UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

IMPLEMENTATION AND ANALYSIS OF LOCATION-BASED ROUTING PROTOCOLS FOR MANETS

A Master's Thesis

Submitted to the Faculty of the

Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

Universitat Pòlitecnica de Catalunya

By

Fabio Perrone

In partial fulfilment

of the requirements for the degree of

MASTER IN TELECOMMUNICATIONS ENGINEERING

(Erasmus+ Exchange Program)

Advisors: Israel Martin Escalona, Enrica Zola

Barcelona, September 2016

POLITECNICO DI TORINO

ABSTRACT

This thesis concerns routing protocols for MANETs with a particular focus on location-based ones. After a deep overview of the literature, one regular routing protocol, DYMO, and two location-based (LB) ones, DYMOselfwd and AODV-Line, have been selected for further study. To this end, they have been implemented and simulated with the OMNET++ simulator. The scenarios are chosen to evaluate the impact of the node density, the nodes' mobility behaviour and of the ping payload on the performance of the routing protocols, in terms of scalability and ability to recover from route disruptions in a mobile scenario. In addition, the impact of an error in the location information is also analysed in the case of the two LB protocols.

Index

Index of Figures

Glossary

AODV-Line: Ad hoc On-Demand Distance Vector - Line

AvgHops: Average number of hops

DYMO: Dynamic Manet On-Demand

DYMOselfwd: Dynamic Manet On-Demand Self Forward

e2eDel: End to End Delay

LB: location-based

MANET: Mobile Adhoc Networks

NRL: Normalized Routing Load

Nthr: Network throughput

PDR: Packet Delivery Ratio

PL: Packet Loss

Pow: Power consumption

RDD: Route Discovery Delay

RERR: Route Error

RLT: Route Life Time

RO: Routing Overhead

RREP: Route Reply

RREQ: Route Request

RTT: Round Trip Time

1. Introduction

A Mobile Ad Hoc Network (MANET) is a network built without the support of an infrastructure. Originally, this kind of network was considered interesting especially for a military environment or in a disaster scenario to allow communication even when infrastructures were not available. In the last decade, importance of MANETs has grown because of all its possible applications and thanks to the easiness to join the network itself; in fact, its nodes could be any laptop, or smartphone, etc.

In a MANET, in addition to being the source and the destination of a packet, nodes also act as relays in order to allow other nodes to communicate with each other. It's also important to keep in mind that the topology of a MANET is often variable due to nodes' mobility (e.g., nodes may join it, leave or physically move away). Actually, facing dynamic topologies is just one of the challenges of MANETs. Other relevant issues are network scalability, energy efficiency, Quality of Service (QoS), network overhead, efficient routing and security [1].

To achieve a successful communication between nodes, many routing protocols were proposed in the literature. These protocols are mainly divided in three categories: 1) proactive protocols or table-driven, where a node has information about every other node in the network; 2) reactive protocols or on-demand, where nodes don't store information about other nodes and find a route to communicate just when needed; 3) hybrid protocols, which are a mixture between the two different approaches previously mentioned.

It is not possible to clearly state that one of the three previously mentioned classes of routing protocols is better than the others because it depends on the scenario used in the study and on the metrics of interest on which the authors have been focusing. In general, a good routing protocol should be able to find a route between source and destination no matter the dimension of the network; it should not let nodes run out of energy too fast or let them consume battery when not necessary; it should not take too much time to build the route and should limit the routing overhead.

The knowledge of the positions of the nodes in the network may improve the performance of a routing protocol. Systems like GPS, for example, can be used to improve routing, allowing to limit flooding of packets, and so to save energy and, in general, to better manage network resources.

This thesis focuses on three routing protocols for MANETs: DYMO, DYMOselfwd and AODV-Line. DYMOselfwd is a modification of the original DYMO, a well-known routing protocol, while AODV-Line is a modification of AODV, another well-known one. Both DYMOselfwd and AODV-Line use the information of the nodes' position to improve the performance of the regular protocol.

These three protocols have been implemented in Omnet++ and then simulated in different scenarios, varying the number of nodes, the packet payload and node speed.

2. Goals

2.1. Scope of the Thesis

Studying and implementing new routing protocols for a MANET necessarily rely on a deep understanding of its protocol's mechanisms and performance.

This work has been necessary to better understand the performance of the main MANET routing protocols, with a particular focus on location-based routing protocols. Also, this thesis has been motivated to better comprehend the impact of the knowledge of nodes' positions in routing, in particular in terms of routing overhead. Finally, a full understanding of the impact of the error in the location information on the routing performance is also needed and then assessed in this study.

2.2. Objectives

The following objectives have been defined in this thesis:

1. Bibliographic research about routing protocol for MANETs
2. Comparison between the most representatives routing protocols
3. Individuation of the interesting metrics for the performance evaluation
4. Implementation of the DYMOselfwd and of the AODV-Line in the Omnet++ simulator
5. Creation of the necessary routines that allow the collection of statistics from the studied routing protocols (DYMO, DYMOselfwd and AODV-Line)
6. Simulation of DYMO, DYMOselfwd, AODV-Line in different scenarios
7. Data collection and analysis

3. State of the art on routing protocols

This chapter offers an overview of routing protocols, but instead of pointing out differences between protocols in a classic way, classifying them as proactive, reactive and hybrid protocols, it focuses most on the differences in performances between location-based protocols and regular ones.

This chapter is organized as follow: Section 3.1 concerns the metrics on which researchers have been focus so far in both development and simulation phase. Section 3.2 gives an overview on taxonomy. Section 3.3 shows some MANETs routing protocols not based on position, while Section 3.4 is devoted to position-based MANETs routing protocols.

3.1. Metrics

Performance evaluation passes through the definition of some metrics of interest. Depending on the protocol and on the goal that it wants to achieve, metrics may change; however, there are some of them that are often present in many of the works in the literature. In the following sections, the most used metrics are presented together with a brief description. Later in this chapter (see Section 6), we will provide a comparison based on these metrics:

- **End-to-end delay (e2eDel)**. It is the overall time between the instant when a packet is generated by the source until the instant when it is received by the destination [1]. It can be expressed as:

$$\boldsymbol{delay_{avg}} = \frac{\sum_{i=1}^{t_{tot}} \left(\boldsymbol{d_d + d_{tr} + d_{pr} + d_q + d_{pp}} \right)_i}{\boldsymbol{t_{tot}}}, \qquad \textbf{Equation 1}$$

  where d_d is the delay for route discovery, d_{tr} is the delay for transmissions, d_{pr} is the delay for processing, d_q is the delay for queuing, d_{pp} is the delay for propagation, *i* represents the single transmission and t_{tot} is the total number of transmissions. It's better to have this value as lower as possible. In some papers, the end-to-end delay is normalized dividing it by the average number of nodes in the simulating scenario.

- **Packet delivery ratio (PDR)**. It is the ratio of the number of data packets received by the destination over the number of data packets generated by the source [1]:

$$\boldsymbol{Packet\ delivery\ ratio} = \frac{\sum_{i=1}^{n} \boldsymbol{P_{d \leftarrow s_i}}}{\sum_{j=1}^{n} \boldsymbol{P_{s \rightarrow d_j}}}, \qquad \textbf{Equation 2}$$

  where $P_{s \rightarrow d}$ is a packet sent by source node *s* to destination node *d*, $P_{d \leftarrow s}$ is a packet received by destination *d* from source *s*, and *n* is the total number of nodes in the network. It's better to have this metric as near as possible to 1.

- **Normalized routing load (NRL)**. It is the ratio of the number of routing packet transmitted (RPT) by all nodes in the network over the number of data packet received (DPR) by all destination nodes [1]:

$$NRL = \frac{\sum_{i=1}^{n} RPT_i}{\sum_{i=1}^{n} DPR_i}, \qquad \text{Equation 3}$$

where RPT_i is the sum of all routing packets transmitted by node *i*, DPR_i is the sum of all data packets received by node *i*. The lower the NRL is, the better performance we get.

- **Routing overhead (RO)**. It is the number of routing packets transmitted by all the nodes in the network. These packets use part of the network bandwidth so it is important to minimize their number as much as possible. The routing overhead can be expressed as:

$$Routing\ overhead = \sum_{i=1}^{n} RPT_i . \qquad \text{Equation 4}$$

Some other metrics that is possible to consider are:

- **Average number of hops (AvgHops)**, which is the average number of nodes that retransmit the packet towards the destination [2].
- **Network throughput (NThr)**, which is the sum of the number of bits per second that every node in the network can transmit.
- **Packet loss (PL)**, which is the number of packets that are dropped by nodes [1].
- **Route discovery delay (RDD)**, which is the time needed to find a route from source to destination.
- **Route life time (RLT)**, which is the average time a route is not broken.
- **Round trip time (RTT)**, which is the time from the moment a node sends a Ping to the moment it receives the Pong for the Ping sent.
- **Power consumption (Pow)**, which is the power used by nodes. Researchers usually decide to personalize this metric, focusing on some particular energetic aspect or normalizing this value with the number of transmissions.

3.2. Taxonomy of MANETs routing protocols

MANETs routing protocols can be distinguished according to several possible criteria. According to a recently published survey [3], protocols can be classified as: 1) static, when routes are maintained independently of traffic condition; 2) adaptive, when the source-destination routes change in order to avoid network congestions. A static protocol is easy to implement in a small network but it can't really manage high nodes' mobility and it is not suited for high variation of traffic conditions especially when the size of the network increases; on the other hand, an adaptive protocol is more complex to implement, but at the same time it can be applied to any type of topology and of traffic, no matter the size of the network.

The same authors in [3] also propose other taxonomies. For example, routing protocols can be divided into distributed and centralized, depending if the assessment of routes is done in a shared way among nodes or not. A centralized protocol has a better management of the network but the nodes in charge of the decisions will run out of battery sooner than other nodes. Yet, another possible distinction could be made based on some specific rules given to the nodes [3]. Thus, a hierarchical routing protocol will have some gateway nodes that can control transmissions of a group of other nodes. In a flat routing protocol all nodes follow the same routing algorithm, instead. A hierarchical protocol can scale much better than a flat one, but the nodes that act as gateways risk running out of battery.

A typical classification is as proactive, reactive or hybrid protocol. The proactive protocols are based on traditional distance vector (DV) and link state (LS) protocols. The main mechanisms adopted in proactive

protocols are: 1. Increase the amount of topology information stored at each node (to avoid loops and speed up protocol convergence); 2. Vary dynamically the size of route updates and the update frequency; 3. Optimize flooding to combine DV and LS features [4]. A proactive protocol is the best option in small networks that are not too dynamic but it cannot scale well with network size because keeping updated the routing tables would get too difficult. On the other hand, reactive protocols are different from proactive, as they don't use routing tables; instead, the routes are discovered when needed through routing request packets (RREQ) and destination replies with Route Reply (RREP) packets [3]. While this process adds some delay, it scales better than proactive protocols in case the number of nodes in the network increases.

Hybrid routing protocols are a mixture between proactive and reactive protocols; they try to take the benefits from both approaches, especially for scalability (i.e., typical of reactive protocols) and for lower delay (i.e., typical of proactive protocols). Usually the area is divided into zones and for an intra-routing zone the approach is proactive, while for an inter-routing zone a reactive strategy is used.

Finally, it is also possible to distinguish between routing protocols that use location information to deliver data as location-based (LB) routing protocols. The knowledge of nodes' position allows preserving bandwidth as the route discovery messages can be flooded in a geo-localized area, and consequently it paves the way to energy saving, as fewer nodes are involved in the route discovery and/or routing. It also opens new possibilities such as geocasting, i.e. multicast a packet to every node in a limited geographic area. The main drawback of LB routing protocols is that position information, obtained with GPS or other systems, comes with an inner error that may severely affect packet routing.

In the following, the main routing protocols for MANETs are briefly described.

3.3. Regular protocols

3.3.1. Proactive protocols

A. Destination-Sequenced Distance-Vector (DSDV) Routing Protocol

DSDV is a proactive protocol based on Bellman Ford algorithm and guarantees loop free routes [5]. It provides a single path to a destination, which is selected using the DV shortest path routing algorithm. Each node exchanges its neighbour (routing) table periodically with its neighbours. In order to reduce the amount of overhead transmitted through the network, two types of update packets are used. These are referred to as a full dump and incremental packets [4]. The first ones carry all the available routing information and the second ones only carry the information changed since the last full dump.

B. Wireless routing protocol (WRP)

The wireless routing protocol (WRP) is also based on Bellman Ford algorithm [6]. The main duty of a node is to keep updated four tables:

- Distance table
- Routing table
- Link-cost table
- Message retransmission list

The maintenance of these tables add a significant memory overhead while the HELLO messages that the nodes exchange, which are needed for ensuring connectivity, consume a significant amount of bandwidth [4].

C. Optimized Link State Routing (OLSR)

The OLSR is a table driven protocol where each node elects a Multipoint Relay (MPR). The MPR is in charge of forwarding control traffic, of announcing periodically the link-state information in their control messages, and is used to form a route for a given node to any destination. The use of MPR allows the network to facilitate efficient flooding of control messages [7].

D. Cluster Gateway Switch Routing Protocol (CGSR)

The CGSR is a protocol where nodes are grouped into clusters in charge of each cluster there is a cluster-head, as shown in Figure 1, whose controls inter-cluster transmissions [8]. When a cluster-head moves or run out of battery, a new one is elected. When a node spatially moves, it needs to update its cluster and the cluster-head handling its transmission.

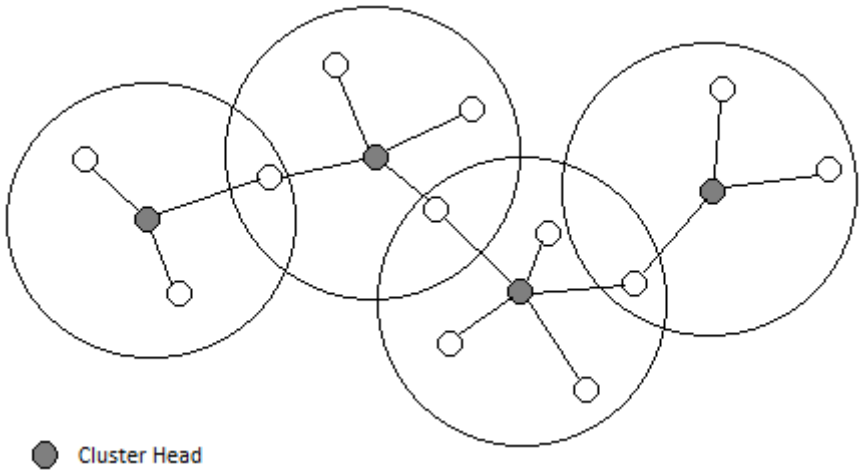

Figure 1: Cluster Gateway Switch Routing Protocol – Cluster organization

E. Global State Routing (GSR)

The GSR protocol is based on LS algorithm and it is a uniform, topology oriented, proactive routing protocol. It modifies and improves the link state algorithm by restricting the update messages between intermediate nodes only. Each node maintains a neighbours' list, a topology table, and a next hop table [4].

F. Fisheye State Routing (FSR)

FSR is a hierarchical proactive protocol based on fisheye technique. In FSR each node has information about the other nodes, but the amount of information on a given node decreases as the distance between them increases [9]. So, referring to Figure 2, the red node will have accurate information about nodes in the grey coloured area and it will exchange link state information about immediate neighbours more frequently. For this reason, the routing accuracy of nodes that retransmit RREQ increases as the destination gets closer.

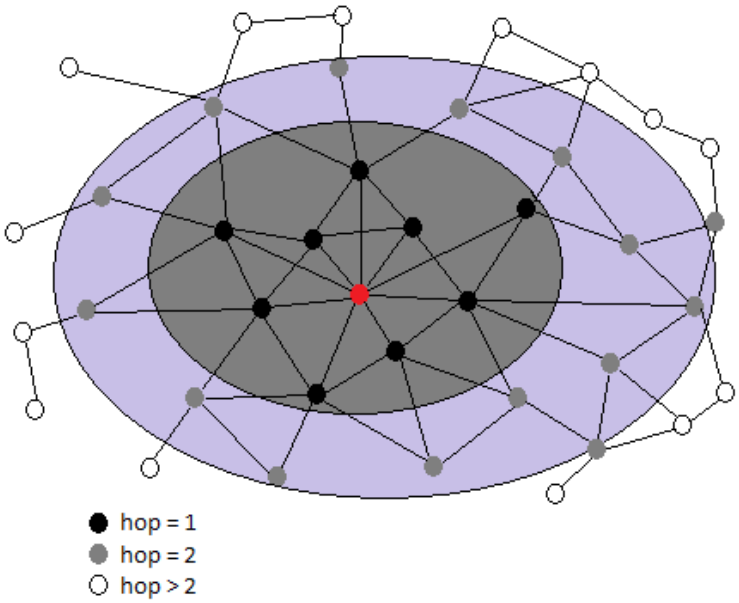

Figure 2: Fisheye State Routing

3.3.2. Reactive protocols

A. Ad hoc On-Demand Distance Vector (AODV)

AODV is a reactive protocol that uses RREQ and RREP to find a route between source and destination, as illustrated in Figure 3. When a source wants to send a packet, it first starts to send route request packets to all its neighbours; each of its neighbours is in charge of sending again this packet until it reaches the destination. At this point, the destination sends back a route reply through the reverse path until the source. The first RREQ packet received by the destination is likely to have traversed a path with low delay and/or hop count. Representing the weight of each link in the network by the delay incurred on the link, AODV reduces to finding a minimum-weight path between the source and the destination [10].

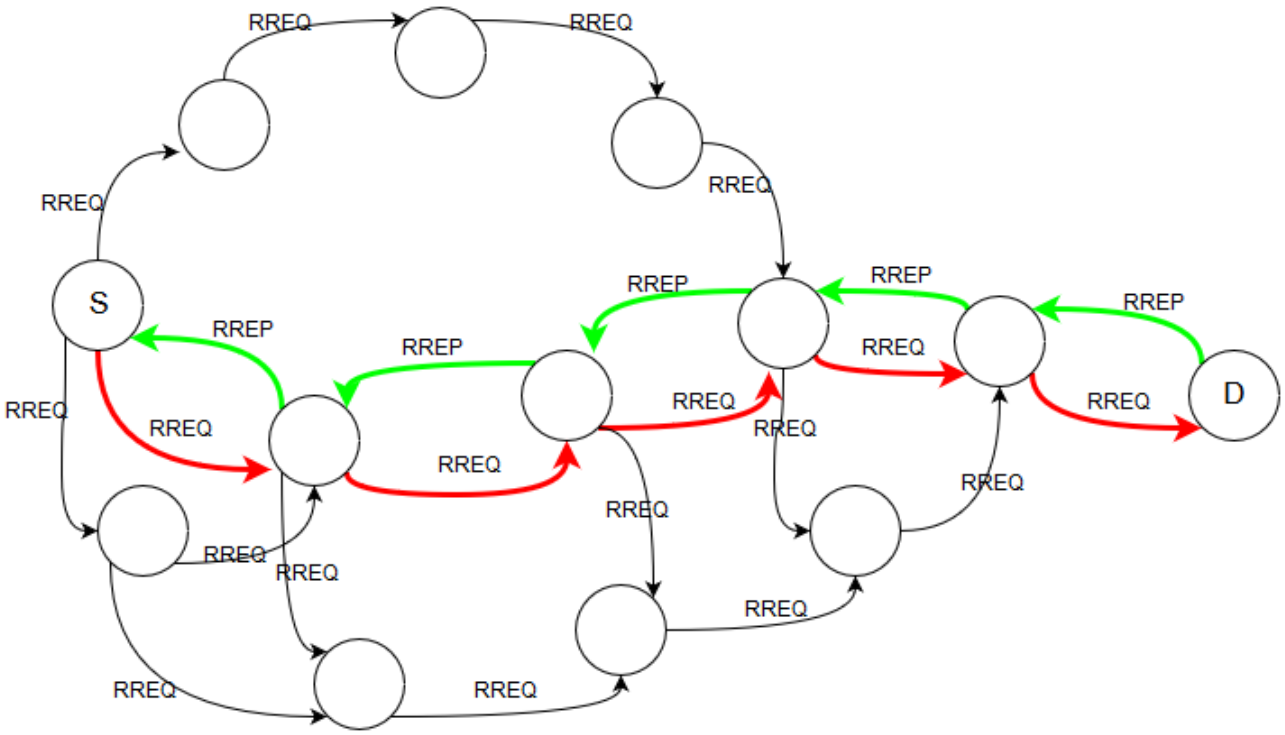

Figure 3: AODV route discovery mechanism

An extension of AODV is the Ad-hoc On-demand Multipath Distance Vector (AOMDV) that computes multiple loop-free and link-disjoint paths. In this protocol every route is tracked using a sequence number and for each destination a node maintains the advertised hop count, which is defined as the maximum hop count for all the paths, which is used for sending route advertisements of the destination. Each duplicated route advertisement received by a node defines an alternate path to the destination [11].

