Evaluation of MANET Routing Protocols in Realistic Environments

Marc Esquius Morote

College of Electronics and Information Engineering,
Tongji University

Escola Tècnica Superior
d'Enginyeria de Telecomunicació de Barcelona,
Universitat Politècnica de Catalunya

Project Tutor: Prof. Guo Aihuang

Novembre 2010

Abstract

Recently, many researchers have become interested in MANET (Mobile Ad-hoc NETworks) to construct a self-configurable network without existing communication infrastructure. This research presents the results of a detailed performance evaluation on several MANET routing protocols working under realistic environments. The routing protocols, mobility models and other aspects are explained and discussed in order to know how to use them properly to model real-life conditions. NS-2 and Bonnmotion were used to create the networks, services and environment characteristics in general.

It is concluded which protocols can handle which applications and which not and that the performance of the protocols can be considerably different when more and more realistic elements are taken into account. This should be considered in further researches since the nowadays evolution of MANET will bring them soon into services of our society.

Keywords: MANET, Routing protocols, Performance evaluation, Realistic environments.

Acknowledgments

First of all I would like to thank Li Hui for her support and advices during the whole time, even when she was in Tongji as well as when she left to the United States. Thanks for helping me along my NS-2 problems! I also want to thank professor Guo Aihuang for his suggestions and for letting me work in his department.

My research period in Shanghai would not have been the same mainly without all the wonderful moments spent with Sarah taking, from time to time, my mind off the research; and without Pablo with who we made all the long working hours on our projects much more enjoyable. Also thank all friends in the dormitories for so pleasant stay and for keeping the working spirit up all time!

Last, but not least, I want to thank my parents for all the support and all the opportunities they always gave me. Without this, I could not be where I am.

Contents

Chapter 1

Introduction

1.1 Why MANET

An ad-hoc network is often described as a collection of mobile platforms or nodes where each node can move freely and arbitrarily without the benefit of any fixed infrastructure except for the nodes themselves. They are often autonomous, self-configuring, adaptive and have proven lots of benefits, that is why they are the subject of much current research. The rapid adoption of wireless networking technology in the commercial sector using IEEE 802.11-based WLAN specifications is evidently seen.

Each node in an ad hoc network is in charge of routing information between its neighbors, thus contributing to and maintaining connectivity of the network. This results in a need to identify a new class of routing protocols. Protocol development efforts for ad-hoc networks are effervescent in the wireless networking research area. These efforts are largely fueled by the formation of the Mobile Ad-hoc NETworking (MANET) working group within the Internet Engineering Task Force (IETF) in 1999, to develop a routing framework for IP-based protocols in ad-hoc networks.

Because of all its properties, MANET is an excellent candidate to be used in places where wireless communication is needed but there is a lack of infrastructure. Examples of its utilization could be in military environments, rescue operations or vehicles communication.

1.2 State of the art

In order to see how MANET research has gain more and more importance during the last year, the number of publications in IEEE digital library that have MANET as a keyword is plotted in figure 1.1. Is shown that during the last decade the amount of publication every year increased considerably and its tendency is to keep increasing even more.

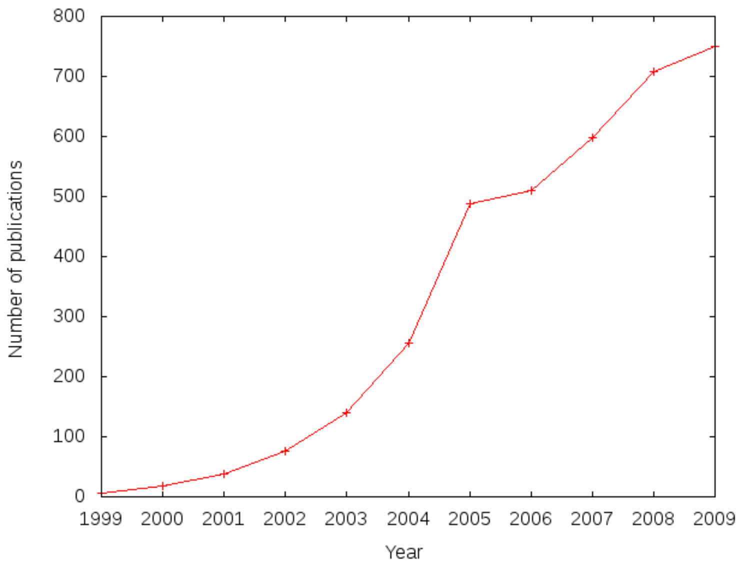


Figure 1.1: Number of publication referring MANET during the years

These publications study and analyze different aspects of the MANET themselves and how the different protocols perform under different circumstances considering different figures of merit.

In [1] [2] [3] [4] [5] [6] [7] they compare the most common routing protocols (AODV, DSDV, DSR, TORA, etc.) between them or with some new implementation under non-realistic circumstances of movement or propagation. For example, [7] studies the performance of DSR, AODV and TORA changing the application data (sensor, text, voice, etc.) through the packet delivery fraction, the end-to-end delay and the routing load. In the other hand, in [1], although is a really recent publication of 2010, they only compare DSDV and AODV sweeping the number of connections in a simple MANET.

The publications [8] [9] [10] [11] study one of the problems that will come up with the growth of MANET (aspect also discussed in this research), the scalability of the protocols, but again without modeling a realistic environment. Using two different simulators (QualNet and NS-2), in [8] they compare some common protocols with a location based protocol (LAR) to check the advantages of geographical information in this problem. A link prediction model is tested in [10] to see how the scalability of the protocols can improve in MANETs from 100 up to 1000 nodes.

On one hand, other publications like [12] [13] [14] try to study the impact of the protocols in real scenarios but not with different mobility models. In [13] they model a network of sensors considering lots of aspects of different propagation models and prove that choosing a wrong model can lead to wrong results in terms of protocol performance. On the other hand, [15] [16] [17] [18] evaluate MANETs using different mobility models but leaving a part some other characteristics to define better a real scenarios. For example in [15] they perform an analysis that helps in discerning the effect of mobility and fading on performance of MANETs in various network scenarios.

After this search we see that, although lately more publication are trying to do their studies in more realistic environments, still lots of them do not consider these aspects. From the ones who try to achieve more realistic conditions, not all of them gather all the factors together so they still have a lack of detail that with the nowadays tools could be covered.

1.3 Aim of the research

The aim of this research is to evaluate the performance of different routing protocols for MANET in realistic environments using the available open-source tools for simulating mobile networks. As it is explained in the previous section, lots of theoretical work has been done during these last years in MANET, but a large percentage of them evaluate protocols under non-realistic conditions: the transmission and receiver devices are not modeled according to commercial ones, but using the default parameters of the simulator; some of the mobility models used have interesting mathematical properties but do not describe the path a person or a car follows; the geographical zone is usually not taken into account, which can affect considerably the performance mainly if we are in an environment with obstacles where fading effects must be considered; etc. In the only environment where lots of research has been carried out under realistic condition is in the case of MANET used in vehicular networks (VANET), but there are much more environments where MANET can be used.

In chapter 2, different routing algorithms are described and then the routing protocols used in this research which use these algorithms are explained. The following chapter contains the explanation of the mobility models considered, a comparison between them and in which situations these mobility models should be used to describe accurately the movement of the wireless devices. The mobility models is one of the aspects that affect more a protocol performance, that is why is so important to study them and their properties well. The next chapter, number 4, presents the simulation tools and methodology followed in this research: for modeling a system and simulate it as well as for how to analyze the results to extract conclusions.

Chapter 5 is the core chapter of the research, several MANET case studies are presented, modeled and analyze using all the information presented in the previous chapters. Each case study has been modeled as accurate as possible using the network simulator NS-2 and other tools to obtain a realistic scenario. Finally in chapter 6 the research conclusions and future work are discussed.

Chapter 2

Routing

In this section first, the main routing algorithms are briefly described to get an idea of how they work. Then, is explained in more detail how the routing protocols analyzed in this research work and in which way they use each routing algorithm. For each routing protocol, since most of them has evolved during the years, the version described is the one implemented in the network simulator.

2.1 Routing algorithms

2.1.1 Distance vector

The methods used to calculate the best path for a network are different between different routing protocols but the fundamental features of distance-vector (DV) algorithms are the same across all DV based protocols. Distance Vector means that Routers are advertised as vector of distance and direction. *Direction* is simply next hop address and exit interface and *Distance* means such as hop count. Routers using distance vector protocol do not have knowledge of the entire path to a destination, as it happen with Source routing.

DV protocols are based on calculating the direction and distance to any link in a network. The cost of reaching a destination is calculated using various route metrics.

Usually updates are performed periodically in a distance-vector protocol where all (or part of) the router's routing table is sent to all its neighbors that are configured to use the same distance-vector routing protocol. Once a router has this information it is able to amend its own routing table to reflect the changes and then inform its neighbors of the changes. This process has been described as "routing by rumor" because routers are relying on the information they receive from other routers and cannot determine if the information is actually valid and true. There are a number of features which can be used to help with instability and inaccurate routing information, these depends of each protocol itself, so it will be explain with each specific protocol.

2.1.2 Source routing

Source routing allows a sender of a packet to partially or completely specify the route the packet takes through the network. In contrast, in non-source routing protocols (like distance-vector), routers in the network determine the path based on the packet's destination. Source routing allows easier troubleshooting, improved trace-route, and enables a node to discover all the possible routes to a host. It also allows a source to directly manage network performance by forcing packets to travel over one path to prevent congestion on another.

In the Internet Protocol, two header options are available which are rarely used: "strict source and record route" (SSRR) and "loose source and record route" (LSRR). Due to security concerns, packets marked LSRR are frequently blocked on the Internet. If not blocked, LSRR can allow an attacker to spoof its address but still successfully receive response packets.

2.1.3 Link-state

The basic concept of link-state routing is that every node constructs a map of the connectivity to the network, in the form of a graph, showing which nodes are connected to which other nodes. Each node then independently calculates the next best logical path from it to every possible destination in the network. The collection of best paths will then form the node's routing table. This contrasts with distance-vector routing protocols, which works by having each node share its routing table with its neighbors. In a link-state protocol the only information passed between nodes is connectivity related, instead of telling the neighbors its routing table, it tell the others about its neighbors.

First, each node needs to determine what other ports it is connected to, over fully-working links (it does this using a simple reachability protocol). Next, each node periodically and in case of connectivity changes makes up a short message, the link-state advertisement to identify the node and which nodes is connected including a sequence number. To create the map of connectivity, the algorithm simply iterates over the collection of link-state advertisements and for each one, it makes links on the map of the network. No link is considered to have been correctly reported unless the two ends agree. For example, if one node reports that it is connected to another, but the other node does not report that it is connected to the first, there is a problem, and the link is not included on the map.

2.1.4 Link-reversal

The link-reversal routing protocol was designed for packet-radio networks, where routes should be maintained among mobile nodes to a central station. Ones of its main points are that changes in network topology should only affect local nodes (and so reduce protocol overhead) and not to find the shortest path to a destination but to find some paths quickly. Looking for the shortest path makes little sense in networks where the topology changes rapidly, so it's better to find fast enough a route before the topology changes again.

Link-reversal routing algorithms are based on a principle known from graph theory. The distributed algorithm tries to maintain a *directed acyclic graph* (DAG) which is destination oriented (it does so by reversing edge directions in the graph when needed). There is one such DAG for every destination and this graph is called *destination oriented* when there is only one vertex (node) which has no outgoing edge (sink), which actually is the destination.

The algorithms works this way: supposing a DAG has no sink, any node is picked (A) and since it is not the sink it must have an outgoing edge to another node (B). Because of A, B is also not the sink and must have an outgoing edge to another node too. When the outgoing edge lead us to a previous node we would have discovered a cycle. As we cannot go on forever by connecting to a new node, so we will have to stop in a node which has no outgoing edges, which will be the sink of the DAG. What the distributed algorithm does is to reverse the direction of links at nodes which have lost their last downstream neighbor (outgoing edges).

2.2 Routing Protocols

An ad-hoc routing protocol is an standard that controls how nodes decide which way to route packets between devices in a mobile ad-hoc network. In MANET, nodes do not start out familiar with the topology of their networks: they have to discover it. The basic idea is that a new node may announce its presence and should listen for announcements broadcast by its neighbors, so each node learns about nodes nearby and how to reach them.

There are lots of routing protocols and they use different routing algorithms (like the ones explained in the previous section) to accomplish their mission. A part from the algorithm, they can also be classified by their routing principle. In this research, proactive (table-driven), reactive (on-demand) and geographical (position-based routing) protocols are used.

- Proactive: This type of protocols maintains fresh lists of destinations and their routes by periodically distributing routing tables throughout the network. The main disadvantages are the respective amount of data for maintenance (even if any information must be sent) and the slow reaction on restructuring and failures.

- Reactive: In this type of protocols, a route is established only when it is required and hence the need to find routes to all other nodes in the network as required by the table-driven approach is eliminated. The main disadvantages are the high latency time in route finding and the fact that an excessive flooding can lead to network congestion.

- Geographical: This type of protocols utilize physical distances and distribution of nodes in areas in order to improve the network performance. The main disadvantages are that the efficiency depends on balancing the geographic distribution versus occurrence of traffic and that negligence calculating the distance may cause overload.

Table 2.1 presents the routing protocols used in this research and which routing algorithm and principle are the based on. In the next subsections, each protocol is explained in more detail to see their other properties, advantages and disadvantages a part from these main two characteristics.

Table 2.1: Characteristics of the routing protocols

Protocol	**Routing algorithm**	**Routing principle**
DSDV	Distance vector	Proactive
DSR	Source routing	Reactive
AODV	Distance vector	Reactive
OLSR	Link-state	Proactive
TORA	Link-reversal	Reactive
LAR	Source routing	Reactive-Geographical

With the protocols chosen, the main type of routing algorithms and principles can be evaluated in the case studies presented in chapter 5. This way, we can tell which algorithms and other protocols properties are more suitable in different real scenarios.

2.2.1 DSDV

DSDV (Destination Sequenced Distance Vector) is a proactive protocol based in the distance vector (Section 2.1.1). DSDV is one of the most well known table-driven routing algorithms for MANETs. In distance vector protocols, every node i maintains for each destination x a set of distances dij(x) for each node j that is a neighbor of i. Node i treats neighbor k as a next hop for a packet destined to x if dik(x) equals minjdij(x). The succession of next hops chosen in this manner leads to x along the shortest path. In order to keep the distance estimates up to date, each node monitors the cost of its outgoing links and periodically broadcasts to all of its neighbors its current estimate of the shortest distance to every other node in the network. The distance-vector which is periodically broadcasted contains one entry for each node in the network which includes the distance from the advertising node to the destination. The distance vector algorithm described above is a classical Distributed Bellman-Ford (DBF) algorithm.

DSDV is a distance-vector algorithm which uses sequence numbers originated and updated by the destination, to avoid the looping problem caused by stale routing information. In DSDV, each node maintains a routing table which is constantly and periodically updated (not on-demand) and advertised to each of the node's current neighbors. Each entry in the routing table has the last known destination sequence number. Each node periodically transmits updates, and it does so immediately when significant new information is available. The data broadcasted by each node will contain its new sequence number and the following information for each new route: the destination's address, the number of hops to reach the destination and the sequence number of the information received regarding that destination, as originally stamped by the destination.

No assumptions about mobile hosts maintaining any sort of time synchronization or about the phase relationship of the update periods between the mobile nodes are made. Following the traditional distance-vector routing algorithms, these update packets contain information about which nodes are accessible from each node and the number of hops necessary to reach them. Routes with more recent sequence numbers are always the preferred basis for forwarding decisions. Of the paths with the same sequence number, those with the smallest metric (number of hops to the destination) will be used.

You can find the detailed description of this protocol and its improvements over the years in [19].

Advantages and disadvantages

As a proactive protocol it requires regular updates of its routing tables, which uses up battery power and a small amount of bandwidth even when the network is idle.

This limits the number of nodes that can join the network. Whenever topology of the network changes, a new sequence number is necessary before the network re-converges and DSDV is unstable until update packets propagate through the network. For this reason DSDV is not suitable for highly dynamic networks.

DSDV is effective for creating ad-hoc networks for small populations of mobile nodes. Even if the number of nodes is higher, DSDV can performs good if the topology does not change quick. Furthermore, as there is no need to ask for the route every time data needs to be sent its delay is considerably small.

2.2.2 DSR

Dynamic source routing protocol (DSR) is an on-demand protocol designed to restrict the bandwidth consumed by control packets in ad-hoc wireless networks by eliminating the periodic table-update messages required in the table-driven approach. As its name tells, this protocol use source routing algorithm to discover routes (Section 2.1.2).The major difference between this and the other on-demand routing protocols is that it is beacon-less and hence does not require periodic hello packet (beacon) transmissions, which are used by a node to inform its neighbors of its presence. The basic approach of this protocol during the route construction phase is to establish a route by flooding RouteRequest packets (RREQ) in the network. The destination node, on receiving a RouteRequest packet, responds by sending a RouteReply packet (RREP) back to the source, which carries the route traversed by the RouteRequest packet received. This is called Route Discovery and is one of its two major phases along with Route Maintenance.

Route discovery and maintenance

Consider a source node that does not have a route to the destination. When it has data packets to be sent to that destination, it floods a RREQ throughout the network. Each node, upon receiving a RREQ, rebroadcasts the packet to its neighbors if it has not forwarded it already, provided that the node is not the destination node and that the packet's time to live (TTL) counter has not been exceeded.

To return the Route Reply, the destination node must have a route to the source node. If the route is in the Destination Node's route cache, the route would be used. Otherwise, the node will reverse the route based on the route record in the RREP message header (this requires that all linkes are symmetric). Nodes can also learn about the neighboring routes traversed by data packets if operated in the promiscuous mode (the mode of operation in which a node can receive the packets that are neither broadcast nor addressed to itself). This route cache is also used during the route construction phase in the way that if an intermediate node that receives a RREQ has a route to the destination node in its own route cache, then it replies to the source node by sending the RREP with the entire route information from the source node to the destination node.

As each node can flood the other with a RREQ it seems that loops could be formed as well as multiple transmissions of the same RREQ, for example by an intermediate node that receives it through multiple paths. To prevent this, each RREQ carries a sequence number generated by the source node and the path it has traversed. A node, upon receiving a RREQ, checks the sequence number on the packet before forwarding it, so it is forwarded only if it is not a duplicate RREQ.

In the event of fatal transmission, the Route Maintenance Phase is initiated whereby the Route Error packets (RERR) are generated at a node. That node send to the others so they will remove the routes that uses that hop, so all routes containing it are truncated at that point. Again, the Route Discovery Phase is initiated to determine the most viable route.

You can find the detailed description of this protocol and its improvements over the years in [20].

Advantages and disadvantages

Advantages of DSR protocol are easily guaranteed loop-free routing, operation in networks containing unidirectional links, use of only *soft state* in routing, and very quick recovery when routes in the network change. The DSR protocol is designed mainly for mobile ad-hoc networks of up to about two hundred nodes and is designed to work well even with very high rates of mobility.

The main disadvantage of this protocol is that the route maintenance mechanism does not locally repair a broken link. Stale route cache information could also result in inconsistencies during the route reconstruction phase. Another important disadvantage is that a considerable routing overhead is involved due to the source routing mechanism employed in DSR. This routing overhead is directly proportional to the path length (even worse in IPv6). In the other hand, although its route discover mechanism is really quick, the connection setup delay is higher than in table-driven protocols like DSDV.

2.2.3 AODV

The AODV (Ad-hoc On demand Distance Vector) is also an on-demand reactive protocol as well as DSR, where their major difference stems out from the fact that DSR uses source routing, in which a data packet carries the complete path to be traversed, however, in AODV, the source node and the intermediate nodes store the next-hop information corresponding to each flow for data packet transmission, it uses distant vector routing algorithm (Section 2.1.1). This protocol also uses the messages RREQ (Route Request), RREP (Route Replies) and RERR (Route Errors) under UDP protocol and its method is almost the same as in DSR where the source node floods the RREQ in the network when a route for a destination is not available (it may obtain multiple routes to different destinations from a single RREQ too).

The major difference between AODV and other on-demand routing protocols is that it uses a destination sequence number (DSN) to determine an up-to-date path to the destination. A node updates its path information only if the DSN of the current packet received is greater than the last DSN stored at the node. A RREQ carries the source identifier (SrcID), the destination identifier (DestID), the source sequence number (SSN), the DSN, the broadcast identifier (BcastID), and the time to live (TTL) field. DSN indicates the freshness of the route that is accepted by the source. When an intermediate node receives a RREQ, it either forwards it or prepares a RREP if it has a valid route to the destination. The validity of a route at the intermediate node is determined by comparing the sequence number at the intermediate node with the destination sequence number in the RREQ. If a RREQ is received multiple times, which is indicated by the BcastID-SrcID pair, the duplicate copies are discarded.

All intermediate nodes having valid routes to the destination, or the destination node itself, are allowed to send RREP packets to the source. Every intermediate node, while forwarding a RREQ, enters the previous node address and its BcastID. A timer is used to delete this entry in case a RREP is not received before the timer expires. This helps in storing an active path at the intermediate node as AODV does not employ source routing of data packets. When a node receives a RREP, information about the previous node from which the packet was received is also stored in order to forward the data packet to this next node as the next hop toward the destination.

All nodes active in the net transmit periodically *hello* messages (considered as special RREP messages). If one node does not receive *hello* from the neighbor means connection lost with them and they modify their routing table deleting that path. It also send a RRER to the other neighbor nodes that used that path. It can be done easily because each node keeps a list of all the active nodes in each communication.

You can find the detailed description of this protocol and its improvements over the years in [21].

Advantages and disadvantages

The AODV routing protocol is designed for MANET with populations of tens to thousands of mobile nodes and can handle low, moderate and relatively high mobility rates, as well as a variety of data traffic levels. AODV has also been designed to reduce the dissemination of control traffic and eliminate overhead on data traffic, in order to improve scalability and performance. Another main advantage of this protocol is that routes are established on demand and destination sequence numbers are used to find the latest route to the destination and the connection setup delay is lower.

One of the disadvantages of this protocol is that intermediate nodes can lead to inconsistent routes if the source sequence number is very old and the intermediate nodes have a higher but not the latest destination sequence number, thereby having stale entries. Also multiple RREP packets in response to a single RREQ packet can lead to heavy control overhead. Another disadvantage of AODV is that the periodic beaconing leads to unnecessary bandwidth consumption. In the other hand, as a reactive distance-vector protocol, the connection setup delay is quite high and considerably more than in the case of other reactive protocols like DSR.

2.2.4 OLSR

The Optimized Link State Routing Protocol (OLSR) operates as a table driven, proactive protocol using the link-state routing algorithm (Section 2.1.3). Each node selects a set of its neighbors nodes as multipoint relays (MPR). In OLSR, only nodes, selected as such MPRs, are responsible for forwarding control traffic, intended for diffusion into the entire network. MPRs provide an efficient mechanism for flooding control traffic by reducing the number of transmissions required.

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

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

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

You can find the detailed description of this protocol and its improvements over the years in [22].

Advantages and disadvantages

OLSR is well suited to large and dense mobile networks, as the optimization achieved using the MPRs works well in this context. The larger and more dense a network, the more optimization can be achieved as compared to the classic link-state algorithm. OLSR uses hop-by-hop routing where each node uses its local information to route packets.

