Ad Hoc Network for Oil Supply Boats

By
Peipei Liu

Thesis submitted in Partial Fulfillment of the Requirements for the Degree Master of Technology in Information and Communication Technology

Faculty of Engineering and Science
University of Agder

Grimstad
May 2009

Abstract

An ad hoc network is a collection of wireless mobile nodes dynamically forming a temporary network without any fixed infrastructure like access points or base stations. Each node in ad hoc network is willing to forward data for other nodes, and the determination of which nodes forward data is made dynamically based on the network connectivity. Mobile ad hoc networking can offer multiple advantages in various environments through its flexibility and its special nature. One such environment is maritime communications.

The MARCOM Project currently has shown that a definite protocol simulation of maritime ad hoc network is needed, which focus on the hand-over problems, as well as the mixture of fixed and mobile nodes interconnected via wireless links to form a multi-hop ad-hoc network, amongst ships, marine beacons and buoys.

In this thesis, it is mainly focus on comparing simulation performance between AODV and OLSR. The simulation results have shown that in definite maritime ad hoc networks neither of them suits without modifications. The scenarios are built on maritime which means the protocols have to be workable in the special situation at sea.

These simulations in my thesis do not give a general view of the protocols, but instead test certain characteristics of the protocols in maritime scenarios. If time allows, there could be more discussion about enhancements needed for both AODV and OLSR, like position aware and parameter aware.

Preface

This report is the result of my master thesis project associated with the MARCOM Project, which is financed by the Research Council of Norway with participation from several research and development institutions, universities and colleges, public authorities and industry.

This master thesis is also the last part of my Master of Information Communication of Technology degree at University of Agder.

The MARCOM Project takes up the challenges in the Norwegian Ministry of Trade and Industry's MARUT-IKT-program, and similar activities in ITS-Norway associations' maritime group. MARCOM will also work with the issues from IMO: eNavigation. The problems and opportunities connected to maritime communication are multiple, need for better security, more efficiency, new services for passengers and new business opportunities for the Norwegian maritime industrial cluster.

I would like to thank my supervisors, Frank Li, Magne Arild Haglund and Ziaul Haq Abbas for the professional tutoring, follow up and support with kind effort throughout the project period. Also thanks to Andreas Prinz and The MARCOM Project for giving this valuable opportunity to learn further protocol simulation.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

In the present era, Ad Hoc Network for oil supply boats is a special technology integrating ad hoc network, wireless LAN (WLAN) and efficiency on the sea. Currently, oil platforms are connected to broadband via cable. These platforms can be used together with the shore installations to establish connectivity to the supply boats that run regularly between the platforms and the ports. The boats and installations on the oil platforms and ports are connected to establish ad hoc network that we can typically consider as a sort of Vehicular Ad Hoc Networks (VANETs). The network can use cellular gateways and WLAN access points to connect to the Internet, gather traffic information or for routing purposes. VANET is different from other kinds of ad hoc networks by their hybrid network architectures, node movement characteristics, and new application scenarios.

Simulation is a very powerful tool to analyze and plan maritime operations. A well-designed simulation model can provide useful insights about complex marine situations; in addition, a right-chosen routing protocol is the critical point for economy and high efficiency shipping traffic communication. An important part of any marine operations simulator is modeling the shipping traffic. Before the communication is built, selection of rational routing protocols has to be faced. This thesis presents the performance evaluation of two popular routing protocols, Ad-Hoc On-Demand Distance Vector (AODV) and Optimized Link State Routing (OLSR), based on some classic marine traffic scenarios. The performance comparison is performed using various metrics like Packet Loss and Throughput based on Network Simulator 2 (NS2).

The thesis describes all the parameters used for the simulations in detail and then compares each routing protocol's simulation results before arriving at a conclusion as to which is the best one for definite marine ad hoc networks.

1.2 Thesis definition

A Vehicular Ad Hoc Network, or VANET, is a form of Mobile ad hoc network, to integrate the communication among nearby vehicles and between vehicles and nearby fixed equipment. This project considers boats and nearby fixed stations, such as oil platforms at sea and base stations on ports, as nodes in a maritime ad hoc network. Each boat with VANET device will be a node in the ad hoc network, the nodes that are within each other's radio range can communicate directly, while distant nodes rely on their neighboring nodes to forward packets. Basically, the idea of the ad hoc network for oil supply boats at sea is that the boats can communicate directly with each other within transmission coverage, otherwise communicate on multi-hops. These protocols supporting wireless ad hoc network enable the exchange of data between distinct pairs of nodes, using intermediate network participants for forwarding packets on their way to the destination.

A lot of routing protocols have been developed for Mobile Ad Hoc Networks (MANETs). Mobile ad hoc networks became a popular subject for research and 802.11/Wi-Fi wireless networking became widespread in the mid to late 1990s. Many of the academic papers evaluate protocols and abilities assuming varying degrees of mobility within a bounded space, usually with all nodes within a few hops of each other, and usually with nodes sending data at a constant rate. However, due to the unreliable channel conditions on the sea, high boat speed and dynamic information exchange make the situation different from those of MANETs. Thus, trying to find and evaluate rational protocols for marine ad hoc networks is necessary and challenging.

In this project, the main task is to focus on maritime routing protocols established for oil supply boats, which means the protocols have to be workable in the special situations at sea. Then the following issues will be discussed definitely:

Can ad hoc networking technology be applied to oil supply boat scenarios? If yes, which protocol performance best?
Are existing protocols sufficient for communication in oil supply boat scenarios? If not, which aspects should be improved?
If time allows, propose enhancements to existing protocols would be discussed.

1.3 Background information of The MARCOM project

The MARCOM project is a joint initiative from several Research and development institutions, Universities and Colleges, public authorities and industry. The industry itself and The Norwegian Research Council fund the project. The project spans for four years starting in 2007.

This case will focus on the hand-over problems, as well as the mixture of fixed and mobile nodes interconnected via wireless links to form a multi-hop ad-hoc network, amongst ships, marine beacons and buoys.

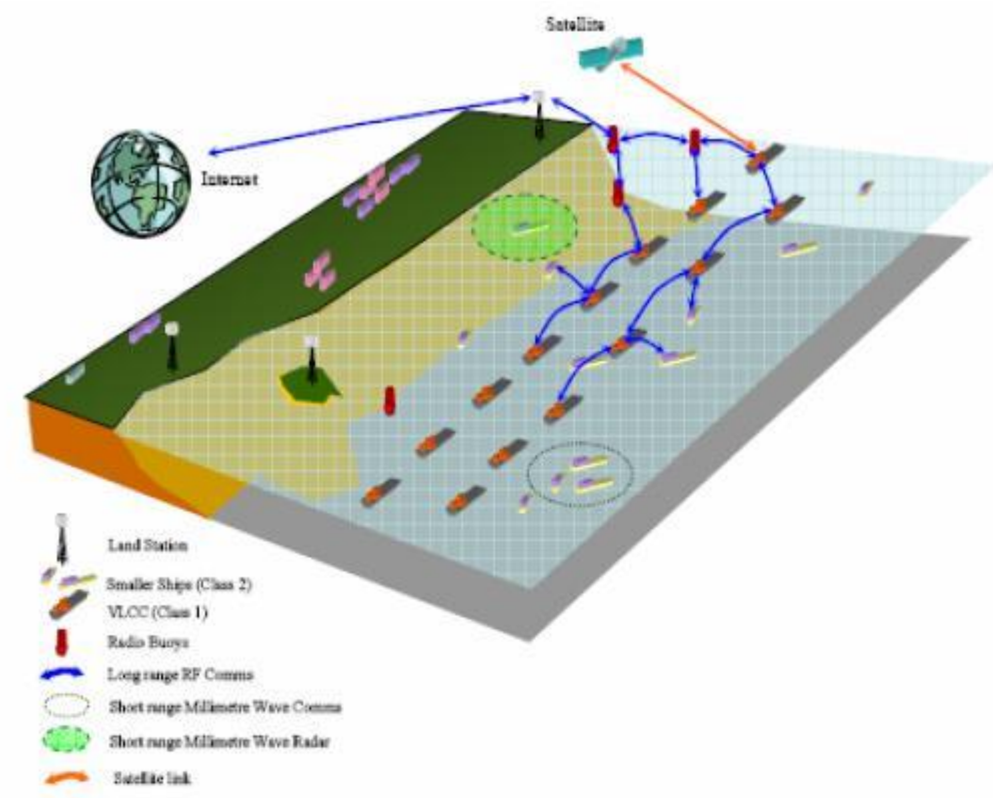


Figure 1 Project realistic architecture (From www.marcom.no)

1.4 Problem arena

The project discuses maritime wireless networks built by communicable boats and platforms which are considered as mobile nodes in the ad hoc networks and can receive and relay others messages through the wireless networks.

Typically, a mobile ad hoc network is a network comprising wireless mobile nodes that communicate with each other without a fixed infrastructure like access points or base stations. Vehicular ad hoc networks are kind of special cases where the nodes can only move on prevision routes with specified directions for all boats at sea. Every mobile node, the communication-needed boat, in the VANET acts as a wireless station and mobile router at the same time. To find a route to a destination a routing protocol is obviously used.

A number of routing protocols for VANET's have been proposed and evaluated. These evaluations often involve simulations, for the reason of the arrangement and management of such a performance evaluation with a large number of vehicles is too expensive.

Routing protocols are divided into two categories: Proactive and Reactive. Proactive routing protocols are table-driven protocols and they always maintain current up-to-date routing information by sending control messages periodically between the host nodes which update their routing tables. The proactive routing protocols use link-state routing algorithms which frequently flood the link information about its neighbor boats. Reactive or on-demand routing protocols create routes when they are needed by the source host and these routes are maintained while they are needed.

My goal is to carry out a systematic performance study of two routing protocols for ad hoc networks, Ad hoc On Demand Distance Vector (AODV) routing protocol and Optimized Link State Routing (OLSR) protocol, and then evaluate performance and compare each routing protocol's simulation results with some metrics in the same scenarios.

1.5 Method and thesis work

In this project, the main task is focus to select some available ad hoc network protocol that can have better performance for the communication around boat-boat and boat-platform at sea.

According to the long communication distance, broadcast is a used routing method in this VANET. When the message needs to be sent beyond the transmission range, multi-hop is used.

The simplest way to implement a broadcast service is flooding in which each node re-broadcasts messages to all of its neighbors except the one it got this message from. Flooding guarantees the message will eventually reach all nodes in the network. Flooding performs relatively well for a limited small number of nodes. But when the number if nodes in the network increase, the performance drops quickly. The bandwidth requested for one broadcast message transmission can increase exponentially. As each node receives and broadcasts the message almost at the same time, this causes contentions and collisions, broadcasts storms and high bandwidth consumption. Flooding may have a very significant overhead and selective forwarding can be used to avoid network congestion. [1]

The first step to evaluate performance of maritime ad hoc network routing protocols requires the use of modeling for generating realistic boat and platform scenarios. For this purpose, five scenarios to define typical network communication situations at sea are presented below:

(1) Pure Cellular Network

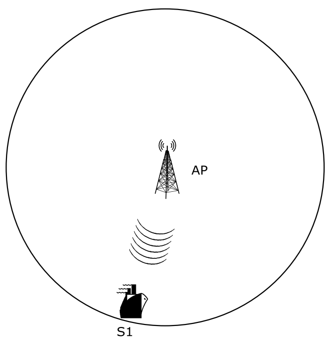


Figure 2 Pure Cellular Network

The cellular network is used where the base station is available and the boat is within the base station coverage.

(2) Pure Ad Hoc Network among boats

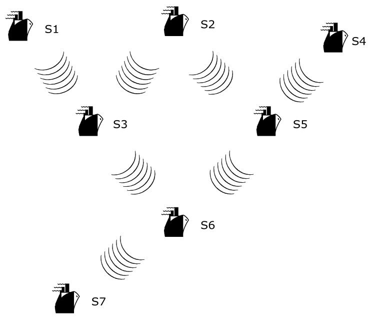


Figure 3 Pure Ad Hoc Network among boats

All boats within the available communication coverage can form a mobile ad hoc network to perform Boat-to-Boat communications.

(3) Hybrid: Boats – Platform

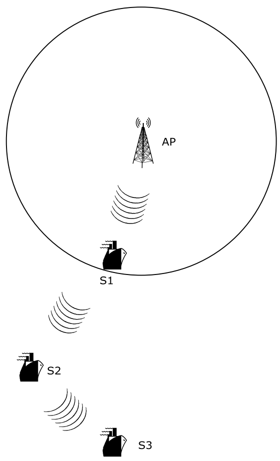


Figure 4 Hybrid: Boats – Platform

The boat S3 performs Boat-Access Point communication through S1 and S2.

(4) Hybrid: Boats – Boats

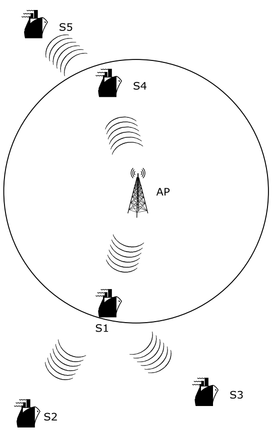


Figure 5 Hybrid: Boats – Boats

The boats S2, S3 and S5 perform communications through Access Point S1 and S4.

(5) General condition

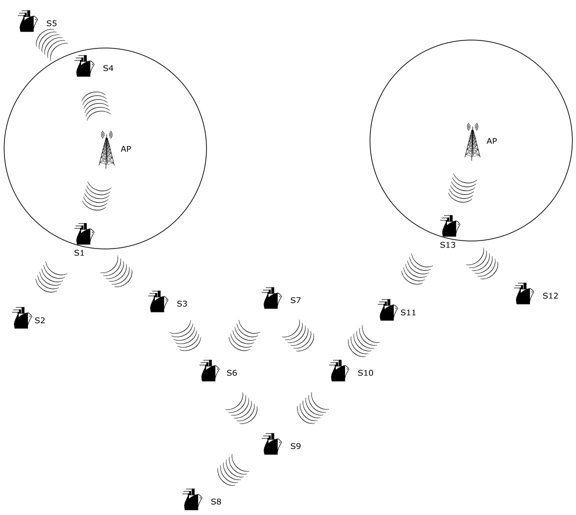


Figure 6 General condition

This composited scenario simulates the realistic condition.

1.6 Project outline

As mentioned before, in this project, the first part of my task is to focus on selecting some available ad hoc network protocols that can work well for the communication between boat-boat and boat-oil platform at sea. After suitable protocols have been chosen, the second part of task is trying to evaluation the performance of the routing protocols.

In order to evaluate the performance of the routing protocol, I choose the Network Simulator 2. It is freely available and widely used for research on mobile ad hoc networks. Furthermore, simulation is done on the packet level. Thus, a detailed analysis of the simulation results is possible.

2 Related Work and Simulation Tool

NS or the network simulator (also popularly called NS-2, in reference to its current generation) is a discrete event network simulator. It is popular in academia for its extensibility (due to its open source model) and plentiful online documentation. ns is popularly used in the simulation of routing and multicast protocols, among others, and is heavily used in ad-hoc networking research. ns supports an array of popular network protocols, offering simulation results for wired and wireless networks alike. It can be also used as limited-functionality network emulator. [2]

2.1 Introduction to NS-2

NS-2 is a packet-level simulator which is essentially a centralized discrete event scheduler to schedule the events such as packet and timer expiration. The centralized event scheduler cannot accurately emulate "events occurred at the same time", instead, it can only handle events occurred one by one in time. However, this is not a serious problem in most network simulations, because the events here are often transitory. Besides, NS-2 implements a variety of network components and protocols. Notably, the wireless extension, derived from CMU Monarch Project [3], has two assumptions simplifying the physical world [4]:

(1) Nodes do not move significantly over the length of time they transmit or receive a packet. This assumption holds only for mobile nodes of high-rate and low-speed.

(2) Node velocity is insignificant compared to the speed of light. In particular, none of the provided propagation models include Doppler effects, although they could.

2.2 GloMoSim and OPNET

GloMoSim[5] is another open-source network simulator which is based on parallel programming. Hopefully, it can emulate the real world more accurately. However, it may be hard to debug parallel programs. Although GloMoSim currently solely supports wireless networks, it provides more physical-layer models than ns-2. There is another simulator OPNET which requires licence. Table 1 compares the wireless physical models used in the three simulators.

Simulator	GloMoSim	Ns-2	OPNET
Noise (SNR) calculation	Cumulative	Comparison of two signals	Cumulative
Signal reception	SNRT based BER based	SNRT based	BER based
Fading	Rayleigh Ricean	Not included	Not included
Path loss	Free space Two ray etc	Free space Two ray	Free space

Table 1 Physical layer and propagation models available in GloMoSim, ns-2 and OPNET

2.3 NS-2 Basics

2.3.1 NS-2 directory structure

NS is 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.

Figure 7 presents the relationship between pure C++ objects and pure OTcl objects.

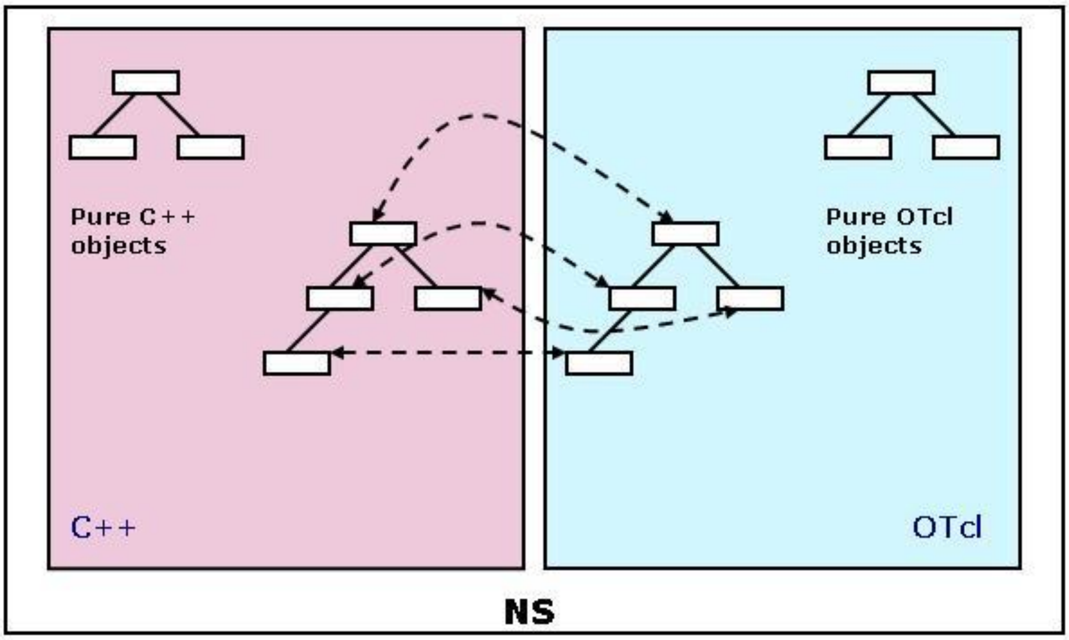


Figure 7 Relationship between C++ and OTcl in NS

As shown in the figure below, the C++ classes of ns-2 network components and protocols are implemented in the subdirectory “ns-2.*”, and the TCL library (corresponding to configurations of these C++ instances) in the subdirectory of “tcl”.

