University of Macedonia
Master in Information Systems (MIS)

Performance Comparison of Manet Routing Protocols based on real-life scenarios

Samara Chrysoula
(mis 1021)

Supervisor
Professor A. A. Economides

Thessaloniki
January, 2012

Abstract

Ad-hoc networking is a concept in computer communications, which means that users wanting to communicate with each other form a temporary network, without any form of centralized administration. Each node participating in the network acts both as host and a router and must therefore be willing to forward packets for other nodes. For this purpose, a routing protocol is needed.

An ad-hoc network has certain characteristics, which imposes new demands on the routing protocol. The most important characteristic is the dynamic topology, which is a consequence of node mobility. Nodes can change position quite frequently, which means can consist of laptops and personal digital assistants and are often very limited in resources such as CPU capacity, storage capacity, battery power and bandwidth.

A variety of routing protocols for MANETs have been developed by network researchers and designers primarily to improve the performance of MANETs with respect to correct and efficient route establishment between a pair of stations for message delivery. Examples of commonly used MANET routing protocols include ad-hoc on-demand distance vector (AODV), highly dynamic destination-sequenced distance vector routing protocol (DSDV) and optimized link state routing protocol (OLSR). A good understanding of the effect of each of these routing protocols on a typical IEEE 802.11 network will assist an efficient design and deployment of appropriate MANETs.

This present thesis aims to study the performance evaluation and comparison of three MANET routing protocols in real life simulation scenarios located in Greece, drawing valuable conclusions and future improvements.

Acknowledgements

It would not have been possible to write this master thesis without the help and support of the kind people around me, to only some of whom it is possible to give particular mention here.

First and foremost, my grateful thanks go to Prof. Anastasios A. Economides, the supervisor of my Master thesis. His invaluable and patient guidance, enthusiastic encourage and support, accompany me in every stage of my thesis research. What I learn from him will provide me with lifetime benefits. I consider myself luck to access his supervision.

My gratitude also goes to Dr. Eirini Karapistoli for her guidance, advice and support throughout the entire thesis. Additionally, I would like to thank Mr. Filippos Kolovos for his support and insightful discussions.

Finally, I want to express my deepest feeling to my family, my parents George and Maria and my brother Apostolos. Without their love and encouragement, I would not pursue new goals again and again.

Samara Chrysoula, 17 January 2012

Contents

1 Introduction

The field of wireless and mobile communications has experienced an unprecedented growth during the past decade. Current second-generation (2G) cellular systems have reached a high penetration rate, enabling worldwide mobile connectivity. Mobile users can use their cellular phones to check their email and browse the Internet.

Recently, an increasing number of wireless local area network (LAN) hot spots is emerging, allowing travelers with portable computers to surf the Internet from airports, railways, hotels and other public locations. Broadband Internet access is driving wireless LAN solutions in the home for sharing access between computers. In the meantime, 2G cellular networks are evolving to 3G, offering higher data rates, infotainment and location-based or personalized services.

However, all these networks are conventional wireless networks, conventional in the sense that as prerequisites, a fixed network infrastructure with centralized administration is required for their operation, potentially consuming a lot of time and money for set-up and maintenance. Furthermore, an increasing number of devices such as laptops, personal digital assistants (PDAs), pocket PCs, tablet PCs, smart phones, MP3 players, digital cameras, etc. are provided with short-range wireless interfaces. In addition, these devices are getting smaller, cheaper, more user friendly and more powerful. This evolution is driving a new alternative way for mobile communication, in which mobile devices form a self-creating, self-organizing and self-administering wireless network, called a mobile ad hoc network (Hoebeke, Moerman, Dhoedt, & Demeester, 2004).

The infrastructureless and the dynamic nature of these networks demands new set of networking strategies to be implemented in order to provide efficient end-to-end communication. This, along with the diverse application of these networks in many different scenarios such as battlefield and disaster recovery, has seen MANETs being researched by many different organizations and institutes. Routing in the MANETs is a challenging task and has received a tremendous amount of attention from researches. This has led to development of many different routing protocols for MANETs, and each author of each proposed protocol argues that the strategy proposed provides an improvement over a number of different strategies considered in the literature for a given network scenario. Therefore, it is quite difficult to determine which protocols may perform best under a number of different network scenarios, such as increasing node density and traffic (Abolhasan, Wysocki, & Dutkiewicz, 2004).

1.1 Previous Research

Several researchers have done the qualitative and quantitative analysis of Ad Hoc Routing Protocols by means of different performance metrics. They have used different simulators for this purpose. Most of this analysis focuses on the main challenges of MANETs which are reliability, bandwidth and battery power. Although the use of simulation has increased, the credibility of the simulation results has decreased (Kurkowski, Camp, & Colagrosso, 2005). The main disadvantage of simulation studies that leads to this is that they are not based on real life scenarios, but their research field is general and based on random assumptions. As a result, the conclusions of these simulation studies cannot be used directly in Mobile Ad Hoc Networks applications, but further study must be done before, mainly because it is difficult for one to choose a proper routing protocol for a given MANET application (Lin, 2004).

In this chapter we analyze the research that has been already done, the simulation parameters that have been used and the conclusions that have been made. This research can be broadly classified as follows based on the methodology used and the subject of study.

1.1.1 General studies

Much of the initial research was based on the comparison of routing protocols. More specifically, they concentrate on the comparison of both proactive and reactive routing protocols. Next are presented the most considerable studies from this section.

J. Broch, D. A. Maltz, D. B. Johnson, Y-C. Hu, and J. Jetcheva, in their paper (Broch, Maltz, Johnson, Hu, & Jetcheva, 1998) have compared the DSDV, TORA, DSR and AODV Protocols using ns-2 simulator. The simulation was done with 50 nodes with varying pause times. The results were obtained for the metrics: packet delivery ratio, routing overhead, number of hops taken by the packet to reach the destination.

S.R. Das, R. Castaneda, J. Yan, and R. Sengupta (Das, Castaneda, Yan, & Sengupta, 1998) evaluated the performance of routing protocols with respect to fraction of packets delivered, end-to-end delay, and routing load by varying the number of conversation per node. The evaluation was done with 30 and 60 nodes using Maryland Routing Simulator. The protocols used in the simulation are SPF, DSDV, TORA, DSR and AODV.

Azzedine Boukerche, in his paper (Boukerche, 2001) has done the performance comparison of AODV, CBRP and DSR Ad Hoc routing protocols using ns-2 simulator. The key performance metrics evaluated in his experiments are Throughput, Average End-to-end delay of data packets and Normalized routing overhead for different data sources and varying pause times of mobile nodes. As per his observation DSR and CBRP has higher throughput than in comparison with AODV.CBRP has high routing overhead than DSR.

M.Saravana Karthikeyan, K.Angayarkanni and Dr.S.Sujatha in their paper (Karthikeyan, Angayarkanni, & Sujatha, 2010) concentrate to the comparison of both proactive and reactive routing protocols using the NS-2 simulator, based on significant performance metrics like latency and throughput for better understanding of functionalities of those protocols.

Finally, P. Manickam1, T. Guru Baskar, M.Girija and Dr.D.Manimegalai present in their paper (Manickam, Guru Baskar, Girija, & Manimegalai, 2011) a performance comparison of proactive and reactive protocols DSDV, AODV and DSR based on metrics such as throughput, packet delivery ratio and average end-to-end delay by using the NS-2 simulator by varying network size, simulation time.

Some researchers focus even more on comparing protocols that belong to the same category. For example, Samir R. Das, Charles E. Perkins et al. (Das, Perkins, & Royer, Performance Comparison of TwoOn-Demand Routing Protocols for Ad Hoc Networks, 2000) who evaluated the DSR and AODV on-demand routing protocols, using the ns-2 simulator, with three performance metrics: Packet delivery fraction, Average End-to-End Delay and Normalized routing load with varying pause times. Based on the observations, recommendations were made as to how the performance of either protocol can be improved.

Furthermore, Jayakumar G. and Ganapathy G. in their paper (Jayakumar & Ganapathy, 2007) have also compared the DSR and AODV protocols using ns-2 simulator. Packet delivery fraction, Average end-to-end delay of data packets, Normalized routing load and Normalized MAC load are the four performance metrics that were considered for evaluation of these two on demand routing protocols. The simulation results brought important characteristic of differences between the two on demand routing protocols.

In contrast, Mohamed Amnai1, Youssef Fakhri and Jaafar Abouchabaka, in their paper (Amnai, Fakhri, & Abouchabaka, 2011) conducted a behavior study of OLSR proactive routing protocol using two traffic types multimedia (VBR) and CBR, by over various mobility models as Random Way Point, Random Direction and Mobgen Steady State. The experimental results illustrate that the behavior of OLSR change according to the model and the used traffics.

1.1.2 Energy Based Performance Comparisons

Energy consumption in ad hoc networks is a very important factor. For many ad-hoc networks, the nodes are small and portable, imposing stringent constraints on the battery size and power. Because batteries carried by each mobile node have limited power supply, processing power is limited, which in turn limits services and applications that can be supported by each node. This becomes a bigger issue in mobile ad hoc networks because, as each node is acting as both an end system and a router at the same time, additional energy is required to forward packets from other nodes.

Many researchers have developed “energy-aware” protocols for ad hoc network routing. Chang and Tassiulas (Chang & Tassiulas, 2000) proposed an algorithm to select the routes and power levels such that the time until the drain out of the batteries of the nodes is maximized. They proposed that in order to maximize the lifetime of the network, the traffic should be routed such that the energy is balanced among the nodes in proportion to their energy consumption, instead of routing to minimize absolute consumed power. Another proposal, by Ryu and Cho (Ryu & Cho, 2001), studied a routing scheme in home ad-hoc networks wherein the packets are routed though the outlet-plugged devices instead of the battery-powered ones to prolong the life of the batteries. Finally, Xu et. al. (Xu, Heidemann, & Estrin, 2000) proposed two algorithms, Basic Energy-Conserving Algorithm (BECA) and Adaptive Fidelity Energy-Conserving Algorithm (AFECA) to reduce the energy consumption of mobile nodes. BECA is an algorithm that puts the radio of a node into “sleep mode” for a certain duty cycle in order to reduce the idle or listening time energy consumption of the node. AFECA makes use of information of local node density to adjust the length of time the node is in sleep mode.

Other researchers did not implement any energy-efficiency algorithms, but instead, compared the relative performance of the more established routing protocols when there are energy constraints in the network. Juan Carlos Cano and Pietro Manzoni concentrate in their work (Cano & Manzoni, 2000) on the power consumption aspects of the routing protocols. More specifically, they measured and compared the energy consumption behavior of the Ad hoc On Demand Distance Vector (AODV), the Direct Source Routing (DSR), the Temporally Ordered Routing Algorithm (TORA) and the Destination Sequenced Distance Vector Routing (DSDV). Their basic methodologies consisted of first selecting the most representative parameters for a MANET, then defining and simulating a basic scenario and finally, by varying the selected parameters, generate and evaluate a wide enough different scenarios. The five selected parameters were the mobile node number, the moving area dimensions, the node’s mobility pattern, the number of actual traffic sources and the data traffic pattern. The simulation results presented in their paper were obtained using the ns-2 simulator.

E. Ahvar, and M. Fathy (Ahvar & Fathy, 2007) using the GLOMOSIM simulator compared the performance of Dynamic Source Routing (DSR), Ad Hoc On-Demand Distance Vector Routing (AODV), location-aided routing (LAR1) .Their evaluation was based on energy consumption in mobile ad hoc networks. The performance differentials were analyzed using varying network load, mobility, and network size. Based on the observations, they made recommendations about when the performance of either protocol can be best.

1.1.3 Mobility Based Performance Comparisons

Mobility pattern, in many previous studies was assumed to be random waypoint. Random waypoint is a simple model that is easy to analyze and implement. This has probably been the main reason for the wide spread use of this model for simulations. Realizing that random waypoint is too general a model, recent research has started focusing on alternative mobility models and protocol independent metrics to characterize them.

Yasser Kamal Hassan, Mohamed Hashim Abd El-Aziz and Ahmed Safwat Abd El-Radi in their paper (Hassan, Abd El-Aziz, & Abd El-Radi, 2010) presented a comparing performance of protocols for routing packets between wireless mobile hosts in an ad-hoc network. The scenarios which they implement consist of dynamic network size and different number of movement speed at invariable pause time which used an AODV and DSR from On-Demand protocols compared with DSDV from proactive table-driven routing protocols. Their goal was to compare the three routing protocols to each other, not to find the optimal performance possible in their scenarios, we observe that the mobility pattern does influence the performance of MANET routing protocols.

1.1.4 Multimedia transmissions Based Performance Comparisons

As the mobile and handheld devices are becoming even more popular, and the use of ad hoc networks is increasingly perceived as significant, there is substantial relative work by the research community, regarding the differences among the existing ad hoc routing protocols. The performance of these protocols has been tested for the case of general traffic but not in respect with to multimedia traffic and especially video transmission.

George Adam, Christos Bouras, Apostolos Gkamas, Vaggelis Kapoulas, Georgios Kioumourtzis and Nikos Tavoularis in their paper (Adam, Bouras, Gkamas, Kapoulas, Kioumourtzis, & Tavoularis, 2011) conducted a number of simulations in order to evaluate the performance of three of the most popular routing protocols, two reactive (AODV, DSR) and one proactive (OLSR) for different number of simultaneous video transmissions. They used the packet delivery ratio, the end-to-end delay, the packet delay variation (jitter) and the routing overhead as evaluation metrics. The mobility scenario simulated the environment of a modern city, where the vehicles (mobile nodes) are connected to each other and communicate. The vehicles are almost always moving, maximizing the routing process complexity.

1.1.5 Network size Based Performance Comparisons

Almost always the network protocols were simulated as a function of pause time (i.e. as a function of mobility), but never as a function of network size. David Oliver Jörg in his paper (Jörg, 2003), presented a performance comparison of four different mobile ad-hoc routing protocols (AODV, DSR, LAR1 and ZRP) as a function of network and area size. The main interest of the project was to test the ability of different routing protocols to react on network topology changes (for instance link breaks, node movement, and so on). Furthermore the focus was set on different network sizes, varying number of nodes and area sizes. His investigations did not include the protocol's operation under heavy load, e.g. its operation in congestion situations. Therefore only rather small packet sizes and one source node were selected.

1.1.6 Security Based Performance Comparisons

Security in MANETs is of prime importance in several scenarios of deployment such as battlefield, event coverage, etc.. However, very few researchers very have considered these protocols in highly demanding real-life scenarios which may impose seemingly contradicting constraints including security, reliability, performance, and power conservation.

Karthik Sadasivam, Vishal Changrani and T. Andrew Yang in their paper (Sadasivam, Changrani, & Yang, 2005) performed a scenario-based evaluation of three routing protocols– DSDV, DSR and SEAD. They also analyzed the tradeoffs between performance and security for specific scenarios of deployment. Their performance evaluation gives an insight into the applicability of the three protocols under consideration and helps identify which protocol is more suitable for a given scenario.

1.1.7 Scenario Based Performance Comparisons

A. Vasiliou and A.A. Economides have made simulation studies based on scenarios mostly or educational purposes. More specifically, they analyze the benefits of collaborative learning through educational trips with the use of Manets, (Vasiliou & Economides, Game-based learning using MANETs, 2007) (Vasiliou & Economides, MANET-based outdoor collaborative learning, 2008), they investigate the scalability and communication among multiple rescue teams (Vasiliou & Economides, Scalable multiple team cooperation using MANET multicasting, 2006), (Vasiliou & Economides, Rescue operations using coordinated multicast MANET, 2006) and they also have studied the environmental monitoring from scientists and the collection of data from sensors (Vasiliou & Economides, MANETs for environmental monitoring, 2006).

Finally, they have made performance evaluations for multicast Manets (Vasiliou & Economides, Evaluation of Multicasting Algorithms in Manets, 2005), (Vasiliou & Economides, Partitioning multicast MANETs, 2005), (Vasiliou & Economides, Multicast groups in MANETs, 2005), (Vasiliou & Economides, Performance of multicast groups moving towards the same direction, 2005).

Table 1 Previous Research Summary

		Simulation Parameters				
		Packet Size	**Network Size**	**Mobility**	**DSSS Rate**	**Traffic**
Simulation Metrics	**PDR**	(Vasiliou & Economides, 2007), (Vasiliou & Economides, 2008), (Ismail & Hassan, 2011), (Tuteja, Gujral, & Thalia, 2010)	(Hassan, Abd El-Aziz, & Abd El-Radi, 2010), (Jörg, 2003), (Vasiliou & Economides, 2005), (Vasiliou & Economides, 2006)	(Broch, Maltz, Johnson, Hu, & Jetcheva, 1998), (Hassan, Abd El-Aziz, & Abd El-Radi, 2010), (Jörg, 2003), (Vasiliou & Economides, 2007), (Vasiliou & Economides, 2005), (Vasiliou & Economides, 2008), (Vasiliou & Economides, 2006), (Al-Maashri & Ould-Khaoua, 2006), (Tuteja, Gujral, & Thalia, 2010)		(Adam, Bouras, Gkamas, Kapoulas, Kioumourtzis, & Tavoularis, 2011), (Vasiliou & Economides, 2007), (Vasiliou & Economides, 2005), (Vasiliou & Economides, 2008), (Tuteja, Gujral, & Thalia, 2010)
	Average Delay	(Vasiliou & Economides, 2007), (Vasiliou & Economides, 2008), (Tuteja, Gujral, & Thalia, 2010)	(Karthikeyan, Angayarkanni, & Sujatha, 2010), (Amnai, Fakhri, & Abouchabaka, 2011), (Hassan, Abd El-Aziz, & Abd El-Radi, 2010), (Jörg, 2003), (Vasiliou & Economides, 2005), (Vasiliou & Economides, 2006)	(Das, Castaneda, Yan, & Sengupta, 1998), (Das, Perkins, & Royer, 2000), (Hassan, Abd El-Aziz, & Abd El-Radi, 2010), (Jörg, 2003), (Sadasivam, Changrani, & Yang, 2005), (Vasiliou & Economides, 2007), (Vasiliou & Economides, 2005), (Vasiliou & Economides, 2008), (Vasiliou & Economides, 2006), (Lakshmikanth, Patel, & Gaiwak, 2009), (Al-Maashri & Ould-Khaoua, 2006), (Tuteja, Gujral, & Thalia, 2010)		(Adam, Bouras, Gkamas, Kapoulas, Kioumourtzis, & Tavoularis, 2011), (Vasiliou & Economides, 2007), (Vasiliou & Economides, 2005), (Vasiliou & Economides, 2008), (Tuteja, Gujral, & Thalia, 2010)
	Throughput	(Ismail & Hassan, 2011), (Tuteja, Gujral, & Thalia, 2010)	(Karthikeyan, Angayarkanni, & Sujatha, 2010), (Amnai, Fakhri, & Abouchabaka, 2011), (Amnai, Fakhri, & Abouchabaka, 2011), (Kanakaris, Ndzi, & Azzi, 2010)	(Lakshmikanth, Patel, & Gaiwak, 2009), (Al-Maashri & Ould-Khaoua, 2006), (Tuteja, Gujral, & Thalia, 2010)		(Tuteja, Gujral, & Thalia, 2010)
	Total Energy Consumption		(Cano & Manzoni, 2000), (Ahvar & Fathy, 2007), (Kanakaris, Ndzi, & Azzi, 2010)	(Cano & Manzoni, 2000), (Ahvar & Fathy, 2007) , (Djenouri, Derhab, & Badache, 2006)		(Cano & Manzoni, 2000), (Ahvar & Fathy, 2007)

Table 1 summarizes many of the papers that were mentioned in this section. They are categorized based in the simulation metrics and parameters which we study in the present thesis. As we can observe, many simulation studies have been made for the mobility and the network size. The total energy consumption has not been extensively studied like the other simulation metrics. We also have to mention that in many studies the traffic load is not determined as a combination of the interval and the packet size. In the present thesis we have defined those parameters separated and so we can determine how both of them affect the performance of the routing protocols. The most important observation is that the DSSS Rate has not been studied for any simulation metric in order to see how it affects the performance of the routing protocols.

1.2 Problem Statement

We briefly introduced the basic simulation studies that have been made the last years on MANET routing protocols. As it has already been mentioned the main disadvantage of these studies is that they are not based on real life scenarios. This is probably the main reason MANETs have not been used extensively in every day applications although they have significant advantages above traditional communication networks (cellular and infracture networks). Some of these advantages are:

- Use of ad hoc networks could increase mobility and flexibility, as ad-hoc networks can be brought up and torn down in very short time.
- Mobile ad hoc networks have better coverage in rough areas due to multi-hop relaying.
- Ad-hoc networks could be more economical in some cases as they eliminate fixed infrastructure costs and reduce power consumption at mobile nodes.
- Ad-hoc networks are more robust than conventional wireless networks because of their non-hierarchical distributed control and management mechanisms.
- Because of short communication links (node-to-node instead of node to a central base station), radio emission levels could be kept at low level. This increases spectrum reuse possibility or possibility of using unlicensed bands.
- Because of multi-hop support in ad-hoc networks, communication beyond Line Of Sight (LOS) is possible at high frequencies.

Despite the aforementioned and the potential application possibilities, ad-hoc networks are yet far from being deployed on large-scale commercial basis. Although various routing protocols are suggested and tested for mobile ad-hoc networks, performance metrics like throughput, delay and protocol overhead in relation to successfully transmitted data need better optimization. One single protocol will probably not be able to work efficiently across entire range of design parameters and operating conditions and that’s the reason why various real life scenarios should be tested. If each protocol could be assigned to a real life scenario with certain parameters, it would be much easier for those scenarios to be implemented.

1.3 Aim of research

The objective of the present master thesis is to evaluate the performance of different routing protocols for MANETs in realistic environments and to specify the best choice for each scenario and for certain parameters. Particularly for Greece, there are many applications where mobile ad hoc networks could be formed and perform even better than the traditional communication networks. Many are the application where traditional fracture networks cannot be formed due to the topology and where the mobile phone technology cannot be exploited in real-time applications.

As it is explained in the previous section, lots of theoretical work has been done during these last years in MANETs, but a large percentage of them evaluate protocols under non-realistic conditions: the transmission and receiver devices are not modeled according to commercial ones, but using the default parameters of the simulator; some of the mobility models used have interesting mathematical properties but do not describe the path a person or a car follows, etc. In the only environment where lots of research has been carried out under realistic condition is in the case of MANET used in vehicular networks (VANET), but there are much more environments where MANET can be used.

The overall goal of this thesis is to develop realistic scenarios which are located in regions around Greece for everyday life applications and through the simulation process of those scenarios to compare the most significant mobile ad hoc routing protocols. Of course under research those results could be adjusted in other worldwide locations which meet the same parameters.

1.4 Thesis Overview

The present thesis consists of 8 chapters and two appendixes. The thesis is organized as follows. In chapter 1 and 2 we provide a brief summary of mobile ad hoc networks. More specifically, we analyze the history, the main features and the applications of MANETs, the IEEE 802.11 protocols, the DSSS Rate technology and also the mobility models. In chapter 3 we explain the concept of routing in general and which are the problems with routing in MANETs. We also give a brief classification of the routing protocols in MANETs and we analyze the three basic routing protocols, AODV, DSDV and OLSR. Chapter 4 is the core of the thesis. We analyze the ns-3 simulator, the performance metrics used in the simulation, we present the real life scenarios and finally, we analyze the results of the simulation for each scenario. In chapter 5 the research conclusion and the future work is discussed. In chapter 6 are presented the references of the thesis. In the first appendix (Chapter 7) we present a brief terminology of general and ad hoc related terms. Finally, in the second appendix (Chapter 8) is the code which was used in the simulation for the three real life scenarios (simulation scripts).

2 Mobile Ad-Hoc Networks

Mobile ad hoc networks (MANETs) represent complex distributed systems that comprise wireless mobile nodes that can freely and dynamically self-organize into arbitrary and temporary, “ad-hoc” network topologies, allowing people and devices to seamlessly internetwork in areas with no pre-existing communication infrastructure, e.g., disaster recovery environments. Ad hoc networking concept is not a new one, having been around in various forms for over 20 years. Traditionally, tactical networks have been the only communication networking application that followed the ad hoc paradigm. Recently, the introduction of new technologies such as the Bluetooth, IEEE 802.11 and Hyperlan are helping enable eventual commercial MANET deployments outside the military domain. These recent evolutions have been generating a renewed and growing interest in the research and development of MANET.