OLSR is well suited for networks, where the traffic is random and sporadic between a larger set of nodes rather than being almost exclusively between a small specific set of nodes. As a proactive protocol, OLSR is also suitable for scenarios where the communicating pairs change over time: no additional control traffic is generated in this situation since routes are maintained for all known destinations at all times.

2.2.5 TORA

The Temporally-Ordered Routing Algorithm (TORA) is an adaptive routing protocol for multihop networks. A key concept in the protocol's design is an attempt to de-couple the generation of far reaching control message propagation from the dynamics of the network topology. The basic, underlying algorithm is neither distance-vector nor link-state; it is a member of a class referred to as link-reversal algorithms (Section 2.1.4). The protocol builds a loop-free multipath routing structure that is used as the basis for forwarding traffic to a given destination. The protocol can simultaneously support both source-initiated, on-demand routing for some destinations and destination-initiated, proactive routing for other destinations. It that possesses the following attributes:

- Distributed execution
- Loop-free routing
- Multipath routing
- Reactive or proactive route establishment and maintenance
- Minimization of communication overhead via localization of algorithmic reaction to topological changes

TORA is distributed, routers need only maintain information about adjacent routers (one-hop knowledge). Like a distance-vector routing approach, TORA maintains state on a per-destination basis. However, TORA does not continuously execute a shortest-path computation and thus the metric used to establish the routing structure does not represent a distance. The destination-oriented nature of the routing structure in TORA supports a mix of reactive and proactive routing on a per-destination basis. During reactive operation, sources initiate the establishment of routes to a given destination on-demand. This mode of operation may be advantageous in dynamic networks with relatively sparse traffic patterns, since it may not be necessary (nor desirable) to maintain routes between every source/destination pair at all times. At the same time, selected destinations can initiate proactive operation, resembling traditional table-driven routing approaches. This allows routes to be proactively maintained to destinations for which routing is consistently or frequently required.

TORA is designed to minimize the communication overhead associated with adapting to network topological changes. The scope of TORA's control messaging is typically localized to a very small set of nodes near a topological change. A secondary mechanism, which is independent of network topology dynamics, is used as a means of route optimization and soft-state route verification. The design and flexibility of TORA allow its operation to be biased towards high reactivity (low time complexity) and bandwidth conservation (low communication complexity) rather than routing optimality making it potentially well-suited for use in dynamic wireless networks.

You can find the detailed description of this protocol and its improvements over the years in [23].

Advantages and disadvantages

TORA quickly creates and maintains loop-free multipath routing to destinations for which routing is required, while minimizing communication overhead. It rapidly adapts to topological changes, and has the ability to detect network partitions and erase all invalid routes within a finite time.

As mentioned earlier, the protocol is designed to decouple (to the greatest extent possible) the generation of far-reaching control message propagation from the dynamics of the network topology. Consequently, there is no distance estimate or link-state information propagation. A negative effect of this design choice is clear: over time, as the link-reversal process proceeds, the destination-oriented DAG (see subsection 2.1.4) may become less optimally directed than it was upon creation.

TORA can be suitable in high density networks, but when the traffic is high and some packets are lost, TORA may perceive the packet loss as link breakage and reacts to it sending more update messages and causing even more congestion. So even TORA is designed to reduce overhead and deliver as fast as possible, it becomes sensible to changes in the amount of data sent and network size reacting always to it by the generation of a huge amount of routing load.

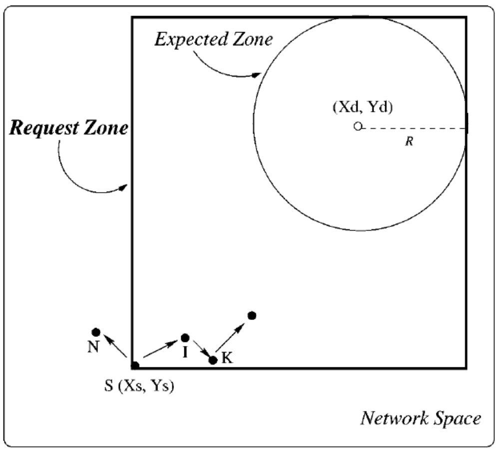
2.2.6 LAR

Like DSR, LAR (Location-Aided Routing) is an on-demand source routing protocol. The main difference between LAR and DSR is that LAR sends location information in all packets trying to decrease the overhead of a future route discovery. In DSR, if the neighbors of S do not have a route to D, S floods the entire ad hoc network with a route request packet for D. LAR uses location information for mobiles nodes to flood a route request packet for D in a forwarding zone instead of in the entire ad hoc network. The authors of [24] propose two methods used by intermediate nodes between S and D to determine the forwarding zone of a route request packet.

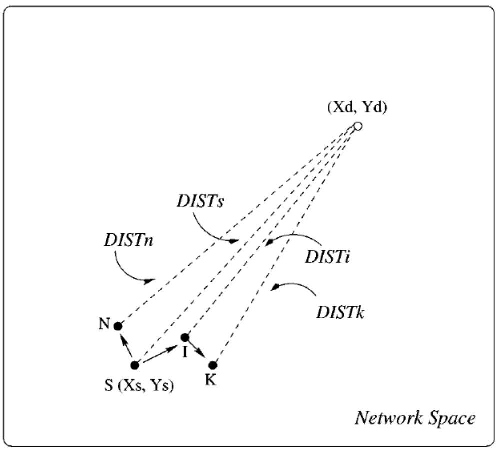
In method 1, called LAR Box, a neighbor of S determines if it is within the forwarding zone by using the location of S and the expected zone for D. The expected zone is a circular area determined by the most recent location information on D, (X_D,Y_D), the time of this location information, (t0), the average velocity of D, (Vavg), and the current time, (t1). This information creates a circle with radius R = Vavg (t1t0) centered at (X_D,Y_D). The forwarding zone is a rectangle with S in one corner, (X_S,Y_S), and the circle containing D in the other corner. If a neighbor of S determines it is within the forwarding zone, it forwards the route request packet further. A node that is not a neighbor of S determines if it is within the forwarding zone by using the location of the neighbor that sent the mobile node the route request packet and the expected zone for D based on the most recent available information. Thus the forwarding zone and the expected zone adapt during transmission.

In method 2, called LAR Step, an intermediate node determines if it is within the forwarding zone if the node is closer to D than the neighbor that sent the node the route request packet. Specifically, if the distance of the neighbor that sent the node the route request packet to D is Sdist, and the distance of the node that received the route request packet to D is Cdist, then the node will forward the route request packet if $Cdist \leq Sdist$.

Figure 2.1 illustrates the difference between the two LAR schemes. Consider figure 2.1(a) for LAR Box. When nodes I and K receive the route request for node D (originated by node S), they forward the route request, as both I and K are within the rectangular request zone. On the other hand, when node N receives the route request, it discards the request, as N is outside the rectangular request zone. Now consider figure 2.1(b) for LAR Step. When nodes N and I receive the route request from node S, both forward the route request to their neighbors, because N and I are both closer to (X_D,Y_D) than node S. When node K receives the route request from node I, node K discards the route request, as K is farther from (X_D,Y_D) than node I. Observe that nodes N and K take different actions when using the two LAR schemes.



(a) LAR Box



(b) LAR Step

Figure 2.1: Comparison of the two LAR methods.

Both LAR Box and LAR Step include a two stage route discovery method. In the first stage, the route request packet is forwarded according to either LAR Box or LAR Step. If a route reply packet is not received within the route request timeout period, then a second route request packet is flooded through the entire ad-hoc network. If a route reply packet is again not received within the route request timeout period, then D is considered unreachable. If D remains unreachable for 30 seconds, packets for D are dropped.

You can find the detailed description of this protocol in [24] and some improvements and its implementation in NS2 in [5].

Advantages and disadvantages

As LAR is based in DSR it has all advantages and disadvantages explained in 2.2.2. The difference is that LAR sends location information in all packets to (potentially) decrease the overhead of a future route discovery in wireless ad-hoc networks. The main disadvantage in LAR it that additional hardware is needed to obtain nodes' location.

Chapter 3

Mobility models

In this chapter the mobility models used in the case studies (chapter 5) are presented, compared and explained for which situations is more suitable one than another. As is pointed out in the section 1.3, a large percentage of the MANET publications do not use realistic environments, and one of the points is that they use mobility models which, although they have some interesting mathematical properties, do not represent real movements (also discussed in [25] and [15]). The most commonly used is the Random Waypoint mobility model. This is the first mobility model explained, so it can be proved why it is not suitable to model the movement of a person or a vehicle and therefore new mobility models are needed. Then, other mobility models developed to model more specific situation are presented: Gauss-Markov, Manhattan Grid and Disaster Area.

3.1 Randow Waypoint

The Random Waypoint (RWP) Mobility Model includes pause times between changes in direction and/or speed. A node begins by staying in one location for a certain period of time (pause time) and once this time expires, the node chooses a random destination in the simulation area and a speed that is uniformly distributed between [minspeed, maxspeed]. Then it travels toward the newly chosen destination at the selected speed. Upon arrival, the node pauses for a specified time period before starting the process again. However, since its behavior is independent of past motion (memoryless), it generates very unrealistic displacements. In figure 3.1 the traveling pattern followed by a node using this mobility model is plotted.

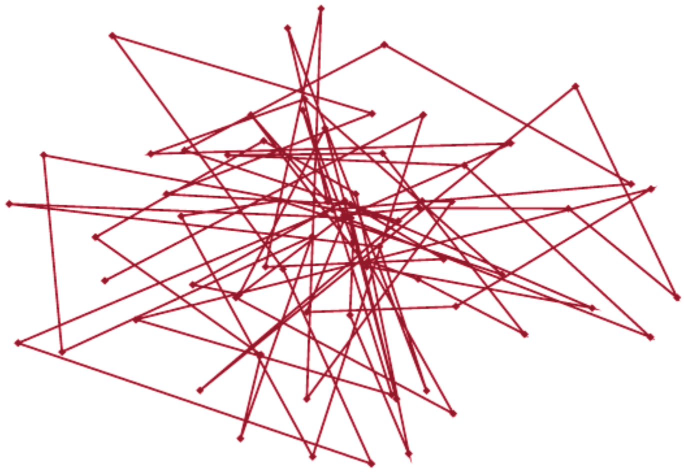


Figure 3.1: Traveling pattern using the Random Waypoint model

The mobility of RWP constantly causes topology change. The pause time and the maximum speed have effect on the mobility behavior of the nodes: if the maximum speed is low and the pause time is high, the network topology of a network becomes relatively stable. On the other hand, if the maximum speed is high and the pause time is small, the network topology is highly dynamic.

Two problems of the RWP mobility model are *sharp turn* and *sudden stop* [26]. *Sharp turn* occurs whenever there is a direction change in the range [90, 180]. *Sudden stop* occurs whenever there is a change of speed that is not relative to the previous speed. These problems can be eliminated by allowing the past speed and direction to effect the future speed and direction. For example, Gauss-Markov mobility model can solve these problems to achieve more realistic movement of mobile nodes.

The mathematical theory and a complete explanation of this model are presented in [27].

3.2 Gauss-Markov

In practical systems, a mobile user usually travels with a destinations in mind, therefore mobile's location and velocity in the future are likely to be correlated with its current location and velocity. The memoryless nature of random-walk models makes unsuitable to represent such behavior. Another widely used mobility model in cellular network analysis is the fluid-flow model, which is suitable for vehicle traffic in highways, but not pedestrian movements with frequent stop-and-go interruptions. The Gauss-Markov (GM) model represents a wide range of user mobility patterns including, as the two extreme cases, the random-walk and the constant velocity fluid-flow models.

In systems with correlated velocity mobility patterns, the largest-probability location of a mobile is generally not the cell where the mobile last reported. Thus a mobility management scheme that takes advantage of the predictability of the mobiles' location can perform better. In the GM model, the future location of a mobile is predicted based on the probability density function of the mobile's location, which is given by the GM model based on its location and velocity at the time of the last location update.

Taking into account the previous speed / direction (s_{n-1}/d_{n-1}), the mean speed / direction (S/D), and a random value ($s_{x(n-1)}/d_{x(n-1)}$) having a Gaussian distribution (these parameters are considered with different weights according to a randomness parameter α) the current speed and direction are then given by:

$$s_n = \alpha s_{n-1} + (1-\alpha)S + \sqrt{1-\alpha^2}s_{x(n-1)} \tag{3.1}$$

$$d_n = \alpha d_{n-1} + (1-\alpha)D + \sqrt{1-\alpha^2}d_{x(n-1)} \tag{3.2}$$

In figure 3.2 the traveling pattern followed by nodes using this mobility model is plotted. We appreciate all its properties where each movement, after an pause, follows a human likely reasoning and is not a completely random choice.

For more information on the mathematical theory and expressions of the GM model, a complete explanation is developed in [28].

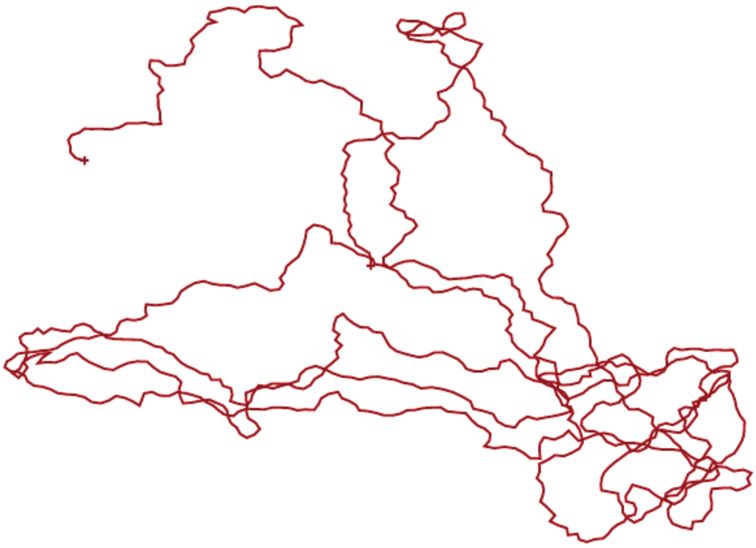


Figure 3.2: Traveling pattern using the Gauss-Markov model

3.3 Manhattan Grid

There are various mobility models in Vehicular Ad-hoc NETworks (VANET). One such model is the Manhattan Grid (MG) mobility model which uses a grid road topology. This mobility model was mainly proposed for the movement in urban area, where the streets are in an organized manner. The mobile nodes move in horizontal or vertical direction on an urban map.

The MG model employs a probabilistic approach in the selection of nodes movements since, at each intersection, a vehicle chooses to keep moving in the same direction or to turn (left or right with the same probability). The pause probability and maximum time of pause can also be defined so to model different situations such as traffic lights or traffic jumps.

The minimum speed the vehicles will present after a pause can also be defined as well as their mean speed and its standard deviation. To help to represent several situation in the city, also the speed change probability is a parameters we can choose, so in fluent traffic it will not change a lot, but in case of accident or traffic jump this probability should be higher.

In figure 3.3 vehicles following the MG model are plotted. We can see that all of them follow a grid topology movements and that due to all random variables explained before, each zone of the city has different density of vehicles.

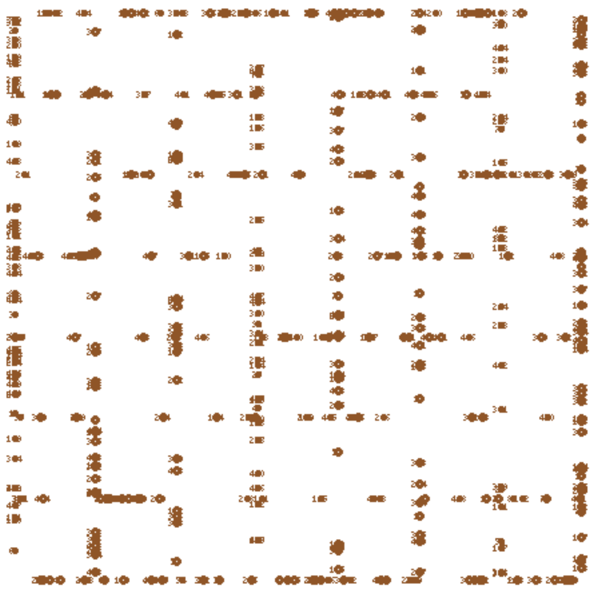


Figure 3.3: Vehicles distribution using the Manhattan Grid model

A more extended explanation of this model as well as other models used for vehicular mobility are presented in [29].

3.4 Disaster Area

In catastrophe situations, public safety units require robust communication systems. These catastrophe situations are considered at network performance evaluation as disaster area (DA) scenario. Due to the fact that any kind of pre-installed infrastructure may have been destroyed by the catastrophe, there is a demand for communication systems independent of such infrastructure. MANETs meet the requirement of being independent of any kind of infrastructure by their very definition.

In these situations, the users of communication systems that need reliable communication are civil protection forces, including rescue teams and fire brigades, so this mobility model is based on an analysis of tactical issues of civil protection. This analysis provides characteristics influencing network performance in public safety communication networks like heterogeneous area-based movement, obstacles, and joining/leaving of nodes.

Civil protection forces are strictly structured and their actions are strictly organized. The units do not walk around randomly, there is one leader or a group of leaders (technical operational command) which tells everybody where and how to move or in which area to work. In general, the movements are driven by tactical reasons called separation of the room. The disaster area and its surrounding is divided into different areas: incident site, casualties treatment area, transport zone and hospital zone (figure 3.4).

- *Incident site*: is the place where the disaster actually happened. In this area affected and injured people as well as fatalities are found, and the disaster has to be minimized. The affected and injured people are brought to the casualties treatment area.
- *The casualties treatment area*: this area consists of two places: the patients waiting for treatment area and the casualties clearing station. The patients waiting for treatment area is usually close to the incident site. The people are rescued from danger and wait there for their treatment. Then they are transported to the casualties clearing station which is still within the disaster area. After an extended first aid they are transported to hospital.
- *The transport zone*: is an area where transport units (ambulance coaches and rescue helicopters) wait in stand-by areas to take these people to hospitals. A the technical operation command is the responsible to coordinate all the transport logistics between the causalities treatment area itself and with the hospital, that is why is in the border with the casualties treatment area.
- *Hospital zon*e: usually far from the incident site, that is why vehicles are needed for the transport. After all the vehicles bring people here, they wait again for new transportation orders.

Each unit belongs to one of these areas. For example, a firefighter belongs to the incident site and a paramedic will work at one place in the casualties treatment area. The units sent to a specific location once will typically stay close to this location. Some

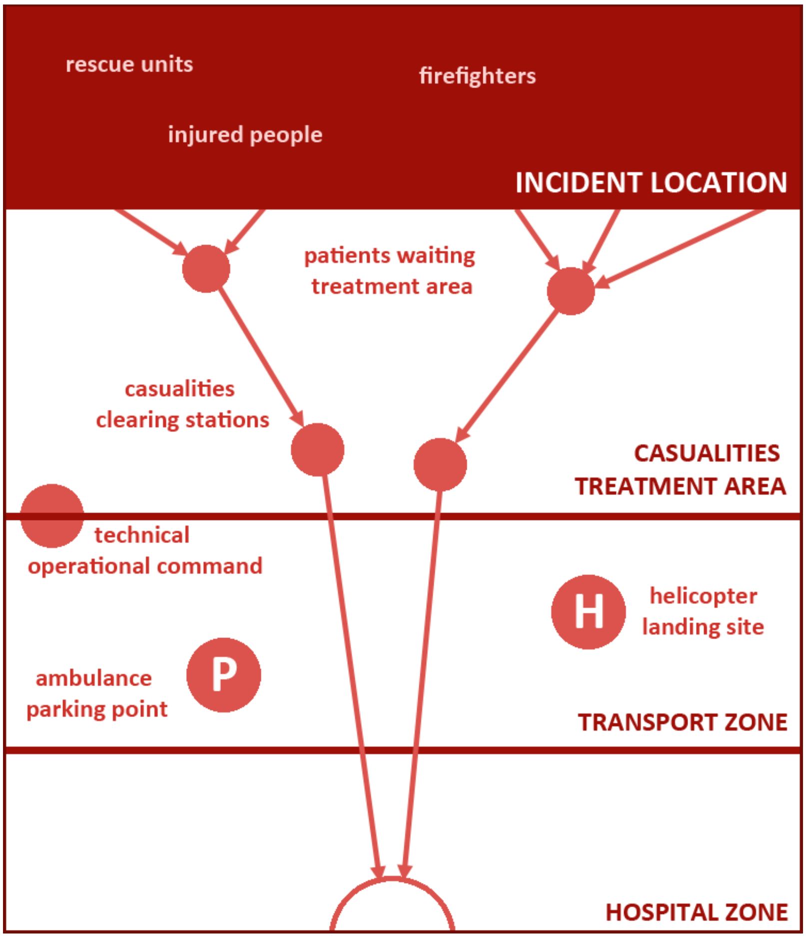


Figure 3.4: Separation of the room in Disaster Area model

of them are transport units which carry patients to the next area, the others do not leave the area. Thus, the area within which a unit moves depends on tactical issues but is restricted to one specific area.

The areas beside the incident location (places where tents are set up) are chosen by humans such that there are no obstacles inside these areas. At the incident location, the units (like firefighters) will destroy larger hindering obstacles. Smaller ones can be ignored, because they only have little impact on the movement. So there are only obstacles between different areas. Thus, they only affect transport units.

For the transport between the patients waiting for treatment area and the casualties clearing station, there are typically transport troops. These troops are pedestrians, for example four that carry a patient on a barrow. These troops pick up a patient and transport him on the direct way to his destination. Thus, they choose the optimal (shortest) paths avoiding obstacles. The hospitals are typically far away and not part of the disaster area communication network. The vehicles of the transport zone (ambulance coaches) transport the patients to the hospital, so these transport units typically arrive and leave the network perpetually. After a unit has arrived in the stand-by area it joins the communication network.

When there is a (tactical) request for a transport, one of the units in the stand-by area picks up the patient and brings him to the hospital. Thus, it leaves the communication network after having picked up the patient. These transport units as well as the troops above choose optimal paths and will avoid obstacles. Beside ambulances there may also be helicopters as faster transport units. However, helicopters typically use different communication channels and do not join the local disaster area communication network. So we see that there are heterogeneous speeds in a disaster area: some units (vehicles) move faster than others (pedestrians).

Especially transport units and troops often move in tactical formation (like four people carrying an injured person on a barrow or three persons in an ambulance coach) and this would imply a group mobility model. However, only the nodes that have a communication device are of interest for communication network analysis. Nowadays, only one node of such a tactical formation has a communication device. Thus, from the communication perspective in disaster areas (at the moment), there is no group movement.

As a conclusion, the analysis yields the following main characteristics:

- Heterogeneous area-based movement
- Movement on optimal paths avoiding obstacles
- Nodes join and leave the scenario in specific areas

To see how is each of these key points implemented, check [30] where you will also find a complete explanation with all the assumptions and algorithms.

3.5 Comparison

Some important mobility characteristics [31] likely to exist in MANETs are the following ones:

- *Temporal dependency of velocity*: This indicates that the velocities of a node are correlated within a specified time. In most real life scenarios, the speed of vehicles and pedestrians will accelerate incrementally. The direction change will also be smooth. Hence the velocity at current time period is dependent on the previous epoch. So the mobility model should have some memory to prevent extreme mobility behavior, such as sudden stop, sudden acceleration and sharp turn, which may frequently occur in the trace generated by the RWP model.