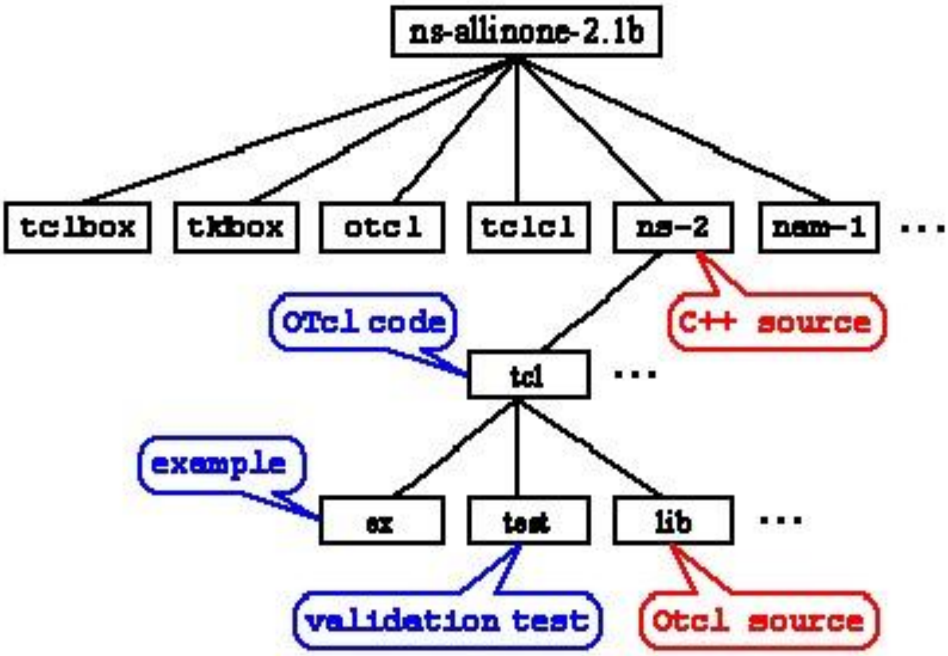


Figure 8 NS components

2.3.2 Network Components

Network components are Node, Link, Queue, etc. Some of them are simple components, that is, they corresponds to a single C++ object; the others are compound components, that is, they combine multiple simple components, e.g. a Link component is composed of a Delay component (emulating propagation delay) and a Queue component. In general, all network components are created, plugged and configured by some TCL scripts when ns-2 is initialized. [6]

Example: Plug MAC into NetIF (Network Interface)

```
Class MAC {
        Void send (Packet* p);
        Void recv(Packet*, Handler* h);
        NsObject*target_//pointing to an instance of NetIF
}
```

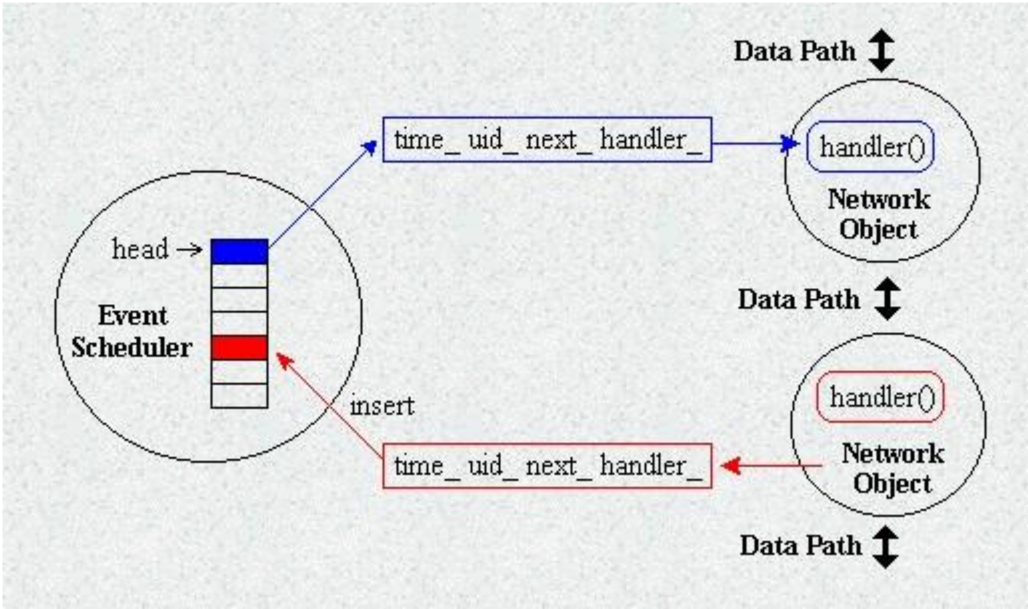


Figure 9 Discrete Event Scheduler

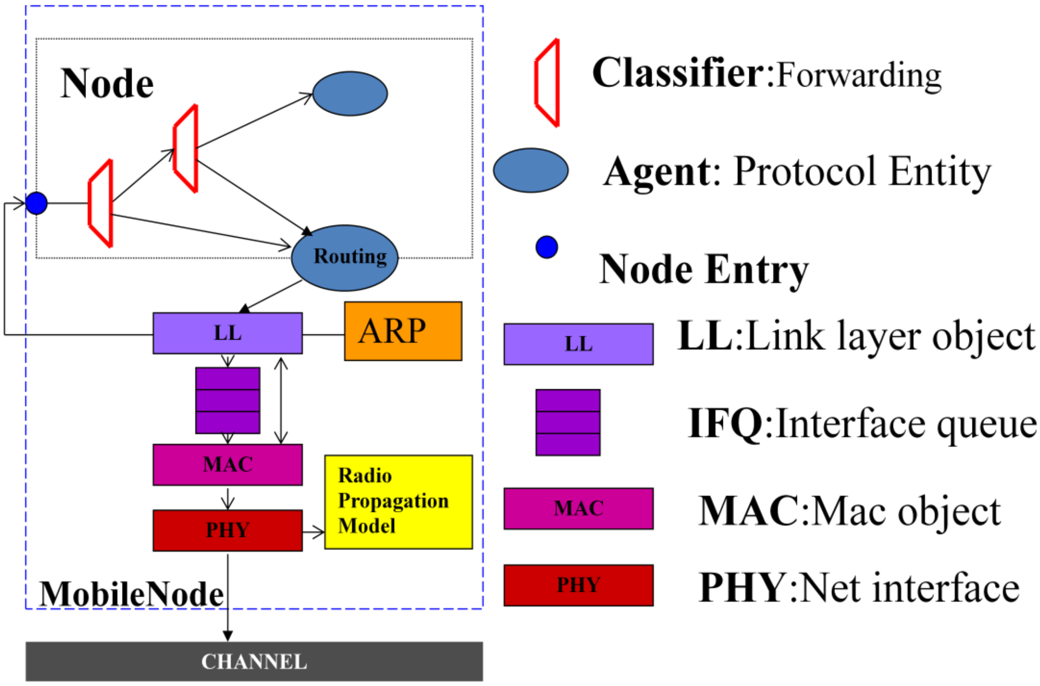


Figure 10 Mobile node structure2

- **Node**

A node is a compound object composed of a node entry object and classifiers as shown in the below figure. There are two types of nodes in NS. A unicast node has an address classifier that does unicast routing and a port classifier. A multicast node, in addition, has a classifier that classify multicast packets and a multicast classifier that performs multicast routing. [7]

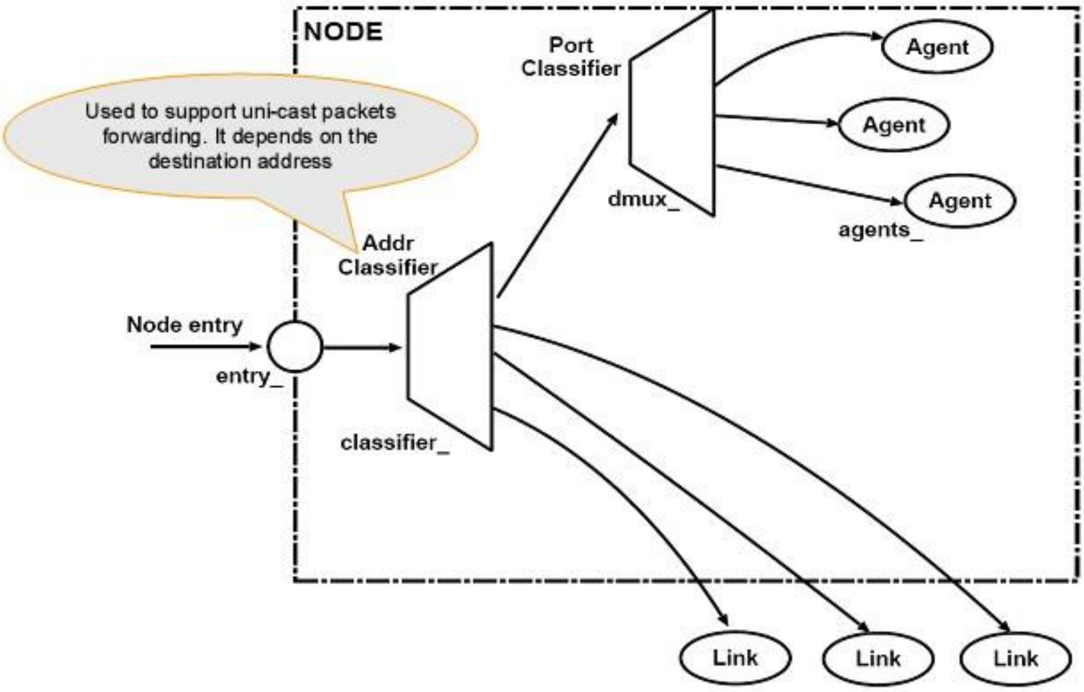


Figure 11 Unicast Node

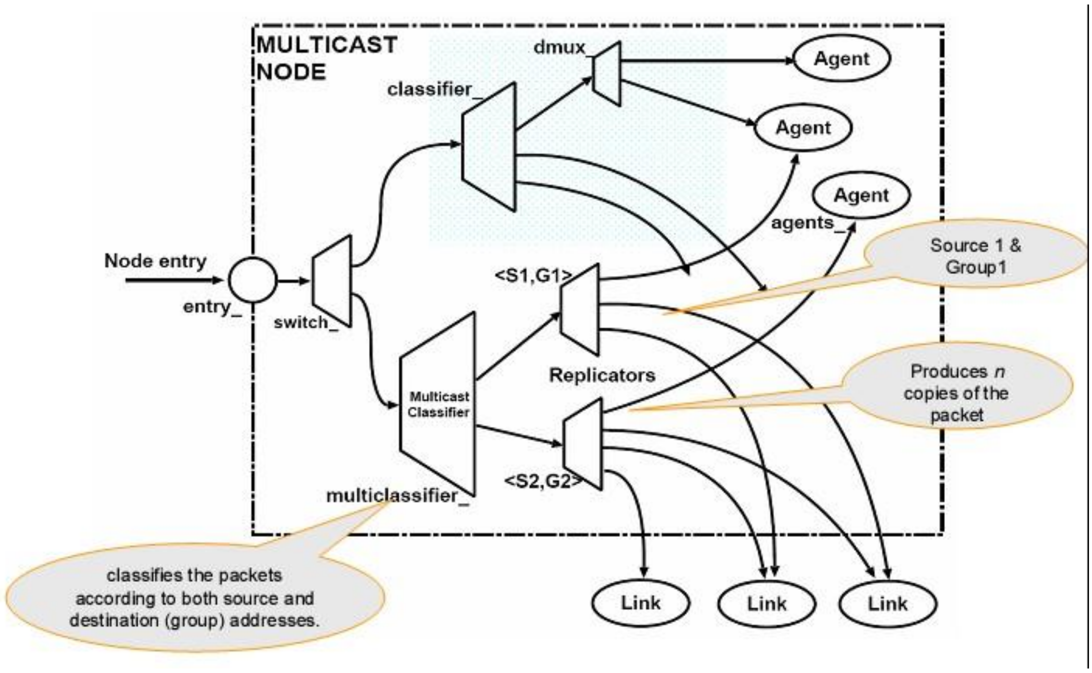


Figure 12 Multicast Node

In NS, unicast nodes are the default nodes. To create multicast nodes the user must explicitly notify in the input OTcl script, right after creating a scheduler object, that all the nodes that will be created are multicast nodes. After specifying the node type, the user can also select a specific routing protocol other than using a default one. [7]

Comparing the type of Unicast node, Multicast node has those characteristics:

(1) Extended structure than the other normal nodes
(2) There is no links between the nodes
(3) They can move inside a certain topology
(4) They should be configured by many parameters to define the physical, MAC, routing, …etc
(5) The routing could be wireless/Wireless-wired

The following parameters should be defined:
(1) Ad hoc Routing: Routing protocol -> AODV, DSDV, TORA, DSR …
(2) LL Type: The link layer -> LL, LL/Sat
(3) Mac Type: The MAC layer -> MAC/802_11, MAC/Sat, MAC/Sat/Unslotted Aloha, MAC/Tdma
(4) Ifq Type: Type of Queue -> Queue/Drop Tail, Queue/Drop Tail/priQueue
(5) Ifq Len: Length of the Queue
(6) Ant Type: Type of Antenna -> Antenna/OmniAntenna
(7) Prop Instance: Wireless propagation model -> Propagation/TwoRayGround, Propagation/Shadowing
(8) Phy Type: Type of physical interfaces -> Phy/WirelessPhy, Phy/Sat
(9) Channel: Type of wireless channel -> Channle/WirelessChannel, Channel/Sat
(10) Topo Instance: The used topology
(11) Wired Routing: Define if the node has a wired interface or not -> ON, OFF
(12) Mobile IP: Define if mobile IP is used or not -> ON, OFF

- **Link**

A link is another major compound object in NS. When a user creates a link using a duplex-link member function of a Simulator object, two simplex links in both directions are created as shown in the below figure.

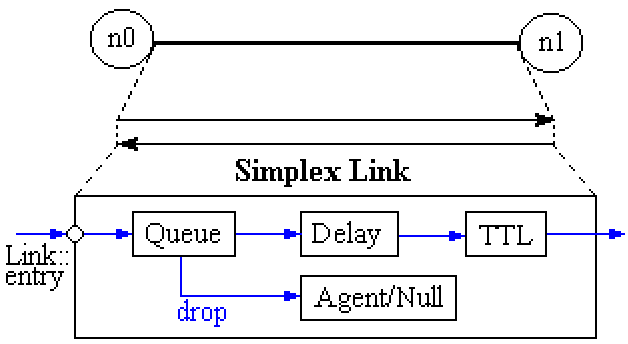


Figure 13 Link

One thing to note is that an output queue of a node is actually implemented as a part of simplex link object. Packets dequeued from a queue are passed to the Delay object that simulates the link delay, and packets dropped at a queue are sent to a Null Agent and are freed there. Finally, the TTL object calculates Time To Live parameters for each packet received and updates the TTL field of the packet.

- **Tracing**

In NS, network activities are traced around simplex links. If the simulator is directed to trace network activities (specified using *$ns* trace-all file or *$ns* nam trace-all *file*), the links created after the command will have the following trace object of type *type* between the given *src* and *dst* nodes using the create-trace {*type file src dst*} command.

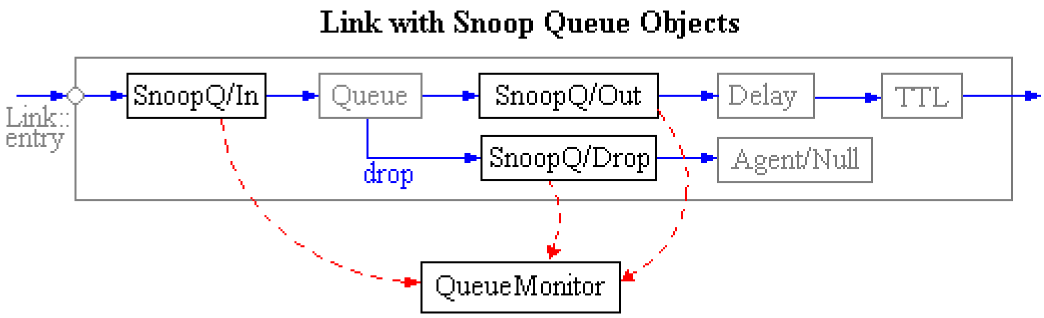


Figure 14 Inserting Trace Objects

When each inserted trace object (i.e. EnqT, DeqT, DrpT and RecvT) receives a packet, it writes to the specified trace file without consuming any simulation time, and passes the packet to the next network object. The trace format will be examined in the General Analysis Example section.

- **Queue Monitor**

Basically, tracing objects are designed to record packet arrival time at which they are located. Although a user gets enough information from the trace, he or she might be interested in what is going on inside a specific output queue. For example, a user interested in RED queue behavior may want to measure the dynamics of average queue size and current queue size of a specific RED queue (i.e. need for queue monitoring). Queue monitoring can be achieved using queue monitor objects and snoop queue objects as shown in Figure below.

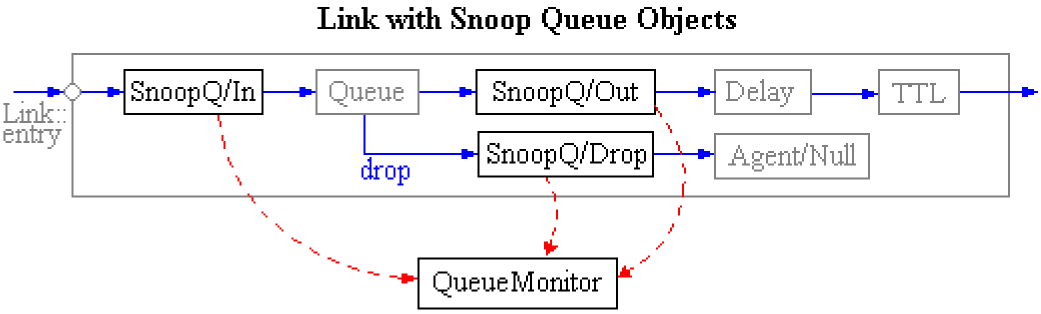


Figure 15 Monitoring Queue

When a packet arrives, a snoop queue object notifies the queue monitor object of this event. The queue monitor using this information monitors the queue. A RED queue monitoring example is shown in the RED Queue Monitor Example section.

2.4 NS-2 Environment

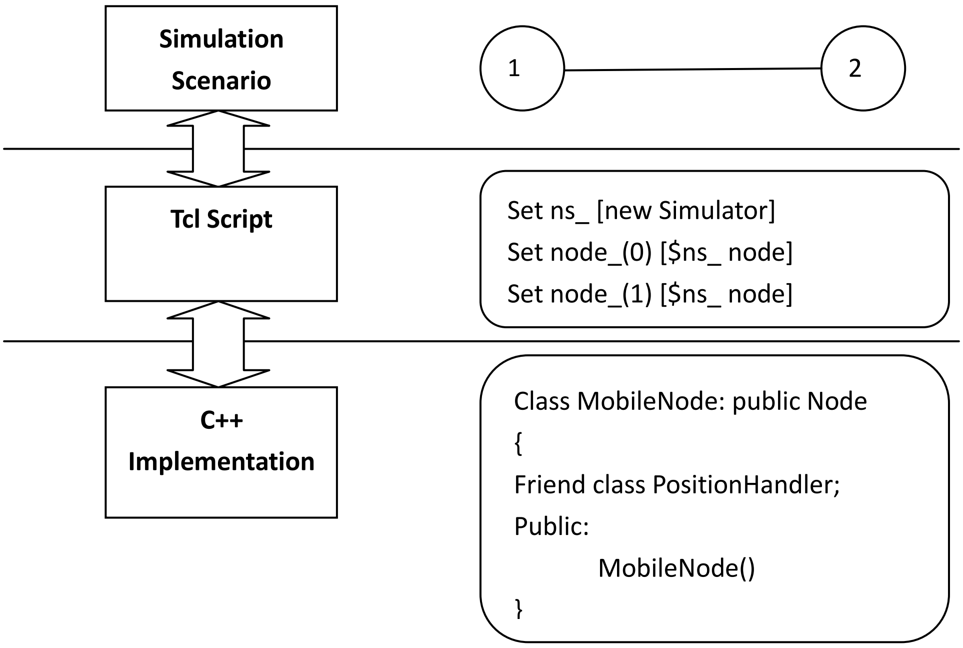


Figure 16 NS-2 Environment

3 Ad hoc routing protocols

A large number of papers have tried to simulate and analyze reactive and proactive protocols in general. The problem of analytical approaches lies in the fact that protocols like OLSR and AODV are complex and can be configured in many ways to achieve high performance in various scenarios. The behaviour of the protocols is mainly triggered by events like timeouts and the reception of routing messages. Nonetheless, the impact that these events have on both protocols is different. Timeouts have a great influence on the route establish and maintenance process in AODV which is characteristic for a reactive protocol. The one and two hop neighbour lists of OLSR are affected by timeouts which results in inefficient flooding of topology control messages as a consequence of errors in the multipoint relay set calculation. [8]

It is clear that the collected simulation results strongly depend on the implementation of the protocols and their configurations. In addition, the node density, the mobility model, and traffic patterns affect the results in many ways.

Mobile nodes, boats in my project, are within each other’s radio range can communicate directly, while distant mobile nodes rely on their neighbouring mobile nodes to forward packets. Each mobile node acts as either a host or router.

In this thesis, I have carried out a systematic performance study of two routing protocols for ad hoc networks namely Ad hoc On Demand Distance Vector (AODV) Routing protocol and Optimized Link State Routing (OLSR) protocol, and then evaluated performance between AODV and OLSR in maritime ad hoc network scenarios for some defined parameters.

Routing protocols are divided into two categories: Proactive and Reactive. Proactive routing protocols are table-driven protocols and they always maintain current up-to-date routing information by sending control messages periodically between the hosts which update their routing tables. The proactive routing protocols use link-state routing algorithms which frequently flood the link information about its neighbours. [9]Reactive or on-demand routing protocols create routes when they are needed by the source host and these routes are maintained while they are needed.

This chapter briefly describes the different ad hoc routing protocols that chosen to simulate and analyze, namely AODV and OLSR.

3.1 Desirable properties

If the conventional routing protocols do not meet our demands, we need a new routing protocol. The question is what properties such protocols should have? There are some of the properties [10] that are desirable [11]:

Distributed operation
The protocol should of course be distributed. It should not be dependent on a centralized controlling node. This is the case even for stationary networks. The difference is that nodes in an ad hoc network can enter/leave the network very easily and because of mobility the network can be partitioned.

Loop free
To improve the overall performance, we want the routing protocols to guarantee that the routes supplied are loop-free. This avoids any waste of bandwidth or CPU consumption.

Demand based operation
To minimize the control overhead in the network and thus not wasting network resources more than necessary, the protocol should be reactive. This means the protocol should only react when needed and that the protocol should not periodically broadcast control information.

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.

Security
The radio environment is especially vulnerable to impersonation attacks, so to ensure the wanted behaviour from the routing protocol, we need some sort of preventive security measures. Authentication and encryption is probably the way to go and the problem here lies within distributing keys among the nodes in the ad hoc network

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 uses sort of stand-by mode to save power. It is therefore important that the routing protocol has support for these sleep-modes. Considering the realistic conditions in my thesis, the mobile communication devices are set in the boats, so the power is not a problem.

Multiple routes

To reduce the number of reactions to topological changes and congestion multiple routes could be used. If one route has become invalid, it is possible that another stored route could still be valid and thus saving the routing protocol from initiating another route discovery procedure.

Quality of service support

Some sort of Quality of Service support is probably necessary to incorporate into the routing protocol. This has a lot to do with what these networks will be used for.

3.2 Ad hoc On Demand Distance vector – AODV

3.2.1 Description

The ad hoc On-Demand Distance Vector (AODV) routing protocol enables multi-hop routing between participating mobile nodes wishing to establish and maintain an ad hoc network. AODV is based upon the distance vector algorithm. The key feature is that AODV is reactive, as opposed to proactive protocols like DSDV. AODV only requests a route when needed and does not require nodes to maintain routes to destinations that are not actively used in communications. As long as the “endpoints” of a communication connection have valid routes to each other, AODV does not play any role.

Features of this protocol include loop freedom and that link breakages cause immediate notifications to be sent to the affected set of nodes, but only that set. Additionally, AODV has support for multicast routing and avoids the Bellman Ford “counting to infinity” problem [12]. The use of destination sequence numbers guarantees that a route is “fresh”.

The algorithm uses different messages to discover and maintain links. Whenever a node wants to try and find a route to another node, it broadcast a Route Request (RREQ) to all its neighbours. The RREQ propagates through the network until it reaches the destination or a node with a fresh enough route to the destination. Then the route is made available by unicasting a RREP back to the source.

The algorithm uses hello messages (a special RREP) that are broadcasted periodically to the immediate neighbours. These hello messages are local advertisements for the continued presence of the node and neighbours using routes through the broadcasting node will continue to mark the routes as valid. If hello messages stop coming from a particular node, the neighbour can assume that the node has moved away and mark that link to the node as broken and notify the affected set of nodes by sending a link failure notification (a special RREP) to that set of nodes.

Route table management

AODV needs to keep track of the following information for each route table entry:

- Destination IP Address: IP address for the destination node.
- Destination Sequence Number: Sequence number for this destination.
- Hop Count: Number of hops to the destination.

- Next Hop: The neighbour, which has been designated to forward packets to the destination for this route entry.
- Lifetime: The time for which the route is considered valid.
- Active neighbour list: Neighbour nodes that are actively using this route entry.
- Request buffer: Makes sure that a request is only processed once.