B. Dynamic Source Routing Protocol (DSR)

The DSR is a beaconless reactive protocol that uses a cache to store already discovered routes to avoid further flooding. It is mainly composed by two mechanisms: 1. route discovery and 2. route maintenance. DSR has been designed to compute correct routes in the presence of asymmetric (unidirectional) links [12].

C. Temporarily-Ordered Routing Algorithm (TORA)

The TORA is an adaptive routing protocol for multi-hop networks [13]. TORA supports multiple path routing, which allows it to reduce the overhead for route discovery, as it does not need to discover a new route when the network topology changes unless all routes in the local route cache fail. Hence, the trade-off is that since multiple paths are used, routes may not always be the shortest ones [14]. In TORA there are four main operations that are creating, maintaining, erasing and optimizing routes.

D. Cluster-Based Routing Protocol (CBRP)

CBRP is a hierarchical protocol where nodes are grouped in cluster coordinated by cluster-heads. Each cluster-head has the duty to manage the route discovery inter clusters in order to minimize the flooding

traffic for route discoveries. Clusters have a diameter of 2 hops and the intra-cluster discovery is done by flooding [15].

E. Channel Quality Based Routing Protocol (CQBR)

CQBR is a variation of AODV with the objective to increase the stability of route that could cause packet loss. To improve the quality of links that makes up the whole route, CQBR monitors every link channel quality, and only allows RREQ packets to pass through such links with a mean RSSI (Received Signal Strength Indicator) value higher than a threshold [16].

F. Load Equilibrium Neighbour Aware Routing (LUNAR)

LUNAR is a protocol that tries to minimize the end to end delay and routing overhead. The main feature stands in the routing discovery process that is as follows: (a) each node determines its neighbours using hello packets, (b) each node uses the concept of node centrality (eccentricity) and uncovered nodes to reduce the RREQ broadcast storm, (c) the destination finally selects the least loaded path [17].

G. Dynamic MANET On-demand (DYMO)

DYMO is a reactive, multi-hop, unicast routing protocol based on AODV[1] [18]. The most important feature of DYMO is that intermediate nodes are able to learn the route to all the predecessor nodes in the path, thus lessening the number of RREQs generated in the network by AODV [19].

An enhancement is E2DYMO, in which, without introducing new protocol messages, it's possible to detect the nodes that reach a critical battery level and choose a different path in order to let the network last more [20].

Another possibility is En-DYMO, a routing mechanism that selects the best route based on route priority function, that is a function of energy (higher is better) and traffic (lower is better) [21].

3.3.3. Hybrid protocols

A. Zone Routing Protocol (ZRP)

The ZRP divides the region of interest in several zones, using a proactive approach inside the same zone and a reactive one between different zones. ZRP is composed of three other protocols [22]:

- Neighbor Discovery Protocol (NDP)
- Intra Zone Routing Protocol (IARP)
- Inter Zone Routing Protocol (IERP)

A problem of ZRP is the presence of overlapping zones that produces duplicate route request. As a solution to this problem, the Hierarchical Zone Routing Protocol (H-ZRP) was proposed. In H-ZRP a fundamental key is the definition of a variable (*N*) that refers to node's ability to manage a network. This variable is affected

[1] In the beginning, it was actually named AODVv2. In 2013, they decided to change back to the old name of AODVv2. However, in this paper it has been named after DYMO.

by a lot of factors: the remaining energy of one node, the transmission power, the available band and the translation frequency [22]. Then, in every area the node with the highest value of *N* is elected as the managing node. Every time a node wants to send a packet to another but it does not know the route it sends it to the managing node, which checks if the destination is in its area, sending back a RREP if yes, or redirecting the RREQ to the gateway if no.

B. Zone-Based Hierarchical Link State (ZHLS)

In ZHLS protocol [23] the region is divided in non-overlapping zones with two topologies: a zone level topology and a node level topology. Inter-zone communication is made thanks to some nodes that act as gateways; every node inside the zone has two routing tables, one for the intra-zone routing and another one for the inter-zone.

C. Threshold based Hybrid Routing Protocol (THRP)

THRP is a successor of the ZRP protocol and it is defined as a velocity-aware routing protocol. In fact, it can adapt the behaviour of a node to the environment, joining a proactive cluster with slow nodes or acting as a single reactive node with highly moving speed [24].

D. AntHocNet

AntHocNet is a hybrid protocol based on Ant Colony Optimization (ACO) algorithm [25]. It's reactive because it obtains route information to destination just if they are involved in the communication session, and proactive because it has a route maintenance mechanism. Routing information is stored in "pheromone tables" that are similar to the ones used in other ACO routing algorithms. Forwarding of control and data packets is done in a stochastic way, using these tables. Link failures are dealt by using specific reactive mechanisms, such as local route repair and the use of warning message [25].

E. ZCR

The ZCG protocol is a hybrid protocol where nodes belong to different zones with a Zone Leader. The Zone Leader is elected according to a parameter called Fitness Factor, calculated taking into account the most desirable attributes: minimum mobility, a high degree of connectivity and plentiful battery power. The Zone Leader of the source starts the route discovery forwarding two Path Discovery Commands(PDC) message to ensure that they reach D via ZL(Destination) and S at the same time [26]. When S and D receive the PDC message, they broadcast route request messages (RREQs) in order to find one another and rebroadcasting continues at intermediate nodes until a positive RREQ-collision occurs [26]. The ZCG supports link failure maintenance similar to that used in the AODV routing protocol [26].

3.4. Location based protocols

In this section we focus on the main location-based routing protocols. In location-based protocols, routing and/or data delivery are aided by the knowledge of the position of the source and of the other nodes. This information is obtained thanks to systems like GPS or Galileo, which are typically affected by an error (i.e., depending on the system position evaluation, due to the mobility of the node while it receives the

information, etc.). However, it is generally assumed by all the following protocols, if not otherwise specified, that the positions of the nodes are not affected by any error during the simulation. Moreover, the nodes positions are assumed to be known without further investigating the impact of gathering such information.

3.4.1. Proactive Protocols

A. Distance Routing Effect Algorithm for Mobility (DREAM)

DREAM [27] was proposed in 1998 and was built upon two main ideas: distance effect (i.e., the greater the distance between two nodes, the slower they appear to be moving with respect to each other) and mobility rate; based on these concepts, a node can optimize the frequency at which it sends updates to the network, thus reducing the routing overhead (RO) and power consumption (Pow) in the network. The protocol stores the location of other nodes in routing tables. Control packets with node's coordinates are broadcasted periodically to keep other nodes' routing tables updated. Based on the routing tables, messages are sent in the direction of the destination node, and delivery is guaranteed by following the direction with a given probability.

The authors in [27] compare DREAM against DSR with respect to the end-to-end delay (e2eDel).

B. Location Based Routing Protocol (LBRP)

More recently, LBRP was proposed [28] in order to cope with the deterioration that unavailability of the GPS signal, e.g. in indoor scenarios, may cause to the performance of LB protocols. In fact, differently from GPSR and DREAM, it also uses the velocity information in geographic routing, which allows keeping the information of nodes' position valid for longer periods of time and in more constrained scenarios (e.g., indoor, dense urban areas, etc.). In addition, the source keeps updated information of the location while the session is still valid thanks to ACK packets. The protocol is composed of three procedures: location management, location discovery and transmission of data packets. Each node has a location table with all the coordinates and the speed of other nodes and does the route discovery by flooding.

The metrics used to compare the performance of this protocol against DREAM and DSR in [28] are: packet delivery ratio (PDR), RO, e2eDel and data load.

3.4.2. Reactive protocols

A. Greedy Perimeter Stateless Routing (GPSR)

In Greedy Perimeter Stateless Routing (GPSR) protocol [29] the location of nodes is used to decide to which node forward the packet. The node chosen will simply be the one closest to the destination. In some cases, greedy forwarding is not possible, so the right hand rule is used to forward in the perimeter of the obstacle.

The authors in [29] evaluate the performance of GPSR against DSR through the following metrics: packet delivery success rate, RO, and optimality of path lengths taken by data packets.

B. *GeoTORA*

GeoTORA is an improvement of TORA protocol that uses geocasting to reduce flooding. The TORA protocol is modified allowing to perform anycast, so a source can perform anycast to any node of the geocast region, when any node in the geocast group receives the packet, it floods the packet such that the flooding is limited to the geocast region [30]. The metrics used in the evaluation of the protocol are: accuracy of geocast delivery, defined as the ratio of the number of group members that actually receive the geocast packet and the number of group members which were in the geocast region at the time when the geocast delivery was initiated; overhead of geocast delivery, defined as the number of geocast packets received by the nodes [30].

C. *Location-Aided Routing (LAR)*

Location Aided Routing (LAR) protocol is based on the idea that the source node sends packets to the coordinative geographic location of the destination node instead of using its network address [31]. The position error is taken into account in the performance evaluation of this protocol [32]. The metrics used in the evaluation of the protocol are the number of routing packets per data packet and the number or routing packets per discovery.

LAR protocol is an evolution of DSR protocol, but in this case there are two important assumptions to make: the source knows its own location and the location of the destination. Having this information, the node forwards the packets only in the direction of the request zone, calculated according with the destination node's position.

D. *Improved Opportunistic Location Aided Routing (IOLAR)*

Based on the observation that there is an unavoidable interruption in the network due to link failures, the application of opportunistic routing in LAR protocol has been recently investigated [33]. An enhancement is proposed in the Improved Opportunistic Location Aided Routing (IOLAR) so to improve the linkability while routing. The location information of the destination node is attached to the packet header; when the source finds an available node to send the data to, it first checks for the link status of the node using the received signal strength of the node of interest.

In their evaluation against LAR protocol, the authors use the following metrics: network throughput (NThr), end-to-end delay and packet loss. As demonstrated in [33], IOLAR outperforms the LAR protocol because of the opportunistic routing mechanism incorporated and also because of the link-ability check at every communication operation.

E. *Predictive Location-Aided Routing (PLAR)*

Another recent improvement of LAR protocol is the Predictive LAR (PLAR) [34], which is proposed for mobility models in which the target of motion is known. The information of a destination node used in the LAR protocol may be out of date, especially in highly dynamic scenarios. Hence, PLAR add to LAR a motion prediction of destination nodes. In addition, when there is no information for a destination/location, LAR protocol turns to pure flooding algorithm. The PLAR covers this weakness by applying the new location service to disseminate the location information of other nodes in the network.

The metrics of evaluation of the protocol are total over ratio (TOR), which is defined as the ratio of the number of routing control packets to the total received control and data packets. Authors in [34] show that

the overhead is less than in LAR; also, they claim that PLAR is suitable for highly dynamic environments, such as vehicular networks.

F. Ad hoc On-Demand Distance Vector Location-Aided Routing (AODV-LAR)

Authors in [35]} propose to enhance the AODV protocol by minimizing the control message overhead. Their first proposal is AODV-LAR that tries to decrease the route discovery flooding of AODV using the LAR scheme. Knowing the location of the destination, AODV-LAR restricts the flooding to a rectangular area that contains the source and destination. If the source node knows the location and speed of the destination node at a given time, it can restrict the flooding to a restricted area instead of the whole network [35].

G. Ad hoc On-Demand Distance Vector Line (AODV-Line)

A second version of AODV-LAR is AODV-Line. Instead of considering a whole region where performing the flooding, it restricts the flooding of control packets to the imaginary line that connects the source and the destination nodes [35].

Authors in [35] compare their two proposal against AODV in terms of: RO, RDD, PDR and normalized routing load. AODV-LAR outperforms original AODV, while AODV-Line performs better than AODV-LAR.

H. *DYMO with selective forwarding (DYMOselfwd)*

Authors in [19] propose to include in the RREQ packet the position information of the transmitting node and of the destination node, in order to reduce the routing overhead of the original DYMO protocol and to save power. An algorithm is proposed so that each node that receives the RREQ is able to delay the retransmission of the RREQ; as the delay is proportional to the node's distance from the destination, the closest node is selected in a distributed fashion and is in charge of retransmission. Due to the broadcast nature of the channel, the other candidate nodes will prevent to send the RREQ, thus reducing the amount of messages in the network.

Authors in [19] compare DYMOselfwd against DYMO according to the following metrics: the time needed to establish the route; the number of collisions at the MAC module level; the total number of the MAC module frames sent by all nodes; the total number of RREQ sent by all nodes. In common scenarios in MANETs, DYMOselfwd performs better than DYMO [19].

I. Location based shortest path routing protocol (LBSPR)

LBSPR [36] is a protocol that uses the position of every node to calculate the shortest path. Location information is obtained by a source node only when needed (route discovery) and there are no intermediate nodes that take part in the route discovery. The protocol also implements an energy efficient approach as it utilizes location information of mobile nodes with the goal of decreasing routing overhead in MANETs. The metrics of evaluation used in [36] are route discovery delay and hop count.

J. Location Aware Sector-Based Routing (LSR)

The Location Aware Sector-Based Routing (LSR) protocol works on the principle of DSR for route setup and is an improvement of LAR protocol based on expected zone of the destination [37]. The route discovery uses

RREQ and RREP but the flooding is minimized thanks to the use of directional antenna (120°) in the direction of the *expected zone*, which is calculated with a prediction algorithm.

The metrics of evaluation of the protocol are: control packets per route discovery, route life time and control overhead comparison for maximum active routes [37].

3.4.3. Hybrid protocols

K. AntMANET

AntMANET [38] is a hybrid protocol based on the ACO algorithm. It's very similar to AntHocNet, but it uses some new table like north neighbour table and the geo table, that stores information such as coordinates and geo-life time of the node. The use of the location information allows to increase the routing efficiency and to reduce the complexity by minimizing the network latency, overhead and the power consumption [38]. The metrics of evaluation used are: traffic overhead, end-to-end delay, Energy-consumed-in-Receive-mode and Energy-consumed-in-Transmitting-mode [38].

L. Improved Hybrid Location-based Ad hoc Routing protocol (IHLAR)

IHLAR combines AODV with ARP (Angular Routing Protocol) that is a position-based routing protocol that uses an improved geographic greedy forwarding to route packets to the destination as and when possible. If the greedy forwarding fails, it uses an angle-based forwarding scheme to circumvent voids in sparser networks [39]. If the destination node resides within a given number of hops from the source or an intermediate node, then the source node or the intermediate node will route the packet using AODV protocol [39]. The metrics of evaluation of the protocol are: average e2eDel, average routing hops and PDR. Simulation results in [39] show that it outperforms both AODV and ARP.

3.5. Discussion

In this section all the previous protocols are compared. The comparison is based on results of simulation and testing done by other authors (see "Ref" in Table 1 and Table 2) and based on the proposed metrics.

A quantitative comparison is complicate as every author uses a specific scenario (i.e., number of nodes, size of the area, mobility parameters, etc.) and also because each protocol is tested using different metrics. Every scenario is characterized by several parameters. The most relevant ones are of course the dimension of the simulated area, the number of nodes, their speed and mobility, their pause time and their transmission range [29]. The scenario deeply depends on the metrics that we want to check. Usually, for a generic MANET routing protocol the goal is to minimize the end-to-end delay, minimize the traffic overhead and maximize the delivery ratio [29]. As regards to the mobility model, the most used is the Random Waypoint Model, in which nodes have a random speed and a random destination.

Therefore, a qualitative comparison among the main routing protocols is provided here, so that it can be used as a general reference for assessing their performance. To this end, a four-level marks system is applied to the expected performance of several metrics for each protocol. The four levels are: 1) Excellent, 2) Good, 3) Fair and 4) Poor. Table 1 shows the performance of the non-location-based protocols according to the

several metrics discussed before, while Table 2 provides the performance of the LB protocols. These tables also provide a reference to the paper where the study is published, together with the year of publication.

Table 1: Regular routing protocols comparison

Name	e2eDel	PDR	NLR	RO	AvgHops	NThr	PL	RDD	Pow	Year	Ref
DSDV	2	3	4	4	1	4	3	2	3	1994	[5]
WRP	2	3	4	4	2	4	3	2	3	1996	[6]
OLSR	2	3	3	4	2	4	3	2	4	1998	[7]
CGSR	3	3	3	3	3	3	3	3	4	1997	[8]
GSR	3	3	3	4	2	3	3	3	3	1998	[4]
FSR	3	3	3	3	2	3	3	3	3	1999	[9]
AODV	3	2	3	3	3	2	2	3	3	1999	[10]
AOMDV	3	2	3	3	3	3	2	3	3	2001	[11]
DSR	3	2	3	2	3	3	2	4	3	1998	[12]
TORA	4	3	3	3	3	2	3	3	4	1997	[13]
CBRP	3	2	2	2	3	3	2	3	3	1999	[15]
CQBR	3	2	3	3	3	2	2	4	3	2014	[16]
LUNAR	2	2	2	2	3	2	2	3	3	2014	[17]
DYMO	3	2	2	2	2	2	2	3	2	2005	[19]
E2DYMO	3	2	2	2	3	2	2	3	1	2013	[20]
ENDYMO	3	1	2	2	3	1	1	3	2	2015	[21]
ZRP	3	3	3	3	2	3	3	3	3	1997	[22]
ZHLS	2	3	3	2	2	3	3	2	3	1999	[23]
THRP	2	2	2	2	2	1	2	2	2	2007	[24]
AntHocNet	3	1	3	3	3	2	1	3	3	2014	[25]
ZCG	3	3	3	3	2	2	3	3	2	2015	[26]

From Table 1, it is possible to notice that the performance generally improved from the first generation of MANETs routing protocols (DSDV, GSR, AODV, etc.) over the time. Overall best performances are achieved by reactive and hybrid protocols especially in terms of packet delivery ratio (PDR), network throughput (NThr), routing overhead (RO) and power consumption (Pow). The protocols that perform best seem to be: DYMO and its evolutions among the reactive protocols, and THRP among the hybrid protocols.

In Table 2, protocols' evaluation takes into account the marks assigned to protocols in Table 1. Among these protocols, LAR and its evolutions have remarkable overall performances. DYMOselfwd seems to have the best performance in terms of RO and NThr; LSR performs better in terms of RO as well and for the normalized routing load (NRL) while AntMANET seems to behave better than the others for the end-to-end delay (e2eDel).

On average, location-based protocols get a better mark compared to regular ones, especially for what concern NThr, Pow and RO, thanks to the possibility of reducing useless flooding of packets.

Among all these location-based routing protocols, only the authors of LAR take into account the position error, running two simulations with and without error. Authors of DYMOselfwd [20] mention the existence of a position error but they do not provide simulation results.

Table 2: Location-based routing protocols comparison

Name	e2eDel	PDR	NLR	RO	AvgHops	NThr	PL	RDD	Pow	Year	Ref
DREAM	2	3	3	3	2	3	3	1	3	1998	[27]
LBRP	2	3	2	3	2	3	3	1	3	2011	[28]
GPSR	2	3	2	2	1	3	3	3	3	2000	[29]
GEOTORA	3	3	2	2	3	2	2	3	3	2000	[30]
LAR	3	2	2	2	3	2	3	3	2	2000	[32]
IOLAR	2	2	2	2	3	1	2	2	2	2014	[33]
PLAR	3	2	1	1	3	2	3	3	2	2013	[34]
AODV-LAR	3	2	2	2	3	2	3	3	2	2012	[35]
AODV-Line	3	2	2	2	3	2	3	2	2	2012	[35]
DYMO-selfwd	3	2	2	1	2	1	2	3	2	2015	[19]
LBSPR	2	2	2	2	1	2	2	2	2	2010	[36]
LSR	3	2	1	1	3	2	2	3	2	2015	[37]
AntMANET	1	2	3	3	2	2	2	2	2	2015	[38]
IHLAR	2	2	3	2	1	2	3	3	3	2011	[2]

4. Routing protocols implemented

Among all the protocols presented in Chapter 3, three of them have been selected for testing through simulation in this study.

Two of these protocols are DYMO-selfwd and AODV-Line, both location-based protocols chosen thanks to their good performances, especially in RO and PDR. In order to evaluate the difference with the not position-based version of DYMO-selfwd, the standard DYMO has been chosen as third protocol to simulate.

In the following, a more detailed explanation of the behaviour of these three routing protocols is provided.

4.1. DYMO (AODV-v2)

The Dynamic MANET On-demand (AODVv2) routing protocol is a routing protocol for MANETs Networks currently named AODV-v2 and defined in an Internet-Draft [40]. The version of the protocol described here is in the Internet-Draft 24 [41].

DYMO is a protocol that determines unicast route in a network in an on-demand fashion. An on-demand or reactive protocol finds a route just when required from the application layer.

The protocol consists of two operations that are route discovery and route maintenance. Every node has its own routing table that is made of the following entries: *Destination Address, Sequence Number, Hop Count, Next Hop Address, Next Hop Interface, Is Gateway, Prefix, Valid Timeout, Delete Timeout*.