- *Spatial dependency of velocity*: This indicates that the velocities of different nodes are correlated in space (within a specified transmission range of each of other). In some scenarios such as battlefield communication, museum touring or rescue operations, a certain leader node in its neighborhood may influence the movement pattern of a mobile node. But RWP model considers a mobile node as an entity that moves independently of other nodes. Group mobility models or DA models capture the interdependent movement of nodes.

- *Geographic restrictions or obstacles*: RWP and its variants assume that the mobile nodes can move freely within the simulation field without any restrictions. However, in realistic applications in urban areas, the movement of a mobile node may be bounded by obstacles, such as buildings and streets or freeways. MG and DA mobility models can be used to study this characteristic.

The mobility characteristics that each model presents are shown in table 3.1.

After doing this comparison we see that the RWP mobility model does not provide some key elements which are necessary to describe different types of movement in real life. For his reason, and as nowadays lots of mobility models are being developed, the type of movement for each case study presented in chapter 5 must be analyzed in order to choose the mobility model that match the most.

Table 3.1: Characteristics of the mobility

Mobility model	**Temporal dependency of velocity**	**Spacial dependency of velocity**	**Geographical restrictions or obstacles**
Random Waypoint	No	No	No
Gauss-Markov	Yes	No	No
Manhattan Grid	Yes	No	Yes
Disaster Area	Yes	Yes	Yes

For example, Gauss-Markov's aim is to describe human likely movements, that is why it has memory with a temporal dependency of velocity. All the environments where the mobile nodes are on people walking may be suitable: conference room, staff of a hotel, shopping mall, etc. although we have to take into account that it does not provide the option of including obstacles.

In the other hand, Manhattan Grid is really focused in generating mobility for a grid section of a city. Although it provides lots of parameters to define the mobility of the vehicles, this mobility model is developed to be used only in this particular environment. It has also temporal dependency of velocity to model the behavior of a vehicle inside a city (traffic lights, turns, etc.) and the obstacles we can define are the city block's size.

Disaster Area gives us lots of flexibility and a really accurate model to use when modeling rescue operations. It allows us to define the shape and the number of the different areas and the obstacle wherever we like (not in fix positions like in the Manhattan Grid) and the velocity of the mobile nodes depends on the position (which area is the node in) and also on the time (if a vehicle or a pedestrian is asked to do some task immediately).

In [32], [25], [16] and [15] is studied and proved that the performance of a protocol can vary significantly with different mobility models and even with the same mobility model with different parameters. They also conclude that the performance of an MANET protocol should be evaluated with the mobility model that most closely matches the expected real-world scenario.

Chapter 4

Simulation tools

4.1 Performing the simulation

So far we have seen that we need at least two components to carry out this research: one that generate all the nodes, connections, set the routing protocol, etc. and another one to generate the mobility pattern that the nodes will follow. The programs chosen in this project are the network simulator NS-2 [33] and the mobility scenario generator Bonnmotion [34].

4.1.1 NS-2

NS is a discrete event simulator targeted at networking research. NS provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. It began as a variant of the REAL network simulator in 1989 and has evolved substantially over the past few years. In 1995 ns development was supported by DARPA through the VINT project at LBL, Xerox PARC, UCB, and USC/ISI. Currently ns development is supported through DARPA with SAMAN and through NSF with CONSER, both in collaboration with other researchers including ACIRI. Ns has always included substantial contributions from other researchers, including wireless code from the UCB Daedelus and CMU Monarch projects and Sun Microsystems.

NS simulator is based on two languages: an object oriented simulator, written in C++, and a OTcl (an object oriented extension of Tcl) interpreter, used to execute user's command scripts. NS has a rich library of network and protocol objects. There are two class hierarchies: the compiled C++ hierarchy and the interpreted OTcl one. with one to one correspondence between them.

The compiled C++ hierarchy allows us to achieve efficiency in the simulation and faster execution times. This is in particular useful for the detailed definition and operation of protocols which allows one to reduce packet and event processing time. Then in the OTcl script provided by the user we can define a particular network topology, the specific protocols and applications that we wish to simulate (whose behavior is already defined in the compiled hierarchy) and the form of the output that we wish to obtain from the simulator. The OTcl can make use of the objects compiled in C++ through an OTcl linkage (done usingtclCL) that creates a matching of OTcl object for each of the C++.

NS is a discrete event simulator, where the advance of time depends on the timing of events which are maintained by a scheduler. An event is an object in the C++ hierarchy with an unique ID, a scheduled time and the pointer to an object that handles the event. The scheduler keeps an ordered data structure with the events to be executed and fires them one by one, invoking the handler of the event.

In this project three different versions of NS-2 will be used for the simulation. Concerning MANET networks researching, the default distribution of NS-2 incorporates AODV, DSDV, DSR and TORA as routing protocols. The problem comes when we want to implement new protocols to run under NS-2. Usually from one NS-2 version to another implies some important changes that make impossible for a protocol implementation to be suitable for all of them, so modifies (sometimes no trivial at all) must be made. So when looking for implementations done by universities or research groups, you must take this into account. In this research, the choice made is shown in table 4.1.

Although TORA is by default in all NS-2 distribution lots of problems has been reported from 2.29 version onwards, so to guarantee a properly and stable behavior the 2.28 version is chosen. The LAR implementation is based on the implementation de-

Table 4.1: NS-2 version used for each protocol

Protocol	NS-2 version
AODV	2.34
DSDV	2.34
DSR	2.34
LAR	2.32
OLSR	2.34
TORA	2.28

veloped by Doctor T. Camp at the Colorado School of Mines [5] and then modified by Claus Christmann at Georgia Tech to be able to work in the version 2.32. OLSR was developed by Doctor F. J. Ros at the University of Murcia based on the RFC 3626 [22] and then improved by themselves to work under version 2.34. DSDV presents some glitch in earlier version for long simulation with lots of nodes, but in 2.34 is solved. AODV and DSR seamed to work well along the different ones so version 2.34 is chosen to be the latest.

In order to get used to the TCL language, the NS-2 simulator and learn how to deal with all the problems explained before, [35], [36] and [37] manuals and reference books were consulted as well as lots of forums for developers.

4.1.2 Bonnmotion

BonnMotion is a Java software which creates and analyses mobility scenarios. It is developed within the Communication Systems group at the Institute of Computer Science 4 of the University of Bonn, Germany, where it serves as a tool for the investigation of mobile ad hoc network characteristics. It will be used to create the different mobility scenarios explained in 3 and then transform them in a format that can be used by NS-2.

A part from the movements of the nodes in a certain scenario, BonnMotion also defines the size of the scenario, the number of nodes, their speed, the duration of the simulation and how many seconds we want to skip in order to avoid the transitory period of the movement of the nodes. A part from this default parameters we can also change some parameters for each mobility model such as the pause time in RWP or the number of nodes in each region in DA. In D you can find all the different parameters that can be configured in each kind of mobility model, and some examples of how some mobility files used in this project has been generated.

BonnMotion also provides tools which are also used in this research to analyze a certain scenario and obtain statistics of it. There are two modes of operation: to calculate overall statistics (averaged over the simulation time) and to calculate progressive statistics (values of metrics for certain points in time).

4.1.3 Considerations

For each simulation, the position and movements of the nodes are set randomly as well as the traffic between them. Bonnmotion is the responsible for the random properties of the positions and movements of the nodes, and for the traffic NS-2 random variables are used. Setting the random variables properly is a key point because if this is done wrong, some simulations can be correlated and we can come up with bad results even if we think we have performed enough simulations to describe a general case. As is explained in [38], lots of publications based in NS-2 have this problem because you must be careful when using NS-2 random generators. The tips in this paper are considered and at least 10 simulations are performed in each case so to provide good results and conclusions.

In order to make a fair comparison between protocols, the random seeds for the position and movements of the nodes as well as the ones for the connection parameters must not change till all the protocols are tested. This way, each protocol is tested under an scenario with exactly the same conditions. For this, in NS-2 the seed value must not be 0, because this means that each time the script runs with a different value.

Bonnmotion is the responsible to generate all movements commands in tcl according to the mobility model chosen. When they are generated, the movement patterns present a transitory period so we have to be careful to skip this first seconds since they do not present the properties of the mobility model wanted. Each model must be checked or analyzed in order to see how much time is necessary to be skipped.

Furthermore, according to [39], CBR traffic under UDP must be used to compare accurately the different protocols. TCP has a slow start mechanism that can influence the overall results.

4.2 Analyzing the results

NS-2 is a really complete network simulator, but it generate complex trace files and does not give the user any tool to extract results from the thousands of code lines generated. Thus, some scripts to analyze the results were created with AWK (programing language specifically designed for processing text-based data). Using this scripts, the following figure of merits can be obtained from a NS-2 trace file:

- *Average throughput*: the average number of bits arrived per second at each destination node. This metric can be used to figure out the suitability of the protocol for applications requiring high throughput.
- *Average packet delivery fraction*: the ratio of the data packets delivered to the destinations to those generated by the sources. Lower value reflects a larger number of packets being dropped due to link failures or network congestion.
- *Average end-to-end delay*: the time a packet takes from its sending till its arrival at the destination. It includes buffering, queuing, retransmission and propagation delays. The end-to-end delay is important because today many applications need a low latency to deliver usable results.
- *Packet delivery variation*: the time difference between the delay from a packet and the previous one. Not the average value, but the instant one, is useful to find out if a protocol is suitable for real time communications or not. If a packet arrives with a considerably different delay than the adjacent, this diminish the properties a real time communication must have so it can make it unreliable.
- *Normalized routing load (in bytes or packets)*: is defined as the number of control packets or bytes transmitted for each data packet or byte delivered. The evaluation of this metric can help capturing the scalability of routing protocols.

Sending more routing packets also increases the probability of packet collision and can delay data packets in the queues.

As not always all of them are useful depending on which is the aim of the system simulated, in each case study is discussed which ones are convenient. In B the main AWK script used to analyze and calculate the figures of merit from a trace files is presented.

4.3 Working methodology

The methodology followed during the research was:

1. Detailed analysis of a possible environment where MANET can be used.
2. Model the devices, geographical zone, mobility, traffic pattern, etc. into NS-2 files.
3. Discuss which figures of merit are important in the modeled system to decide which is the best protocol to use.
4. Create the shell script describing the whole simulation process and the variables to sweep and run it.
5. Analyze the results with AWK scripts.
6. Check if the results are coherent with the theoretical expectations. If not, explain why or check point 2 for mistakes and repeat the process.

Chapter 5

Case studies

In this section, several real implementation of MANET are analyzed and simulated. In each case study is explained why a MANET is a good choice in that situation, how all its properties are modeled and finally which routing protocol performs the best.

5.1 CS1: Shopping mall

Some companies are doing some research in the *intelligent shopping malls*. When the costumers go inside a commercial center, they would like to be informed of the special promotions, new products, etc. of their favorite companies. The idea of this research is to find a system which provides each costumer with this information after activating one application in a personal device, for example the mobile phone.

Each device with this service activated will join the network formed by the other devices and the transmission points, the shops. Even if the shops do not have a long transmission range (because of fading effects the transmission range inside buildings decrease a lot), the shops can reach a costumer that just entered from a far entrance, through the multi-hop network created. This system clearly describes a MANET structure.

5.1.1 Simulation model

In this MANET there are some static nodes, shops, which send information to some mobile nodes, customers. Differently to other MANETs, not all nodes send information to all others, only shops send information to the customers, and these ones are used as a repeater if a shop is not able to reach the addressee.

The simulation environment chosen is a shopping mall in the *Xujiahui* electronic shopping area in Shanghai (see figure 5.1). This zone is famous for selling all kind of electronic devices, so it is a perfect location where this system could be tested. The dimensions of the shopping mall are 60x100m^2 and depending of the time of the day, the flux of clients who has this service activated can vary (this is not the total number of clients in the shopping mall, but the ones with this add service on).

Figure 5.1: Xujiahui Shopping mall satellite picture

The mobility pattern of the mobile nodes (people) is the one described by the equations of Gauss-Markov which represents a realistic movement of people since this mobility pattern has memory, so do the trajectories of people walking. This model is more human likely than, for example, the Random Waypoint which after each pause time each node can take a complete random direction and this is not what a person does. The maximum speed of the people is 2 m/s, which is considered the limit between walking and running. To know more about this mobility model check section 3.2.

The Distributed Coordination Function (DCF) of IEEE 802.11 for wireless LANs is used as the MAC layer. The frequency band of 2.472 Ghz (channel 13) is chosen since this is a technology that all new personal devices (like PDAs, mobile phones, laptops...) incorporate. As this service is supposed not to decrease the device features and the information sent are text messages with pictures or small videos, using a transmission rate of 32kbps is suitable. The connection duration is described by a uniform random variable between 3 and 6 seconds, enough time to send the information at the previous bit rate. The advertisements demands are distributed along the time using an exponential variable to generate situations with more pressure than others, but not as stressful as a pareto distribution.

As this is an indoor simulation, a propagation model that takes into account multi-path propagation effects (also known as fading effects) must be used. The NS-2 shadowing model is used according to App. A.3 and a path loss exponent of 3 (in-building scenario but not so obstructed) and shadowing deviation of 5 (shopping mall is more likely as a factory than office but not so obstructed) are chosen. Taken into account this propagation model, the shops and customers are static terminals with a transmission range of 40m, which is a suitable range for commercial not so expensive devices in indoor environments.

The probability that a costumer subscribed to this service enters the shopping mall and wants news from one shop is Pc. Since is known that there are Ns shops offering this service and Nn new clients during the simulation time, the number of connection, Nc, is:

$$Nc = Pc\ Ns\ Nn\ St \tag{5.1}$$

$$Nc = Pc[\frac{connect}{client\ min\ shop}] \times Ns[shops] \times Nn[client] \times 8[min] = 8\ Pc\ Ns\ Nn \tag{5.2}$$

Where St is the simulation time set to 8 minutes. Nn varies depending on the day and the hour the shopping mall is analyzed, and Ns varies in dependence of the number of shops offering this service.

Furthermore, three different scenarios are studied: an scenario where few shops in the shopping mall offer this service (8), another one where all shops offer it (60) and a third one with an intermediate number of shops (24). In each one, the number of new clients during the simulation time is the parameter being swept from 10 to 90, because is assumed that a flux of 90 clients in 8 minutes is already a lot of customers.

Performing these simulations we cover all possible situation: from where few people is going shopping in the morning asking for lots of information, to a huge shopping day (like before Christmas) with lots of people but, for example, with still few shops have this service active. Figure 5.2 presents the simulation sketch of how is the inside of the shopping mall in the case of 8 shops and 90 clients. The big circles represent the shops and the small ones the clients.

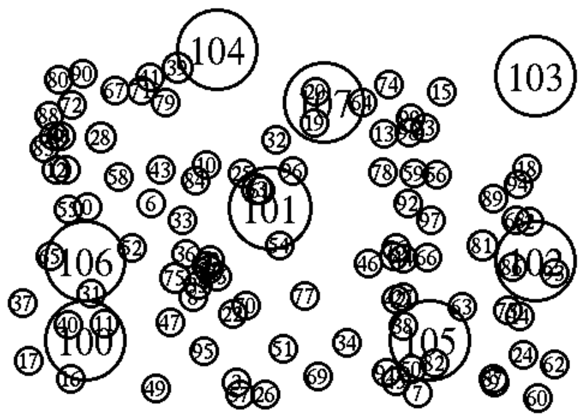


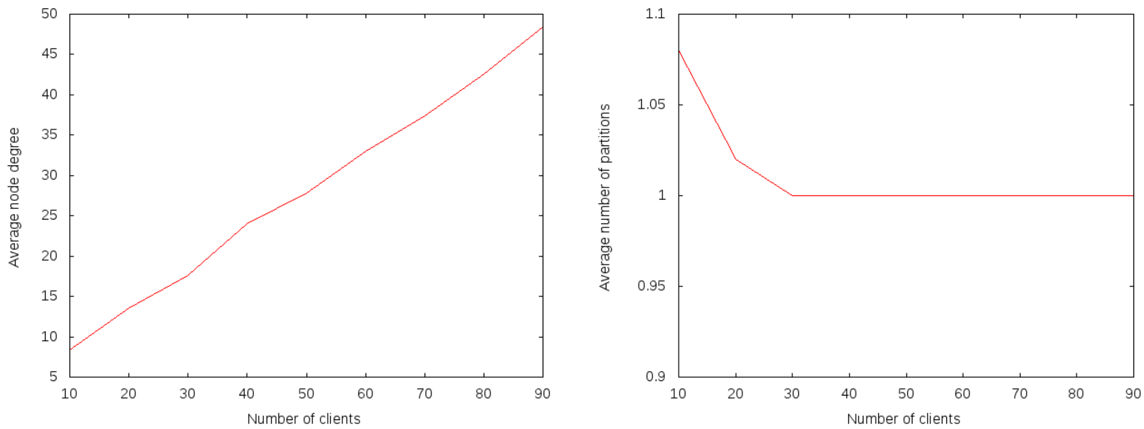
Figure 5.2: Shopping mall sketch of 100 clients

In App. C.1 you can check the TCL script used to describe this simulating environment in one of the scenarios of this case study, and in App. D how the mobility pattern can be generated for the clients as well as the static position for the shops. To avoid the transitory period in the generation of this mobility model, at least 3600 seconds must be skipped as it is explained in [34].

Model statistics

As the number of clients is increased, the density of nodes per square meters becomes high. The environments under this condition are called dense MANET networks, since there are lots of nodes in a small space. This type of networks become of specific interest because lots of protocols can not afford this scalability problem. As all the different scenarios are studied, we will be able to decide which protocol is the best to use at anytime, so it has enough scalability to work well independently of the number of shops or clients activated.

Bonnmotion can be used to calculate the statistics of the scenarios, so it can help us with the determination of the performance quality of the different protocols. Two interesting scenario parameters are plotted in figure 5.3 for the 8 shops scenario (the other scenario present almost the same properties): the average node degree and the average number of partition. The average node degree tells us to how many of the other nodes is one node connected, while the average number of partition tells us if the whole network is connected at all times (value 1) or if network partitions appear (value larger than 1).



(a) Average node degree

(b) Average number of partitions

Figure 5.3: 8 shops scenario statistics vs. number of nodes

The fact that the number of partition is almost always 1 and that the average node degree increases in a linear way means that this is an stable scenario, which is a property of the dense MANET with low nodes speed. In the next subsection is explained how the different protocols should perform under these circumstances.

5.1.2 Protocols and figures of merit

The protocols tested are AODV, DSDV, DSR, OLSR and TORA. DSDV can give good performance because we are in an stable scenario (as proved in the previous subsection) and this situation is an advantage for this proactive protocol. DSR may crash because is not designed to handle lots of nodes and connection but in the other hand, as source routing protocol, it delivers the packets really quick. One of AODV's purpose is just to be able to handle networks with lots of nodes. So AODV, DSDV and DSR are chosen to be the most widely used proactive and reactive routing protocols, which perform well under different of circumstances.

OLSR was initially developed for dense networks and it maybe be convenient, but if only a set of nodes are sending information (the shops) and not all of them, the route maintenance may not be done properly. TORA, as is explained in [8] and [40] presents bad scalability in terms of connections and data load and it fails quickly under these circumstances, so this aspect is checked. LAR is not tested because this system may not be able to give location information. In the table 5.1 the protocols constants chosen are listed.

Table 5.1: Shopping mall protocols parameters

(a) DSR

Timehop for hop route request	30 ms
Retransmit route request	500 ms
Size of header with n addresses	4n + 4 bytes
Buffer size	64 packets
Packet lifetime in buffer	30 s
Max rates for routes replies	1/s

(b) AODV

Route Request retry	5
Route Request rate limits	10 pkts/s
Gratuitous route reply flag	Enable
Advice route timeout	30 s
Hello interval	1 s
Allowed hello loss	10 s
Timeout buffer	2 s

(c) DSDV

Startup jitter	2 s
Instantaneous jitter	100 ms
Broadcast jitter	10 ms
Min. triggered updates	1 s

(d) TORA and IMEP

Beacon Periods	3 s
Max Beacon Timer	9 s
Max Tries	3

(e) OLSR

Willingness	Always
Hello Interval	2 s
TC Interval	5 s
Neighbour Hold Time	6 s
Topology Hold Time	15
Duplicate Message Hold Time	30 pkts/s

The figures of merit considered are the packet delivery fraction, throughput and the routing load (to see if a protocol cause or can cause network congestion). The delay and the packet delivery variation is not taken into account because this is not a real time communication scenario and is not important if someone receives a message some seconds later than someone else.

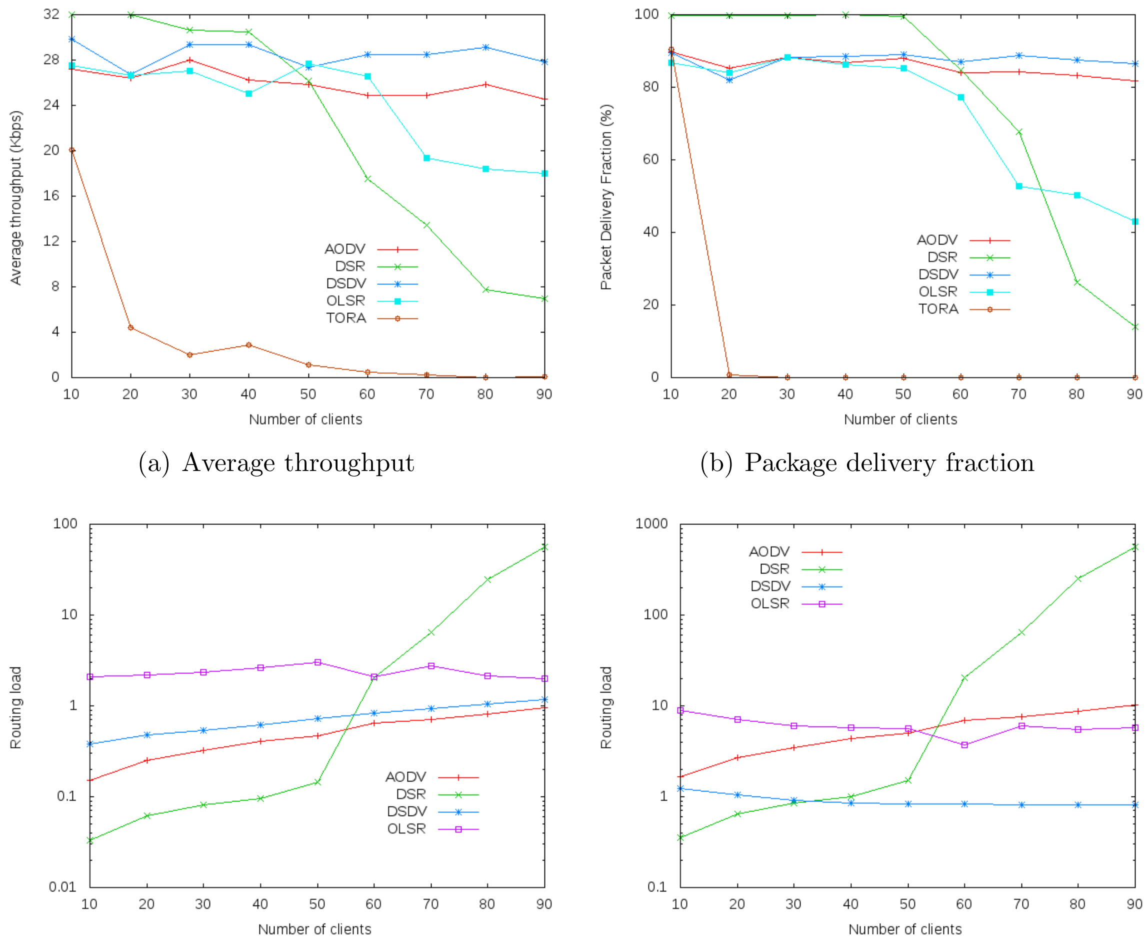
5.1.3 Results

8 shops scenario

As there are few shops, the probability that someone wants information from a shop is higher since there are not a lot of options. In this case, the Pc is set to 4% and according to equation 5.2 it means that, for example, for a flux of 50 new clients every 8 minutes, the 8 shops will send 128 messages. If everyone asked for information from all shop it will result in a total of 320 messages, but is very unlikely that in 8 minutes everyone ask for all shops. So the hypothesis is a realistic stressful situation to test the protocols, but still far from an unrealistic extreme condition. Figure 5.4 presents the results.

