination of both DSR and DSDV. It borrows the basic on-demand mechanism of Route Discovery and Route Maintenance from DSR, plus the use of hop by-hop routing, sequence numbers, and periodic beacons from DSDV.

2.3.1 Basic Mechanisms

When a node S as shown in the figure 2.1, needs a route to some destination D [11], it broadcasts a Route request message to its neighbors, including the last known sequence number for that destination. The Route request is flooded in a controlled manner through the network until it reaches a node that has a route to the destination. Each node that forwards the Route request creates a reverse route for itself back to node S.

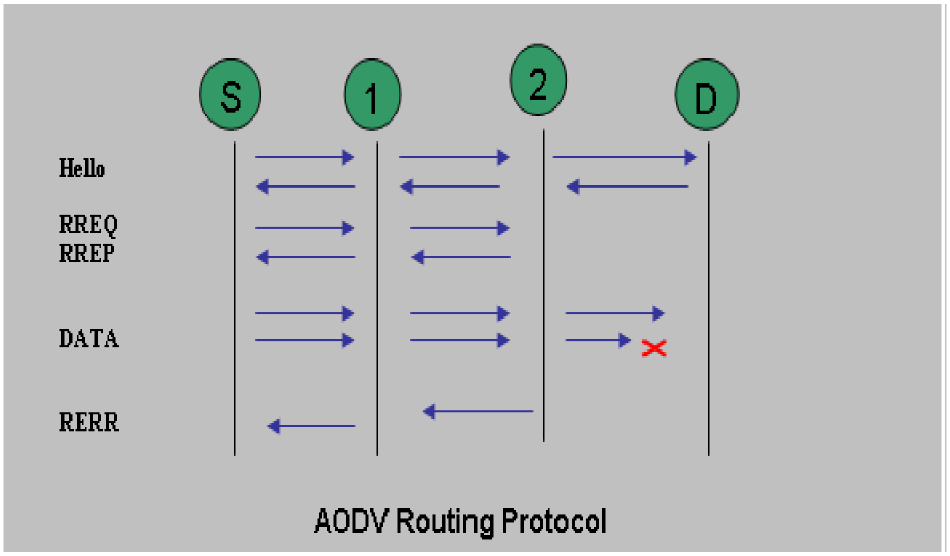


Figure 2.1- Basics workflow mechanism of AODV [11]

When the Route request reaches anode with a route to D, that node generates a Route reply that contains the number of hops necessary to reach D and the sequence number for D most recently seen by the node generating the REPLY. Each node that participates in forwarding this REPLY back toward the originator of the Route (node S), creates forward route to D. The state created in each node along the path from S to D is hop by hop state that is, each node remembers only the next hop and not the entire route, as would be done in source routing.

In order to maintain routes, AODV normally requires that each node periodically transmit a hello message, with a defect rate of once per second Failure to receive three consecutive HELLO messages from a neighbor is taken as an indication that the fink to the neighbor in question is down. Alternatively, the AODV specification briefly suggests that anode may use physical layer or link layer methods to detect link breakages to nodes that it considers neighbors.

When it goes down, any upstream node that has recently forwarded packets to a destination using that link is notified via an unsolicited Route containing an infinite metric for that destination. Upon receipt of such a Route reply, a node must acquire a new route to the destination using Route Discovery.

2.3.2 Flooding for Control Packet Delivery

Sender S broadcasts a control packet P to all its neighbors. Each node receiving P forwards P to its neighbors. Sequence numbers help to avoid the possibility of forwarding the same packet more than once. Packet P reaches destination D provided that D is reachable from sender S. Node D does not forward the packet. Many protocols perform flooding of control packets e.g. AODV & DSR, instead of data packets .The control packets are used to discover routes Discovered routes are subsequently used to send data packet(s) Overhead of control packet flooding is amortized over data packets transmitted between consecutive control packet floods.

2.3.3 Advantages

- Simplicity
- Good in smaller networks when devices are in range of each other
- Best when link failures are minimum

2.3.4 Disadvantages

- Very high overhead (enormous HELLO messages)

- Data packets may be delivered to too many nodes who do not need to receive them
- Potentially lower reliability of data delivery
- Flooding uses broadcasting -- hard to implement reliable broadcast delivery without significantly increasing overhead.

2.3.5 Link Failure Detection

- Hello messages: Neighboring nodes periodically exchange hello message
- Absence of hello message is used as an indication of link failure

2.4 DSDV(Destination Sequenced Distance Vector) – Proactive

Destination Sequenced Distance Vector protocols (DSDV) [12] is based on Bellman-Ford shortest path algorithm. Each node has a table, which contains the shortest path to every other node in the network. These tables are constantly updated and forwarded to other nodes in the network whenever a change is detected. When a node receives an update it can either update the tables or hold it for a while in order to select shortest route. Figure2.2 shows an example where node 1 is the source and node 15 is the destination. The routing table of node 1, Table shows that the shortest route to the receiving node is through node 8 while the distance to it is 4 hops. When broken link is detected, the end node initiates a table update. The update message has “infinity” assigned to it and sequence number for that destination. When a node receives a message with infinity weight it quickly forwards it to neighboring nodes.

For example if node 11 moves to different location and communication is lost between 14 and 10, node 10 detects broken link and sets the broken link between them to infinity. The “infinity message” starts the table update process.

Each update might increase or decrease the number of hops between any two nodes. In this example, the distance between 1 and 14 has increased from 3 to 5 hops.

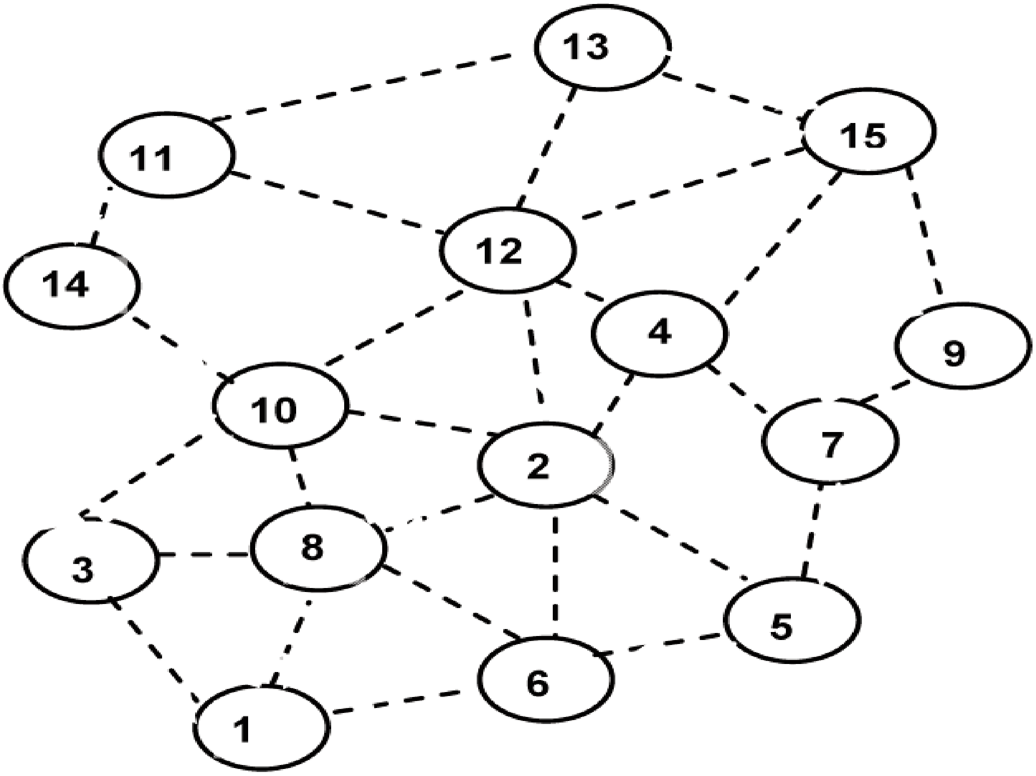


Figure 2.2- Route Establishment in DSDV [12]

Dest	Next Node	Dist	Seq No
2	8	2	12
3	3	1	23
4	8	3	37
5	6	2	49
6	6	1	67
7	6	3	111
8	8	1	128
9	6	4	134
10	8	2	155
11	8	4	167
12	8	3	170
13	8	4	173
14	8	3	182
15	8	4	185

Table 1- Routing table for node 1 [12]

Incremental update let wireless network be easily incorporated. The update creates lots of traffic and slows the network down. Nodes need to wait for a table update message by the same destination node creating delays. The sequence number tags are used to prevent the formation of loops, to counter the count-to infinity problem and for faster convergence.

The Destination Sequence Distance Vector (DSDV) is a proactive unicast mobile ad hoc network routing protocol. DSDV is based on the traditional Bellman-Ford algorithm. However, their mechanisms to improve routing performance in mobile ad hoc networks are quite different. In routing tables of DSDV, an entry stores the next hop towards a destination, the cost metric for the routing path to the destination and a destination sequence number that is created by the destination. Sequence numbers are used in DSDV to distinguish stale routes from fresh ones and avoid formation of route loops.

The route updates of DSDV can be either time-driven or event-driven. Every node periodically transmits updates including its routing information to its immediate neighbors. While a significant change occurs from the last update, a node can transmit its changed routing table in an event-triggered style. Moreover, the DSDV has two ways when sending routing table updates. One is "full dump" update type and the full routing table is included inside the update. A “full dump” update could span many packets. An incremental update contains only those entries that with metric have been changed since the last update is sent. Additionally, the incremental update fits in one packet.

Each entry in the routing table contains a sequence number, the sequence numbers are generally even if a link is present; else, an odd number is used. The number is generated by the destination, and the emitter needs to send out the next update with this number. Routing information is distributed among nodes by sending full dumps infrequently and smaller incremental updates more frequently.

2.4.1 Selection of Route:

If a router receives new information, then it uses the latest sequence number. If the sequence number is the same as the one already in the table, the route with the better metric is used. Stale entries are those entries that have not been updated for a while. Such entries as well as the routes using those nodes as next hops are deleted.

2.5 DSR (Dynamic Source Routing) – Reactive

The DSR [13] protocol is composed of two main mechanisms that work together to allow the discovery and maintenance of source routes in the ad hoc network:

- Route Discovery is the mechanism by which a node S wishing to send a packet to a destination node D obtains a source route to D. Route Discovery is used only when S attempts to send a packet to D and does not already know a route to D.

- Route Maintenance is the mechanism by which node S is able to detect, while using a source route to D, if the network topology has changed such that it can no longer use its route to D because a link along the route no longer works. When Route Maintenance indicates a source route is broken, S can attempt use any other route it happens to know to D, or can invoke Route Discovery again to find a new route for subsequent packets to D. Route Maintenance for this route is used only when S is actually sending packets to D. In DSR, Route Discovery and Route Maintenance each operate entirely “on demand". In particular, unlike other protocols, DSR requires no periodic packets of any kind at any layer within the network. For example, DSR does not use any periodic routing advertisement, link status sensing, or neighbor detection packets, and does not rely on these functions from any underlying protocols in the network. This entirely on-demand behavior and lack of periodic activity allows the number of overhead packets caused by DSR to scale all the way down to zero, when all nodes are approximately stationary with respect to each other and all routes needed for current communication have already been discovered. As nodes begin to move more or as communication patterns change, the routing packet overhead of DSR automatically scales to only that needed to track the routes currently in use. Network topology changes not affecting routes currently in use are ignored and do not cause reaction from the protocol.