When a node *S* needs a route to communicate to node *D* and it doesn't have a valid one in its routing table it starts a Route Discovery. It creates a RREQ, route request, and broadcasts it in the network. Every RREQ is re-broadcasted by other nodes, which elaborate the information, update its routing table and append its information to the RREQ before retransmitting it. If the RREQ reaches the destination *D* an answer called RREP, route reply message, is unicasted back to the source node *S*. At this moment a route is built. If no RREP arrives to the source node *S* until a RREQ_WAIT_TIME, the node can create a new RREQ.

In order to ensure stability to a communication even when the topology changes, the process of route maintenance is performed. Every node has to check the validity time of the stored routes and update routes status.

A node creates a RRER message if it receives a data packet whose final destination has a not valid route and adds in the message all the entries from the routing table that are linked to the unreachable node. Than the RRER is broadcasted and every node that receives it update its routing table and then re-broadcast it, if necessary, or just drop the packet.

4.2. DYMO-selfwd

The protocol DYMO-selfwd is a modification of DYMO protocol in fact it gets advantage of using position information.

When a node starts a route discovery it creates a RREQ with two more fields, transmitter position and receiver position. Every intermediate node first checks if its distance to the destination is less than the distance between the previous node to the destination, if it is bigger it drops the packet, otherwise it calculates a time to wait, that depends on of the distance to the destination, before retransmitting the RREQ packet.

Figure 4 helps to explain how the delay is calculated. The node that has sent the RREQ packet is *pn* (previous node), *d* is the final destination while *n1* and *n2* are two intermediate nodes that can retransmit the RREQ. The red segment is the radius of transmission and is the max possible transmission range. The segment red-yellow will be called *s1*, the green line will be called *d1* and it is the distance between node *n1* to the destination, and the distance of the previous node from the destination will be called *d_pn*.

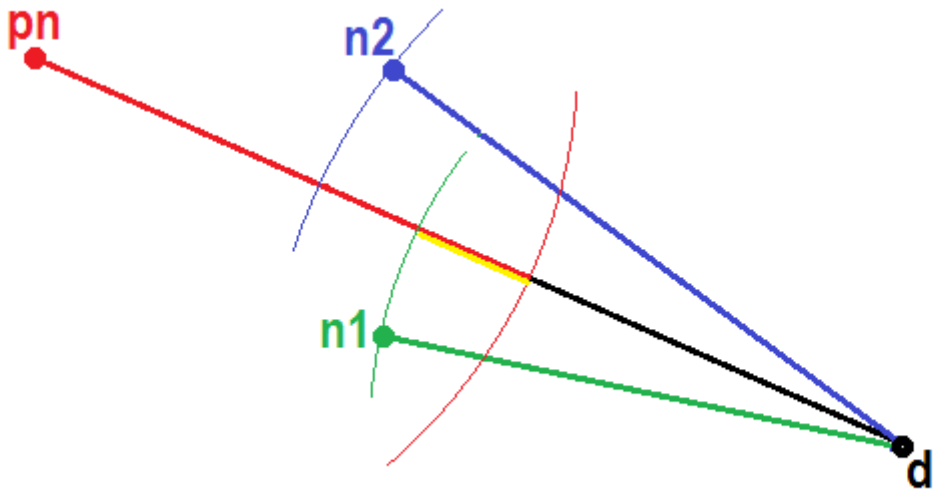

Figure 4: Explanation of calculation of retransmitting delay

Knowing that and having all necessary data, it is possible to calculate *s1*:

$$s_1 = d_1 + maxrange - d_{pn} \qquad \text{Equation 5}$$

s2 is calculable in the same way of *s1.*

Then *s1* is normalized with the max range:

$$s^\circ = \frac{s_1}{maxrange} \qquad \text{Equation 6}$$

Then, after defining a minimum delay to wait, the final delay is calculable in the following way:

$$delay_{fin} = delay_m * \exp(s^\circ) \qquad \textbf{Equation 7}$$

The node with the minimum delay among all nodes is the best selected node and will transmit for first, of course. All the other nodes that listen to the best node retransmitting the packet will drop their one.

4.3. AODV-Line

AODV-Line routing protocol is a modification of original AODV routing protocol. In AODV protocol, when a node needs a route to a destination it starts a route discovery creating a RREQ message that is then flooded by other nodes until it reach the destination that creates a RREP message. The RREP is unicasted back to source.

Route maintenance is done through Hello messages that are used by nodes to communicate to their neighbours of their existence. In fact, if node B sends to node A an Hello message, it refreshes all entries of its routing table, but If node A does not receive by node B any Hello messages (or other messages), within an Expunge timer, node A assumes that node B is unreachable, and set as not valid all the routes in which B is the next hop.

A node creates a RRER, in the same manners described in DYMO routing protocol.

AODV-Line routing protocol limits the excessive flooding caused by RREQ and RREP reducing the area of re-broadcasting these messages.

The area in which is possible to flood a RREQ is a rectangle equal to the distance of the source to the destination, multiplied by *W*, a value called *window*.

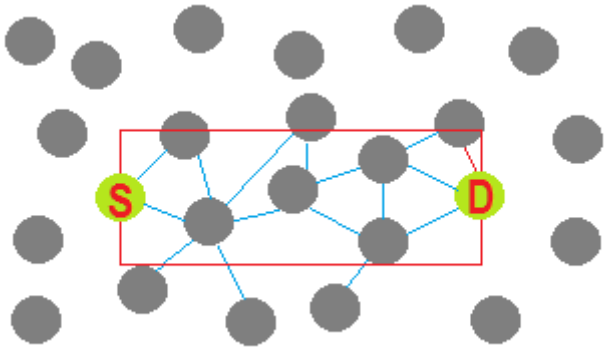

Figure 5: Restricted flooding area in AODV-Line

The value *W* depends on the transmission range *T* and of the density of nodes *σ* and is equal to:

$$W = \frac{1}{2*T*\sigma}$$ **Equation 8**

Only nodes inside the rectangle can retransmit the RREQ.

In order to allow nodes to do this calculation, RREQ and Hello messages are modified adding *Location information* and *timestamp* field for source and destination in RREQ and just for the source in Hello messages.

Location information contained in RREQ and Hello messages are also stored in a Location table to be used again later. The Location table has three entries: *Node Address, Node Location, Timestamp* that are updated every time a node receives information that are newer.

5. Simulation tools

Simulation tools are a fundamental key in developing new technologies or techniques. Such a simulation imitates some system behaviour without a real implementation, with a consequent substantial benefit in terms of economic resources and time saving.

It can be used for already existing systems to test them in particular scenarios, allowing us to explore new procedures without disrupting the real system, to compare different solutions and to elect the best one. In general, simulating can answer to the question "*What if?*".

Nowadays, there are several simulation tools in networking field and the main ones are NS-2, NS-3, OPNET and OMNET++.

5.1. Main simulation tools

Ns-2 is a discrete event simulator for networking systems born as a variant of Real World Simulator in in 1989. From 2010 it is not maintained and it is not accepted for publications anymore but statistics says about 70% education purposes use ns2 [42]. Ns-2 code comprises OTCL and C++, linked to achieve efficiency. Running the simulation will produce two outputs or two files namely "trace file" and "namfile". Trace file defines the event discreet simulators and can be given input to a new scenario file called NS- VISUAL TRACE ANALYZER, while Nam file is a visual graphical window [42].

Ns-3 is a new software development effort focused on improving the core architecture, software integration, models, and educational components of ns-2 but it has been written from scratch so it is not based on Ns-2. The project started in July 2006 and the first release was made on June 30, 2008 [43].

OPNET is a high level event-based network level simulation tool that supports four simulation technologies as: *Discrete Event Simulator, Flow Analysis, ACE Quick predict* and *Hybrid Simulation* [42].

OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators. It offers an Eclipse-based IDE, a graphical runtime environment, and a host of other tools. There are extensions for real-time simulation, network emulation, database integration, SystemC integration, and several other functions [44]. OMNeT++ has a commercial edition called OMNesT.

5.2. State of the art of already available implementations

Another key factor in the selection of the most suitable simulator to work with was the presence or not of already available implementations. A trustable implementation of protocols of interests have been searched:

- The DYMOselfwd and AODV-Line protocols were not available online as downloadable project but it was possible to find DYMO and AODV protocols.
- DYMO protocol was available for Ns-2 on *Surgeforce.net* [45] but a version made for Ns-3 seems not to be available for download. Instead, DYMO is one of the already implemented protocols in the INET Framework of OMNET++.

- A downloadable implemented version of AODV protocol can be easily found online for Ns-2 and for Ns-3 [46] too. AODV protocol is also one of the already implemented protocols in the INET Framework of OMNET++.

For all the above, we decided to work with OMNET++.

5.3. OMNET++ simulator

OMNeT++ is an object-oriented modular discrete event network simulation framework available for the most common operating systems (Linux, the MAC module OS/X, Windows). It has a generic architecture, so it is not a simulator of anything concrete, but rather provides infrastructure and tools for writing simulations [47].

The main advantage of OMNeT++ is its modularity, in fact Models are built using different reusable modules. Modules can communicate each other with gates (or ports) through Message Passing. Simple modules are programmed in C++ and are at the lowest level of module hierarchy. Modules can also use parameters that allow to easily customize module behaviour and topology. Simple modules can be grouped to form compound modules, and so on; for this reason, OMNeT++ itself is considered a compound module [47].

Simulations can be run with a graphical interface (Tkenv), that allows the user to see messages sent and nodes movements, or by command line.

An OMNeT++ model consists of the following parts [47]:

- NED language topology description(s) (.ned files). A file written in a high level language (NED) in which the topology of the network is described;
- Message definitions (.msg files). A file that defines messages types and their structure;
- Simple module sources. They are C++ files, with .h/.cc suffix.

A simulation produces a vectorial file and a scalar file in which simulation statistics are collected depending on their nature. In addition, OMNeT++ provides a user friendly IDE for data analysis.

5.3.1. The INET framework

The INET framework is an open source library for OMNeT++ that contains models for the Internet stack (TCP, UDP, IPv4, IPv6, OSPF, BGP, etc.), wired and wireless link layer protocols (Ethernet, PPP, IEEE 802.11, etc), support for mobility, MANET protocols, DiffServ, MPLS with LDP and RSVP-TE signalling, several application models, and many other protocols and components [48].

The INET framework work with the same logic of OMNeT++ so it is based on different modules that communicate each other. For every communication layers of the ISO/OSI there are several options that give to the user enough freedom in testing the most various scenarios.

Hosts and routers in the INET Framework are OMNeT++ compound modules that are composed of [49]: *Interface Table,* module that contains the possible interfaces (eth0, wlan0, etc.)*; Routing Table,* module that contains the IPv4 module or IPv6 routing tables and that is accessed by other modules for adding or replacing a route of finding the best one*; Mobility Module,* module that is responsible for nodes movement and carries nodes positions*; NICs,* a compound module made of a queue and a the MAC module*; Network Layer,* module that represents protocols of the network layer*; Transport Layer Protocols,* that are represented by modules connected to the network layer(Currently available: TCP, UDP, SCTP)*; Application,* Application modules typically connect to TCP and/or UDP, and model the user behaviour as well as the application program (e.g.

browser) and application layer protocol (e.g. HTTP*); Routing Protocols,* this modules the routing protocols*; MPLS Modules,* Additional modules are needed for MPLS simulations; *Relay Unit,* that could be contained in Ethernet switch modules*.*

6. Implementation

In this section the implementation of every routing protocol and of other layers will be explained. The flow of the overall procedure is illustrated in Figure 6.

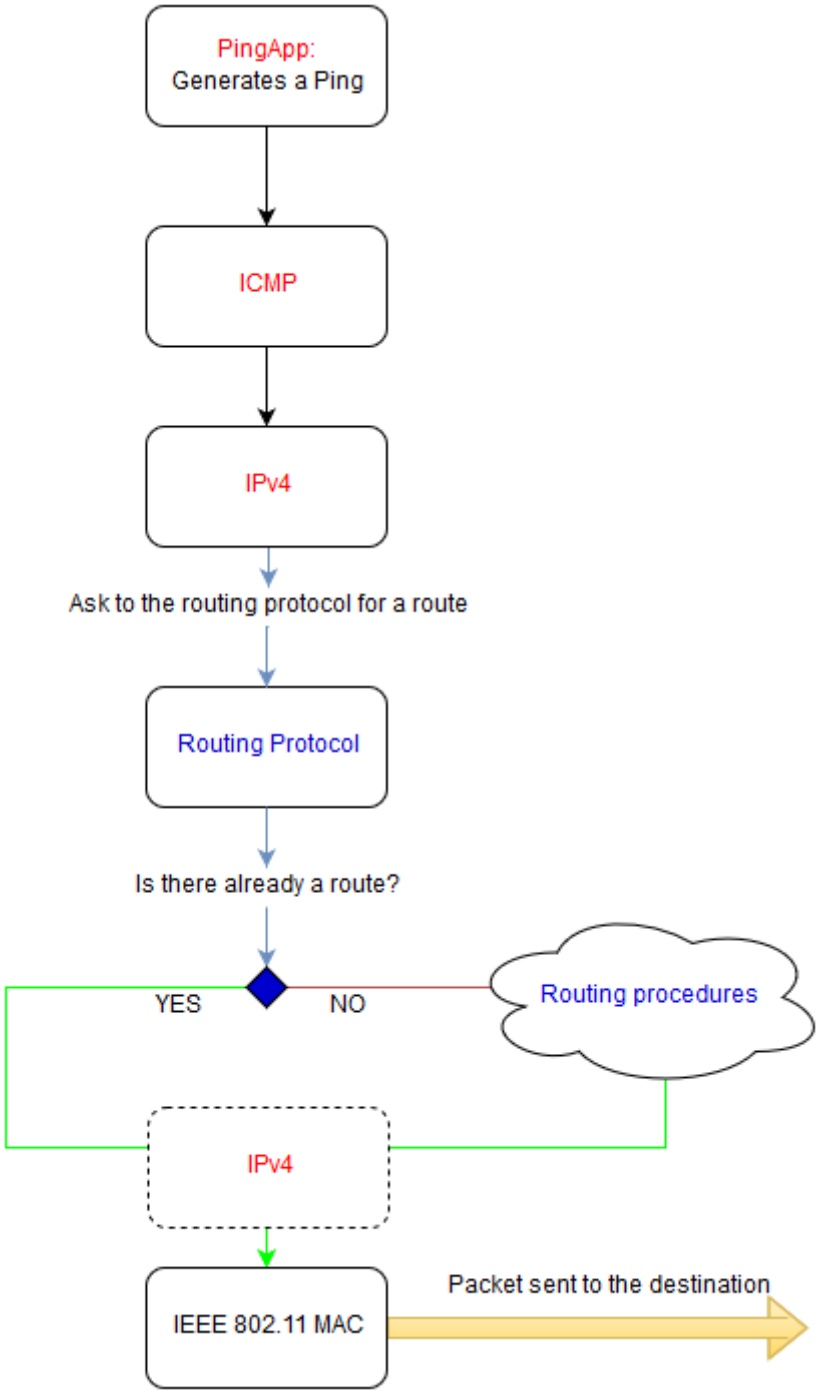

Figure 6: Flow of sending a data packet process

The Pingapp module generates a Ping and sends the packet to the ICMP module that sends it to the IPv4 module, which asks for a route to the routing protocol. If it has already a route, the routing protocol module sends the route back to the IPv4 module; if not, it first starts for routing procedures and then gives the answer to the IPv4 module. When a route is available, the IPv4 module sends the packet to the lower level that sends it to the destination.

The routing procedure depends of course on the routing protocol used. However, the behaviour of the system can be simplified, as shown in Figure 7.

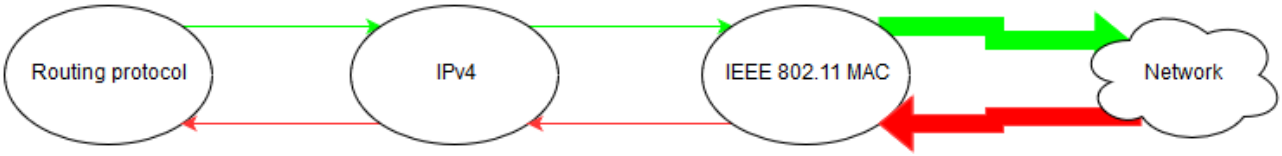

Figure 7: General internal flow in routing procedures

Every time the Routing protocol module has to send a packet (e.g. RREQ) to other hosts, it first sends it to the IPv4 module that relays to the MAC module layer.

6.1. DYMO

The DYMO protocol was already implemented in the INET Framework. For this reason, in this section there will be a brief explanation of how the protocol was implemented and of the modifications done to collect the statistics.

In DYMO when a node wants to start a route discovery (

Figure 8) it creates a RREQ and sends it to the other nodes. At the same time, the node schedules a *RREQwaitRREPTimer.* If a RREP arrives before the time expires, the route discovery is completed and the timer is deleted; if not, a *RREQBackoffTimer* is scheduled and the route discovery is restarted when the self-message (*RREQBackoffTimer*) arrives. There is a condition to check before creating a *RREQBackoffTimer*: if the number of attempts is equal to the maximum allowed minus one, then the timer scheduled is different and it is called *RREQHoldDownTimer*. When this last timer expires, if there is a delayed datagram, the route discovery is retrieved once again.

When a node receives a RREQ, it first updates its routing table, and then it checks whether it is the destination. If the node is just an intermediate node, it broadcast the RREQ. Otherwise, if the node is the final destination of the route discovery, it creates a RREP and sends it back to the node that just sent the RREQ. Every node that receives the RREP, checks in the routing table for the next hop to the destination (originally, the source of the source discovery) and forwards the packet.

Every time a node finds out that a link is broken it creates and broadcasts a RERR packet. Every node that receives the RERR re-broadcasts it.

For what concern statistics, several metrics have been implemented. Values as total number or RREQ, or of RREP or of RERR were considered as scalar quantities and increased every time was necessary while values as route life time and route discovery delay have been considered as vectors of values.

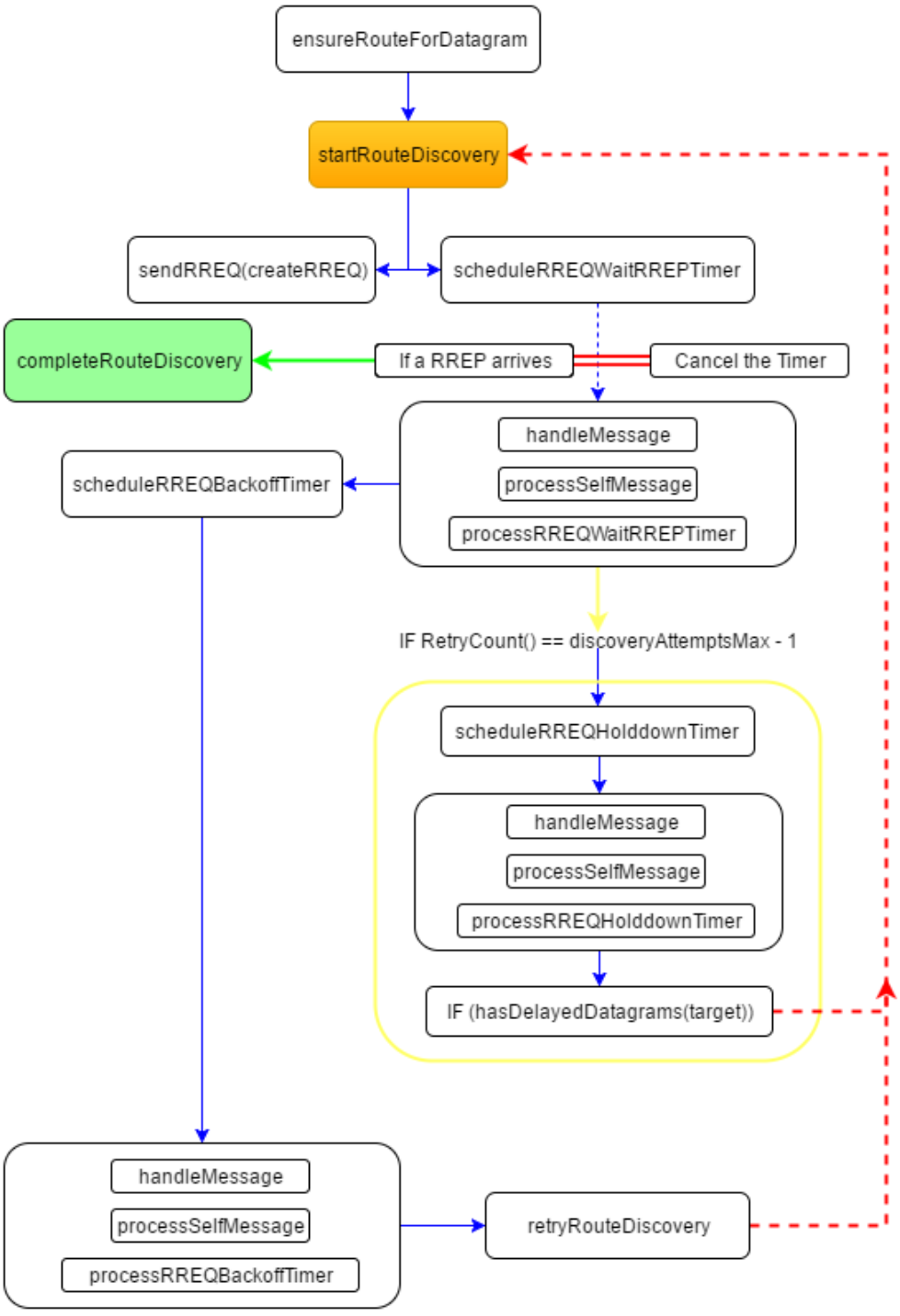

Figure 8: Route discovery in DYMO

6.2. DYMO-selfwd

The protocol DYMO-selfwd has been implemented patching the protocol DYMO, exposed in Section 6.1, and all the information already provided is still valid.