As I explained in the previous section, this is a dense MANET environment and the mobility pattern is a GM with low speed in order to simulate people walking. All these conditions make this scenario really stable what explains why DSDV is the protocol that performs better with around 90% of PDF and 30kbps of throughput along all the region of interest. As the routing load is not high, even with more connections there are few probabilities of causing congestion in the network because of this.

AODV is designed to deal with networks with lots of nodes, so it also performs quite good: around 85% of PDF, but 26kbps of throughput (as a reactive distance-vector protocol is not as fast as DSDV in delivering the packets). The problem is that there are lots of connections and AODV does not take advantage of this aspect like DSDV. DSDV reduces, not the number of routing bytes, but the number of routing packets needed as the number of clients and connections increase (around 10 for AODV and less than 1 for DSDV, figure 5.4(d)), while AODV needs more bytes and packets.



(a) Average throughput (b) Package delivery fraction

(c) Routing load (bytes) (d) Routing load (packets)

Figure 5.4: 8 shops figures of merit vs. number of clients

About DSR, it performs the best in all merit figures till 50 clients (nearly 100% of PDF and less than 1 of routing load in packets), but then as the number of clients increases (so does the number of connections), it breaks completely because its source routing nature makes the header of the packets too big to deal with them. We can note this effect in the routing load: till 50 clients it is really low, but afterwards it increases exponentially. Thus, we can define the breakpoint of this protocol between 50 and 60 clients.

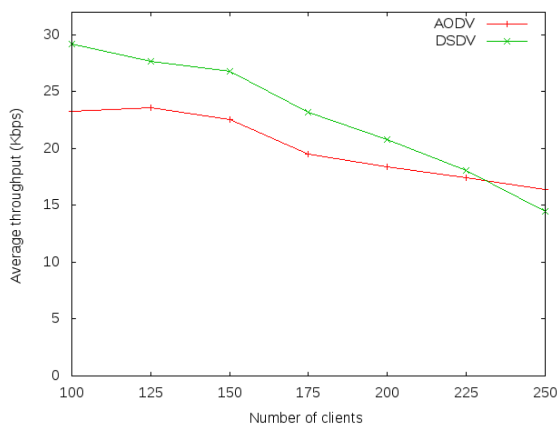
The conditions in this scenario should help OLSR to have one of the best performance but, the results show that from 60 clients onwards its performance decreases a lot (a PDF of around 90% drops down till almost 40% in the case of 90 clients). As it is described in the subsection 2.2.4, OLSR is convenient for to large and dense networks and, when the traffic is sporadic and between a large set of nodes, it contributes to the optimization of its link-state algorithm.

The problem here is that in this system not all nodes send information to each other, but only few nodes of the network, the shops. When the packets are sent only by a small specific set of nodes it does not perform as expected because the way to maintain the routes cannot be done properly, so old and maybe invalid information remains and is not updated. That is why, when the number of shops becomes much less than the total number of nodes, OLSR cannot work well anymore.

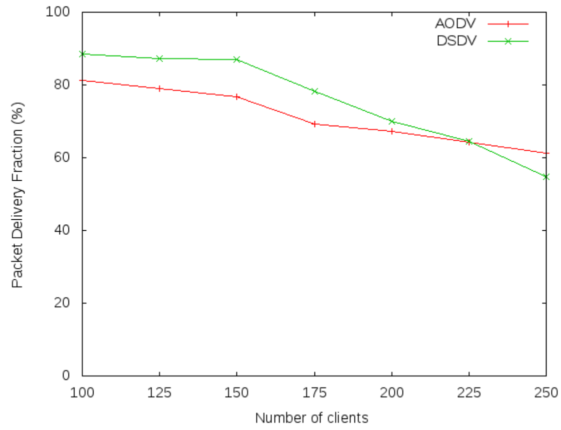
Finally, as is pointed out in the previous section, TORA was not developed to be used in dense MANET networks with lots of connections and it has few scalability and we see that it barely works with more than 10 clients. TORA is layered on top of IMEP, the Internet MANET Encapsulation Protocol, which is required to provide reliable, in-order delivery of all routing messages from a node to each of its neighbors, as well as notification to the routing protocol whenever a link is created or broken. The congestive collapse observed is most probably happening due to a positive feedback loop developed in TORA/IMEP.

The number of routing packets sent cause numerous collisions, which in turn cause data packets to be lost. The loss of these packets cause IMEP to erroneously believe that links to its neighbors are breaking. TORA reacts to the perceived link breakages by sending more update messages, which in turn cause more congestion. Moreover each update requires reliable delivery, which increases the exposure to additional erroneous links failure detections. The routing load for TORA is not presented because since the beginning it does not work well and its values reach the range of 10^6.

According to the good performance of DSDV and AODV in our region of interest, I wanted to stress more these protocols to see is they can afford it with still more clients, which would mean an hypothetic situation where the shopping mall is really full of people and everyone has a device with this service activated asking for shops information. The results are presented in figure 5.5 where we can see that both protocols are still able to handle the system till around 200 clients (DSDV always better) with a PDF of 70-75% and then they drop till less than 60% which would mean insufficient quality of service. This confirms that DSDV should be the protocol used in this scenario and, furthermore, as a proactive protocol it has less end-to-end delay than a distance vector reactive protocol like AODV.



(a) Average throughput



(b) Package delivery fraction

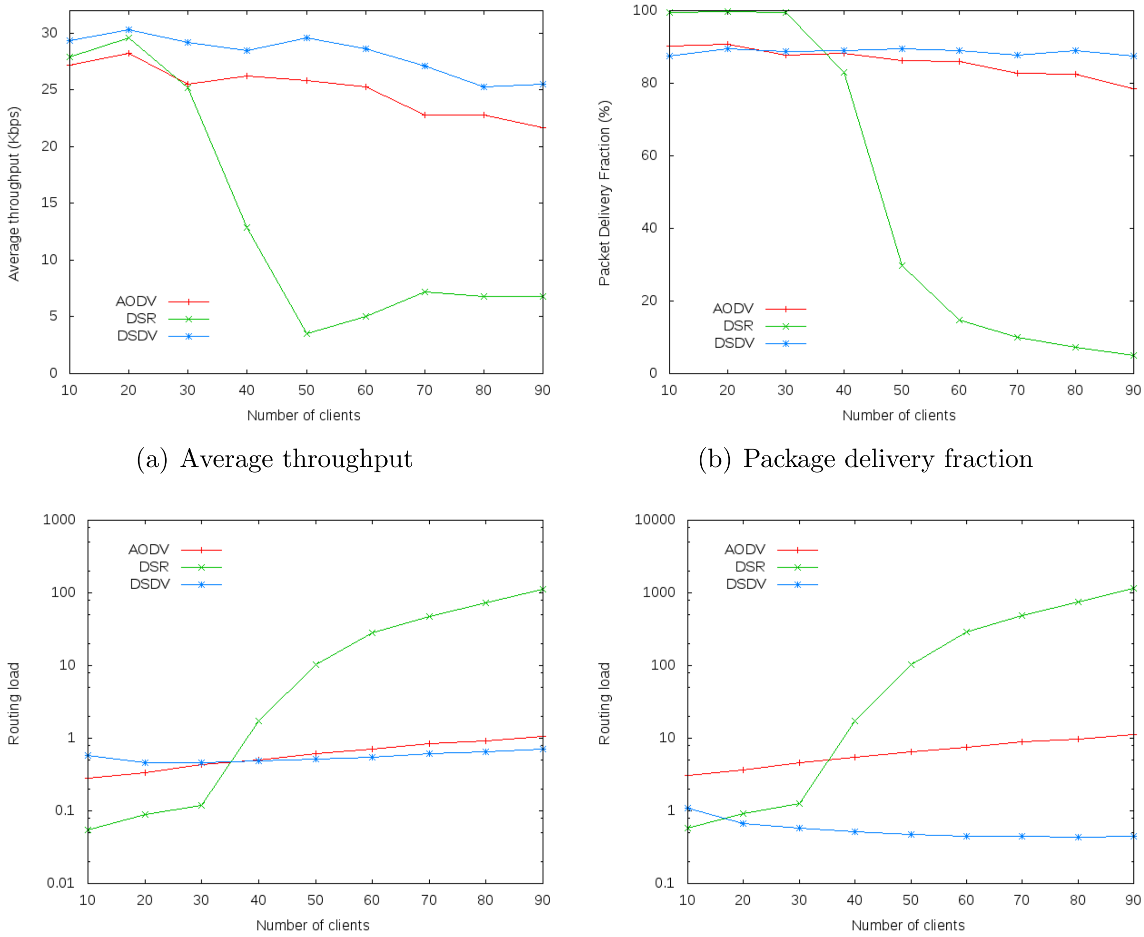
Figure 5.5: 8 shops figures of merit vs. high number of clients

24 shops scenario

In this scenario, as there are more shops where the clients can choose from, the probability that someone wants information from one specific shop is lower. In this case the Pc is set to 3% and according to equation 5.2 we get that, for example, for a flux of 50 new clients every 8 minutes, the 24 shops will send 240 messages. As is explained in the 8 shops scenario, this is a situation to stress the protocols, but in its usual work this system will not present worse stressful conditions.

In the previous section, it was proved that OLSR and TORA where not suitable for this MANET application and have not been tested in this scenario which present even more connections. DSR was also proved not to work well along all the region of interest, but it is tested in order to see where is its breakpoint so it can help to predict if AODV and DSDV will perform well as the number of clients increases more. Figure 5.6 presents the results.

DSDV and AODV perform almost in the same way as in the 8 shops scenario in terms of PDF and routing load, while the throughput decreases a little bit due to the increased difficulty in finding or maintaining a route now that there are more connections. As it



(a) Average throughput

(b) Package delivery fraction

(c) Routing load (bytes)

(d) Routing load (packets)

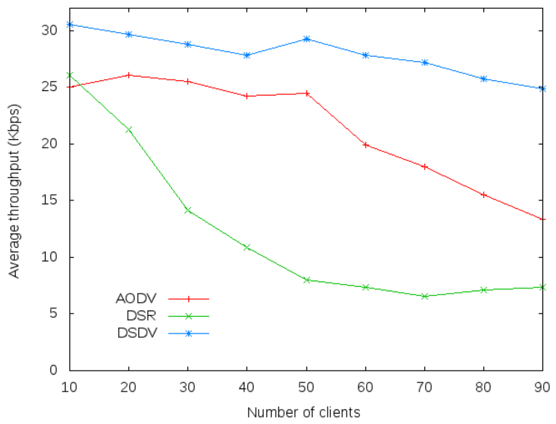
Figure 5.6: 24 shops figures of merit vs. number of clients

was explained in the description of this case study, the delay is not a key point, and while the PDF is high enough (90% for DSDV and 80-85% for AODV) and the routing load keeps low (in terms of packets less than 10 for AODV and less than 1 for DSDV), the systems will work properly.

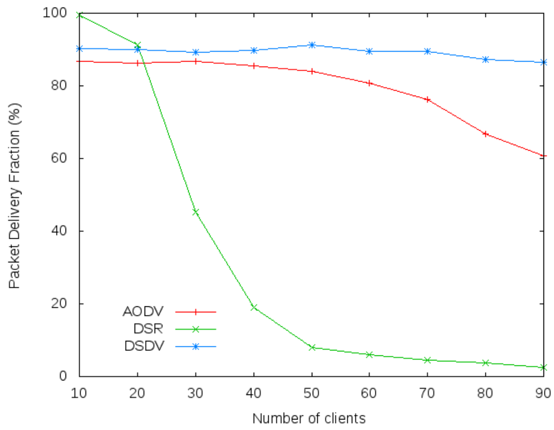
Under this conditions, DSR's breakpoint is between 30 and 40 clients, proving again that DSR is not a useful protocol to handle lots of connections. As now the breakpoint arrives before, we see that also the throughput and PDF of the AODV start to decrease before than in the 8 shops scenario, but not in the case of DSDV. So this breakpoint seams to be related with the reactive properties that also presents AODV. In the next section, this point is checked again in order to extract accurate conclusions.

60 shops scenario

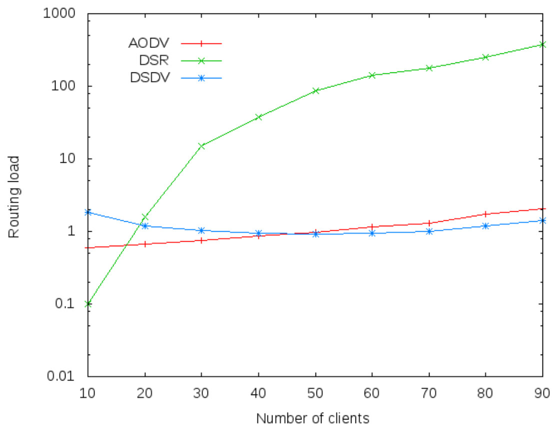
In this last scenario (where the hypothesis is that all the shops provides the add service) as there are much more shops than before, the people become very selective when choosing from which shop they want information. Thus, this case the Pc is set to 1.5%. According to equation 5.2 we get that, for example, for a flux of 50 new clients every 8 minutes, the 60 shops will send 360 messages. For the same reasons explained in the previous scenario, AODV, DSDV and DSR have been the protocols tested.



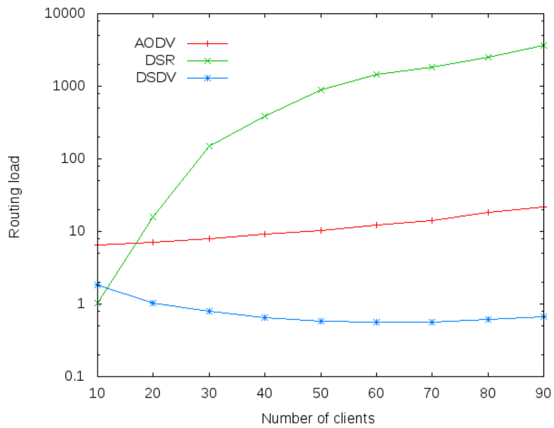
(a) Average throughput



(b) Package delivery fraction



(c) Routing load (bytes)



(d) Routing load (packets)

Figure 5.7: 60 shops figures of merit vs. number of clients

With this last case scenario, it can be concluded that DSDV is the only protocol able to handle all different situation in this shopping mall MANET. DSDV keeps the same values of throughput (more than 25kbps) and PDF (around 90%) as the previous scenarios. In the other hand, at 90 nodes AODV reaches only 60% of PDF and less than

13kbps. Furthermore, DSDV still presents a few packets routing load, around 1, while AODV surpass 10 and with some more connections it can cause network congestion.

As I stated in the previous section, in this last scenario we clearly see that the DSR breakpoint (now around 20 customers) is not related with the DSDV performance (because the breakpoint changed but the DSDV results maintain) but with the AODV's one. In the 24 shops scenario, the breakpoint was at 30-40 customers and at 90 clients AODV presented an 80% of PDF. Now, in the 60 shops scenario, the breakpoint occurs with 15-20 customers less than before, so also does the value of 80% of PDF for AODV which is around 70 customers.

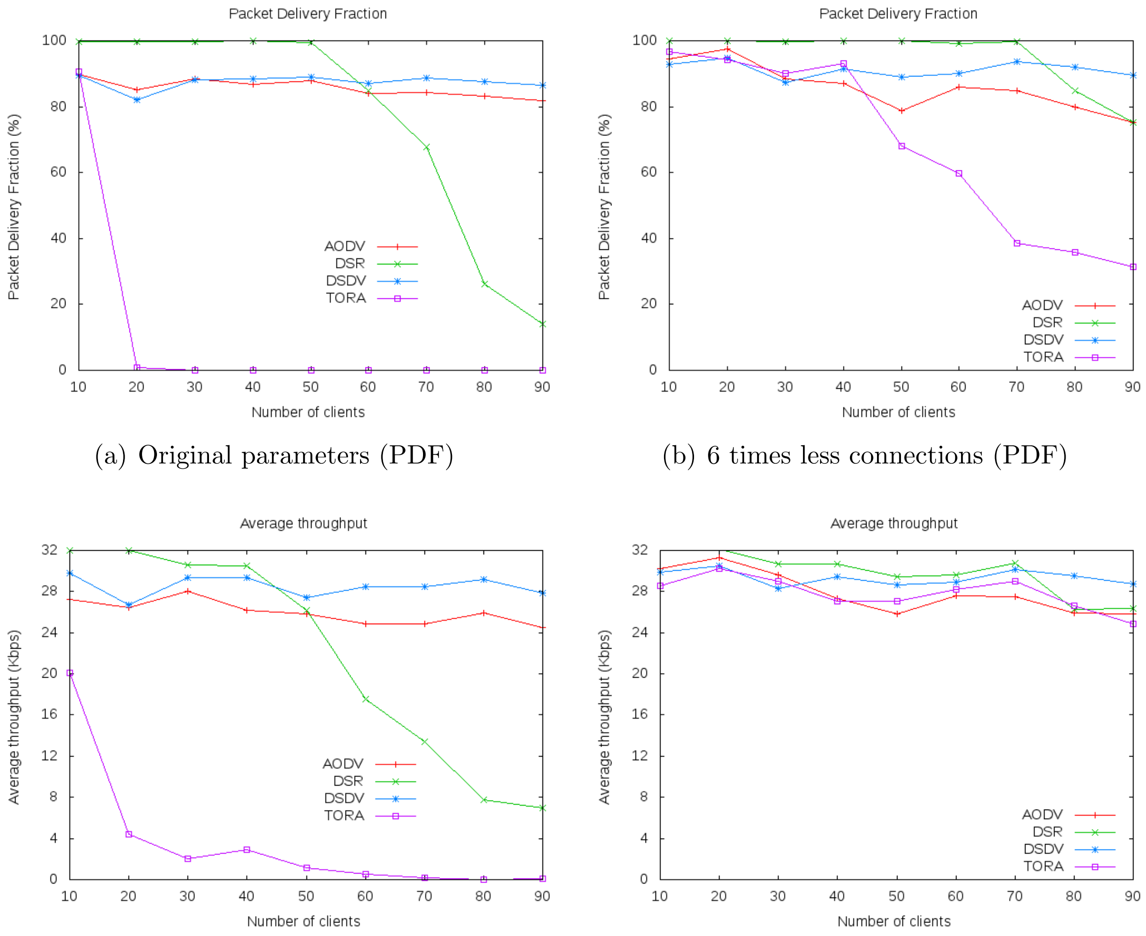
After analyzing all the different scenarios, it has been proved that DSDV performs the best independently of the number of shops or clients active in the shopping mall. It can be concluded that in the kind of systems which presents a high density of nodes with low speed and lots of connections, DSDV is a protocol that should be considered to obtain a good quality of service.

Particularity

To emphasize the particularity of this case study, I would like to prove that the number of connections established is an aspect that affects much more the performance of the protocols rather than only taking into account the whole amount of data transmitted. This is pointed out, because in lots of MANET publications they set a bit rate and decide the length distribution of the connections, but usually they do not consider the number connection as a relevant aspect and default or standard values are chosen.

The main handicap of this case study is not a high bit rate, but the huge amount of connections. For a routing protocol is much more challenging to establish lots of news connections than to maintain old ones, even if the amount of data to send is the same. To model more realistic MANET environments, which is the aim of this research, this point must be considered otherwise invalid results can be obtained.

To prove the previous explanation, two simulation have been carried out in the 8 shops scenario: one with the original parameters used in this case study and another one with 6 times less connections, but 6 times longer. At the end, the amount of data transmitted is approximately the same, but in figure 5.8 we see that the behavior of the protocols changed: now some perform better than others.



(a) Original parameters (PDF)

(b) 6 times less connections (PDF)

(c) Original parameters (throughput)

(d) 6 times less connections (throughput)

Figure 5.8: 8 shops scenario with different connection properties

In the 90 clients case, the original scenario demands 230 different connections while the modified scenario demands only 38. This is a great help for DSR because it does not need to carry a big overhead in the route discovery process so often, and now it is the protocol that performs the best in almost all the region of interest. TORA, till 40 clients even outperform the most of the other protocols, while in the real situation was unable to work at all with more than 10 clients. So it has been confirmed that is this point is not considered, completely different and inaccurate results can be obtained.

5.2 CS2: Basic vehicular network

Since lots of years ago different systems for intelligent houses are being studied, called domotica. Now some companies are trying to implement networks of sensors in a city to make the cities intelligent. *Urbiotica* is a company that is developing these networks in the city of Barcelona, which will offer, for example, the possibility to monitor the state of the garbage containers to offer a better and more optimal service or to know the parking places available in a street. Being able to send information to the vehicles nearby would offer a really better service since you would be able to inform them about the state of the traffic or to let them know where are some free parking places. Furthermore, if the cars could communicate with each other it will increase the vehicle safety by relaying required information from vehicle to vehicle.

With the growth and expansion of wireless communication technologies, considerable research efforts have occurred in the area of inter-vehicle communications. To allow inter-vehicle communication, vehicles must form some kind of network, called Vehiclar Ad-hoc NETwork (VANET), which is basically a MANET optimized in the vehicles environment.

5.2.1 Simulation model

In VANET, although is a type of MANET, there are lots of researches being done using really complex mobility models, scenarios and traffic pattern. Since is not the aim of this research to analyze VANET networks, an specific area of the city of Barcelona will be used, *Eixample*, which has a grid structure. The portion of the area is 650x650m^2 and each bloc of houses is 90x90m^2. In figure 5.9 you can see a satellite picture of this area of Barcelona where the yellow lines represent the streets.

As I said before and can also be confirmed by the street distribution seen in figure 5.9, the mobility model used is the Manhattan Grid (section 3.3). As the simulation takes place in the center of the city, the minimum speed when the vehicles are moving is 15 km/h and its mean is 40 km/h (considering that the maximum velocity permitted in this area of the city of Barcelona is 50 km/h). Inside a city, the speed of a vehicle is

Figure 5.9: Barcelona *Eixample* neighborhood satellite picture

not as constant as, for example, in a highway, that is why the speed change probability is set to 60% and the turn probability in each block is 30%.