2.1 History and Definition of Mobile ad Hoc Networks

The concept of mobile ad hoc networking is not a new one and its origins can be traced back to the DARPA Packet Radio Network project in 1972 (Freebersyser & Leiner, 2001). Then, the advantages such as flexibility, mobility, resilience and independence of fixed infrastructure, elicited immediate interest among military, police and rescue agencies in the use of such networks under disorganized or hostile environments. For a long time, ad hoc network research stayed in the realm of the military, and only in the middle of 1990, with the advent of commercial radio technologies, did the wireless research community become aware of the great potential and advantages of mobile ad hoc networks outside the military domain, witnessed by the creation of the Mobile Ad Hoc Networking working group within the IETF (IETF MANET Working Group). In our days because of the emergence of real-time applications and the widespread use of wireless and mobile devices, mobile Ad Hoc networks are receiving attention due to many potential military and civilian applications (Ade & Tijare, 2010).

Mobile Ad-hoc Networks are a collection of two or more devices equipped with wireless communications and networking capability. These devices can communication with other nodes that immediately within their radio range or one that is outside their radio range. For the later, the nodes should deploy an intermediate node to be the router to route the packet from the source toward the destination (Han, 2004).

Furthermore, devices are free to join or leave the network and they may move randomly, possibly resulting in rapid and unpredictable topology changes. In this energy-constrained, dynamic, distributed multi-hop environment, nodes need to organize themselves dynamically in order to provide the necessary network functionality in the absence of fixed infrastructure or central administration. Manets may operate by themselves or may be connected to the larger Internet [1].

Mobile nodes are autonomous units that are capable of roaming independently. Typical mobile ad hoc wireless nodes are Laptops, PDAs, Pocket PCs, Cellular Phones, Internet Mobile Phones, Palmtops or any other mobile wireless devices. Mobile ad hoc wireless devices are typically lightweight and battery operated. A mobile ad hoc network (MANET) is an adaptive, self-configurable, selforganizing, infrastructure-less multi-hop wireless network with unpredictable dynamic topologies (Cheng, Huang, & Du, 2006). By adaptive, self-configurable and self-organizing, we mean that an ad hoc network can be formed, merged together or partitioned into separated networks on the fly depending on the networking needs (Swarnapriyaa, Vinodhini, Anthoniraj, & Anand, 2011).

The specific characteristics and complexities, which are summarized in Table 1, impose many design challenges to the network protocols. In addition, these networks are faced with the traditional problems inherent to wireless communications such as lower reliability than wired media, limited physical security, time varying channels, interference, etc.

Table 2 Characteristics and complexities of mobile ad hoc networks

Autonomous and infrastructureless
Multi-hop routing
Dynamic network topology
Device heterogeneity
Energy constrained operation
Bandwidth constrained variable capacity links
Limited physical security
Network scalability
Self-creation, self-organization and self-administration

2.2 Main Features

A Mobile Ad Hoc Network consists of a group of mobile hosts forming a temporary network on wireless links without the aid of any centralized administration or standard support services regularly available on the wide-area network to which the host may normally be connected.

[1] http://en.wikipedia.org/wiki/Mobile_ad_hoc_network

The RFC 2501 by MANET working group in IETF (Corson & Macker, 1999) points out the following ones as some of the relevant characteristics of Manets:

1. Dynamic topologies: nodes can move freely in arbitrary directions and with capricious speed. Therefore, the network must adapt itself to unpredictable changes in its topology, which is typically multihop.
2. Bandwidth constrained: the restrictions imposed by the wireless channel, such as multiple access, multipath interference, noise, fading and limited availability of spectrum, together with the inherent problems the medium access control protocol has to deal with them, make the throughput for each node be much less than the radio's maximum data transmission rate.
3. Energy-constrained operation: the devices that form part of the Ad Hoc Networks may be power limited due to the circumstances of their functioning (as in sensors networks, for instance, where maximizing the average network life is a design criteria), hence, the routing algorithms must manage properly this issue, which can be complicated if *dozing mode* is accepted for the terminals.
4. Limited physical security: mobile wireless networks are susceptible to have security lacks and can be attacked quite easily. Existing security techniques are applied in the date link layer in order to reduce the risk, but some mechanisms can also be introduced in the network layer. On the other hand, the fact of being a decentralized network provides additional robustness against single point of failures.

In addition, some networks (e.g. mobile military networks or highway networks) may be relatively large (e.g. tens or hundreds of nodes per routing area), although the need for scalability is not unique to Manets. However, owing to the preceding characteristics, the mechanisms required to achieve scalability likely are more complicated. Actually, most of the existing protocols break down for large networks.

These characteristics create a set of underlying assumptions and performance concerns for protocol design that extend beyond those guiding the design of routing within the higher-speed, semi-static topology of other packet networks such as the fixed Internet.

2.3 Advantages and Drawbacks

Ad hoc networks are really an alternative to fixed networks in some operational situations, but an analysis of their advantages and drawbacks, that was summarized in (Navarro, 2001)[2 & 3], will help us know the applications and the contexts they may be useful for. Below, we present some of those (Anton, 2000).

[2] The Mobile Ad Hoc Networks Study Group (GEDOC) http://www.siam.dcc.ufmg.br/gedoc
[3] http://www.ietf.org/html.charters/manet-charter.html

Among the advantages of Manets are:

- Fast installation: the level of flexibility for setting up Manets is high, since they do not require any previous installation or infrastructure and, thus, they can be brought up and torn down in very short time.
- Dynamic topologies: nodes can arbitrarily move around the network and can disappear temporally from the Manet, so the network topology graph can be continuously changing at undetermined speed.
- Fault tolerance: owing to the limitations of the radio interfaces and the dynamic topology, Manets support connection failures, because routing and transmission control protocols are designed to manage these situations.
- Connectivity: the use of centralized points or gateways is not necessary for the communication within the Manet, due to the collaboration between nodes in the task of delivering packets.
- Mobility: the wireless mobile nodes can move at the same time in different directions. Although the routing algorithms deal with this issue, the performance simulations show that there is a threshold level of node mobility such that protocol operation begins to fail.
- Cost: Manets could be more economical in some cases as they eliminate fixed infrastructure costs and reduce power consumption at mobile nodes.
- Spectrum reuse possibility: owing to short communication links (node to- node instead of node to a central base station), radio emission levels could be kept at low level. This increases spectrum reuse possibility or possibility of using unlicensed bands.

Some of the problems Manets have are:

- Bandwidth constraints: as commented above, the capacity of the wireless links is always much lower than in the wired counterparts. Indeed, several Gbps are available for wired LAN, while, nowadays, the commercial applications for wireless LANs work typically around 2 Mbps.
- Processing capability: most of the nodes of the AND are devices without a powerful CPU. Furthermore, the network tasks such as routing and data transmission cannot consume the power resources of the devices, intended to play any other role, such as sensing functions.
- Energy constraints: the power of the batteries is limited in all the devices, which does not allow infinitive operation time for the nodes. Therefore, energy should not be wasted and that is why some energy conserving algorithms have been implemented (COMPOW (Narayanaswamy, Kawadia, Sreenivas, & Kumar, 2001), PARO (Gmez, Campbell, Naghshineh, & Bisdikian, 2001) and MBCR (Toh, 2001) are some examples). A study of

the energy saving and capacity improvement potential of power control in multihop wireless networks is done in (Monks, Ebert, Wolisz, & Hwu, 2001).

- High latency: when an energy conserving design has been applied it means that the nodes are sleeping or idle when they do not have to transmit any data. When the data exchange between two nodes goes through nodes that are sleeping, the delay may be higher if the routing algorithm decides that these nodes have to wake up.
- Transmission errors: attenuation and interferences are other effects of the wireless links that increase the error rate.
- Security: (Lundberg., 2000) analyses some of the vulnerabilities and attacks Manets can suffer. The authors divide the possible attacks in passive ones, when the attacker only attempts to discover valuable information by listening to the routing traffic; and active attacks, which occur when the attacker injects arbitrary packets into the network with some proposal like disabling the network. Other security issues such as availability, authenticity, integrity, confidentiality and privacy are discussed in (Vanhala, 2000).
- Location: the addressing is the another problem for the network layer in Manets, since the information about the location the IP addressing used in fixed networks offers some facilities for routing that cannot be applied in Manets. The way of addressing in Manets, of course, has nothing to do with the position of the node. In (Weniger & Zitterbart, 2002) a recent proposal on an IPv6-based addressing scheme for Manets can be seen.
- Roaming: the continuous changes in the network connectivity graph involve that the roaming algorithms of the fixed network are not applicable in Manets, because they are based on the existence of guaranteed paths to some destinations
- Commercially unavailable: Manets are yet far from being deployed on large-scale commercial basis.

2.4 Applications

Once known the features of ad hoc networks, the examples of their potential practical use are only limited by imagination. So, the set of applications for Ad ho Networks is diverse, from small, static networks that are constrained by power sources, to large-scale, mobile, highly dynamic networks. Typical applications are those in which survivable efficient and dynamic communications must be established. Table 2 provides an overview of present and future MANET applications (Chlamtac, Conti, & Liu, 2003).

Table 3 Mobile ad hoc network applications

Application	Possible scenarios/services
Tactical networks	• Military communication and operations • Automated battlefields
Emergency services	• Search and rescue operations • Disaster recovery • Replacement of fixed infrastructure in case of environmental disasters • Policing and fire fighting • Supporting doctors and nurses in hospitals
Commercial and civilian environments	• E-commerce: electronic payments anytime and anywhere • Business: dynamic database access, mobile offices • Vehicular services: road or accident guidance, transmission of • road and weather conditions, taxi cab network, inter-vehicle • networks • Sports stadiums, trade fairs, shopping malls • Networks of visitors at airports
Home and enterprise networking	• Home/office wireless networking • Conferences, meeting rooms • Personal area networks (PAN), Personal networks (PN) • Networks at construction sites
Education	• Universities and campus settings • Virtual classrooms • Ad hoc communications during meetings or lectures
Entertainment	• Multi-user games • Wireless P2P networking • Outdoor Internet access • Robotic pets • Theme parks
Sensor networks	• Home applications: smart sensors and actuators embedded in • consumer electronics • Body area networks (BAN) • Data tracking of environmental conditions, animal movements, chemical/biological detection
Context aware services	• Follow-on services: call-forwarding, mobile workspace • Information services: location specific services, time dependent services • Infotainment: touristic information
Coverage extension	• Extending cellular network access • Linking up with the Internet, intranets, etc.

2.5 IEEE 802.11 (WI-FI) Communication Protocol

The aim of the IEEE 802.11 standard (Ferro & Potorti, 2004) is to provide wireless connectivity to devices that require a quick installation, such as portable computers, PDAs, or generally mobile devices inside a WLAN (Wireless Local Area Network). 802.11 WLANs are already commonly used in several large vertical markets. The 802.11b standard is the first standard to make WLANs usable in the general workplace by providing robust and reliable 11 Mbps performance, five times faster than the original standard.

2.5.1 IEEE 802.11 and 802.11b Technology

As the globally recognized LAN authority, the IEEE 802 committee has established the standards that have driven the LAN industry for the past two decades, including 802.3 Ethernet, 802.5 Token Ring, and 802.3z 100BASE-T Fast Ethernet. In 1997, after seven years of work, the IEEE published 802.11, the first internationally sanctioned standard for wireless LANs. In September 1999 they ratified the 802.11b "High Rate" amendment to the standard, which added two higher speeds (5.5 and 11 Mbps) to 802.11 (3ComCorporation, 2000)[4].

With 802.11b WLANs, mobile users can get Ethernet levels of performance, throughput, and availability. The standards-based technology allows administrators to build networks that seamlessly combine more than one LAN technology to best fit their business and user needs.

Like all IEEE 802 standards, the 802.11 standards focus on the bottom two levels of the ISO model, the physical layer and data link layer (Figure 1). Any LAN application, network operating system, or protocol, including TCP/IP and Novell NetWare, will run on an 802.11-compliant WLAN as easily as they run over Ethernet.

[4] http://www.wlana.com/

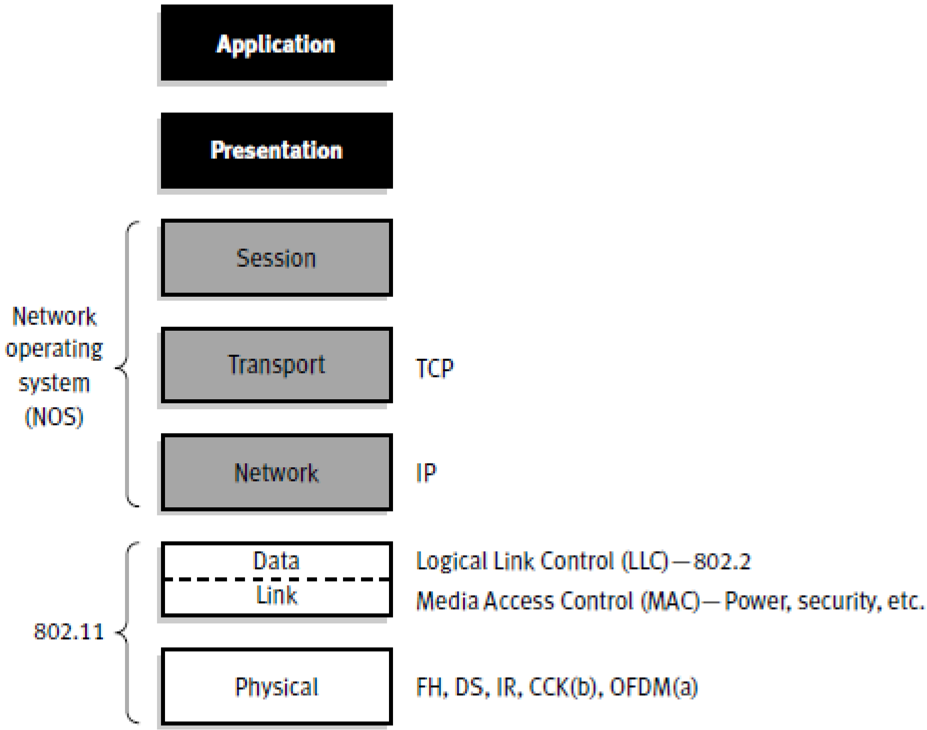


Figure 1 802.11 and the ISO Model

The basic architecture, features and services of 802.11b are defined by the original 802.11 standard. The 802.11b specification affects only the physical layer, adding higher data rates and more robust connectivity.

The 802.11 standard defines two modes: *infrastructure* mode and *ad hoc* mode. In infrastructure mode, the wireless network consists of at least one access point connected to the wired network infrastructure and a set of wireless end stations. This configuration is called a *Basic Service Set (BSS)*. An *Extended Service Set (ESS)* is a set of two or more BSSs forming a single subnetwork. Since most corporate WLANs require access to the wired LAN for services (file servers, printers, Internet links) they will operate in infrastructure mode.

Ad hoc mode (also called peer-to-peer mode or an Independent Basic Service Set, or IBSS) is simply a set of 802.11 wireless stations that communicate directly with one another without using an access point or any connection to a wired network.

This mode is useful for quickly and easily setting up a wireless network anywhere that a wireless infrastructure does not exist or is not required for services, such as a hotel room, convention center, or airport, or where access to the wired network is barred (such as for consultants at a client site).

2.5.2 Direct Sequence Spread Spectrum (DSSS) Technology

The key contribution of the 802.11b addition to the wireless LAN standard was to standardize the physical layer support of two new speeds, 5.5 Mbps and 11 Mbps. To accomplish this, Direct Sequence Spread Spectrum (DSSS) had to be selected as the sole physical layer technique for the standard.

DSSS is a modulation method applied to digital signals (Goldsmith, 2005). It increases the signal bandwidth to a value much larger than needed to transmit the underlying information. In DSSS, spreading codes that are independent of the original signal are used to achieve the goal of bandwidth expansion. Both a sender and a receiver agree on a spreading code, which is regarded as a shared secret between them. A spreading code is usually a sequence of bits valued 1 and −1 (polar) or 1 and 0 (non-polar), which has noise-like properties (Liu, Ning, Dai, & Liu, 2010).

The resulting signal wave looks much like white noise. This white noise can be filtered at the receiving end in order to recover the original data. This filtering happens by again multiplying the same pseudo-random sequence by the received signal (because $1 \times 1 = 1$, and $-1 \times -1 = 1$). This process, known as “de-spreading”, mathematically constitutes a correlation of the transmitted pseudo-random sequence with the receiver’s assumed sequence. For allowing de-spreading to work correctly, the transmit and received sequences must synchronized (Hogie, 2007).

To support very noisy environments as well as extended range, 802.11b WLANs use dynamic rate shifting, allowing data rates to be automatically adjusted to compensate for the changing nature of the radio channel. Ideally, users connect at the full 11 Mbps rate. However when devices move beyond the optimal range for 11 Mbps operation, or if substantial interference is present, 802.11b devices will transmit at lower speeds, falling back to 5.5, 2, and 1 Mbps. Likewise, if the device moves back within the range of a higher-speed transmission, the connection will automatically speed up again. Rate shifting is a physical layer mechanism transparent to the user and the upper layers of the protocol stack.
The main benefits from the DSSS Rate technology are:

- Resistance to intended or unintended jamming
- Sharing of a single channel among multiple users
- Reduced signal/background-noise level hampers interception
- Determination of relative timing between transmitter and receiver

2.6 Mobility Models

The mobility model is designed to describe the movement pattern of mobile users, and how their location, velocity and acceleration change over time. Since mobility patterns may play a significant role in determining the protocol performance, it is desirable for mobility models to emulate the movement pattern of targeted real life applications in a reasonable way.

Otherwise, the observations made and the conclusions drawn from the simulation studies may be misleading. Thus, when evaluating MANET protocols, it is necessary to choose the proper underlying mobility model.

In the previous studies on mobility patterns in wireless cellular networks (Lam, Cox, & Widom, 1999) (Markdoulidakis, Lyberopoulos, Tsirkas, & Sykas, 1997), researchers mainly focus on the movement of users relative to a particular area (i.e., a cell) at a macroscopic level, such as cell change rate, handover traffic and blocking probability. However, to model and analyze the mobility models in MANET, we are more interested in the movement of individual nodes at the microscopic-level, including node location and velocity relative to other nodes, because these factors directly determine when the links are formed and broken since communication is peer-to-peer (Bai & Helmy, 2006).

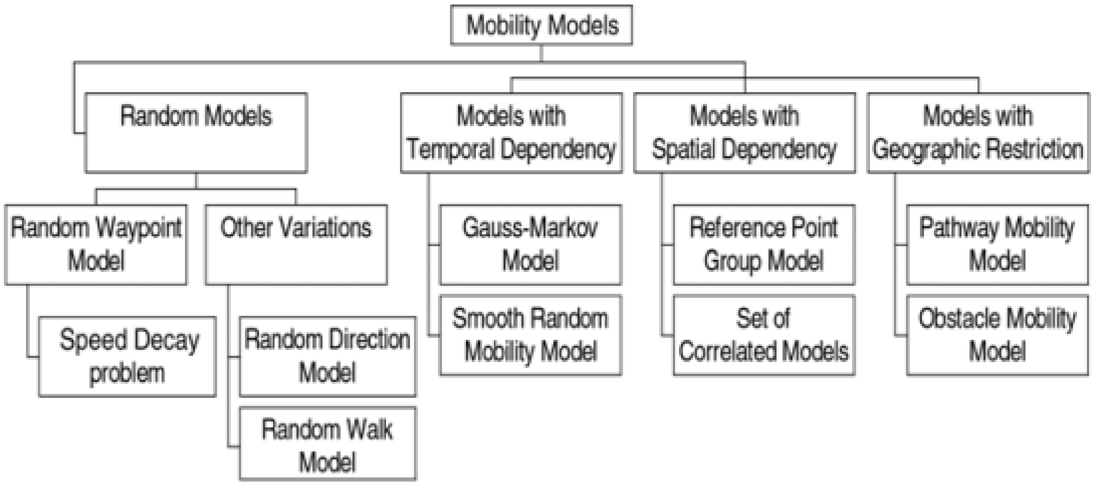


Figure 2 Mobility models classification.

In Figure 2 we provide a categorization for various mobility models into several classes based on their specific mobility characteristics. For some mobility models, the movement of a mobile node is likely to be affected by its movement history. We refer to this type of mobility model as mobility models with temporal dependency. In some mobility scenarios, the mobile nodes tend to travel in a correlated manner. We refer to such models as mobility models with spatial dependency. Another class is the mobility models with geographic restriction, where the movement of nodes is bounded by streets, freeways or obstacles.

2.6.1 Random-Based Mobility Models

In random-based mobility models, the mobile nodes move randomly and freely without restrictions. To be more specific, the destination, speed and direction are all chosen randomly and independently of other nodes. This kind of model has been used in many simulation studies.

2.6.1.1 Random Direction Mobility Model

The Random Direction Mobility Model (Royer, Melliar-Smith, & Moser, 2000) was created in order to overcome a flaw discovered in the Random Waypoint Mobility Model. MNs using the Random Waypoint Mobility Model often choose new destinations, and the probability of choosing a new destination that is located in the center of the simulation area, or requires travel through the middle of the simulation area, is high. Royer states that MNs moving with the Random Waypoint Mobility Model appear to converge, disperse, converge again, etc.1 In order to alleviate this type of behavior and promote a semi-constant number of neighbors, the Random Direction Mobility Model was developed. In this model, MNs choose a random direction in which to travel instead of a random destination. After choosing a random direction, an MN travels to the border of the simulation area in that direction. As soon as the boundary is reached the MN stops for a certain period of time, chooses another angular direction (between 0 and 180 degrees) and continues the process. Figure 3 shows an example path of an MN, which begins at the center of the simulation area or (250,250), using the Random Direction Mobility Model (Saad & Zukarnain, 2009).

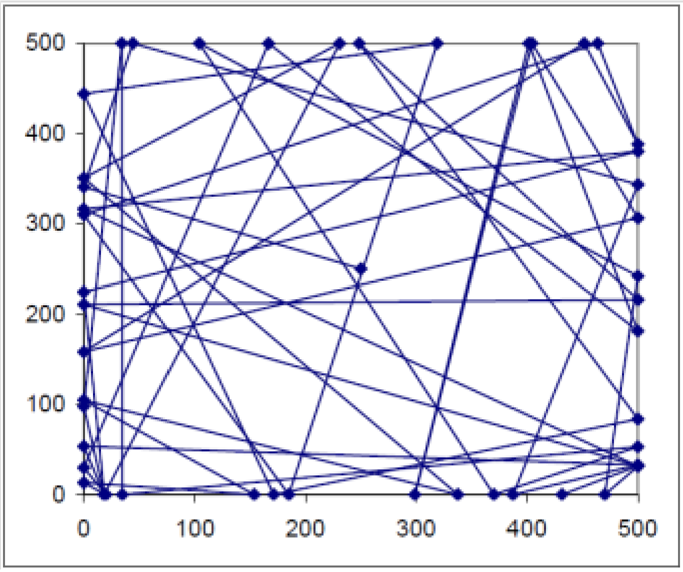


Figure 3 Traveling pattern of an MN using the Random Direction Mobility Model

2.6.2 Mobility Models with Temporal Dependency

Mobility of a node may be constrained and limited by the physical laws of acceleration, velocity and rate of change of direction. Hence, the current velocity of a mobile node may depend on its previous velocity. Thus the velocities of single node at different time slots are 'correlated'. We call this mobility characteristic the Temporal Dependency of velocity.

2.6.2.1 Gauss-Markov Mobility Model

The Gauss-Markov Mobility Model was first introduced by Liang and Haas (Liang & Haas, 1999) and widely utilized (Camp, Boleng, & Davies, 2002) (Hu & Johnson, 2000). The Gauss-Markov mobility model is a relatively simple memory-based model with a single tuning parameter, alpha α, which determined the amount of memory and variability in node movement. Setting α between zero and one allows the tuning of the Gauss-Markov model with degrees of memory and variation. In order to analyze the impact of α on mobility, many simulations have been conducted using the parameters given in Table 3. Figure 4 shows variation in node movement with varying values of α. We observe that as α increases, the node paths become less random and more predictable (Bai & Helmy, 2006) (Broyles, Jabbar, & Sterbenz, 2011).

Table 4 Simulation setup while studying the impact of α.