A first modification has been done to the dymo.msg file, adding two fields to the *RREQ* packet, *Coord destinationPosition* and *Coord sourcePosition*, and creating a *RREQDelayed* packet as shown in Figure 9.

These fields are important because they allow to carry the position information in the packets sent in the network.

```
88
89 //
90 // DYMOselfwd RREQ packet
91 //
92 packet RREQ extends RteMsg {
93     Coord destinationPosition; //patch
94     Coord sourcePosition;      //patch
95 }
96
97 packet RREQDelayed extends RREQ{
98     //patch
99 }
100
```

Figure 9: RREQ Packet in DYMOselfwd

Nodes position information is stored in a position table defined in gpsr protocol and this table is maintained refreshing position information every time a *self-message* called *gpsTimer* arrives (at every arrival the message is re-scheduled too).

In the patched version of *DYMOselfwd::processRREQ(RREQ *rreqIncoming)* there is a first check inherent in nodes positions in order to discard the incoming RREQ if *this* node is farther than the one that sent the packet, and, if so, to also delete a *RREQDelayed* packet, waiting to be sent where it exists. If this node is nearer to the destination than the one that sent the packet, instead to create and send a RREQ, it creates a RREQDelayed and it sends it to itself after a time equal to the one calculated in $\boldsymbol{delay_{fin} = delay_m * \exp(s°)}$ Equation 7 and the key of the RREQ is saved in a map.

Every time a RREQDelayed arrives it is casted into a RREQ packet. The RREQ packet is than normally sent.

6.3. AODV-Line

AODV-Line is implemented as patch of AODV protocol and the overall behaviour is not different from the original one. The only modification is about the rules for retransmission and in managing the position table.

Figure 10 shows how a source performs the route discovery. It first creates and sends a RREQ packet and in the meantime it schedules the max time to wait for a RREP, through sending a *self-message rrepTimermsg.*

If a RREP arrives within this time, the route discovery is completed, else, the route discovery is performed again.

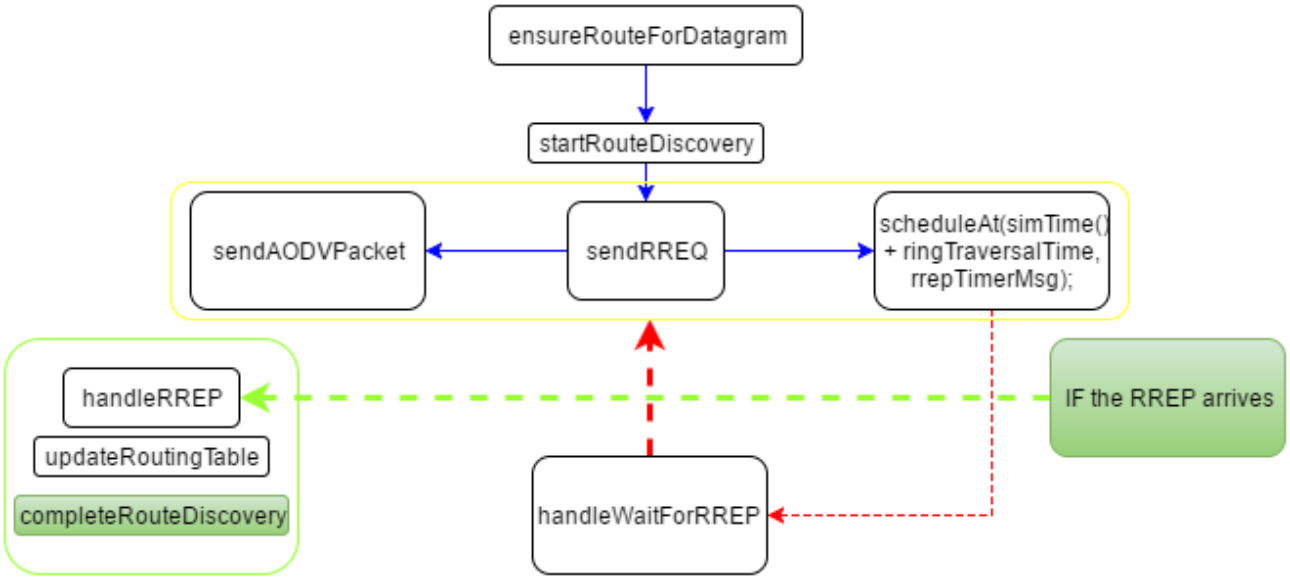

Figure 10: Route discovery in AODV-Line at the source

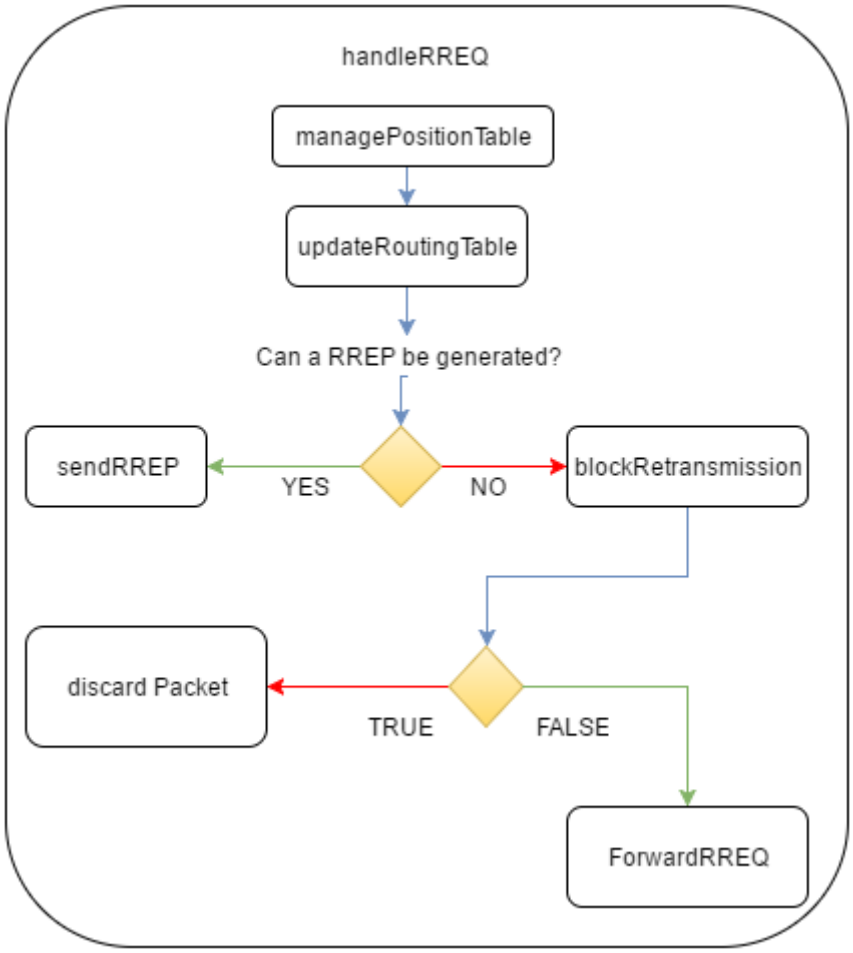

Figure 11: handleRREQ - AODV-Line

When a node receive a RREQ (Figure 11) it can, after checking the possibility of sending a RREP, decide to forward or not the RREQ packet. This decision is made according to the principle of the area of retransmission illustrated in Figure 5 and this duty is on *AODVRouting::blockRetransmission* (Figure 12).

```
bool AODVRouting::blockRetransmission(Coord sourcePosition, Coord destPosition, Coord selfPosition) {

    float s_i = (sourcePosition - selfPosition).length(); //distance source node - intermediate node
    float s_d = (sourcePosition - destPosition).length(); //distance source node - destination node
    float i_d = (selfPosition - destPosition).length(); //distance intermediate node - destination node

    double alfa = acos( ( (s_i*s_i) + (s_d*s_d) - (i_d*i_d) ) / (2*s_i*s_d) );
    // check: if alfa > 90° then the intermediate node don't have to retransmit (aodv-line)
    if(alfa > 1.5708)
        return true;

    double beta = acos( ( (i_d*i_d) + (s_d*s_d) - (s_i*s_i) ) / (2*i_d*s_d) );
    // check: if beta > 90° then the intermediate node don't have to retransmit (aodv-line)
    if(beta > 1.5708)
        return true;

    float height = s_i* sin(alfa);
    // check: if the height > than aodv-line window for retransmission, we've to discard the packet
    if(height > window)
        return true;

    return false;
}
```

Figure 12: AODVRouting::blockRetransmission - AODV-Line

Every created and sent RREP is then unicasted back by every node in the path until it reaches the original source. Instead, when a node finds out that a link is broken, or a node is unreachable, it creates a RRER and broadcast it. Route maintenance is realized even through using HelloMessages. These messages are different from the original AODV ones because, together with RREQs, they are responsible to carry position information, and respective timestamps, that are then elaborated and used to update the position table.

6.4. Other layers

6.4.1. The ICMP module

The ICMP module has been modified to collect some statistics. The first one is the end-to-end delay and the number of data packets received (for every source).

The reason why the e2eDel has not been implemented in the Pingapp module but in the ICMP module is because when a Ping is created and sent, it does not arrive until the application layer of the destination but it is received by the ICMP module, as illustrated in Figure 13, that then creates a Pong. This Pong will arrive to the Pingapp module of the original source, allowing the source to calculate the Round Trip Time.

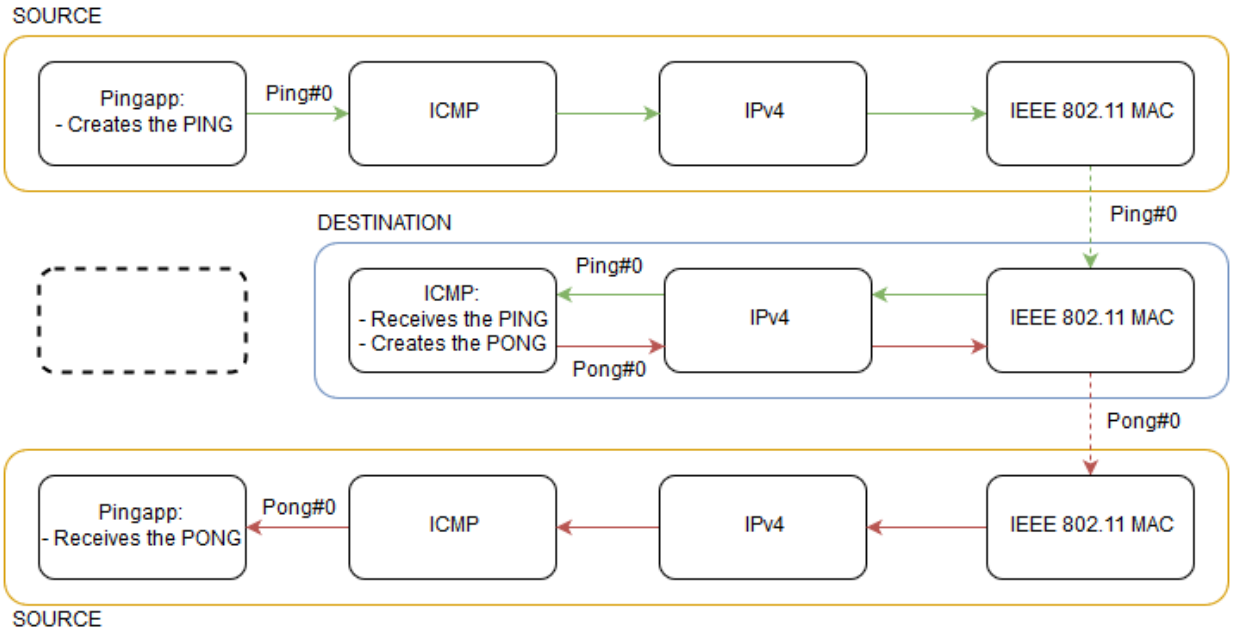

Figure 13: Ping-Pong Transmission

6.4.2. The IPv4 module

The IPv4 module is a complex module with several duties in the overall system work. A simplified representation of Pings handling is illustrated in Figure 14. A modification has been done in *reassembleAndDeliverFinish*, in which for every Ping that arrives a value called hopCount is incremented of the number of hops that every Ping has done. This value is then divided by the number of Pings received (*numLocalDeliver*), allowing to obtain the average number of hops per ping.

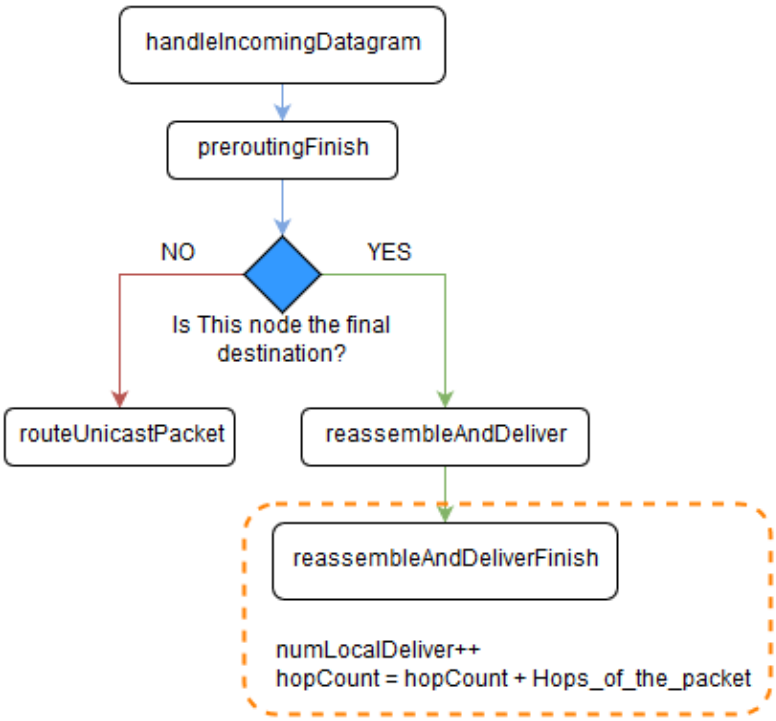

Figure 14: IPv4 - Pings handling

7. Simulation procedure and scenarios

7.1. Procedure followed for the assessment

To carry out the set of simulations of DYMO, DYMOselfwd and AODV-Line we have to follow some necessary steps:

- Decide which scenarios to simulate. For our goals it was interesting to test the protocols varying the number of nodes (40, 70 and 100) keeping the area fixed (1000m*1000m), varying the payload of the application layer (56 bytes, 500 bytes, 1472 bytes) in both pedestrian and vehicular speeds.
- The .ini file has been created with an external program written for these scopes: every node can be a source with the 30% of probability to any other node and the Seeds for the random number generators were randomly generated.
- Every configuration for every protocol was run for five times.
- When simulations finished, all data were extracted with a *.sh* script.
- All extracted data were then analysed with Matlab.

7.2. Metrics provided

Routing protocols testing passed through evaluation of several metrics. Metrics have been chosen as the ones that appear in Literature to be the most relevant or the most used, in order to give the most complete overview as possible about protocols performances to the reader.

Metrics provided are:

- End to end delay (e2eDel)
- Packet delivery ratio (PDR)
- Normalized routing load (NRL)
- Route discovery delay (RDD)
- Route life time (rlt)
- Round trip time (rtt)
- Routing Overhead (RO)
- Average number of hops (AvgHops)

A deep explanation of these metrics has already been addressed in the section 3.1 (Metrics).

7.3. Simulated scenarios

The scenarios simulated were 18 as result of combination of some key configuration features: the number of nodes, the payload of the Ping and the nodes' speed.

The area considered is a square of 1000m*1000m and is fixed. For this reason, changing the number of nodes allows to change their density. Another factor used to decide the number of nodes is the maximum transmission range (e.g. 250m in this study); under the assumption of uniform distribution of the nodes, their number should guarantee connectivity in the network. Thus, a minimum of 25 nodes are required to fully cover the area; however, as nodes start to move, the topology changes and the probability of disconnections in the mesh is really high. Specific simulations have been run in order to find a good trade-off between the minimum number of nodes that guarantees full coverage in the whole area and users' mobility; 40 nodes has been found to be a good compromise to guarantee a fully connected mesh for most of the time even when In this work, three users' densities have been considered to estimate the scalability of the three protocols: 1) 40 nodes for normal density; 2) 70 nodes for high density; and 3) 100 nodes for very high density.

In order to test the ability of the routing protocols to route packets in a more congested network, three values for the ping payload have been simulated: 56, 500 and 1472 bytes.

Nodes that generate traffic are chosen randomly. Every node has a chance of the 30% to be a traffic generator.

Finally, we considered that the nodes move according to the Gauss Markov mobility model, in which temporal dependency is fundamental in determining the mobility behaviour [50]. As such, the randomness of the users' movement can be tuned through specific parameters, thus allowing different users' behaviours compared to other mobility models that are commonly used in the literature (i.e., the Random Waypoint). Initially, each node is assigned a current speed and direction. At fixed intervals of time the movements are simulated by updating the speed (*s*) and the direction (*d*) of each node. The next location is computed based on the current location, speed, and direction of movement according to the following equations:

$$s_i = \alpha\, s_{i-1} + (1-\alpha)\mu_s + \sqrt{(1-\alpha^2)} \cdot x_{i-1},$$
$$d_i = \alpha\, d_{i-1} + (1-\alpha)\mu_d + \sqrt{(1-\alpha^2)} \cdot y_{i-1},$$

Equation 9

where s_i and d_i are the new speed and direction of the node at time interval *i*. α (0≤α≤1) is the tuning parameter through which modelling different mobility patterns: as α approaches zero, a drifting Random Walk is obtained, while with α=1, linear motion is generated. μ_s and $\mu_d$are constants representing the asymptotic mean value of speed and direction as $i \to \infty$; and x_{i-1} and y_{i-1} are independent, uncorrelated, and stationary Gaussian processes with a mean of zero and a standard deviation equal to the asymptotic standard deviation of speed and direction as $i \to \infty$. Figure 15 shows an example of configuration in the omnetpp.ini file; in the "mobility" section, the parameters for the Gauss Markov model can be set.

In order to assess the ability of the protocols to deal with the mobility of the nodes, two mobility scenarios have been considered: one with nodes moving as pedestrian (calculated as a truncnormal with mean equal to 1mps and standard deviation equal to 0.3mps), and one with nodes moving at higher speeds typical of vehicular networks (calculated as a truncnormal with mean equal to 20mps and standard deviation equal to 6mps).

```
[General]
network = DYMONetwork
sim-time-limit = 2000s

# simulation area
**.mobility.constraintAreaMinZ = 0m
**.mobility.constraintAreaMaxZ = 0m
**.mobility.constraintAreaMinX = 0m
**.mobility.constraintAreaMinY = 0m
**.mobility.constraintAreaMaxX = 1000m
**.mobility.constraintAreaMaxY = 1000m

# mobility
**.host[*].mobilityType = "GaussMarkovMobility"
**.host[*].mobility.angle = uniform(0deg,360deg)
**.host[*].mobility.alpha = 0.6
**.host[*].mobility.margin = 1m
**.host[*].mobility.variance = 0.325
**.host[*].mobility.speed = truncnormal(1mps, 0.3mps)

# ping app
*.host[*].numPingApps = 1
*.host[*].pingApp[0].startTime = uniform(0s,5s)

# nic settings
**.wlan[*].bitrate = 54Mbps
**.wlan[*].mgmt.frameCapacity = 10
**.wlan[*].mac.address = "auto"
**.wlan[*].mac.maxQueueSize = 14
**.wlan[*].mac.rtsThresholdBytes = 3000B
**.wlan[*].mac.retryLimit = 7
**.wlan[*].mac.cwMinData = 7
**.wlan[*].mac.cwMinMulticast = 31
**.wlan[*].radio.transmitter.power = 2mW
```

Figure 15: Setting example in omnetpp.ini

The simulation time is set to 2000 seconds to allow users to move in the area and gather the average behaviour. Also, every configuration is repeated 5 times, using different random seeds, for a total of 90 simulations run for every protocol. The results presented in Chapter 8 are averaged over the 5 repetitions.