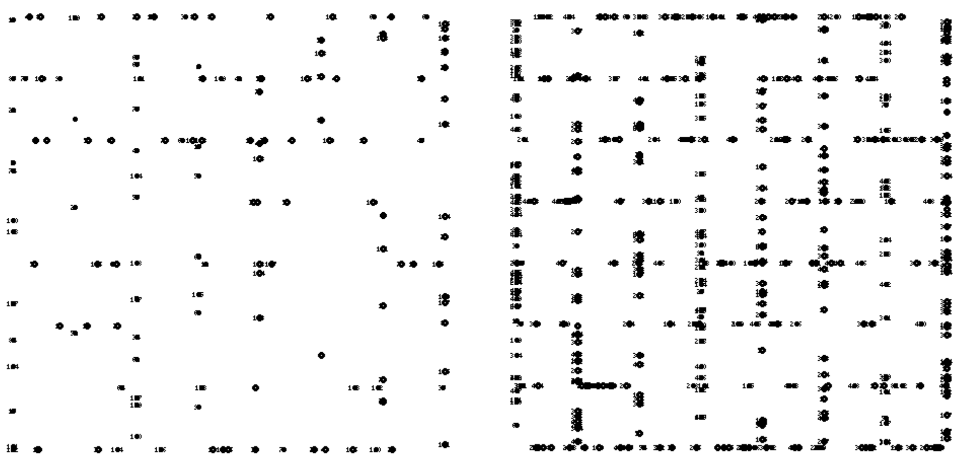
The maximum time a vehicle can be paused and the pause probability will vary depending of the status of the traffic, construction work, accidents, etc. When the traffic is continuous (few pause probability), the pause time is considered the time of the longest traffic light, 25 seconds. When probability increases, this maximum pause time increments due to the previous reasons mentioned and can reach the value of 40 seconds.

This area of Barcelona is considered as a shadowed urban area, so to take into account the fading effects, the Shadowing model is used. According to how is considered this propagation model in NS-2 (see App. A.3) and the propagation model analysis for VANET networks presented in [41], a path loss exponent of 3.7 (outdoor city environment not lots of obstacles) and shadowing deviation of 7 (outdoor obstructed environment) have been chosen. Taken into account this propagation model, the vehicles wireless devices posses a 200m transmission range.

Some applications already developed for the communication between vehicles uses a bit rate of 16kbps and they exchange information about the traffic state, possible routes to avoid congestions, advices of the states of the roads ahead, geographical information, etc. For these purposes and as the connection time is short, its duration is defined with a uniform random variable between 2 and 5 seconds.

A pareto distribution is used to model the arrival time of the connections in order to model possible situation such as when an accident occurs and the vehicles need to communicate this new information to the others at similar time. Although a pareto distribution is used, its properties are not set to generate a possible scenario where almost all connection are concentrated in a really short time. This will not recreate realistic conditions because even in an accident scenario, the different vehicles will not realize of it all at the same moment, so not all of them will send information at once.

Furthermore, two different scenarios are tested: one with 150 and another one with 500. This last one represents the worse case where the streets of the city have a big congestion of cars. This will point out if a protocol presents a good scalability in terms of nodes moving at high speed. Figure 5.10 presents the situations in the case of 150 and 500 vehicles within this area, so the vehicles density can be appreciated.



(a) 150 vehicles (b) 500 vehicles

Figure 5.10: Vehicles density in different situations

The simulation time is defined at 6 minutes, time enough to check how the protocols deals with the connectivity problems in this quick changing conditions. The amount of connections generated is the number of the simulation duration divided by 10 so, in average, every 10 seconds a vehicles sends information to another one in the network.

In App. C.2 you can check the TCL script used to describe this simulating environment in one of the scenarios of this case study, and in App. D how the mobility pattern can be generated with all its options. As it is proved in [34], a safety margin of 3000 seconds must be skipped to not test the protocols in the transitory period of the movement generator.

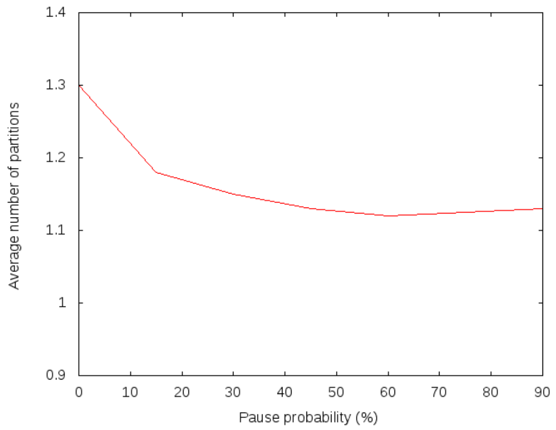
Model statistics

As it was explained in the model statistics of the previous case study (subsection 5.1.1), the scenario statistics of the average node degree and the average number of partition are studied. The results for these parameters along the pause probability for both scenarios (150 and 500 vehicles) are presented in figure 5.11.

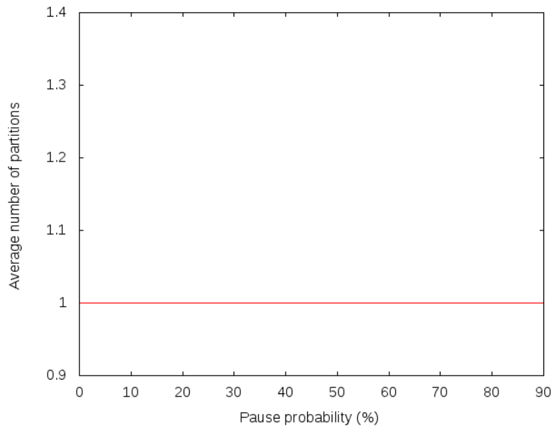
In both cases the average node degree presents the same constant behavior (less than 0.5% of variation from the average value). Even if the vehicles moves continuously or not in this grid region, they are equidistantly distributed all over it. This creates an stables condition, so the average number of vehicles that can be reached by another one tends to the same value.

The point here is that, actually, the number of other vehicles that can be reaches without doing any intermediate hope is a really few portion of the total. This aspect cause the need to follow long multi hope paths in some occasions and, adding the high speed of the nodes, this generate situations that some protocols are not able to handle: they must be fast in discovering a long route and sending the data or the high mobility will invalidate the path.

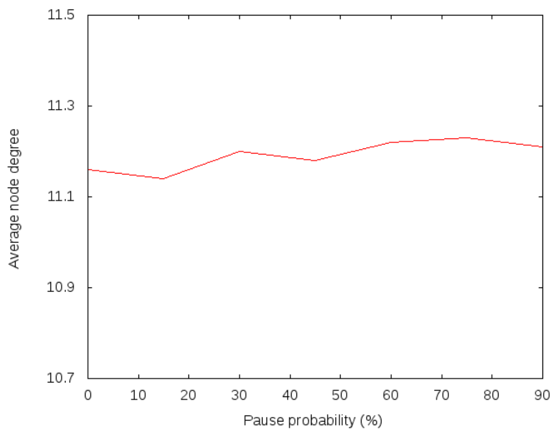
In the other hand, the 500 vehicles scenario present better condition in terms of network partition. This is not strange since we saw in figure 5.10(b) that with this number of vehicles this grid region is really crowded. Some vehicles of the 150 vehicles scenario can become unreachable due to the empty space regions, so data will be lost if trying to send them during this time.



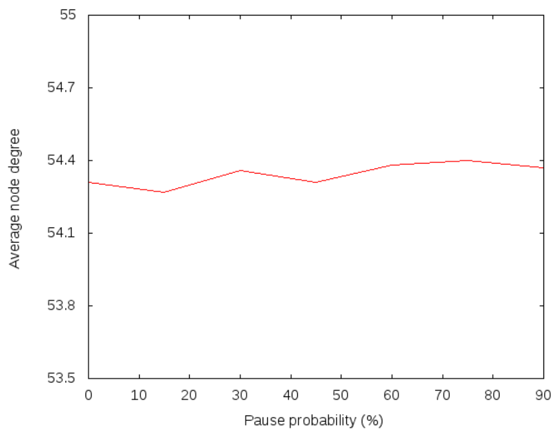
(a) Number of partitions (150 vehicles)



(b) Number of partitions (500 vehicles)



(c) Node degree (150 vehicles)



(d) Node degree (500 vehicles)

Figure 5.11: Scenario statistics vs. pause probability

5.2.2 Protocols and figures of merit

As it was pointed out with the scenario statistics, a handicap of VANET is that the high and continuous speed of the nodes cause the routes to change quick, so a key point is to figure out a new route as quick as possible. To see this effect and know which protocol can deal better as well as in static situation (traffic jam in rush hour for example) as well as in continuous traffic flow situation (for example during the night) the pause probability of the cars is the parameters swept.

AODV is tested due to its properties of high scalability in number of nodes and dealing good with high speed. I assume that the car devices provide geographical information, so LAR is also tested because in this case of large dimensions the geographical information is useful. DSR do not have good properties in scalability but as a source routing protocol, it delivers the packets fast and this can help against the quick changing in the routes.

Finally, TORA is also tested because its main purpose is not to find the optimal path, but to find a path as fast as possible, a good point for this scenario where the network is continuously changing. In the table 5.2 the protocols constants chosen are listed.

In other publication analyzing metropolitan environment for VANET such as [42] and [43], AODV is found to be one of the protocols performing the best while TORA is completely unsuitable, although theoretically it seams a good option. These results are checked.

Table 5.2: VANET protocols parameters

(a) AODV

Route Request retry	5
Route Request rate limits	10 pkts/s
Gratuitous route reply flag	Enable
Advice route timeout	30 s
Hello interval	1 s
Allowed hello loss	10 s
Timeout buffer	2 s

(b) DSR

Timehop for hop route request	30 ms
Retransmit route request	500 ms
Size of header with n addresses	4n + 4 bytes
Buffer size	50 packets
Packet lifetime in buffer	30 s
Max rates for routes replies	1/s

(c) LAR (both methods)

Timehop for hop route request	30 ms
Retransmit route request	500 ms
Size of header with n addresses	4n + 40 bytes
Buffer size	50 packets
Packet lifetime in buffer	30 s

(d) TORA and IMEP

Beacon Periods	3 s
Max Beacon Timer	9 s
Max Tries	3

The figures of merit considered are the packet delivery fraction, throughput and the end-to-end delay. Usually it will not be the situation, but some information may need to be delivered fast in emergency situations so, if some protocol present good enough PDF, the throughput and the delay can help to decide the best protocol to use.

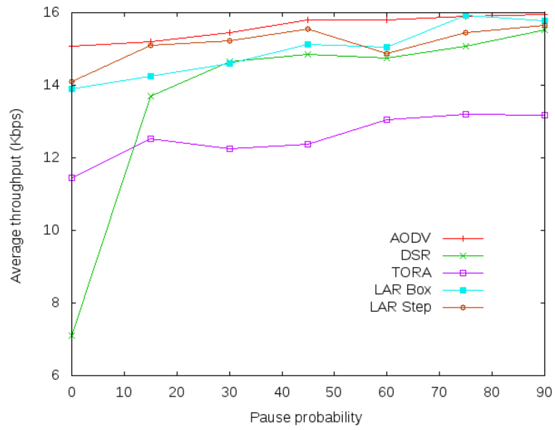
5.2.3 Results

Network of 150 vehicles

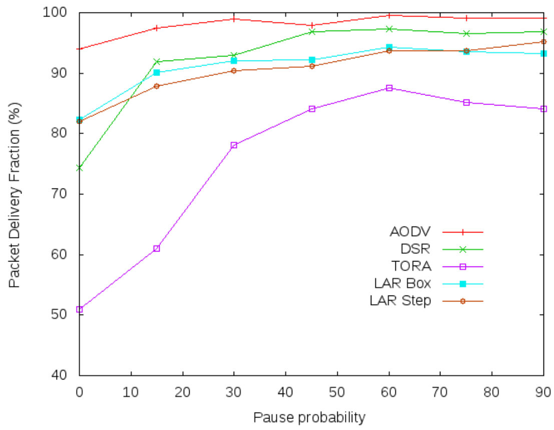
As I explained before, for each scenario the parameter swept is the pause probability which goes from 0% (which represent completely fluent traffic) until 90% (which is the situation of a traffic jam) with steps of 15%. For every step the maximum pause time also increases (linear increase of 3 seconds) because with fluent traffic the maximum time can only be caused by a traffic light but, in the other cases, the vehicles can be stopped for more time due to a traffic jam or an accident. Figure 5.12 presents the results for this first scenario with 150 vehicles.

As expected, AODV performs really well with always more than 95% of PDF and 15kpbs of throughput, and when the pause probability increases, so the network is more stable, this values are 99% and 15.8kbps respectively. The distance-vector nature of AODV helps in the previous figures of merit but, as it is already known, it makes the end-to-end delay higher than other protocol, around 0.4 seconds. As I said in the in previous sections, this figure of merit is not so important while it does not reach an extreme high value.

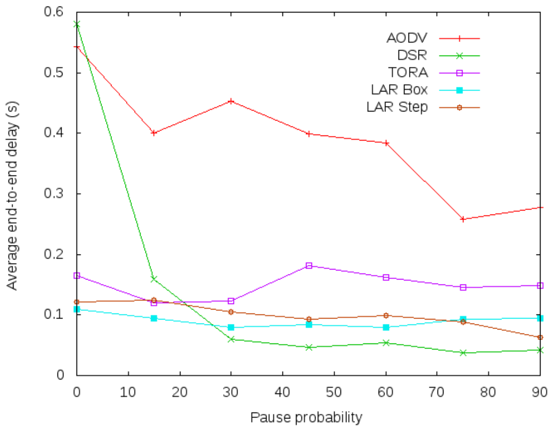
When the pause probability is more than 15%, DSR does not perform bad neither, more than 90% of PDF and considerably less delay than AODV (less than 0.1 seconds). Its problem comes when the mobility is high, where DSR presents poor results (73% of PDF, only 7kbps of throughput and a high delay of 0.6 seconds) because the source routing algorithm with high speed is difficult to achieve: when the route discovery packet reaches the destination, it has to go back fast carrying a big header or the quickly moves of the vehicles will make the path change again. In the other hand, both LAR methods are also source routing but, with the help of the location information, they improve the performance in high mobility conditions (more than 80% of PDF) and the rest present similar values as DSR, so their performance is much better.



(a) Average throughput



(b) Package delivery fraction



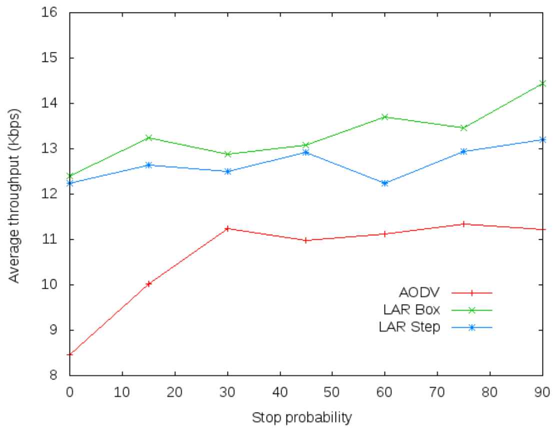
(c) Average end-to-end delay

Figure 5.12: 150 vehicles figures of merit vs. pause probability

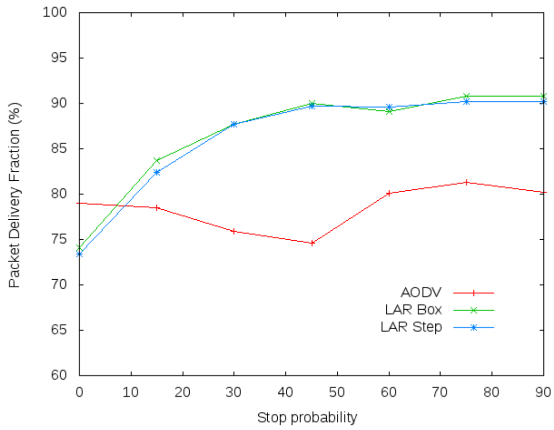
Finally TORA's performace is always worse than the other protocols (a part from the end-to-end delay) and in high mobility it only reaches 50% of PDF and in the best case of high pause probability 85%. Although the TORA's aim is to find a path as fast as possible to avoid the route changing (which was the problem that was facing DSR) its link-reversal algorithm is not suitable to deal with so many nodes, which is already stated as a disadvantage in its description.

Network of 500 vehicles

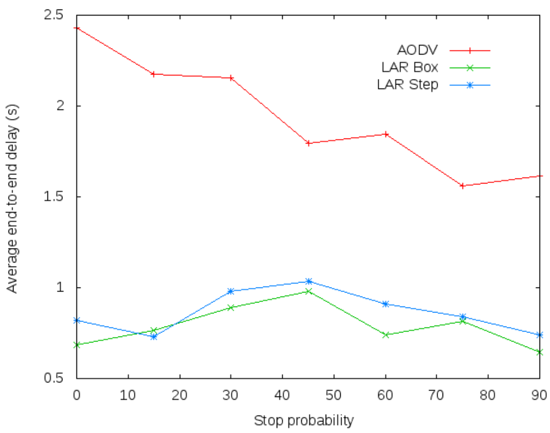
As it is checked in the previous scenario, DSR performs well when the pause probability increases, but when the nodes moves continuously its performance is poor. For this reason, and also because this new scenario present much more nodes than the previous one (which makes an even worse situation for source routing protocols), DSR is not tested. As TORA's performance decreases as the number of nodes increases, is neither tested. Figure 5.13 presents the results for this new 500 vehicles scenario using AODV, LAR Box and LAR Step.



(a) Average throughput



(b) Package delivery fraction



(c) Average end-to-end delay

Figure 5.13: 500 vehicles figures of merit vs. pause probability

AODV is supposed to work well with lots of nodes even if they move fast, while a source routing based protocol like LAR might decrease its performance, but the results does not show this. In both LAR implementation we can see the same effect as before where in high mobility the performance is worse, 75% of PDF, but when the pause probability increases it reaches 90%. As source routing, their delay is low (always less than 1 seconds) so the throughput is also good (from 12kbps until more than 14kbps) and LAR Box always performs better than the other method, Step. Thus, as it was explained when deciding which protocols to test (subsection 5.2.2), the location information in this scenario helps a lot to increase the performance of a routing protocol.

In the other hand, AODV never achieve more than 80% of PDF neither more than 11kbps of throughput. In the previous scenario we saw that the performance of AODV was almost independent from the mobility grade, and when the traffic was fluent its results were almost the same. Now this has changed and with few pause probability the delay is really high (almost 2.5 seconds) and the throughput decreases until 8.5kbps. The rest of the probability steps present a delay between 1.5 and 2.2 seconds, which starts to be considerable if urgent information needs to be resend to more vehicles creating a chain of a large delay.

Most of the nowadays applications being tested in this new VANET environments do not use more than 150 vehicles in the area considered in this case study. Thus, for the moment, and because its performance in the first scenario is much better than the rest of the protocols, AODV can be a good option. However, as the applications grow and more vehicles use them, it should be considered that AODV could not be suitable anymore so more research in looking for better protocols must be done. LAR Box could be an option but, as it uses source routing algorithm, is not supposed to work with so many nodes. Other protocols taking advantage of the geographical information may be a better option. For example, GPSR which uses greedy forwarding to forward packets to nodes that are always progressively closer to the destination.

5.3 CS3: Rescue operation

In this case study a real rescue operation environment is modeled and analyzed. Rescue operation are performed after a catastrophe like a train crash, which is the scenario chosen in this case study. The train crashed against one station so the incident location (IL) is surrounded by destroyed building and rubble. Thus, is necessary to enter in this area from two different points in order to reach everyone inside, which implies that two waiting patient treatment areas (WPTA) are necessaries. A casualty clearing station (CCS) is placed close to each one of the WPTA and also another one is in an intermediate position as backup. An ambulance parking point (APP) is situated in the east using clear transportation routes to each one of the CCS and hospitals. Everything is coordinated by a technical operation command (TOC).

The area of the disaster and the zone used to carry out all rescue efforts, medical assistance and transportation is 225x300m^2. The number of units working on it is about 700, but not all of them have communication devices. In the next section you can see how many units with communication devices are in each zone and a sketch of the operation.

5.3.1 Simulation model

As this is a rescue operation performed after a train crash disaster, the mobility model used is the DA described in the section 3.4. According to the model and the previous explanation, the size, the personal and vehicles with communication devices of each zone are:

1 IL (30m x 150m): 15 firefighters and 5 transportation units are going inside this zone to take the people out.

2 WPTA (16m x 36m): 17 vehicles help with the transport of people to the casualties clearing stations while 3 pedestrian team coordinate it.

3 CCS (26m x 50m): In each area 15 units of medics are working.

1 TOC (20m x 25m): To coordinate all the transport logistics 6 tactical units are responsible.

1 APP (40m x 40m): A total number of 25 ambulances transport people to the hospital. There are also 5 pedestrian teams to help.

For each zone, the entrance and exit points are set on the middle of the areas' borders. The scenario entry and exit points for the vehicular transport units (like ambulances) are set at the down right border of the scenario. Also some obstacles are added to the scenario to represent, for example, the rubble or buildings in fire. A sketch of how the scenario looks like is plotted in figure 5.14.

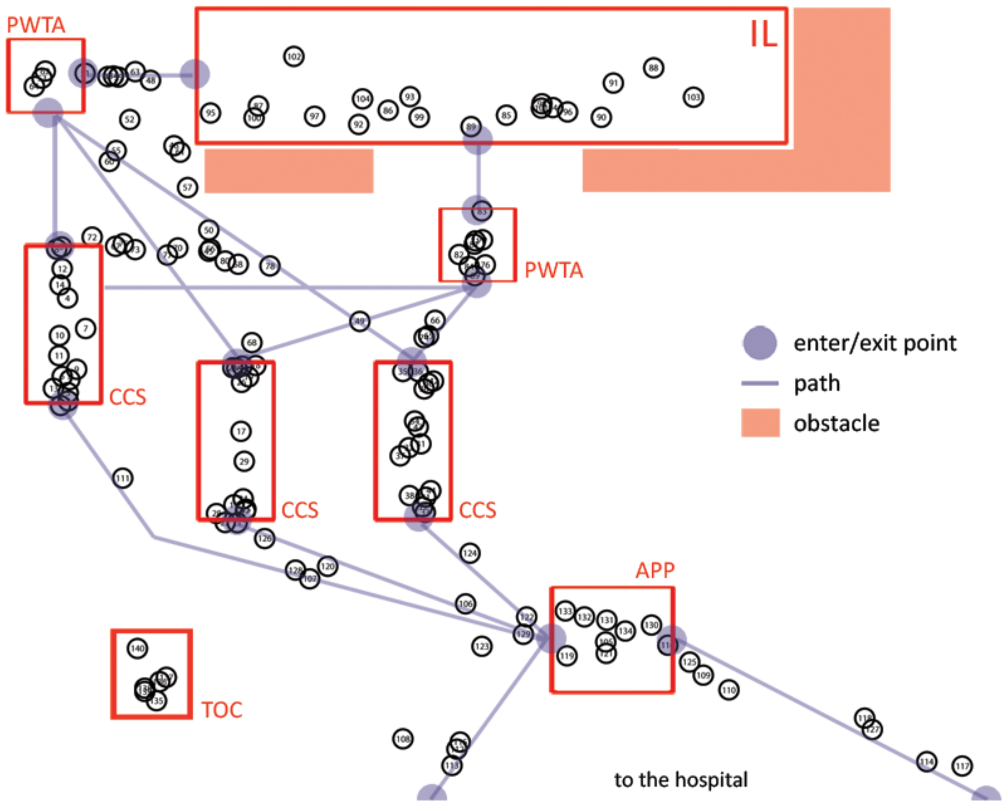


Figure 5.14: Train crash rescue operation sketch

In total there are 141 units, remember that more people is participating in this rescue operation, but only the ones with wireless devices are considered. The speed ranges for pedestrians and vehicles are 1-2m/s and 5-12m/s respectively and it depends on the zone they are working in (it was already explained in section 3.5 that this mobility model provides spatial dependency of velocity). As the area around the disaster is quite flat and without building around, the propagation model considered is the two ray ground explained in the App. A.2.