Simulation parameter	Value
MeanVelocity	[800 1200] m/s
MeanDireaction	[0 2π] radians
MeanPitch	[-0.05 0.05] radians
NormalVelocity	N(0, 80)
NormalDirection	N(0, 1.4)
NormalPitch	N(0, 0.2)

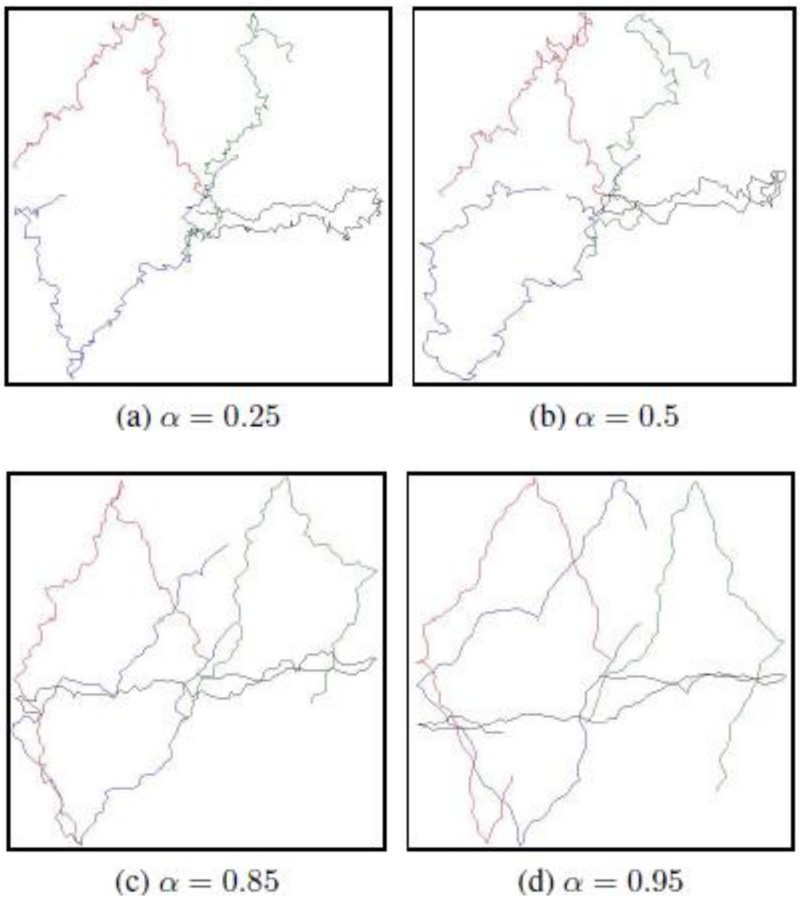


Figure 4 Gauss-Markov model with a = 0.25, 0.5, 0.85, 0.95

3 Routing in Mobile Ad-Hoc Networks

Routing is the act of moving information from a source to a destination in a network. During this process, at least one intermediate node within the network is encountered. The routing concept basically involves, two activities: firstly, determining optimal routing paths and secondly, transferring the information groups (called packets) through a network. The later concept is called as packet switching which is straight forward, and the path determination could be very complex (Cisco, 2002).

Routing protocols use several metrics to calculate the best path for routing the packets to its destination. These metrics are a standard measurement that could be number of hops, which is used by the routing algorithm to determine the optimal path for the packet to its destination. The process of path determination is that, routing algorithms initialize and maintain routing tables, which contain the total route information for the packet. This route information varies from one routing algorithm to another.

Routing tables are filled with a variety of information which is generated by the routing algorithms. Most common entries in the routing table are IP-address prefix and the next hop. Routing table's Destination/next hop associations tell the router that a particular destination can be reached optimally by sending the packet to a router representing the "next hop" on its way to the final destination and ip-address prefix specifies a set of destinations for which the routing entry is valid for.

Switching is relatively simple compared with the path determination. The concept of switching is like, a host determines like it should send some packet to another host. By some means it acquires the routers address and sends the packet addressed specifically to the routers MAC address, with the protocol address of the destination host. The router then examines the protocol address and verifies whether it know how to transfer the data to its destination. If it knows how to transfer the data then it forwards the packet to its destination and if it doesn't then it drops the packet.

In mobile ad-hoc networks where there is no infrastructure support as is the case with wireless networks, and since a destination node might be out of range of a source node transmitting packets; a routing procedure is always needed to find a path so as to forward the packets appropriately between the source and the destination. Within a cell, a base station can reach all mobile nodes without routing via broadcast in common wireless networks. In the case of ad-hoc networks, each node must be able to forward data for other nodes. This creates additional problems along with the problems of dynamic topology which is unpredictable connectivity changes (Feeney, 1999), (Schiller, 2000).

3.1 Problems with routing in Mobile Ad-hoc Networks

The problems with routing in Mobile Ad-hoc Networks are the following:

- Asymmetric links: Most of the wired networks rely on the symmetric links which are always fixed. But this is not a case with ad-hoc networks as the nodes are mobile and constantly changing their position within network. For example consider a MANET where node B sends a signal to node A but this does not tell anything about the quality of the connection in the reverse direction (Schiller, 2000).
- Routing Overhead: In wireless ad hoc networks, nodes often change their location within network. So, some stale routes are generated in the routing table which leads to unnecessary routing overhead.
- Interference: This is the major problem with mobile ad-hoc networks as links come and go depending on the transmission characteristics, one transmission might interfere with another one and node might overhear transmissions of other nodes and can corrupt the total transmission.
- Dynamic Topology: This is also the major problem with ad-hoc routing since the topology is not constant. The mobile node might move or medium characteristics might change. In ad-hoc networks, routing tables must somehow reflect these changes in topology and routing algorithms have to be adapted. For example in a fixed network routing table updating takes place for every 30sec (Schiller, 2000). This updating frequency might be very low for ad-hoc networks.

3.2 Classification of routing Protocols in MANET's

Classification of routing protocols in MANET's can be done in many ways, but most of these are done depending on routing strategy and network structure (Hong, Xu, & Gerla, 2002) (Royer & Toh, 1999). According to the routing strategy the routing protocols can be categorized as Pro-active (table-driven) and Reactive (on-demand), while depending on the network structure these are classified as flat routing, hierarchical routing and geographic position assisted routing (Hong, Xu, & Gerla, 2002). Both the Table-driven and Reactive protocols come under the Flat routing category[5] (Figure 5).

[5] http://en.wikipedia.org/wiki/List_of_ad_hoc_routing_protocols

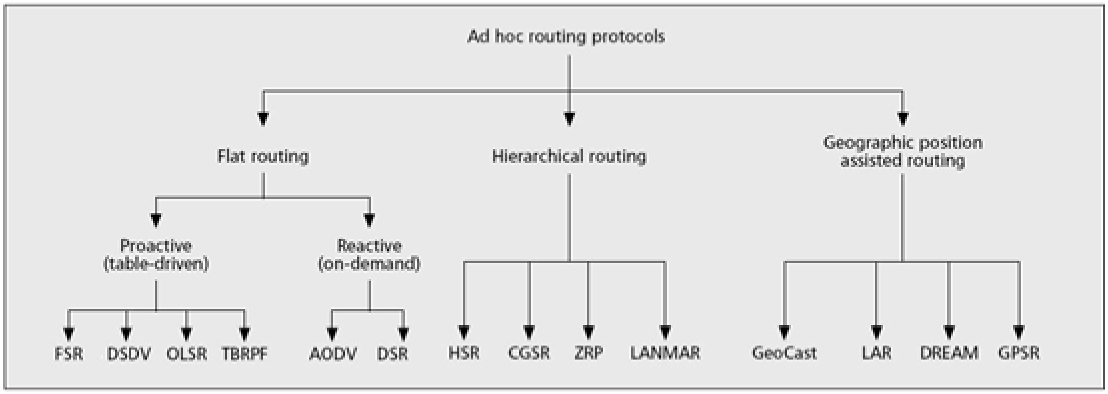


Figure 5 Classification of Routing Protocols In Mobile Ad-hoc Networks

3.2.1 Routing in Flat Network Structure

In a flat structure, all nodes in a network are at the same level and have the same routing functionality. Flat routing is simple and efficient for small networks. The problem is that when a network becomes large, the volume of routing information will be large and it will take a long time for routing information to arrive at remote nodes (Khan, Zaman, & Reddy, 2008).

The protocols that we review here fall into two categories, namely, proactive routing and reactive routing. Many proactive protocols stem from conventional link state routing. Reactive routing, on the other hand, is a more new emerging routing philosophy in the ad hoc area than proactive. It differs from conventional routing protocols in that no routing activities and no permanent routing information is maintained at network nodes if there is no communication, thus providing a scalable routing solution to large populations.

3.2.1.1 On-Demand or Reactive Protocols

A network using an on-demand protocol will not maintain correct routing information on all nodes for all times. Instead, such routing information is obtained on demand. If a node wants to transmit a message, and does not have enough routing information to send the message to the destination, the required information has to be obtained (unless the protocol is using directly a flooding approach to deliver the messages). The node needs at least to know the next hop (among its neighbors) for the packet. Although the node could just broadcast the packet to all neighbors this leads to serious congestion in many cases. However, such broadcasts must be used in a route discovery process, since there is no next-hop information available, yet.

Usually this consists of a broadcast message from the originating node, indicating the desired route. Nodes which have the required information will respond to the originating node, which will eventually choose a route from the replies it received. The broadcast may be limited to travel only a few hops first, before a net-wide broadcast will be issued (which would flood the whole network).

Of course, the route request and selection process must be finished, before the message can be sent. This leads to an initial setup delay for messages, if their route is not known to the node. To limit the impact of this delay, most protocols will use a route cache for once established routes. However, the information in this cache will time out, since in a mobile environment, the routes will be invalid after some time.

Clearly, applications that are used over an on-demand routing protocol need to be tolerant for such an initial setup delay.

The advantage of on-demand routing protocols lies in the fact that the wireless channel (a scarce resource) does not need to carry a lot of routing overhead data for routes, which are not even used. This advantage may diminish in certain scenarios where there is a lot of traffic to a large variety of nodes.

Thus the scenario will have a very significant impact on the performance. In such a scenario with lots of traffic to many nodes, the route-setup traffic can grow larger than constant background traffic to maintain correct routing information on each node. Still, if enough capacities would be available, the reduced efficiency (increased overhead) might not affect other performance measures, like throughput or latency.

I also consider some location based protocols as on-demand protocols, since they determine the direction in which to send the packet on demand and some protocols may even initiate a location query of the destination nodes for their packets on demand (Lang, 2006).

Thus, examples for on-demand protocols are the following: ABR, AODV, CEDAR, DSR, FORP, GEDIR, LAR, SSR, WAR.

3.2.1.2 Proactive Protocols

Proactive routing protocols will try to maintain correct routing information on all nodes in the network at all times. This can be achieved in different ways, and thus divides the protocols into two subclasses: event driven and regular updated protocols (Lang, 2006).

Event driven protocols will not send any routing update packets, if no change in topology occurs. Only if a node detects a change of the topology (usually a change in the neighbor set, or the reception of a message indicating a change in some other nodes neighbor set), this is reported to other nodes, according to the strategy of the routing protocol.

Protocols that are updated in regular intervals will always send their topology information to other nodes at regular intervals. Many link state protocols work in such a manner (but varying the maximum distance of an update message with the length of the interval). Nodes farther away get updates less frequently than close nodes, thus balancing the load imposed on the network.

Proactive protocols of either subclass impose a fixed overhead to maintain the routing tables. Even if many of the entries are not used at all. Their advantage is that the routes can be used at once and there is no setup delay.

Many studies (Jacquet & Viennot, June 2000) compare “flooding protocols” with “hello protocols” (those that periodically announce their neighbors and routes) in terms of overhead in an analytical way.

Event driven proactive routing protocols are the following: CBRP, CGSR, DSDV, GSR, LMR, TORA and WRP.

Regular updated protocols are: DDR, FSLS, FSR, GPSR, LANMAR, OLSR, STAR and TBRPF.

A comparison of these two categories of routing protocols is presented, highlighting their features, differences, and characteristics in Table 4 (Mohseni, Hassan, Patel, & Razali, 2010).

Table 5 Comparison of Proactive and Reactive Routing Protocols in Manet.

Routing class	Proactive(Table-Driven)	Reactive(On-Demand)
Routing structure	Both Flat and hierarchical structures	Mostly Flat
Availability of route	Always available	Determined when needed
Control Traffic volume	Usually high	Lower than proactive routing protocols
Periodic updates	Yes, some may use conditional.	Not required. Some nodes may require periodic beacons
Control Overhead	High	Low
Route acquisition delay	Low	High
Storage Requirements	High	Depends on the number of routes kept or required. Usually lower than proactive protocols
Bandwidth requirement	High	Low
Power requirement	High	Low
Delay level	Small since routes are predetermined	Higher than proactive
Scalability problem	Usually up to 100 nodes.	Source routing protocols up to few hundred nodes. Point-to-point may scale higher.

Handling effects of mobility	Occur at fixed intervals.	Usually updates ABR introduced LBQ AODV uses local route discovery
Quality of service support	Mainly shortest path as the QoS metric	Few can support QoS, although most support shortest path

3.2.2 Hierarchical Routing Protocols

Typically, when wireless network size increase (beyond certain thresholds), current "flat" routing schemes become infeasible because of link and processing overhead. One way to solve this problem and to produce scalable and efficient solutions is hierarchical routing. An example of hierarchical routing is the Internet hierarchy, which has been practiced in wired network for a long time. Wireless hierarchical routing is based on the idea of organizing nodes in groups and then assigning nodes different functionalities inside and outside of a group. Both routing table size and update packet size are reduced by including in them only part of the network (instead of the whole), thus control overhead is reduced. The most popular way of building hierarchy is to group nodes geographically close to each other into explicit clusters. Each cluster has a leading node (clusterhead) to communicate to other nodes on behalf of the cluster. An alternate way is to have implicit hierarchy. In this way, each node has a local scope. Different routing strategies are used inside and outside the scope. Communications pass across overlapping scopes. More efficient overall routing performance can be achieved through this flexibility. As mobile nodes have only a single omnidirectional radio for wireless communications, this type of hierarchical organization will be referred to as "logical hierarchy" to distinguish from the physically hierarchical network structure.

3.2.3 Geographic Position Information Assisted Routing

The advances in the development of Global Positioning System (GPS) nowadays make it possible to provide location information with a precision in the order of a few meters. They also provide universal timing. While location information can be used for directional routing in distributed ad hoc systems, the universal clock can provide global synchronizing among GPS equipped nodes. Research has shown that geographical location information can improve routing performance in ad hoc networks. Additional concern must be taken into account in a mobile environment, i.e., locations may not be accurate by the time the information is used. All the protocols surveyed below assume that the nodes know their positions.

3.3 Ad-hoc On-Demand Distance Vector (AODV) Protocol

AODV is a very simple, efficient, and effective routing protocol for Mobile Ad-hoc Networks which do not have fixed topology. This algorithm was motivated by the limited bandwidth that is available in the media that are used for wireless communications. It borrows most of the advantageous concepts from DSR and DSDV algorithms. The on demand route discovery and route maintenance from DSR and hop-by-hop routing, usage of node sequence numbers from DSDV make the algorithm cope up with topology and routing information. Obtaining the routes purely on-demand makes AODV a very useful and desired algorithm for MANETs (Gorantala, 2006).

3.3.1 Introduction to AODV

The information in this section concerning the Ad Hoc On-Demand Distance Vector Protocol (AODV) protocol is taken from the RFC (Perkins, Belding-Royer, & Das, Ad hoc On-Demand Distance Vector (AODV) Routing, 2003). AODV is a reactive protocol, i.e., so the routes are created and maintained only when they are needed. The routing table stores the information about the next hop to the destination and a sequence number which is received from the destination and indicating the freshness of the received information. Also the information about the active neighbors is received throughout the discovery of the destination host. When the corresponding route breaks, then the neighbors can be notified.

The route discovery is used by broadcasting the RREQ message to the neighbors with the requested destination sequence number, which prevents the old information to be replied to the request and also prevents looping problem, which is essential to the traditional distance vector protocols (Hong, Xu, & Gerla, 2002). The route request does not add any new information about the passed hosts only it increases its hop metric. Each passed host makes update in their own routing table about the requested host. This information helps the destination reply to be easily routed back to the requested host. The route reply use RREP message that can be only generated by the destination host or the hosts who have the information that the destination host is alive and the connection is fresh.

New version of the AODV routing protocol (Perkins, Belding-Royer, & Das, Ad hoc On-Demand Distance Vector (AODV) Routing, 2003) has also a feature that only the destination host can reply to the sent request. When the reply is sent back to the requested host the actual hop metric is counted. The intermediate hosts records information about the replied host upon receiving the reply message. The hosts must record and forward new information only when the sequence number is greater or if the sequence number is the same and hop metric is smaller. The additional RREP-ACK message must be sent in response to the RREP message when the message has an active acknowledgment option. The acknowledgment option is set up when there is possibility that the route may be unidirectional. This feature enables that the unidirectional links can be detected. When the breakage of the route is noticed the host sends RERR message to the neighbors. The Hello message is periodically sent for maintaining the route information.

Usually messages are transmitted by using IP limited broadcast address, but the messages are checked for the content so that they will not be broadcasted throughout the entire network. Some of the messages are supposed to be spread widely in the network, for example route request message (RREQ). So their distribution is restricted by the TTL field in the IP header. Usually the fragmentation of the IP packet is not required (Huhtonen, 2004).

3.3.2 Routing

3.3.2.1 Sequence numbers

The sequence numbers are the key idea for removing the old and invaluable information from the network. The sequence number act as timestamps and prevent this distance vector protocol from the loop problem (Huishan, Huimin, & Nam). The destination sequence number for each possible destination host is stored in the routing table. The destination sequence numbers are updated in the routing table when the host receives the message with the greater sequence number. The host can change the destination sequence number in the routing table if it is offering a new route to itself or if some route expires or simply breaks.

The host also keeps its own sequence number, which must be incremented only in two different cases: before it sends RREQ message and when the host sends a RREP message responding to the RREQ message. In the second case the sequence number must be incremented to the maximum of the current sequence number and the sequence number in the received RREQ message. The sequence numbers must be treated as unsigned integers so that the possible rollovers can occur, AODV protocol supports the sequence number to be rolled over without any problems.

3.3.2.2 RREQ, RREP and RREP-ACK messages

The route request message (RREQ) is sent when the host does not know the route to the needed destination host or the existed route is expired. The RREQ message includes the destination sequence number which is the last known sequence number of the destination host entry found in the routing table. If there contains no entry for the destination host, then the unknown sequence number flag must be set. The RREQ message also contains the requesting hosts sequence number, which must be incremented beforehand. The RREQ ID field is incremented by one which is found from the last used RREQ message, which was sent by this host. Also the hop count metric must be set to zero and before sending the RREQ message the

RREQ ID and its own address must be saved to the buffer for the specified amount of time, so that it recognize the replies.

There is possibility that some hosts can be unidirectional then the G field can be set in RREQ message, so that every intermediate host will generate the RREP message and unicast it to the requesting host. Also the intermediate host must generate the gratuitous RREP to the destination host. There is a limit of RREQ messages that the host can send per minute, waiting before retransmitting the RREQ message, and number of RREQ message retries it can send overall. All the repeat attempts must be sent using binary exponential backoff. The expanding ring search technique is used for preventing the RREQ messages from unnecessary spreading out through the network for more information about the technique is found from.

First when the host receives RREQ message, it checks the time period between the last RREQ messages from the same host and discards the message if it is under the specified limit. Next host increases the hop count by one in the RREQ message and makes update in own routing table basing on the sequence number and the requested host's address. Also the hop count is copied from the RREQ message. The host marks that the route is valid to requested host and adds information about the next hop specifying to which host the message should be forwarded to. Host needs to count the lifetime of the route to the requested host. The host must set the destination sequence number in the RREQ message if the sequence number is greater in the routing table than in the received message, but the host should not modify the sequence number in the routing table. Lastly the host should broadcast the request and decrease its TTL field in the IP header.

The host can generate the route reply message (RREP) if the destination is the host itself or if the route to the destination is valid and has the same or greater destination sequence number, but only if the D field is not set. D field in the RREQ message indicates that only the destination host can reply to the RREQ message. When generating the RREP message host copies the destination address and the requested host's sequence number to the corresponding RREP message's fields. If the receiver is the destination host then its own sequence number is incremented and copied to the destination sequence number field. In addition, the hop count is set to zero and the lifetime field of the RREP message is set to the initial timeout value of the host. If the receiver is the intermediate host, then it just copies destination sequence number from the routing table and adds the host address from where it has received RREQ message to the destination address field. Also the host must add the hop count with the lifetime from the routing table to the RREP. The lifetime is calculated by subtracting the current time and the expiration time from the routing table. When the RREP message is created it is sent using unicast to the next hop in order to be delivered to the requested host. The hop count metric is incremented along the path, so at the end, it corresponds to the actual distance between the hosts.

The gratuitous RREP is like the original RREP only it is sent to the destination host and all of the fields are generated in the same manner only gratuitous RREP destination address is set to the requested host's address. If the gratuitous node is sent to the destination node and the destination node has already sent its own RREQ message, then the contents of the RREQ message and RREP message which was sent in response to the earlier requested host are actually the same.

When the host receives the RREP message it searches for the previous hop and increases hop metric by one. If there is no routing entry for the previous hop, then the route is created but without a valid sequence number. Also it is necessary that the route to the destination host is created in the routing table. First the host must compare the destination sequence numbers. The routing table entry is modified only in the following situations: the sequence number is marked as invalid in routing table, the destination sequence number is greater than the routing table entry and the route is marked as valid, the sequence number is the same but the route is marked as inactive, the sequence number is same and the hop count metric is smaller than the information in the routing table. If the routing table is updated or created then the route must be marked as active and the destination sequence number field as valid. Also in the routing table the next hop is assigned to the host address from which the RREP message was received. The hop count is increased and the expiry time is set to current time plus the lifetime from the RREP message. The destination sequence number is copied from the message. Finally RREP message is forwarded to the requester using the next hop address from the routing table. If the address to which the RREP message is forwarded can have errors or maybe unidirectional then the A flag is set, which correspond to the receiver of the message to generate the RREP-ACK message back to the sender.

3.3.2.3 RERR messages, route expiry and route deletion

When the link breakage happens the host must invalidate the existing route in the routing table entry. The host must list the affected destinations and determine which neighbors can be affected with this breakage. Finally the host must send the route error (RERR) message to the corresponding neighbors. The RERR message can be broadcasted if there are many neighbors which need that information or unicasted if there is only one neighbor. The host can also iteratively unicast the message to needed neighbors if the broadcast is not possible. However, iterative unicasting must be considered as a single broadcast RRER message, so the RERR messages per second limit, is essential.

If the host detects the link breakage of the active route, then the host makes a list of unreachable destinations based on the routing table entries where the unreachable neighbor acts as a next hop address. If host gets RERR messages, then the unreachable destinations is consisted from the routing table which has same addresses as in RERR message and routing table next hop address entries. The destination sequence numbers for the entries in the routing table for the unreachable destinations must be incremented or if the host received RERR message, then simply copied from it. After this the entry for the unreachable hosts must be set to invalid lifetime. Lifetime is set to the current time plus specific deletion time, so that the entry is not deleted from the routing table before the lifetime expires. Then the RERR message with the unreachable destinations should be unicasted for one neighbor or broadcasted to the many neighbors with TTL value set to 1. The DestCount field in the RERR message describes the number of the unreachable host addresses.

3.3.2.4 Repairing

When the link breakage occurs then the host can try to locally repair the link if the destination is no further than specified amount of hops. In order to repair the link the host increase the destination sequence number and broadcasts the RREQ message to the host. The TTL for the IP header must be calculated, so that locally repair process would not spread throughout the network. The host waits for the RREP messages to its RREQ message for specified amount of time. If the RREP message is not received, then it changes the routing table status for the entry to invalid. If host receives the RREP message then the hop count metric is compared. If the hop metric from the message is greater than the previous one then the RERR with the N field set up is broadcasted. The N field in the RERR message indicates that the host has locally repaired the link and the entry in the table should not be deleted. The received RREP message is handled as original RREP message. The repairing of the link before the data is sent to unavailable host is a proactive repairing. Proactive repairing can be inefficient because the risk of repairing the routes that are not used anymore. So the proactive repairing can be used basing on the local traffic and the workload of the network.

3.3.2.5 Hello messages

Although AODV is a reactive protocol it uses the Hello messages periodically to inform its neighbors that the link to the host is alive. The Hello messages are broadcasted with TTL equals to 1, so that the message will not be forwarded further. When host receives the Hello message it will update the lifetime of the host information in the routing table. If the host does not get information from the host’s neighbor for specified amount of time, then the routing information in the routing table is marked as lost. This action generates needed RRER message to inform other hosts of the link breakage. The routes that were created by the Hello message and were not used for any routing actions should not generate the RERR message when the link breakage occurs.

3.3.2.6 Routing table structure

This is the main data structure where all needed information about the routes is stored. The routing table must include at least the following fields: destination address, destination sequence number, hop count, next hop, lifetime, precursor list, and route state. The precursor list contains the information about which hosts can possible forward the messages to this route. Precursor list contains the information to which neighbor the errors should be forwarded when the possible break occurs.