Route discovery
A node broadcasts a RREQ when it needs a route to a destination and does not have one available. This can happen if the route to the destination is unknown, or if a previously valid route expires. After broadcasting a RREQ, the node waits for RREP. If the rely is not received within a certain time, the node may rebroadcast the RREQ or assume that there is no route to the destination.

Forwarding of RREQs is done when the node receiving a RREQ does not have a route to the destination. It then rebroadcast the RREQ. The node also creates a temporary reverse route to the Source IP Address in its routing table with next hop equal to the IP address field of the neighbouring node that sent the broadcast RREQ. This is done to keep track of a route back to the original node making the request, and might be used for an eventual RREQ to find its way back to the requesting node. The route is temporary in the sense that it is valid for a much shorter time, than an actual route entry.

When the RREQ reaches a node that either is the destination node or a node with a valid route to the destination, a RREP is generated and unicasted back to the requesting node. While this RREP is forwarded, a route is created to the destination and when the RREP reaches the source node, there exists a route from the source to the destination.

Route maintenance
When a node detects that a route to a neighbour no longer is valid, it will remove the routing entry and send a link failure message, a triggered route reply message to the neighbours that are actively using the route, informing them that this route no longer is valid. For this purpose AODV uses a active neighbour list to keep track of the neighbours that are using a particular route. The nodes that receive this message will repeat this procedure. The message will eventually be received by the affected sources that can chose to either stop sending data or requesting a new route by sending out a new RREQ.

3.2.2 Properties

The advantage with AODV compared to classical routing protocols like distance vector and link-state is that AODV has greatly reduced the number of routing messages in the network. AODV achieves this by using a reactive approach. This is probably necessary in an ad hoc network to get reasonably performance when the topology is changing often.

The sequence numbers that AODV uses represents the freshness of a route and is increased when something happens in the surrounding area.

AODV only support one route for each destination. It should be fairly easy to modify AODV, so that it supports several routes per destination. Instead of requesting a new route when an old route becomes invalid, the next stored route to that destination could be tried. The probability for that route to still be valid should be rather high.

AODV uses hello messages at the IP-level. This means that AODV does not need support from the link layer to work properly. The hello messages add a significant overhead to the protocol.

AODV does not support unidirectional links. When a node receives a RREQ, it will setup a reverse route to the source by using the node that forwarded RREQ as next hop. This means that the route reply, in most cases is unicasted back the same way as the route request used.

3.3 Optimized Link State Routing – OLSR

3.3.1 Description

The Optimized Link State Routing Protocol (OLSR) is developed for mobile ad hoc networks. It operates as a table driven and proactive protocol, thus exchanges topology information with other nodes of the network regularly. The nodes which are selected as a multipoint relay (MPR) by some neighbour nodes 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 MPR. In route calculation, the MPRs are used to form the route from a given node to any destination in the network. The protocol uses the MPRs to facilitate efficient flooding of control messages in the network. OLSR inherits the concept of forwarding and relaying from HIPERLAN (a MAC layer protocol) which is standardized by ETSI. [13]

In wireless ad-hoc networks, there is different notion of a link, packets can and do go out the same interface; hence, a different approach is needed in order to optimize the flooding process. Using Hello messages the OLSR protocol at each node discovers 2-hop neighbour information and performs a distributed election of a set of multipoint distribution relays (MPRs). Nodes select MPRs such that there exists a path to each of its 2-hop neighbours via a node selected as an MPR. These MPR nodes then source and forward TC messages which contain the MPR selectors. This functioning of MPRs makes OLSR unique from other link state routing protocols in a few different ways: The forwarding path for TC messages is not shared among all nodes but varies depending on the source, only a subset of nodes source link state information, not all links of a node are advertised but only those which represent MPR selections.

3.3.2 Properties

Being a proactive protocol, routes to all destinations within the network are known and maintained before use. Having the routes available within the standard routing table can be useful for some systems and network applications as there is no route discovery delay associated with finding a new route.

The original definition of OLSR does not include any provisions for sensing of link quality; it simply assumes that a link is up if a number of hello packets have been received recently. This assumes that links are bi-modal (either working or failed), which is not necessarily the case on wireless networks, where links often exhibit intermediate rates of packet loss. Implementations such as the open source OLSRd (commonly used on Linux-based mesh routers) have been extended (as of v. 0.4.8) with link quality sensing. This is sometimes called "fish-eye" or Radio-Aware OLSR or

RA-OLSR and is one of the two protocols included in the 802.11s draft standard. It was influenced by the HSLS protocol.

Being a proactive protocol, OLSR uses power and network resources in order to propagate data about possibly unused routes. While this is not a problem for wired access points and laptops, it makes OLSR unsuitable for sensor networks.

Being a link-state protocol, OLSR requires a reasonably large amount of bandwidth and CPU power to compute optimal paths in the network. In the typical networks where OLSR is used (which rarely exceed a few dozens of nodes), this does not appear to be a problem.

By only using MPRs to flood topology information, OLSR removes some of the redundancy of the flooding process, which may be a problem in networks with large packet loss rates - however the MPR mechanism is self-pruning (which means that in case of packet losses, some nodes which would not have retransmitted a packet, may do so).

OLSR makes use of "Hello" messages to find its one hop neighbours and its two hop neighbours through their responses. The sender can then select its multipoint relays (MPR) based on the one hop node which offer the best routes to the two hop nodes. Each node has also an MPR selector set which enumerates nodes that have selected it as an MPR node. OLSR uses Topology Control (TC) messages along with MPR forwarding to disseminate neighbour information throughout the network. Host Network Address (HNA) messages are used by OLSR to disseminate network route advertisements in the same way TC messages advertise host routes.

4 Simulations and numerical results

The protocols that the thesis has simulated are AODV and OLSR.

4.1 Measurements

Before the thesis go into the actual simulations, I will discuss which parameters that are interesting to measure when studying routing protocols in a maritime ad hoc network. There are two main performance measures that are substantially affected by the routing algorithm, the average end-to-end throughput and the average end-to-end delay.

4.1.1 Quantitative metrics

The measurements that the thesis has conducted can be seen from two angles: externally and internally. The external view is what the application sees and the internal view is how the routing protocol behaves. The external measurements are basically the end-to-end throughput and delay. The internal behaviour can further be divided into routing efficiency and routing accuracy.

- Routing Efficiency: How much of the sent data is actually delivered to the destination? How much routing overhead is required to find routes?
- Routing Accuracy: How accurate, measured in number of hopes are the supplied routes compared to the optimal shortest path.

4.1.2 Parameters

The metrics have to be measured against some parameter that describe the characteristic behaviour of an ad hoc network and can be varied in a controlled way. The parameters that I have chosen to simulate with are:

- Mobility, which probably is one of the most important characteristics of an ad hoc network. This will affect the dynamic topology.
- Offered network load. This can be characterized by three parameters: packet size, number of connections and the rate that we are sending the packets with.
- Network size: number of nodes, the size of the area that the nodes are moving within. The network size basically determines the connectivity.

4.1.3 Scenario

The scenario is very important for the simulation. In this thesis, I have discussed three kinds of scenarios based on maritime situations. The scenarios that I created then analyzed in terms of unreachable hosts. And all scenarios with extremely high degree of unreachable hosts where discarded.

4.2 Simulation scenarios

In this chapter I am going to describe how the simulations work. Here will be three different types of simulations:

- Scenario1, One-mobile-node simulations:
  A model with the topology consists of three stationary nodes and one mobility node.
- Scenario2, Two-mobile-nodes simulations:
  A model with the topology consists of three static nodes and two mobility nodes which move on the same route but in opposite directions.
- Scenario3, Realistic simulations:
  A model with the topology based on the realistic boat traffic trace captured in some seaport of Bergen.

In all scenarios simulated in my thesis, I use UPD traffic type. Why the scenario does not use TCP for the simulation? Because I want to get a general view of how the routing protocol behaves, and TCP uses flow control, retransmit features and so on. The communication pattern is randomly created. The parameters are specified when randomizing the communication pattern. In these simulations, I want to analyse how the protocols affect the communication and why different simulation performance duel to different protocols at the same scenario.

These simulations do not give a general view of the protocols, but instead test certain characteristics of the protocols in certain scenarios.

4.3 Scenario 1 - One mobile node simulation

As the initial scenario, here is the simplest model with three stationary nodes as access points and one moving node as a boat, as shown in the below.

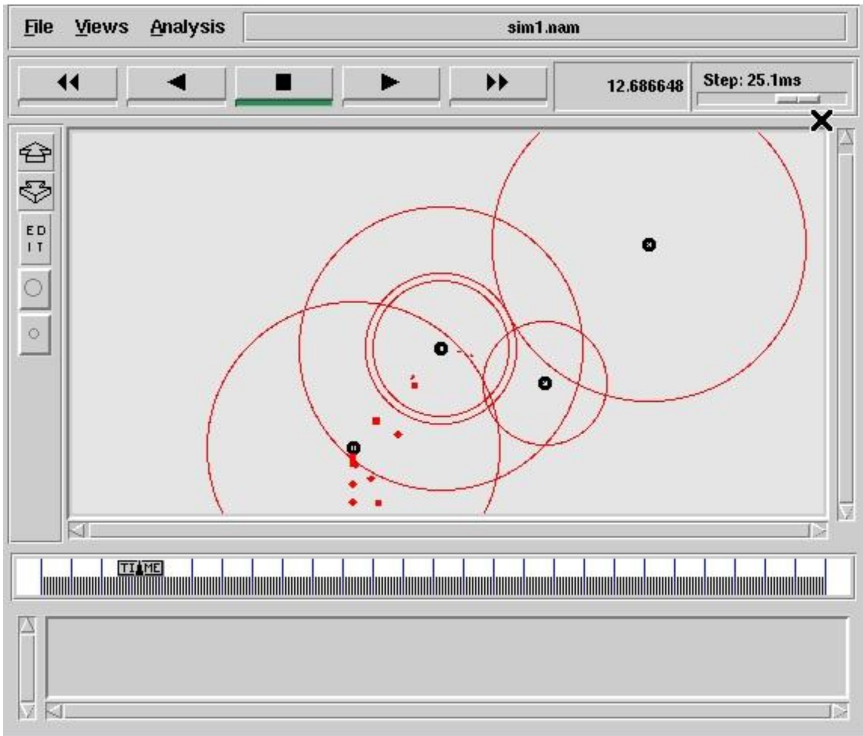
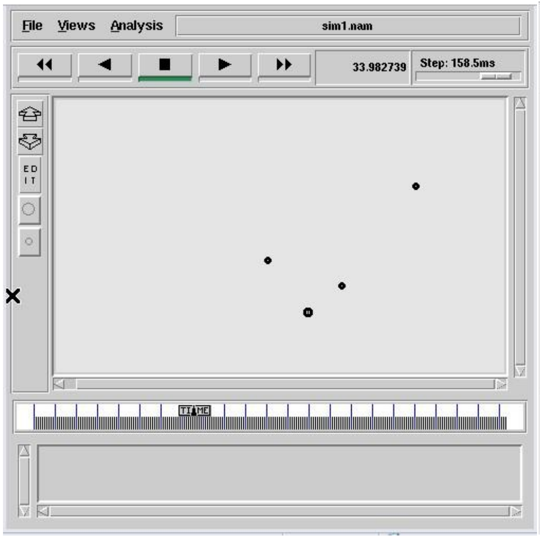


Figure 17 Topology for scenario 1 of AODV

In this simulation, the parameters are set as follows:

- Number of nodes: This is constant during the simulation. This scenario uses 4 nodes for the simulation, one is mobility and the other three are stationary.
- Mobility state: Node 0 moves following a fixed route and changes speed and direction after simulation starts 35sec and 50sec.
- Environment size: Determines the size of the environment. I use a size of 1000*500 meters for the simulation.
- Pause time: Pause time is the time for which a node stands still before starts moving. In this scenario, here is no pause time.
- Network interface type: Phy/WirelessPhy, 914MHz Lucent WaveLAN DSSS radio interface.
- Mac type: 802.11 Mac-layer type.
- Link layer: Supported.
- Protocols: AODV and OLSR.
- Traffic type: Constant Bit Rate,
- Simulation time: 100sec
- Bandwidth: 10Mbit



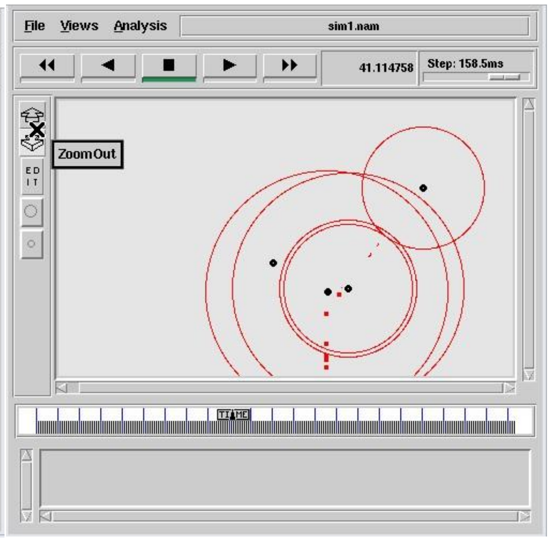
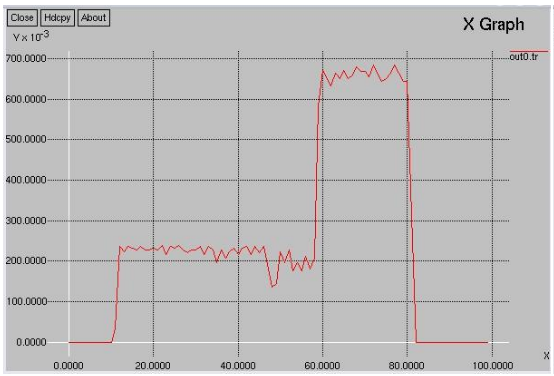


Figure 18 Topology for scenario 1 of OLSR, time before 40sec

Figure 19 Topology for scenario 1 of OLSR, time after 40sec

An UDP connection is set up between node 0 and node 3. On top of this I run constant-bit-rate (CBR) traffic.

In this scenario,
Packet size = 1480bps,
CBR interval = 0.0066 seconds,
The source bit rate = 1500*8/0.006=1 776 Kbps.



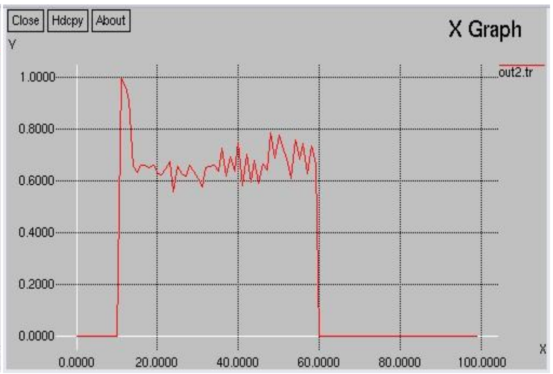
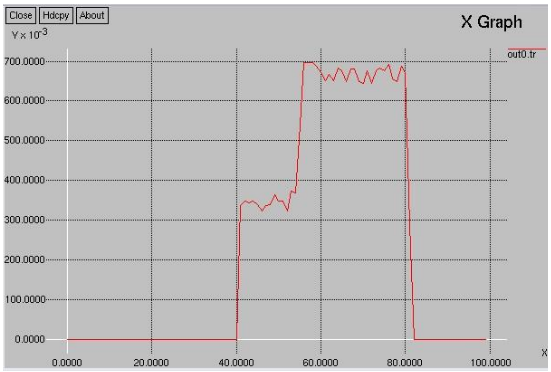


Figure 20 Bandwidth for scenario 1 of AODV

Figure 21 Packet Loss Ratio for scenario 1 of AODV

Total bandwidth = 3 129 500 Bps,
Average bandwidth = 250.36 Kbps.



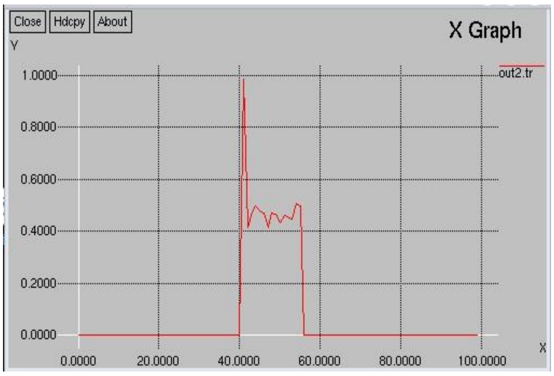


Figure 22 Bandwidth for scenario 1 of OLSR

Figure 23 Packet Loss Ratio for scenario 1 of OLSR

Bandwidth for scenario 1 of OLSR

Total bandwidth = 2 807 500 Bps

Average bandwidth = 224.60 Kbps

<u>Summary:</u>

Compared to the same simulation environment, both the total bandwidth and the average bandwidth are reduced. As can read from the graph, there are no packets sent before about 40sec after the simulation starts when the protocol is OLSR, instead AODV is 10sec. This can be explained that OLSR is a proactive routing protocol. OLSR uses some time to build the Multipoint Relays (MPR-sets). The idea of MPR is reducing the same broadcast in some regions in the network. When the nodes are stable, better performance throughout the scenario is achieved in OLSR compared to AODV, but worse when the commutating node is moving.

For the packet loss ratio we can find that OLSR has less packet loss than AODV.

When the protocol is AODV, it is obvious clear that AODV isn't that "smart" at choosing route. As we can see, during the time when node 0 moves pass node 1 following the route towards node 3, it "should" refresh routing table to choose optimal path, the shorter path, "node 0 – 2 – 3" instead of "node 0 – 1 – 2 – 3". But the truth isn't what we expect. Node 0 use "old" routing table till it communicates with node 3 directly.

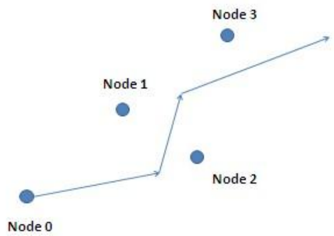


Figure 24 Simulation of movement

So, I assume the protocols can become "smarter" with changing some parameter, such as interval time.

➢ First I try AODV; I decreased interval time to half, to check if it would help to do with get protocol smarter.

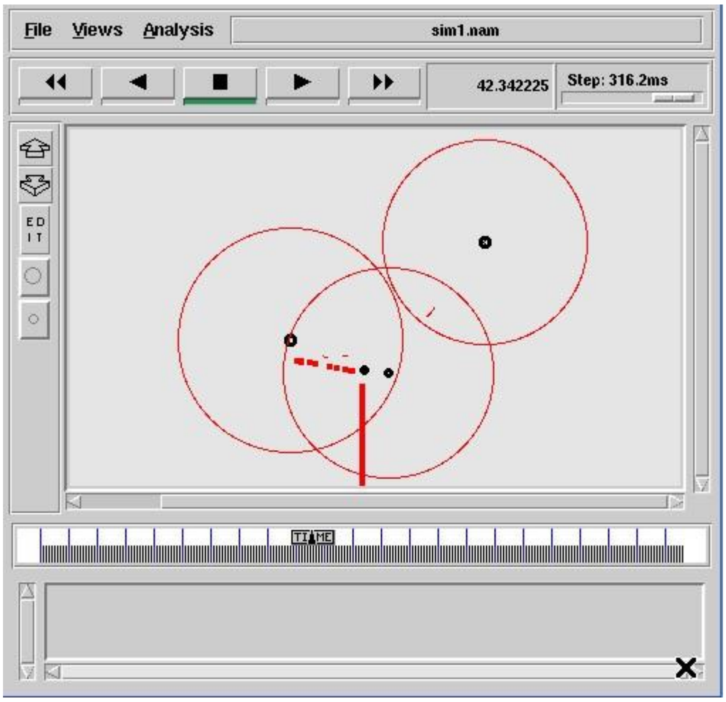
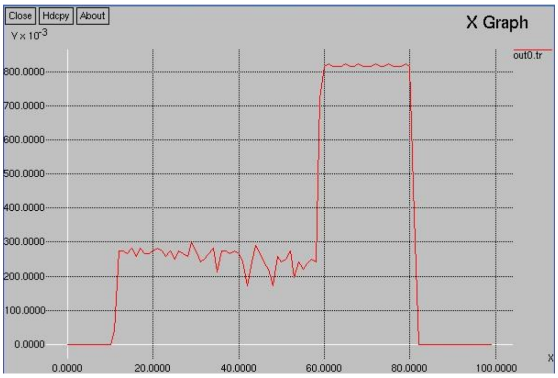


Figure 25 Topology for interval-decreased-scenario 1 with AODV



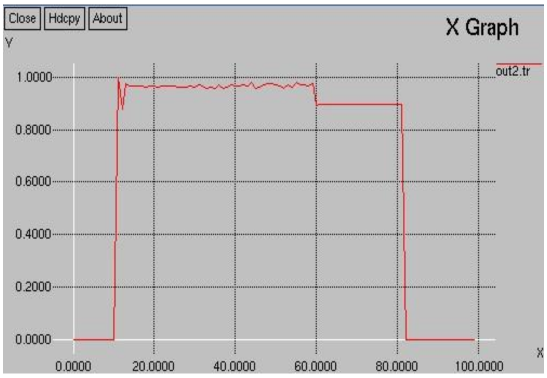


Figure 26 Bandwidth for interval-decreased-scenario 1 with AODV

Figure 27 Packet Loss Ratio for interval-decreased-scenario 1 with AODV

From graphs, here are only 2 "jumps" in bandwidth graph which means node 0 changed routing paths only 2 times not that "smart" to change to shorter path even when node 0 comes to node 2 closely enough. For the reason of that is the algorithm of AODV protocol itself.

➢ Second, I try OLSR with interval time decreased.