2.5.1 Multiple Route discovery

In response to a single Route Discovery (as well as through routing information from other packets overheard), a node may learn and cache multiple routes to any destination. This support for multiple routes allows the reaction to routing changes to be much more rapid, since a node with multiple routes to a destination can try another cached route if the one it has been using should fail. This caching of multiple routes also avoids the overhead of needing to perform a new Route Discovery each time a route in use breaks. The sender of a packet selects and controls the route used for its own packets, which together with support for multiple routes also allows features such as load balancing to be defined. In addition, all routes used are easily guaranteed to be loop-free, since the sender can avoid duplicate hops in the routes selected.

2.5.2 Advantages

- Multiple routes
- Good in networks having high mobility

2.6 OLSR (Optimized Link State Routing) - Proactive

The protocol [9] is an optimization of the classical link state algorithm tailored to the requirements of a mobile wireless LAN. The key concept used in the protocol is that of multipoint relays (MPRs). MPRs are selected nodes which forward broadcast messages during the flooding process. This technique substantially reduces the message overhead as compared to a classical flooding mechanism, where every node retransmits each message when it receives the first copy of the message. In OLSR, link state information is generated only by nodes elected as MPRs. Thus, a second optimization is achieved by minimizing the number of control messages flooded in the network. As a third optimization, an MPR node may chose to report only links between itself and its MPR selectors. Hence, as contrary to the classic link state algorithm, partial link state information is distributed in the network. This information is then used for route calculation. OLSR provides optimal routes (in terms of number of hops). The protocol is particularly suitable for large and dense networks as the technique of MPRs works well in this context.

The Optimized Link State Routing Protocol (OLSR) is developed for mobile ad hoc networks. It operates as a table driven, proactive protocol, i.e., exchanges topology information with other nodes of the network regularly. Each node selects a set of its neighbor nodes as "multipoint relays" (MPR). In OLSR, only nodes, selected as such MPRs are responsible for forwarding control traffic, intended for diffusion into the entire network. MPRs provide an efficient mechanism for flooding control traffic by reducing the number of transmissions required.

Nodes, selected as MPRs, also have a special responsibility when declaring link state information in the network. Indeed, the only requirement for OLSR to provide shortest path routes to all destinations is that MPR nodes declare link-state information for their MPR selectors. Additional available link-state information may be utilized, e.g., for redundancy.

Nodes, which have been selected as multipoint relays by some neighbor node(s) announce this information periodically in their control messages. Thereby a node announces to the network, that it has reach ability to the nodes which have selected it as an MPR. In route calculation, the MPRs are used to form the route from a given node to any destination in the network. Furthermore, the protocol uses the MPRs to facilitate efficient flooding of control messages in the network.

A node selects MPRs from among its one hop neighbors with "symmetric", i.e., bi-directional, linkages. Therefore, selecting the route through MPRs automatically avoids the problems associated with data packet transfer over uni-directional links (such as the problem of not getting link-layer acknowledgments for data packets at each hop, for link-layers employing this technique for unicast traffic).

OLSR is developed to work independently from other protocols. Likewise, OLSR makes no assumptions about the underlying link-layer.

2.6.1 Main address

The main address of a node, which will be used in OLSR control traffic as the "originator address" of all messages emitted by this node. It is the address of one of the OLSR interfaces of the node. A single OLSR interface node must use the address of its only OLSR interface as the main address.

2.6.2 Neighbor node

A node X is a neighbor node of node Y if node Y can hear node X (i.e., a link exists between an OLSR interface on node X and an OLSR interface on Y).

2.6.3 2-hop neighbor

A node heard by a neighbor.

2.6.4 Multipoint relay (MPR)

A node which is selected by its 1-hop neighbor, node X, to "re-transmit" all the broadcast messages that it receives from X, provided that the message is not a duplicate, and that the time to live field of the message is greater than one.

2.6.5 Multipoint relay selector (MPR selector, MS)

A node, which has selected its 1-hop neighbor, node X, as its multipoint relay, will be called a multipoint relay selector of node X.

2.6.6 Protocol Overview

OLSR is a proactive routing protocol for mobile ad hoc networks. The protocol inherits the stability of a link state algorithm and has the advantage of having routes immediately available when needed due to its proactive nature. OLSR is an optimization over the classical link state protocol, tailored for mobile ad hoc networks.

OLSR minimizes the overhead from flooding of control traffic by using only selected nodes, called MPRs, to retransmit control messages. This technique significantly reduces the number of retransmissions required to flood a message to all nodes in the network. Secondly, OLSR requires only partial link state to be flooded in order to provide shortest path routes. The minimal set of link state information required is, that all nodes, selected as MPRs, MUST declare the links to their MPR selectors. Additional topological information, if present, MAY be utilized e.g., for redundancy purposes.

OLSR MAY optimize the reactivity to topological changes by reducing the maximum time interval for periodic control message transmission. Furthermore, as OLSR continuously maintains routes to all destinations in the network, the protocol is beneficial for traffic patterns where a large subset of nodes are communicating with another large subset of nodes, and where the [source, destination] pairs are changing over time. The protocol is particularly suited for large and dense networks, as the optimization done using MPRs works well in this context. The larger and more dense a network, the more optimization can be achieved as compared to the classic link state algorithm.

OLSR is designed to work in a completely distributed manner and does not depend on any central entity. The protocol does NOT REQUIRE reliable transmission of control messages: each node sends control messages periodically, and can therefore sustain a reasonable loss of some such messages. Such losses occur frequently in radio networks due to collisions or other transmission problems.

Also, OLSR does not require sequenced delivery of messages. Each control message contains a sequence number which is incremented for each message. Thus the recipient of a control message can, if required, easily identify which information is more recent - even if messages have been re-ordered while in transmission.

Furthermore, OLSR provides support for protocol extensions such as sleep mode operation, multicast-routing etc. Such extensions may be introduced as additions to the protocol without breaking backwards compatibility with earlier versions.

OLSR does not require any changes to the format of IP packets. Thus any existing IP stack can be used as is: the protocol only interacts with routing table management.

2.6.7 Multipoint Relays

The idea of multipoint relays is to minimize the overhead of flooding messages in the network by reducing redundant retransmissions in the same region. Each node in the network selects a set of nodes in its symmetric 1-hop neighborhood, which may retransmit its messages. This set of selected neighbor nodes is called the "Multipoint Relay" (MPR) set of that node. The neighbors of node N which are not in its MPR set, receive and process broadcast messages but do not retransmit broadcast messages received from node N.

Each node selects its MPR set from among its 1-hop symmetric neighbors. This set is selected such that it covers (in terms of radio range) all symmetric strict 2-hop nodes. The MPR set of N, denoted as MPR (N), is then an arbitrary subset of the symmetric 1-hop neighborhood of N, which satisfies the following condition: every node in the symmetric strict 2-hop neighborhood of N must have a symmetric link towards MPR (N). The smaller a MPR set, the less control traffic overhead results from the routing protocol. [14] gives an analysis and example of MPR selection algorithms.

Each node maintains information about the set of neighbors that have selected it as MPR. This set is called the "Multipoint Relay Selector set" (MPR selector set) of a node. A node obtains this information from periodic HELLO messages received from the neighbors.

A broadcast message, intended to be diffused in the whole network, coming from any of the MPR selectors of node N is assumed to be retransmitted by node N, if N has not received it yet. This set can change over time (i.e., when a node selects another MPR-set) and is indicated by the selector nodes in their HELLO messages.

2.6.8 MPR Selection and MPR Signaling

The objective of MPR selection is for a node to select a subset of its neighbors such that all nodes 2 hops away will receive a broadcast message, retransmitted by these selected neighbors. The MPR set of a node is computed such that it, for each interface, satisfies this condition. The information required to perform this calculation is acquired through the periodic exchange of HELLO messages.

2.6.9 Topology Control Message Diffusion

Topology Control messages are diffused with the purpose of providing each node in the network with sufficient link-state information to allow route calculation. Topology Control messages are diffused.

2.6.10 Route Calculation

Given the link state information acquired through periodic message exchange, as well as the interface configuration of the nodes, the routing table for each node can be computed.

Chapter 3 - Network Simulator (NS)

NS [15] is a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. NS is popularly used in the simulation of routing and multicast protocols, among others, and is heavily used in ad-hoc networking research.

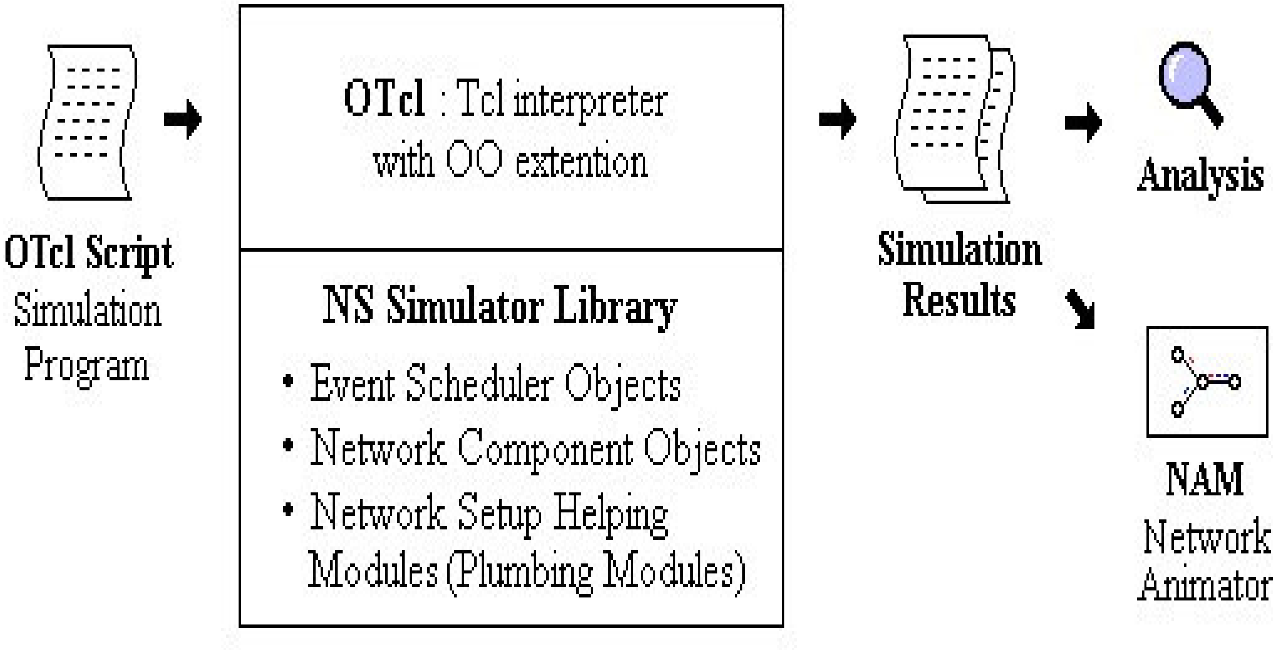

OTcl Script
Simulation Program
OTcl : Tcl interpreter with OO extention
NS Simulator Library
• Event Scheduler Objects
• Network Component Objects
• Network Setup Helping Modules (Plumbing Modules)
Simulation Results
Analysis
NAM
Network Animator

Figure 3.1-Simplified User's View of NS-2 [14]

Ns began as a variant of the REAL network simulator in 1989 and has evolved substantially over the past few years. In 1995 ns development was supported by DARPA through the VINT project at LBL, Xerox PARC, UCB, and USC/ISI. Currently ns development is support through DARPA with SAMAN and through NSF with CONSER, both in collaboration with other researchers including ACIRI. Ns has always included substantial contributions from other researchers, including wireless code from the UCB Daedelus and CMU Monarch projects and Sun Microsystems.