The radio communication devices use the voice codification based on the algorithm MELPe (Mixed Excitation Linear Prediction [enhanced]) which was developed by the US government as the voice codec Military Standard [44]. Later on, it was implemented in lots of equipment for tactical and rescuing operation due to its high voice quality using only 2.4kbps. The radio channels used are in the frequency range of 146-174 MHz giving a transmission range of 100 meters in this kind of scenario. The average height of the antennas knowing where the vehicular and pedestrian units have it, is 1.6 meters.

The connections follow a pareto distribution which is usually applied to model self-similar arrival in packet traffic. As each connection represent a call, an exponential variable with an average of 20 seconds is used to decide its duration. Usually the calls will be used to inform quickly and to give or receive instructions, so they will not be longer.

The parameter swept along the simulations is the number of connections. Using the pareto distribution for the calls arrival to model situations with more or less stress is not enough to create a realistic stressful situation because, if the are not enough connections, a possible network congestion will not be seen. Increasing the number of connection contributes to emphasize this effect and see if a protocol can deal with a critical situation where lots of units have to contact with each other at the same time.

As this is a rescue operation completely organized and coordinated, even in the worse case of an emergency, the number of calls in our 5 minutes simulation (time enough to represents the duration of the critical part of the communications when an emergency occurs) will not be more than 80-90. In order to see which safety margin each protocol provides, they are tested from 20 up to 120 connections.

In App. C.1 you can check the TCL script used to describe this simulating environment in one of the scenario of this case study, and in App. D how the mobility pattern can be generated. In the disaster area model generated by Bonnmotion the transitory period lasts around 5000 seconds [30], so at least the first 5000 seconds of the simulation have to be skipped to get truthful results.

Model statistics

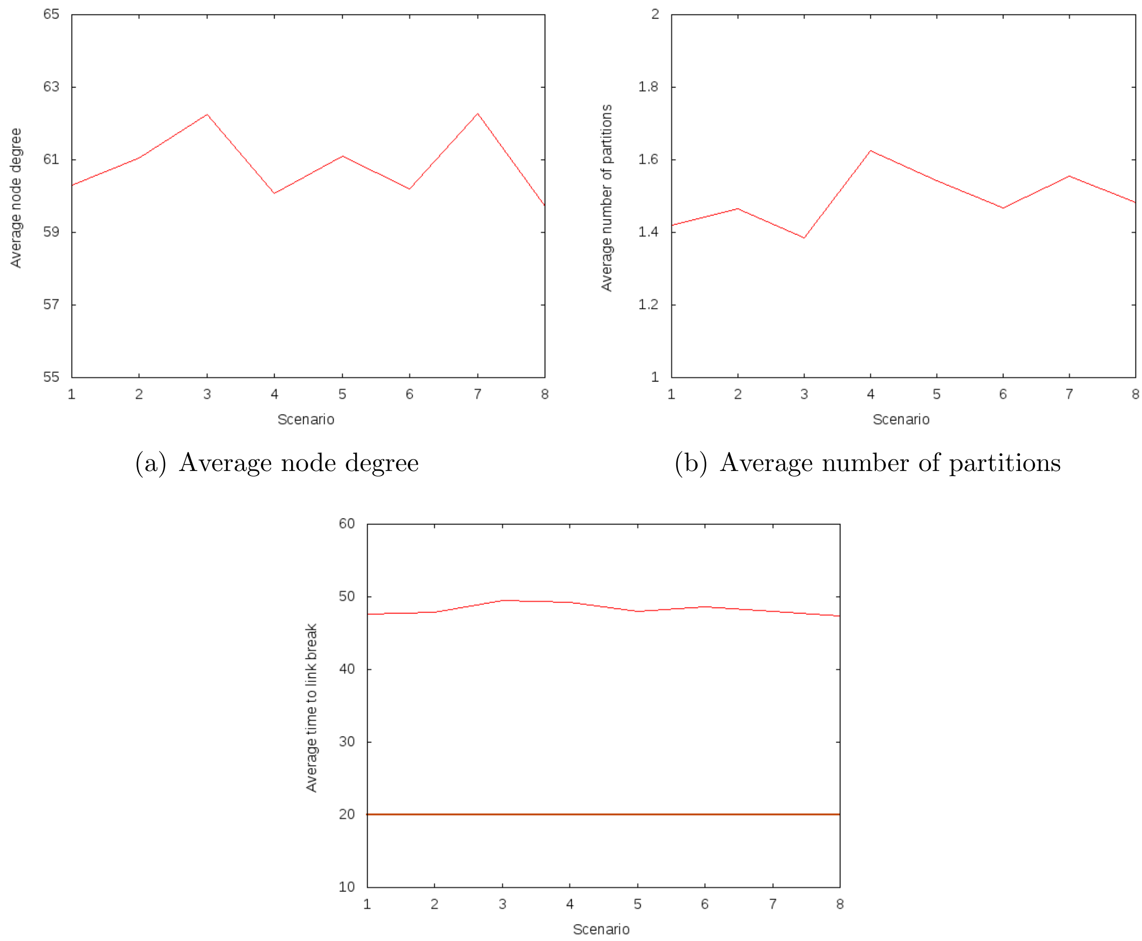
Referring the explanation of the scenarios' statistics in the previous case studies (subsection 5.1.1), the average node degree and the average number of partition are studied. One of the main differences between this case study and the others is the fact that here a real time communication is needed. Thus, another scenario statistic useful to see if this scenario can handle the type of communication required is the average time to link break. This aspect is not pointed out in the two previous case studies because the communication required is really short (only few seconds of duration) and the link break time in their situations is always much more.

The results for these three statistics in different scenarios is presented in figure 5.15. Since the model created represents an specific rescue operations, the position of the areas as well as their number is not changed. Slightly few differences in the number of pedestrians or vehicles has been done in each scenario, which could represent several situations in the same rescue operation.

As the nodes' movements between region is quickly and continuously, the average number of partitions is not so close to 1 as it occurred in the other case studies, but around 1.5. This means that, more or less, the half of the time the network is not completely connected and subnetworks appear.

Although the movement between regions is fast, the movement inside the regions is more stable and slower and this is the reason why the average node degree nearly does not change while changing the scenario parameters. Despite the presence of obstacles, as the region is small in comparison with the number of nodes and transmission range, during the whole time almost the half of the nodes can be reached by another one (completely opposite situation as in 5.2). Thus, it will not be necessary for the protocols to discover and maintain routes of lots of hops.

In the other hand, the average time to link break is close to 50 seconds, so more than twice the average time of the durations of the calls. Although an exponential random variable can reach values much different than its average value, without considering the properties of the protocols themselves, this scenario seams to be able to handle the communication's requirements.



(a) Average node degree

(b) Average number of partitions

(c) Average time to link break

Figure 5.15: Rescue operation statistics vs. different scenarios

5.3.2 Protocols and figures of merit

The protocols tested are AODV, DSDV, DSR and LAR (with its two different methods). AODV and DSR are chosen to be the most widely reactive routing protocols, which perform well under the scenario statistics explained before (can deal with network partitions, no need to discover and maintain long paths, etc.).

This scenario has a considerable number of nodes not distributed in the whole region, but in sections, and changing quickly between them. This is what makes the average node degree stable, but as it is stable only in small regions can cause DSDV to not reach its optimal performance. To check this point, it will be evaluated. For this same reason, OLSR which mainly performs good under more stable conditions is not tested.

In this case study, the mobility change a lot and is fast in the case of the vehicles creating network partitions as shown in the scenario statistics. Thus, TORA is not evaluated because as is proved in [8], [40] and in the first case study of this research (5.1), it can not handle such circumstances of speed change and huge connectivity. In table 5.3 the protocols constants chosen are listed.

Table 5.3: Rescue operation protocols parameters

(a) DSR

Timehop for hop route request	30 ms
Retransmit route request	500 ms
Size of header with n addresses	4n + 4 bytes
Buffer size	50 packets
Packet lifetime in buffer	30 s
Max rates for routes replies	1/s

(b) AODV

Route Request retry	5
Route Request rate limits	10 pkts/s
Gratuitous route reply flag	Enable
Advice route timeout	30 s
Hello interval	1 s
Allowed hello loss	10 s
Timeout buffer	2 s

(c) DSDV

Startup jitter	2 s
Instantaneous jitter	100 ms
Broadcast jitter	10 ms
Min. triggered updates	1 s

(d) LAR (both methods)

Timehop for hop route request	30 ms
Retransmit route request	500 ms
Size of header with n addresses	4n + 40 bytes
Buffer size	50 packets
Packet lifetime in buffer	30 s

First, the figures of merit considered are the average packet delivery fraction, the average throughput and the average end-to-end delay. From these results, the protocols that clearly do not accomplish the requirements are ruled out and then, from the chosen ones, the instant packet delivery variation (PDV) is checked because, as it is explained and exemplified afterwards with the results, this is a really important point in real time communications.

5.3.3 Results

Sweeping the number of connections

As is explained in the previous section, first the results for different number of connections are presented in figure 5.16.

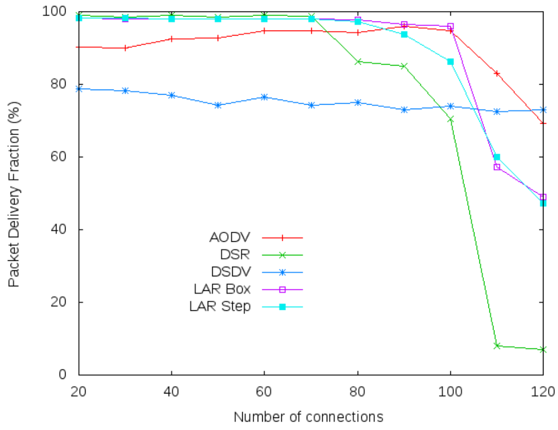
In terms of PDF and delay, and so also throughput, from 70-80 connections onwards DSR starts to fail considerably presenting less than 85% of PDF at 80 connections, much less than LAR and AODV and also a throughput of 2kbps. This is for the same reason as happened in the other case study in section 5.1: as DSR is a source routing protocol, as the amount of connections increase, the more the big overhead of the packets increases and create network congestion. This is also reflected in the routing load, where from 70-80 connections onwards it drastically rises, also because of the packets retransmission when their are lost due to the network congestion. So DSR performs good almost in all the region of interest (till 80-90 connections) but at the end it can not handle it which gives us no safety margin at all.

As LAR is based in DSR, it presents the same problem but thanks to the location information it can handle more connections before reaching the breaking point and it starts to fail from 100-110 connection onwards. Thus, inside our region of interest, it performs really good offering more than 95% of PDF and an average delay around 0,15 seconds. Between the two LAR methods, LAR Box always performs better than LAR Step (see subsection 2.2.6), so the first one is chosen to continue its test in the communication instant values.

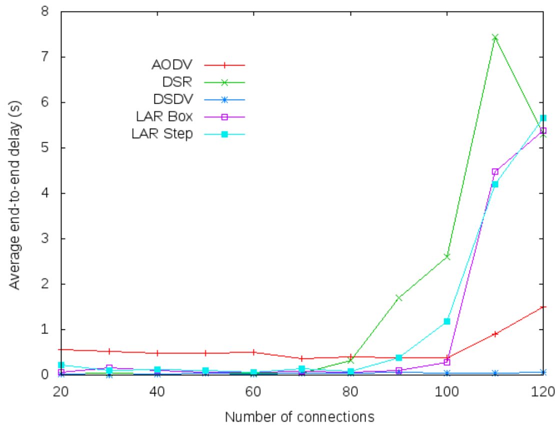
In this scenario the nodes in each area move slowly but between areas they can achieve high speeds, so we can say that not the whole region is stable, but locally yes. Because of this, DSDV maintain its performance independently of the number of connection with small delay thanks to its proactive nature (important point since this is a real time scenario) and routing load decreasing (more connections means more information that data packets can carry themselves without the need of sending packets only for routing). But as our region of interest ends at 80-90 connections, it is not considered to continue because in this region other protocols perform much better in terms of

PDF and throughput. DSDV cannot achieve more than 78% of PDF and 1.9kbps of throughput while LAR and AODV get always more than 90% and 2.2kbps respectively.

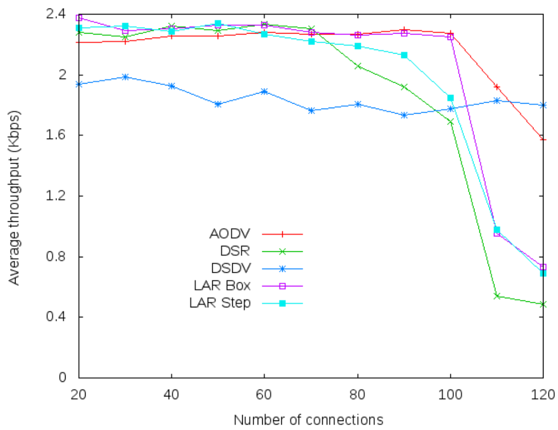
As I just asserted, AODV also performs good and, although is not as quick as the source routing ones because it uses a distance-vector algorithm, it is the most stable at the end of our region of interest (it does not fail so drastically as DSR or LAR) providing a wider safety margin. As stated before, the delay is important in this real time communication, but as the average is not so bad (less than 0,5 seconds) and the other figures of merits give good results, AODV is also taken into account.



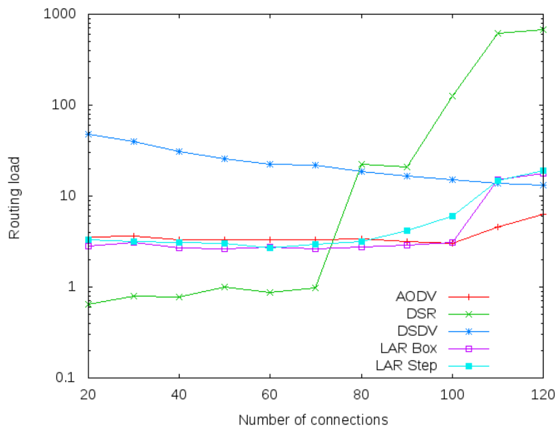
(a) Packet delivery fraction



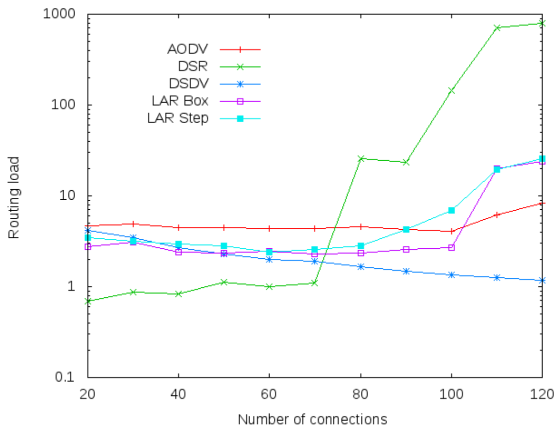
(b) End-to-end delay



(c) Throughput



(d) Routing load (bytes)



(e) Routing load (packets)

Figure 5.16: Rescue operation figures of merit vs. number of connections

Communication instant values

With these first results I decided to continue to discuss which is the best protocol to use between AODV and LAR Box, since the other ones have been proved not suitable. I want to remember that this is a real time communication so the instant delay and PDV are key points. AODV average delay is quite higher than LAR Box, but we have to check the instant delay during a whole communication, since with the average value is not possible to say which one is better.

The PDV is useful to check the health of a call in order to see if the communication between nodes can flow smoothly without a huge change in the delay along the communication (which will make it ineffective). If the communication delay of a protocol is a bit higher than another one but its value is more constant value, this is preferable in front of a one with less delay but changing a lot during the connection.

According to the explanations and results presented in [45] we can say that in a optimum real time communication, the PDV should be less than 0.15 seconds. Between 0.15 and 0.3 seconds is considered as average and from 0.3 to 0.5 as poor real time communication. Delays of more seconds are tagged as unacceptable delays for real time communications in these emergency situations.

In figure 5.19 you can compare the instant PDV and delay of a simulation with 90 nodes with a pareto arrival distribution (which is the most stressful situation) and in figure 5.17 is shown which percentage of the communication is optimum, average, poor or unacceptable.

The results show that, in these figures of merit, LAR Box performs much better than AODV. Although AODV presents a good service during the 87.3% of the connection, a 11.02% of the rest if considered unacceptable because the packets arrive with more than 0.5 seconds of PDV. In the case of LAR Box, the 97.01% of the packets present a PDV below 0.15 seconds and the major part of the rest, less than 0.3 seconds. Really good result for a MANET network taking into account that these results are from the most stressful situation.

Checking a more relaxing situation (50 connections in a exponential arrival distribution) we see in figure 5.18 and figure 5.20 that the performance of LAR Box is a bit

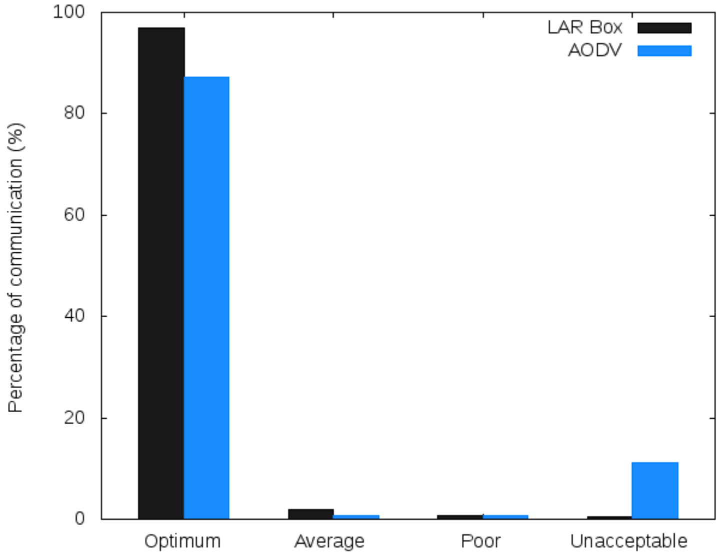


Figure 5.17: Communication type in a stressful situation

better than before: 98.31% of package with PDV below 0.15 seconds and only an irrelevant part of 0.17% of the communications is considered as unacceptable. This is because the great amount of connections concentrated in a short period of time (which before was generated by the pareto arrival) is an aspect that makes source routing protocols (LAR) fail quicker, and now with an exponential distribution is gone.

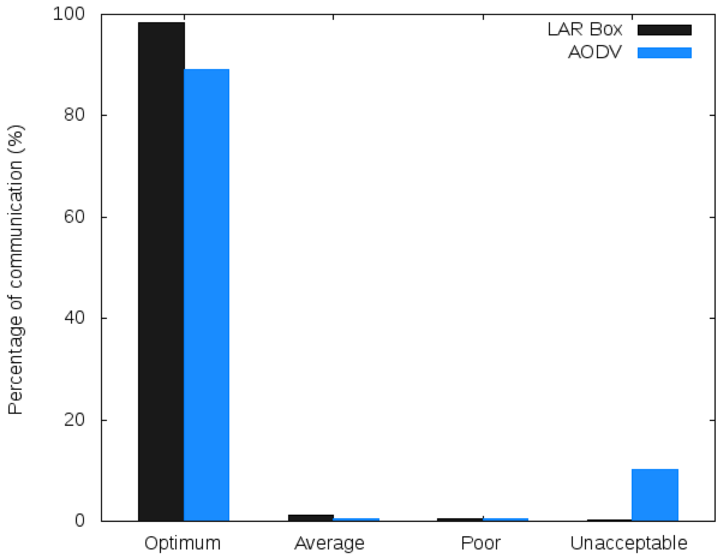


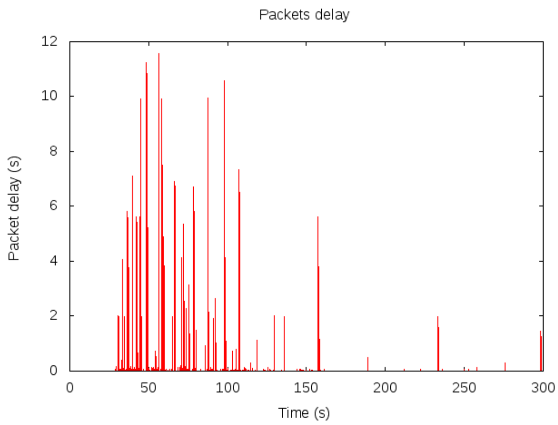
Figure 5.18: Communication type in a more relaxing situation

In the other hand, AODV still have unacceptable delays: 10.13% of packets still with more than 0.5 second of PDV. We see that the big delays of AODV do not seam to be much related with the arrival distribution, but an effect of the distance-vector routing itself which along the whole communications it turns it not poor, but unacceptable. This routing algorithm (as is explained in 2.1.1 and 2.2.3) a part from generating more end-to-end delay, it specific short intervals can cause a considerably large delay.

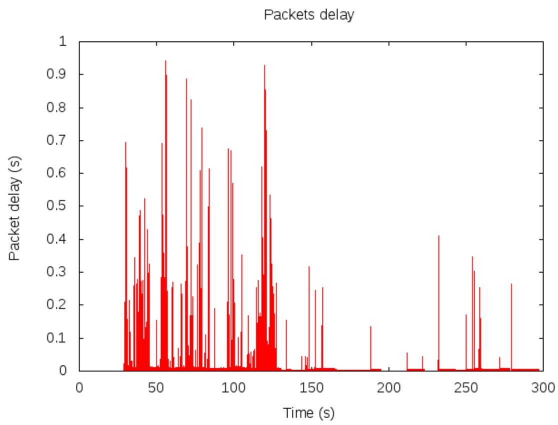
In table 5.4 the exact percentage of communication type in both scenarios is shown. Thus, we can conclude that the most suitable protocol to manage the communication in a MANET system for a rescue operation is LAR in its Box working method. In this case, as LAR Box outperforms so much AODV in terms of delay, checking the PDV was not completely necessary. But, for example, if we had consider DSR as an option, as both are source routing protocols, the PDV would have been the figure of merit that would have decided which one is the best.