For technical reasons it has not been possible to simulate the original DYMO protocol in a 100 nodes scenarios.

Additionally, for DYMOselfwd and AODV-Line those simulations have been run twice considering the case in which the information of the position of every node has got an error.

8. Simulation results

In this section results of DYMO, DYMOselfwd and AODV-Line in some of the scenarios simulated are compared. In order avoid any confusion with possible names of configurations a little bit of taxonomy is necessary. A configuration name is organized as follows:

X_Y_Z

X is the number of nodes. It can be 40, 70 or 100.

Y is the payload of the ping. It can be 56, 500 or 1472bytes.

Z is the speed of nodes. It can be “ped” (pedestrian speed) or “veh” (vehicular speed).

Summarizing, the configurations taken into account are:

##	Configuration
01	40_56_ped
02	40_56_veh
03	40_500_ped
04	40_500_veh
05	40_1472_ped
06	40_1472_veh
07	70_56_ped
08	70_56_veh
09	70_500_ped
10	70_500_veh
11	70_1472_ped
12	70_1472_veh
13	100_56_ped
14	100_56_veh
15	100_500_ped
16	100_500_veh
17	100_1472_ped
18	100_1472_veh

Among these configurations, the 70_56_ped has been chosen as the reference one and results are provided in Section 8.1. The impact of the nodes density will be explored in Section 8.2., showing results in a scenario where a lower number of nodes are involved (i.e., 40) in the routing, thus facing the possibility of higher route failures; a scenario with a very high density of nodes (i.e., 100) is considered also, in order to assess the scalability of the three protocols. The impact of different users’ mobility is studied in Section 8.3., while Section 8.4 is devoted to the impact of different ping payloads. How an error in position information affects the routing performance is evaluated and explained in Section 8.5.

8.1. Configuration 70_56_ped

Figure 16 shows the performance of the three protocols in a scenario of 70 nodes that move as pedestrians and that transmit pings with a payload of 56B. DYMOselfwd outperforms the others in the packet delivery ratio (PDR), route lifetime (RLT) and route discovery delay (RDD). Also, considering that AODV-line is only able to deliver, on average, 32 % of the packets sent, we can also conclude that DYMOselfwd behaves better that DYMO in the end to end delay (e2eDel), round trip time (RTT), routing overhead (RO) and network throughput (NRL). Thus, DYMOselfwd is slightly worse than DYMO in the average number of hops (AvgHops) only; however, the numbers are very similar.

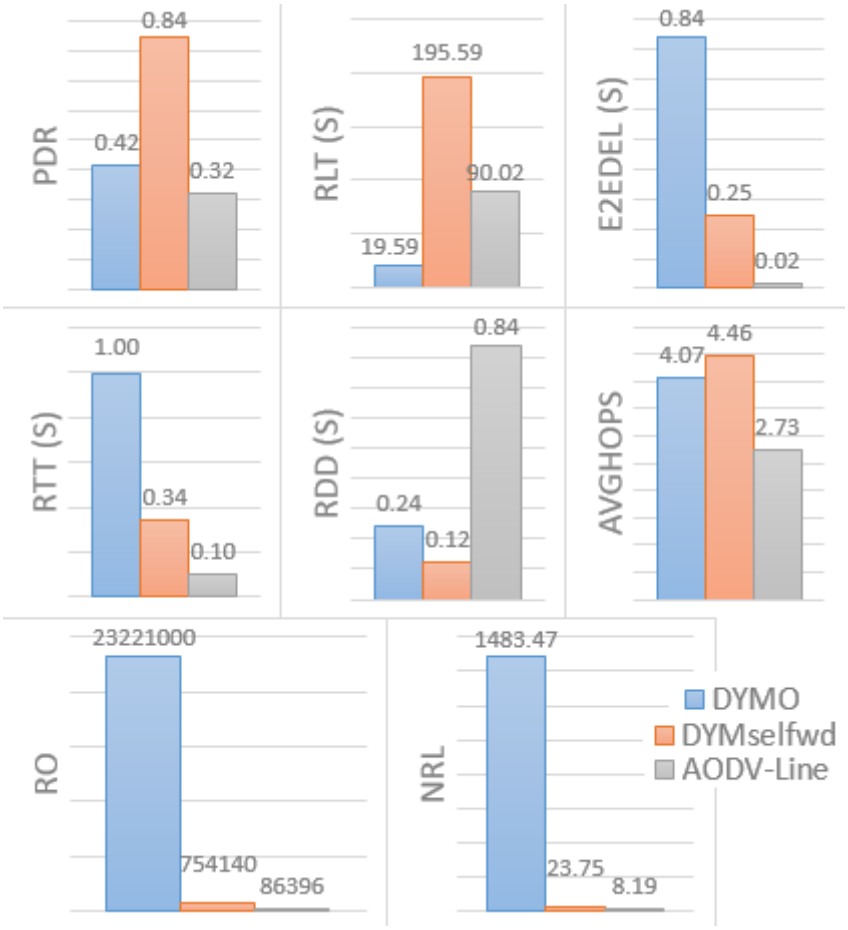

Figure 16: 70_56_ped performances

It is worth stressing here that, in the original DYMO, the nodes send 23,221,000 routing packets during 2,000 seconds that last the simulation, while with DYMOselfwd this amount reduces to 754,140. This amount is further decreased with AODV-line (86,396); however, this has a clear negative impact on the PDR, as the protocol seems to be too selective when deciding whether to retransmit a routing packet or not. This is also reflected in the NRL, which indicates the number of routing packets sent versus the number of data packets that have been received correctly. As the PDR for AODV-Line is just of 32% and the NRL is 8.19, most of the delivered packets have followed routes of really few hops (the average number of hops is 2.73), while the

protocol has not been able to find a route for further destinations. This also explains the low e2eDel and the high value of route discovery delay. Short range deliveries take also advantage of using Hello Messages. The high route discovery delay does not affect the average e2eDel, because the number of route discoveries is not high.

8.2. Impact of node density

Node density is a key factor that affects routing protocols performance in a relevant manner. Keeping the area of the region fixed (1km^2), increasing the number of nodes means that more traffic will be generated but, at the same time, the connectivity, in terms of absences of isolated nodes, should increase.

On the other hand, less nodes leads to less traffic, which means less route discoveries, and in general, less control packets, but it also means that the risk of low connectivity is higher. The effects of lower or higher users density is investigated in this section.

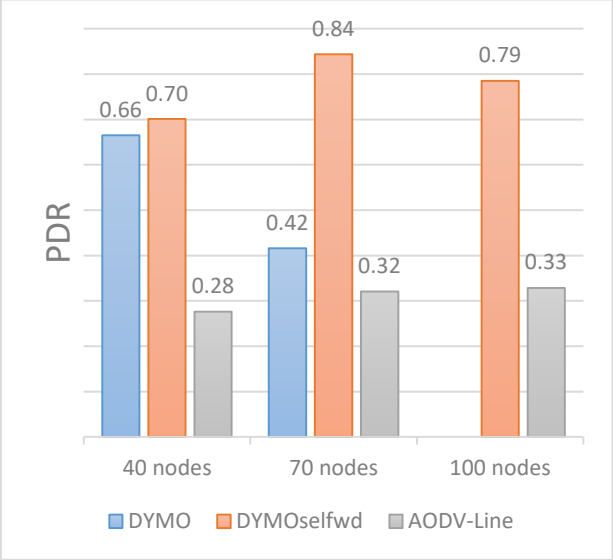

Figure 17: 40/100_56_ped - Packet delivery ratio

Figure 17 shows the packet delivery ratio for the three protocols and different user densities. With 40 nodes, DYMO delivers 66.5% of the packets, increasing its performance compared to the previous case (41%). Instead, DYMOselfwd and AODV-Line, get worse performance due to the reduced number of nodes. With 100 nodes, DYMOselfwd get worse performance due to the increased number of nodes that generate traffic. However, again the DYMOselfwd outperforms the others in terms of PDR. Again, the bad performance of the AODV-Line is due to the higher selectivity of nodes that retransmit packets that these two location-based routing protocol have. Less nodes in the network favours DYMO because it has a consistent number of control packets generated. A similar behaviour (i.e., high PDR for DYMOselfw and low for AODV-line) is found when the number of nodes is increased to 100; please note that the results for the DYMO protocol for 100 nodes cannot be displayed because the simulations were taking too long and it was not possible to get the results. This scalability problem should be fixed in the future and further studied.

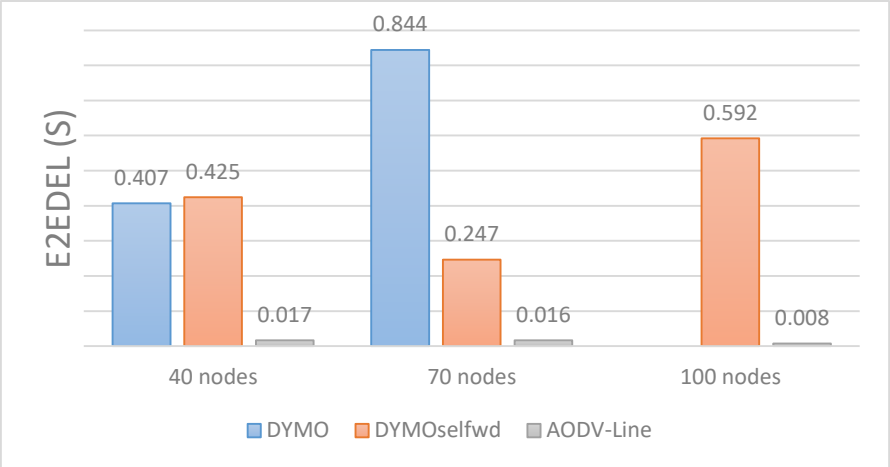

Figure 18: 40/70/100_56_ped - e2eDel

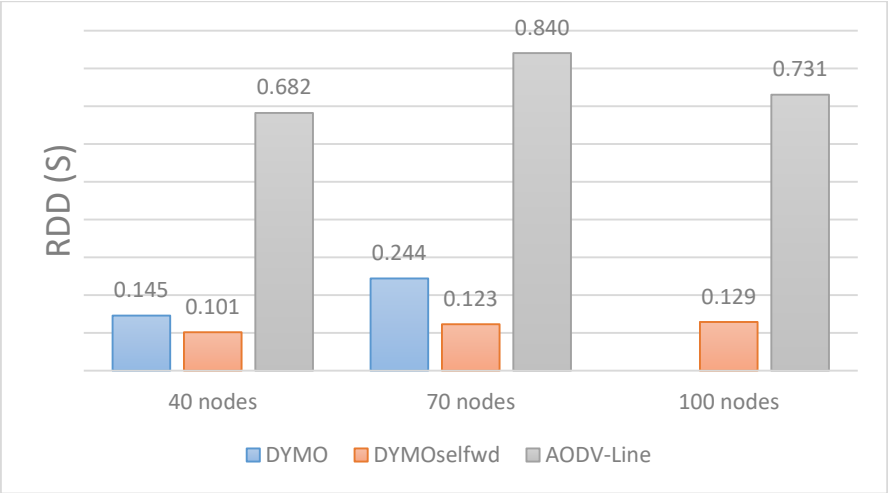

Figure 19: 40/70/100_56_ped – RDD

Regarding the end to end delay (see Figure 18), it reduces in a lower density scenario for DYMO, while for DYMOselfwd it increases. In the 100 nodes case, it further increases for DYMOselfwd. Instead, AODV-Line keeps being stable with almost the same value for 40 and 70 nodes but reduces it by the half for 100 nodes.

The route discovery delay, shown in Figure 19, decreases with 40 nodes for all three protocols. With 100 nodes, the RDD remains stable with DYMOselfwd while it decreases a bit with AODV-Line compared with the reference case. Again, the RDD for AODV-line is very high for all the scenarios, which is in line with the low PDR. Also, for DYMOselfwd, the RDD keeps stable with user density, while the e2eDel increases when reducing or increasing the density compared to the reference case. This behaviour can be explained by the low density of the network that leads to a low connection of the nodes, confirmed by the reduction of the packed delivery ratio and by the increasing of the routing overhead. For this reasons e2eDel and RTT increase in a noticeable way.

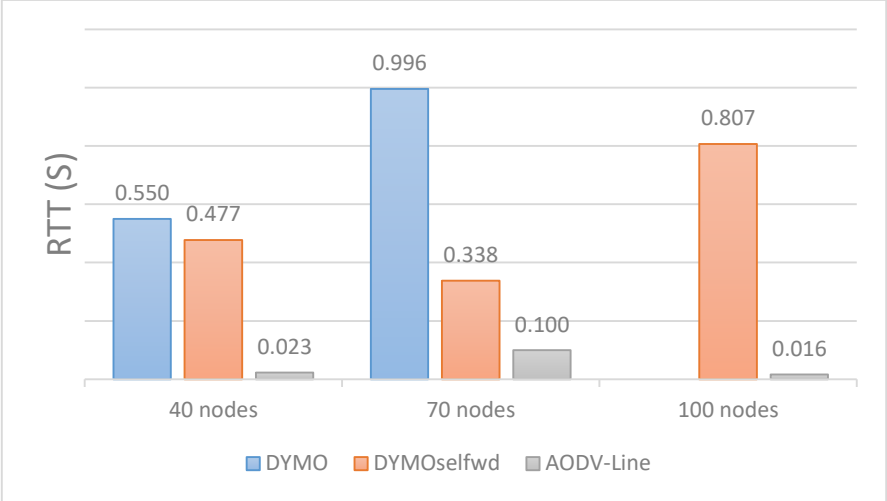

Figure 20: 40/70/100_56_ped - RTT

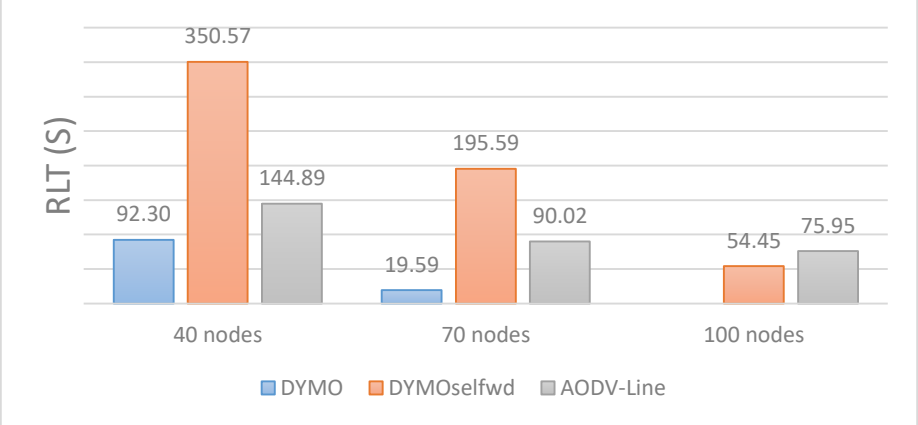

Figure 21: 40/70/100_56_ped - Route life time

As shown in Figure 20, the round trip time of DYMO and AODV-line decreases with a lower density of nodes, while for DYMOselfwd it increases. When a higher number of nodes is considered, the rtt is more than doubled for DYMOselfwd, while it decreases for AODV-Line.

The route lifetime has a common trend in all the three protocols as illustrated in Figure 21. It decreases as the number of nodes increases because with less nodes it is more probable that the same identical route is maintained. Also, DYMOselfwd outperforms the others in terms of RLT a part from the higher density scenario; however, results for AODV-line may be skewed by the low PDR it shows.

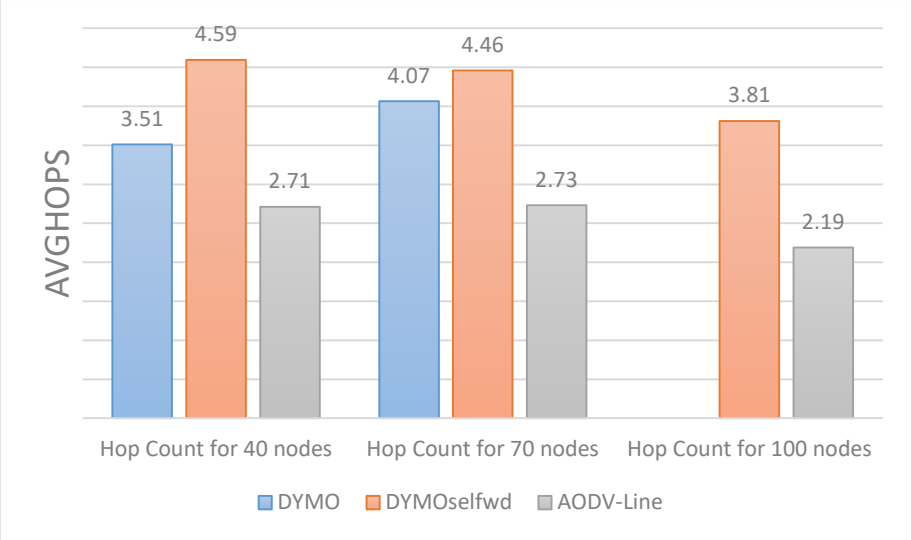

Figure 22: 40/70/100_56_ped – Average number of hops

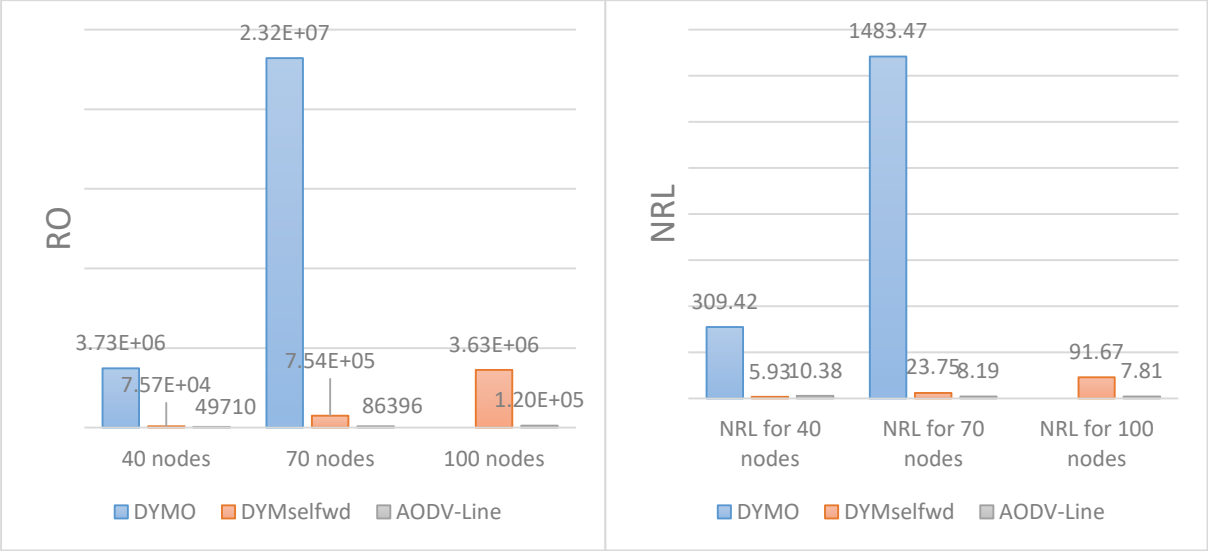

Figure 23: 40/70/100_56_ped - Routing overhead – Normalized routing load

The average number of hops is quite stable no matter the number of nodes in the area for the three protocols, as shown in Figure 22. The most remarkable differences are a decrease in DYMO protocol with respect to the reference scenario with 40 nodes, a decrease in DYMOselfwd and AODV-line with 100 nodes.

As expected, the routing overhead and the normalized routing load, shown in Figure 23, increase for all three protocols as the number of nodes increase. Again, it is worth noticing the huge difference between regular routing protocols (i.e., DYMO) and location-based protocols in terms of RO.

Again, we can conclude that DYMOselfwd seems to outperform the others in both lower and higher density scenarios, in terms of PDR, RDD, RLT, RO and NRL, while clearly having a higher average number of hops compared to the others.

8.3. Impact of node speed

Node speed can affect performances greatly and a vehicular speed leads to worse performances than with pedestrian speed, as the probability of having disconnections in the mesh increases.

The packet delivery ratio is quite significant, because it indicates the real possibility of communication. As such, for this metric we compare the behaviour with pedestrian and vehicular speeds for the three node densities presented in the previous section (40, 70 and 100). Figure 24 clearly shows the effect of a higher nodes' speed. As expected, the PDR for the vehicular scenario is always lower than in the pedestrian case. In the 100 nodes scenario the performance drop is even worse: despite the higher number of nodes (i.e., a higher possibility to have an alternative route) the nodes move faster, thus increasing the probability that the route gets lost.