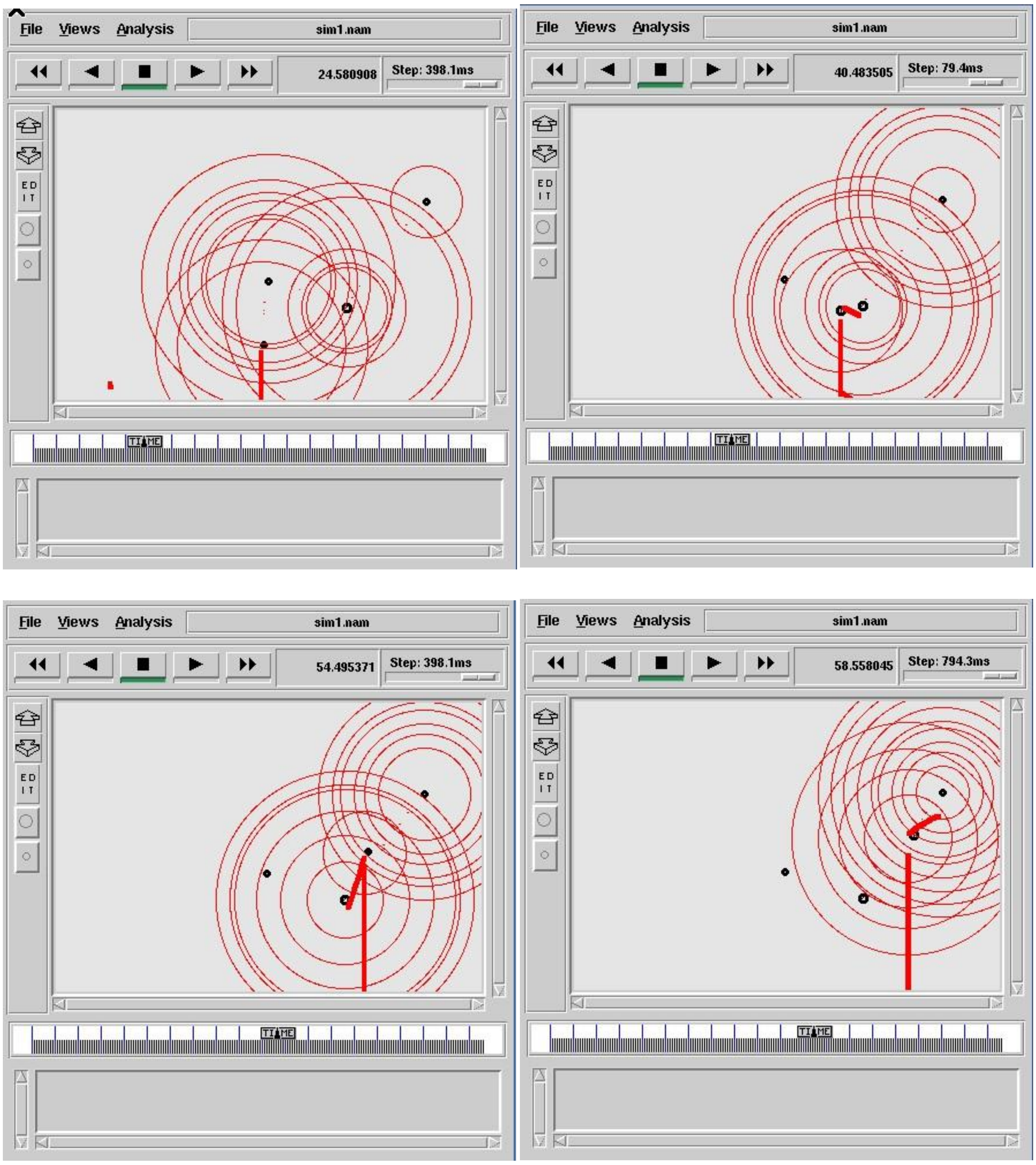
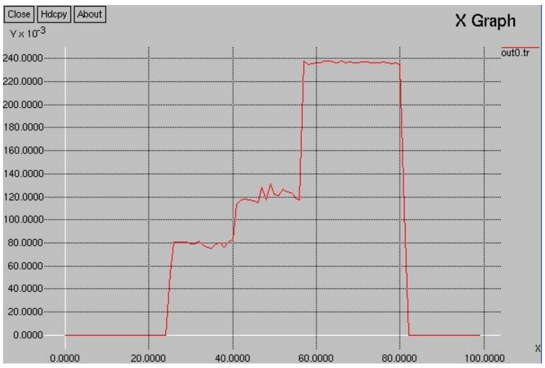


Figure 28 Topology for interval-decreased-scenario 1 with OLSR



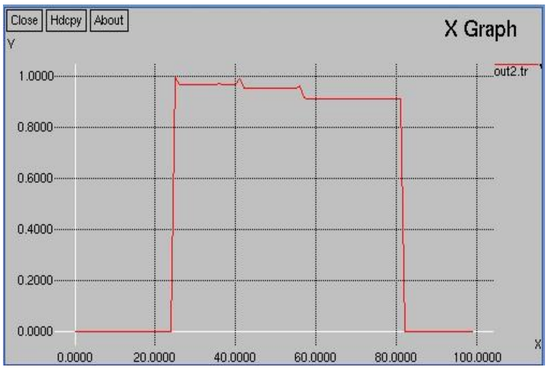


Figure 29 Bandwidth of interval-decreased-scenario 1 with OLSR

Figure 30 Packet Loss Ratio of interval-decreased-scenario 1 with OLSR

Finally, we get what we except result here. At 25sec, node 0 changed packets translation path from “0-1-2-3” to “0-2-3”. At 55sec, node 0 changed packets translation path from “0-2-3” to “0-3”. It is also clearly found from the graph shows that 3 “jumps” in bandwidth graph means routing path has changed 3 times throughout the whole simulation process.

However, packet loss increases meanwhile. Then the issue here is to try to find optimal values for these parameters, such as interval time. The choice of these parameters is also very dependent on the behaviour the scenarios are desired.

Now, let’s try to change different parameters to see how the specified parameters affect the performance in different protocols, and compare the results to the results from initial scenario.

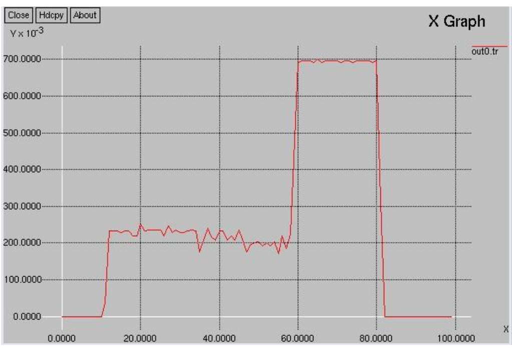
4.3.1 Change of data rates

- ***Source data rates***

The model is much the same like the initial one, but the source data will be changed, to examine in which way this affects the throughput and packet loss and different performance in different protocols.

Here I use the way of changing two parameters, pocketsize and interval for the current CBR.
The bit rate in the initial code is very high. Now we want to investigate what happens when we decrease the bit rate.

I reduced the packet size to 480bps and increased the CBR-interval to 0.02 seconds. This gives source bit rate 480*8/0.02=192Kbps



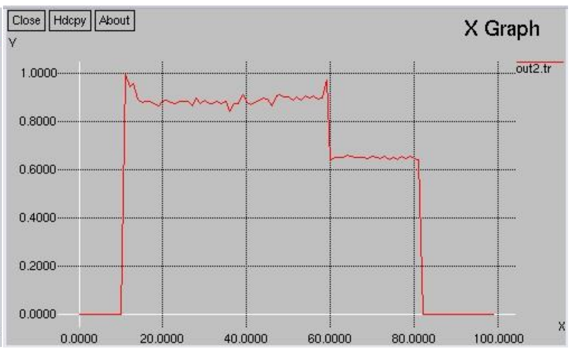
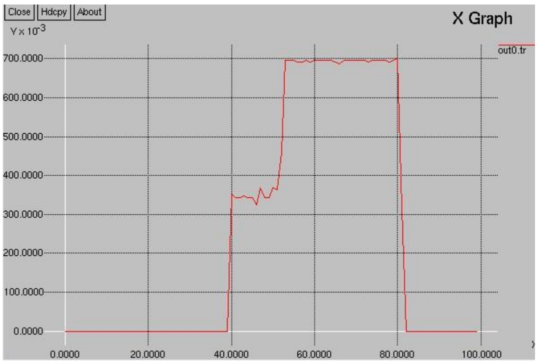


Figure 31 Bandwidth for scenario 1 with AODV

Figure 32 Packet Loss Ratio for scenario 1 with AODV

Total bandwidth = 32 090 000 Bps
Average bandwidth = 256.72 Kbps



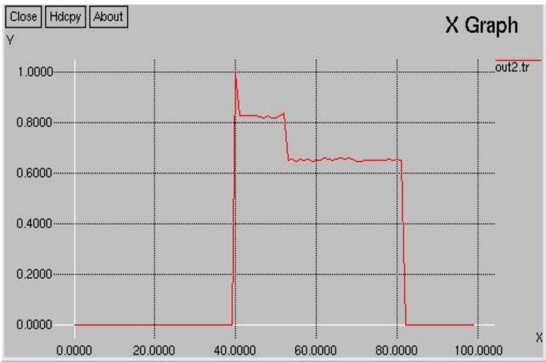


Figure 33 Bandwidth for scenario 1 of OLSR
Figure 34 Packet Loss Ratio for scenario 1 of OLSR

Total bandwidth = 3 053 000 Bps
Average bandwidth = 244.24 Kbps

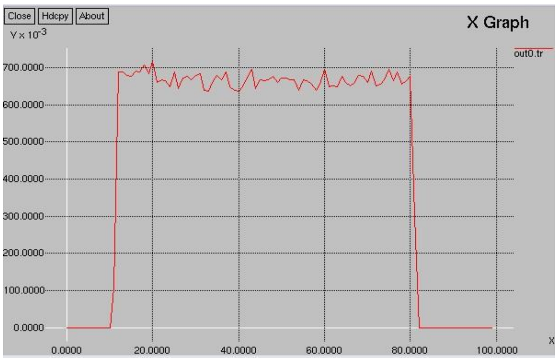
Summary:

When the protocol is AODV, read from graphs, we can see that bandwidth performance is almost 10 times than initial case and average bandwidth nearly no difference, and packet loss ratio increased hugely. Because the interval is increased, means link breakages are detected later than before, which makes the protocol is sending packets on a broken route that it thinks is valid, the packets in the buffers were dropped because of congestion and timeouts.
For OLSR, both total bandwidth and average bandwidth didn't change that much, but packet loss ratio increased hugely also.

➢ ***Transmission rates***

In this part, I look into the effects of increasing the transmission rate. The transmission data rate for the DATA is changed to 10Megabit/s, while BasicRate for RTS/CTS, and ACK remained unchanged.
This is done by changing the parameter Mac/802_11 set DataRate_ 1e7.

As expected, the total and average bandwidth grows. This can be verified by the bandwidth graph underneath. The total bandwidth is now 5 812 500 bps, and the average bandwidth is 465 Kbps.



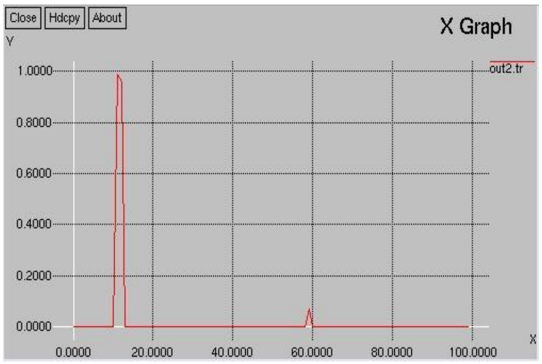
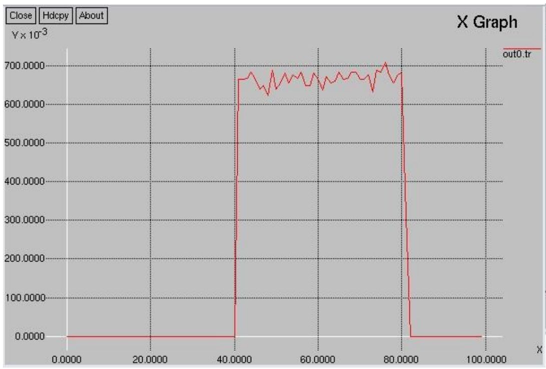


Figure 35 Bandwidth for scenario 1 of AODV

Figure 36 Packet Ratio Loss for scenario 1 of AODV

The second graph shows the packet loss ratio. It much decreases than the initial one.



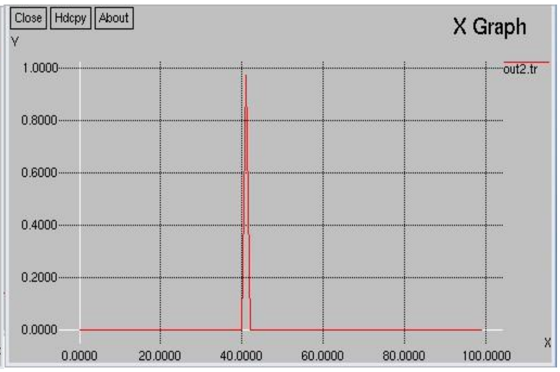


Figure 37 Bandwidth for scenario 1 of OLSR

Figure 38 Packet Loss Ratio for scenario 1 of OLSR

The changes are more or less like AODV. The total and average bandwidth grow, meanwhile, the packet loss ratio hugely decreases. Now, the total bandwidth are 3 371 000 bps, average bandwidth are 269.68 Kbps.

Summary:

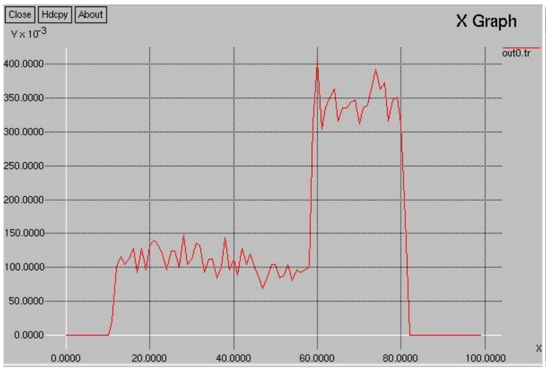
In the case of changing of data rate, the performance of change form AODV and OLSR is almost the same.

4.3.2 Change the number of traffic flows

In this part, I set up an additional UDP-flow, equal to the first one.

The graphs demonstrate that this has a very large impact on both the throughput and the packet loss. In this case, the total bandwidth is 1 601 000 Bps, and the average bandwidth is 128.08 Kbps.

We can see that the bandwidth has decreased to about the half the initial scenario, and that packet loss ratio is going to close 1. This is due to the fact that we are trying to transmit way much more than the saturation throughput.



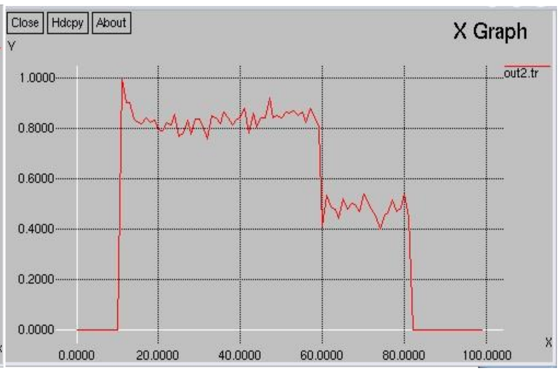
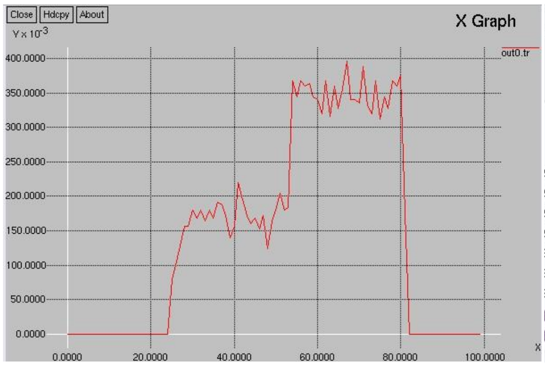


Figure 39 Bandwidth for scenario 1 of AODV

Figure 40 Packet Loss Ratio for scenario 1 of AODV

Instead of OLSR, this has a very large impact on both the throughput and the packet loss. The bandwidth has decreased to about the half the initial scenario, and that packet loss ratio is going to close 1. In this case, total bandwidth is 1 800 500 Bps, and average bandwidth is 144.04 Kbps.



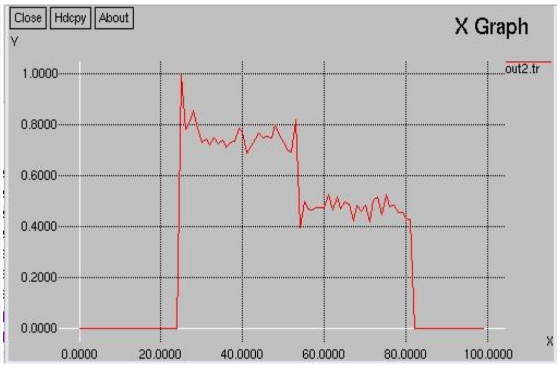


Figure 41 Bandwidth for scenario 1 of OLSR

Figure 42 Packet Loss Ratio for scenario 1 of OLSR

Summary:

In the case of changing of traffic flows, the performance shows that OLSR has larger impact than AODV.

4.3.3 Change the number of nodes

In this part, I add simulation nodes to 10 instead of 4. In this case, the network topology is also changed. We will investigate in which way the change of network topology influences the throughput and packet loss. The network topology is changed according to the figure underneath. The traffic-flow is set up between nodes 0 and 9.

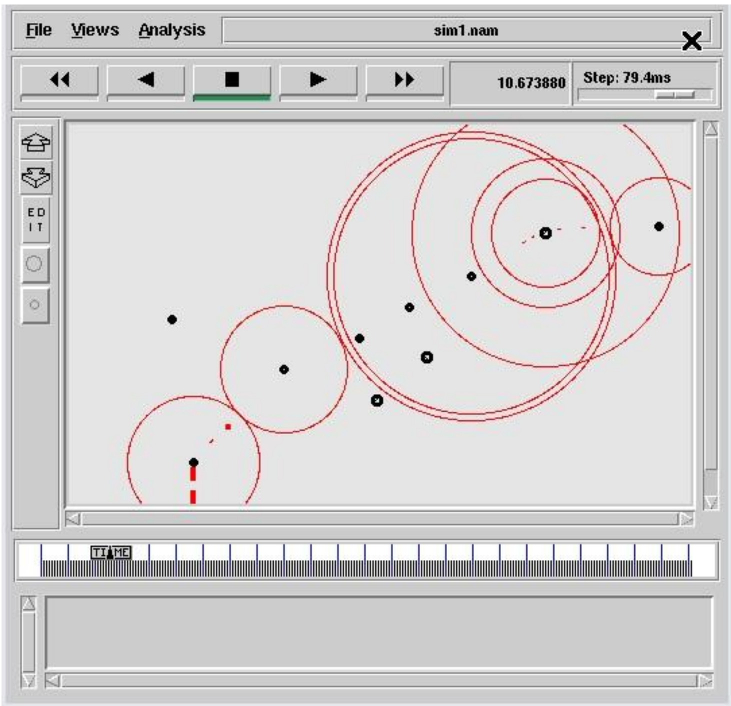
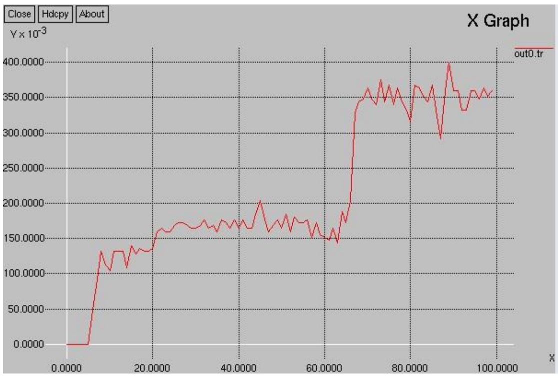


Figure 43 Topology for 10-node-scenairo 1 with AODV



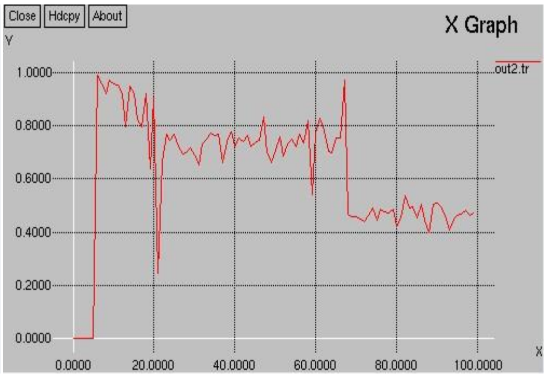
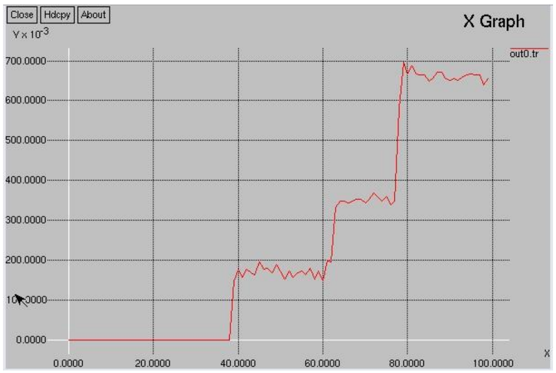


Figure 44 Bandwidth for 10-node-scenario 1 of AODV

Figure 45 Packet Loss Ratio for 10-node-scenario 1 of AODV

Total Bandwidth = 2 641 500 Bps

Average Bandwidth = 211.32 Kbps



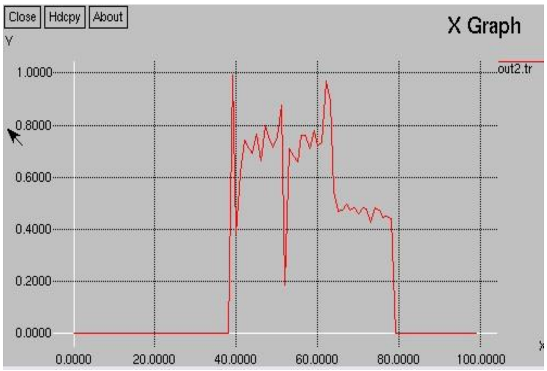


Figure 46 Bandwidth for 10-node-scenario 1 with OLSR
Figure 47 Packet Loss Ratio for 10-node-scenario 1 with OLSR

Total bandwidth = 2 981 000 Bps
Average bandwidth = 238.48 Kbps

Summary:

The performance of AODV is worse than OLSR when the joined-nodes increases, when means the node density increases. But OLSR shows consistent performance.

Since every time when the communication is built or re-built, OLSR needs more time to “refresh” routing table for the reason of proactive protocol characteristic, but it does better performance after the connection is established.

Explain the results:

Neither of them is sufficient well for communication in this scenario.