3.1 Design

NS [14] was built in C++ and provides a simulation interface through OTcl, an object-oriented dialect of Tcl. The user describes a network topology by writing OTcl scripts, and then the main NS program simulates that topology with specified parameters.

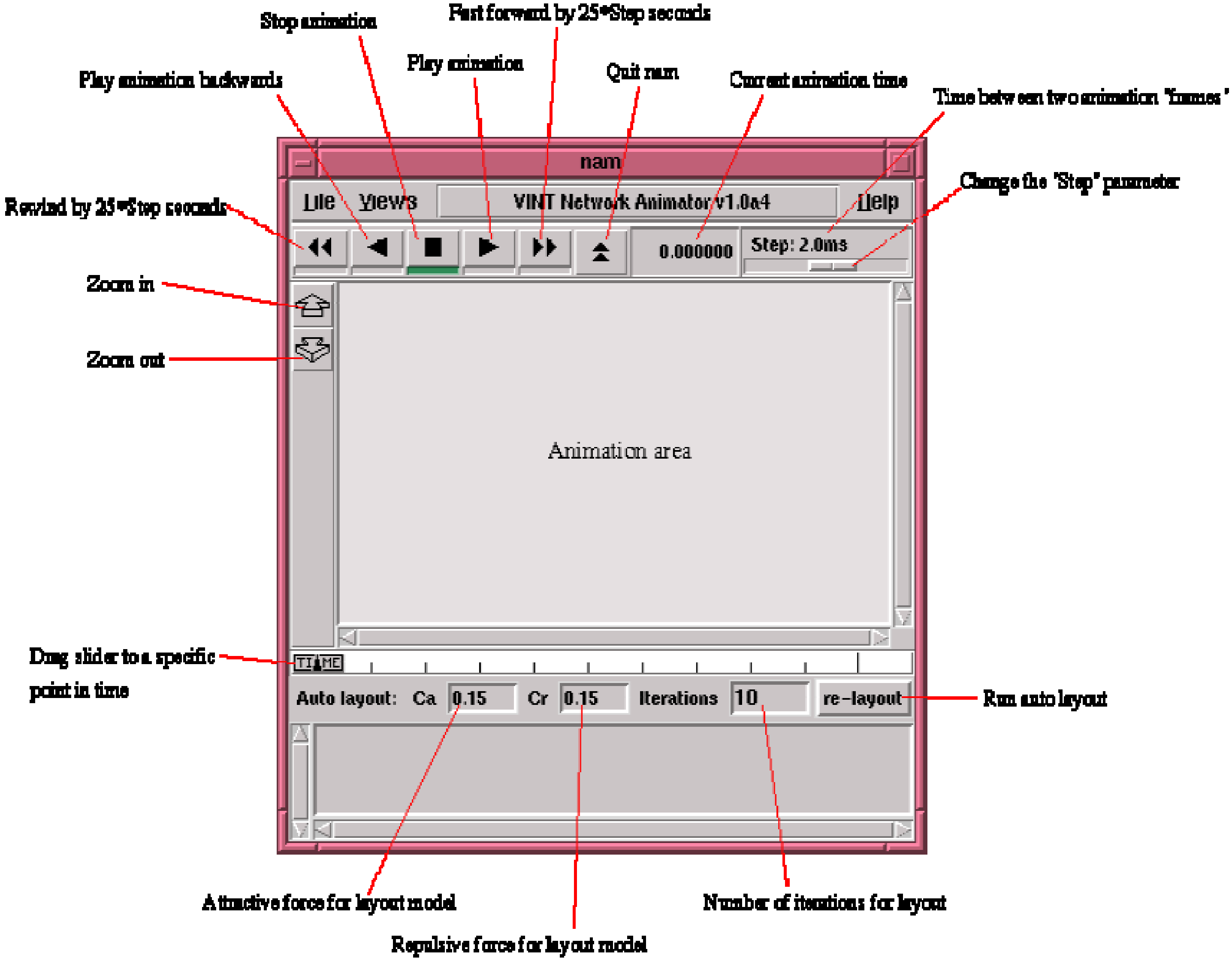
3.1.1 NS-3

Generation 3 of ns has begun development as of July 1, 2006 and was projected to take four years. It is funded by the institutes like University of Washington, Georgia Institute of Technology and the ICSI Center for Internet Research with collaborative support from the Planète research group at INRIA Sophia-Antipolis. Currently ns-3 is in development phase. It is an event based network simulator written in C++ and Python.

3.2 NAM

Network protocol designers face difficult tasks, including simultaneously monitoring state in a potentially large number of nodes, understanding and analyzing complex message exchanges, and characterizing dynamic interactions with competing traffic. Researchers have explored network protocol visualization in many contexts, beginning with static protocol graphs and visualization of large-scale traffic, and more recently including simulation visualizations and editors. Network visualization tools allow designers to take in large amounts of information quickly, visually identify patterns in communication, and better understand causality and interaction.

Nam is a network animator that provides packet-level animation, protocol graphs, traditional time-event plots of protocol actions, and scenario editing capabilities. Nam benefits from a close relationship with ns, which can collect detailed protocol information from a simulation. The authors describe how nam uses preprocessing to visualize data taken directly from real network traces. They also feel that visualization tools such as nam make protocol design and debugging easier.


Stop animation
Fast forward by 25*Step seconds
Play animation
Quit nam
Current animation time
Time between two animation 'frames'
Play animation backwards
Change the 'Step' parameter
Rewind by 25*Step seconds
Zoom in
Zoom out
nam
File
Views
VINT Network Animator v1.0a4
Help
0.000000
Animation area
Drag slider to a specific point in time
TIME
re-layout
Run auto layout
Attractive force for layout model
Number of iterations for layout
Repulsive force for layout model

Step: 2.0ms

Auto layout: Ca 0.15 Cr 0.15 Iterations 10

Figure 3.2- Figure showing the NAM tool description [15]

To use NS-2, a user programs in the OTcl script language.

An OTcl script will do the following.

•Initiates an event scheduler.

•Sets up the network topology using the network objects.

•Tells traffic sources when to start/stop transmitting packets through the event scheduler.

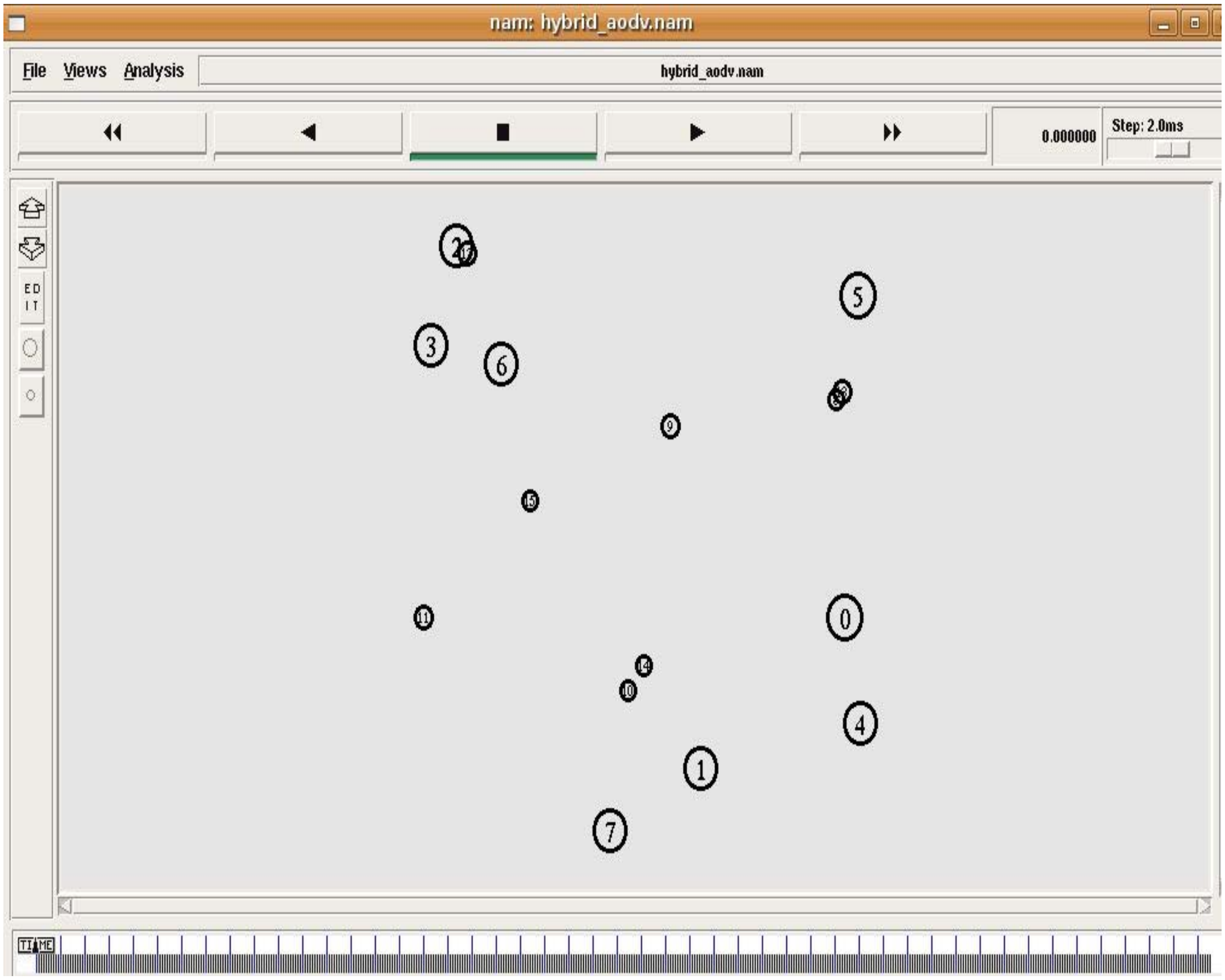


Figure 3.3- A practical NAM window

3.2.1 Uses

Provides a visual interpretation of the network created

· Can be executed directly from a Tcl script

· Controls include play, stop ff, rw, pause, a display speed controller and a packet

Monitor facility.

· Presents information such as throughput, number of packets on each link.

· Provides a drag and drop interface for creating topologies.

3.3 Tracing

A Trace file contains all information needed for animation purposes- both on a static network layout and on dynamic events such as packet arrivals, departures, drops and link failures.