3.3.3 Interesting concepts of AODV

The concepts of AODV that make it desirable for MANETs with limited bandwidth include the following:

- Minimal space complexity: The algorithm makes sure that the nodes that are not in the active path do not maintain information about this route. After a node receives the RREQ and sets a reverse path in its routing table and propagates the RREQ to its neighbors, if it does not receive any RREP from its neighbors for this request, it deletes the routing info that it has recorded.
- Maximum utilization of the bandwidth: This can be considered the major achievement of the algorithm. As the protocol does not require periodic global advertisements, the demand on the available bandwidth is less. And a monotonically increased sequence number counter is maintained by each node in order to supersede any stale cached routes. All the intermediate nodes in an active path updating their routing tables also make sure of maximum utilization of the bandwidth. Since, these routing tables will be used repeatedly if that intermediate node receives any RREQ from another source for same destination. Also, any RREPs that are received by the nodes are compared with the RREP that was propagated last using the destination sequence numbers and are discarded if they are not better than the already propagated RREPs.
- Simple: It is simple with each node behaving as a router, maintaining a simple routing table, and the source node initiating path discovery request, making the network self-starting.
- Most effective routing info: After propagating an RREP, if a node finds receives an RREP with smaller hop-count, it updates its routing info with this better path and propagates it.
- Most current routing info: The route info is obtained on demand. Also, after propagating an RREP, if a node finds receives an RREP with greater destination sequence number, it updates its routing info with this latest path and propagates it.
- Loop-free routes: The algorithm maintains loop free routes by using the simple logic of nodes discarding non better packets for same broadcast-id.

- Coping up with dynamic topology and broken links: When the nodes in the network move from their places and the topology is changed or the links in the active path are broken, the intermediate node that discovers this link breakage propagates an RERR packet. And the source node re-initializes the path discovery if it still desires the route. This ensures quick response to broken links.
- Highly Scalable: The algorithm is highly scalable because of the minimum space complexity and broadcasts avoided when it compared with DSDV (Ramachandran, 2006).

3.3.4 Advanced uses of AODV

- Because of its reactive nature, AODV can handle highly dynamic behavior of Vehicle Ad-hoc networks (Schwingenschlogl & Kosch, 2002).
- Used for both unicasts and multicasts using the ’J’ (Join multicast group) flag in the packets (Ramachandran, 2006).

3.3.5 Advantages

- Because it is a flat routing protocol it does not need any central administrative system to handle the routing process. Reactive protocols like AODV tend to reduce the control traffic messages overhead at the cost of increased latency in finding new routes (Jacquet, Mühlethaler, Clausen, Laouiti, Qayyum, & Viennot, 2001).
- Tries to keep the overhead of the messages small. If host has the route information in the Routing Table about active routes in the network, then the overhead of the routing process will be minimal. The AODV has great advantage in overhead over simple protocols which need to keep the entire route from the source host to the destination host in their messages. The RREQ and RREP messages, which are responsible for the route discovery, do not increase significantly the overhead from these control messages. AODV reacts relatively quickly to the topological changes in the network and updating only the hosts that may be affected by the change, using the RRER message. The Hello messages, which are responsible for the route maintenance, are also limited so that they do not create unnecessary overhead in the network (Huishan, Huimin, & Nam).
- Loop free and avoids the counting to infinity problem, which were typical to the classical distance vector routing protocols, by the usage of the sequence numbers (Hong, Xu, & Gerla, 2002).

3.3.6 Limitations/Disadvantages of AODV

- Requirement on broadcast medium: The algorithm expects/requires that the nodes in the broadcast medium can detect each others’ broadcasts.

- Overhead on the bandwidth: Overhead on bandwidth will be occurred compared to DSR, when an RREQ travels from node to node in the process of discovering the route info on demand, it sets up the reverse path in itself with the addresses of all the nodes through which it is passing and it carries all this info all its way.
- No reuse of routing info: AODV lacks an efficient route maintenance technique. The routing info is always obtained on demand, including for common case traffic (Dr. Roth, 2004).
- It is vulnerable to misuse: The messages can be misused for insider attacks including route disruption, route invasion, node isolation, and resource consumption (Ning & Sun, 2003). AODV lacks support for high throughput routing metrics: AODV is designed to support the shortest hop count metric. This metric favors long, low bandwidth links over short, high-bandwidth links (Ramachandran, 2006).
- High route discovery latency: AODV is a reactive routing protocol. This means that AODV does not discover a route until a flow is initiated. This route discovery latency result can be high in large-scale mesh networks.

3.4 Destination Sequenced Distance Vector (DSDV) Protocol

Destination sequenced distance vector routing (DSDV) is adapted from the conventional Routing Information Protocol (RIP) to ad hoc networks routing. It adds a new attribute, sequence number, to each route table entry of the conventional RIP. Using the newly added sequence number, the mobile nodes can distinguish stale route information from the new and thus prevent the formation of routing loops (He, 2007).

3.4.1 Packet Routing and Routing Table Management

In DSDV (Perkins & Bhagwat, Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, 1994), each mobile node of an ad hoc network maintains a routing table, which lists all available destinations, the metric and next hop to each destination and a sequence number generated by the destination node. Using such routing table stored in each mobile node, the packets are transmitted between the nodes of an ad hoc network. Each node of the ad hoc network updates the routing table with advertisement periodically or when significant new information is available to maintain the consistency of the routing table with the dynamically changing topology of the ad hoc network.

Periodically or immediately when network topology changes are detected, each mobile node advertises routing information using broadcasting or multicasting a routing table update packet. The update packet starts out with a metric of one to direct connected nodes. This indicates that each receiving neighbor is one metric (hop) away from the node. It is different from that of the conventional routing algorithms. After receiving the update packet, the neighbors update their routing table with incrementing the metric by one and retransmit the update packet to the corresponding neighbors of each of them. The process will be repeated until all the nodes in the ad hoc network have received a copy of the update packet with a corresponding metric. The update data is also kept for a while to wait for the arrival of the best route for each particular destination node in each node before updating its routing table and retransmitting the update packet. If a node receives multiple update packets for a same destination during the waiting time period, the routes with more recent sequence numbers are always preferred as the basis for packet forwarding decisions, but the routing information is not necessarily advertised immediately, if only the sequence numbers have been changed. If the update packets have the same sequence number with the same node, the update packet with the smallest metric will be used and the existing route will be discarded or stored as a less preferable route. In this case, the update packet will be propagated with the sequence number to all mobile nodes in the ad hoc network. The advertisement of routes that are about to change may be delayed until the best routes have been found. Delaying the advertisement of possibly unstable route can damp the fluctuations of the routing table and reduce the number of rebroadcasts of possible route entries that arrive with the same sequence number.

The elements in the routing table of each mobile node change dynamically to keep consistency with dynamically changing topology of an ad hoc network. To reach this consistency, the routing information advertisement must be frequent or quick enough to ensure that each mobile node can almost always locate all the other mobile nodes in the dynamic ad hoc network. Upon the updated routing information, each node has to relay data packet to other nodes upon request in the dynamically created ad hoc network.

In the routing information updating process, the original node tags each update packet with a sequence number to distinguish stale updates from the new one. The sequence number is a monotonically increasing number that uniquely identifies each update from a given node.

As a result, if a node receives an update from another node, the sequence number must be equal or greater than the sequence number of the corresponding node already in the routing table, or else the newly received routing information in the update packet is stale and should be discarded. If the sequence number of one node in the newly received routing information update packet is same as the corresponding sequence number in the routing table, then the metric will be compared and the route with the smallest metric will be used.

In addition to the sequence number and the metric for each entry of the update packet, the update route information contains also both the address of the final destination and the address of the next hop. There are two types of update packets, one is called full dump, which carries all of the available routing information. The other is called incremental, which carries only the routing information changed since the last full dump.

Each node in an ad hoc network must periodically transmit its entire routing table (full dump) to its neighbors most likely using multiple network protocol data units (NPDUs). The full dumps of the nodes can be transmitted relatively infrequently when little movement of mobile nodes is occurring. Incremental update packets are transmitted between the full dumps for partial changes of the routing table such as receiving new sequence numbers and fewer significant route changes. The incremental routing update should be fitted in one NPDU. The mobile nodes are expected to determine the significance of the routing information changes to be sent out with each incremental advertisement. When the significant changes increase with frequent varying of the network topology and the size of an incremental approaches the maximum size of a NPDU, a full dump is scheduled to make the next incremental become smaller.

3.4.2 Responding to Topology Changes

Links can be broken when the mobile nodes move from place to place or have been shut down etc. The broken link may be detected by the communication hardware or be inferred if no broadcasts have been received for a while from a former neighbor. The metric of a broken link is assigned infinity. When a link to next hop has broken, any route through that next hop is immediately assigned an infinity metric and an updated sequence number. Because link broken qualifies as a significant route change, the detecting node will immediately broadcast an update packet and disclose the modified routes.

To describe the broken links, any mobile node other than the destination node generates a sequence number, which is greater than the last sequence number received from the destination. This newly generated sequence number and a metric of infinity will be packed in an update message and flushed over the network. To avoid nodes themselves and their neighbors generating conflicting sequence numbers when the network topology changes, nodes only generate even sequence numbers for themselves, and neighbors only generate odd sequence numbers for the nodes responding to the link changes.

The routes to a lost node will be re-established when the lost node comes back to the network and broadcasts its next update message with an equal or later sequence number and a finite metric. The update message will be disseminated over the whole network to indicate that the broken links have come back into service again. In any case, the entry containing a finite metric and an equal or later sequence number will supersede the corresponding entry with a metric of infinity in the routing table of a node.

3.4.3 Advantages of DSDV

- DSDV protocol guarantees loop free paths (Dr. Roth, 2004).
- Count to infinity problem is reduced in DSDV (Dr. Roth, 2004).
- We can avoid extra traffic with incremental updates instead of full dump updates.
- Path Selection: DSDV maintains only the best path instead of maintaining multiple paths to every destination. With this, the amount of space in routing table is reduced.

3.4.4 Limitations of DSDV

- Wastage of bandwidth due to unnecessary advertising of routing information even if there is no change in the network topology (Patel, 2000).
- DSDV doesn't support Multi path Routing.
- It is difficult to determine a time delay for the advertisement of routes (He, 2007).
- It is difficult to maintain the routing table's advertisement for larger network. Each and every host in the network should maintain a routing table for advertising. But for larger network this would lead to overhead, which consumes more bandwidth.
- DSDV assumes that all wireless links in an ad hoc network are bi-directional. However, this is not true in reality. Wireless media is different from wired media due to its asymmetric connection. Unidirectional links are prevalent in wireless networks.

3.5 Optimized Link State Routing Protocol (OLSR)

The proactive OLSR (Clausen T. H., Hansen, Christensen, & Behrmann, 2001) adapts a classical link state protocol for mobile ad hoc routing. As a proactive routing protocol, it uses periodic messages to update topology information at each node. In a classical link state protocol, the link state packet includes the entire neighbor list along with the associated link cost metric, thus generating large control packet overheard. Furthermore, these packets are broadcast to the entire network which does not scale well to the low bandwidth requirements of wireless ad-hoc networks. OLSR optimizes the classical link state protocol by reducing the control packet overhead and creating efficient flooding mechanisms (Annamalai, 2005).

3.5.1 Introduction of OLSR

The information in this section concerning the Optimized Link State Protocol is taken from its RFC 3561 (Clausen & Jacquet, 2003). Optimized Link State Protocol (OLSR) is a proactive routing protocol, so the routes are always immediately available when needed. OLSR is an optimization version of a pure link state protocol. So the topological changes cause the flooding of the topological information to all available hosts in the network. To reduce the possible overhead in the network protocol uses Multipoint Relays (MPR). The idea of MPR is to reduce flooding of broadcasts by reducing the same broadcast in some regions in the network, more details about MPR can be found later in this chapter. Another reduce is to provide the shortest path. The reducing the time interval for the control messages transmission can bring more reactivity to the topological changes (Ge, Kunz, & Lamont, 2003), (Jacquet, Laouiti, Minet, & Viennot, Performance of multipoint relaying in ad hoc mobile routing protocols, 2002), (Clausen T. , Hansen, Christensen, & Behrmann, 2001), (Laouiti, Qayyum, & Viennot, 2002), (Jacquet, Laouiti, Minet, & Viennot, 2001).

OLSR uses two kinds of the control messages: Hello and Topology Control (TC). Hello messages are used for finding the information about the link status and the host's neighbors. With the Hello message the Multipoint Relay (MPR) Selector set is constructed which describes which neighbors has chosen this host to act as MPR and from this information the host can calculate its own set of the MPRs. the Hello messages are sent only one hop away but the TC messages are broadcasted throughout the entire network. TC messages are used for broadcasting information about own advertised neighbors which includes at least the MPR Selector list. The TC messages are broadcasted periodically and only the MPR hosts can forward the TC messages.

There is also Multiple Interface Declaration (MID) messages which are used for informing other host that the announcing host can have multiple OLSR interface addresses. The MID message is broadcasted throughout the entire network only by MPRs. There is also a "Host and Network Association" (HNA) message which provides the external routing information by giving the possibility for routing to the external addresses. The HNA message provides information about the network- and the netmask addresses, so that OLSR host can consider that the announcing host can act as a gateway to the announcing set of addresses. The HNA is considered as a generalized version of the TC message with only difference that the TC message can inform about route cancelling while HNA message information is removed only after expiration time. The MID and HNA messages are not explained in more details in this chapter, the further information concerning these messages can be found in (Huhtonen, 2004).

3.5.2 Routing

3.5.2.1 Neighbor Sensing

The link in the ad hoc network can be either unidirectional or bidirectional so the host must know this information about the neighbors. The Hello messages are broadcasted periodically for the neighbor sensing. The Hello messages are only broadcasted one hop away so that they are not forwarded further. When the first host receives the Hello message from the second host, it sets the second host status to asymmetric in the routing table. When the first host sends a Hello message and includes that, it has the link to the second host as asymmetric, the second host set first host status to symmetric in own routing table. Finally, when second host send again Hello message, where the status of the link for the first host is indicated as symmetric, then first host changes the status from asymmetric to symmetric. In the end both hosts knows that their neighbor is alive and the corresponding link is bidirectional (Laouti, Mühlethaler, Najid, & Plakoo, 2002).

The Hello messages are used for getting the information about local links and neighbors. The Hello messages periodic broadcasting is used for link sensing, neighbor's detection and MPR selection process. Hello message contains: information how often the host sends Hello messages, willingness of host to act as a Multipoint Relay, and information about its neighbor. Information about the neighbors contains: interface address, link type and neighbor type. The link type indicates that the link is symmetric, asymmetric or simply lost. The neighbor type is just symmetric, MPR or not a neighbor. The MPR type indicates that the link to the neighbor is symmetric and that this host has chosen it as Multipoint Relay.

3.5.2.2 Multipoint Relays

The Multipoint Relays (MPR) is the key idea behind the OLSR protocol to reduce the information exchange overhead. Instead of pure flooding the OLSR uses MPR to reduce the number of the host which broadcasts the information throughout the network. The MPR is a host's one hop neighbor which may forward its messages. The MPR set of host is kept small in order for the protocol to be efficient. In OLSR only the MPRs can forward the data throughout the network.

Each host must have the information about the symmetric one hop and two hop neighbors in order to calculate the optimal MPR set. Figure 6 illustrates these concepts. Information about the neighbors is taken from the Hello messages. The two hop neighbors are found from the Hello message because each Hello message contains all the hosts' neighbors. Selecting the minimum number of the one hop neighbors which covers all the two hop neighbors is the goal of the MPR selection algorithm. Also each host has the Multipoint Relay Selector set, which indicates which hosts has selected the current host to act as a MPR.

When the host gets a new broadcast message, which is need to be spread throughout the network and the message's sender interface address is in the MPR Selector set, then the host must forward the message. Due to the possible changes in the ad hoc network, the MPR Selectors sets are updated continuously using Hello messages.

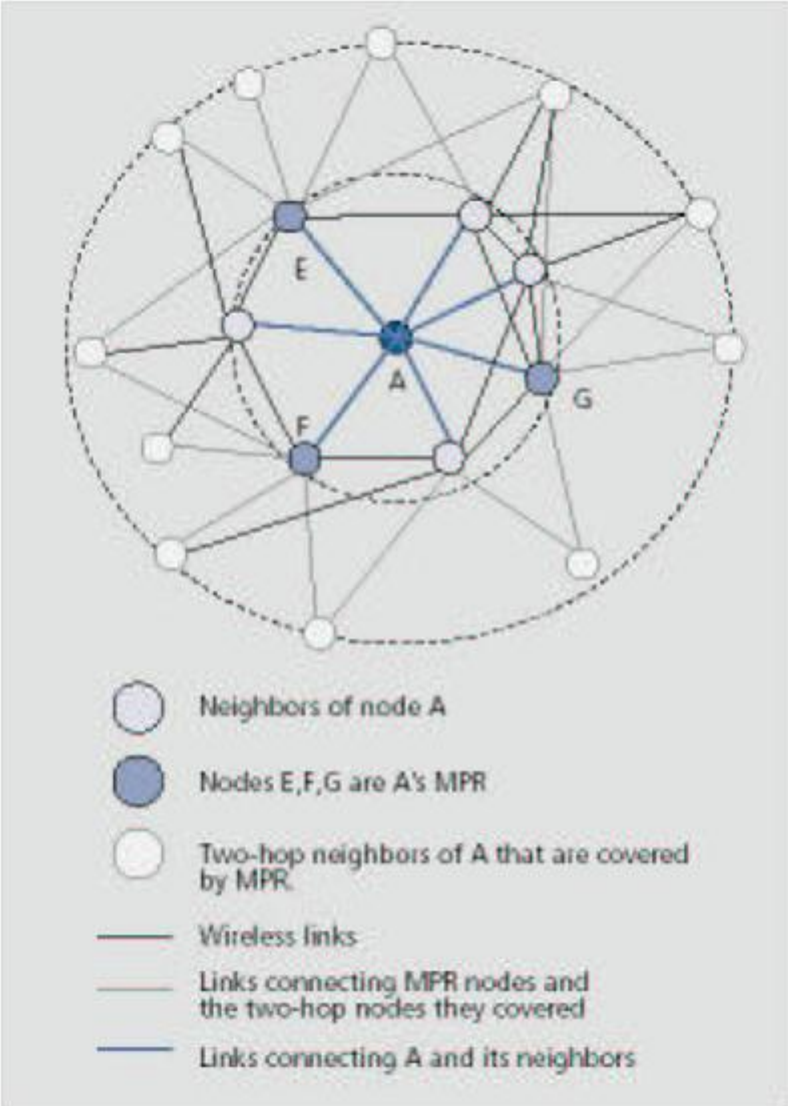


Figure 6 The basic concepts of Multipoint Relays

3.5.2.3 Multipoint Relays Selection

In this section the proposed algorithm for the selection of Multipoint Relay set is described. The algorithm constructs the MPR set which includes minimum number of the one hop symmetric neighbors from which it is possible to reach all the symmetrical strict two hop neighbors. The host must have the information about one and two hop symmetric neighbors in order to start the needed calculation for the MPR set. All the exchange of information are broadcasted using Hello messages. The neighbors which have status of willingness different than WILL_NEVER in the Hello message can be chosen to act as MPR. The neighbor must be symmetric in order to become an MPR.

Proposed algorithm for selecting Multipoint Relay set:

1. Take all the symmetric one hop neighbors which are willing to act as an MPR.

2. Calculate for every neighbor host a degree, which is a number of the symmetric neighbors, that are two hops away from the calculating source and does not include the source or its one hop neighbors.
3. Add the neighbor symmetric host to the MPR set. If it is the only neighbor from which is possible to get to the specific two hop neighbor, then remove the chosen host neighbors from the two hop neighbor set.
4. If there are still some hosts in the two hop neighbor set, then calculate the reachability of the each one hop neighbor, meaning the number of the two hop neighbors, that are yet uncovered by MPR set. Choose the node with highest willing value, if the values are the same then takes the node with greater number of reachability. If the reachability is the same, then take the one with greater degree counted in the second step. After choosing the neighbor for MPR set remove the reachable two hop neighbor from the two hop neighbor set.
5. Repeat previous step until the two hop neighbors set is empty.
6. For the optimization, set the hosts in the MPR set in the increasing order basing on the willingness. If one host is taken away and all the two hop neighbors, covered by at least one host and the willingness of the host is smaller than WILL_ALWAYS, then the host may be removed.

The possible improvements of this algorithm are needed, for example, when there are multiple possible interface addresses for one host. The finding the optimum MPR set for the two hop neighbor coverage is considered to be an NP problem based on.

3.5.2.4 Topology Information

In order to exchange the topological information and build the topology information base the host that were selected as MPR need to sent the topology control (TC) message. The TC messages are broadcasted throughout the network and only MPR are allowed to forward TC messages. The TC messages are generated and broadcasted periodically in the network.

The TC message is sent by a host in order to advertise own links in the network. The host must send at least the links of its MPR selector set. The TC message includes the own set of advertised links and the sequence number of each message. The sequence number is used to avoid loops of the messages and for indicating the freshness of the message, so if the host gets a message with the smaller sequence number it must discard the message without any updates. The host must increment the sequence number when the links are removed from the TC message and also it should increment the sequence number when the links are added to the message. The sequence numbers are wrapped around. When the hosts advertised links set becomes empty, it should still send empty TC messages for specified amount of time, in order to invalidate previous TC messages. This should stop sending the TC messages until it has again some information to send.

The size of the TC message can be quite big, so the TC message can be sent in parts, but then the receiver must combine all parts during some specified amount of time. Host can increase its transmission rate to become more sensible to the possible link failures. When the change in the MPR Selector set is noticed, it indicates that the link failure has happened and the host must transmit the new TC message as soon as possible.

3.5.2.5 Routing Table Calculations

The host maintains the routing table, the routing table entries have following information: destination address, next address, number of hops to the destination and local interface address. Next address indicates the next hop host. The information is got from the topological set (from the TC messages) and from the local link information base (from the Hello messages). So if any changes occur in these sets, then the routing table is recalculated. Because this is proactive protocol then the routing table must have routes for all available hosts in the network. The information about broken links or partially known links is not stored in the routing table.

The routing table is changed if the changes occur in the following cases: neighbor link appear or disappear, two hops neighbor is created or removed, topological link is appeared or lost or when the multiple interface association information changes. But the update of this information does not lead to the sending of the messages into the network. For finding the routes for the routing table entry the shortest path algorithm is used.

3.5.3 Advantages

- OLSR is also a flat routing protocol, it does not need central administrative system to handle its routing process. The proactive characteristic of the protocol provides that the protocol has all the routing information to all participated hosts in the network. However, as a drawback OLSR protocol needs that each host periodic sends the updated topology information throughout the entire network, this increase the protocols bandwidth usage. But the flooding is minimised by the MPRs, which are only allowed to forward the topological messages.
- The reactiveness to the topological changes can be adjusted by changing the time interval for broadcasting the Hello messages. It increases the protocols suitability for ad hoc network with the rapid changes of the source and destinations pairs. Also the OLSR protocol does not require that the link is reliable for the control messages, since the messages are sent periodically and the delivery does not have to be sequential.

- Due to the OLSR routing protocol simplicity in using interfaces, it is easy to integrate the routing protocol in the existing operating systems, without changing the format of the header of the IP messages. The protocol only interacts with the host's Routing Table.
- OLSR protocol is well suited for the application which does not allow the long delays in the transmission of the data packets. The best working environment for OLSR protocol is a dense network, where the most communication is concentrated between a large number of nodes.
- OLSR has also extensions to allow for hosts to have multiple OLSR interface addresses and provide the external routing information giving the possibility for routing to the external addresses. Based on this information there is possibility to have hosts in the ad hoc network which can act as gateways to another possible network.

3.5.4 Limitations/Disadvantages of OLSR

- Constantly use of bandwidth.
- Although a path from source to destination is being provided, it is not necessarily the shortest path, because every route involves forwarding through a MPR node.
- OLSR has routing delays and bandwidth overhead at the MPR nodes as they act as localized forwarding routers.
- The routing table size grows nonlinearly and the control messages can block the actual packets.

A comparison of the characteristics of the above three ad hoc routing protocols AODV, DSDV, and OLSR is given in Table 5.

Table 6 Comparison of AODV, DSDV and OLSR routing protocols.