In AODV, the network is silent until a connection is needed. At that point the network node that needs a connection broadcasts a request for connection. After the connection has been built, there is no renewing route table anymore until the link breaks. Basically, there are no new routes as long as old routes hold.

In OLSR, since it is a proactive protocol, routes to all destinations within the network are known and maintained before use. The route table updates as network topology changes.

4.4 Scenario 2 – Two mobile node simulation

In scenario 2, here are two nodes move toward to each other. Ideally, multi-hop would be used first, then direct link when node0 and node4 get close to each other.

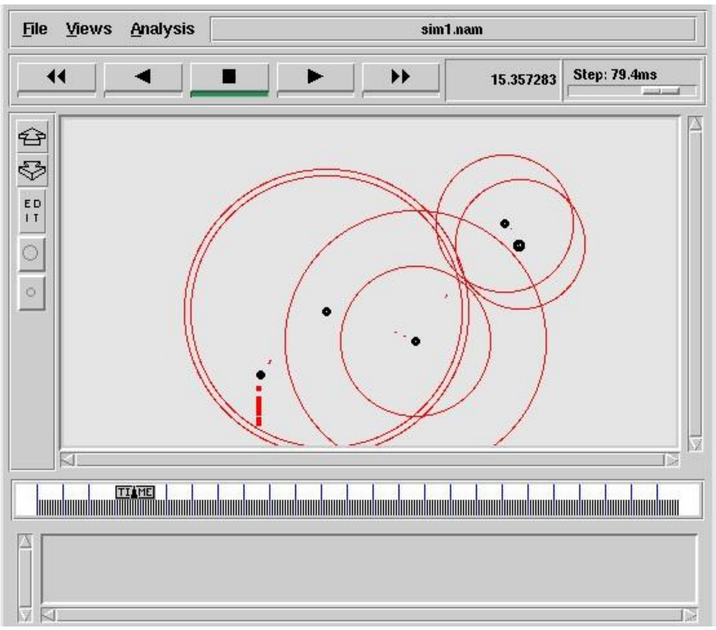
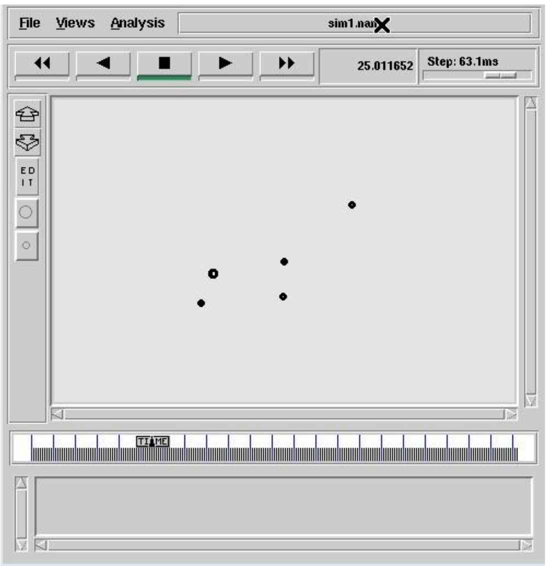


Figure 48 Topology for scenario 2 with AODV

In this scenario, the parameters are set as follows:

- Number of nodes: This is constant during the simulation. This scenario uses 5 nodes for all simulation, two are mobility and the other three are stationary.
- Mobility state: Node 0 and node 4 moves oppositely following a fixed route and keeps a fixed speed and direction till the end of simulation.
- Environment size: Determines the size of the environment. I use a size of 1000*500 meters for the simulation.
- Pause time: Pause time is the time for which a node stands still before starts moving. In this scenario, here is no pause time.
- Network interface type: Phy/WirelessPhy, 914MHz Lucent WaveLAN DSSS radio interface.
- Mac type: 802.11 Mac-layer type.
- Link layer: Supported.
- Protocols: AODV and OLSR.
- Traffic type: Constant Bit Rate,
- Simulation time: 100sec
- Bandwidth: 10Mbit



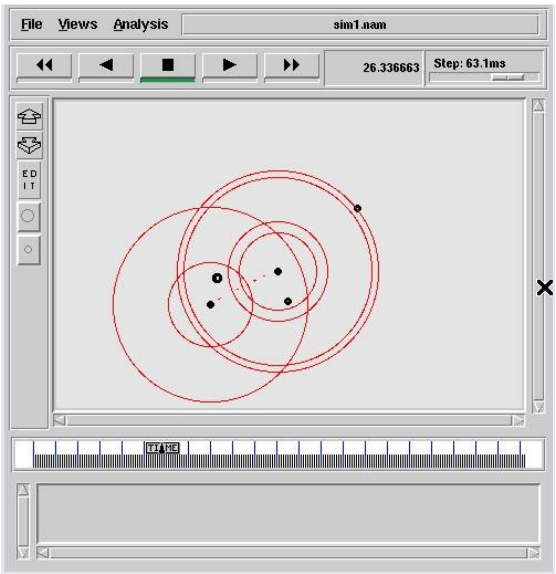
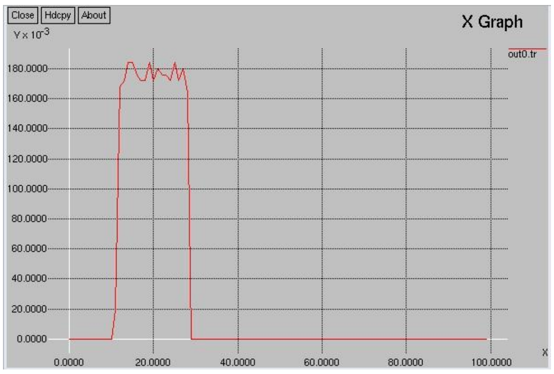


Figure 49 Topology for scenario 2 with OLSR, time before 26sec

Figure 50 Topology for scenario 2 with OLSR, time after 26sec

An UDP connection is set up between node 0 and node 3. On top of this I run constant-bit-rate (CBR) traffic.

In this scenario,
Packet size = 1480bps,
CBR interval = 0.0066 seconds,
The source bit rate = 1480*8/0.0066=1760 Kbps.



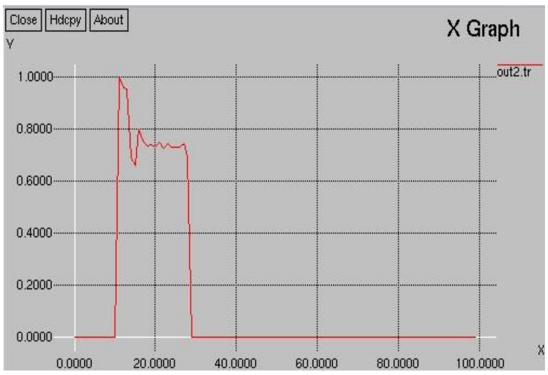
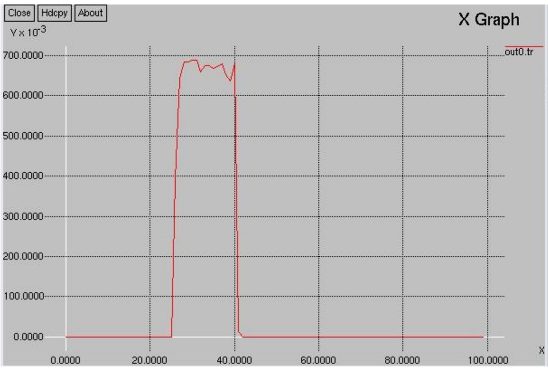


Figure 51 Bandwidth for scenario 2 with AODV

Figure 52 Packet Loss Ratio for scenario 2 with AODV

Total bandwidth = 376 000 Bps,
Average bandwidth = 30.08 Kbps.



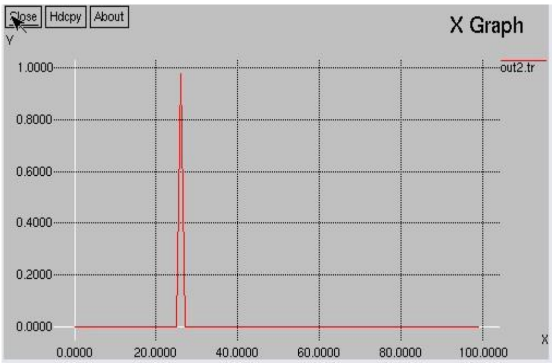


Figure 53 Bandwidth for scenario 2 of OLSR
Figure 54 Packet Loss Ratio for scenario 2 of OLSR

Total bandwidth = 1 224 000 Bps
Average bandwidth = 97.92 Kbps

Summary:
In this case, OLSR needs some more time to build routing table before they start send packets. Performance of OLSR is obviously better than AODV.

From the graph, we can see network topology maintain the initial routing path throughout the simulation. However, the protocol “should” be smarter to change optimal path when node 0 and node 1 come to enough short distance. See the supposed pictures below.

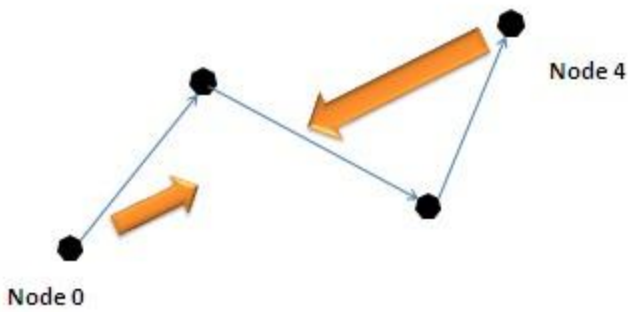
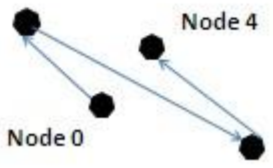


Figure 55 Simulation of protocol routing path while movement



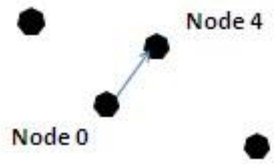
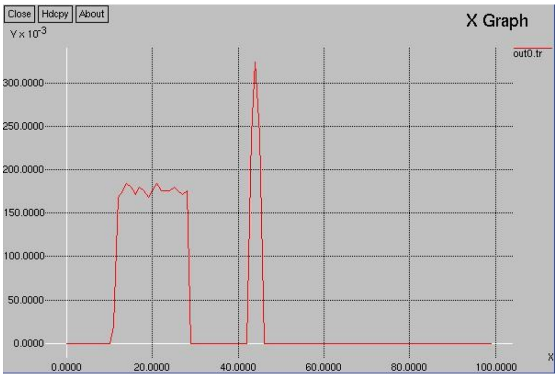


Figure 56 Real protocol routing path

Figure 57 Expected protocol routing path

As we can see, node 0 “should” change routing path like the supposed pictures show, but the initial simulation didn’t perform that .To make protocol working “smarter”, I try to change some parameter, interval time, to check if it can get the result what we expect.

➢ First, I try to decrease interval time in AODV.



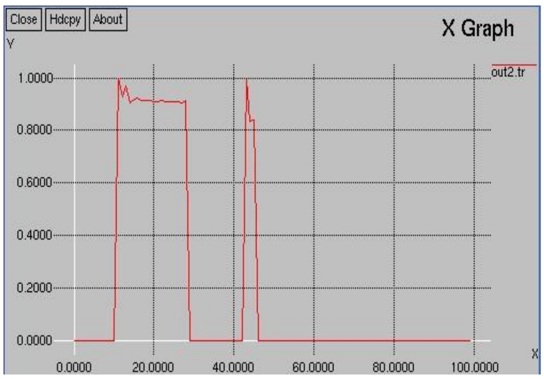


Figure 58 Bandwidth for decreased-interval-scenario 2 of AODV

Figure 59 Packet Loss Ratio for decreased-interval-scenario 2 of AODV

Total bandwidth = 477 500 Bps

Average bandwidth = 38.20 Kbps

However, the result is still not what we expect.

➢ Then, let’s try to do that change in OLSR.

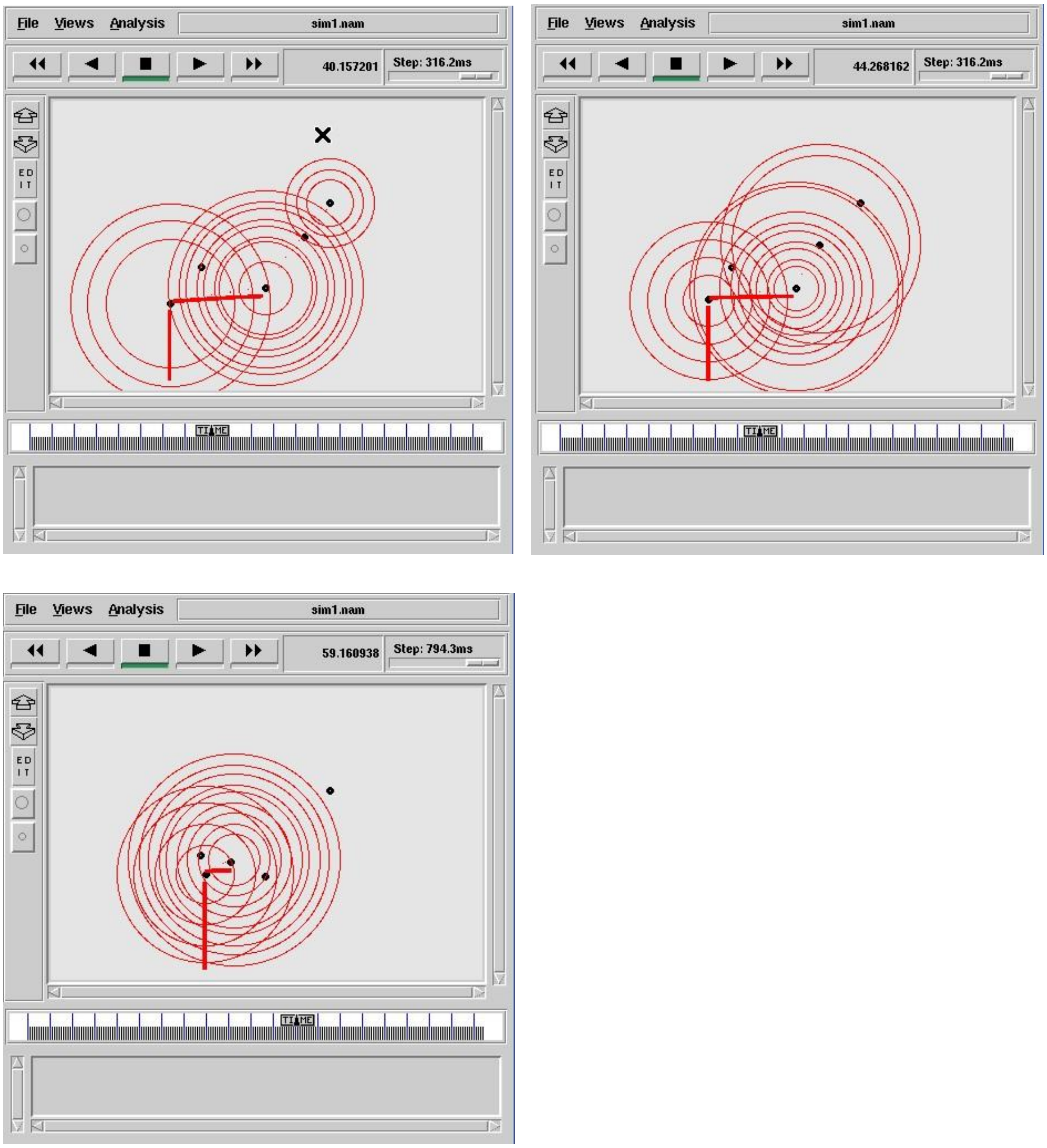
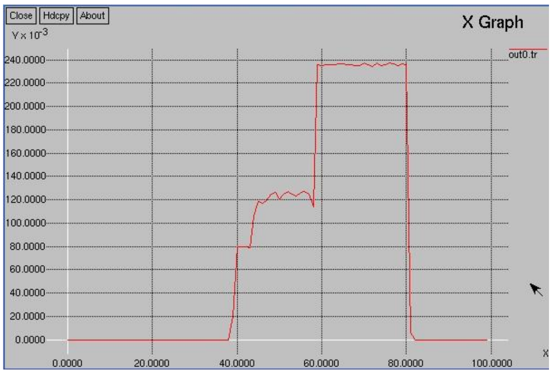


Figure 60 Topology for interval-decreased-scenario 2 with OLSR



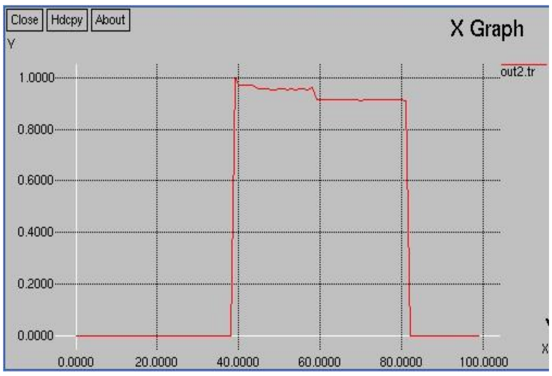


Figure 61 Bandwidth of interval-decreased-scenario 2 with OLSR

Figure 62 Packet Loss Ratio of interval-decreased-scenario 2 with OLSR

From graph, we can see that bandwidth line has 3 times “jumps” which means the routing path has changed 3 times. The result came to what we expect. At 40sec, the packet deliver path follows node “0-1-2-3-4”, “0-1-2-4” then “0-4”.

The basic simulation seems “smarter” than it did before, but it did much worse in packet loss ratio. So the proactive approach is not acceptable when interval time decreases to an unrealistic number (0.0002 second I used here). For an optimal value for these parameters, the choice of that is also dependent on the behaviour we are try to desire.

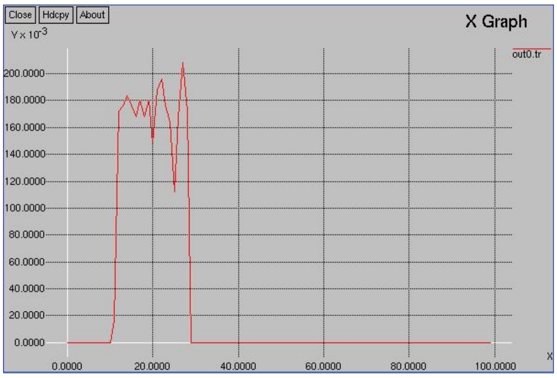
4.4.1 Change of data rates

➢ ***Source data rates***

The model is much the same like the initial one, but the source data will be changed, to examine in which way this affects the throughput and packet loss and different performance in different protocols.

Here I use the way of changing two parameters, pocketsize and interval for the current CBR.
The bit rate in the initial code is very high. Now we want to investigate what happens when we decrease the bit rate.

I reduced the packet size to 480bps and increased the CBR-interval to 0.02 seconds. This gives source bit rate 480*8/0.02=192Kbps



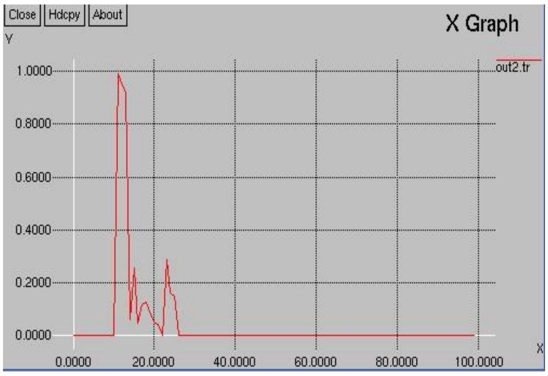
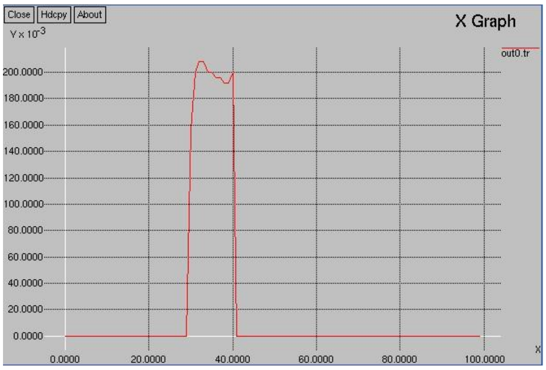


Figure 63 Bandwidth of scenario 2 with AODV

Figure 64 Packet Loss Ratio of scenario 2 with AODV

Total bandwidth = 369 000 Bps

Average bandwidth = 29.52 Kbps



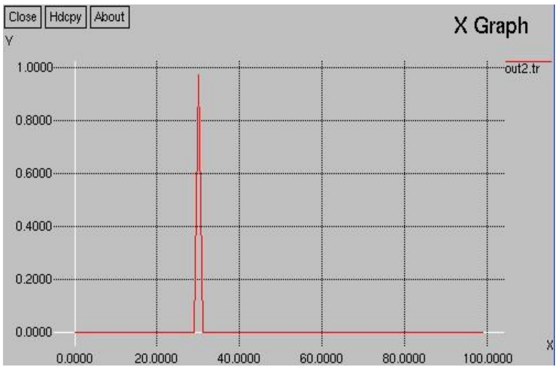


Figure 65 Bandwidth of scenario 2 with OLSR

Figure 66 Packet Loss Ratio of scenario 2 with OLSR

Total bandwidth = 268 520 Bps

Average bandwidth = 21.48 Kbps

Summary:

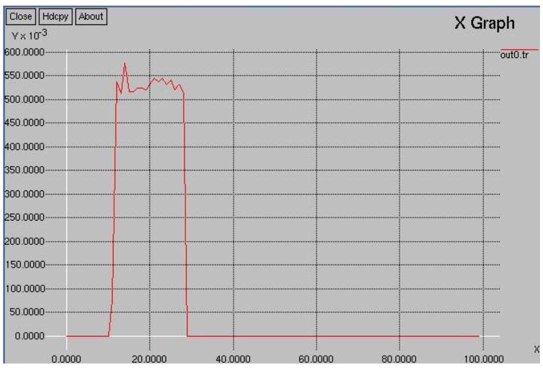
In this case, performance of AODV changed little, but both total and average bandwidth of OLSR decreased much than initial.

- ***Transmission rates***

In this part, I look into the effects of increasing the transmission rate. The transmission data rate for the DATA is changed to 10Megabit/s, while BasicRate for RTS/CTS, and ACK remained unchanged.

This is done by changing the parameter Mac/802_11 set DataRate_ 1e7.

As expected, the total and average bandwidth grows. This can be verified by the bandwidth graph underneath.