To trace packets on all links

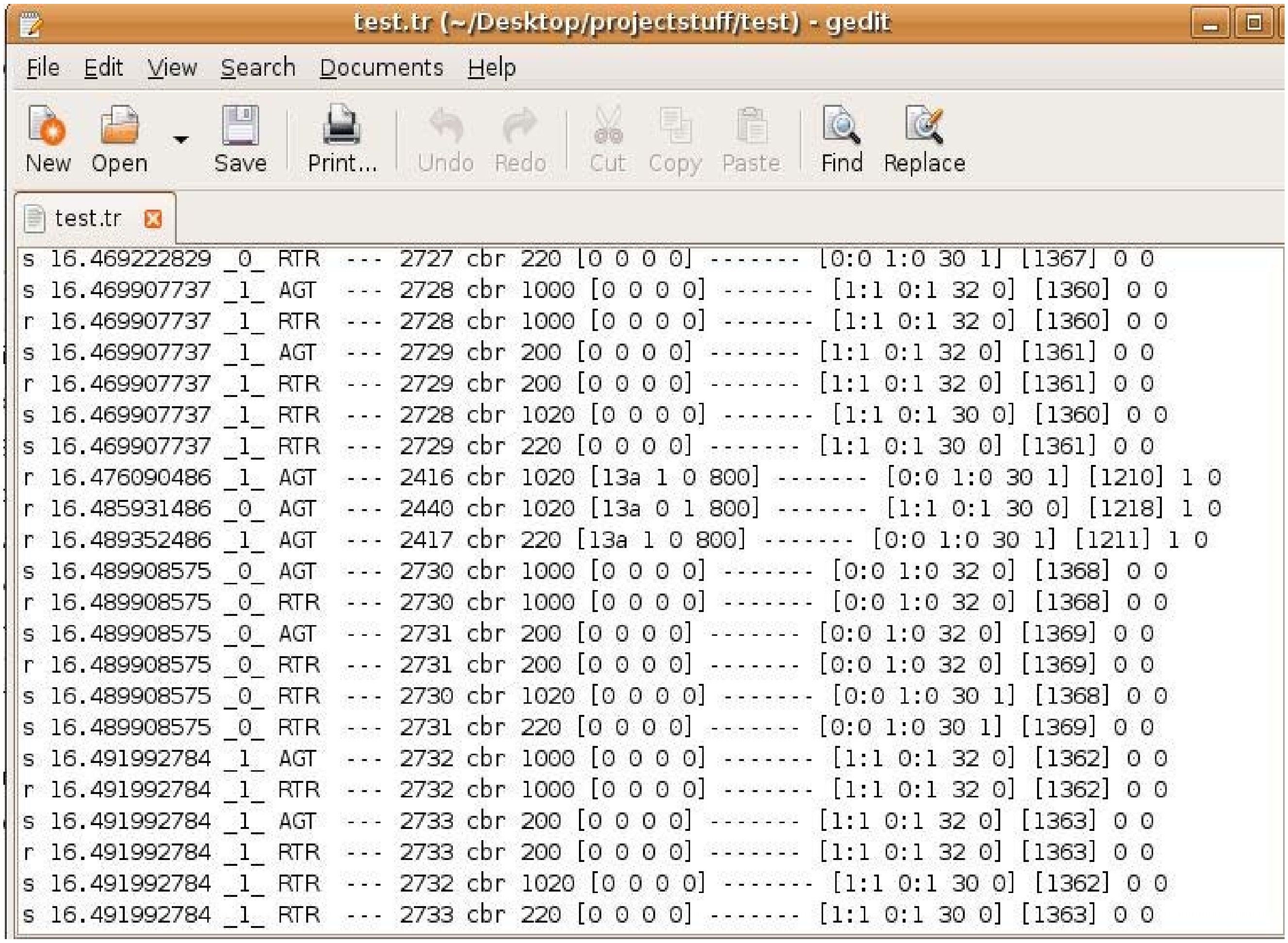
· set trace_file [open out.tr w]

· $ns trace-all $trace_file

· $ns flush-trace

· close $trace_file

An example of a standard trace file in NS-2 follows and its format is shown in Figure 3.4.

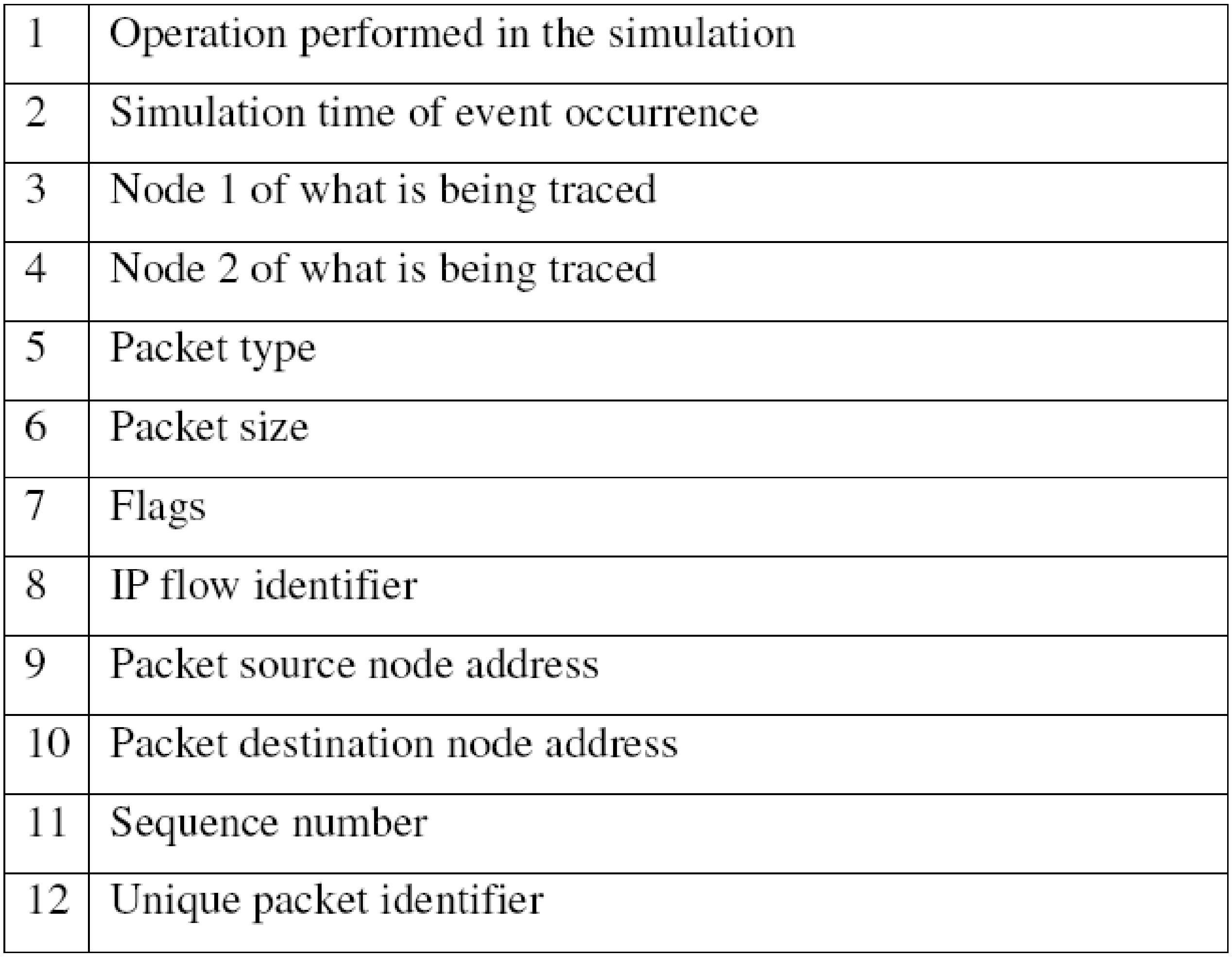


```
s 16.469222829 _0_ RTR  --- 2727 cbr 220 [0 0 0 0] ------- [0:0 1:0 30 1] [1367] 0 0
s 16.469907737 _1_ AGT  --- 2728 cbr 1000 [0 0 0 0] ------- [1:1 0:1 32 0] [1360] 0 0
r 16.469907737 _1_ RTR  --- 2728 cbr 1000 [0 0 0 0] ------- [1:1 0:1 32 0] [1360] 0 0
s 16.469907737 _1_ AGT  --- 2729 cbr 200 [0 0 0 0] ------- [1:1 0:1 32 0] [1361] 0 0
r 16.469907737 _1_ RTR  --- 2729 cbr 200 [0 0 0 0] ------- [1:1 0:1 32 0] [1361] 0 0
s 16.469907737 _1_ RTR  --- 2728 cbr 1020 [0 0 0 0] ------- [1:1 0:1 30 0] [1360] 0 0
s 16.469907737 _1_ RTR  --- 2729 cbr 220 [0 0 0 0] ------- [1:1 0:1 30 0] [1361] 0 0
r 16.476090486 _1_ AGT  --- 2416 cbr 1020 [13a 1 0 800] ------- [0:0 1:0 30 1] [1210] 1 0
r 16.485931486 _0_ AGT  --- 2440 cbr 1020 [13a 0 1 800] ------- [1:1 0:1 30 0] [1218] 1 0
r 16.489352486 _1_ AGT  --- 2417 cbr 220 [13a 1 0 800] ------- [0:0 1:0 30 1] [1211] 1 0
s 16.489908575 _0_ AGT  --- 2730 cbr 1000 [0 0 0 0] ------- [0:0 1:0 32 0] [1368] 0 0
r 16.489908575 _0_ RTR  --- 2730 cbr 1000 [0 0 0 0] ------- [0:0 1:0 32 0] [1368] 0 0
s 16.489908575 _0_ AGT  --- 2731 cbr 200 [0 0 0 0] ------- [0:0 1:0 32 0] [1369] 0 0
r 16.489908575 _0_ RTR  --- 2731 cbr 200 [0 0 0 0] ------- [0:0 1:0 32 0] [1369] 0 0
s 16.489908575 _0_ RTR  --- 2730 cbr 1020 [0 0 0 0] ------- [0:0 1:0 30 1] [1368] 0 0
s 16.489908575 _0_ RTR  --- 2731 cbr 220 [0 0 0 0] ------- [0:0 1:0 30 1] [1369] 0 0
s 16.491992784 _1_ AGT  --- 2732 cbr 1000 [0 0 0 0] ------- [1:1 0:1 32 0] [1362] 0 0
r 16.491992784 _1_ RTR  --- 2732 cbr 1000 [0 0 0 0] ------- [1:1 0:1 32 0] [1362] 0 0
s 16.491992784 _1_ AGT  --- 2733 cbr 200 [0 0 0 0] ------- [1:1 0:1 32 0] [1363] 0 0
r 16.491992784 _1_ RTR  --- 2733 cbr 200 [0 0 0 0] ------- [1:1 0:1 32 0] [1363] 0 0
s 16.491992784 _1_ RTR  --- 2732 cbr 1020 [0 0 0 0] ------- [1:1 0:1 30 0] [1362] 0 0
s 16.491992784 _1_ RTR  --- 2733 cbr 220 [0 0 0 0] ------- [1:1 0:1 30 0] [1363] 0 0
```

Figure 3.4- Trace file

3.3.1 Explanation of TRACE file

Now the explanation of trace file is:



1	Operation performed in the simulation
2	Simulation time of event occurrence
3	Node 1 of what is being traced
4	Node 2 of what is being traced
5	Packet type
6	Packet size
7	Flags
8	IP flow identifier
9	Packet source node address
10	Packet destination node address
11	Sequence number
12	Unique packet identifier

Table 2- Table showing the contents of Trace file [15]

3.3.2 Advantages:

- sometimes cheaper
- find bugs (in design) in advance
- generality: over analytic/numerical techniques
- detail: can simulate system details at arbitrary level

Chapter 4 - Scenarios

A simulation life cycle can be divided into three phases: specification of the scenario, simulation of the events and analysis of the results of the simulation. In everyday life, we come across enormous scenarios.