Protocol Property	AODV	DSDV	OLSR
Multicast Routes	No	No	Yes
Distributed	Yes	Yes	Yes
Unidirectional Link Support	No	No	Yes
Multicast	Yes	No	Yes
Periodic Broadcast	Yes	Yes	Yes
QoS Support	No	No	Yes
Reactive	Yes	No	No

4 Simulation

For our simulation analysis, we have used NS-3[6] as our network simulator (version ns-3.13). The NS-3 simulator is an open source discrete event network simulator targeted primarily for networking research and educational purposes. Previous to NS-3, NS-2 was the tool for academic networking research. But it had many disadvantages. It required the involvement of more than one language, like oTcl and C++. For new modules and features it required a lot of manual recoding and compilations. NS-3 is not an extension of NS-2. Although both simulators are written in C++, yet NS-3 is a new simulator, which does not support the NS-2 APIs. NS-3 is written entirely in C++, with optional Python bindings. Simulation scripts can therefore be written in C++ or in Python. For the scenarios studied in the present document the scripts were written in C++. NS-3 no longer uses oTcl scripts to control the simulation, thus abandoning the problems which were introduced by the combination of C++ and oTcl in NS-2. Thus, NS-3 is much easier to use and a more readily extensible platform. The Simulation Network Architecture looks just like IP architecture stack. The nodes in NS-3 may or may not have mobility. The nodes have "network devices", which transfer packets over channel and incorporates Layer 1 (Physical Layer) & Layer 2 (Data Link layer). The network devices acts as an interface with Layer 3 (Network Layer: IP, ARP). The Layer 3 supports the Layer 4 (Transport Layer: UDP, TCP), which is used by the Layer 5 (Application Layer) objects.

By default, ns-3 simulations use a fixed seed. To obtain randomness across multiple simulation runs, we set the seed differently and found the averages for al runs.

4.1 Performance Metrics

The performance metrics for the evaluation of the three routing protocols (AODV, OLSR and DSDV) are packet delivery ratio (PDR), throughput, average delay and total energy consumption.

- **Packet Delivery Ratio (PDR)**: The number of packets received divided by the number of packets sent by the application.
- **Throughput**: The average rate of successful message delivery over a communication channel.
- **Average Delay**: the time taken by the packet to reach the destination node's MAC protocol from the source node's MAC protocol.
- **Total Energy Consumption**: The total energy consumption of the device.

[6] http://www.nsnam.org/

4.2 Real-Life Simulation Scenarios

In this section we describe the three real-life scenarios based on which the simulation was performed. All three scenarios are located in Greece. The main reason for this choice is because the basic aim of the present thesis is to lay the foundations for the use of Manets in Greece.

4.2.1 School Field Trip Scenario

The first scenario describes a field trip in Parnitha Mountain performed by a school, the purpose of which is the observation for education purposes, non-experimental research and to provide students experiences outside their everyday activities. More specifically, the aim of this research is to observe the geology and the flora of a certain region of the mountain and possibly collect samples. The students should be able to record their activities while moving and be able to communicate with their teacher in every region of the mountain. The use of cell phones won't be the best option because there might be certain areas where the reception will be very poor to non-existent and also sending videos might not be fully supported by all the networks the devices have. For this reasons, in our scenario temporary and localized telecommunication is demanded because the usual fracture networks cannot be achieved and so the use of Manet is imperative.

New mobile technology, such as hand-held cellular based devices, is playing a large role in redefining how we receive information and even students in our days have access in such devices and are capable of using them. With the use of those devices, a Manet can be formed and serve the needs of the group.

In our scenario, we have a group of 20 students and a teacher (21 nodes). Each one of them is equipped with a hand-held device. All of them are walking in a certain direction with a constant speed of 2 m/s. For this reason the mobility model is the Gauss-Markov mobility model. In our scenario we assume that the students will cover a certain area of 1,000 x 500 m. The aim of this group is to observe the flora and exchange pictures, videos, text messages for questions and notes. More specifically, the teacher gives an assignment to the students based on the lessons they have made in the classroom. The students have to find and take a video or a photo of the subject (for example a flower or a tree) and send it to the teacher. When the assignment is completed the teacher will process all the photos and the videos the students have sent and evaluate them. The students can also take photographs of trees, flowers or something that they find interesting and send it to the teacher in order to discuss it in the classroom. In the table that follows (Table 6) are presented the parameters for the present scenario corresponding to the parameters in the simulation study.

Table 7 Correlation between parameters in simulation and parameters in the School Field Trip scenario

Parameters in Simulation	Parameters in real-life scenario
Nodes	students and teacher
Sender	the students
Receiver	the teacher
Speed	2 m×sec^{-1} (average walking speed)
Messages	text messages, photographs, videos, audio

During the field trip, the main factor that can seriously affect the communication reliability and delay is the amount of the sent and received traffic by the participants. We define traffic as the number of packets that are sent in 1 second, therefore the interval plays an important role in the simulation. Also the traffic is being affected by the packet size and the number of packets being sent. Furthermore, the traffic is Constant Bit Rate (CBR), which means that packets are send continuously with the same rate. For the purpose of an educational visit to the mountain, it will be very common to transmit multimedia applications like videos that produce heavy traffic to the network. Moreover, the number of the students that participate in the trip affects the quality of the communication. Primary goal in communications is the minimization interference and the maximization usage of the bandwidth available on the wireless channel especially when multiple senders exist. The simulation time for this experiment is 180 seconds. In the following table (Table 7) the simulation parameters are presented in detail.

Table 8 Simulation parameters for the School Field Trip scenario

Simulation Parameters	Values
Number of Nodes	21 (20 students and a teacher)
Nodes' Speed	2 m×sec^{-1}
Senders	20 (the students)
Receivers	1 (the teacher)
Movement	Gauss-Markov Mobility Model
Area	1000 × 500 m
Protocols	AODV or DSDV or OLSR
DsssRate	1Mbps or 2Mbps or 5.5Mbps or 11Mbps
Packet Size	256 bytes or 512 bytes or 1024 bytes or 2048 bytes or 4096 bytes
Number of Packets	250
Interval	0 - 0,05 - 0,1 - 0,15 - 0,2 - 0,25 - 0,3 - 0,35 - 0,4 - 0,45 - 0,5
Simulation Time	180 sec

4.2.2 Rescue Operation Scenario

This rescue scenario takes place in the mountain Olympus. After a snow slide, the traces of a group of mountaineers are lost. A rescue procedure starts immediately to find any survivors. Rescue and emergency operations are characterized by very hectic and dynamic environments, where time is a critical factor. There is a lot of movement and activity on the site as personnel may arrive and leave the site at different times, e.g., in cases where personnel or other resources (ambulances, helicopters) are called out to other incidents in the area. Several organizations are involved in the operation, e.g., paramedics, fire fighters and police, in addition to a number of other organizations, some of which are voluntary. Fracture networks do not exist in this environment and so the form of a Manet is essential.

The on-scene coordinator/commander (OSC) is the person who has the main role and responsibility of the team. Every member of the team informs the OSC about evidence they find and other important information, by sending him files or voice messages. In this way the OSC has the full overview of all members of the team at all times. In the table that follows (Table 8) are presented the parameters for the present scenario corresponding to the parameters in the simulation study.

Table 9 Correlation between parameters in simulation and parameters in the Rescue Operation scenario

Parameters in Simulation	Parameters in real-life scenario
Nodes	team members
Sender	Rescuers
Receiver	OSC
Data Rate	164 Kbps
Messages	files, text messages, audio (voice messages)

In this scenario we examine how the number of the team members and the speed they have affects the communication. We assume that the operation is at face one and that the team has no information about the missing persons only about their location the last time they communicate with their base. For this reason the team will search in blind at first the location and therefore the mobility model is Random Direction. The team will search a certain area of 500 x 500 m^2 at every face. The next search area will be determined by the evidence they will collect from the present area. The team members inform the OSC with files, text or voice messages and so the data rate is defined at 164 Kbps. The Dsss Rate is 11Mbps because the team is gradually increases for the purpose of the simulation and so the channel will be shared among multiple users. In the scenario the team members might search the area on foot or use special vehicles specifically designed for the mountains and so the speed the team members might have is from 5m/s (which corresponds to the average human walking) to 55m/s (which corresponds to the speed of a vehicle). The simulation time for this experiment is 180 seconds. In the following table (Table 9) the simulation parameters are presented in detail.

Table 10 Simulation parameters for the Rescue Operation scenario

Simulation Parameters	Values
Number of Nodes	10 – 20 – 30 – 40 – 50 – 60 – 70 – 80 – 90 - 100
Nodes' Speed	5 – 10 – 15 – 20 – 25 – 30 – 35 – 40 – 45 – 50 – 55 $m \times sec^{-1}$
Senders	the team members
Receivers	1 (the OSC)
Movement	Random Direction Mobility Model
Area	500 × 500 m
Protocols	AODV or DSDV or OLSR
DsssRate	11Mbps
Packet Size	2048 bytes
Number of Packets	150
Interval	0,1 seconds
Data Rate	164 Kbps
Simulation Time	180 sec

4.2.3 Archaeological Site Scenario

Using wireless technology in museums is not a pioneer idea. In many museums visitors are given a handheld device, and after a 15 minute of training from Explainers (high school students, volunteers etc) the visitor could see the exhibits, find particular online resources and even see information about the exhibit as they come close to it. A lot of the in formations are using sound, so headsets are important. As this use of technology seems to offer great advantages in addition to a classical visit to a museum, isolation phenomenon was observed to most of the users. There were no interaction with other visitors and the use of headset makes the handheld device the primary exhibit. No collaboration was succeeded in any point. In this scenario, we assume that we have a group of tourists who want to make an organized tour to the archeological site of Vergina. In hills around Vergina are burial sites of the kings of Macedon, including the tomb of Philip II, father of Alexander the Great. In this environment fracture networks and Wi-Fi spots do not exist. Also the mobile phone reception is very poor to non-existent With the use of handheld devices (PDAs, smart phones, and any device with a wireless connectivity) and headsets every group of tourists can form a Manet and the tour guide will be able to send multimedia packets to the tourists describing the exhibitions. In this way the tours can easily become private and so many tours could be organized at the same time without interfering with each other. In the table that follows (Table 10) are presented the parameters for the present scenario corresponding to the parameters in the simulation study.

Table 11 Correlation between parameters in simulation and parameters in the Archeological Site scenario

Parameters in Simulation	Parameters in real-life scenario
Nodes	tourists
Sender	Tour guide
Receiver	the tourists
Speed	2 m×sec^{-1} (average walking speed)
Data Rate	2000 Kbps
Messages	high quality videos, audio

In this scenario we examine how the number of the team members and the number of packets being send affects the communication. Form their nature; tours follow a certain pattern of mobility. The tourists follow the tour guide to a certain direction with a constant speed of 2m/s which is the average walking speed. For this reason the mobility model is the Gauss-Markov mobility model. The archeological area the tour takes place is 1000 x 1000 m^2. The tour guide sends high quality videos and voice messages to the tourists and so the data rate is defined at 2200 Kbps. The Dsss Rate is 1Mbps because the number of the tourists is relatively small due to the restricted number of visitors in some areas such as the tombs. The simulation time for this experiment is 180 seconds. In the following table (Table 11) the simulation parameters are presented in detail.

Table 12 Simulation parameters for the Archeological Site scenario

Simulation Parameters	Values
Number of Nodes	5 – 10 – 15 – 20 – 25 – 30
Nodes' Speed	2 m×sec^{-1}
Sender	the tour guide
Receivers	The tourists
Movement	Gauss – Markov Mobility Model
Area	1000 × 1000 m
Protocols	AODV or DSDV or OLSR
DsssRate	1Mbps
Packet Size	4096 bytes
Number of Packets	50 – 100 – 150 – 200 – 250 – 300 – 350 – 400 – 450 – 500
Interval	0,015 seconds
Data Rate	2200 Kbps
Simulation Time	180 sec

4.3 Simulation Results

In this section we present the simulation results for each scenario, protocol and performance metric (Packet Delivery Ratio, Average Delay, Throughput and Total Energy Consumption).

4.3.1 School Filed Trip Scenario

The simulation results for the School Field Trip Scenario are presented on the four sections that follow. In each section we investigate the behavior of each protocol and we also compare them based on different values of interval, packet size and DSSS Rate. In the end of each section we compare the three routing protocols for every case that has been studies and also based on the average price of the interval.

4.3.1.1 Packet Delivery Ratio

In this section we investigate the communication reliability for different types of traffic and DSSS Rate. We measure the PDR. PDR is the percentage from the send messages that was actually delivered. It represents how reliable the communication is. The higher the PDR, the better the communication reliability is. Also, the different values of packet size and interval determine how heavy the traffic is because it defines the data rate. We use five and different packet sizes (256, 512, 1024, 2048, 4096 bytes) and eleven different interval values (0s – 0.5s with 0.05 step). The biggest the packet size and smallest the interval, the heavier the traffic is. Finally, the simulations were made for different DSSS Rate values. We will see how the four different values of the DSSS Rate (1Mbps, 2Mbps, 5.5Mbps and 11Mbps) affect the sharing of a single channel among multiple users.

- AODV

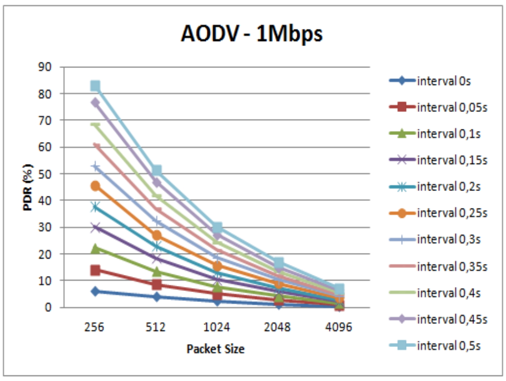


Figure 7 PDR vs. Packet Size with various intervals, DSSS Rate is 1Mbps (AODV).

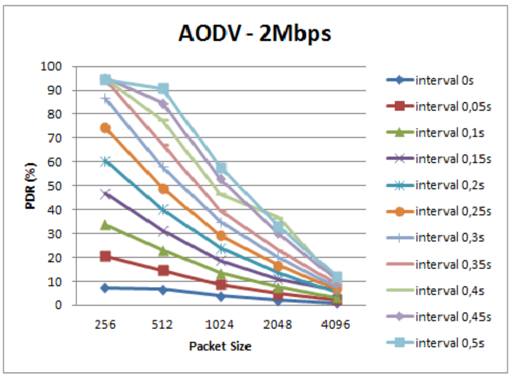


Figure 8 PDR vs. Packet Size with various intervals, DSSS Rate is 2Mbps (AODV).

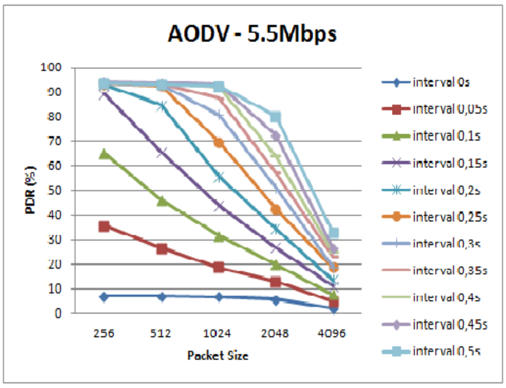


Figure 9 PDR vs. Packet Size with various intervals, DSSS Rate is 5.5Mbps (AODV).

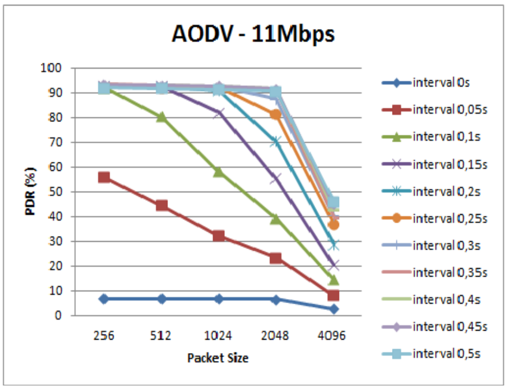


Figure 10 PDR vs. Packet Size with various intervals, DSSS Rate is 11Mbps (AODV).

In Figures 7-10 we observe the PDR for different types of traffic and DSSS Rate when the AODV routing protocols is used. As expected, using smaller packet sizes and higher interval values, we achieve better PDR. More specifically, for DSSS Rate 1Mbps (Figure 7), when the packet is 256 bytes and the interval is 0.5s the PDR is 83%. As the interval is reduced, the PDR is reduced by 8% for every 0.05 seconds. As the packet size is growing this rate of decrease is being reduced. For packet size 512 bytes the maximum PDR is 51.39% and is being reduced by approximately 5%. For packet size 1024 bytes the maximum PDR is 30.14% and is being reduced by approximately 3%. For packet size 2048 bytes the maximum PDR is 17.06% and is being reduced by approximately 2% and for packet size 4096 bytes the maximum PDR is 7.2% and is being reduced by approximately 0.5%. When the DSSS Rate is 2Mbps we observe that the maximum PDR for every packet size and interval is slightly higher. When the DSSS Rate is 1Mbps only when the packet size is 256 and the interval is 0.5 at the same time, we have PDR more than 80%. But when the DSSS Rate is 2Mbps (Figure 8), we have PDR more than 80% when the packet size is 256 bytes and the interval is higher than 0.3 seconds, or when the packet size is 512 bytes and the interval is higher than 0.45 seconds. The highest PDR at this DSSS Rate is 94.37%, for packet size 256 bytes and interval 0.5seconds. As it is expected, when the DSSS Rate is 5.5Mbps (Figure 9) the PDR levels are even higher. More precisely, the PDR is higher than 80% when the packet size is 512 bytes and the interval is more than 0.15 seconds, or when the packet size is 512 bytes and the interval is more than 0.2 seconds, or when the packet size is 1024 and the interval is more than 0.3 seconds or when the packet size is 2048 and the interval is 0.5 seconds. The highest PDR is 94.31% for packet size 256 bytes and interval 0.45 seconds. Finally, when the DSSS Rate is 11Mbps (Figure 10), we have the highest PDR values. We have PDR more than 80% when the packet size is 256 and 512 bytes and the interval is more than 0.1 seconds, when the packet size is 1024 bytes and the interval is more than 0.15 seconds and when the packet size is 2048 bytes and the interval is more than 0.25 seconds.

In conclusion we should mention that as the DSSS Rate is growing, the PDR is growing too and the protocol performs better in more demanding traffic levels. Also, we can observe that with higher DSSS Rates, bigger packet sizes and also smaller interval values can be used. The only case that the PDR stays in low levels whatever the DSSS Rate is, is when the interval is 0 seconds. Of course, this is an ideal situation because this means that the data rate is infinite.

- DSDV

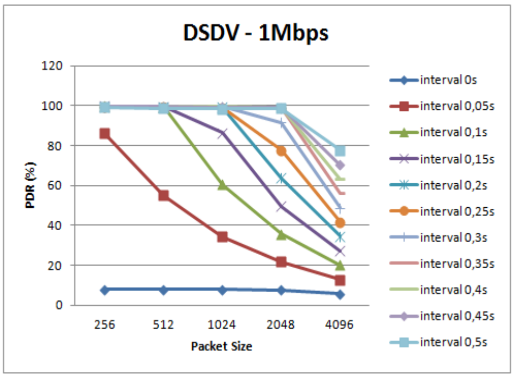


Figure 11 PDR vs. Packet Size with various intervals, DSSS Rate is 1Mbps (DSDV).

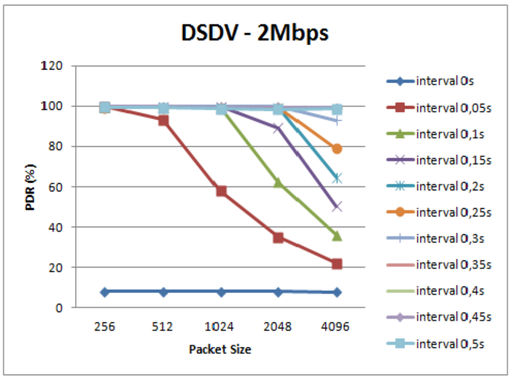


Figure 12 PDR vs. Packet Size with various intervals, DSSS Rate is 2Mbps (DSDV).

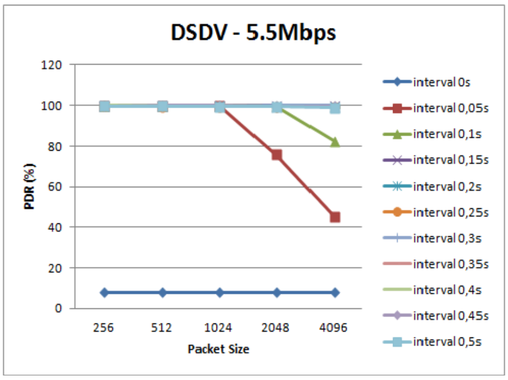


Figure 13 PDR vs. Packet Size with various intervals, DSSS Rate is 5.5Mbps (DSDV).

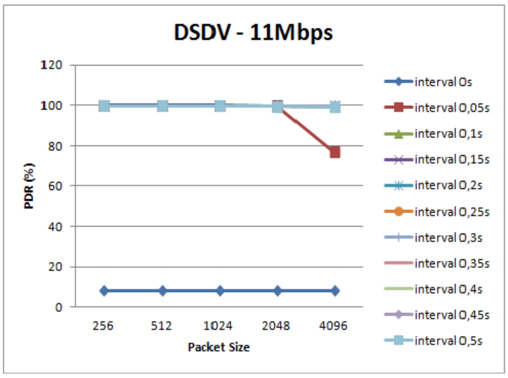


Figure 14 PDR vs. Packet Size with various intervals, DSSS Rate is 11Mbps (DSDV).

A first look in Figures 11-14 can clearly distinguish that the DSDV protocol achieves better PDR levels than AODV. In detail, when DSSS Rate is 1Mbps (Figure 11) we have PDR almost 100% when the packet size is 256 and 512 bytes and the interval is 0.1 second and more, when the packet size is 1024 bytes and the interval is 0.2 seconds and more and when the packet size is 2048 bytes and the interval is 0.35 seconds and more. When the DSSS Rate is 2Mbps (Figure 12), the PDR is even better as it reaches almost the 100% for every packet size when the interval is more than 0.3 seconds and for every interval when the packet size is 256 or 512 bytes. When the DSSS Rate is 5.5 Mbps (Figure 13) only when the interval is lower than 0.05 seconds and the packets size is 2048 or 4096 bytes the PDR is lower than 80%. Finally, when the DSSS Rate is 11Mbps (Figure 14), DSDV has approximately 100%

PDR for almost every case. As we can see, only when the interval is 0.05 seconds and the packet size is 4096 bytes the PDR is 76.70%.

For every DSSS Rate we have the extremes when the interval is 0 seconds and so the data rate is infinite.

- OLSR Results

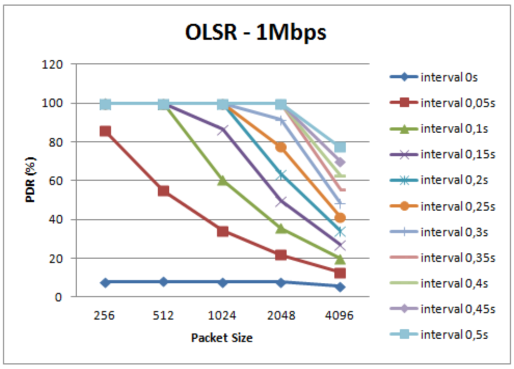


Figure 15 PDR vs. Packet Size with various intervals, DSSS Rate is 1Mbps (OLSR).

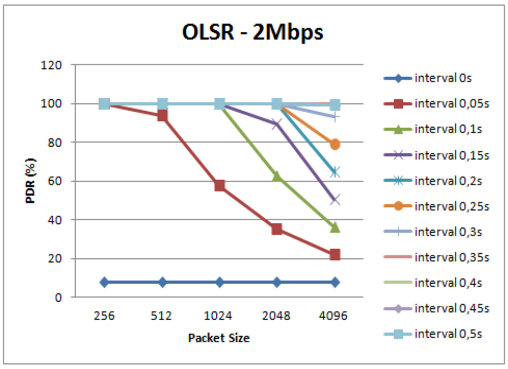


Figure 16 PDR vs. Packet Size with various intervals, DSSS Rate is 2Mbps (OLSR).

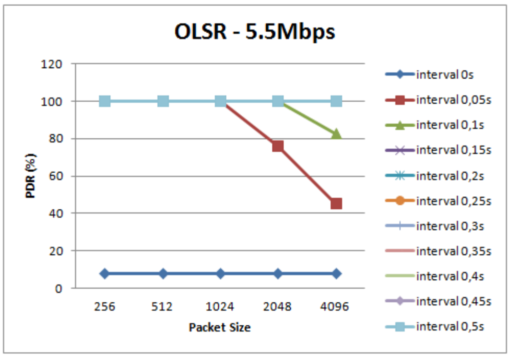


Figure 17 PDR vs. Packet Size with various intervals, DSSS Rate is 5.5Mbps (OLSR).