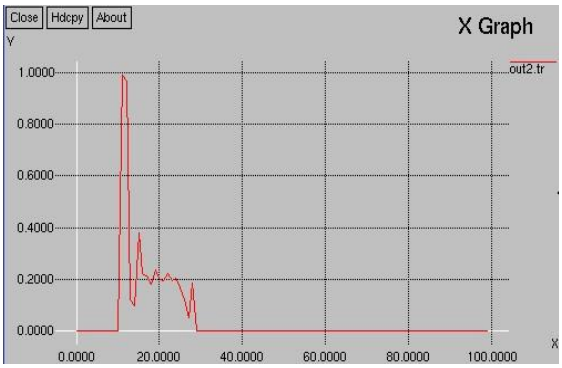


Figure 67 Bandwidth of scenario 2 with OLSR

Figure 68 Packet Loss Ratio of scenario 2 with OLSR

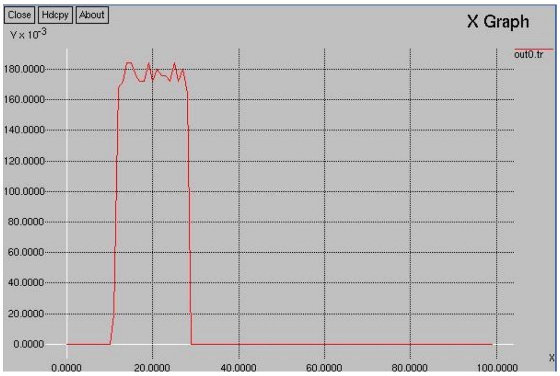
Total bandwidth = 1 136 500 Bps

Average bandwidth = 90.92 Kbps

Summary:
In this case, the performance of OLSR nearly changed.

4.4.2 Change the number of traffic flows

In this part, I set up an additional UDP-flow, equal to the first one.



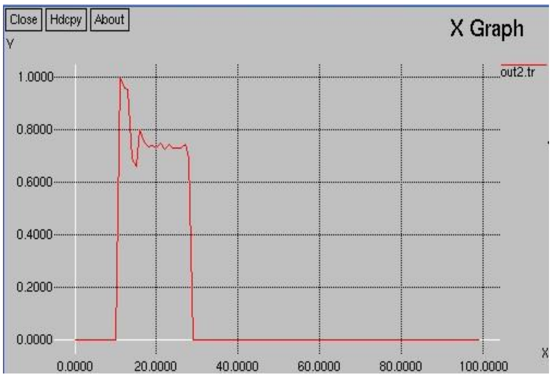
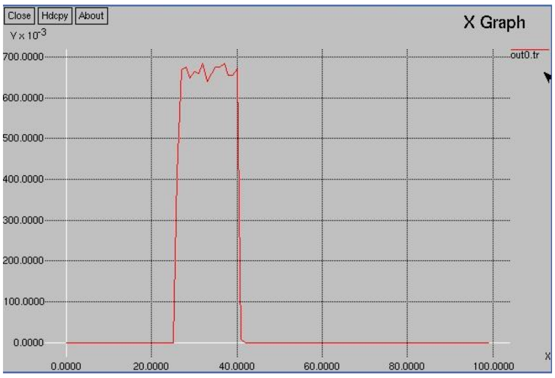


Figure 69 Bandwidth of scenario 2 with AODV

Figure 70 Packet Loss Ratio of scenario 2 with AODV

Total bandwidth = 376 000 Bps
Average bandwidth = 30.08 Kbps



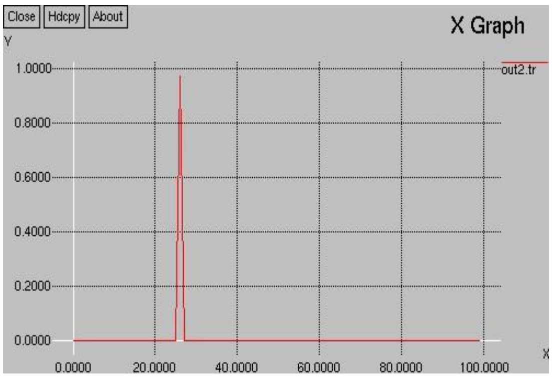


Figure 71 Bandwidth of scenario 2 with OLSR

Figure 72Packet Loss Ratio of scenario 2 with OLSR

Total bandwidth = 1 217 500 Bps
Average bandwidth = 97.4 Kbps

Summary:
In this case, no changes happened at all both of AODV and OLSR. Because of saturation of bandwidth we have supported in initial case already.

4.4.3 Change the number of nodes

In this part, I add simulation nodes to 10 instead of 5. In this case, the network topology is also changed. We will investigate in which way the change of network topology influences the throughput and packet loss. The network topology is changed according to the figure underneath. The traffic-flow is set up between nodes 0 and 9.

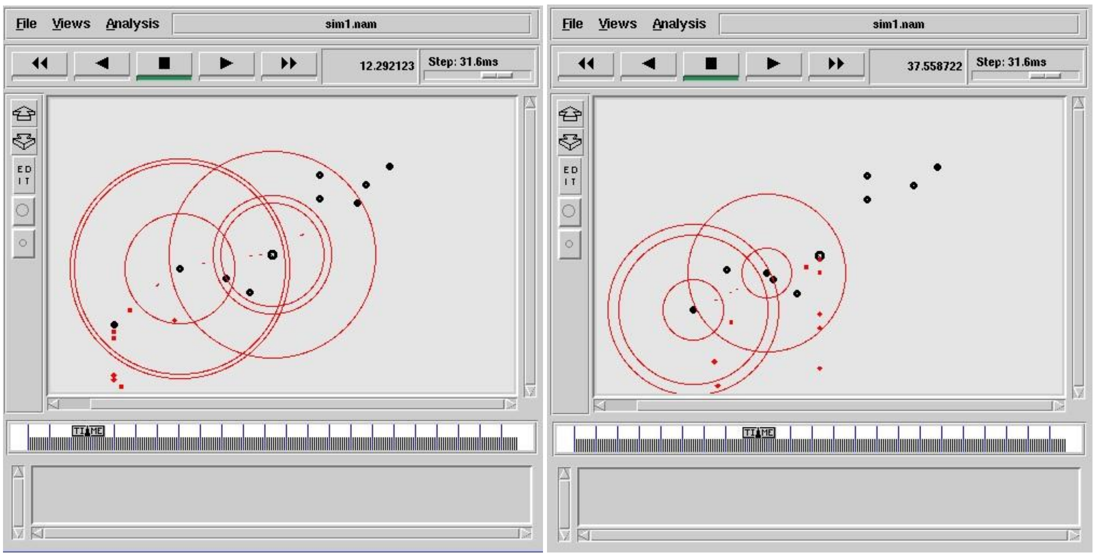
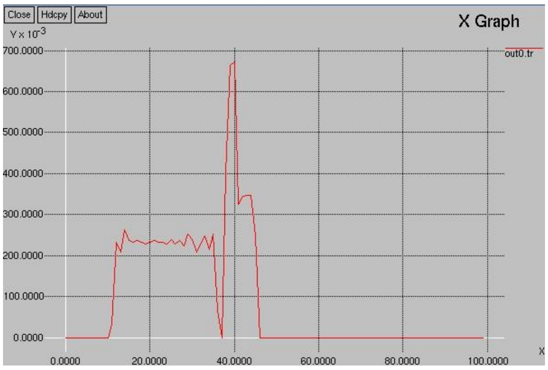


Figure 73 Topology of 10-node-scenairo 2 with AODV



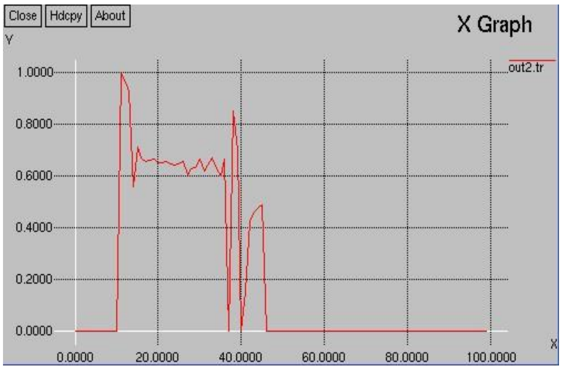


Figure 74 Bandwidth of 10-node-scenario 2 with AODV

Figure 75 Packet Loss Ratio of 10-node-scenario 2 with AODV

Total bandwidth = 1 131 000 Bps

Average bandwidth = 90.48 Kbps

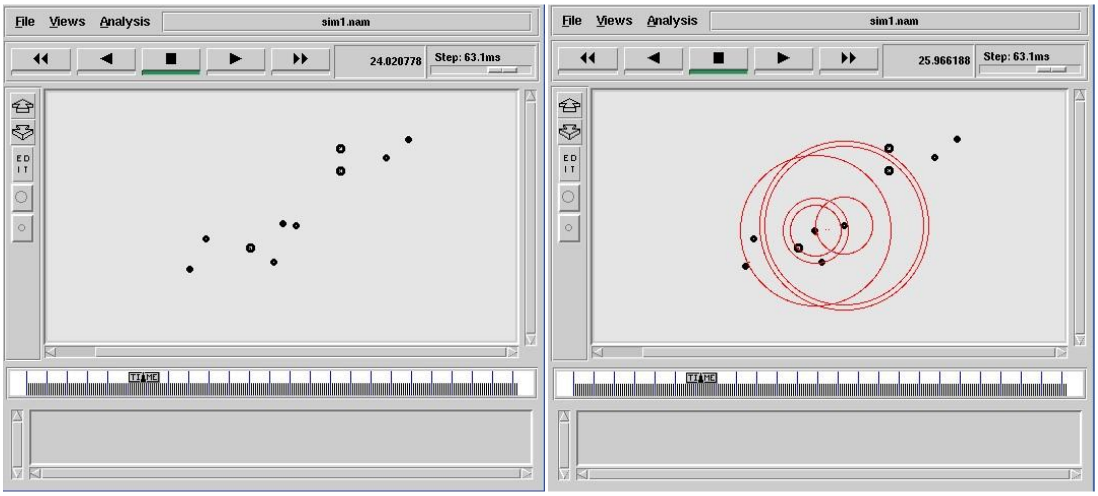
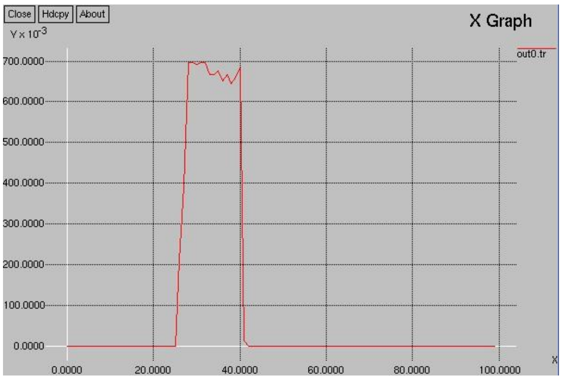


Figure 76 Topology of 10-node-scenairo 2 with OLSR



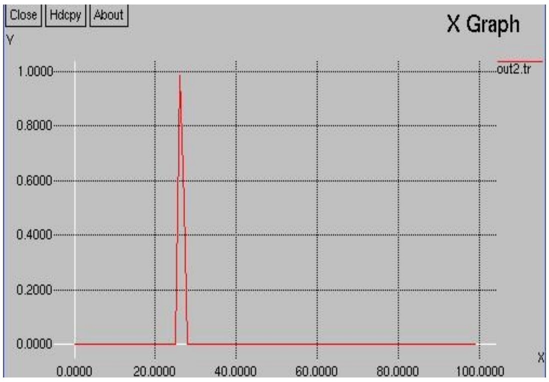


Figure 77 Bandwidth of 10-node-scenairo 2 with OLSR

Figure 78 Packet Loss Ratio of 10-node-scenario 2 with OLSR

Total bandwidth = 11 765 000 Bps

Average bandwidth = 94.12 Kbps

Summary:

More joined-node means more complex routing path in the network. In this case, specified parameters like interval time and node moving speed, OLSR performs better and more consistent than AODV. Since the time before communication set up, OLSR needs more time to build routing table for the reason of proactive protocol characteristic, it avoid prior period in which the protocol routing path is much complex.

Explain the results:

Same explanation as scenario1.

4.5 Scenario 3 – Realistic scenario

4.5.1 Setup

As we all know that there are two types of mobility models used in the simulation, traces and synthetic models. The initial two models are typical simple synthetic models. Synthetic models attempt to realistically represent the behaviours of mobile nodes without the use of traces, which means the models attempt to mimic the movements of boats in real life.

However, the truth is that we would not want boats to navigate in such a restricted route, just like it has been discussed in scenario 2 that boats are moving in straight lines at constant speeds throughout the course of the entire simulation.

The synthetic simulations I have done, gives a very good general picture of how the protocols behave in respect to certain parameters, such as mobility, size, traffic control and network load. This kind of simulations also has those problems:

- It is hard to identify the situations where the protocols fail or not act as we expect.
- It has nearly connection to a real life situation.
- It may favour complex protocols, while simpler protocols can find the routes as effectively in real life scenarios.

And then, it is very interesting to analyse the realistic scenarios. However, the realistic scenarios do not give a great view of general protocols behaves, but some sense of weak points instead.

4.5.2 Scenario characteristics

In this part, I am going to implement the third scenario using part of topology framework of Bergen according to the satellite map which has been captured on May 4, 2009.

The environment size is 1000*500 meters for my considering area, which is typically attempt to show the normal boat traffic in the sea area of Bergen.

This size is scaled according to the range of the transmitters. All the parameters used in this realistic simulation are shown in the below table.

Parameter	Value
Transmitter range	250 m
Bandwidth	10 Mbit
Simulation time	720 s
Number of nodes	5
Environment size	1000*500 m
Traffic type	Constant Bit rate
Packet rate	2 Mbps
Packet size	512 byte
Speed of a boat	0 ~ 20.0 nautical mile/hour
Test protocols	AODV & OLSR

Table 2 Parameters of scenario3

I generate the trace for the boat movements for 720 seconds. I also develop a monitor for the simulator as shown in the below pictures.

The traces are recorded during the time from 11:59 am Monday 04 May, 2009 to 12:07 pm Monday 04 May 2009. Location is a seaport of Bergen. [14]
(Note: The pictures are captured from screenshots using AIS (Automatic Identification System) server, it is permitted through the MARCOM Project and keep confidential.)

AIS is a shipboard broadcast system that acts like a continuous and autonomous transponder, operating in the VHF maritime brand. It allows ships to easily track, identify, and exchange pertinent navigation information from one another or ashore, for collision avoidance, security and VTS reporting. The AIS information from the vessels can be gathered via base stations ashore, passing the information to dedicated AIS data collectors. [15]

This scenario simulates 5 boats that attempt to build ad hoc network wireless network among them. It involves communication between some of the boats sometimes, but loses communication for the reason of out of communication range. Here I mark up them as node 0-4, where node 0 keeps try to communicate with node 4 throughout all the simulation time.

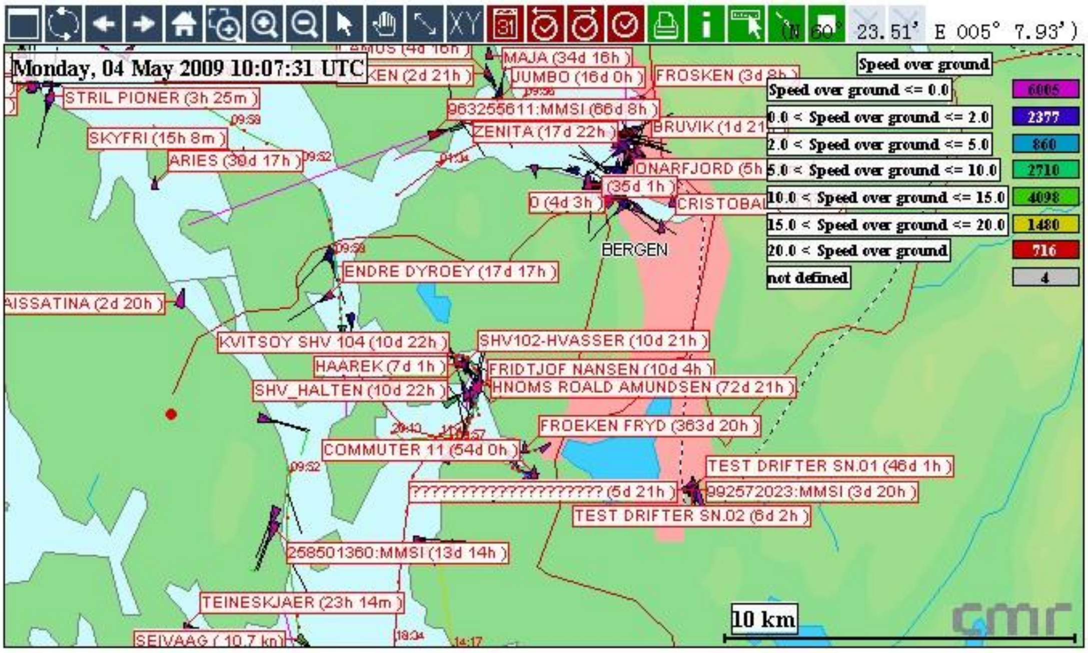


Figure 79 Zoomed in Satellite map of Scenario3

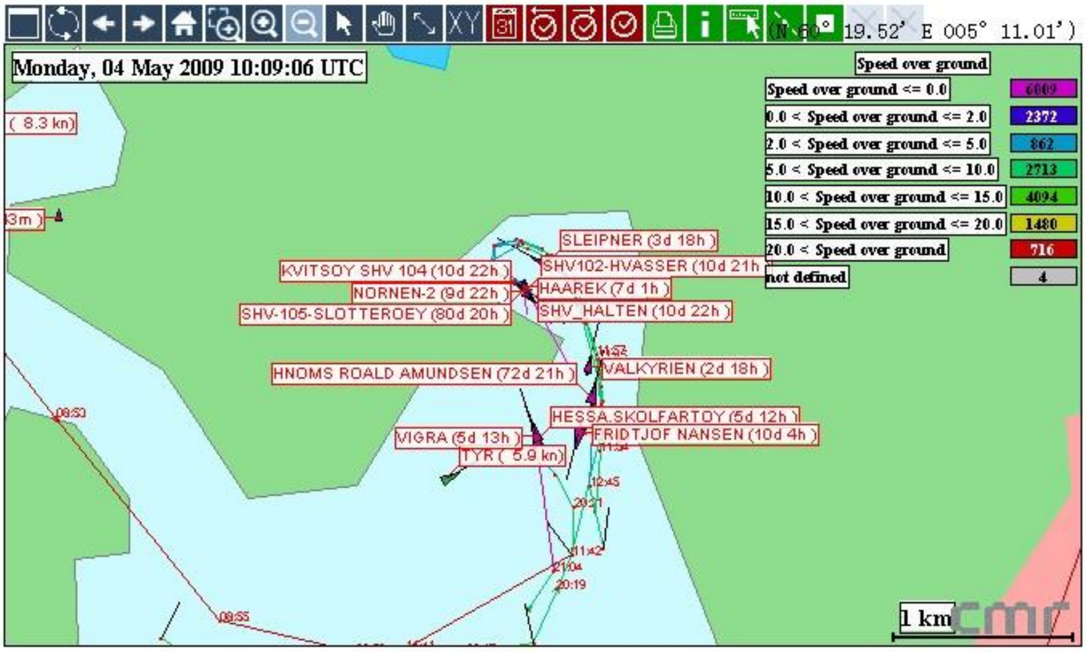


Figure 80 Zoomed Satellite map for scenario3

Here shows the detail trace area with more maps zoomed.

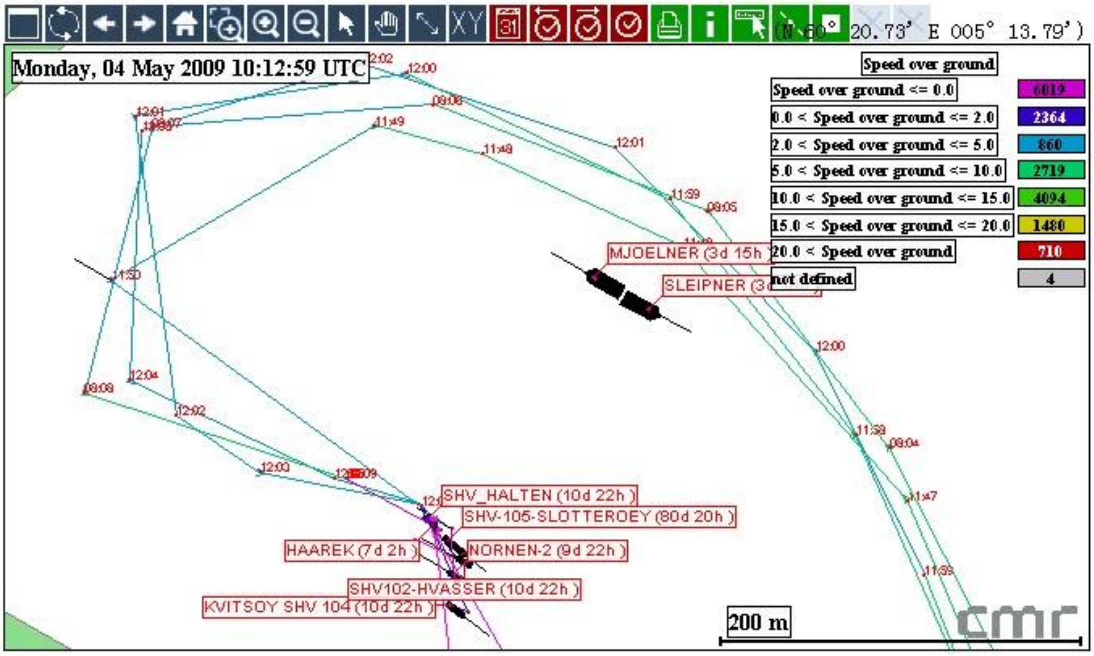


Figure 81Further zoomed Satellite map for scenario3

According to the recorded trace and time, I build the simulation in NS-2. It is assumed that boats use the IEEE 802.11 wireless LAN standard. The transmission range is set to the standard value of 250 m. The simulation environment is 1000*500 m. Each boat follows an unstable speed in the range from 0 to 20.0 nautical mile/hour (36 kilometre/hour).