4.1 General scenarios

- which can have different ranges of transmission
- different transmission times, different area sizes
- different situations, static or mobile
- hybrid (both at the same time), real on ground base scenarios
- ideal scenarios

4.1.2 Practical life Scenarios

In the practical life, the above- explained scenarios can practical be defined as:

- students using laptop computers to participate in an interactive lecture(few static users few mobile)
- sharing information during a meeting
- soldiers relaying information(mobile scenario)
- situational awareness on the battlefield (hybrid scenario)
- emergency disaster relief personnel coordinating efforts after a hurricane or earthquake.(hybrid scenario, both mobile and static users)

4.2 Scenarios implemented in our work:

From police and fire department dispatches to aircraft communication, two-way radio communication works to ensure public safety. This medium becomes particularly important in emergency situations. We have chalked out four scenarios which encompass almost every situation that can be confronted. Lots of battles were fought in built up areas. The gist of these battles was the streets were killing zones, and the fighting was in the buildings, and involved lots of close combats. This scenario could be played in virtually any theatre with any troops. Similarly peace keeping missions, rescue missions and business communication all can have different scenarios.

4.2.1 On foot

Usually in the practical life, on foot personals are in communication. Our first scenario is on foot in which any node can move randomly in any direction. The pause time has been specified and also the maximum speed limit is incorporated. All the above mentioned four protocols have been implemented and then the analysis has been done.

These can be applied to a situation where soldiers are performing a mission in a battlefield, disaster management teams on foot and rescue workers etc.

Key features

Parameters	Values (Units)
Simulation Time	1000 s
Area	500 square meters
Maximum Pause time	2 seconds
Maximum speed of moving devices	3 m/s
Number of devices	16

Table 3- Key features of on foot scenario

After the implementation of these scenarios, all the protocols were applied and then the results were analyzed in detail. Than the best protocol was selected that suits the best in on foot scenario.

4.2.2 Vehicular (City)

Safe navigation [16] support through wireless car to car and car to curb communications has become an important priority for car manufacturers as well as transportation authorities and communications standards organizations. These emerging applications span many fields, from office on- wheels to entertainment, crime investigation, and civic defense, maintain peace and order etc.

In urban cities, the roads and turns are mostly diagonal and at right angles to each other. The roads are built. Urban areas may be cities, towns or conurbations, but the term is not commonly extended to rural settlements such as villages and hamlets. So in this scenario, our nodes will move in diagonal, straight and to 90 degrees angle.

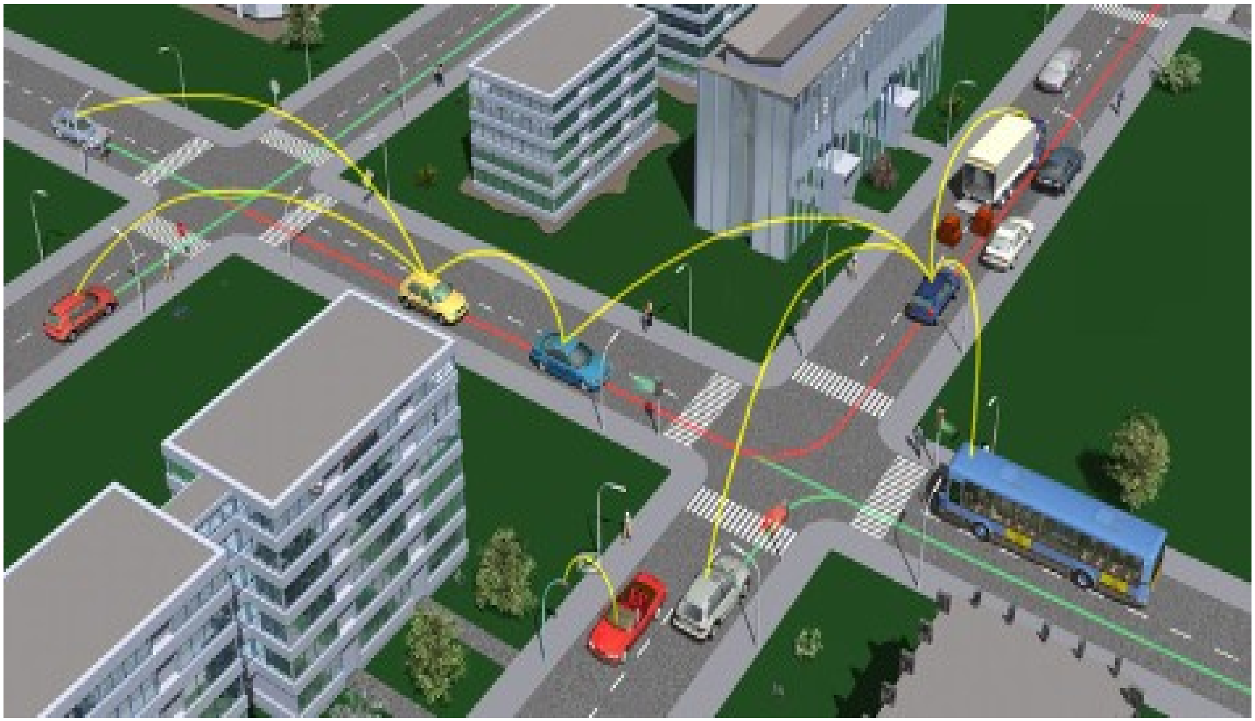


Figure 4.1- Vehicular Ad hoc network [18]

Key features

Parameters	Values (Units)
Simulation Time	900 s
Area	4 square kilometers
Maximum Pause time	15 seconds
Maximum speed of moving devices	14 m/s
Number of devices	10
Street Length	300 meters

Table 4- Key features of Vehicular (Urban) scenario

All the five protocols were applied and there trace files were studied in detail. Than keeping in view the protocol functions, the delays and throughputs were checked and a suitable protocol is selected.

4.2.4 Hybrid Scenario

The most practical scenario in the real world is the hybrid scenario in which both vehicles and on foot communication is taking place for many desired reasons. Patrols assigned area in vehicle and on foot; performs active campus community problem identification; attend campus neighborhood meetings; maintains high patrol visibility to assist in crime prevention; actively performs routine beat patrol, concentrating on high incident areas, to detect possible criminal activities or needs for service; regularly checks businesses, university property, and residential areas; monitors radio broadcasts by Communications and other officers to ensure awareness of activities in area and to provide assistance, if needed; identifies, reports, and responds to suspicious activities or needs for service.

In hybrid scenarios duties relating to service and assistance (lost child, arguments, injured persons, walk-away, lock-outs, prowlers, abandoned vehicles, dog bites, civil law disputes, alarms, vehicle inspections, etc.); responds to scene through radio runs, notification, or observation; evaluates situation to determine needs assistance from others, other agency contact, ambulance, etc.); identifies and implements appropriate course of action.

Hence in our scenario, the movement of nodes has variable speeds, different pause times, random turns and both on foot and vehicular nodes communicate with each other.

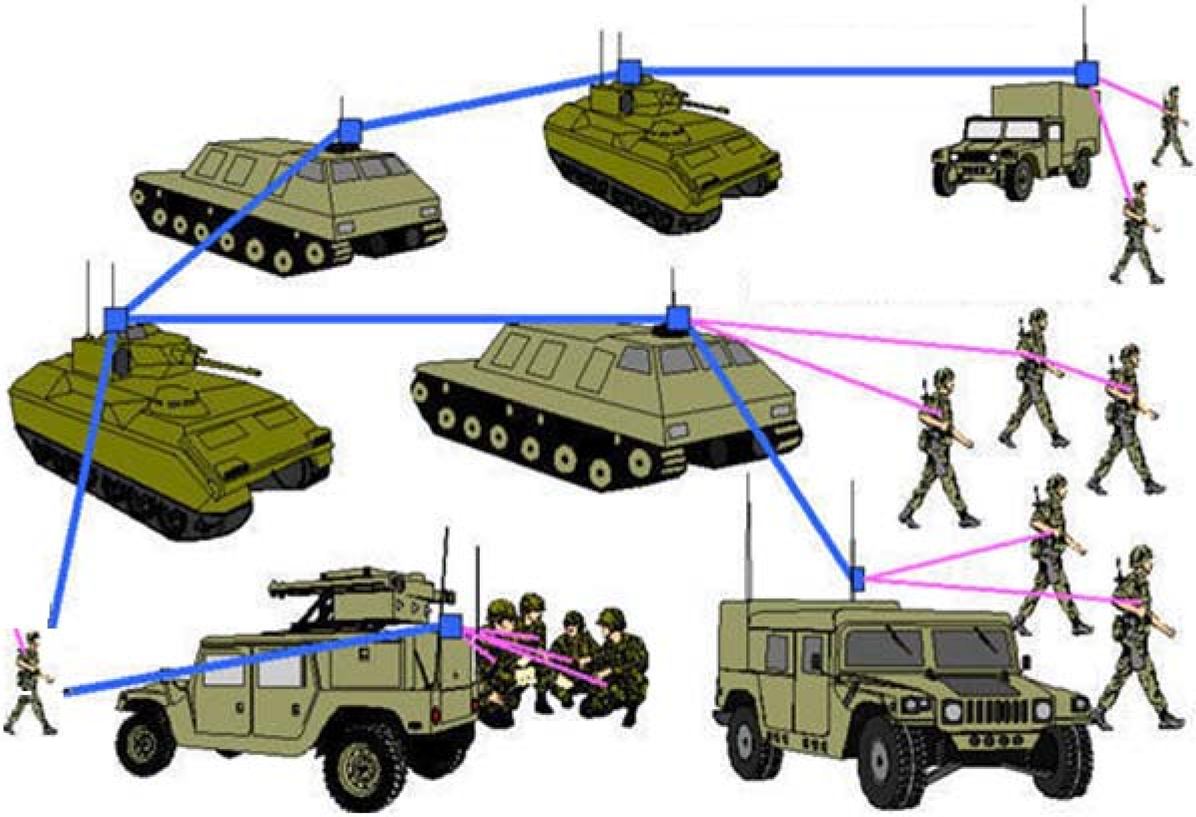


Figure 4.2- Hybrid Ad hoc network [19]

Key feature:

Parameters	Values (Units)
Simulation Time	900 s
Area	1500 square meters
Maximum Pause time on foot	2 seconds
Maximum pause time of vehicles	15 seconds
Maximum speed of moving devices on foot	8 m/s

Maximum speed of moving devices on foot	3 m/s
Number of on foot devices	8
Number of Vehicular devices	8

Table 5- key features of Hybrid scenario

The trace files were studied in detail after applying all the four protocols. Throughput and latency was checked keeping in view the functioning of the protocols and the best protocol was selected.

4.3 Traffic Behaviors

4.3.1 CBR (Constant Bit Rate)

A CBR [13] traffic object generates traffic according to a deterministic rate. Packets are a constant size.

4.3.2 Variable bit Rate (VBR)

VBR randomly change rate or bursts depending on parameters set. The traffic type used in the analysis of these situations is VBR, since it depicts the real world scenario in a more realistic way. The packet sizes change variably in between the defined limits.

4.4 Poisson’s Distribution

The packet size and the time intervals are following the Poisson’s distribution. the Poisson distribution is a discrete probability distribution that expresses the probability of a number of events occurring in a fixed period of time if these events occur with a known average rate and independently of the time since the last event. (The Poisson distribution can also be used for the number of events in other specified intervals such as distance, area or volume.)