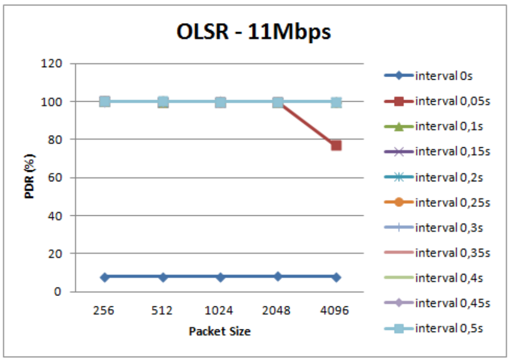


Figure 18 PDR vs. Packet Size with various intervals, DSSS Rate is 11Mbps (OLSR

In Figures 15-18 we observe that OLSR performs almost the same as DSDV, with very small differences which can be barely distinct from the diagrams. When DSSS Rate is 1Mbps (Figure 15) we have PDR almost 100% when the packet size is 256 and 512 bytes and the interval is 0.1 second and higher, when the packet size is 1024 bytes and the interval is 0.2 seconds and higher and when the packet size is 2048 bytes and the interval is 0.3 seconds and higher. When the DSSS Rate is 2Mbps (Figure 16), the PDR is even better as it reaches almost the 100% for every packet size when the interval is more than 0.3 seconds and for every interval when the packet size is 256 or 512 bytes. When the DSSS Rate is 5.5 Mbps (Figure 17) only when the interval is lower than 0.05 seconds and the packets size is 2048 or 4096 bytes the PDR is lower than 80%. Finally, when the DSSS Rate is 11Mbps (Figure 18), DSDV has approximately 100% PDR for almost every case. As we can see, only when the interval is 0.05 seconds and the packet size is 4096 bytes the PDR is 76.93%.

For every DSSS Rate we have the extremes when the interval is 0 seconds and so the data rate is infinite.

4.3.1.2 Delay

In this section we investigate the average delay time. The delay affects the quality of the communication and so it is an important factor for the evaluation of the protocols.

- AODV Results

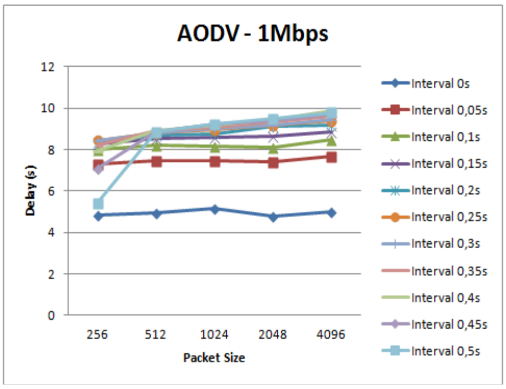


Figure 19 Delay vs. Packet Size with various intervals, DSSS Rate is 1Mbps (AODV).

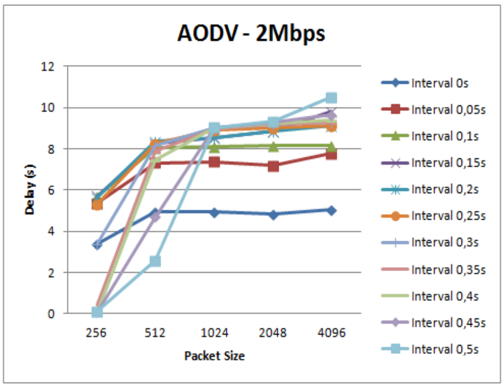


Figure 20 Delay vs. Packet Size with various intervals, DSSS Rate is 2Mbps (AODV).

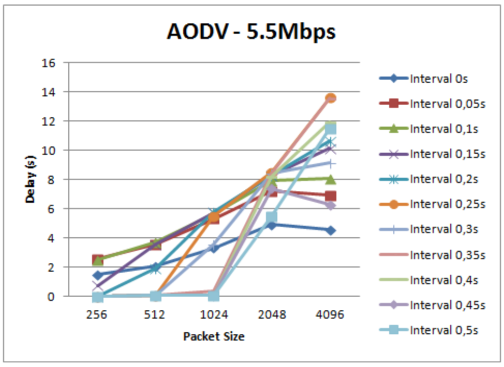


Figure 21 Delay vs. Packet Size with various intervals, DSSS Rate is 5.5Mbps (AODV).

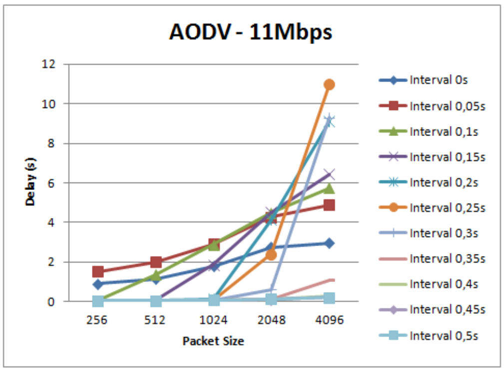


Figure 22 Delay vs. Packet Size with various intervals, DSSS Rate is 11Mbps (AODV).

In Figures 19-22 we can observe the delay for the four different DSSS Rates when the AODV protocol is used. In Figure 19 we can see that when the DSSS Rate is 1Mbps, the delay is from 4.7 to 9.7 seconds. As the interval is growing, the delay is growing too. In general, the delay is being affected more from the interval than from the packet size. When the DSSS Rate is 2Mbps (Figure 20) the delay levels are lower for smaller packet sizes. When the packet size is 256 bytes we even have almost 0 seconds delay for interval higher than 0.35 seconds. The highest delay is 10.5 seconds when the packet size is 4096 bytes and the interval is 0.5 seconds. When the packet size is 256 bytes, the highest delay is 5.7 seconds and when the packet size is 4096 bytes the lowest delay is 5 seconds. When the DSSS Rate is 5.5Mbps (Figure 21), the delay is even better than the other two rates. The delay is almost zero when the packet size is 256, 512 and 1024 bytes and the interval is lower than 0.2 seconds. The highest delay is 13,62 seconds when the packet size is 4096 bytes and the interval is 0.25 seconds. The highest delay when the packet size is 256 bytes is lower than before (2.5 seconds) and the lowest delay when the packet size is 4096 bytes is lower than before (4.5 seconds). Finally, when the DSSS Rate is 11Mbps (Figure 22), the delay is almost zero when the interval is 0.4 seconds or higher. The highest delay is 11 seconds when the packet size is 4096 bytes and the interval is 0.25 seconds and when the packet size is 1024 bytes or lower, the delay is lower than 2.8 seconds.

A very interesting result is the delay is more balanced when the interval is 0.05 seconds. This means that the difference between the lowest and the highest delay is smaller as the interval is reduces. This is also being affected by the DSSS Rate. When the DSSS Rate is 1, 2 or 5.5Mbps, the delay is more balanced.

- DSDV Results

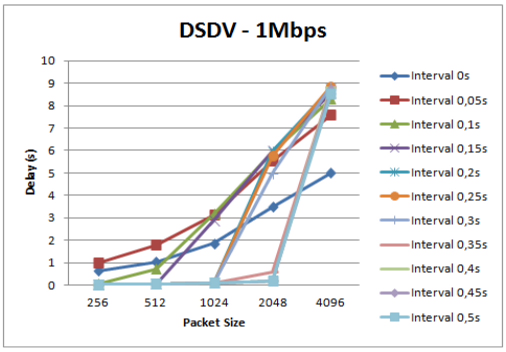


Figure 23 Delay vs. Packet Size with various intervals, DSSS Rate is 1Mbps (DSDV).

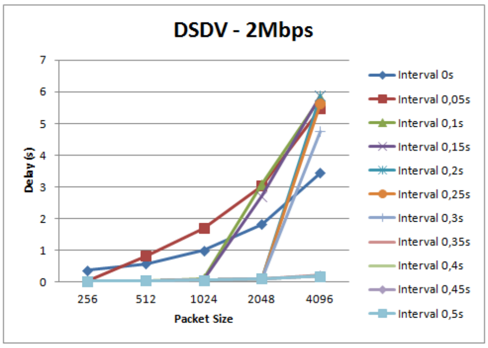


Figure 24 Delay vs. Packet Size with various intervals, DSSS Rate is 2Mbps (DSDV).

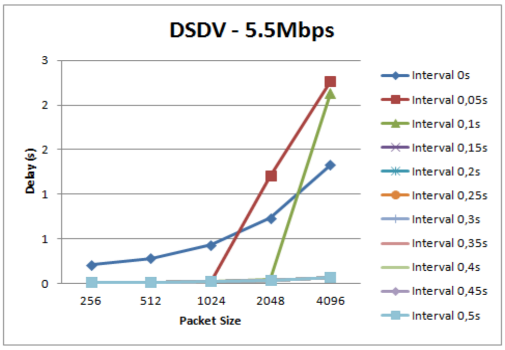


Figure 25 Delay vs. Packet Size with various intervals, DSSS Rate is 5.5Mbps (DSDV).

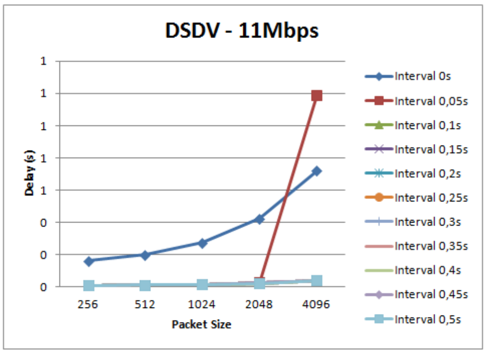


Figure 26 Delay vs. Packet Size with various intervals, DSSS Rate is 11Mbps (DSDV).

In Figures 23-26 we can observe the delay for the four different DSSS Rates when the DSDV protocol is used. When the DSSS Rate is 1Mbps (Figure 23), the delay for packet size 256 and 512 bytes is almost zero. The highest delay is 8.85 seconds when the packet size is 2049 bytes and interval 0.25 seconds. When the DSSS Rate is 2Mbps (Figure 24) the delay levels are better than before. The delay is almost zero when the packet size is 256 and 512 bytes or when the interval is higher than 0.35 seconds. The highest delay is 5.87 seconds when the packet size is 4096 bytes and the interval is 0.15 seconds. When the DSSS Rate is 5.5Mbps (Figure 25), the delay is almost zero when the interval is higher than 0.15 seconds or when the packet size is 256, 512 and 1024 bytes. The highest delay is 2.26 seconds, when the packet size is 4096 seconds and the interval is 0.05 seconds. Finally, when the DSSS Rate is 11Mbps (Figure 26), the delay is almost zero for every case, except when the packet size is 4096 bytes and the interval is 0.05 seconds (the delay is 1.1 seconds).

- OLSR Results

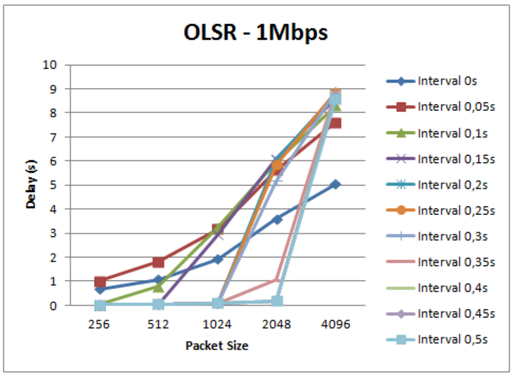


Figure 27 Delay vs. Packet Size with various intervals, DSSS Rate is 1Mbps (OLSR).

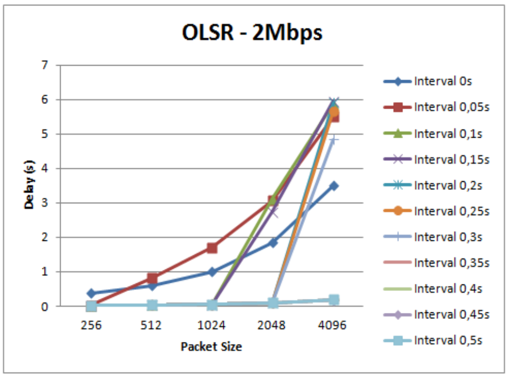


Figure 28 Delay vs. Packet Size with various intervals, DSSS Rate is 2Mbps (OLSR).

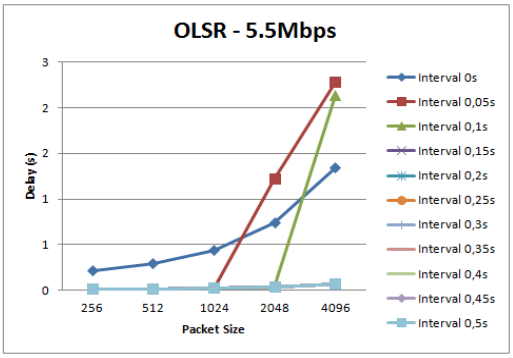


Figure 29 Delay vs. Packet Size with various intervals, DSSS Rate is 5.5Mbps (OLSR).

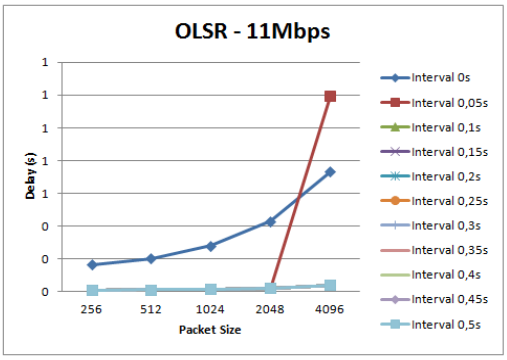


Figure 30 Delay vs. Packet Size with various intervals, DSSS Rate is 11Mbps (OLSR).

In Figures 27-30 we can observe the delay for the four different DSSS Rates when the OLSR protocol is used. As in PDR results in the previous section, we observe that OLSR performs almost the same as DSDV, with very small differences which can be barely distinct from the diagrams. When the DSSS Rate is 1Mbps (Figure 27), the delay for packet size 256 and 512 bytes is almost zero. The highest delay is 8.88 seconds when the packet size is 2049 bytes and interval 0.3 seconds. When the DSSS Rate is 2Mbps (Figure 28) the delay levels are better than before. The delay is almost zero when the packet size is 256 and 512 bytes or when the interval is higher than 0.35 seconds. The highest delay is 5.94 seconds when the packet size is 4096 bytes and the interval is 0.15 seconds. When the DSSS Rate is 5.5Mbps (Figure 29), the delay is almost zero when the interval is higher than 0.15 seconds or when the packet size is 256, 512 and 1024 bytes. The highest delay is 2.27 seconds, when the packet size is 4096 seconds and the interval is 0.05 seconds. Finally, when the DSSS Rate is 11Mbps (Figure 30), the delay is almost zero for every case, except when the packet size is 4096 bytes and the interval is 0.05 seconds (the delay is 1.19 seconds).

An interesting observation in the results in this section is that when the DSSS Rate is 1Mbps, which is the worst case for the DSDV and the OLSR protocols considering the results for higher DSSS Rates, the results are almost identical when the DSSS Rate is 11Mbps and the routing protocol is AODV. Therefore, the "best" results for AODV are the "worst" results for DSDV and OLSR.

4.3.1.3 Throughput

In this section we investigate the throughput for different types of traffic and DSSS Rate. The throughput is an important factor as well which affects the communication and the network performance.

- AODV Results

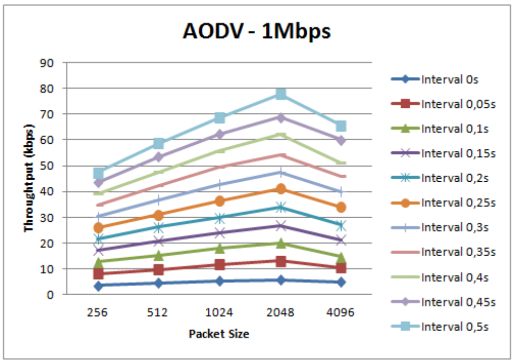


Figure 31 Throughput vs. Packet Size with various intervals, DSSS Rate is 1Mbps (AODV).

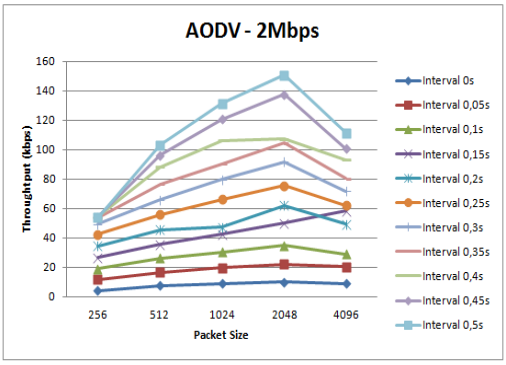


Figure 32 Throughput vs. Packet Size with various intervals, DSSS Rate is 2Mbps (AODV).

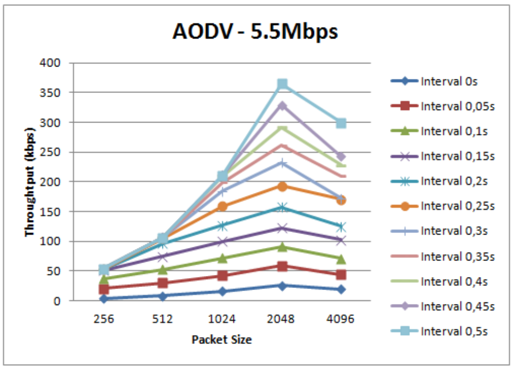


Figure 33 Throughput vs. Packet Size with various intervals, DSSS Rate is 5.5Mbps (AODV).

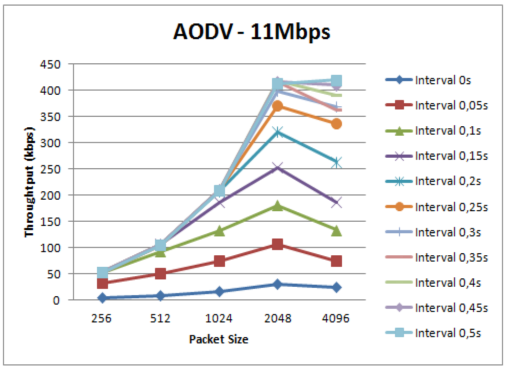


Figure 34 Throughput vs. Packet Size with various intervals, DSSS Rate is 11Mbps (AODV).

In Figures 31-34 we can observe the delay for the four different DSSS Rates when the AODV protocol is used. When the DSSS Rate is 1Mbps (Figure 31), the highest throughput is 77.6kbps when the packet size is 2048 bytes and the interval is 0.5 seconds. As we can see,

for every packet size, the higher the interval, the higher the throughput is. When the DSSS Rate is 2Mbps (Figure 32), the results are better. The highest throughput is 150.45 kbps, when the packet size is 2048 bytes and the interval is 0.5 seconds like before. When the DSSS Rate is 5.5Mbps (Figure 33), the protocol performs better. The highest throughput is 364.63kbps when the packet size is 2048 bytes and the interval is 0.5 seconds. Finally, when the DSSS Rate is 11Mbps (Figure 34), the protocol performs better than any other rate. The highest throughput is 419kbps when the packet size is 4096 bytes and the interval is 0.5 seconds.
As we can see, the best results when the DSSS Rate is 11Mbps is the worst when the DSSS Rate is 1Mbps. It is interesting that the best results regardless the DSSS Rate are when the packet size is 2048 bytes.

- DSDV Results

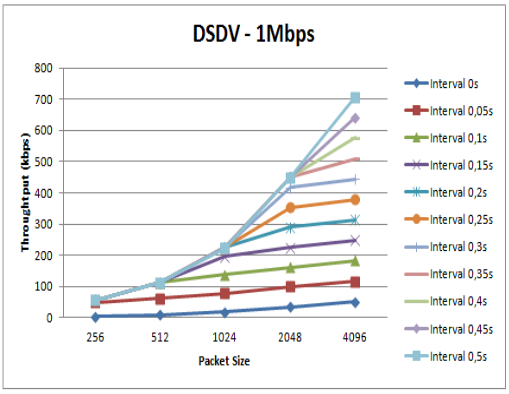


Figure 35 Throughput vs. Packet Size with various intervals, DSSS Rate is 1Mbps (DSDV).

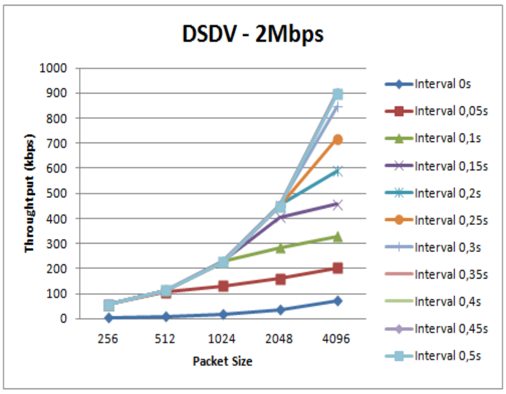


Figure 36 Throughput vs. Packet Size with various intervals, DSSS Rate is 2Mbps (DSDV).

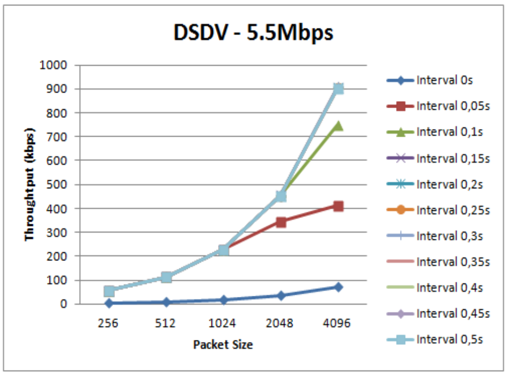


Figure 37 Throughput vs. Packet Size with various intervals, DSSS Rate is 5.5Mbps (DSDV).

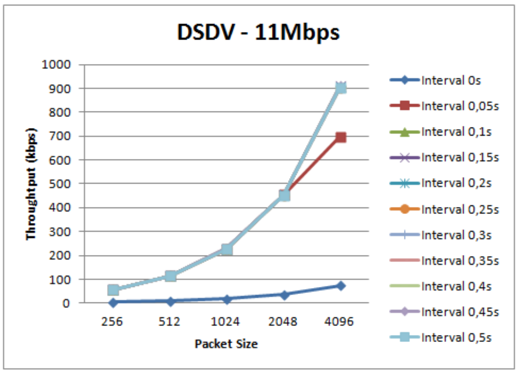


Figure 38 Throughput vs. Packet Size with various intervals, DSSS Rate is 11Mbps (DSDV).

In Figures 35-38 we can observe the delay for the four different DSSS Rates when the DSDV protocol is used. When the DSSS Rate is 1Mbps (Figure 35), the highest throughput is 706.3kbps when the packet size is 4096 bytes and the interval is 0.5 seconds. When the packet size is 256 bytes the throughput is lower than 60kbps. When the DSSS Rate is 2Mbps (Figure 36), the highest throughput is 902kbps when the packet size is 4096 bytes and the interval is 0.45 seconds. For DSSS Rate 5.5Mbps and 11Mbps (Figure 37-38)we can observe that the results is almost the same with only exceptions when the packet size is 2048 and the interval

0.05 seconds and when the packet size is 4096 bytes and the interval is 0.1 seconds and lower. When the DSSS Rate is 11Mbps the protocol performs slightly better than when the DSSS Rate is 5.5Mbps. In 5.5Mbps the highest throughput is 908.5kbps and in 11Mbps the highest throughput is 909kbps. As we can see, for every packet size, the highest the interval, the highest the throughput is.

- OLSR Results

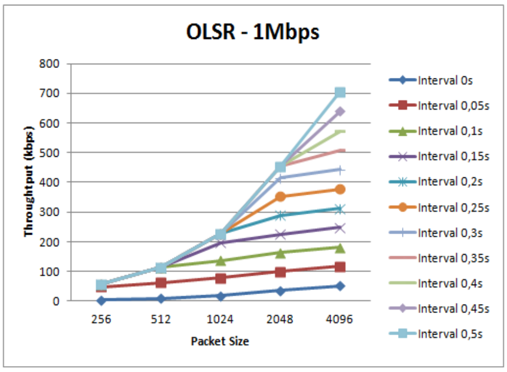


Figure 39 Throughput vs. Packet Size with various intervals, DSSS Rate is 1Mbps (OLSR).

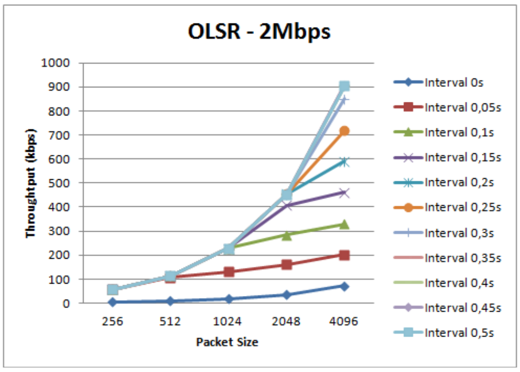


Figure 40 Throughput vs. Packet Size with various intervals, DSSS Rate is 2Mbps (OLSR).