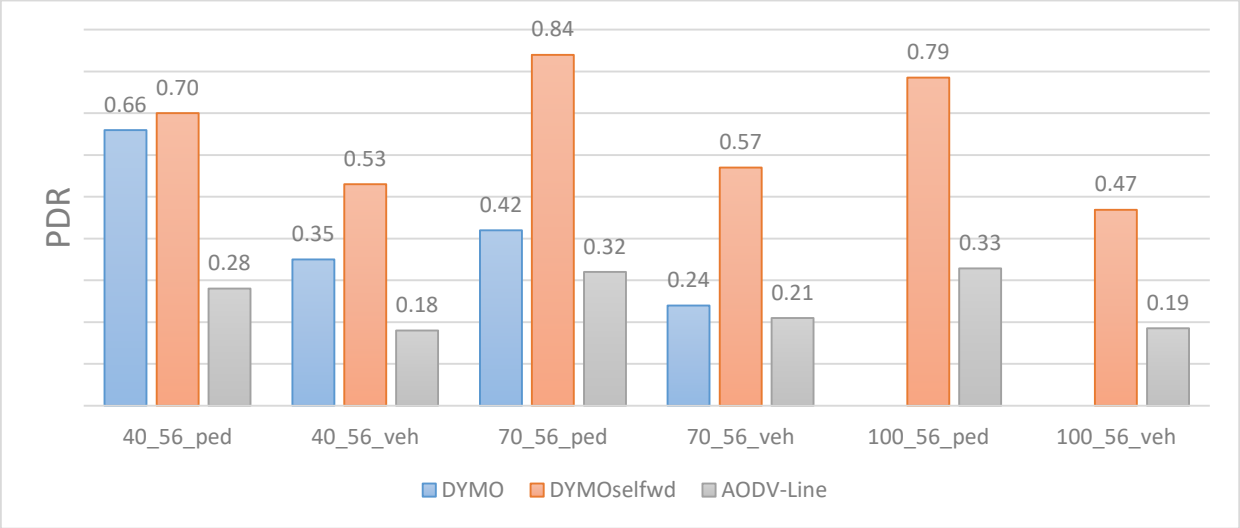

Figure 24: 40/70/100_56_ped/veh - Packet delivery ratio

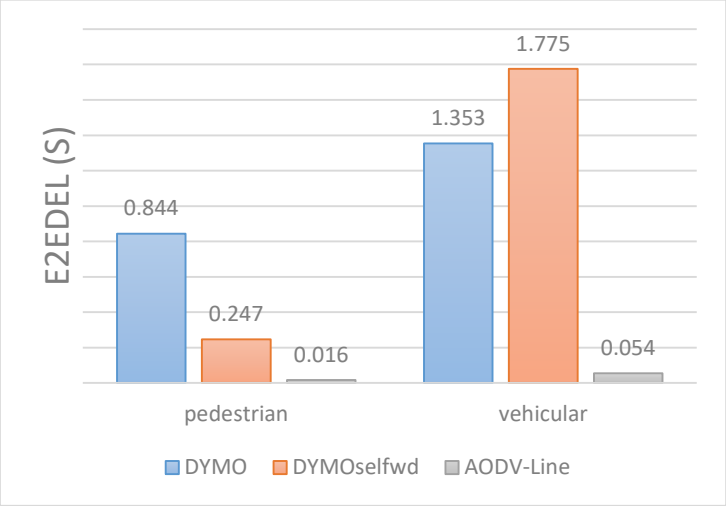

Figure 25: 70_56_ped/veh - e2eDel

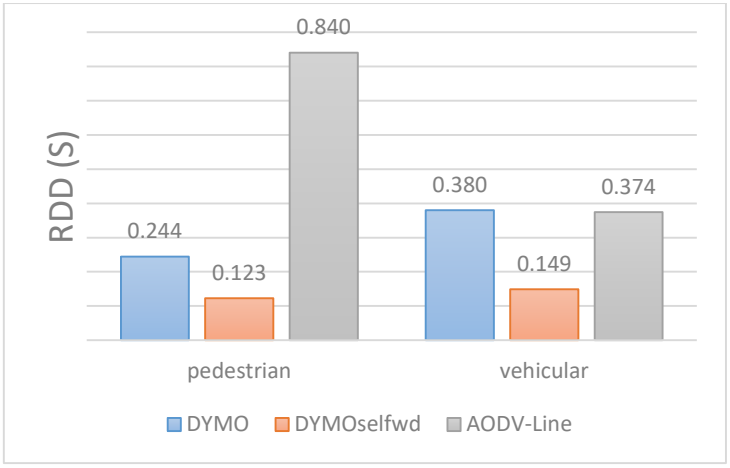

Figure 26: 70_56_ped/veh – RDD

This effect of worsening performances can be found in the other metrics too. For simplicity, figures only display the behaviour for the 70 nodes scenario. All the result for all the use cases are provided through tables in Section **Errore. L'origine riferimento non è stata trovata.**. The e2eDel and RTT increase for all the three protocols; however, the e2eDel and RTT for DYMOselfwd increase up to values higher than DYMO ones, as shown in Figure 25 and in Figure 27. This behaviour can be explained considering the location-based nature of the protocol, which is much more affected by nodes' position than DYMO.

While the RDD increase as expected for DYMO and DYMOselfwd, it decreases for AODV-Line, as shown in Figure 26. This can be explained knowing that the RDD is calculated on successful routes discoveries. The high selectivity in RREQs retransmission of AODV-Line plus a higher mobility of the nodes, allow only routes discoveries to the nearest nodes to be successful. This is also confirmed by the really low average number of hops, as illustrated in Figure 29.

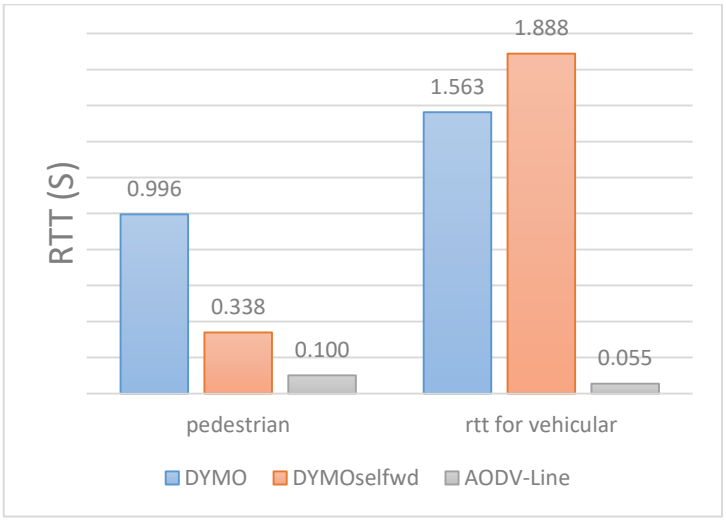

Figure 27: 70_56_ped/veh - RTT

Figure 28 shows the route life time; it decrases with increasing speed, as expected, for DYMOselfwd and AODV-Line, but it increases for DYMO. This can be explained looking at the low packet delivery ratio of DYMO (24%) in Figure 24. The protocol has been able to build few routes, the ones to nodes that are close enough to be easily maintained with high number of control packets in the network. This is also confirmed by the lower average number of hops, as shown in Figure 29.

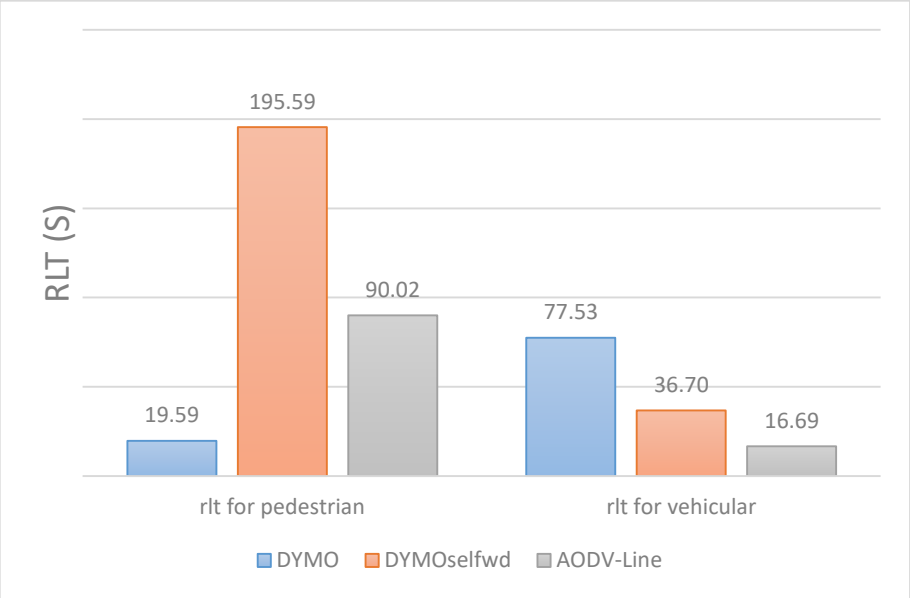

Figure 28: 70_56_ped/veh – Route life time

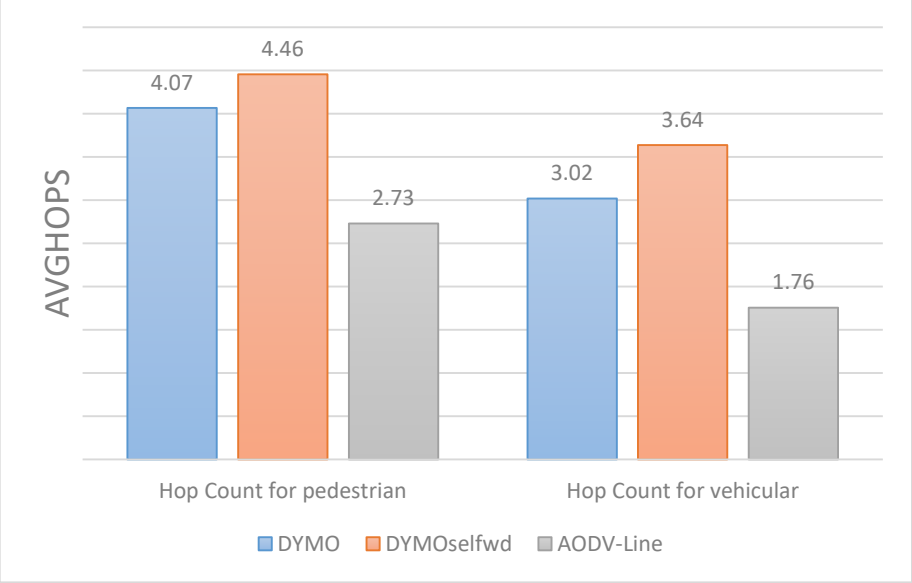

Figure 29: 70_56_ped/veh – Average number of hops

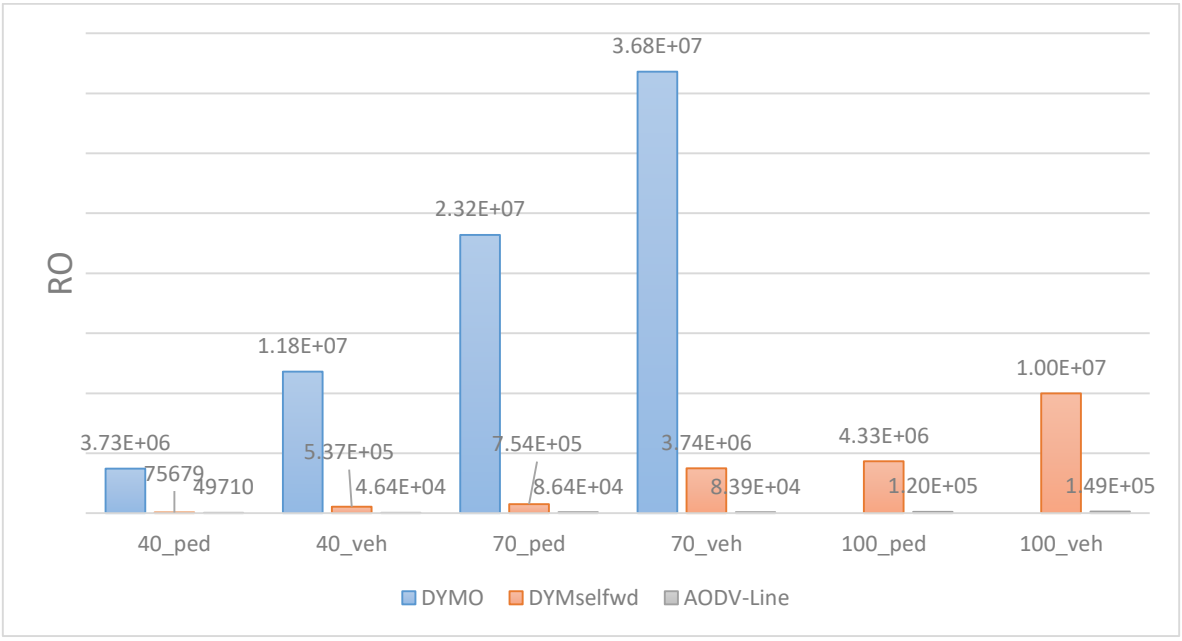

Figure 30: 40/70/100_56_ped/veh - Routing overhead

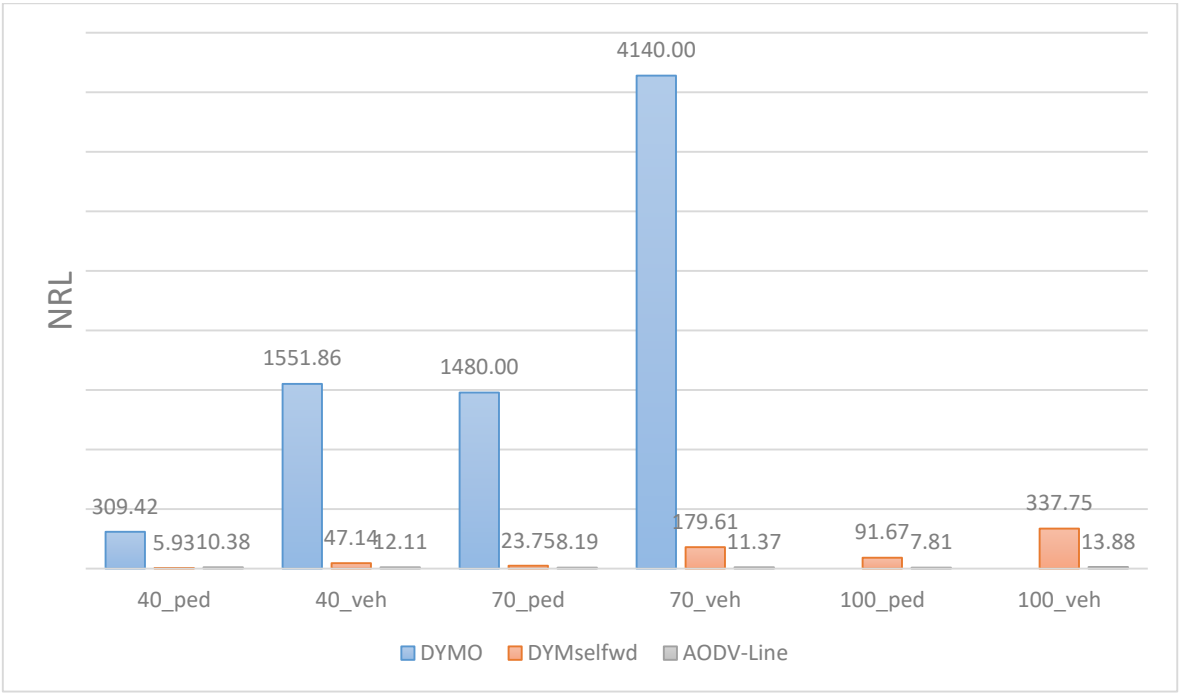

Figure 31: 40/70/100_56_ped/veh - Normalized routing load

Figure 30 and Figure 31 show that the number of routing packets, as well as the normalized routing load, always increase in a vehicular scenario compared to a scenario with lower mobility, as expected.

8.4. Impact of payload

The effect of different ping payload has been considered interesting because heavier payloads congest the network more. In this section, a scenario with 70 pedestrian users is considered, while varying the payload of the ping from 56 bytes to 500 and 1472 bytes.

The packet delivery ratio in the three cases is displayed in Figure 32. As expected, in general the PDR decreases as the payload increases. This decrease is more evident for DYMO compared to the two LB protocols, which display a stable behaviour. Both location-based protocols are resistant to the increase of the ping payload, because their network is less congested than the DYMO's, considering that they create less control packets.

Overall, the DYMOselfwd outperforms the others in terms of PDR; also, with data carrying higher payloads, both DYMO and AODV-line do not meet the quality of service required in a network.

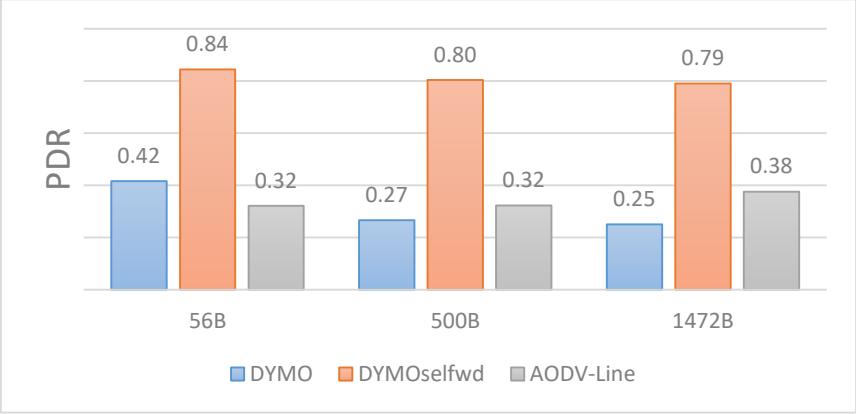

Figure 32: 70_56/500/1472_ped - Packet delivery ratio

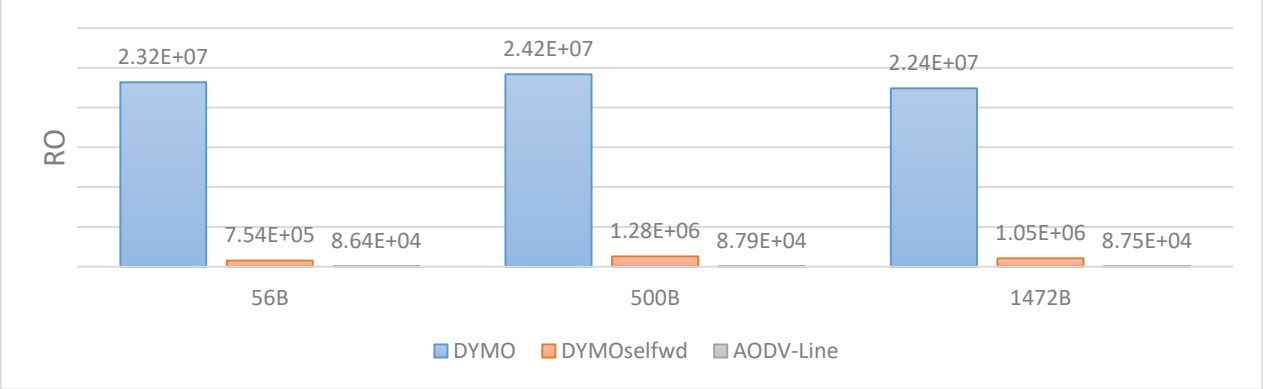

Figure 33: 70_56/500/1472_ped - Routing overhead

This is confirmed by the trend shown in Figure 33 and in Figure 34, where the routing overhead and the NRL are shown, respectively. The RO of DYMO is two orders of magnitude higher than DYMOselfwd in 56 bytes payload case and one order of magnitude higher in both 500 bytes and 1472 bytes payload cases; as the PDR is very low, the difference with AODV-Line is even larger. The DYMO protocol is much more affected by the size of the data because it also creates a lot of control packets, and the network is much more congested. In

fact, the normalized routing load of DYMO increases with the load, while for DYMOselfwd is more stable; however, it increases when the payload is increased to 500 bytes, and slightly decreases when the payload is further increased. The stability of the behaviour of AODV-Line reflects again how selective this routing protocol is in retransmitting routing packets.