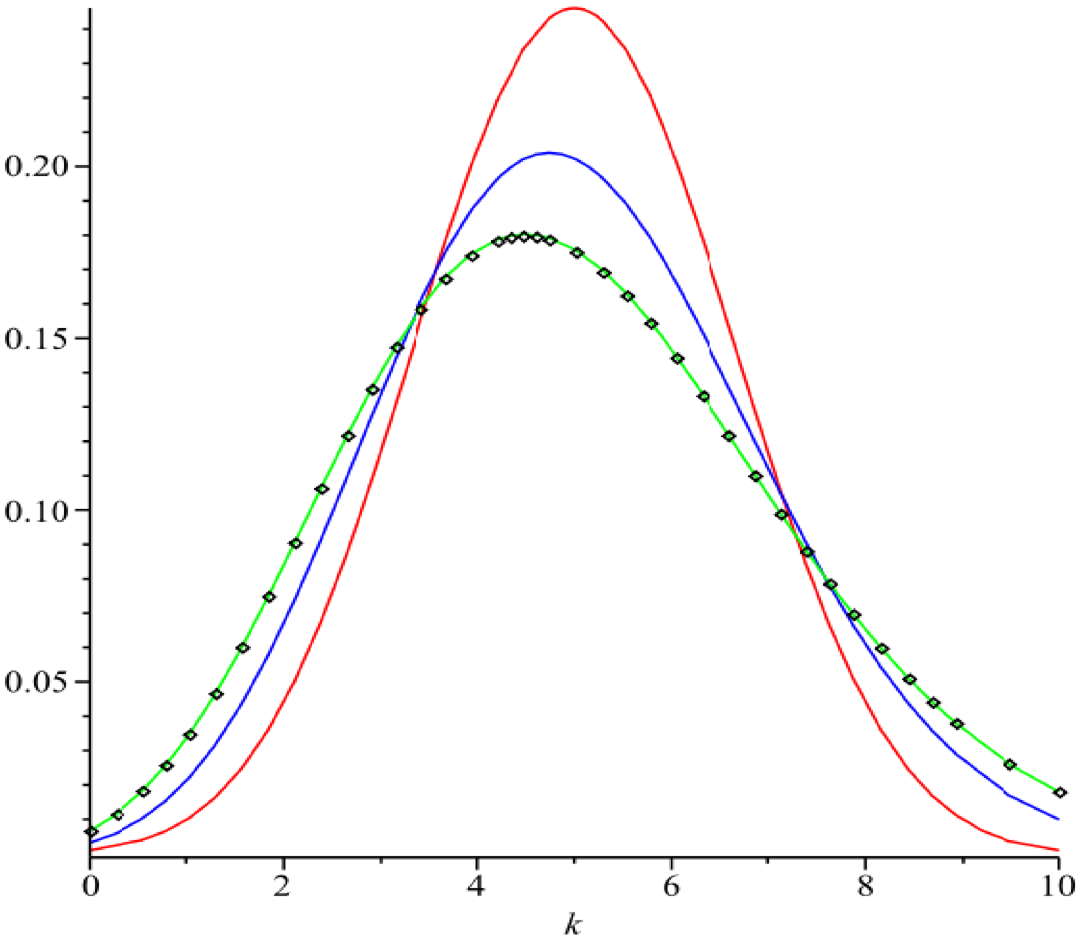


Figure 4.3- Graph of Poisson's distribution [20]

The Poisson distribution can be applied to systems with a large number of possible events, each of which is rare. A classic example is the nuclear decay of atoms.

Chapter 5 –Implementation and Results

5.1 Introduction to AWK

AWK [20] is an extremely versatile programming language for working on files. A file is treated as a sequence of records, and by default, each line is a record. Each line is broken up into a sequence of fields, so we can think of the first word in a line as the first field, the second word as the second field, and so on. An AWK program is of a sequence of pattern-action statements. AWK reads the input a line at a time. A line is scanned for each pattern in the program, and for each pattern that matches, the associated action is executed.AWK is one of the early tools to appear in Version 7 Unix and gained popularity as a way to add computational features to a Unix pipeline. A version of the AWK language is a standard feature of nearly every modern Unix-like operating system available today. AWK is mentioned in the Single UNIX Specification as one of the mandatory utilities of a Unix operating system. Besides the Bourne shell, AWK is the only other scripting language available in a standard Unix environment.

5.2 AWK commands

AWK commands are the statement that is substituted for action in the examples above. AWK commands can include function calls, variable assignments, calculations, or any combination thereof. AWK contains built-in support for many functions.

5.3 Comparison Parameters:

5.3.1 Latency:

Latency is the time required for a packet to traverse the network from source to destination. All kind of delays like transmission, processing and propagation delays are included in it. Latency of every protocol varies because of its routing mechanism. It is also affected by parameters like routing load, number of control packets etc. Latency of MANETs varies due to its mobility and dynamic nature. Due to high mobility link breakages increases which causes the delay to become higher. Latency of the protocols discussed is calculated and compared in order to determine the effectiveness of each of these protocols in Mobile Ad hoc Networks (MANETs).

5.3.2 Throughput:

Throughput or network throughput is the average rate of successful message delivery over a communication channel**.** The system throughput or aggregate throughput is the sum of the data rates that are delivered to all terminals in a network. This number is closely related to the channel capacity of the system, and is the maximum possible quantity of data that can be transmitted under ideal circumstances. Throughput of the protocols discussed is calculated and compared in order to determine the effectiveness of each of these protocols in Mobile Ad hoc Networks (MANETs).

5.4 Results and Analysis:

5.4.1 Results of On-Foot Scenario

Here OLSR and DSDV are proactive in nature and hence route discovery time is less as compared to the reactive protocols (i.e. DSR and AODV). The transmission of packets in proactive protocols takes less time as compared to the on-demand.

In a scenario of low mobility and less number of nodes, AODV out-performs DSR because DSR stores multiple routes in the route discovery, whereas AODV only stores the shortest path. Since mobility is low in this scenario, so probability of link failure is less, hence path established by AODV lasts longer and performs better.

While comparing reactive with the proactive protocols generally, proactive protocols establish routing tables. Since here the number of nodes is less and mobility is less, so tables require comparatively less periodic updating , hence outperforming the reactive protocols.

OLSR performs better than DSDV because it uses MPRs, which reduces the time to discover the destination. As the offered traffic increases, OLSR takes less time in finding the destination, hence increasing the throughput

In DSR and AODV, DSR always demonstrates lower routing load than AODV. This happens because AODV has more route requests than DSR. DSR uses aggressive caching and is likely to find a route in cache, consequently lowering the route requests as compared to AODV.

The routing load of OLSR is high because it keeps sending HELLO messages to detect its MPRs set. This link sensing increases the routing load of OLSR.

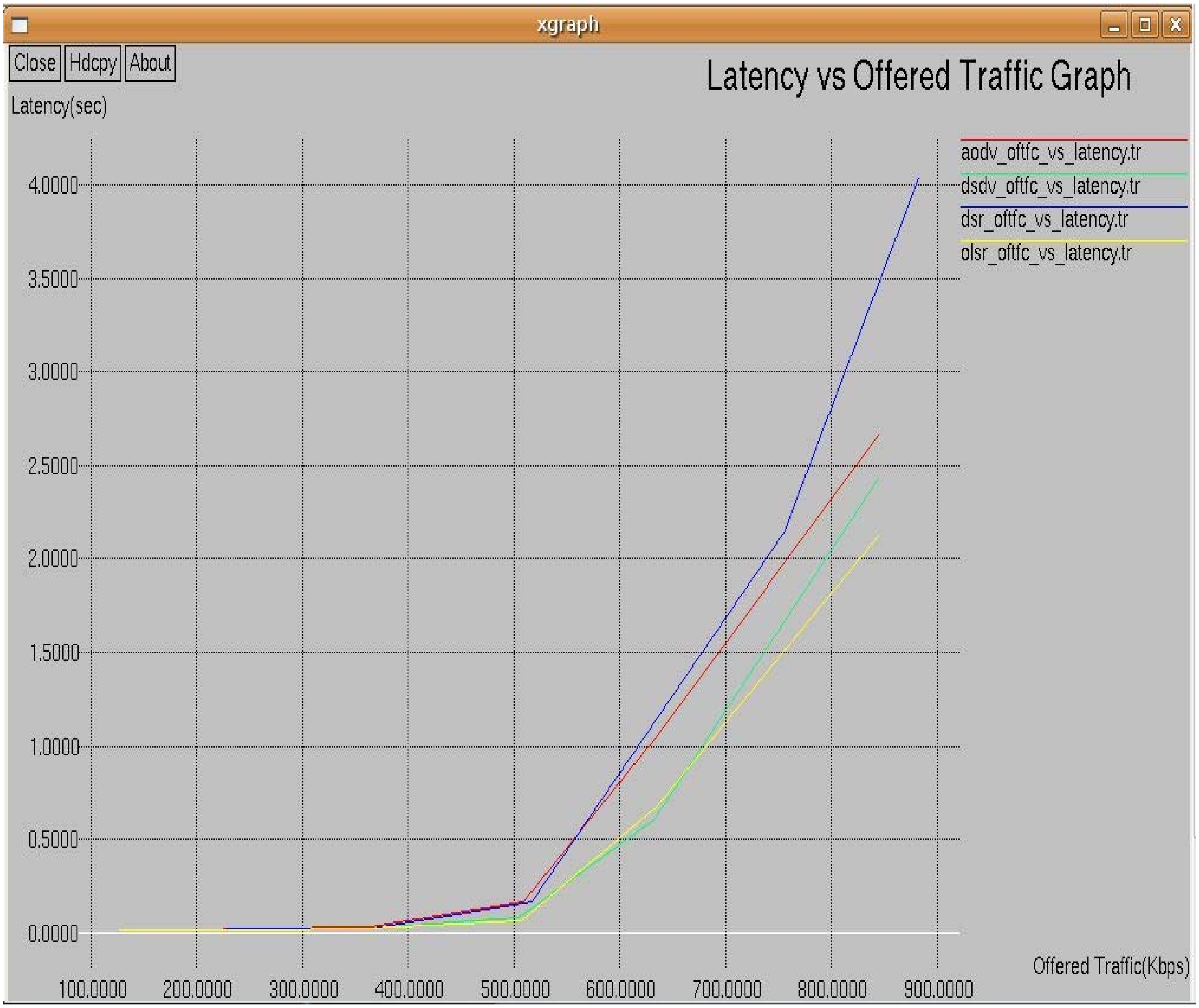


Figure 5.1 Latency vs Offered Traffic (on foot scenario)