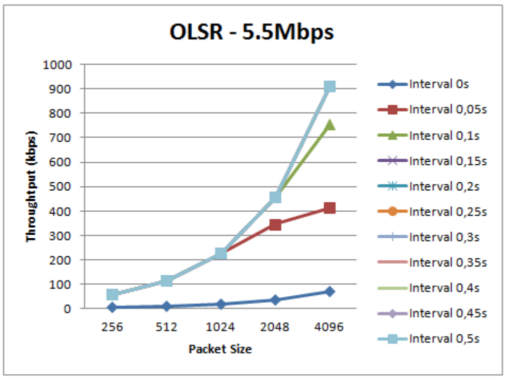


Figure 41 Throughput vs. Packet Size with various intervals, DSSS Rate is 5.5Mbps (OLSR).

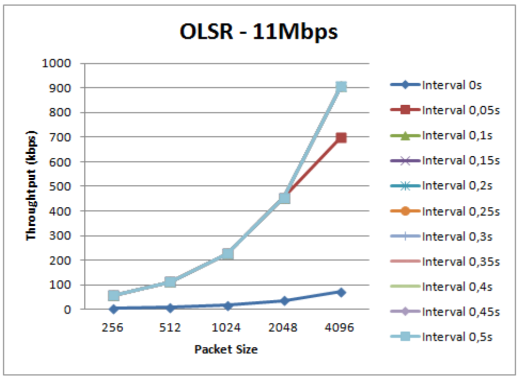


Figure 42 Throughput vs. Packet Size with various intervals, DSSS Rate is 11Mbps (OLSR).

In Figures 39-42 we can observe the delay for the four different DSSS Rates when the OLSR protocol is used. As we can see, the results are almost the same as when the routing protocol is DSDV. When the DSSS Rate is 1Mbps (Figure 39), the highest throughput is 704.9kbps when the packet size is 4096 bytes and the interval is 0.5 seconds. When the packet size is 256 bytes the throughput is lower than 60kbps. When the DSSS Rate is 2Mbps (Figure 40), the highest throughput is 906kbps when the packet size is 4096 bytes and the interval is 0.35 seconds. For DSSS Rate 5.5Mbps and 11Mbps (Figure 41-42) we can observe that the results is almost the same with only exceptions when the packet size is 2048 and the interval 0.05 seconds and when the packet size is 4096 bytes and the interval is 0.1 seconds and lower. When the DSSS Rate is 11Mbps the protocol performs slightly better than when the DSSS Rate is 5.5Mbps. In 5.5Mbps the highest throughput is 908.7kbps and in 11Mbps the highest throughput is 909kbps. As we can see, for every packet size, the highest the interval, the highest the throughput is.

An interesting observation is that for DSDV and OLSR routing protocols as the packet size doubles the throughput is almost being doubled too. Also, in those two protocols we can reach throughput of 909kbps when AODV has highest throughput 419kbps. In very demanding applications, DSDV and OLSR perform better than AODV too.

4.3.1.4 Total Energy Consumption

The final perfume metric which is being studied is energy. Energy is a very important factor in routing protocols for MANETS, because in our scenario portable devices may not have the chance to be recharged and so the total energy consumption should be reduced as far it can be.

- AODV Results

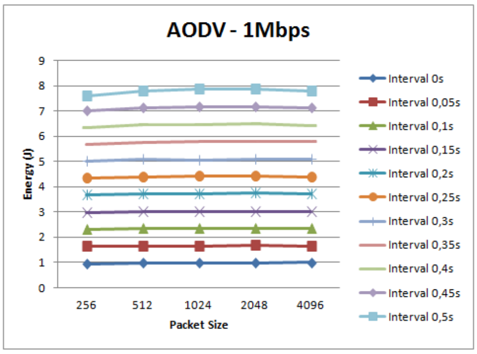


Figure 43 Total Energy Consumption vs. Packet Size with various intervals, DSSS Rate is 1Mbps (AODV).

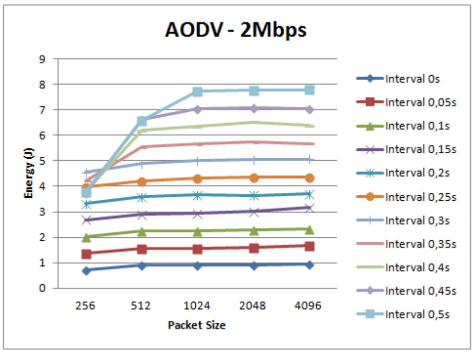


Figure 44 Total Energy Consumption vs. Packet Size with various intervals, DSSS Rate is 5.5Mbps (AODV).

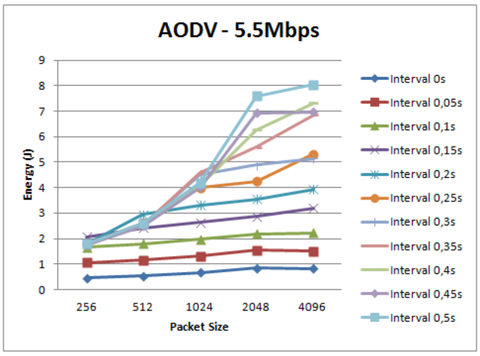


Figure 45 Total Energy Consumption vs. Packet Size with various intervals, DSSS Rate is 2Mbps (AODV).

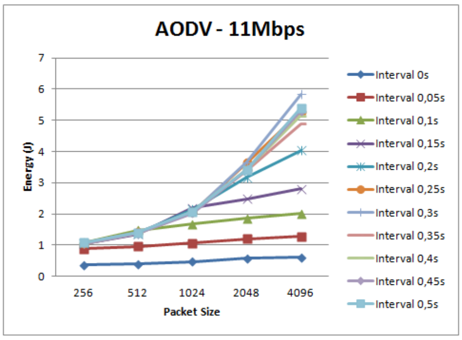


Figure 46 Total Energy Consumption vs. Packet Size with various intervals, DSSS Rate is 11Mbps (AODV)

In Figures 43-46 we can observe the total energy consumption for the four different DSSS Rates when the AODV protocol is used. When the DSSS Rate is 1Mbps (Figure 43), the results for each interval are almost the same regardless the packet size. The higher the interval, the higher the total energy consumption is. The highest energy consumption is 7.86J when the packet size is 2048 bytes and the interval 0.5 seconds. When the DSSS Rate is 2Mbps (Figure 44) the results for each interval are almost the same for packet size 1024 and bigger. When the packet size is 512 the highest energy consumption is 6.6J and when the packet size is 256 bytes the total energy consumption is 4.5J. When the DSSS Rate is 5.5Mbps (Figure 45), the highest energy consumption is 8J when packet size is 4096 bytes and interval is 0.05 seconds and as we can see the consumption is being reduced even more than with the previous DSSS Rates. Finally, when the DSSS Rate is 11Mbps (Figure 46), the results are even better. The highest total energy consumption is 5.8J when the packet size is 4096 bytes and the interval is 0.3 seconds. In general, the bigger the packet size, the highest the total energy consumption is.

- DSDV Results

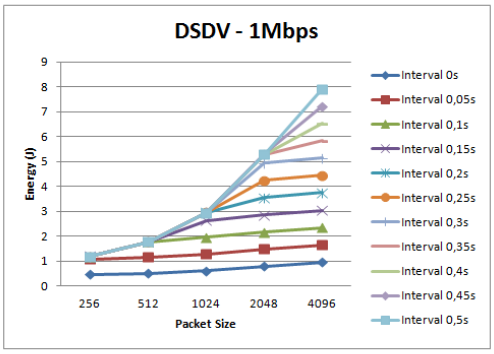


Figure 47 Total Energy Consumption vs. Packet Size with various intervals, DSSS Rate is 1Mbps (DSDV).

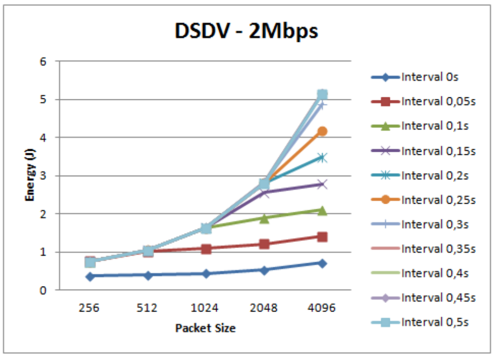


Figure 48 Total Energy Consumption vs. Packet Size with various intervals, DSSS Rate is 5.5Mbps (DSDV).

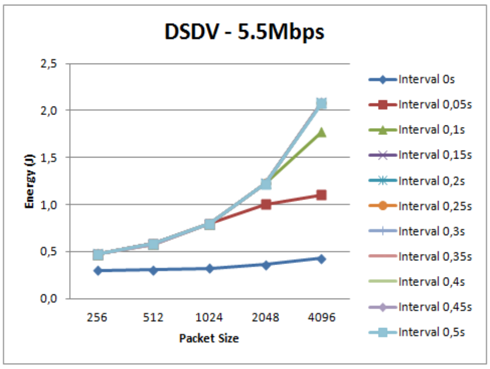


Figure 49 Total Energy Consumption vs. Packet Size with various intervals, DSSS Rate is 2Mbps (DSDV).

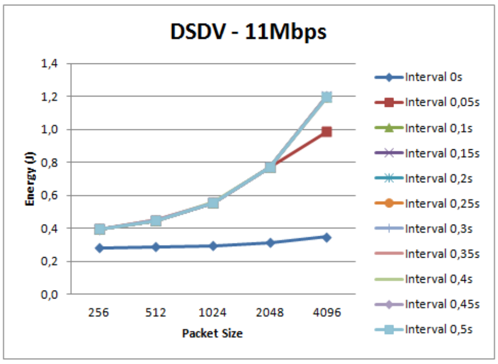


Figure 50 Total Energy Consumption vs. Packet Size with various intervals, DSSS Rate is 11Mbps (DSDV).

In Figures 47-50 we can observe the total energy consumption for the four different DSSS Rates when the DSDV protocol is used. When the DSSS rate is 1Mbps (Figure 47), the highest energy consumption is 7.9J when the packet size is 4096 bytes and the interval is 0.5 seconds. For packet size 256 and 512 bytes the total energy consumption is lower than 2J. When the DSSS Rate is 2Mbps (Figure 48), the energy consumption is lower than before. The highest energy consumption is 5.17J for packet size 4096 and interval 0.45 seconds. When the DSSS Rate is 5.5Mbps (Figure 49), the highest energy consumption is 2.08 seconds when the packet size is 4096 bytes and the interval is 0.45 seconds. Finally, when the DSSS Rate is

11Mbps (Figure 50), the total energy consumption does not outreach the 1.2J in every case. For each interval the energy consumption is almost the same regardless the packets size apart from the case when the packet size is 4096 bytes and the interval is 0.05 seconds.

As we can see, the total energy consumption in every case regardless the DSSS Rate, is being reduced as the packet size is being reduced. As it is mentioned before the only case that the energy consumption stays in low levels whatever the DSSS Rate is, is when the interval is 0 seconds. Of course, this is an ideal situation because this means that the data rate is infinite.

- OLSR Results

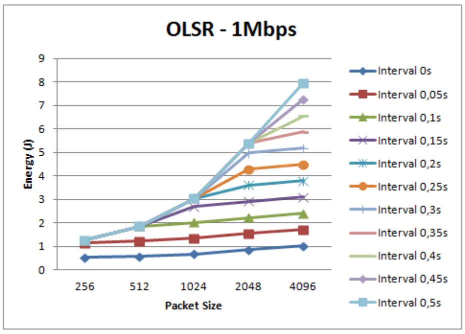


Figure 51 Total Energy Consumption vs. Packet Size with various intervals, DSSS Rate is 1Mbps (OLSR).

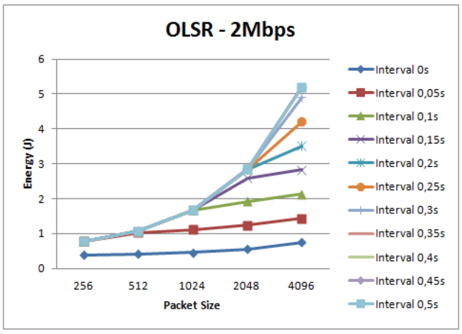


Figure 52 Total Energy Consumption vs. Packet Size with various intervals, DSSS Rate is 2Mbps (OLSR).

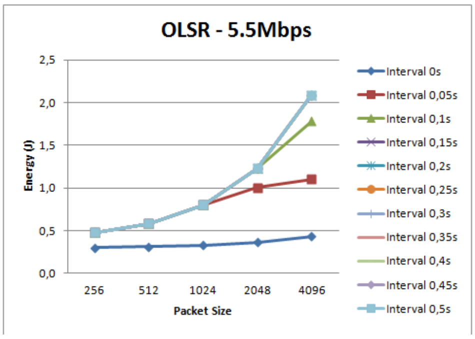


Figure 53 Total Energy Consumption vs. Packet Size with various intervals, DSSS Rate is 5.5Mbps (OLSR).

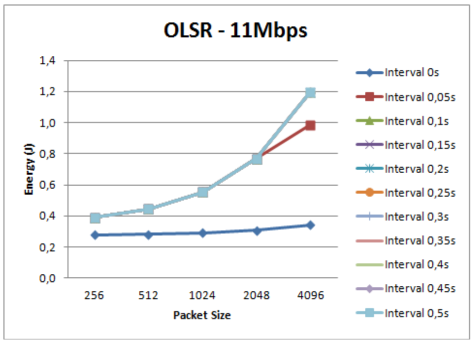


Figure 54 Total Energy Consumption vs. Packet Size with various intervals, DSSS Rate is 11Mbps (OLSR).

In Figures 51-54 we can observe the total energy consumption for the four different DSSS Rates when the OLSR protocol is used. Like the previous section, the OLSR protocol presents almost the same results as DSDV. When the DSSS rate is 1Mbps (Figure 51), the highest energy consumption is 7.9J when the packet size is 4096 bytes and the interval is 0.5 seconds. For packet size 256, 512 bytes the total energy consumption is lower than 2J. When the DSSS Rate is 2Mbps (Figure 52), the energy consumption is lower than before. The highest energy consumption is 5.2J for packet size 4096 and interval 0.5 seconds. When the DSSS Rate is 5.5Mbps (Figure 53), the highest energy consumption is 2.09 seconds when the packet size is 4096 bytes and the interval is 0.55 seconds. Finally, when the DSSS Rate is 11Mbps (Figure 54), the total energy consumption does not outreach the 1.2J in every case.

For each interval the energy consumption is almost the same regardless the packets size apart from the case when the packet size is 4096 bytes and the interval is 0.05 seconds.

As we can see, the total energy consumption in every case regardless the DSSS Rate, is being reduced as the packet size is being reduced. Again, the only case that the energy consumption stays in low levels whatever the DSSS Rate is, is when the interval is 0 seconds. Of course, this is an ideal situation because this means that the data rate is infinite.

4.3.1.5 Comparison of Routing Protocols

In this section we compare the three routing protocols. We categorize the diagrams firstly based on the performance metrics and secondly based on the DSSS Rate. These diagrams will provide a better understanding of the on differences between protocols. In the end of this section will be a summary of this comparison.

Packet Delivery Ratio (PDR)

In Figures 55-74 we can compare the three routing protocols based on the PDR they achieve. The results are categorized based on the different DSSS Rates.

1. DSSS Rate 1Mbps

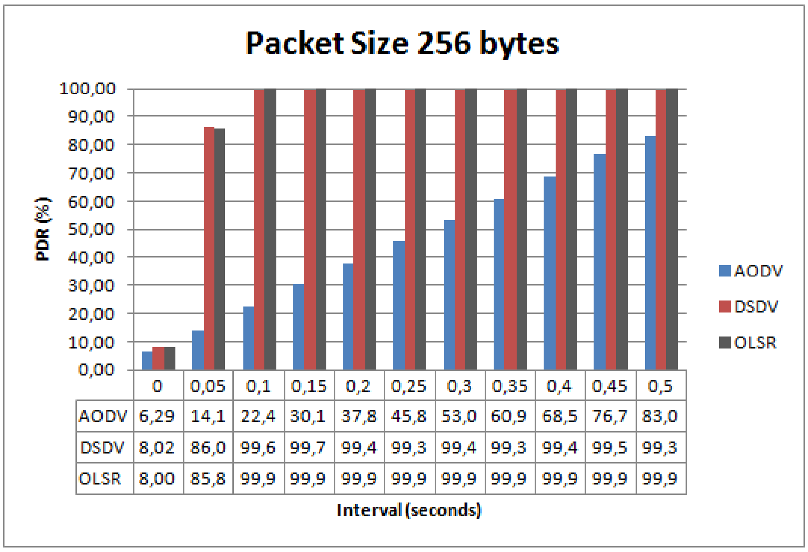
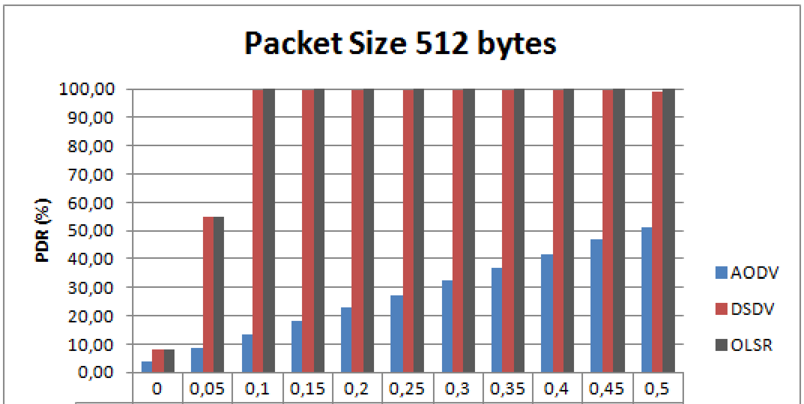


Figure 55 PDR vs. interval, DSSS Rate 1Mbps, Packet Size 256 bytes.

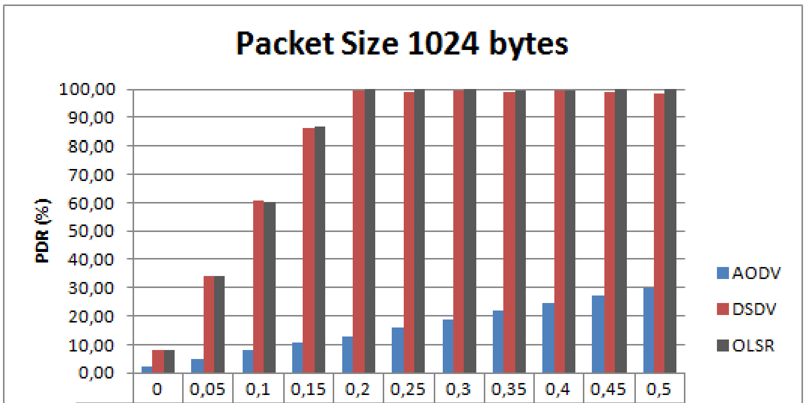
In Figure 55 we can see that routing protocols DSDV and OLSR outperform AODV in every case. DSDV outperforms OLSR only when the interval is 0 and 0.05 seconds.



	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
AODV	4,11	8,68	13,5	18,4	23,0	27,2	32,4	37,0	41,8	47,1	51,3
DSDV	8,02	55,0	99,4	99,5	99,5	99,2	99,5	99,2	99,3	99,5	98,7
OLSR	8,01	54,9	99,9	99,9	99,8	99,9	99,9	99,9	99,8	99,9	99,9

Figure 56 PDR vs. interval, DSSS Rate 1Mbps, Packet Size 512 bytes.

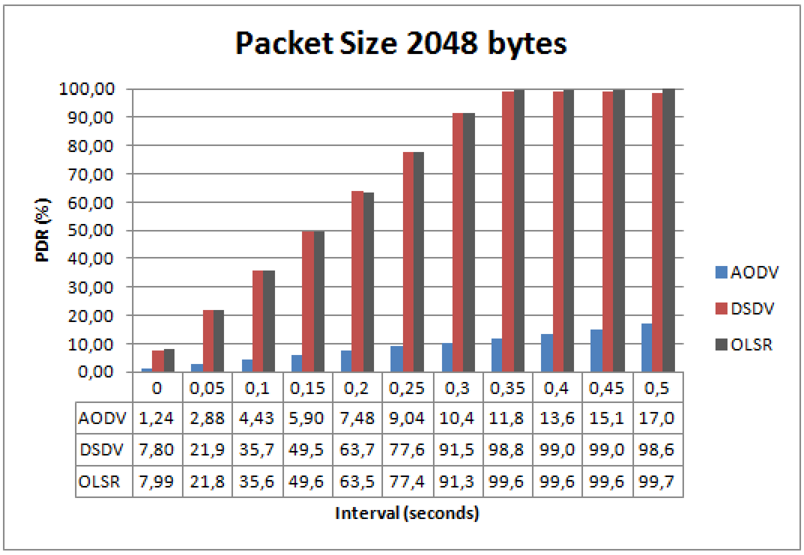
In Figure 56 we can see that routing protocols DSDV and OLSR outperform AODV in every case. DSDV outperforms OLSR only when the interval is 0 and 0.05 seconds.



	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
AODV	2,40	5,15	7,90	10,5	13,0	15,9	18,7	21,6	24,4	27,3	30,1
DSDV	8,02	34,2	60,5	86,3	99,2	99,0	99,5	99,0	99,3	99,1	98,3
OLSR	7,98	34,1	60,4	86,6	99,8	99,8	99,8	99,7	99,7	99,8	99,8

Figure 57 PDR vs. interval, DSSS Rate 1Mbps, Packet Size 1024 bytes.

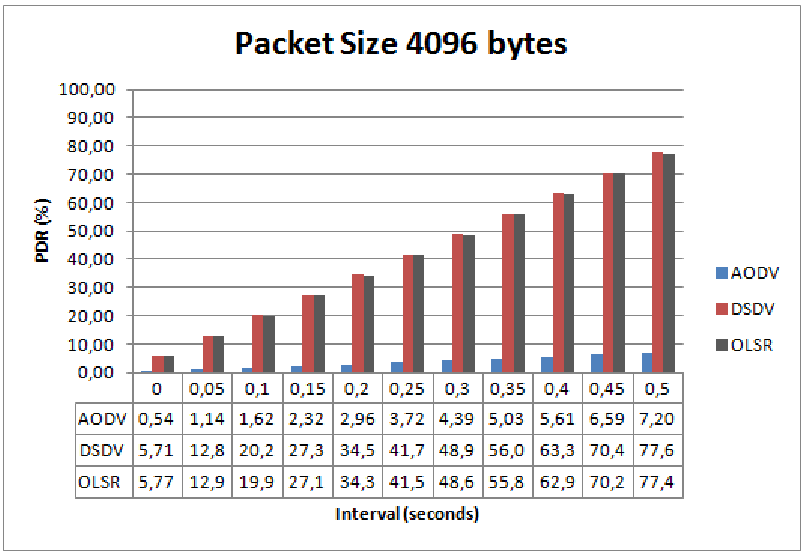
In Figure 57 we can see that routing protocols DSDV and OLSR outperform AODV in every case. DSDV outperforms OLSR only when the interval is 0, 0.05 and 0.1 seconds.



	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
AODV	1,24	2,88	4,43	5,90	7,48	9,04	10,4	11,8	13,6	15,1	17,0
DSDV	7,80	21,9	35,7	49,5	63,7	77,6	91,5	98,8	99,0	99,0	98,6
OLSR	7,99	21,8	35,6	49,6	63,5	77,4	91,3	99,6	99,6	99,6	99,7

Figure 58 PDR vs. interval, DSSS Rate 1Mbps, Packet Size 2048 bytes.

In Figure 58 we can see that routing protocols DSDV and OLSR outperform AODV in every case. DSDV outperforms OLSR only when the interval is 0.05, 0.1, 0.2, 0.25 and 0.3 seconds.



	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
AODV	0,54	1,14	1,62	2,32	2,96	3,72	4,39	5,03	5,61	6,59	7,20
DSDV	5,71	12,8	20,2	27,3	34,5	41,7	48,9	56,0	63,3	70,4	77,6
OLSR	5,77	12,9	19,9	27,1	34,3	41,5	48,6	55,8	62,9	70,2	77,4

Figure 59 PDR vs. interval, DSSS Rate 1Mbps, Packet Size 4096 bytes.

In Figure 59 we can see that routing protocols DSDV and OLSR outperform AODV in every case. OLSR outperforms DSDV only when the interval is 0 and 0.05 seconds.