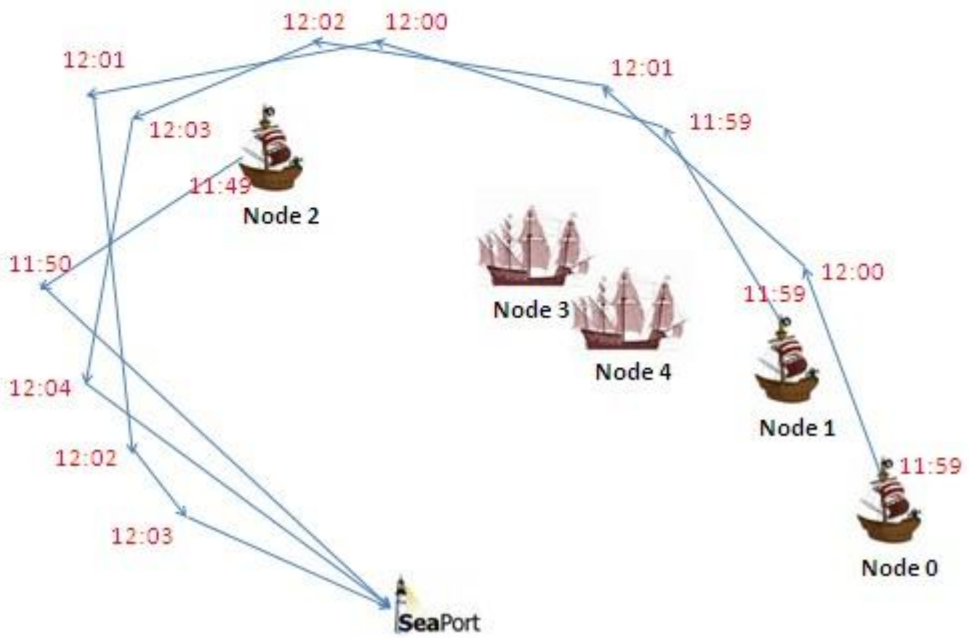


Figure 82 Boat trace simulation of scenario3

In this simulation, the parameters are set as follows:

- Number of nodes: This is constant during the simulation. This scenario uses 5 nodes for all simulation, two are stationary and the other three are mobility.
- Mobility state: Node 0, node 1 and node 2 moves following a fixed route and keeps an unstable speed and direction till the end of simulation.
- Environment size: Determines the size of the environment. I use a size of 1000*500 meters for the simulation.
- Pause time: Pause time is the time for which a node stands still before starts moving. In this scenario, here is no pause time.
- Network interface type: Phy/WirelessPhy, 914MHz Lucent WaveLAN DSSS radio interface.
- Mac type: 802.11 Mac-layer type.
- Link layer: Supported.
- Protocols: AODV and OLSR.
- Traffic type: Constant Bit Rate,
- Simulation time: 720sec
- Bandwidth: 10Mbit

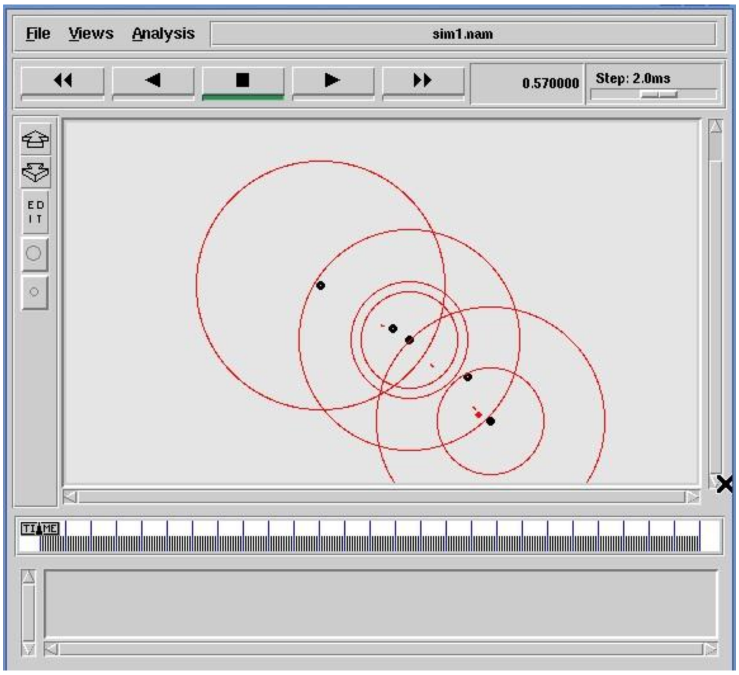


Figure 83 Topology for scenario 3 of AODV

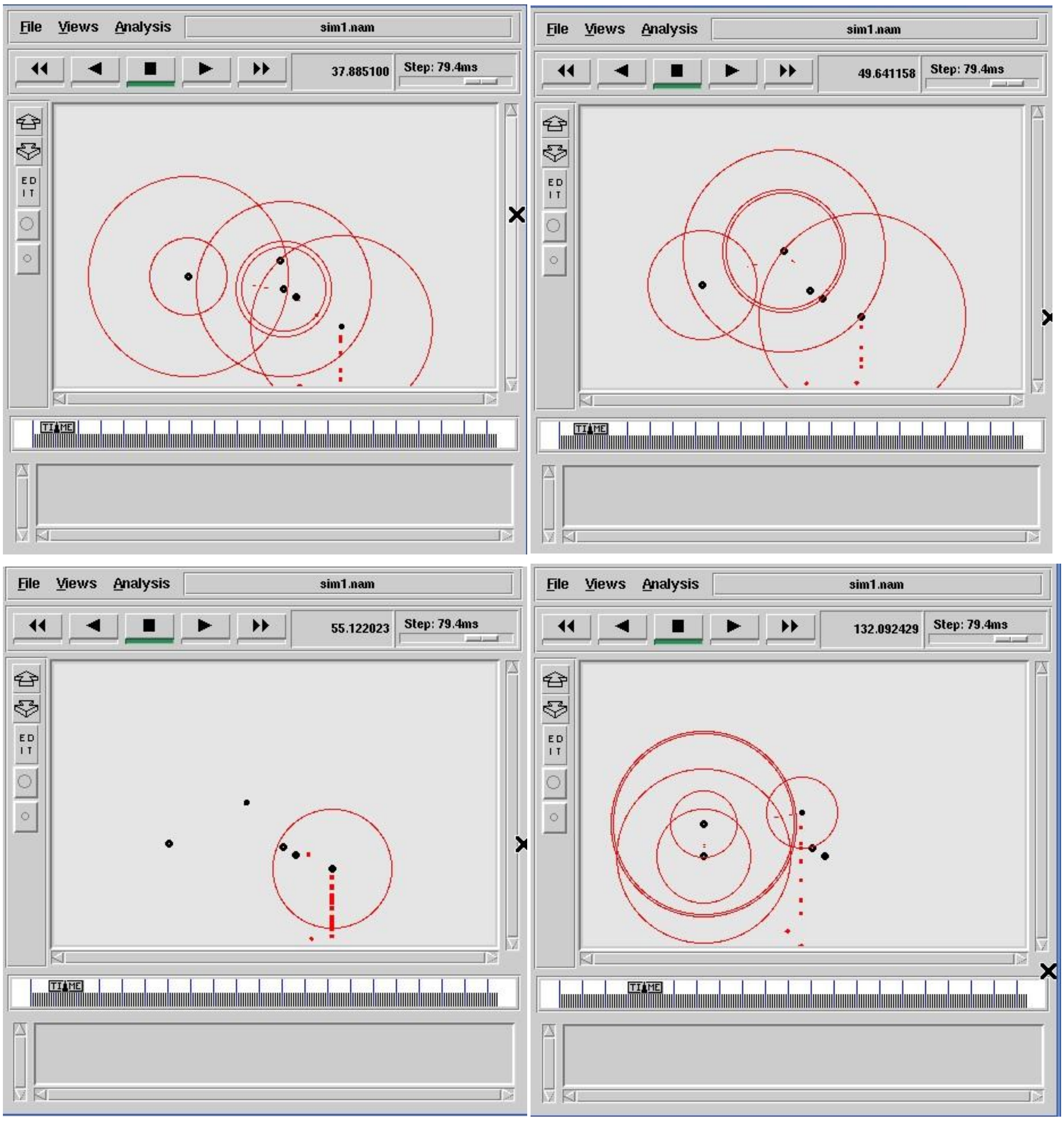
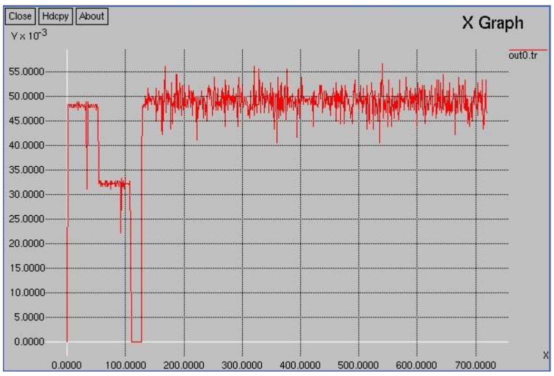


Figure 84 Topology for scenario 3 of OLSR



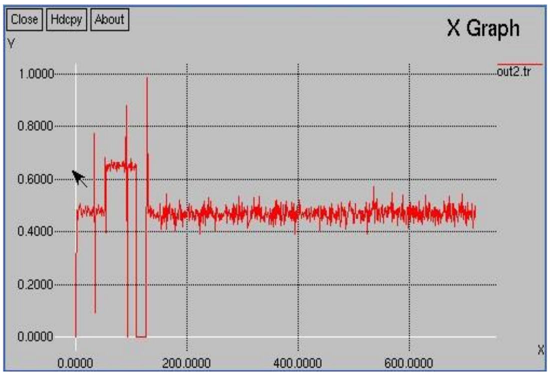
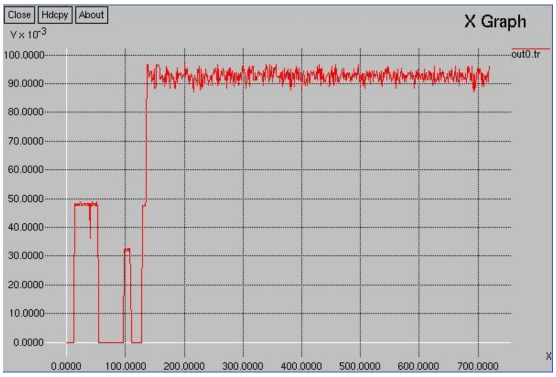


Figure 85 Bandwidth for scenario 3 of AODV

Figure 86 Packet Loss Ratio for scenario 3 with AODV

Total bandwidth = 29 966 500 Bps

Average bandwidth = 332.96 Kbps



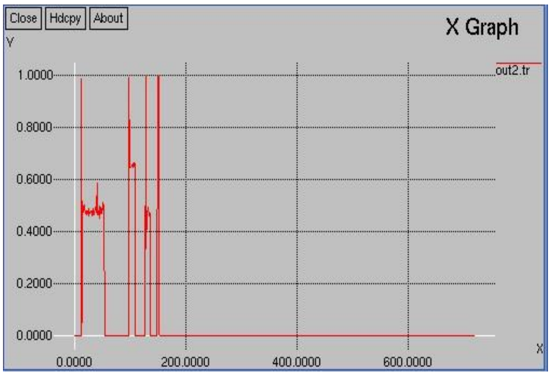


Figure 87 Bandwidth for scenario 3 of OLSR

Figure 88 Packet Loss Ratio for scenario 3 with OLSR

Total bandwidth = 51 034 000 Bps
Average bandwidth = 567.04 Kbps

Read from the graphs, we can see that:

- OLSR is a proactive routing protocol, so the routes are always immediately available when needed. The topological changes cause the flooding of the topological information to all available hosts in the network. The proactive characteristic of the protocol provides that the protocol has all the routing information to all participated hosts in the network. So before the communication transport is built or re-built, there needs more time to build routing table than it does in AODV. OLSR protocol requires each host periodically to send the updated topology information throughout the entire network, which makes protocol bandwidth huge increased used.
- After the communication transport has been built, the performance of OLSR is better than AODV's.
- Obviously the large fraction of dropped packets is not acceptable, almost 50%. The reason for these drops has to do with the interval of the "hello messages". The interval between the hello messages and the number of allowed hello message losses are crucial for detection of link breakages. If the interval is decreased, link breakages are detected earlier, but it would also make the control overhead in the network increases.

So, the interesting issue here is to try to find optimal values for these parameters to get what we except state. The choice of these parameters is much more dependent on the definite situation.

Now, let's try to change different parameters to see how the specified parameters affect the performance in different protocols, and compare the results to the results from initial scenario.

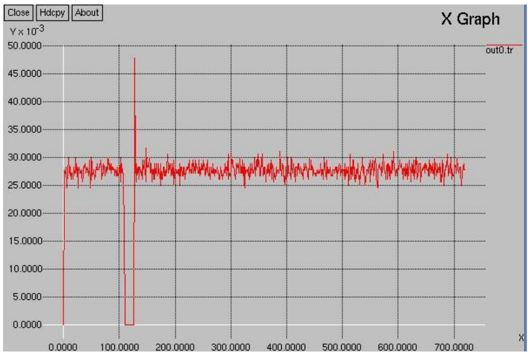
4.5.3 Change of data rates

➢ ***Source data rates***

The model is much the same like the initial one, but the source data will be changed, to examine in which way this affects the throughput and packet loss and different performance in different protocols.

Here I use the way of changing two parameters, pocketsize and interval for the current CBR.
The bit rate in the initial code is very high. Now we want to investigate what happens when we decrease the bit rate.

I reduce the packet size to 480 bps and increase the CBR-interval to 0.02 seconds. This gives source bit rate 480*8/0.02=192Kbps



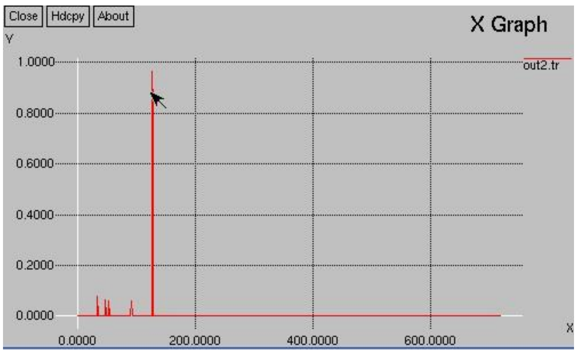
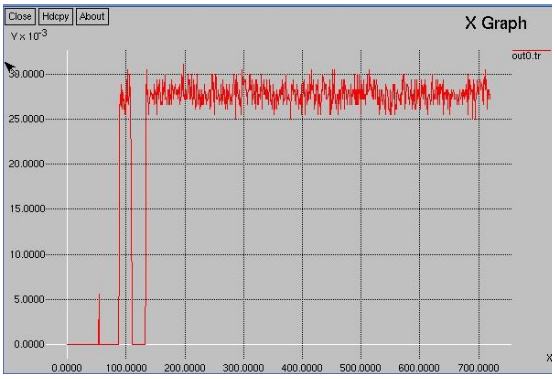


Figure 89 Bandwidth for scenario 3 with AODV

Figure 90 Packet Loss Ratio for scenario 3 with AODV

Total bandwidth = 17 525 500 Bps

Average bandwidth = 194.73 Kbps



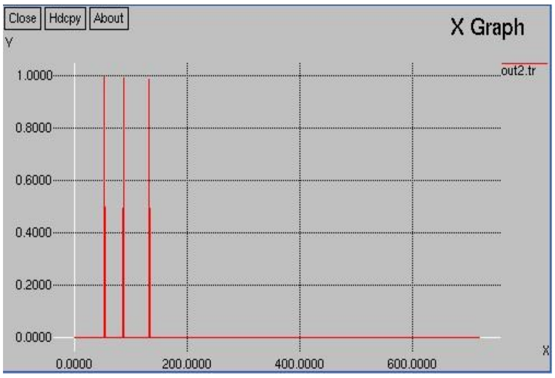


Figure 91 Bandwidth for scenario 3 with OLSR

Figure 92 Packet Loss Ratio for scenario 3 with OLSR

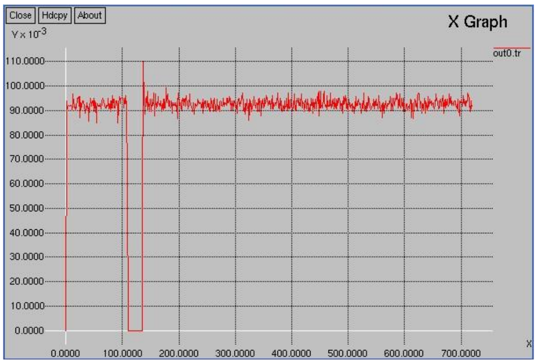
Total bandwidth = 15 211 040 Bps
Average bandwidth = 169.01 Kbps

<u>Summary:</u>
This change effects AODV heavily, instead OLSR not that much. The reason for this is that CBR-interval time increases which makes link breakages are detected later and more packets are dropped. That because the protocol is sending packets on a broken route that it thinks it valid and that packet in the buffers are dropped because of congestion and timeouts.

➢ *Transmission rates*

In this part, I look into the effects of increasing the transmission rate. The transmission data rate for the DATA is changed to 10Megabit/s, while BasicRate for RTS/CTS, and ACK remained unchanged.
This is done by changing the parameter Mac/802_11 set DataRate_ 1e7.

As expected, the total and average bandwidth grows. This can be verified by the bandwidth graph underneath.



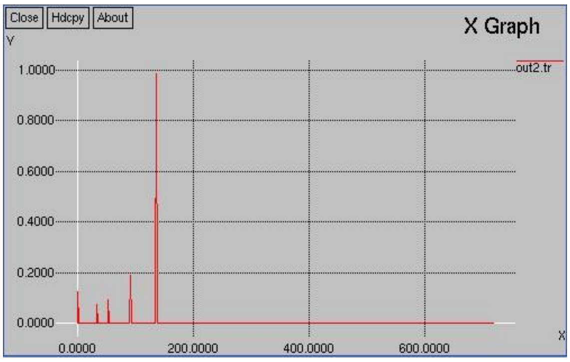
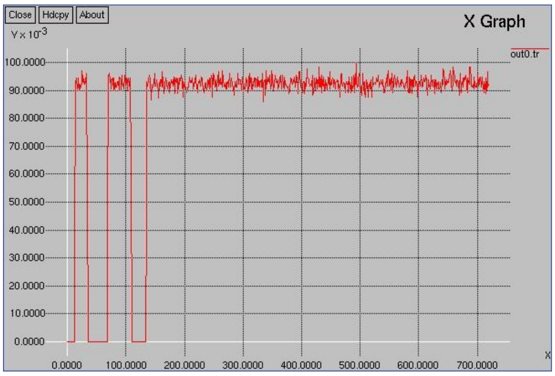


Figure 93 Bandwidth for scenario 3 with AODV
Figure 94 Packet Loss Ratio for scenario 3 with AODV

Total Bandwidth = 57 693 500 Bps
Average Bandwidth = 641.04 Kbps



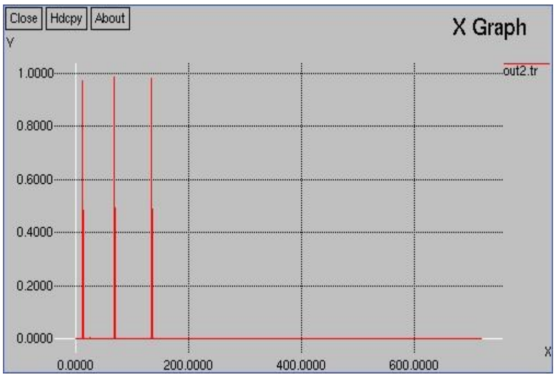


Figure 95 Bandwidth for scenario 3 with OLSR

Figure 96 Packet Loss Ratio for scenario 3 with OLSR

Total bandwidth = 53 838 500 Bps
Average bandwidth = 598.21 Kbps

Summary:

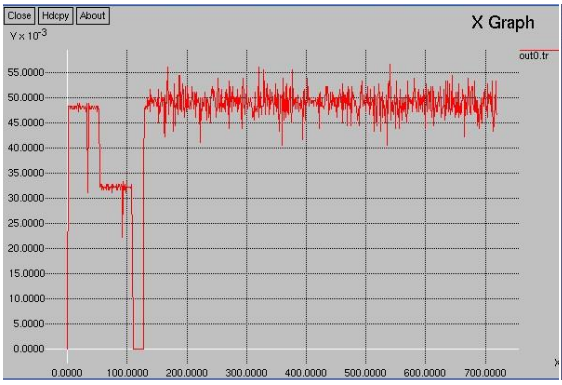
This change effects AODV heavily, but nearly OLSR. As the graph shows, we can see that total bandwidth of AODV is almost two time than initial one's. Even though, the performance of AODV is much better than before because of less packet loss happened in case of enough bandwidth support.

4.5.4 Change the number of traffic flows

In this part, I decrease half of the UDP-flow.

The graphs demonstrate that this has a very large impact on both the throughput and the packet loss.

We can see that the bandwidth has decreased to about the half the initial scenario, and that packet loss ratio is going to close 1. This is due to the fact that we are trying to transmit way much more than the saturation throughput.



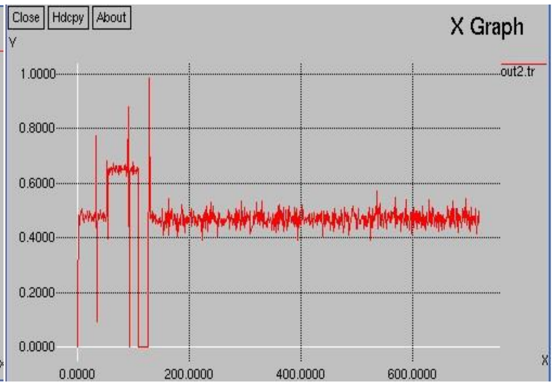
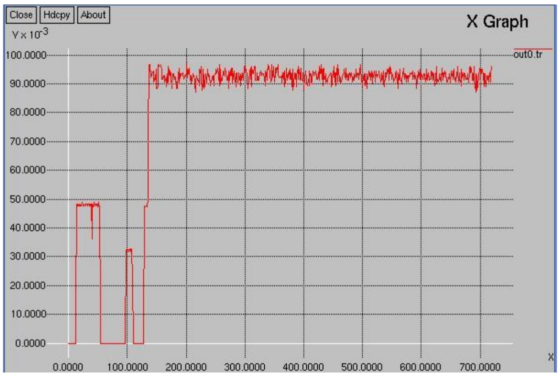


Figure 97 Bandwidth for scenario 3 with AODV

Figure 98 Packet Loss Ratio for scenario 3 with AODV

Total bandwidth = 29 966 500 Bps

Average bandwidth = 332.96 Kbps



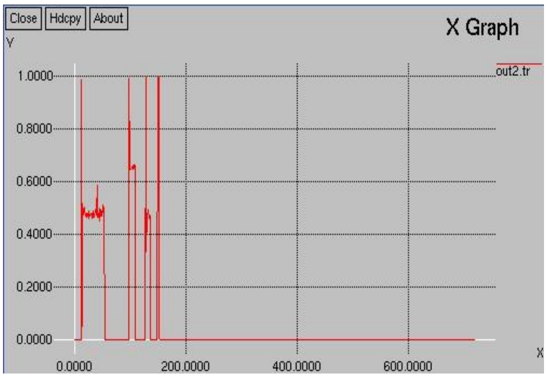


Figure 99 Bandwidth for scenario 3 with OLSR

Figure 100 Packet Loss Ratio for scenario 3 with OLSR

Total bandwidth = 51 034 000 Bps

Average bandwidth = 567.04 Kbps

Summary:

This change affects nearly.