Table 5.4: Percentage of communication type

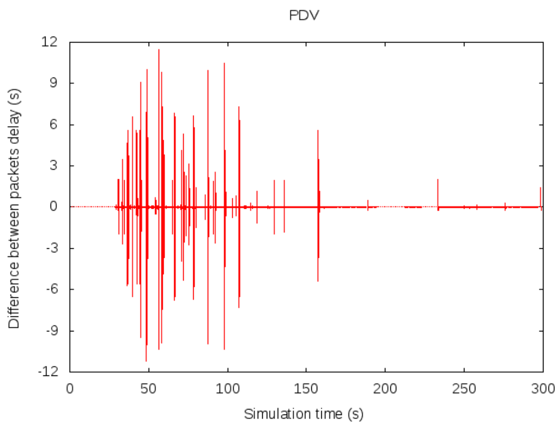
	Stressful		Relaxing	
	AODV	LAR Box	AODV	LAR Box
Optimum (%)	87.3	97.01	89.01	98.31
Average (%)	0.76	1.98	0.39	1.19
Poor (%)	0.65	0.62	0.47	0.36
Unacceptable (%)	11.02	0.39	10.13	0.17



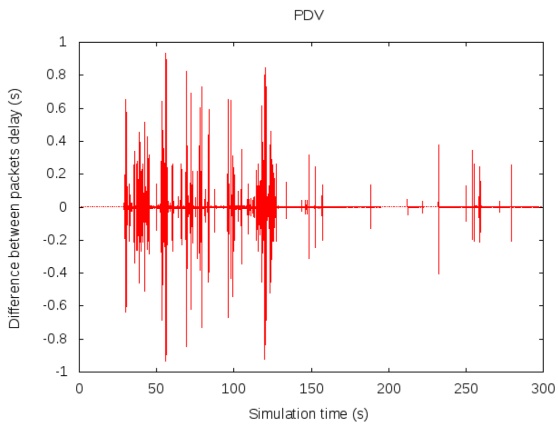
(a) AODV delay



(b) LAR Box delay

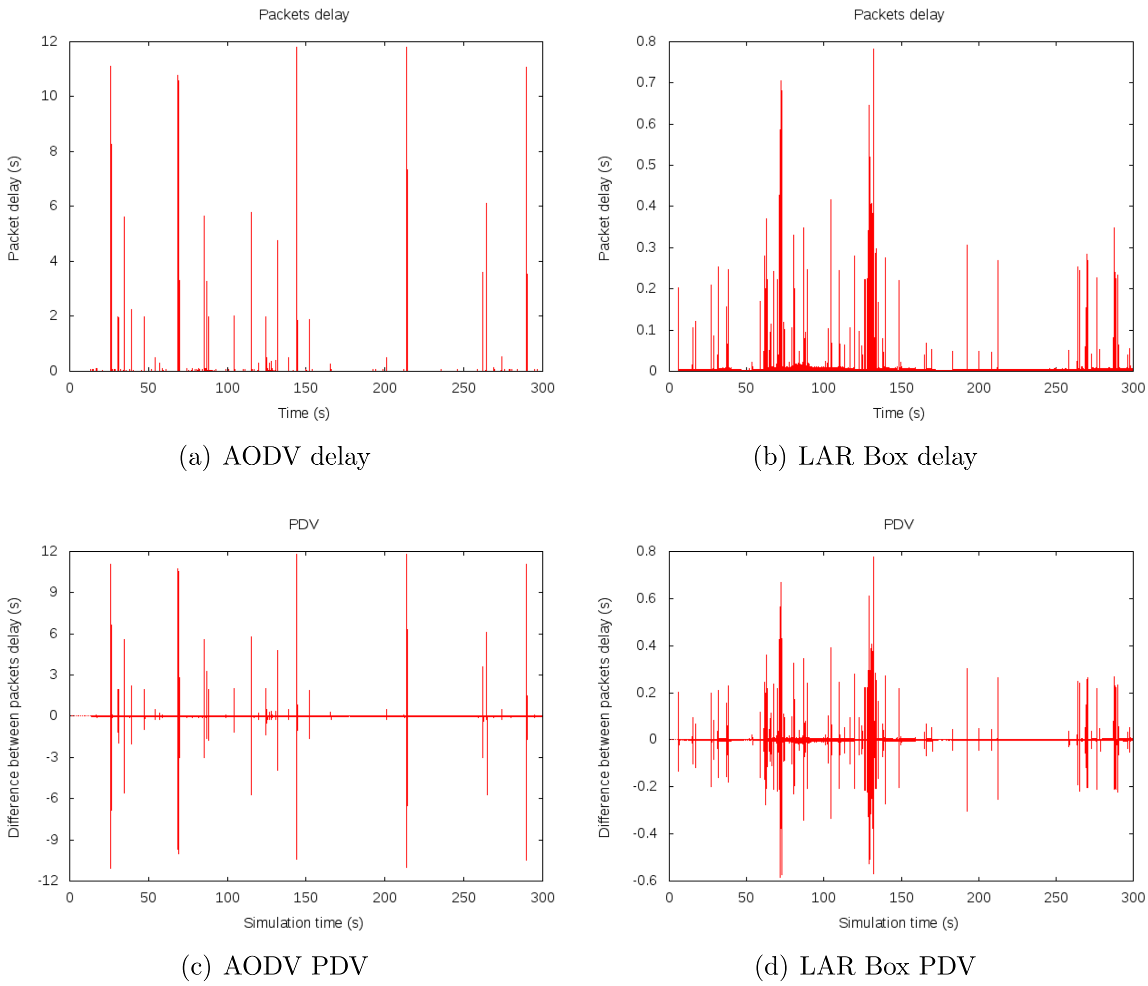


(c) AODV PDV



(d) LAR Box PDV

Figure 5.19: Instant PDV and delay in a stressful situation



(a) AODV delay

(b) LAR Box delay

(c) AODV PDV

(d) LAR Box PDV

Figure 5.20: Instant PDV and delay in a more relaxing situation

Chapter 6

Conclusions

Three case studies were analyzed and discussed in this research, each one representing a real-life application that could be created with a MANET. The aim of this research was to model a realistic environment for each case study (using the available open-source tools for simulating mobile ad-hoc networks) and evaluate the performance of different routing protocols under these conditions. This is a step forward since lots of publications still compare protocols performance in ideal assumptions which, as has been proved in this research as well as in [13] or [15], can lead to completely different results as if the MANET were modeled according to a real scenario.

The first case study (section 5.1) was an information service created with people's personal devices and shops inside a shopping mall. The main points a routing protocol had to face were:

1. High density of nodes (dense MANET).
2. Huge amount of short connections.
3. Stable scenario due to the slow speed of the nodes in a small area.

DSDV was the protocol performing the best due to its better scalability properties in terms of nodes and connections under the stability characteristics of this MANET.

AODV seamed a suitable option since is developed for working in networks with high node density, but in the worse scenario the number of connection were too much for using the distance-vector algorithm in a reactive way. OLSR failed to perform because as a link-state protocol, when only some few nodes send information (the shops) its way to maintain the routes can not be done properly.

Furthermore, it was proved in 5.1.3 that for a protocol is more challenging to handle lots of new connections than to maintain existing ones even if the amount of data to transfer is the same. Protocols using link-reversal algorithms like TORA or source routing algorithms like DSR can route well lots of information, but what they can not handle is when the information is split in lots of different connections.

An VANET in an urban area was the second case study (section 5.2). It was checked in 5.2.1 that its main characteristics were:

1. Nodes move at high speed.
2. Lots of hops are needed to reach the destination.
3. Network partitions may appear when the density of nodes is low.

The best performance was given by AODV when the density of nodes was not so high, and by LAR Box when the density increases. When the mobility was high, DSR was not able to work well due to the need of discovering long routes in short time, but the geographical information used by LAR helped it to achieve a good performance even being also a source routing protocol. In the other hand, TORA again could not give good results, this time because the link-reversal algorithm was not implemented to be used with so many nodes.

The third case study 5.3 consisted of a rescue operation communication system. After the study of the model's statistics, the main properties obtained were:

1. Real time communication needed (small end-to-end delay and PDV).
2. Locally stable in the different regions but fast and constant moves between them.

3. Few hops needed to reach the destination, but network partition can appear anytime.

The only protocol able to manage this communication system was LAR Box. Its delay and PDV were close to the one of a proactive protocol thanks to the geographical information combined with its source routing nature. The distance-vector algorithm used by AODV made impossible to handle real time communication because the delays generated were too big (although this protocol could be suitable if the communication between nodes did not require real time properties). The stable condition of each zone was the characteristic needed by DSDV to perform good, but as between zones this behavior was completely the opposite, it could not reach its optimal performance. Although in terms of delay and PDV was suitable for real time communication, DSDV did not present enough throughput.

Gathering all the results and conclusions of each case study, the following general situation where some protocols should be used or not can be asserted:

- DSDV gives good results in terms of PDF, delay and routing load only if the scenario evaluated is globally stable.
- DSR and TORA are not good candidates when an application requires a great amount of connections (compared with the number of nodes). LAR may also have problems dealing with it, but it can afford more than DSR and TORA.
- AODV presents good performance in terms of PDF in large networks and in different situations, but it generates to much delay to be used in real time applications.
- OLSR can fail to perform if only a small part of nodes in a network send information and the rest only receive.
- If geographical information is available, LAR is a suitable option in large networks as well as in real time applications since it provides high PDF and few end-to-end delay and PDV, so also obtaining high throughput.

As future work, more detailed simulations could be performed in each one of these case studies considering more aspect to get closer to a more realistic environment. The main problem is that with all standard open source tools may be impossible to achieve more accurate scenarios modeling each application and its environment, so specific modules or other tools should be developed.

Furthermore, since today we have the technology available for it, more researches should be done to find out in which other scenarios MANETs could be implemented. It was also pointed out that the performance of the protocols can be considerably different when more and more realistic elements are taken into account. This aspect have to be considered in further researches since the nowadays evolution of MANET will bring them soon into services of our society.

Appendix A

Propagation models

A.1 Free space model

The free space propagation model assumes the ideal propagation condition that there is only one clear line-of-sight path between the transmitter and receiver. H. T. Friis presented the following equation to calculate the received signal power in free space at distance d from the transmitter [46].

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \tag{A.1}$$

where P_t is the transmitted signal power. G_t and G_r are the antenna gains of the transmitter and the receiver respectively. $L(L \geq 1)$ is the system loss, and λ is the wavelength.

The free space model basically represents the communication range as a circle around the transmitter. If a receiver is within the circle, it receives all packets, otherwise, it loses all.

A.2 Two-ray ground reflection model

A single line-of-sight path between two mobile nodes is seldom the only means of propation. The two-ray ground reflection model considers both the direct path and a ground reflection path. It is shown [47] that this model gives more accurate prediction at a long distance than the free space model. The received power at distance d is predicted by

$$P_r(d) = \frac{P_t G_t G_r {h_t}^2 {h_r}^2}{d^4 L} \tag{A.2}$$

where h_t and h_r are the heights of the transmit and receive antennas respectively. Note that the original equation in [47] assumes $L = 1$. To be consistent with the free space model, L is added here.

The above equation shows a faster power loss than Eqn. (A.1) as distance increases. However, The two-ray model does not give a good result for a short distance due to the oscillation caused by the constructive and destructive combination of the two rays. Instead, the free space model is still used when d is small.

Therefore, a cross-over distance d_c is calculated in this model. When $d < d_c$, Eqn. (A.1) is used. When $d > d_c$, Eqn. (A.2) is used. At the cross-over distance, Eqns. (A.1) and (A.2) give the same result. So d_c can be calculated as

$$d_c = (4\pi h_t h_r)/\lambda \tag{A.3}$$

A.3 Shadowing model

The free space and the two-ray model predict the received power as a deterministic function of distance. They both represent the communication range as an ideal circle. In reality, the received power at certain distance is a random variable due to multipath propagation effects, which is also known as fading effects. In fact, the above two models predicts the mean received power at distance d.

The shadowing model consists of two parts. The first one is known as path loss model, which also predicts the mean received power at distance d, denoted by $\overline{P_r(d)}$. It uses a close-in distance d_0 as a reference. $\overline{P_r(d)}$ is computed relative to $P_r(d_0)$ as follows.

$$\frac{P_r(d_0)}{\overline{P_r(d)}} = \left(\frac{d}{d_0}\right)^{\beta} \tag{A.4}$$

β is called the path loss exponent, and is usually empirically determined by field measurement. From Eqn. (A.1) we know that $\beta = 2$ for free space propagation. Table A.1 gives some typical values of β. Larger values correspond to more obstructions and hence faster decrease in average received power as distance becomes larger. $P_r(d_0)$ can be computed from Eqn. (A.1).

Table A.1: Some typical values of path loss exponent β

Environment		β
Outdoor	Free space	2
	Shadowed urban area	2.7 to 5
In building	Line-of-sight	1.6 to 1.8
	Obstructed	4 to 6

Table A.2: Some typical values of shadowing deviation σ_{dB}

Environment	σ_{dB} (dB)
Outdoor	4 to 12
Office, hard partition	7
Office, soft partition	9.6
Factory, line-of-sight	3 to 6
Factory, obstructed	6.8

The path loss is usually measured in dB. So from Eqn. (A.4) we have

$$\left[\frac{\overline{P_r(d)}}{P_r(d_0)}\right]_{dB} = -10\beta \log\left(\frac{d}{d_0}\right) \tag{A.5}$$

The second part of the shadowing model reflects the variation of the received power at certain distance. It is a log-normal random variable, that is, it is of Gaussian distribution if measured in dB. The overall shadowing model is represented by

$$\left[\frac{P_r(d)}{P_r(d_0)}\right]_{dB} = -10\beta \log\left(\frac{d}{d_0}\right) + X_{dB} \tag{A.6}$$

where X_{dB} is a Gaussian random variable with zero mean and standard deviation σ_{dB}. σ_{dB} is called the shadowing deviation, and is also obtained by measurement. Table A.2 shows some typical values of σ_{dB}. Eqn. (A.6) is also known as a log-normal shadowing model.

The shadowing model extends the ideal circle model to a richer statistic model: nodes can only probabilistically communicate when near the edge of the communication range. As Shadowing model simulates shadow effect of obstructions between the transmitter and receiver, it is mainly used to simulate wireless channel in in-door environment.

Appendix B

Main AWK script

```
BEGIN {
   recvd_size = 0
   sent_pkts = 0
   recvd_pkts = 0
   prot_pkts = 0
   prot_size = 0
   sent_size = 0
   #must change the packet size in each case
   packet = 512
   numcon = 0
   all_times = 0
  }

function abs(value) {
   if(value < 0) return -value
   else return value
}

  {
   event = $1
   time = $2
   node_id = $3
   level = $4
   event_id = $6
   pkt_type = $7
   pkt_size = $8

  #### Data packets sent

  if (event == "s" && level == "AGT") {
   # Update total sent packets and the starting time of
```

```
 # the event to calculate its delay
 sent_pkts++
 sent_size += pkt_size
 sent_times[event_id+1]=time

 # Register of all the active connection in the simulation
 exist=0
 sub(/\[/,"",$14)
 split($14, src, ":")
 split($15, dst, ":")

 # Check if there is a new connection between to nodes
 for (i=0;i<=numcon;i++) {
    # this connection is already stored
    if(src[1]==conSour[i+1] && dst[1]==conDest[i+1]){
       exist=1
       break
    }
 }

 # This connection is new, store its values (src, dst, start)
 # and update the number of connection
 if (exist==0){
    numcon++
    conSour[numcon]=src[1]
    conDest[numcon]=dst[1]
    conTimeStart[numcon]=time
 }
}

#### Data packets received

else if (event == "r" && level == "AGT") {
 # Every time a packet is received, the final time of the
 # connection is updated by this time
 sub(/\[/,"",$14)
 split($14, src, ":")
 split($15, dst, ":")

 for (i=0;i<=numcon;i++) {
    if(src[1]==conSour[i+1] && dst[1]==conDest[i+1]){
       conTimeStop[i+1]=time
       break
       }
 }
```

```
  # Store the stop time of the event
  recv_times[event_id+1]=time
  last_event=event_id+1
  # Rip off the header
  hdr_size = pkt_size % packet
  pkt_size -= hdr_size
  # Store received packet's size and update total received packets
  recvd_size += pkt_size
  recvd_pkts++
 }

 #### Protocol packets

 else if ((event == "s" || (event == "f")) && (pkt_type == "AODV" ||
 pkt_type == "DSR" || pkt_type == "message" || pkt_type == "OLSR" ||
 pkt_type == "IMEP")) {

  prot_pkts++
  prot_size+=pkt_size
  prot = pkt_type
 }
}

 END {

  # To plot the times of start and stop of different connections found
  for (i=1;i<=numcon;i++) {
  printf("\n%d) Traffic from %d to %d:\tStart Time = %.3f\tStop Time = %.3f",
  i, conSour[i],conDest[i], conTimeStart[i], conTimeStop[i])
     if(conTimeStop[i]!=0){
        all_times += conTimeStop[i]-conTimeStart[i]
     }
  }
  # Max delay difference = max{d(1),d(2),...d(n)} - min{d(1),d(2),...d(n)}
  maxD = 0
  minD = 1000
  PDV[1]=0
  for (i=1;i<=last_event;i++) {
     #If a packet is dropped the value of its received time is 0,
     # and it must not be counted
     if(recv_times[i]!=0){
        delay[i]=recv_times[i]-sent_times[i]
        if(maxD<=delay[i]) maxD=delay[i]
        if(minD>=delay[i]) minD=delay[i]
        sum_delay+=delay[i]
        # jitter ((recvtime(j)-sendtime(j))-(recvtime(i)-sendtime(i)))/(j-i)
        # j > i (seq number because of the lost packages)
```

```
      if(i>1){
         PDV[i]=(delay[i]-delay[before])/(i-before)
         sum_pdv+=abs(PDV[i-1])
      }
      before=i
   }
   print recv_times[i] "\t" delay[i] > "packets_delay_unsorted.txt"
   print recv_times[i] "\t" PDV[i] > "PDV_unsorted.txt"
   # later on this must be sorted to present accurate results
}

if (recvd_pkts == 0){
   printf("\n\n\t### No data transfer could be stablished ###\n\n")
}

else{

# Throughput: bytesAllConnections/AlltimeConnection (Kbites/s)
avg_trh = recvd_size/all_times/1000*8
pack_del = recvd_pkts/sent_pkts
avg_delay = sum_delay/recvd_pkts
avg_pdv = sum_pdv/(recvd_pkts-1)
highest_delay = (maxD-minD)

if (prot == "message") {prot = "DSDV"}
else if (prot == "IMEP") {prot = "TORA"}

if (prot == "AODV"){
print avg_delay >> "0avg_delay_AODV.txt"
print pack_del*100 >> "0pack_del_AODV.txt"
print prot_size/recvd_size >> "0rout_load_AODV.txt"
print prot_pkts/recvd_pkts >> "0rout_load_pack_AODV.txt"
print avg_trh >> "0avg_thr_AODV.txt"
}

else if (prot == "DSDV"){
print avg_delay >> "0avg_delay_DSDV.txt"
print pack_del*100 >> "0pack_del_DSDV.txt"
print prot_size/recvd_size >> "0rout_load_DSDV.txt"
print prot_pkts/recvd_pkts >> "0rout_load_pack_DSDV.txt"
print avg_trh >> "0avg_thr_DSDV.txt"
}

else if (prot == "DSR"){
print avg_delay >> "0avg_delay_DSR.txt"
print pack_del*100 >> "0pack_del_DSR.txt"
print prot_size/recvd_size >> "0rout_load_DSR.txt"
```

```
    print prot_pkts/recvd_pkts >> "0rout_load_pack_DSR.txt"
    print avg_trh >> "0avg_thr_DSR.txt"
    }

    else if (prot == "TORA"){
    print avg_delay >> "0avg_delay_TORA.txt"
    print pack_del*100 >> "0pack_del_TORA.txt"
    print prot_size/recvd_size >> "0rout_load_TORA.txt"
    print prot_pkts/recvd_pkts >> "0rout_load_pack_TORA.txt"
    print avg_trh >> "0avg_thr_TORA.txt"
    }

    else if (prot == "OLSR"){
    print avg_delay >> "0avg_delay_OLSR.txt"
    print pack_del*100 >> "0pack_del_OLSR.txt"
    print prot_size/recvd_size >> "0rout_load_OLSR.txt"
    print prot_pkts/recvd_pkts >> "0rout_load_pack_OLSR.txt"
    print avg_trh >> "0avg_thr_OLSR.txt"
    }

    printf("\n\nSent packages = %d",sent_pkts)
    printf("\nReceived packages = %d",recvd_pkts)
    printf("\nPDF = %.3f",pack_del*100)

    printf("\n\nAll delays[s] = %.3f", sum_delay)
    printf("\nAverage delay[s] = %.3f",avg_delay)
    printf("\nHighest delay difference[ms] = %.3f",highest_delay)

    printf("\n\nAll connetions time[s] = %.3f", all_times)
    printf("\nAverage Throughput[kbps] = %.3f",avg_trh)

    printf("\n\nAverage PDV[ms] = %.3f",avg_pdv)

    printf("\n\nProtocol = %s\n",prot)
    printf("\nProtocol packets = %d\n",prot_pkts)
    printf("\nProtocol load[pkts] = %.3f",prot_pkts/recvd_pkts)
    printf("\nProtocol load[bytes] = %.3f\n\n",prot_size/recvd_size)
    }
}
```

Appendix C

TCL scripts

C.1 Shopping mall

```
### Define options ###

set opt(chan)   Channel/WirelessChannel;   # channel type
set opt(prop)   Propagation/Shadowing;      # radio-propagation model
set opt(netif)   Phy/WirelessPhy;      # network interface type
set opt(mac)   Mac/802_11;         # MAC type
set opt(ll)   LL;            # link layer type
set opt(ant)   Antenna/OmniAntenna;      # antenna model
set opt(ifqlen)   64;            # max packet in ifq
set opt(rp)   AODV;    #AODV, DSDV, DSR, TORA, OLSR routing protocol
set opt(sc)    "./shopping_scenario1.ns_movements"
set opt(nnsh)   24;   #number of shops


#### Define de transmition range ####

$opt(prop) set pathlossExp_ 3.0; #path loss exponent
$opt(prop) set std_db_ 5.0; #shadowing deviation
$opt(prop) set dist0_ 1.0

$opt(netif) set L_ 1.0
$opt(netif) set freq_ 2.472e+09      ;#channel 13
$opt(netif) set Pt_ 0.281838   ;#transmission power (W)
$opt(netif) set RXThresh_ 6.18186e-11   ;#for 40m Shadowing
```

```
#### Load the scenario parameters ####

# grid size, number of nodes and simulation time
source shopping_scenario1.ns_params

#### Load connection properties ####

set opt(nc) [expr $opt(nn)*3.6] ;#number of connections
set packSize 256   ;#in bytes
set min_duration 3
set max_duration 6
set rateC 32kbps   ;#bit rate

set val(nn) $opt(nn)         ;# number of clients
set opt(nn) [expr $val(nn)+$opt(nnsh)]   ;# number of clients + shops
puts "Number of connections: $opt(nc)"

# to avoid DSR segmentation fault
if { $opt(rp) == "DSR" } {
   set opt(ifq) CMUPriQueue;
} else {
   set opt(ifq) Queue/DropTail/PriQueue;
}

set ns_ [new Simulator]Script
$ns_ use-scheduler Heap
set tracefd [open randomtrace.tr w]
#set namtrace [open randomnam.nam w]
set chan [new $opt(chan)]
set prop [new $opt(prop)]

$ns_ trace-all $tracefd
#$ns_ namtrace-all-wireless $namtrace $opt(x) $opt(y)

# set up topography object
set topo [new Topography]
$topo load_flatgrid $opt(x) $opt(y)
create-god $opt(nn)
$prop topography $topo


#### Create nn mobilenodes and attach them to the channel ####

$ns_ node-config -adhocRouting $opt(rp) \
        -llType $opt(ll) \
        -macType $opt(mac) \
```

```
        -ifqType $opt(ifq) \
        -ifqLen $opt(ifqlen) \
        -antType $opt(ant) \
        -propType $opt(prop) \
        -phyType $opt(netif) \
        -channel [new $opt(chan)] \
        -topoInstance $topo \
        -agentTrace ON \
        -routerTrace $routeT \
        -macTrace OFF \
        -movementTrace OFF

# create clients and shops with diferent names to make it easier
for {set i 0} {$i < $val(nn) } { incr i } {
   set node_($i) [$ns_ node]
   $node_($i) random-motion 0
   $ns_ initial_node_pos $node_($i) 4
}
for {set i 0} {$i < $opt(nnsh) } { incr i } {
   set shop_($i) [$ns_ node]
   $shop_($i) random-motion 0
   $ns_ initial_node_pos $shop_($i) 10
}


#### Set up the connection between two random nodes ####

set rngNumber [new RNG]
$rngNumber seed 157

#create UDP connection between two random nodes with CBR traffic
for {set i 0} {$i < $opt(nc)} {incr i} {
  set same 1
  #to not create more than once a connection between the same
  #shop and the same client
  while {$same == 1} {
   set same 0
   set num_udpG_($i) [expr [$rngNumber integer $opt(nnsh)]]
   set num_udpS_($i) [expr [$rngNumber integer $val(nn)]]
   for {set j 0} {$j < $i} {incr j} {
      if{$num_udpG_($i)==$num_udpG_($j) && $num_udpS_($i)==$num_udpS_($j)}{
         set same 1
         break
      }
    }
  }
```

```
set udp_($i) [new Agent/UDP]
$ns_ attach-agent $shop_($num_udpG_($i)) $udp_($i)

set null_($i) [new Agent/Null]
$ns_ attach-agent $node_($num_udpS_($i)) $null_($i)

$ns_ connect $udp_($i) $null_($i)

set cbr_($i) [new Application/Traffic/CBR]
$cbr_($i) attach-agent $udp_($i)
$cbr_($i) set packetSize_ $packSize
$cbr_($i) set rate_ $rateC
$cbr_($i) set random_ false

}

#### Set up the traffic pattern between nodes ####

set rngStart [new RNG]
$rngStart seed 311
set RVStart [new RandomVariable/Exponential]
$RVStart use-rng $rngStart
$RVStart set avg_ [expr $opt(stop)/2]

set rngStop [new RNG]
$rngStop seed 99
set RVStop [new RandomVariable/Uniform]
$RVStop use-rng $rngStop

for {set i 0} {$i < $opt(nc)} {incr i} {
   set start_time_($i) [expr [$RVStart value]]
   while {$start_time_($i) > $opt(stop)} {
      set start_time_($i) [expr [$RVStart value]]
   }
   $RVStop set min_ [expr [$start_time_($i)+$min_duration]]
   $RVStop set max_ [expr [$start_time_($i)+$max_duration]]
   set stop_time_($i) [expr [$RVStop value]]
   if { $stop_time_($i) > $opt(stop) } {
      set stop_time_($i) $opt(stop)
   }
   $ns_ at $start_time_($i) "$cbr_($i) start"
   $ns_ at $stop_time_($i) "$cbr_($i) stop"
}

# loading the shops position generated as a random static model
source shops_scenario.position
```

```
puts "Loading scenario file..."
source $opt(sc)


#### Ending nam and the simulation ####

for {set i 0} {$i < $val(nn) } { incr i } {
   $ns_ at $opt(stop) "$node_($i) reset"
}

for {set i 0} {$i < $opt(nnsh) } { incr i } {
   $ns_ at $opt(stop) "$shop_($i) reset"
}

proc stop {} {
    global ns_ tracefd
    $ns_ flush-trace
    close $tracefd
    #close $namtrace
}

#$ns_ at $opt(stop) "$ns_ nam-end-wireless $opt(stop)"
$ns_ at $opt(stop) "stop"
$ns_ at [expr $opt(stop)+0.01] "puts \"end simulation\" ; $ns_ halt"

$ns_ run
```