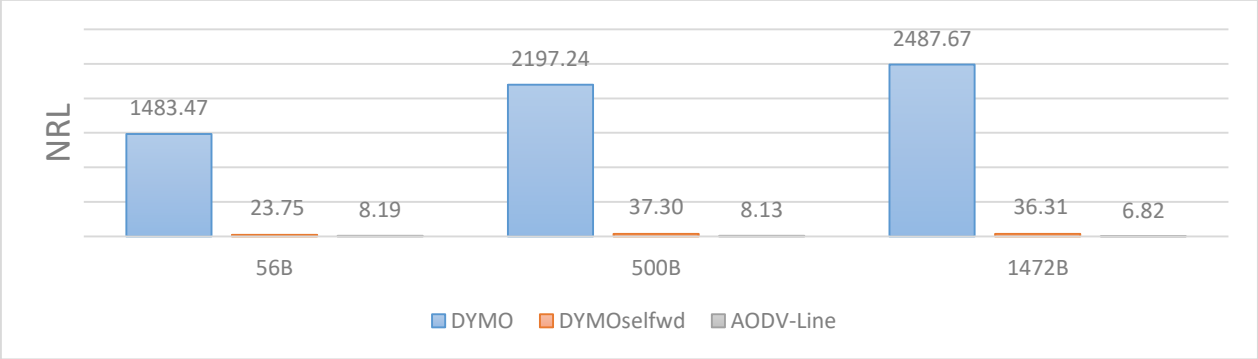

Figure 34: 70_56/500/1472_ped - Normalized routing load

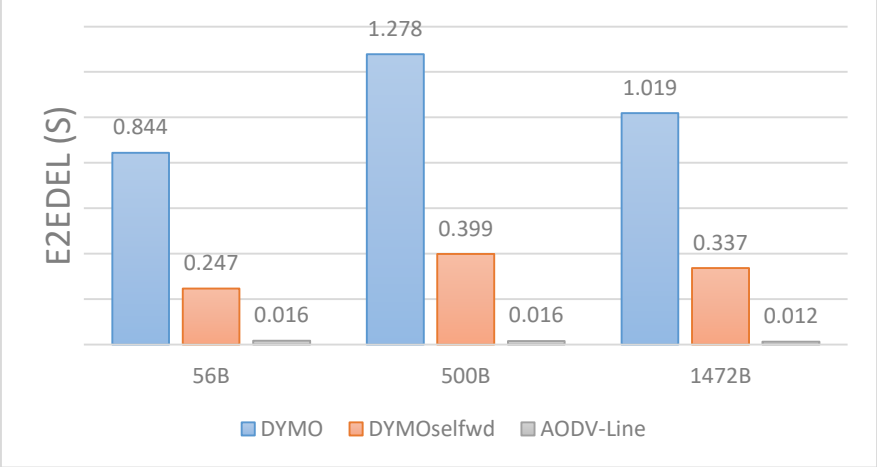

Figure 35: 70_56/500/1472_ped - e2eDel

Figure 35 shows the end to end delay of the three protocols in the cases of ping payload equal to 56, 500 and 1472 bytes. The e2eDel of DYMO and DYMOselfwd increases when the payload is increased from 56 to 500 bytes and decreases when it is further increased to 1472 bytes. On the other hand, the e2eDel of AODV-Line is pretty stable, due to the pretty stable but very low PDR. As the e2eDel is calculated only on the successfully delivered packets, the drop of the PDR also explains why the end to end delay with the greatest ping payload, 1472 bytes, is better than with 500 bytes. This effect is even more emphasized for the round trip time, as illustrated in Figure 36, that follows the same trend of the end to end delay.

Figure 37 shows the route discovery delay for the three protocols in all possible cases of ping payload. The RDD is pretty stable for all the protocols; however, again related to the trend in the PDR, a little decrease in the RDD is observed for DYMOselfwd and AODV-Line in the case with 1472 bytes ping payload.

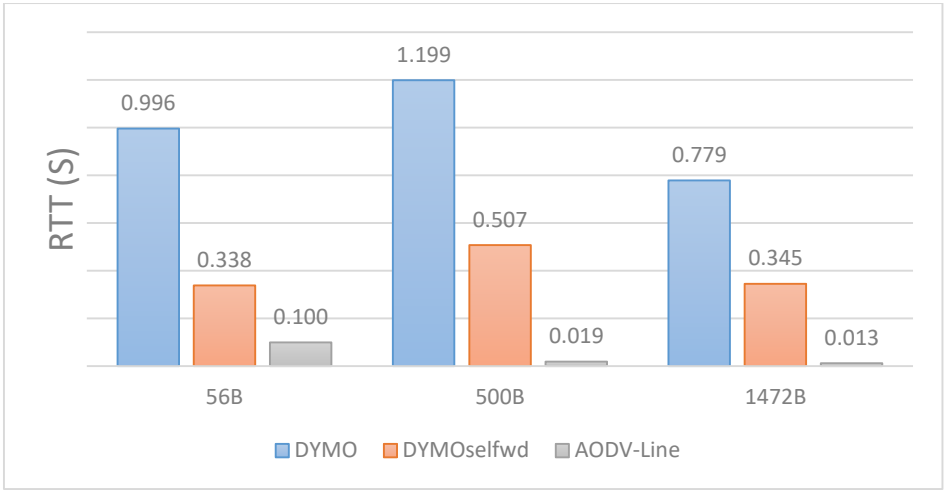

Figure 36: 70_56/500/1472_ped - RTT

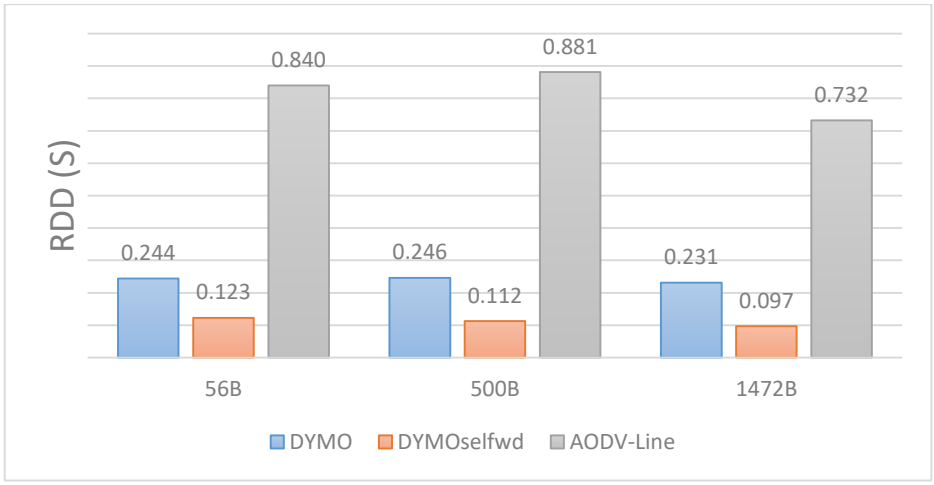

Figure 37: 70_56/500/1472_ped – RDD

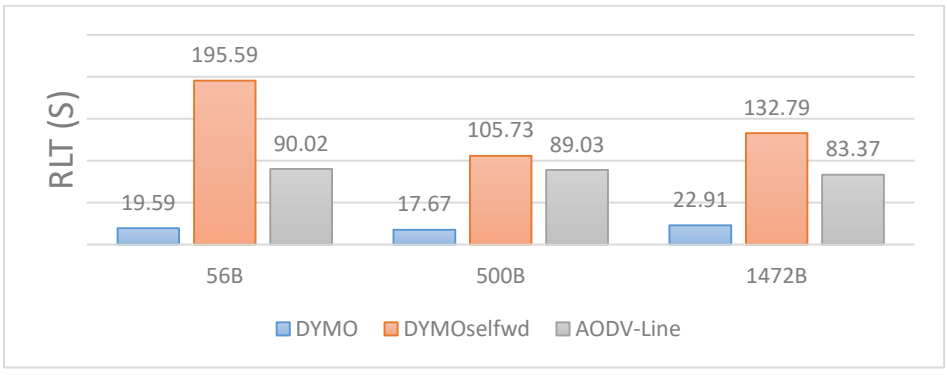

Figure 38: 70_56/500/1472_ped - Route life time

The route lifetime is pretty stable for DYMO and AODV-line, as illustrated in Figure 38; on the other hand, RLT in DYMOselfwd decreases when the payload is increased to 500 bytes, and increases when the payload is increased again. This behaviour may be understood after looking to the average number of hops, which is

illustrated in Figure 39. In the case of DYMOselfwd with 1472 bytes, the AvgHops decreases compared to lower payloads; thus, a lower number of route may suffer a disruption, thus justifying the increase in the RLT.

The average number of hops is pretty stable for DYMO and DYMOselfwd when the payload is increased from 56 to 500 bytes. A further increase in the payload means a decrease of the AvgHops, too, due to a little decrease of performance that allow to have successful transmission, especially between nodes that are close to each other. This decrease is also observed in AODV-line when the payload is increased.

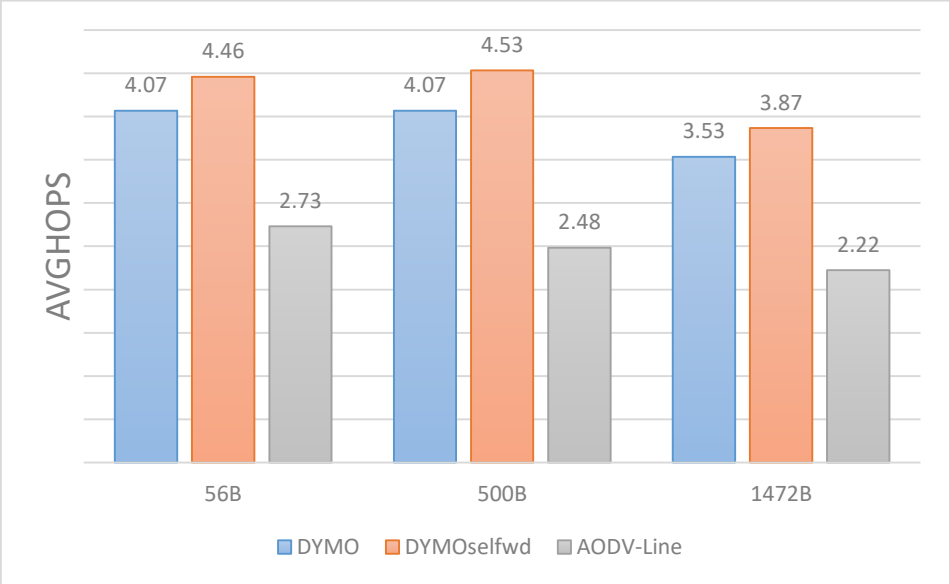

Figure 39: 70_56/500/1472_ped – Average number of hops

8.5. Impact of positioning error

One of the most relevant aspects to assess the reliability of a location-based routing protocol is the impact of the error that comes with the position information. In the simulations, the error is uniformly distributed between 0 to 5 meters; this error is then summed to the real position on both X and Y axes.

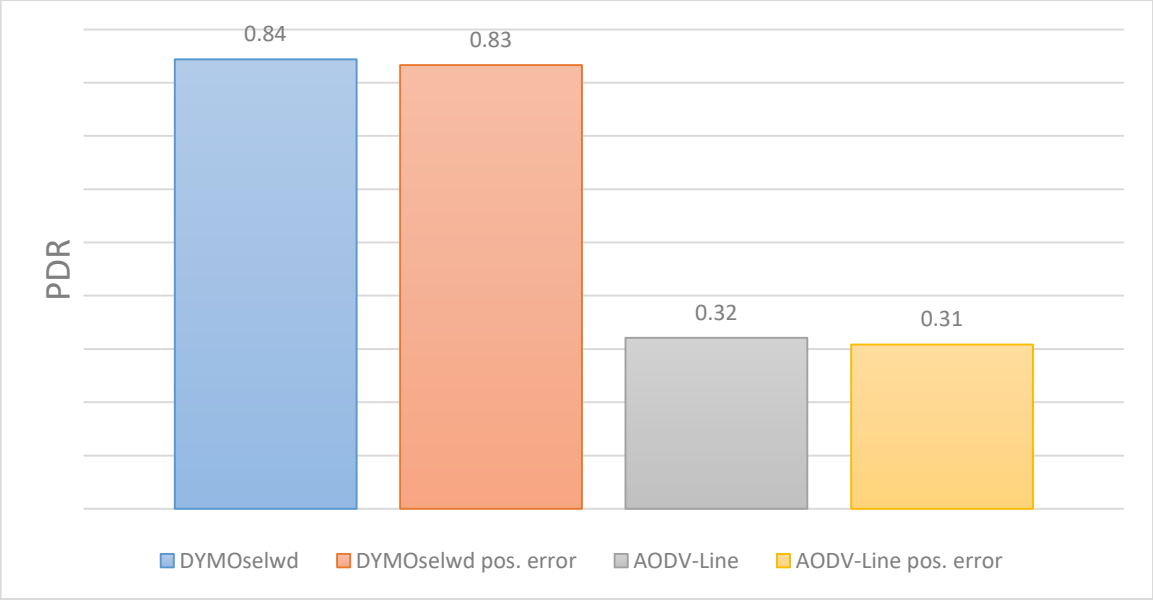

Figure 40: 70_56_ped - Packet delivery ratio (with pos. Error)

Surprisingly, the impact of the position error is insignificant in the packet delivery ratio, as illustrated in Figure 40, which depicts a pretty stable behaviour. A similar behaviour is found for the end to end delay and the round trip time, shown in Figure 41, the route life time, shown in Figure 42, the average number of hops, shown in Figure 43, the routing overhead, shown in Figure 44, and the normalized routing load, shown in Figure 45. In order to make it easy for the reader, all these metrics are summarized in Table 3.

The reason is that the LB strategies used by the two protocols under study seem to be robust to the position error. Similar behaviour can be found in the vehicular scenario (see from Table 5 to **Errore. L'origine riferimento non è stata trovata.**), thus confirming the robustness of the proposed strategy.

Table 3: 70_56_ped performances with position error

	e2eDel	RTT	RLT	Avg #hops	RO	NRL
DYMOselfwd	0.246	0.338	195.59	4.45	7.54E+05	23.74
DYMOselfwd + Error	0.252	0.338	190.46	4.36	7.71E+05	24.61
AODV-Line	0.016	0.100	90.02	2.72	8.64E+04	8.19
AODV-Line + Error	0.014	0.097	95.06	2.79	8.55E+04	8.48

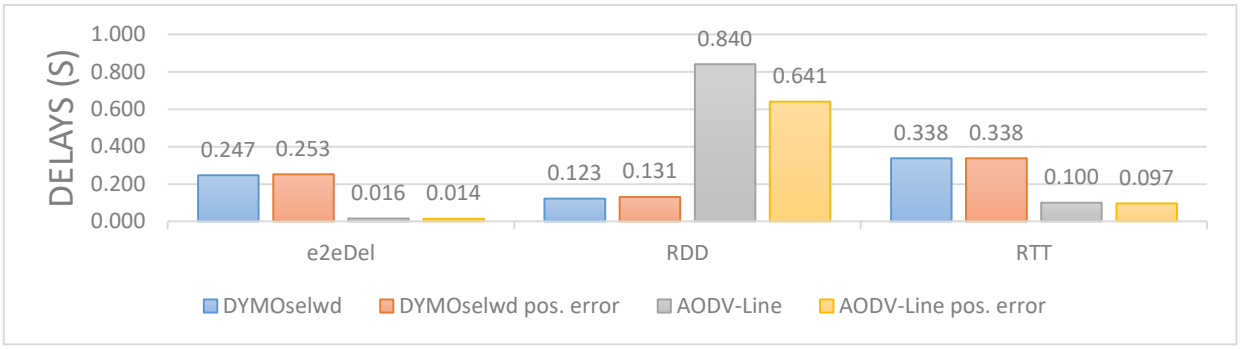

Figure 41: 70_56_ped - delays (with pos. Error)

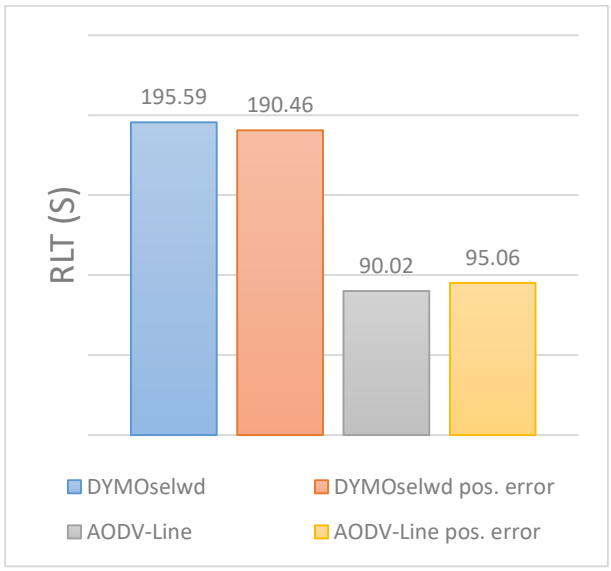

Figure 42: 70_56_ped - Route life time (with pos. Error)

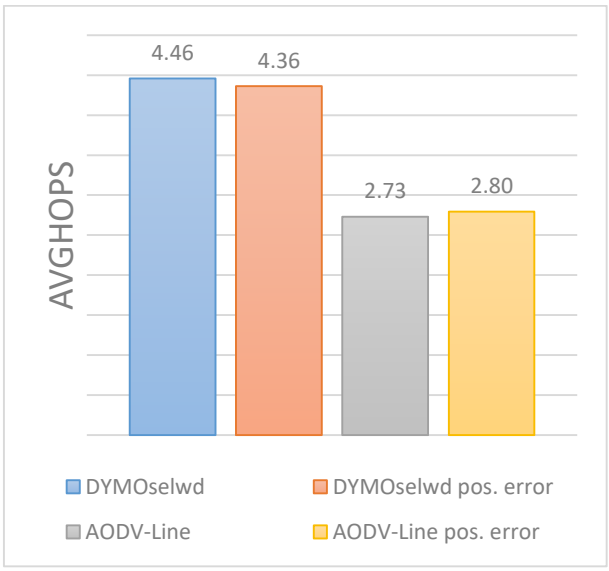

Figure 43: 70_56_ped – Average number of hops (with pos. Error)

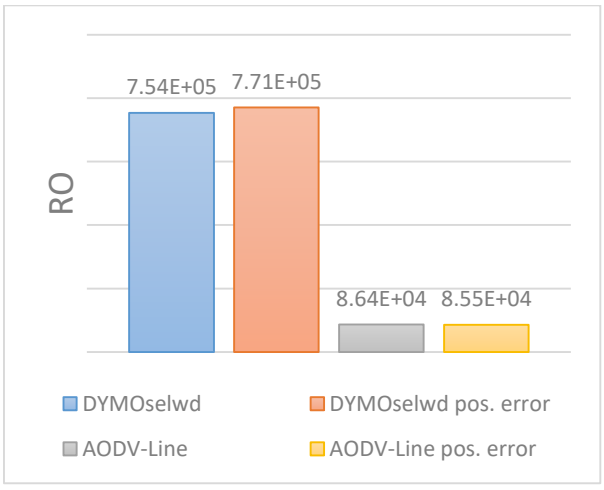

Figure 44: 70_56_ped – Routing overhead (with pos. Error)

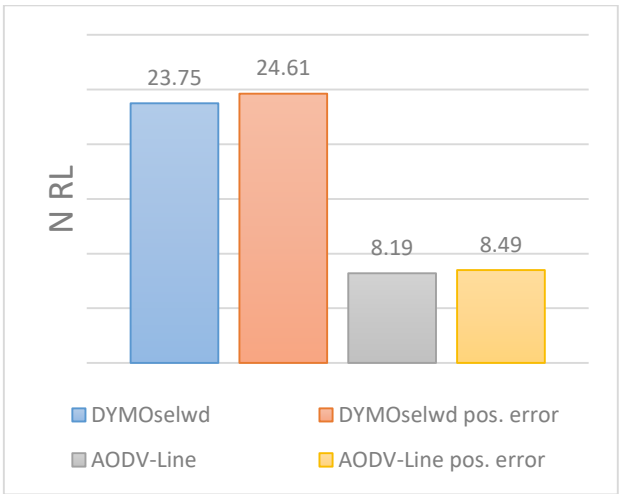

Figure 45: 70_56_ped – Normalized routing load (with pos. Error)

9. Conclusion and future work

This thesis deals with some of the routing protocols that have been developed for MANETs so far; special effort has been given to those protocols that are considered important for historical reasons, excellent performance or novelty. Instead of pointing differences between proactive and reactive and hybrid routing protocols by following the classical approach, this work focuses on the differences in performance between regular protocols and location-based (LB) ones.

Location-based routing protocols can use location information to improve the route discovery process, and their performance is usually higher than regular ones. Among LB protocols, DYMOselfwd was one of the most promising ones and AODV-Line was an interesting option to compare against. Even if these protocols are both LB, the way they use location information is completely different.