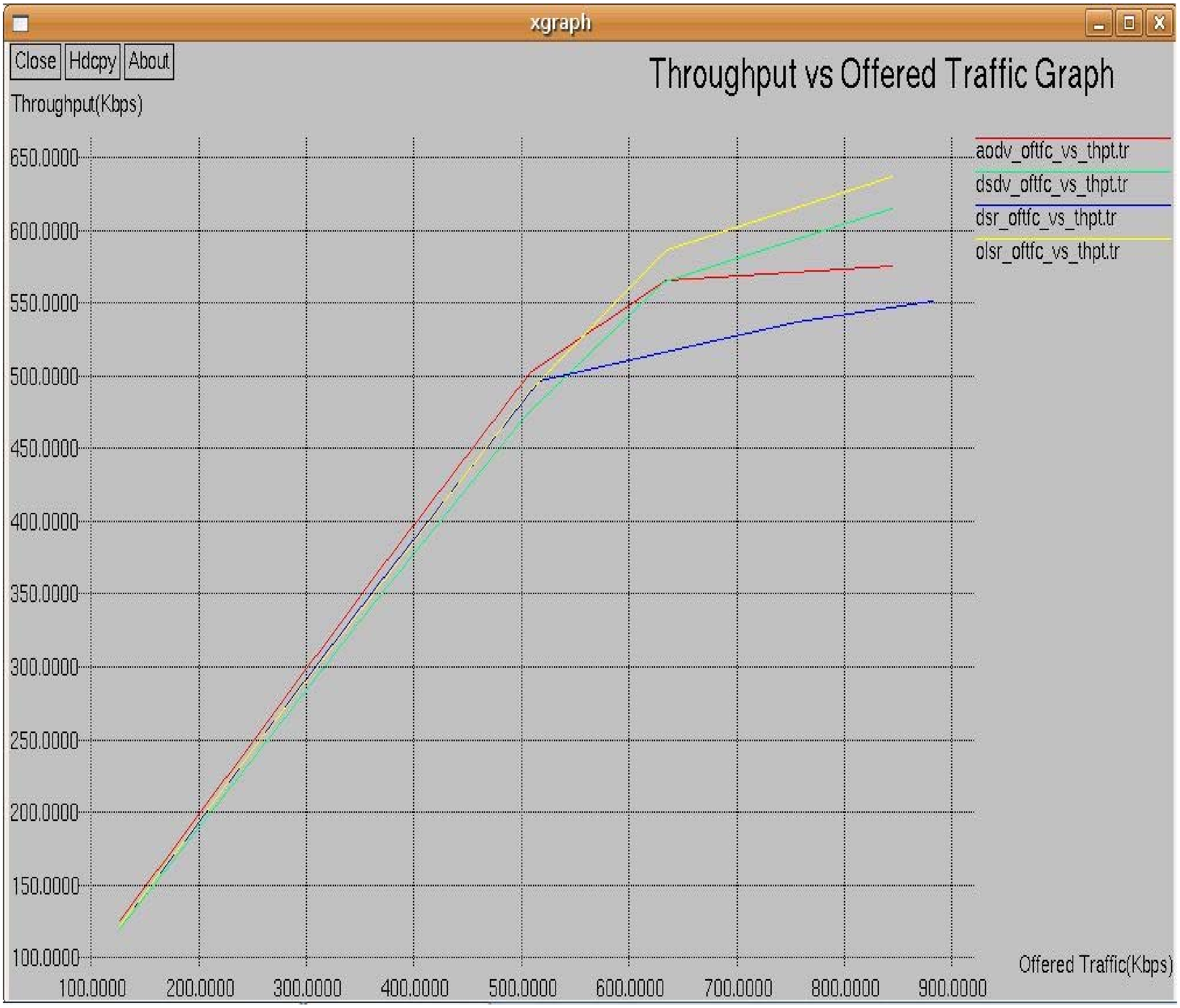


Figure 5.2 Throughput vs Offered traffic (on foot scenario)

.

5.4.2 Results of City Scenario:

In vehicular city scenario, the mobility has increased causing link failures more frequent. Since AODV has one route per destination in the routing table, it has maximum throughput because it is adaptive to the route changes that occur frequently during high mobility.

With high mobility, most of the cashed routes have chances of being stale. Hence, more route requests are made and DSR performs poorly. Therefore, its throughput also falls badly.

OLSR and DSDV perform almost similarly as both are proactive. Since route changes are rapid so both these protocols are not adaptive to these rapid changes consequently giving lesser throughput as compared to AODV.

In a highly mobile scenario, DSR has a larger latency because it tries to send the data packet to destination through any of the route available in cache. As the offered traffic also increases along with increased mobility, DSR has an increasing latency.

AODV, DSDV and OLSR behave almost similarly in the beginning. As the offered traffic increases, (as it is already mentioned that) throughput of AODV increases, it implies that its latency will decrease as it is shown from the graph. AODV has lowest latency since it has highest throughput. Similarly DSDV and OLSR will have values in between that of AODV and DSR (as shown in graph).

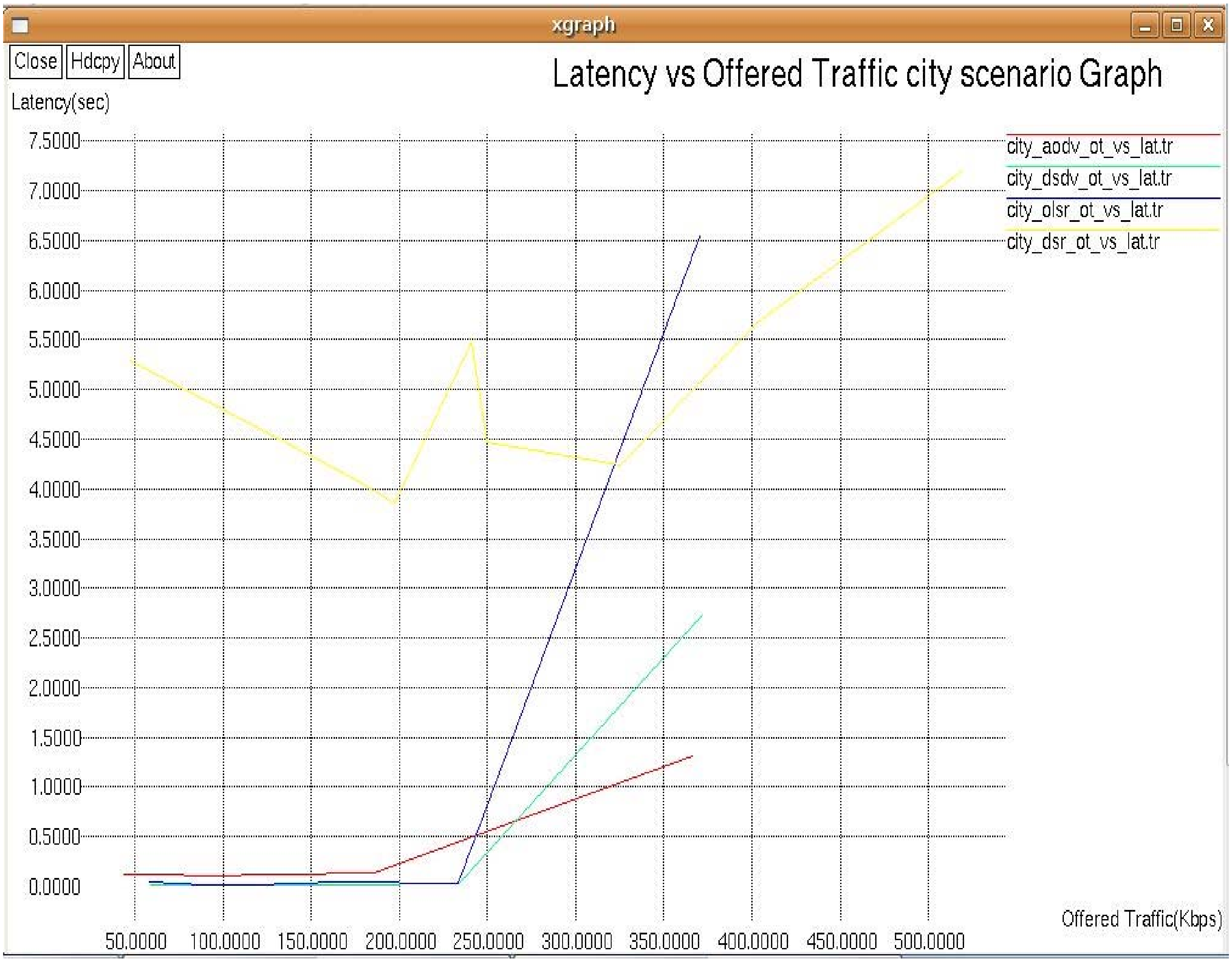


Figure 5.3 Latency vs Offered traffic (vehicle city scenario)

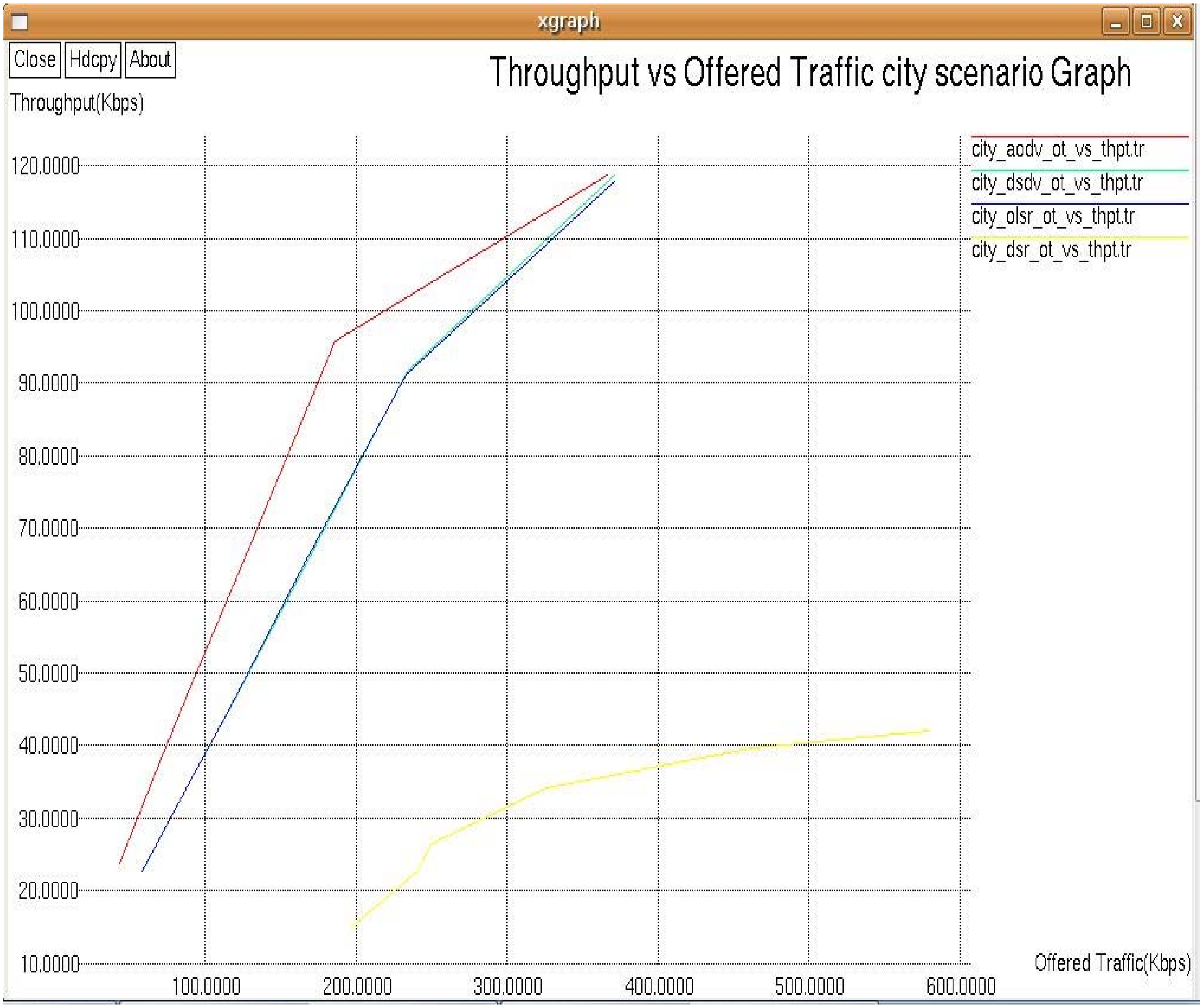


Figure 5.4 Throughput vs Offered Traffic (vehicle city scenario)

5.4.3 Results of Hybrid Scenario:

In hybrid scenario, throughput generally increases since this scenario contains both on-foot as well as vehicular devices, so mobility is not that rapid as in vehicular scenario. DSR performs better in this scenario as far as throughput is concerned. Since DSDV and OLSR are proactive and they are not adaptive to rapid topology changes, so DSR (having multiple routes in cache) out performs them.