4.5.5 Change the number of nodes

In this part, I add simulation nodes to 10 instead of 5. In this case, the network topology is changed. New nodes are not placed on the real-life data. We will investigate in which way the change of network topology influences the throughput and packet loss. The network topology is changed according to the figure underneath. The traffic-flow is set up between nodes 0 and 9.

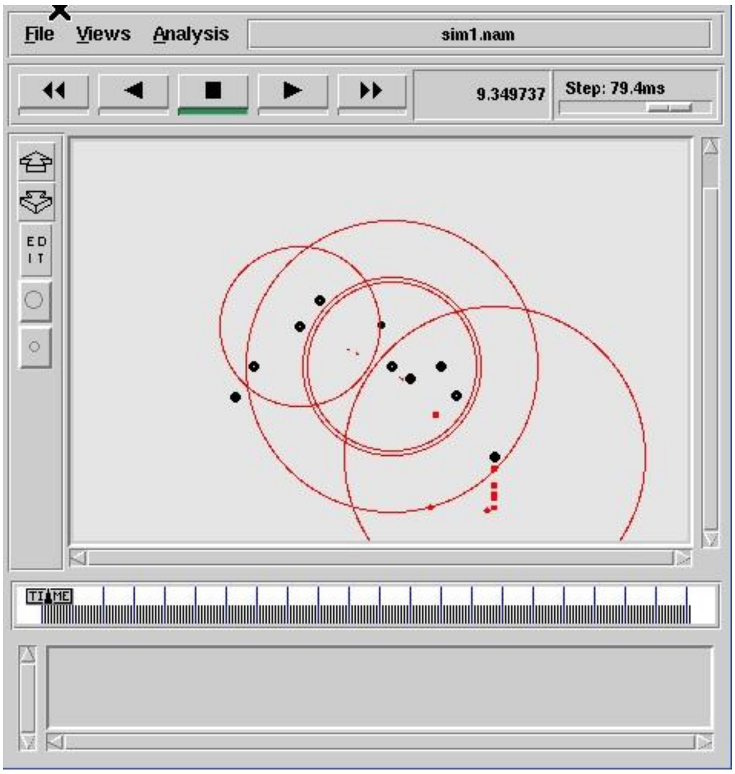
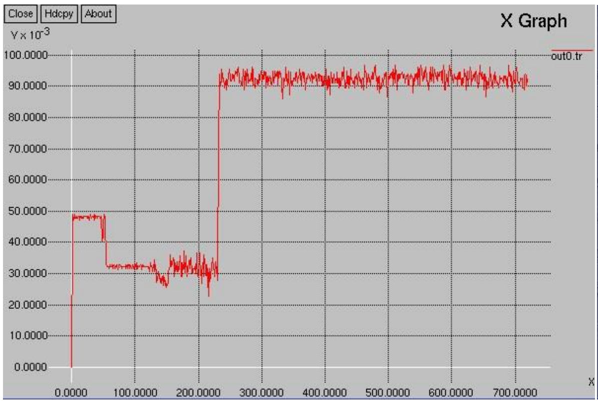


Figure 101 Topology for 10-node-scenario 3 with AODV



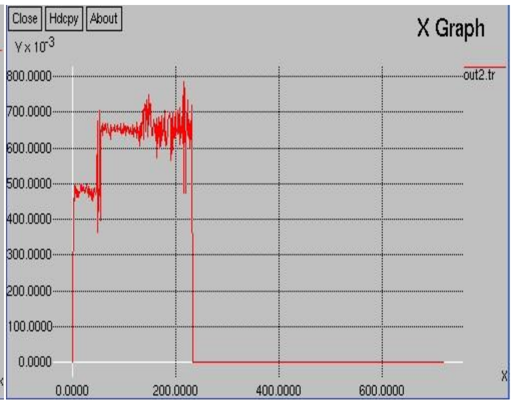


Figure 102 Bandwidth for 10-node-scenario 3 with AODV

Figure 103 Packet Loss Ratio for 10-node-scenario 3 with AODV

Total bandwidth = 47 949 000 Bps

Average bandwidth = 532.77 Kbps

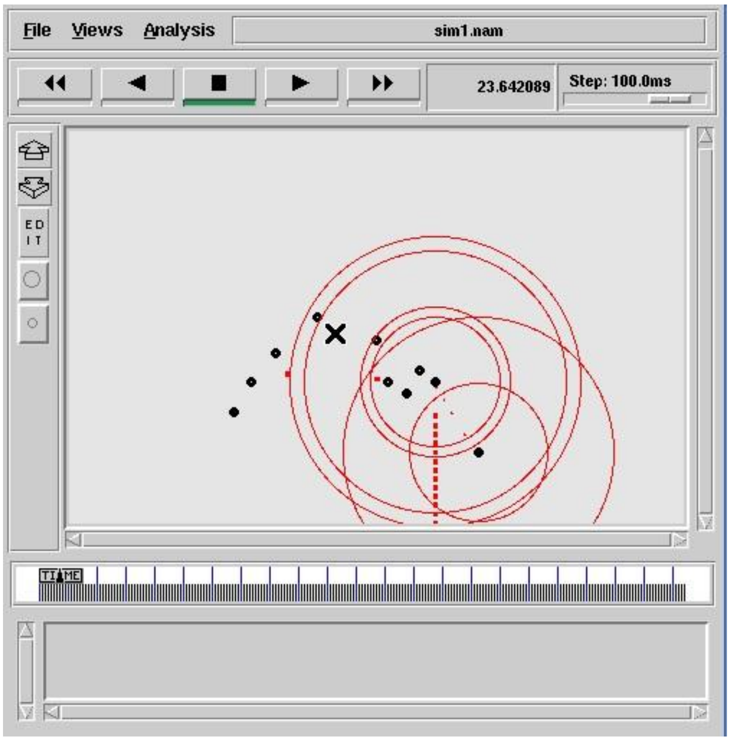
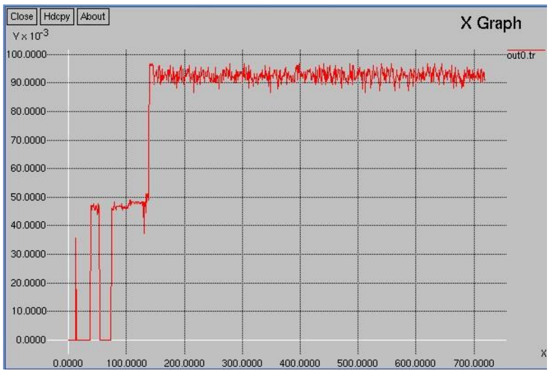


Figure 104 Topology of 10-node-scenario 3 with OLSR



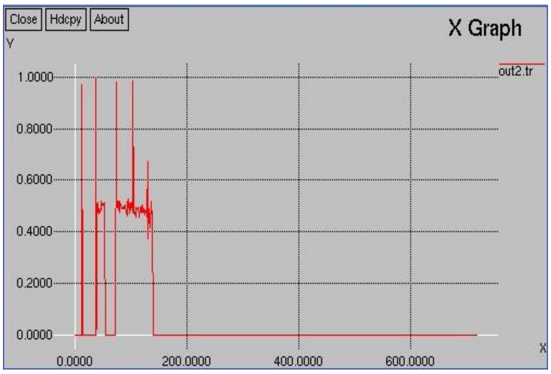


Figure 105 Bandwidth for 10-node-scenario 3 with OLSR

Figure 106 Packet Loss Ratio for 10-node-scenario 3 with OLSR

Total bandwidth = 51 835 500 Bps

Average bandwidth = 575.95 Kbps

Summary:

The joined-nodes increase, which means the node density increases. When the topology changes, the truth has to be faced that network has to find the optimal route. Because of the periodic updates, OLSR needs some time before it converges to a steady state for the reason of proactive protocol type. This happens when the network has a lot topology changes. Most of the packets that are sent during this time are dropped and the rest of them get higher hop count. AODV has a similar behaviour and better packet loss state.

5 Improve – Proposed enhancements to AODV and OLSR for Maritime Ad Hoc Network application

Due to time limited, this thesis would make further discussion about proposed enhancements to AODV and OLSR for maritime ad hoc network application.

5.1 Position-aware algorithm

It is assumed that all boats have installed GPS system.

The key phase of position-aware enhancement is that it could use GPS system to find the position of nearby boats and then automatically identify them on an interactive map display, the real-time location of all boats which have joined or will join network communication. Depending on this system, boats would be able to monitor those routes and give a prospective protocol and routing path.

5.2 Traffic load-aware algorithm

A good routing algorithm needs to balance the traffic load to help improving network capacity by avoiding routing traffic through congested areas.

Routing metrics are very critical to network performance. Good routing metric should carry enough information about the link quality so that a node can determine the best path to reach a gateway. The proposed routing metric consists of two parts: the congestion level and the channel utilization on a given node. The congestion level on each link of the node represents how hard to transmit successfully a frame on that link. The channel utilization represents the fraction of channel time in which the channel is sensed busy. [16]

5.3 Multiple gateways

A method of distributing packets to different gateways in maritime ad hoc network would be drawing much attention to get well performance to lower the communication throughput.

6 Conclusions, Contributions and Future Work

The simulations have shown that there certainly is a need for a special ad hoc network routing protocol when maritime topology changes for a good performance required.

6.1 Results

In network with a dynamic topology, proactive protocols such as OLSR have considerable difficulties in maintaining valid routes, and it loses much many packets while the network topology changes rapidly. With increasing mobility, its network traffic load grows larger to maintain routes to every node.

This study presents that a reactive routing protocol (AODV) is superior to a proactive one (OLSR). The proactive protocol (OLSR) offers better performance for CBR sources given but with much more bandwidth. OLSR protocols sends routing packets to discover and maintain routes to all destinations, that is the reason of the number of delivered packets decrease then the traffic load increase.

The realistic scenario is taken to get a clear understanding on how the protocols would behave in an environment more realistic than synthetic model. The results confirm that OLSR gets poor performance even though the mobility is kept rather low, instead AODV handled not good enough in this case. AODV get better performance in network where paths have many hops and low overhead is preferred.

However, the simulation results have show that neither of chosen protocols suit well in the scenarios without further modifications. Changing parameters and configuration may help performance enhancement. Obviously optimal protocols for maritime ad hoc network are still needed.

6.2 Contributions

Basically, the objective of this project is to improve my knowledge of the topological properties of a network built over nodes moving according to realistic vehicular mobility models, through a low-level study of the network connectivity, and try to find a suitable routing protocol for the ad hoc network for oil supply boats and perform it.

More specifically, the following four aspects reflect the main contributions:
1, a clear demonstration by different scenarios of maritime mobility analytical

descriptions, an attempt to finding a suitable routing protocol for project requirement (focus on comparing AODV and OLSR), ad hoc network protocol for oil platform, perform it on NS2.

2, a detailed study of the connectivity properties of realistic scenario at sea, which gives the most interesting and challenging background for network topological studies.

3, an attempt has been made to compare between AODV and OLSR protocols under the maritime environment. The results show that enhancements still need for both protocols.

6.3 Future Work

Ad hoc network is a quite hot concept in network communications. There is much valuable research going on and many issues keep waiting to be solved. Due to limited time, this report has only focused on two routing protocols simulation. However here should be further subject studies related to many issues.

- For the realistic environment discussion, the simulation scenarios should be more diverse.
- More routing protocols could be simulated, for instance, DSDV, DSR, and ZRP.
- Enhancement for both protocols.
- Other traffic than CBR.
- Evaluation of multicast routing protocols

7 References

[1] Fan Li, Yu Wang, “Routing in Vehicular Ad Hoc Networks: A Survey”, IEEE Vehicular technology magazine, June 2007

[2] http://en.wikipedia.org/wiki/NS (simulator)

[3] The CMU Monarch Project’s Wireless and Mobility Extensions to NS, http://www.monarch.cs.cmu.edu/

[4] http://nile.wpi.edu/NS/

[5] GloMoSim, http://pcl.cs.ucla.edu/projects/glomosim/

[6] Effects of Wireless Physical Layer Modelling in Mobile Ad Hoc Networks, MOBICOM 2001

[7] Network components, http://nile.wpi.edu/NS/

[8] Tony Larsson and Nicklas Hedman, “Routing Protocols in Wireless Ad-hoc Network – A Simulation Study”, 1998

[9] T. Clausen and P. Jacquet (2003), “Optimized Link State Routing Protocol (OLSR)”, RFC 3262, IETF Network Working Group, October 2003.

[10] Scott Corson and Joseph Macker, “Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations”. Internet-Draft, draft-ietf-manet-issues-01.txt, March 1998.

[11] Tony Larsson and Nicklas Hedman, “Routing Protocols in Wireless Ad-hoc Network – A Simulation Study”, 1998

[12] Martha Steenstrup, “Routing in communication networks”. New Jersey, Prentice Hall. ISBN 0-13-010752-2.

[13] http://hipercom.inria.fr/olsr/

[14] http://aisutland.aisonline.com/stat/, AIS server
(Password protected)

[15] Kåre P. Villanger and Kjell Røang, “Web Map Services at CMR”, 2004

[16] Lan Tien Nguyen, Razvan Beuran and Yoichi Shinoda, “A Load-aware Routing Metric for wireless Mesh Metworks”, 2008

Appendix A – Code of Scenairo3

```
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)            5                           ;# number of mobilenodes
set val(rp)            AODV                        ;# routing protocol
set val(tr_f)      sim1.tr                 ;# trace file
set val(tr_n)          sim1.nam                    ;# nam file
set val(seed)      1


# set up the antennas to be centered in the node and 1.5 meters above it
Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 1.5
Antenna/OmniAntenna set Gt_ 1.0
Antenna/OmniAntenna set Gr_ 1.0

# Initialize the SharedMedia interface with parameters to make
# it work like the 914MHz Lucent WaveLAN DSSS radio interface
Phy/WirelessPhy set CPThresh_ 10.0
Phy/WirelessPhy set CSThresh_ 1.559e-11
Phy/WirelessPhy set RXThresh_ 3.652e-10
Phy/WirelessPhy set Pt_ 0.28183815
Phy/WirelessPhy set freq_ 914e+6
Phy/WirelessPhy set L_ 1.0


#you can set dataRate for DATA here
Mac/802_11 set dataRate_ 1e6
#you can set basicRate for RTS/CTS, and ACK here
Mac/802_11 set basicRate_ 1e6

# ======================================================================
# Main Program
# ======================================================================
```

```
#
# Initialize Global Variables
#
set ns_		[new Simulator]
set tracefd	[open $val(tr_f) w]
$ns_ trace-all $tracefd

set namtrace [open $val(tr_n) w]
$ns_ namtrace-all-wireless $namtrace 600 600

#Open the output files
set f0 [open out0.tr w]
set f1 [open out1.tr w]
set f2 [open out2.tr w]
set f3 [open out3.tr w]

#set color
$ns_ color 0 red
$ns_ color 1 blue

# set up topography object
set topo	[new Topography]

$topo load_flatgrid 1000 500

#
# Create God
#
create-god $val(nn)

#
#  Create the specified number of mobilenodes [$val(nn)] and "attach" them
#  to the channel.



# create channel
set chan_1_ [new $val(chan)]

# configure node

	$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) \
                                -channel $chan_1_ \
                        -topoInstance $topo \
                        -agentTrace ON \
                        -routerTrace ON \
                        -macTrace ON \
                        -movementTrace OFF

        for {set i 0} {$i < $val(nn) } {incr i} {
                set node_($i) [$ns_ node]
         #      $node_($i) random-motion 0              ;# disable random motion
        }

#set positions for created five nodes
#we start simulation at 11:59AM

$node_(0) set X_ 500
$node_(0) set Y_ 30
$ns_ at 0.0 "$node_(0) setdest 430 160 2.46"
$ns_ at 60.0 "$node_(0) setdest 330 270 2.48"
$ns_ at 120.0 "$node_(0) setdest 180 320 3.06"
$ns_ at 180.0 "$node_(0) setdest 70 285 1.92"
$ns_ at 240.0 "$node_(0) setdest 50 100 3.1"
$ns_ at 300.0 "$node_(0) setdest 240 50 1.09"


$node_(1) set X_ 460
$node_(1) set Y_ 110
$ns_ at 0.0 "$node_(1) setdest 350 250 5.93"
$ns_ at 30.0 "$node_(1) setdest 210 315 5.145"
$ns_ at 60.0 "$node_(1) setdest 60 290 2.53"
$ns_ at 120.0 "$node_(1) setdest 80 120 2.85"
$ns_ at 180.0 "$node_(1) setdest 240 50 0.97"


$node_(2) set X_ 190
$node_(2) set Y_ 280
$ns_ at 0.0 "$node_(2) setdest 40 200 2.83"
```

```
$ns_ at 60.0 "$node_(2) setdest 240 50 0.42"


$node_(3) set X_ 320
$node_(3) set Y_ 200
$ns_ at 0.0 "$node_(3) setdest 320 200 0.0"

$node_(4) set X_ 350
$node_(4) set Y_ 180
$ns_ at 0.0 "$node_(4) setdest 350 180 0.0"


# only one UDP connection at very high bitrate (>saturation throughput)
# the results on different nodes are obtained by setting sink0 at each node
set udp1 [new Agent/UDP]
set sink0 [new Agent/LossMonitor]
$ns_ attach-agent $node_(0) $udp1
$ns_ attach-agent $node_(2) $sink0
$ns_ connect $udp1 $sink0
set cbr1 [new Application/Traffic/CBR]
#src bitrate: 1500*8/0.006=2000 Kbps
$cbr1 set packetSize_ 1480
#src bitrate: 500*8/0.002=2000 Kbps
$cbr1 set packetSize_ 480
$cbr1 set interval_ 0.006
$cbr1 set random_ 1
#$cbr1 set maxpkts_ 10000
$cbr1 attach-agent $udp1
$ns_ at 0.0 "$cbr1 start"

# UDP2 connections from node_(2) to node_(0)
set udp2 [new Agent/UDP]
set sink1 [new Agent/LossMonitor]
$ns_ attach-agent $node_(0) $udp2
$ns_ attach-agent $node_(2) $sink1
$ns_ connect $udp2 $sink1
set cbr2 [new Application/Traffic/CBR]
#I add tunable parameters for CBR
$cbr2 set packetSize_ 512
$cbr2 set interval_ 0.02
$cbr2 set random_ 2
#$cbr2 set maxpkts_ 10000
$cbr2 attach-agent $udp2
#$ns_ at 2.0 "$cbr2 start"
```

```
#Define a 'finish' procedure
#the simulation results can be shown by xgraph
proc finish {} {
        global f0 f1 f2 f3
        #Close the output files
        close $f0
        close $f1
            close $f2
            close $f3
            #execute nam
            exec nam sim1.nam &
        #Call xgraph to display the results
        exec xgraph out0.tr out1.tr    -geometry 600x400 &
            exec xgraph out2.tr out3.tr    -geometry 500x300 &
            exit 0
}

set tot_bw0 0
#Define a procedure which periodically records the bandwidth received by the
proc record {} {
            global sink0 sink1 f0 f1 f2 f3 tot_bw0
        #Get an instance of the simulator
            set ns [Simulator instance]
        #Set the time after which the procedure should be called again
            set time 1.0
        #How many bps have been received by the traffic sinks?
            set bw0 [$sink0 set bps_]
            #how many packets are dropped
            set drop1 [$sink0 set nlost_]
            set rec1    [$sink0 set npkts_]
            #Get the current time
            set now [$ns now]
            #puts "$now, $drop1, $drop2, $rec1, $rec2, $pkid0, $pkid1"
           set rec_time [$sink0 set lastPktTime_]
            puts "$now, $bw0, $rec_time"
            #count the total number of bps received
            set tot_bw0 [expr $tot_bw0 + $bw0]
            #Calculate the bandwidth (in MBit/s) and write it to the files
            puts $f0 "$now [expr $bw0/$time*8/7200000]"
            #calculate loss ratio, to avoid divided-by-zero error
      if {$rec1==0} {set rec1 1}
            puts $f2 "$now    [expr    double($drop1)/double($drop1+$rec1)]"
```

```
        #Reset the bps_ values on the traffic sinks
            $sink0 set bps_ 0
            #reset the nlost_ values to zero
            $sink0 set nlost_ 0
            #reset the npkts_ values to zero
            $sink0 set npkts_ 0
        #Re-schedule the procedure
            $ns at [expr $now+$time] "record"
}

proc average {} {
          global tot_bw0
          #average throughput in Kbps, 720 sec is session time
          set ave_bw0 [expr double($tot_bw0*8)/720000]
          puts "tot_bw0= $tot_bw0    ave_bw0= $ave_bw0"
}

#Start logging the received bandwidth
$ns_ at 0.0 "record"

#calculate the average throughput
$ns_ at 720.0 "average"

# Tell nodes when the simulation ends
#
for {set i 0} {$i < $val(nn) } {incr i} {
      $ns_ at 720.0 "$node_($i) reset";
}



$ns_ at 720.01 "stop"
$ns_ at 720.01 "puts \"NS EXITING...\" ; $ns_ halt"
proc stop {} {
      global ns_ tracefd namtrace
      $ns_ flush-trace
      close $namtrace
      close $tracefd
      exec nam sim1.nam &
      exit 0
}



#Call the finish procedure
```