DYMO, DYMOselfwd and AODV-Line have been chosen to be implemented with OMNET++ and then simulated in different scenarios in order to understand which one offers the best performance. In case one needs to limit the number of control packets as much as possible, AODV-Line seems the best option. The drawback of this protocol is that it is not reliable: the packet delivery ratio is too low in all the tested configurations and it would not allow to communicate in a good way, and it is probably not functional to most cases.

On the other hand, DYMOselfwd has solid and functional performance at the cost of a sustainable routing overhead, which is further lower than the one for DYMO. The time needed to find routes is the lowest, but, at the same time, its routes have a quite long life, which means a good quality of the routes built. DYMO performance is fair, in some cases better than AODV-Line, but with the cost of an enormous quantity of control packets.

As a conclusion, our results show that the position information can be a huge advantage and that DYMOselfwd is an effective improvement to the original DYMO. Every test on the impact of considered factors showed that DYMOselfwd is the most reliable protocol; however, it may not be suitable for high speed scenarios. This is due to the fact that, if a routing protocol uses position information to build routes, the quality of those routes depends on how fresh is the position information and not only on how fast nodes can move after having built the route. This aspect is also noticeable in AODV-Line. Original DYMO has worse performance in the vehicular scenario, too, but it is clearly less affected by the nodes speed.

In this work, the impact of the error in the location information provided to the LB protocols has been assessed through simulation in several scenarios. To the best of our knowledge, very few authors have been interested in taking into account this error in their analysis. Our results show that the two LB protocols under study seem robust to the position error, as their behaviour is very stable in all the metric under study.

The simulated scenarios explore the most important factors that may affect performance. For technical reasons, it has not been possible to simulate the original DYMO protocol in a 100 nodes scenarios and it would be interesting to continue DYMO simulations as future work. Also, it would be useful to conduct more tests on the impact of position information error. Finally, it would be interesting to extend the study and comparison with other LB routing protocols, in order to fully understand the scalability, reliability and general performance of the DYMOselfwd.

10. Tables with all the results

Table 4: DYMO - results

	PDR	E2EDEL (s)	RDD (s)	RTT (s)	RLT (s)	Hops	RO (#pkt)	NRL
40_56_ped	0.66496	0.40674	0.14517	0.54956	92.296	3.5122	3.73E+06	3.09E+02
40_56_veh	0.35371	1.2936	0.23135	1.3286	55.447	3.0018	1.18E+07	1.55E+03
40_500_ped	0.59164	0.66799	0.14847	0.67778	32.172	3.6038	5.96E+06	4.76E+02
40_500_veh	0.35956	1.2675	0.21869	1.5294	76.4	2.793	1.02E+07	1.63E+03
40_1472_ped	0.50522	0.7977	0.1465	0.75692	18.741	3.7612	7.36E+06	6.13E+02
40_1472_veh	0.31231	1.5757	0.25415	1.5387	44.899	2.9466	1.28E+07	1.85E+03
70_56_ped	0.41587	0.84406	0.24412	0.99577	19.59	4.067	2.32E+07	1.48E+03
70_56_veh	0.24445	1.3533	0.38042	1.5629	77.533	3.0191	3.68E+07	4.14E+03
70_500_ped	0.26546	1.2777	0.24638	1.1992	17.67	4.0651	2.42E+07	2.20E+03
70_500_veh	0.1628	1.6879	0.34931	1.6697	62.285	3.0743	3.42E+07	5.22E+03
70_1472_ped	0.24993	1.0193	0.23129	0.77903	22.91	3.5321	2.24E+07	2.49E+03
70_1472_veh	0.13937	1.8023	0.4118	1.6438	67.946	2.8958	3.75E+07	6.55E+03

Table 5: DYMOselfwd - results

	PDR	E2EDEL (s)	RDD (s)	RTT (s)	RLT (s)	Hops	RO (#pkt)	NRL
40_56_ped	0.70109	0.42477	0.10144	0.4774	350.57	4.593	75679	5.928726
40_56_veh	0.52653	2.173	0.098835	2.1967	95.551	3.8159	5.37E+05	47.13917
40_500_ped	0.71729	0.34819	0.13757	0.38298	295.84	4.5096	92098	6.05064
40_500_veh	0.53619	2.0139	0.10343	2.0635	114.17	3.4696	4.82E+05	50.58031
40_1472_ped	0.75845	0.35895	0.10798	0.3899	225.79	4.4766	1.30E+05	7.177245
40_1472_veh	0.50149	2.0625	0.092942	1.9711	75.202	3.359	5.86E+05	51.93746
70_56_ped	0.84398	0.24676	0.12288	0.3382	195.59	4.4581	7.54E+05	23.74788
70_56_veh	0.5693	1.7749	0.14924	1.8884	36.7	3.6386	3.74E+06	179.6086
70_500_ped	0.80308	0.39875	0.1124	0.50728	105.73	4.5335	1.28E+06	37.29712
70_500_veh	0.52254	1.9222	0.1439	2.0737	26.379	3.8118	3.63E+06	171.7465
70_1472_ped	0.79025	0.33666	0.096693	0.34522	132.79	3.8665	1.05E+06	36.31383
70_1472_veh	0.47156	1.8377	0.14139	1.8895	23.164	3.5897	4.72E+06	240.0692
100_56_ped	0.78539	0.59206	0.12858	0.80707	54.447	4.5384	4.33E+06	91.67009
100_56_veh	0.46922	2.1961	0.19832	2.443	14.95	4.0038	1.00E+07	337.7466
100_500_ped	0.58905	1.0729	0.12287	1.2925	25.237	4.4513	6.54E+06	180.8778
100_500_veh	0.4212	2.1297	0.1843	2.3633	20.894	3.6859	1.04E+07	404.6409
100_1472_ped	0.46162	1.1263	0.11923	1.1474	31.877	4.2926	6.20E+06	261.9615
100_1472_veh	0.34795	2.1117	0.17328	2.1322	19.868	3.4966	9.58E+06	475.0005

Table 6: DYMOselwd with position error - results

	PDR	E2EDEL (s)	RDD (s)	RTT (s)	RLT (s)	Hops	RO (#pkt)	NRL
40_56_ped	0.70411	0.41818	0.14826	0.44798	372.33	4.4027	65647	5.144749
40_56_veh	0.51585	2.0927	0.089648	2.1515	101.62	3.6856	5.64E+05	50.66384
40_500_ped	0.71919	0.38015	0.093577	0.43235	285.15	4.4086	1.08E+05	7.075379
40_500_veh	0.54229	1.9656	0.10599	2.0882	109.95	3.574	5.20E+05	53.2296
40_1472_ped	0.77391	0.37011	0.094974	0.40737	218.11	4.5438	1.31E+05	7.090817
40_1472_veh	0.50627	2.0652	0.098333	1.9715	60.405	3.5239	6.15E+05	54.45176
70_56_ped	0.83285	0.25291	0.13133	0.33804	190.46	4.3622	7.71E+05	24.61057
70_56_veh	0.55998	1.819	0.14294	1.9027	35.983	3.754	3.88E+06	188.8482
70_500_ped	0.79573	0.38169	0.12083	0.47023	119.12	4.4961	1.24E+06	36.45343
70_500_veh	0.5348	1.9357	0.13672	2.0314	25.149	3.7447	3.54E+06	163.5255
70_1472_ped	0.79363	0.35703	0.11252	0.3769	132.27	3.973	1.03E+06	35.5344
70_1472_veh	0.4734	1.765	0.13736	1.8183	26.28	3.4492	4.43E+06	223.6741
100_56_ped	0.78953	0.58096	0.12932	0.8191	51.817	4.6164	4.44E+06	93.54616
100_56_veh	0.47618	2.1126	0.19075	2.3718	16.269	3.9618	9.15E+06	305.1363
100_500_ped	0.5916	1.0762	0.12145	1.3224	25.72	4.4773	6.44E+06	177.3518
100_500_veh	0.42376	2.0868	0.17845	2.3418	19.832	3.636	1.02E+07	395.2845
100_1472_ped	0.46803	1.108	0.11836	1.1215	33.277	4.3123	6.07E+06	253.2311
100_1472_veh	0.34045	2.1549	0.17389	2.1967	21.716	3.4276	9.82E+06	496.9082

Table 7: AODV-Line - results

	PDR	E2EDEL (s)	RDD (s)	RTT (s)	RLT (s)	Hops	RO (#pkt)	NRL
40_56_ped	0.27612	0.017161	0.68222	0.02279	144.89	2.7116	49710	10.38134
40_56_veh	0.18367	0.027593	0.3069	0.03226	18.392	1.6582	4.64E+04	12.10911
40_500_ped	0.28288	0.026978	0.99713	0.031489	141.77	2.6038	4.70E+04	9.675659
40_500_veh	0.22261	0.064273	0.28486	0.071885	15.522	1.9034	4.89E+04	13.96011
40_1472_ped	0.28838	0.013927	0.77832	0.021898	126.58	2.731	4.60E+04	7.892527
40_1472_veh	0.20431	0.074147	0.59782	0.05165	17.747	1.665	4.60E+04	10.70458
70_56_ped	0.32066	0.016458	0.84015	0.10007	90.023	2.7298	8.64E+04	8.191834
70_56_veh	0.21435	0.054448	0.37431	0.055104	16.687	1.7565	8.39E+04	11.3688
70_500_ped	0.32289	0.015854	0.88144	0.018891	89.032	2.4849	8.79E+04	8.126063
70_500_veh	0.2005	0.036436	0.36124	0.040107	15.592	1.8018	8.66E+04	11.3172
70_1472_ped	0.37566	0.01185	0.73196	0.013222	83.37	2.2237	8.75E+04	6.815619
70_1472_veh	0.18697	0.030821	0.38515	0.033498	14.502	1.6883	8.87E+04	11.79246
100_56_ped	0.32866	0.008058	0.73077	0.015867	75.951	2.1881	1.20E+05	7.808636
100_56_veh	0.18501	0.042052	0.3986	0.062582	12.052	1.8806	1.49E+05	13.87828
100_500_ped	0.2918	0.008409	0.57964	0.016513	64.699	2.2087	1.28E+05	8.585533
100_500_veh	0.19806	0.047766	0.25259	0.089899	13.016	1.7491	1.29E+05	11.28293

100_1472_ped	0.30221	0.035059	0.60791	0.046493	64.921	2.3372	1.22E+05	9.492095
100_1472_veh	0.19626	0.045257	0.37699	0.046653	15.375	1.7137	1.25E+05	11.15394

Table 8: AODV-Line with position error - results

	PDR	E2EDEL (s)	RDD (s)	RTT (s)	RLT (s)	Hops	RO (#pkt)	NRL
40_56_ped	0.26654	0.021488	0.68264	0.025544	146.69	2.7077	47556	10.2271
40_56_veh	0.18172	0.046676	0.38204	0.076374	17.896	1.6868	4.64E+04	12.35859
40_500_ped	0.28082	0.026471	0.73428	0.030828	139.94	2.6002	4.72E+04	9.776218
40_500_veh	0.2062	0.062968	0.47426	0.075533	15.772	1.8304	4.90E+04	14.60409
40_1472_ped	0.3001	0.016879	0.6346	0.027298	120.45	2.7559	4.63E+04	7.616453
40_1472_veh	0.19136	0.072284	0.7195	0.051441	17.572	1.6753	4.61E+04	11.5116
70_56_ped	0.3083	0.014353	0.64138	0.096837	95.061	2.7955	8.55E+04	8.489462
70_56_veh	0.22133	0.047314	0.28751	0.051229	11.22	1.8606	1.06E+05	13.5173
70_500_ped	0.3429	0.014893	0.79117	0.018177	85.199	2.5223	8.70E+04	7.61268
70_500_veh	0.20225	0.03593	0.32303	0.043029	13.528	1.8082	9.39E+04	12.031
70_1472_ped	0.38594	0.01227	0.87076	0.013719	80.695	2.2572	8.86E+04	6.667344
70_1472_veh	0.19755	0.024972	0.28533	0.025417	13.889	1.6559	9.13E+04	11.24418
100_56_ped	0.33857	0.007007	0.65938	0.016667	74.935	2.1831	1.20E+05	7.577767
100_56_veh	0.18072	0.061826	0.37705	0.10109	11.14	1.8598	1.52E+05	14.15639
100_500_ped	0.2692	0.010188	0.59858	0.016997	77.75	2.1767	1.25E+05	8.956978
100_500_veh	0.201	0.036681	0.41009	0.06124	14.633	1.7255	1.18E+05	10.0376
100_1472_ped	0.31247	0.03605	0.59312	0.045976	61.342	2.3706	1.26E+05	9.487809
100_1472_veh	0.18212	0.028459	0.33458	0.030876	15.401	1.6821	1.23E+05	11.79539

11. Acknowledgments

I want to say thank you to Israel and Enrica for helping me in these six months.

Muchas gracias!

12. Bibliography

[1] B. B. S. a. S. T. A. Makkar, "Behavioral Study of MANET Routing Protocols," *International Journal of Innovation, Management and Technology,* vol. 2, no. 3, pp. 210-216, 2011.

[2] J. Z. Dan Luo, "An Improved Hybrid Location-Based Routing Protocol for Ad Hoc Networks," in *Networking and Mobile Computing (WiCOM)*, 2011.

[3] A. M. B. U. A. k. Y. Ashish Srivastava, "Survey and Overview of Mobile Ad-hoc Network Routing Protocols," in *IEEE International Conference on Advances in Engineering &Technology Research (ICAETR - 2014)*, 2014.

[4] P. R. A. P. T. Prasanna venkatesan, "Overview of Proactive Routing Protocols in MANET," in *2014 Fourth International Conference on Communication Systems and Network Technologies*, 2014.

[5] C. Bhagwat, "Highly Dynamic Destination Sequence Distance Vector Routing (DSDV) for Mobile Networks," *proceedings of ACM SIGCOMM,* pp. 234-244, 1994.

[6] S. M. a. J. Garcia, "A Routing Protocol for Packet Radio Networks," *proc. of ACM MOBICOM,* 1995.

[7] P. J. C. A. A. L. P. M. P. M. A. Q. L. V. Thomas Clausen, *Optimized Link State Routing Protocol (OLSR),* 2003.

[8] H. W. W. L. a. M. G. C.C. Chiang, "Routing in Clustered Multi Hop Mobile Wireless Networks with Fading Channel," in *proceedings of IEEE ICN*, 1997.

[9] M. G. a. T. W. C. G. Pei, *Fisheye state routing: a routing scheme for ad hoc wireless networks,* IEEE International Conference on Communications, 2000.

[10] D. P. Hrituparna Paul, "Performance Evaluation of MANET Routing Protocols," *IJCSI International Journal of Computer Science Issues,* vol. 9, no. 2, p. issue 4, 2012.

[11] A. M. N. A. S. A. V. Rais Khanl, *Enhancement of Manet Routing Protocol,* IT in Business, Industry and Government (CSIBIG), 2014.

[12] D. B. J. D. A. M. J. Broch, *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks,* IETF MANET Working Group, 1998.

[13] V. P. a. S. Corson, *Temporally Ordered Routing Algorithm (TORA) Version 1 Functional Specification,* Internet Engineering Task Force (IETF) draft, 2001.

[14] D. P. U. M. B. S. G. S. Rajeswari M, *Performance analysis of AODV, DSR, TORA and OLSR to achieve group communication in MANET,* IEEE - Fourth International Conference on Advanced Computing, 2012.

[15] J. L. a. Y. T. M. Jiang, *Cluster Based Routing Protocol,* IETF Draft, 1999.

[16] M. W. W. L. D. W. Fan Guo, *Channel Quality Based Routing Protocol (CQBR) and Realization on MANET Platform,* 21st International Conference on Telecommunications (ICT), 2014.

[17] V. T. R. Shitalkumar A Jain, *Load Equilibrium Neighbor Aware Routing in Mobile Ad Hoc Network,* Annual IEEE India Conference (INDICON), 2014.

[18] E. B.-R. S. D. C. Perkins, *Ad hoc On-Demand Distance Vector (AODV) Routing,* RFC 3561, 2003.

[19] F. B.-A. a. I. M.-E. Enrica Zola, *A Modification of DYMO Routing Protocol with Knowledge of Nodes' Position: Proposal and Evaluation,* 2013.

[20] M. H. R. J. Nazari Talooki, *Energy efficient dynamic MANET on-demand (E2DYMO) routing protocol,* World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013.

[21] S. C. P. C. D. S. Maya C Aravind, *Enhanced Dynamic MANET On-demand(En-DYMO) Routing Protocol for Mobile Adhoc Networks,* Communication Technologies (GCCT), 2015.

[22] X. C. X. J. Zhang Xijie, *Hierarchical ZRP's Performance vs ZRP's Performance in MANET,* Communication Software and Networks (ICCSN), 2015.

[23] M. J.-N. a. I. Lu, "A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications,* p. 1415–1425, 1999.

[24] R. G. S. Nair, *Performance analysis of threshold based hybrid routing protocol for MANET,* Signal Processing, Communication and Networking (ICSCN), 2015.

[25] N. E. A. Taraka, *"Routing in Ad Hoc Networks Using Ant Colony Optimization,* Intelligent Systems, Modelling and Simulation (ISMS), 2014.

[26] M. D. V. J. P. Y. J. T. L. S. A. Shadi S. Basurra, "Energy efficient zone based routing protocol for MANETs," in *Ad hoc network*, Elsevier, 2015, pp. 16-37.

[27] I. C. V. S. B. W. S. Basagni, "A distance routing effect algorithm for mobility (DREAM)," *Proceedings of the ACM MOBICOM,* pp. 76-84, 1998.

[28] E. S. S. L. J. Raj, *LBRP: Geographic routing protocols for MANETs,* Recent Trends in Information Technology (ICRTIT), 2011.

[29] B. K. a. H. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proceedings of ACM MobiCom,* p. 243–254, 2000.

[30] N. H. V. Young-Bae Ko, *GeoTORA: A Protocol for Geocasting in Mobile Ad Hoc Networks,* Network Protocols, 2000.

[31] D. K. S. C. G. Ashutosh Srivastava, *Geographic and Reactive Routing Protocols for MANET,* European Modelling Symposium, 2013.

[32] N. H. V. Young-Bae Ko, *Location-Aided Routing (LAR) in Mobile Ad Hoc Networks,* Wireless networks, 2000.

[33] D. S. R. M. Anitha Veerasamy, *An Improved Opportunistic Location Aided Routing (IOLAR) in Mobile Ad hoc Networks,* 2nd International Conference on Current Trends in Engineering and Technology, ICCTET'14, 2014.

[34] E. S. P. A. R. M. N. Rozita Aboki, *Predictive Location Aided Routing in Mobile Ad hoc Network,* 11th Malaysia International Conference on Communications, 2013.

[35] M. M. Mohannad Ayash, *Improved AODV Routing Protocol to Cope With High Overhead in High Mobility MANETs,* Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2012.

[36] S. S. B. D. B. D. DasGupta S., *LBSPR: Location based shortest path routing protocol in MANET,* Computational Intelligence and Computing Research (ICCIC), 2010.

[37] S. B. G. Malwe, *Location aware Sector-based Routing in Wireless Ad hoc Networks,* Green Computing and Internet of Things (ICGCIoT), 2015.

[38] K. R. I. W. Mabrouka Abuhmida, *Evaluating the Performance of ANTMANET Protocol for MANET,* Internet Technologies and Applications (ITA), 2015.

[39] J. Z. Dan Luo, *An Improved Hybrid Location-Based Routing Protocol for Ad Hoc Networks,* Wireless Communications, Networking and Mobile Computing (WiCOM), 2011.

[40] S. R. J. D. L. S. a. V. M. C. Perkins, *Ad Hoc On-demand Distance Vector Version 2 (AODVv2) Routing,* draft-ietf-manet-aodvv2-16, 2016.

[41] I. C. C. Perkins, *Dynamic MANET On-demand (AODVv2) Routing,* draft-ietf-manet-dymo-24, 2012.

[42] N. m. C. J. m. S. D. Christhu raj M.R, "A Comprehensive Overview on Different," *International Journal of Engineering and Technology (IJET),* p. Vol 5 No 1, Feb-Mar 2013.

[43] "ns2-ns3," [Online]. Available: https://www.nsnam.org/support/faq/ns2-ns3/. [Accessed 20 August 2016].

[44] [Online]. Available: https://omnetpp.org/intro. [Accessed 20 August 2016].

[45] "sourceforge," [Online]. Available: https://sourceforge.net/p/dymo2010/code/ci/master/tree/.

[46] [Online]. Available: https://www.nsnam.org/doxygen/aodv_8cc_source.html.

[47] A. Varga, OMNeT++ User Manual Version 5.0.

[48] "inet.omnetpp," [Online]. Available: https://inet.omnetpp.org/Introduction.html. [Accessed 20 August 2016].

[49] INET Framework for OMNeT++ Manual, 2016.

[50] A. H. Fan Bai, A SURVEY OF MOBILITY MODELS in Wireless Adhoc Networks, 2006.