C.2 Basic vehicular network

```
### Define options ###

set opt(chan)   Channel/WirelessChannel;   # channel type
set opt(prop)   Propagation/Shadowing;      # radio-propagation model
set opt(netif)   Phy/WirelessPhy;      # network interface type
set opt(mac)   Mac/802_11;         # MAC type
set opt(ll)   LL;            # link layer type
set opt(ant)   Antenna/OmniAntenna;      # antenna model
set opt(ifqlen)   50;            # max packet in ifq
set opt(rp)   AODV;             # routing protocol

# mobility file generated by Bonnmotion
set opt(sc)    "./manhattan1.ns_movements";

#### Define de transmition range ####

$opt(ant) set Gt_ 1
$opt(ant) set Gr_ 1
$opt(ant) set Z_ 1.5

$opt(prop) set pathlossExp_ 3.7; #path loss exponent
$opt(prop) set std_db_ 7.0; #shadowing deviation
$opt(prop) set dist0_ 1.0

$opt(netif) set L_ 1.0
$opt(netif) set RXThresh_ 1.02625e-14   ;#200m
$opt(netif) set Pt_ 0.5         ;#500 mW
$opt(netif) set freq_ 2.45e+09


#### Load the scenario parameters ####

# grid size, number of nodes and simulation time
source manhattan1.ns_params

#### Load connection properties ####

set opt(nc) 24 ;#number of connections
set packSize 512 ;#size in bytes
set interv 0.256
set con_min 3
set con_max 5
set rateC 16kbps
```

```
# to avoid DSR segmentation fault
if { $opt(rp) == "DSR" } {
   set opt(ifq) CMUPriQueue;
} else {
   set opt(ifq) Queue/DropTail/PriQueue;
}

set ns_ [new Simulator]
$ns_ use-scheduler Heap
set tracefd [open randomtrace.tr w]
#set namtrace [open randomnam.nam w]
set chan [new $opt(chan)]
set prop [new $opt(prop)]

$ns_ trace-all $tracefd
#$ns_ namtrace-all-wireless $namtrace $opt(x) $opt(y)

# set up topography object
set topo [new Topography]
$topo load_flatgrid $opt(x) $opt(y)
create-god $opt(nn)
$prop topography $topo

#### Create nn mobilenodes [$opt(nn)] and attach them to the channel ####

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


for {set i 0} {$i < $opt(nn) } { incr i } {
   set node_($i) [$ns_ node]
   $node_($i) random-motion 0
   $ns_ initial_node_pos $node_($i) 5
}
```

```
#### Set up the connection between two random nodes ####

set rngNumber [new RNG]
$rngNumber seed 157

#create UDP connection between two random nodes with CBR traffic

for {set i 0} {$i < $opt(nc)} {incr i} {
  set same 1
  set num_udpS_($i) [expr [$rngNumber integer $opt(nn)]]
  while {$same == 1} {
   set same 0
   set num_udpG_($i) [expr [$rngNumber integer $opt(nn)]]
   # we can not have the same node as source and destination
   while {$num_udpS_($i) == $num_udpG_($i)} {
      set num_udpS_($i) [expr [$rngNumber integer $opt(nn)]]
   }
   for {set j 0} {$j < $i} {incr j} {
     if {$num_udpG_($i)==$num_udpG_($j) && $num_udpS_($i)==$num_udpS_($j)}{
        set same 1
        break
     }
   }
  }

set udp_($i) [new Agent/UDP]
$ns_ attach-agent $node_($num_udpG_($i)) $udp_($i)

set null_($i) [new Agent/Null]
$ns_ attach-agent $node_($num_udpS_($i)) $null_($i)

$ns_ connect $udp_($i) $null_($i)

set cbr_($i) [new Application/Traffic/CBR]
$cbr_($i) attach-agent $udp_($i)
$cbr_($i) set packetSize_ $packSize
$cbr_($i) set rate_ $rateC
$cbr_($i) set random_ false

}


#### Set up the traffic pattern between nodes ####

set rngStart [new RNG]
$rngStart seed 311
```

```
set RVStart [new RandomVariable/Exponential]
$RVStart use-rng $rngStart
$RVStart set avg_ [expr $opt(stop)/2]

set rngStop [new RNG]
$rngStop seed 229
set RVStop [new RandomVariable/Uniform]
$RVStop use-rng $rngStop

for {set i 0} {$i < $opt(nc)} {incr i} {
   set start_time_($i) [expr [$RVStart value]]
   while {$start_time_($i) > $opt(stop)} {
      set start_time_($i) [expr [$RVStart value]]
   }
   $RVStop set min_ [expr $start_time_($i)+$con_min]
   $RVStop set max_ [expr $start_time_($i)+$con_max]
   set stop_time_($i) [expr [$RVStop value]]
   if { $stop_time_($i) > $opt(stop) } {
      set stop_time_($i) $opt(stop)
   }
   $ns_ at $start_time_($i) "$cbr_($i) start"
   $ns_ at $stop_time_($i) "$cbr_($i) stop"
}

puts "Loading scenario file..."
source $opt(sc)

####  ending nam and the simulation ####

for {set i 0} {$i < $opt(nn) } { incr i } {
   $ns_ at $opt(stop) "$node_($i) reset"
}

proc stop {} {
    global ns_ tracefd
    $ns_ flush-trace
    close $tracefd
    #close $namtrace
}

#$ns_ at $opt(stop) "$ns_ nam-end-wireless $opt(stop)"
$ns_ at $opt(stop) "stop"
$ns_ at [expr $opt(stop)+0.01] "puts \"end simulation\" ; $ns_ halt"

$ns_ run
```

C.3 Rescue operation

```
### Define options ###

set opt(chan) Channel/WirelessChannel; # channel type
set opt(prop) Propagation/TwoRayGround; # radio-propagation model
set opt(netif) Phy/WirelessPhy; # network interface type
set opt(mac) Mac/802_11; # MAC type
set opt(ll) LL; # link layer type
set opt(ant) Antenna/OmniAntenna; # antenna model
set opt(ifqlen) 50; # max packet in ifq
set opt(rp) AODV;  # routing protocol

# nodes movements file generated by Bonnmotion
set opt(sc)  "./disaster_area1.ns_movements"


#### Define de transmition range ####

$opt(ant) set Gt_ 1
$opt(ant) set Gr_ 1
$opt(ant) set Z_ 1.6

$opt(netif) set L_ 1.0
$opt(netif) set RXThresh_ 1.31072e-07 ;#100m
$opt(netif) set Pt_ 2 ;#2 Watts
$opt(netif) set freq_ 1.5e+08 ;#radio channel


#### Load the scenario parameters ####

# grid size, number of nodes and simulation time
source disaster_area1.ns_params


#### Load connection properties ####

set opt(nc) 50 ;#number of connections
set packSize 64 ;#size in bytes
set rateC 2.4kbps ;#bitrate
set exp_avg 20 ;#average duration of the calls
set par_avg [expr $opt(stop)/3] ;#pareto average
set par_shape 1.3 ;#pareto shape

# to avoid DSR segmentation fault
if { $opt(rp) == "DSR" } {
```

```
set opt(ifq) CMUPriQueue;
} else {
set opt(ifq) Queue/DropTail/PriQueue;
}

set ns_ [new Simulator]
$ns_ use-scheduler Heap
set tracefd [open randomtrace_aodv.tr w]
#set namtrace [open randomnam.nam w]
set chan [new $opt(chan)]
set prop [new $opt(prop)]

$ns_ trace-all $tracefd
#$ns_ namtrace-all-wireless $namtrace $opt(x) $opt(y)


####  set up topography object ####

set topo [new Topography]
$topo load_flatgrid $opt(x) $opt(y)
create-god $opt(nn)
$prop topography $topo


#### Create opt(nn) mobilenodes and attach them to the channel ####

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

for {set i 0} {$i < $opt(nn) } { incr i } {
   set node_($i) [$ns_ node]
   $node_($i) random-motion 0
   $ns_ initial_node_pos $node_($i) 5
}
```

```
#### Set up the connection between two random nodes ####

set rngNumber [new RNG]
$rngNumber seed 79

#create UDP connection between two random nodes with CBR traffic

for {set i 0} {$i < $opt(nc)} {incr i} {
  set same 1
  set num_udpS_($i) [expr [$rngNumber integer $opt(nn)]]

  while {$same == 1} {
   set same 0
   set num_udpG_($i) [expr [$rngNumber integer $opt(nn)]]

   # we can not have the same node as source and destination
   while {$num_udpS_($i) == $num_udpG_($i)} {
      set num_udpS_($i) [expr [$rngNumber integer $opt(nn)]]
   }
   for {set j 0} {$j < $i} {incr j} {
      if {$num_udpG_($i)==$num_udpG_($j) && $num_udpS_($i)==$num_udpS_($j)}{
         set same 1
         break
      }
   }
  }

set udp_($i) [new Agent/UDP]
$ns_ attach-agent $node_($num_udpG_($i)) $udp_($i)

set null_($i) [new Agent/Null]
$ns_ attach-agent $node_($num_udpS_($i)) $null_($i)
$ns_ connect $udp_($i) $null_($i)

set cbr_($i) [new Application/Traffic/CBR]
$cbr_($i) attach-agent $udp_($i)
$cbr_($i) set packetSize_ $packSize
$cbr_($i) set rate_ $rateC
$cbr_($i) set random_ false

}


#### Set up the traffic pattern between nodes ####

# pareto distribution for arrivals
set rngStart [new RNG]
```

```
$rngStart seed 173
set RVStart [new RandomVariable/Pareto]
$RVStart use-rng $rngStart
$RVStart set avg_ $par_avg
$RVStart set shape_ $par_shape

# exponential distribution for duration
set rngStop [new RNG]
$rngStop seed 109
set RVStop [new RandomVariable/Exponential]
$RVStop use-rng $rngStop
$RVStop set avg_ $exp_avg

for {set i 0} {$i < $opt(nc)} {incr i} {

   set start_time_($i) [expr [$RVStart value]]
   set stop_time_($i) [expr $start_time_($i)+[$RVStop value]]

   if { $stop_time_($i) > $opt(stop) } {
      set stop_time_($i) $opt(stop)
   }

   $ns_ at $start_time_($i) "$cbr_($i) start"
   $ns_ at $stop_time_($i) "$cbr_($i) stop"
}

puts "Loading scenario file..."
source $opt(sc)

####  ending nam and the simulation ####

for {set i 0} {$i < $opt(nn) } { incr i } {
   $ns_ at $opt(stop) "$node_($i) reset"
}

proc stop {} {
    global ns_ tracefd
    $ns_ flush-trace
    close $tracefd
    #close $namtrace
}

#$ns_ at $opt(stop) "$ns_ nam-end-wireless $opt(stop)"
$ns_ at $opt(stop) "stop"
$ns_ at [expr $opt(stop)+0.01] "puts \"end simulation\" ; $ns_ halt"

$ns_ run
```

Appendix D

Bonnmotion parameters

To generate a mobility model using *Bonnmotion* we have to write the following sequence:

```
bm <parameters> <application> <application parameters>
```

Where *parameters* means for example the place where we want to keep the generated mobility file, *application* the type of mobility model we want to use and *application parameters* the parameters of this model. When a mobility model is generated, the following parameters have to be defined independently of which model is chosen:

- -d Scenario duration
- -i Number of seconds to skip
- -n Number of nodes
- -x Width of simulation area
- -y Height of simulation area
- -R Random seed

Then each type of mobility model has its own variables. In the table D.1 the parameters of the three mobility models used in this research are listed. The following are some examples of how to generate the one of each case study in chapter 5.

Shopping Mall (Gauss-Markov)

bm -f shopping1 GaussMarkov -x 100 -y 60 -h 2 -n 50 -d 600 -i 3600

Basic vehicular network (Manhattan Grid)

bm -f manhattan1 ManhattanGrid -x 650 -y 650 -c 0.6 -e 5 -m 11 -o 31 -p 0.45 -t 0.3 -u 7 -v 7 -n 150 -d 360 -i 3600

Rescue operation (Disaster Area)

bm -f disasterarea1 DisasterArea -n 141 -x 275 -y 205 -p 20 -a 1 -g 140 -r 3 -q 3 -d 300 -e 6 -i 5000 -j 1 -b CCS1 -b CCS2 -b PWFTA1 -b IL1 -b APP1 -b TEL -o OBST1

IL1 = 75,200,75,170,225,200,225,170,75,182,140,170,0,20,15

PWFTA1 = 20,190,20,174,56,190,56,174,38,174,56,182,1,20,17

CCS1 = 20,140,46,140,20,90,46,90,33,140,33,90,2,15,0

CCS2 = 66,110,92,110,66,60,92,60,79,110,79,60,2,15,0

TEL = 40,10,65,10,65,35,40,35,50,10,50,11,3,6,0

APP1 = 160,15,160,55,200,15,200,55,270,0,130,0,160,25,200,25,4,30,25

OBST1 = 230,200,230,140,270,200,270,140

Table D.1: Configurable parameters for each mobility model

Mobility model	Parameters
Static	-a [attractor parameters] -l [no. density levels]
Gauss Markov	-a [angle standard deviation] -h [max. speed] -q [speed, angle update frequency] -s [speed standard deviation]
Manhattan Grid	-c [speed change probability] -e [min. speed] -m [mean speed] -o [max. pause] -p [pause probability] -q [update distance] -s [speed standard deviation] -t [turn probability] -u [no. of blocks along x-axis] -v [no. of blocks along y-axis]
Disaster area	-p [max. pause] -a [group size] -g [circle vertices] -r [distance] -q [min. dist] -e [max. areas] -j [factor] -b [new area: size, position and type of area] -o [new obstacle: size and position]

Bibliography

[1] V.Ramesh, Dr.P.Subbaiah, N. K. Rao, and M. Raju, "Performance comparison and analysis of DSDV and AODV for MANET", *IJCSE*, 2010.

[2] H. Ehsan and Z. A. Uzmi, "Performance comparison of ad-hoc wireless network routing protocols", *INMIC*, 2004.

[3] S. R. Biradar, H. H. D. Sarma, K. Sharma, S. K. Sarkar, and P. C., "Performance comparison of reactive routing protocols of MANETs using group mobility model", *ICSPS*, 2009.

[4] W. Lin-zhu, F. Ya-qin, and S. Min, "Performance comparison of two routing protocols for ad-hoc networks", *WASE*, 2009.

[5] T. Camp, J. Boleng, B. Williams, L. Wilcox, and W. Navidi, "Performance comparision of two location based routing protocols for ad-hoc networks", *IEEE INFOCOM*, pp. 1678–1687, 2002.

[6] F. Yongsheng, W. Xinyu, and L. Shanping, "Performance comparison and analysis of routing strategies in mobile ad-hoc networks", *CSSE*, 2008.

[7] K. S. Lee, S. J. Lee, and Y. K. Chung, "A performance comparison of on-demand routing protocols for application data in mobile ad-hoc networks", *SERA*, 2005.

[8] I. Broustis, G. Jakllari, T. Repantis, and M. Molle, "A comprehensive comparison of routing protocols for large-scale wireless manets", *IEEE*, 2006.

[9] H. K. Ong, H. L. Ong, E. R. Tee, and R. Sureswarant, "Scalability study of ad-hoc wireless mobile network routing protocol in sparse and dense networks", *IEEE*, 2006.

[10] P. Singh and D. Lobiyal, "Scalability of routing in MANET", *ICCSN*, 2010.

[11] A. Shrestha and F. Tekiner, "On manet routing protocols for mobility and scalability", *PDCAT*, 2009.

[12] Y. Peiyan and L. Layuan, "Performance evaluation and simulations of routing protocols in ad-hoc networks", *BWAN*, 2006.

[13] T. Yang, M. Ikeda, G. D. Marco, and L. Barolli, "Performance behavior of AODV, DSR and DSDV protocols for different radio models in ad-hoc sensor networks", *ICPPW*, 2007.

[14] J. A. Ferreiro-Lage, J. F. Galvez-Galvez, O. R. Lopez, and F. A. Agelet, "Influence of propagation models in 802.11p car-to-car network simulations", *IEEE*, 2010.

[15] P. Prabhakaran and R. Sankar, "Impact of realistic mobility models on wireless networks performance", *IEEE*, 2006.

[16] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad-hoc network research", *WCMC*, 2002.

[17] B. Divecha, A. Abraham, C. Grosan, and S. Sanyal, "Analysis of dynamic source routing and destination-sequenced distance-vector protocols for different mobility models", *AMS*, 2007.

[18] R. Baumann, S. Heimlicher, and M. May, "Towards realistic mobility models for vehicular ad-hoc networks", *IEEE*, 2007.

[19] C. E. Perkins and P. Bhagwat, *DSDV Routing over a Multihop Wireless Network of Mobile Computers*. 2001.

[20] D. Johnson, Y. Hu, and D. Maltz, "The dynamic source routing protocol (DSR) for mobile ad-hoc networks for IPv4", *IETF*, 2007.

[21] C. Perkins, E. Belding-Royer, and S. Das, "ad-hoc on-demand distance vector (AODV) routing", *IETF*, 2003.

[22] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)", *IETF*, 2003.

[23] V. Park and S. Corson, "Temporally-ordered routing algorithm (TORA) functional specification", *IETF*, 2001.

[24] Y. Ko and N. Vaidya, "Location-aided routing (LAR) in mobile ad-hoc networks", *Mobicom*, pp. 66–67, 1998.

[25] V. Borrel, M. D. de Amorim, and S. Fdida, "On natural mobility models", *LIP6/CNRS*, 2005.

[26] C. A. V. Campos, D. C. Otero, and L. F. M. de Moraes, "Realistic individual mobility markovian models for mobile ad-hoc networks", *IEEE*, 2004.

[27] C. Bettstetter, H. Hartenstein, and X. PerezCosta, "Stochastic properties of the random waypoint mobility model", *ACM/Kluwer Wireless Networks*, 2003.

[28] B. Liang and Z. J. Haas, "Predictive distance-based mobility management for PCS networks", *IEEE*, 2002.

[29] M. Fiore, "Mobility models in inter-vehicle communications literature", 2007.

[30] N. Aschenbruck, E. Gerhards-Padilla, M. Gerharz, M. Frank, and P. Martini, "Modelling mobility in disaster area scenarios", *Proc. IEEE*, 2007.

[31] F. Bai and A. Helmy, *A survey of mobility modeling and analysis in wireless adhoc networks.* 2004.

[32] J. Ariyakhajorn, P. Wannawilai, and C. Sathitwiriyawong, "A comparative study of random waypoint and gauss-markov mobility models in the performance evaluation of MANET", *IEEE*, 2006.

[33] NS2 official website "www.isi.edu/nsnam/ns/"

[34] Bonnmotion official website "net.cs.uni-bonn.de/wg/cs/applications/"

[35] E. Altman and T. Jimnez, *NS Simulator for beginners.* 2003.

[36] T. Issariyakul and E. Hossain, *Introduction to Network Simulator NS2.* Springer, 2009.

[37] T. V. Project, *The ns Manual.* 2008.

[38] M. Umlauft and P. Reichl, "Experiences with the ns-2 network simulator - explicitly setting seeds considered harmful", *Proc. IEEE*, 2007.

[39] J. B. et al., "A performance comparison of ad-hoc multihop wireless networks routing protocols", *Proc. IEEE/ACM MOBICOM*, 1998.

[40] S. R. Chaudhry, A. Al-Khwildi, Y. Casey, H. Aldelou, and H. S. Al-Raweshidy, "A performance comparison of multi on-demand routing in wireless ad-hoc networks", *IEEE*, 2005.

[41] E. van Eenennaam, "A survey of propagation models used in vehicular ad-hoc network (VANET) research", 2009.

[42] I. Gruber and H. Li, "Behavior of ad-hoc routing protocols in metropolitan environments", *IEEE*, 2004.

[43] S. Jaap, M. Bechler, and L. Wolf, "Evaluation of routing protocols for vehicular ad-hoc networks in city traffic scenarios", *IEEE*, 2005.

[44] J. Collura and D. Rahikka, "Interoperable secure voice communications in tactical systems", *IEE Colloq. on Speech Coding Algorithms for Radio Channels*, 2000.

[45] R. Datta, *Real-Time Communications.* CSE, IIT Kharagpur, 2007.

[46] H. T. Friis, "A note on a simple transmission formula", *Proc. IRE*, 1946.

[47] T. S. Rappaport, *Wireless communications, principles and practice.* Prentice Hall, 1996.

List of Figures

List of Tables