As the offered traffic increases, latency of DSR decreases, but latency of DSDV and OLSR increases. Routing load of DSR increases with increased offered traffic, it reaches its maximum value of approx. 190 Kbps and then again starts decreasing when offered traffic is further increased due to aggressive caching.

However, OLSR and DSDV show a smooth curve for routing load. OLSR has much higher routing load as compared to DSDV due to MPRs tracking.

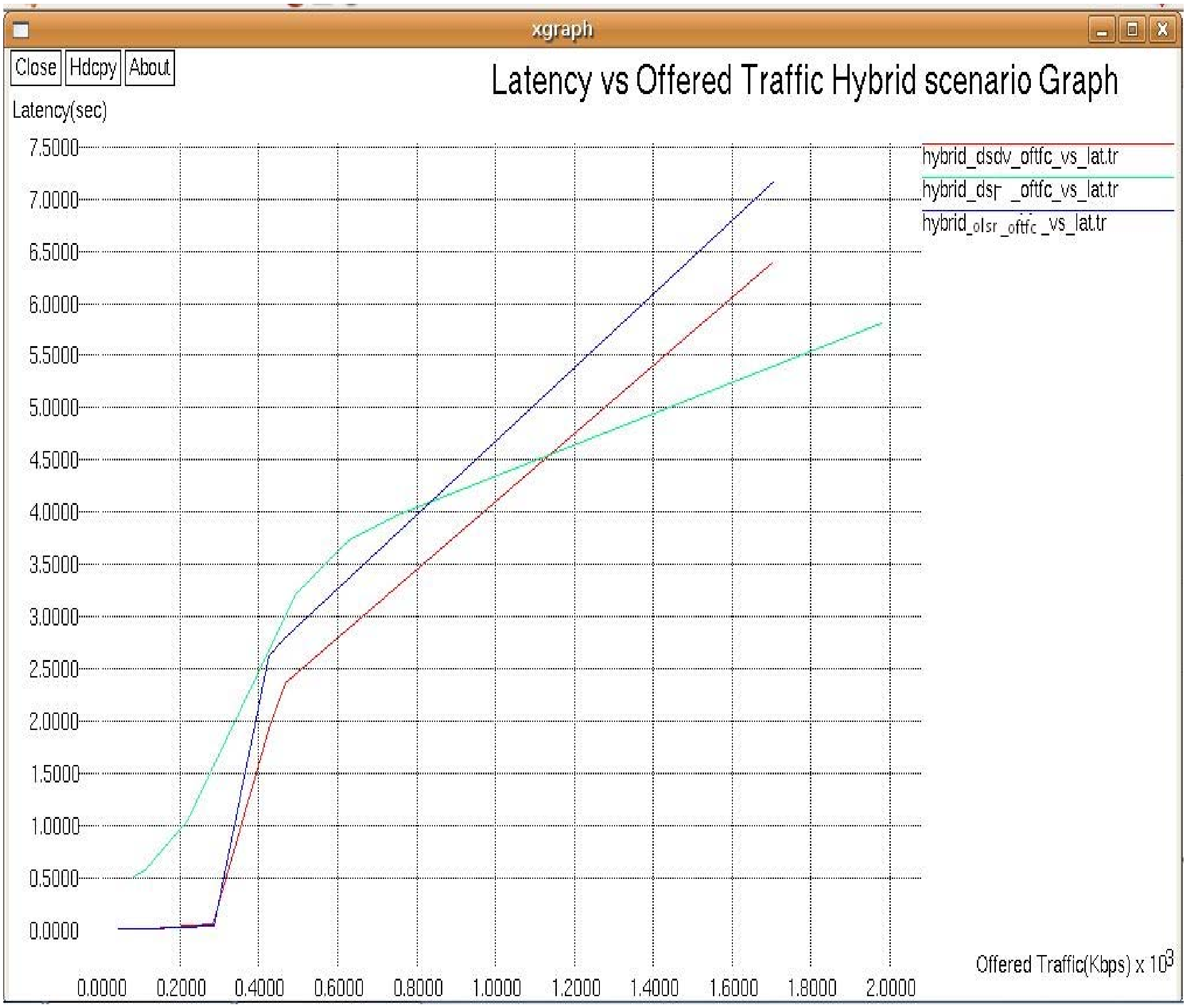


Figure 5.5 Latency vs Offered Traffic (hybrid scenario)

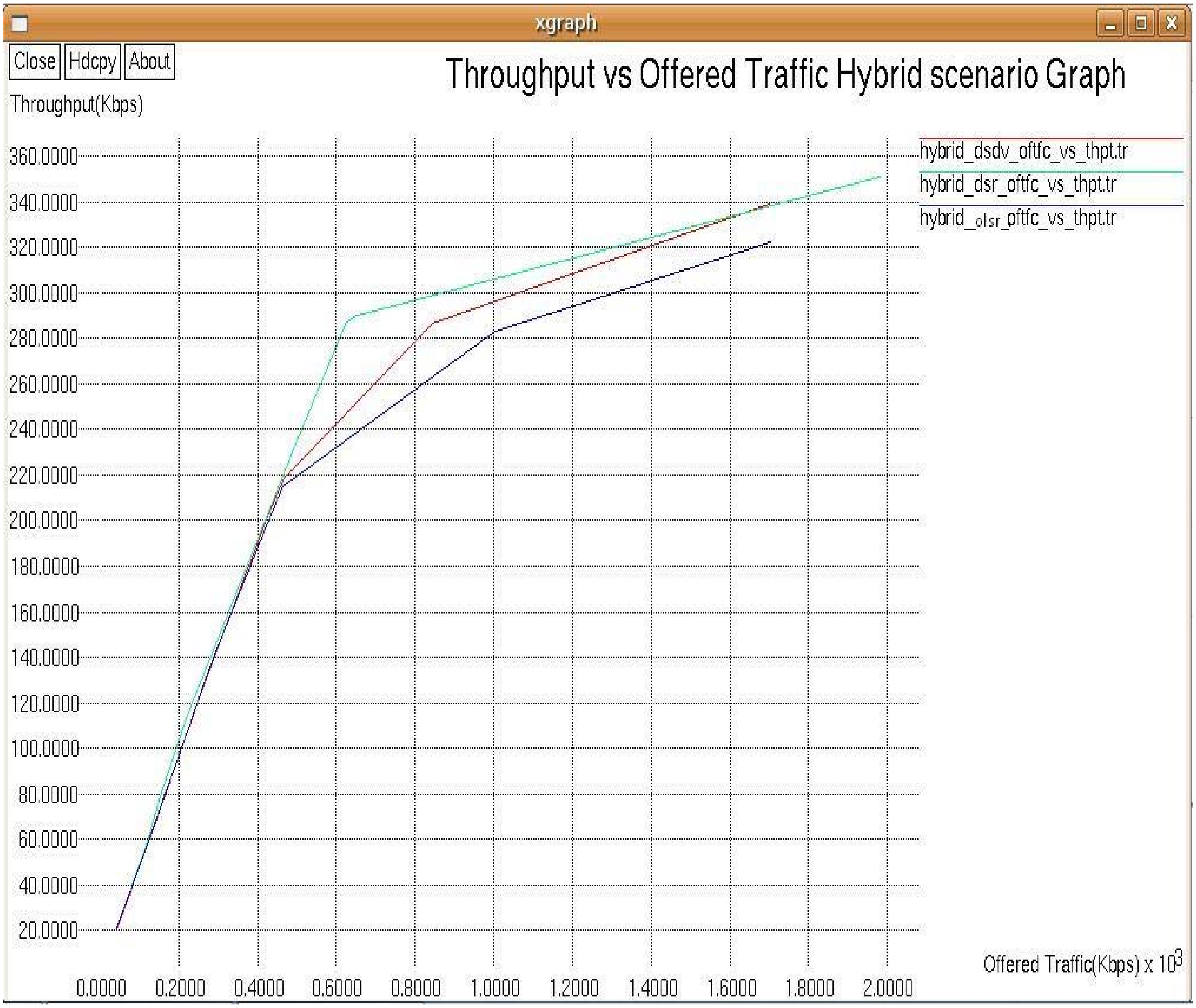


Figure 5.6 Throughput vs Offered Traffic (hybrid scenario)

Future Work:

In future, more protocols can be compared on different scenarios. Our results can be used in the future and ideas can be taken from them for further use. Our codes can be implemented to find latency, throughput and routing load for these protocols and for other protocols also, with some amendments.

Bibliography

[1] “An Optimal Caching Technique for Wireless Ad hoc Network using Connected Dominating” Set by Naveen Nahata , Dr. Shasikala Tapaswi, Tony Johri, Namit Mishra

[2] “MANET versus WSN by” JA Garcia-Macias1 and Javier Gomez

[3] “Performance Comparison and Analysis of DSDV and AODV for MANET” by V.Ramesh , Dr.P.Subbaiah ,N. Koteswar Rao, M.Janardhana Raju

[4] “A Performance Comparison of Multi-Hop Wireless Ad Hoc NetWork Routing Protocols” by Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu and Jorjeta Jetcheva

[5] “An efficient and secure routing protocol for mobile adhoc networks” by N. Ch. Sriman Narayana Iyengar.

[6] Charles E. Perkins and Pravin Bhagwat. “Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers”. In Proceedings of the SIGCOMM ’94 Conference on Communications Architectures, Protocols and Applications, pages 234–244, August 1994. A revised version of the paper is available from http://www.cs.umd.edu/projects/mcml/papers/Sigcom m94.ps.

[7] C. Perkins Ad hoc On-Demand Distance Vector (AODV) RoutingRFC3561 [S] July 2003

[8] David B. Johnson, David A. Maltz and Yih-Chun Hu. “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks”. Internet-Draft, draft-ietfmanet- dsr-10.txt, July 2004

[9] “Optimized Link State Routing Protocol (OLSR)” Project Hipercom, INRIA by T. Clausen, Ed. And P. Jacquet, Ed.

[10] “Performance analysis of adhoc network routing protocols” by P. Chenna Reddy and Dr. P. Chandrasekhar Reddy ^{2}

[11] “Throughput Enhancement in Scalable MANETs using Proactive and Reactive Routing Protocols” by M.Saravana karthikeyan, K.Angayarkanni , and Dr.S.Sujatha , *Member*, IAENG

[12] “A Survey of Secure Mobile Ad Hoc Routing Protocols” by Loay Abusalah, Ashfaq Khokhar, and Mohsen Guizani I 1553-877X/08/$25.00 © 2008 IEEE Manuscript received June 13, 2006

[13] INTERNET-DRAFT 19 July 2004 David A. Maltz, Carnegie Mellon University ,Yih-Chun Hu, Rice University “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)”

[14] A. Qayyum, L. Viennot, A. Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks.35th Annual Hawaii International Conference on System Sciences (HICSS'2001).

[15] An Introduction to NS, Nam and OTcl scripting by Paul Meeneghan and Declan Delaney

[16] “Vehicular grid communications: the role of the internet infrastructure” by Mario Gerla, Biao Zhou,

[17] http://www.ercim.eu/publication/Ercim_News/enw57/santi.gif

[18] http://cache.jalopnik.com/assets/resources/2006/12/car_2_car.jpg

[19] http://www.atacwireless.com/pics/image001.jpg

[20] http://en.wikipedia.org/wiki/File:Binomial_versus_poisson.svg

[21] A Tutorial and Introduction - by Bruce Barnett