2. DSSS Rate 2Mbps

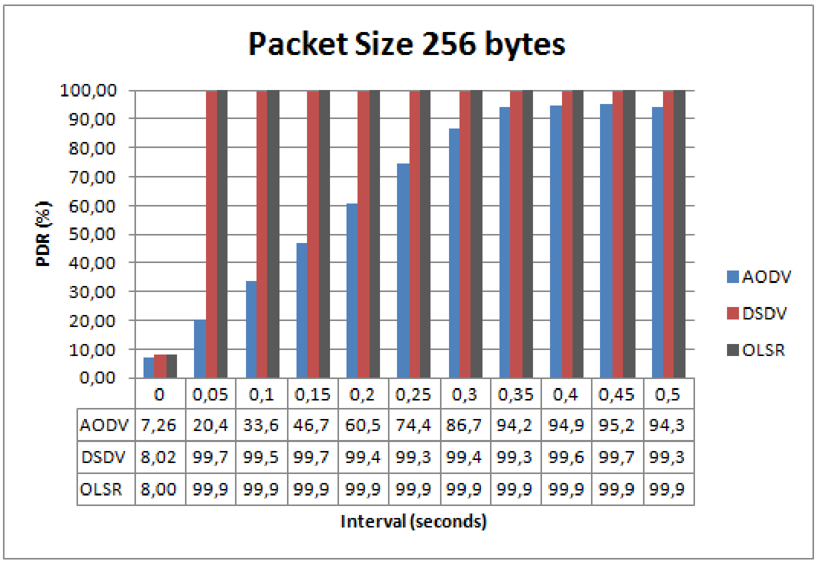


Figure 60 PDR vs. interval, DSSS Rate 2Mbps, Packet Size 256 bytes.

In Figure 60 we can see that routing protocols DSDV and OLSR outperform AODV in every case. DSDV outperforms OLSR only when the interval is 0 seconds.

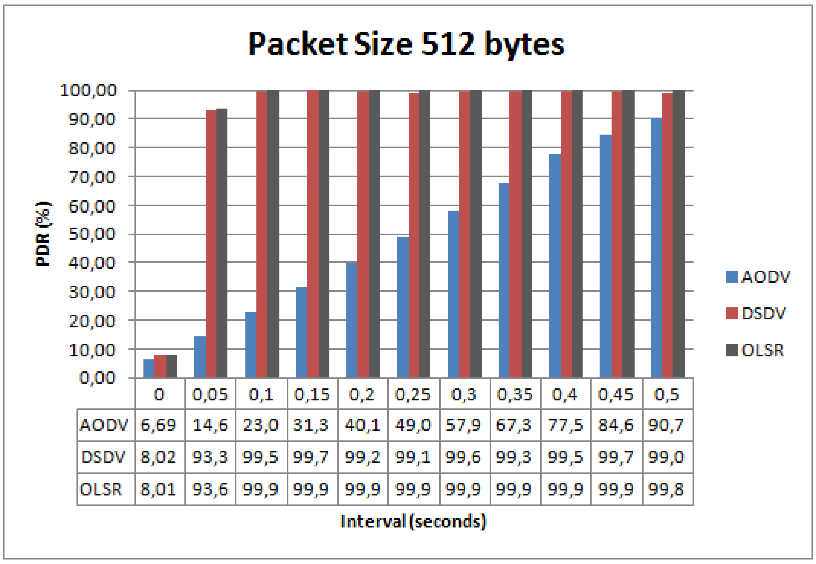
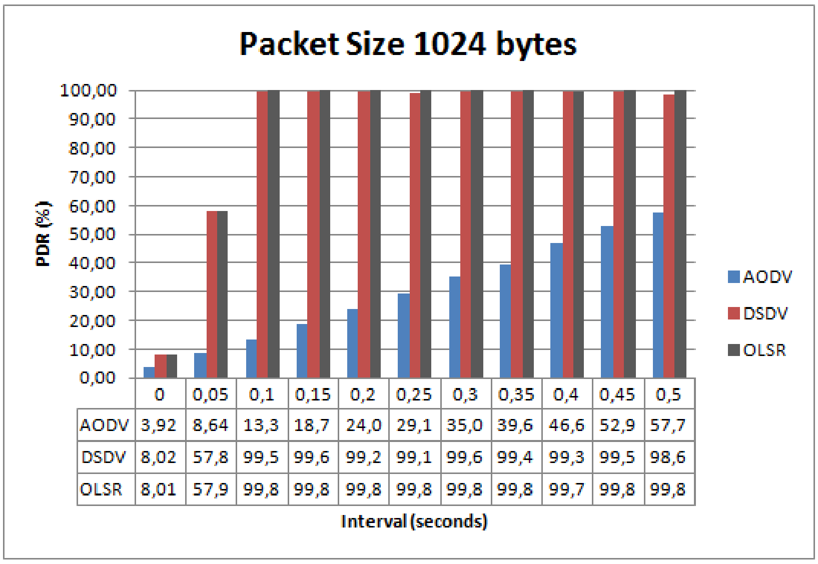


Figure 61 PDR vs. interval, DSSS Rate 2Mbps, Packet Size 512 bytes.

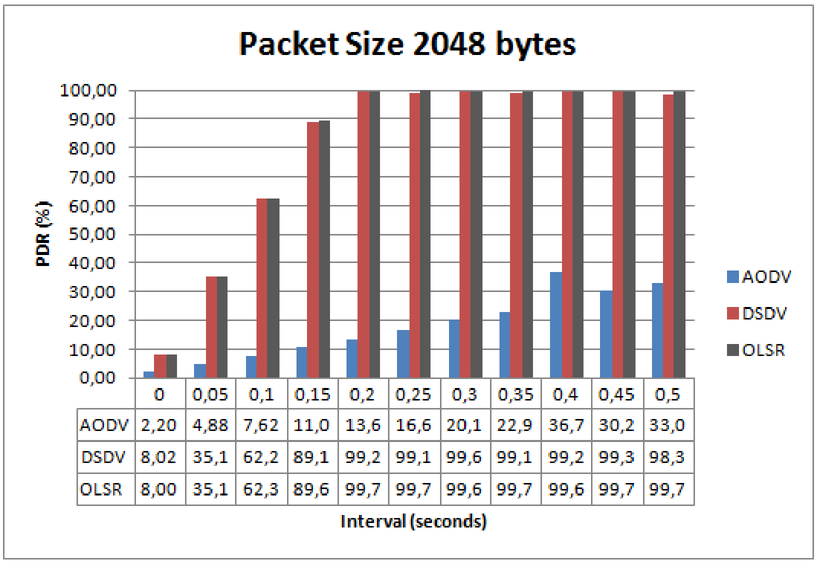
In Figure 61 we can see that routing protocols DSDV and OLSR outperform AODV in every case. DSDV outperforms OLSR only when the interval is 0 seconds.



	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
AODV	3,92	8,64	13,3	18,7	24,0	29,1	35,0	39,6	46,6	52,9	57,7
DSDV	8,02	57,8	99,5	99,6	99,2	99,1	99,6	99,4	99,3	99,5	98,6
OLSR	8,01	57,9	99,8	99,8	99,8	99,8	99,8	99,8	99,7	99,8	99,8

Figure 62 PDR vs. interval, DSSS Rate 2Mbps, Packet Size 1024 bytes.

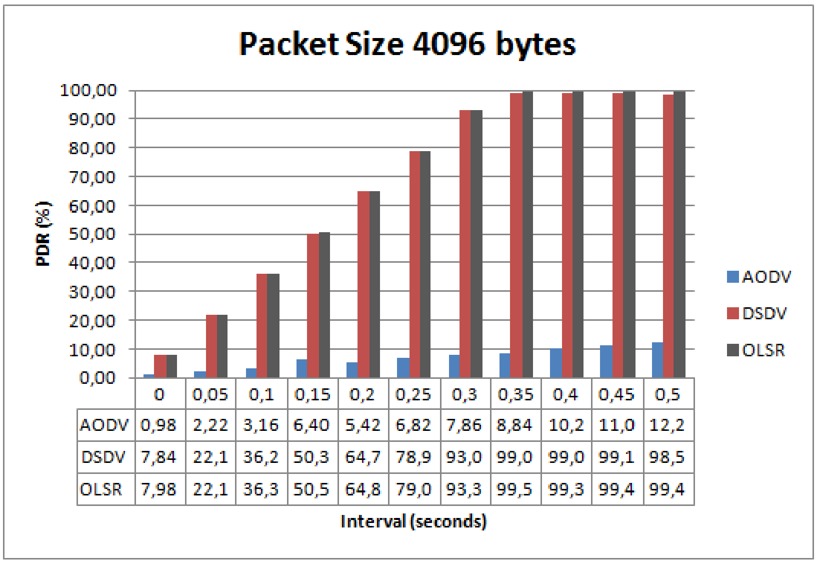
In Figure 62 we can see that routing protocols DSDV and OLSR outperform AODV in every case. DSDV outperforms OLSR only when the interval is 0 seconds.



	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
AODV	2,20	4,88	7,62	11,0	13,6	16,6	20,1	22,9	36,7	30,2	33,0
DSDV	8,02	35,1	62,2	89,1	99,2	99,1	99,6	99,1	99,2	99,3	98,3
OLSR	8,00	35,1	62,3	89,6	99,7	99,7	99,6	99,7	99,6	99,7	99,7

Figure 63 PDR vs. interval, DSSS Rate 2Mbps, Packet Size 2048 bytes.

In Figure 63 we can see that routing protocols DSDV and OLSR outperform AODV in every case. DSDV outperforms OLSR only when the interval is 0 seconds.

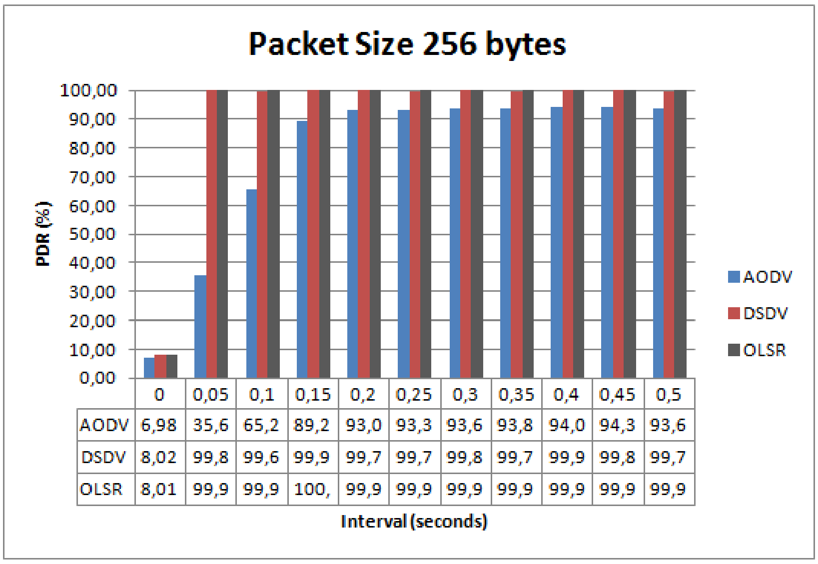


	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
AODV	0,98	2,22	3,16	6,40	5,42	6,82	7,86	8,84	10,2	11,0	12,2
DSDV	7,84	22,1	36,2	50,3	64,7	78,9	93,0	99,0	99,0	99,1	98,5
OLSR	7,98	22,1	36,3	50,5	64,8	79,0	93,3	99,5	99,3	99,4	99,4

Figure 64 PDR vs. interval, DSSS Rate 2Mbps, Packet Size 4096 bytes.

In Figure 64 we can see that routing protocols DSDV and OLSR outperform AODV in every case. DSDV in no case outperforms OLSR.

3. DSSS Rate 5.5Mbps



	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
AODV	6,98	35,6	65,2	89,2	93,0	93,3	93,6	93,8	94,0	94,3	93,6
DSDV	8,02	99,8	99,6	99,9	99,7	99,7	99,8	99,7	99,9	99,8	99,7
OLSR	8,01	99,9	99,9	100,	99,9	99,9	99,9	99,9	99,9	99,9	99,9

Figure 65 PDR vs. interval, DSSS Rate 5.5Mbps, Packet Size 256 bytes.

In Figure 65 we can see that routing protocols DSDV and OLSR outperform AODV in every case. DSDV outperforms OLSR only when the interval is 0 seconds.

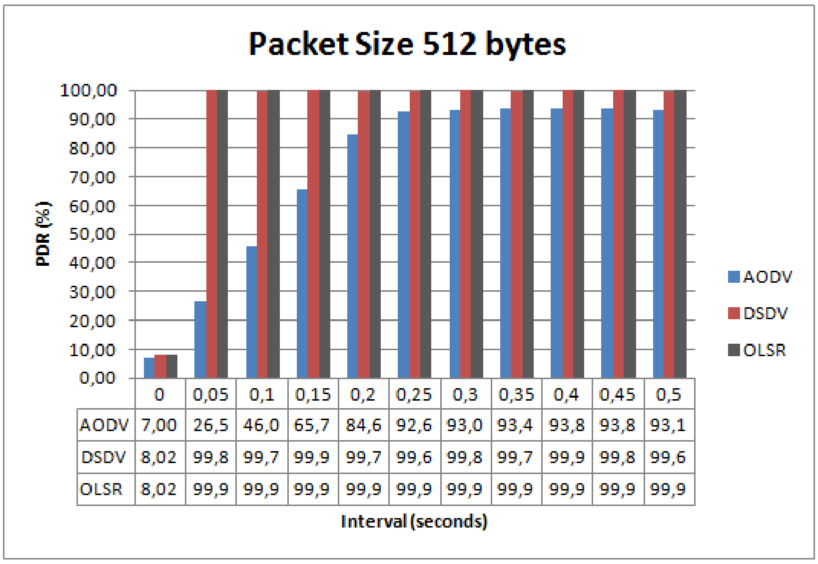


Figure 66 PDR vs. interval, DSSS Rate 5.5Mbps, Packet Size 512 bytes.

In Figure 66 we can see that routing protocols DSDV and OLSR outperform AODV in every case. DSDV never outperforms OLSR.

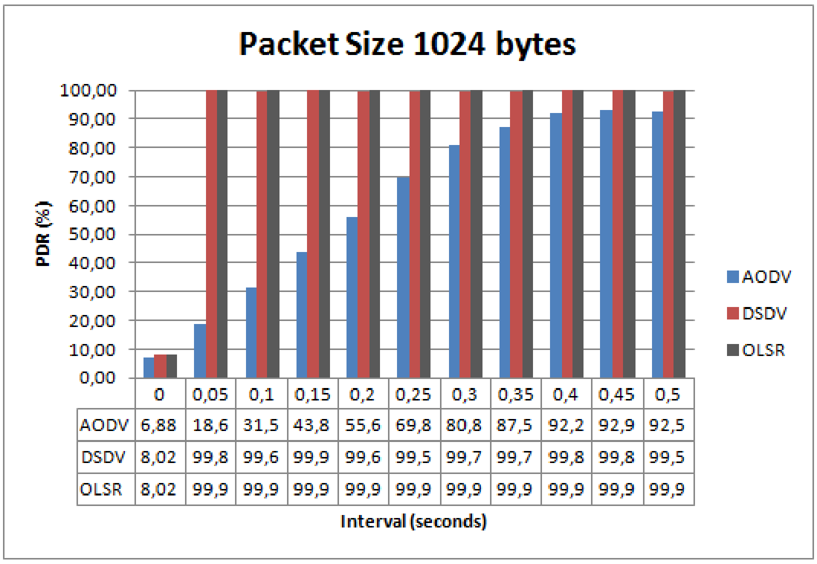
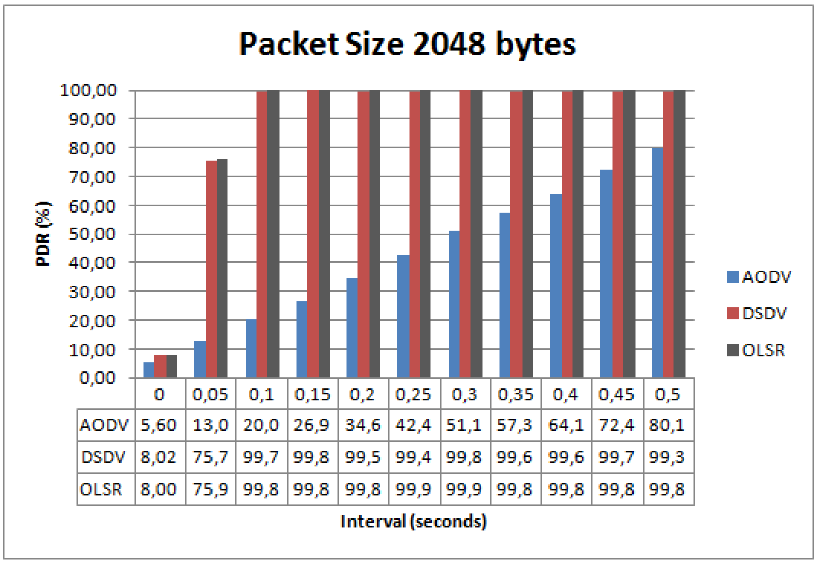


Figure 67 PDR vs. interval, DSSS Rate 5.5Mbps, Packet Size 1024 bytes.

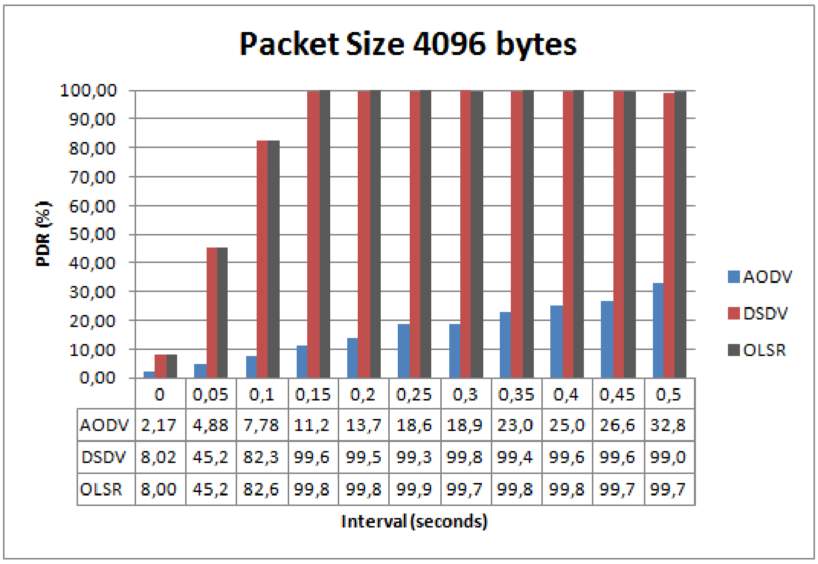
In Figure 67 we can see that routing protocols DSDV and OLSR outperform AODV in every case. DSDV never outperforms OLSR.



	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
AODV	5,60	13,0	20,0	26,9	34,6	42,4	51,1	57,3	64,1	72,4	80,1
DSDV	8,02	75,7	99,7	99,8	99,5	99,4	99,8	99,6	99,6	99,7	99,3
OLSR	8,00	75,9	99,8	99,8	99,8	99,9	99,9	99,8	99,8	99,8	99,8

Figure 68 PDR vs. interval, DSSS Rate 5.5Mbps, Packet Size 2048 bytes.

In Figure 68 we can see that routing protocols DSDV and OLSR outperform AODV in every case. DSDV outperforms OLSR only when the interval is 0 seconds.



	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
AODV	2,17	4,88	7,78	11,2	13,7	18,6	18,9	23,0	25,0	26,6	32,8
DSDV	8,02	45,2	82,3	99,6	99,5	99,3	99,8	99,4	99,6	99,6	99,0
OLSR	8,00	45,2	82,6	99,8	99,8	99,9	99,7	99,8	99,8	99,7	99,7

Figure 69 PDR vs. interval, DSSS Rate 5.5Mbps, Packet Size 4096 bytes.

In Figure 69 we can see that routing protocols DSDV and OLSR outperform AODV in every case. DSDV outperforms OLSR only when the interval is 0 and 0.3 seconds.

4. DSSS Rate 11Mbps

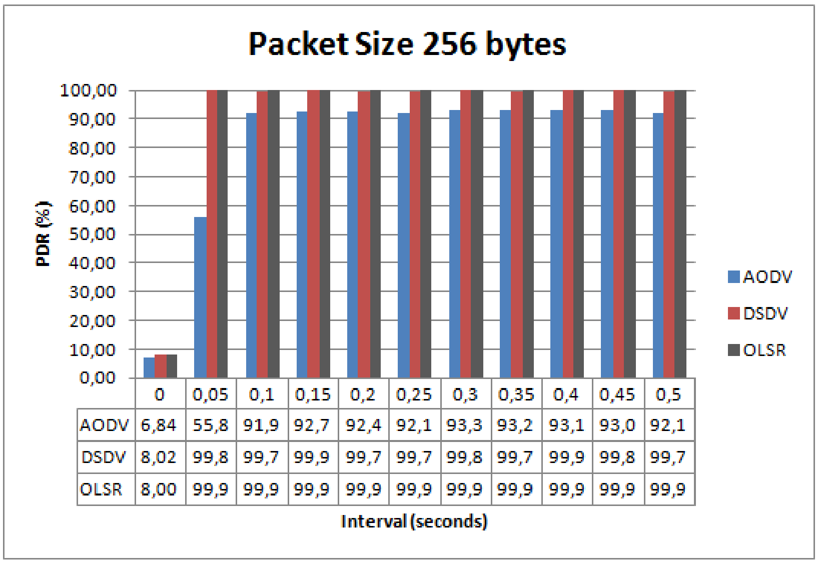


Figure 70 PDR vs. interval, DSSS Rate 11Mbps, Packet Size 256 bytes.

In Figure 70 we can see that routing protocols DSDV and OLSR outperform AODV in every case. DSDV outperforms OLSR only when the interval is 0 seconds.

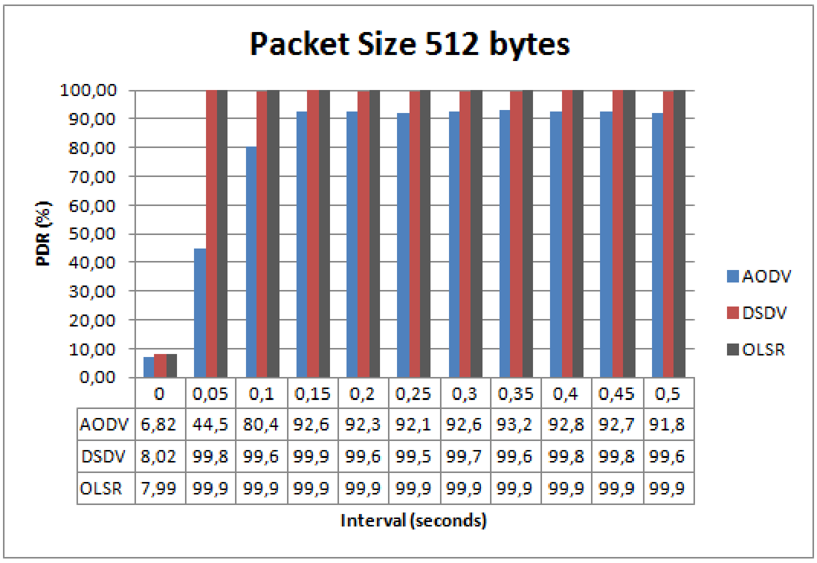
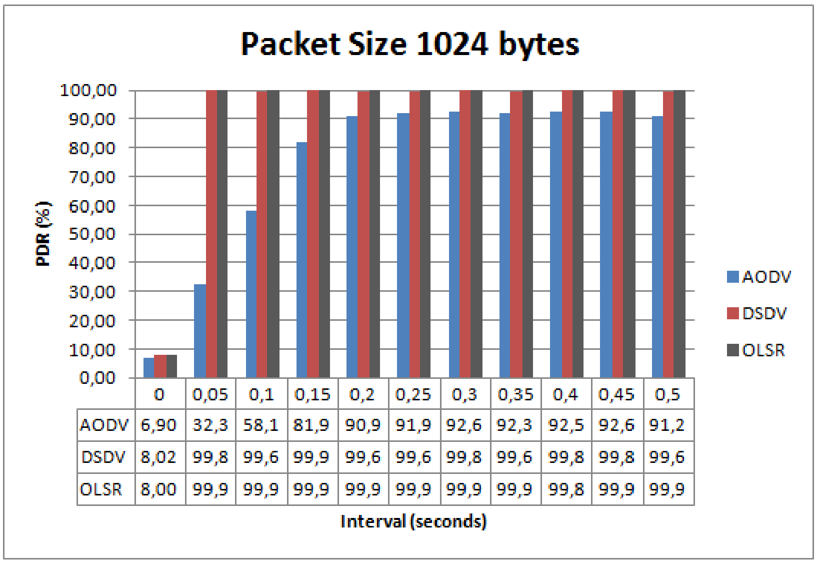


Figure 71 PDR vs. interval, DSSS Rate 11Mbps, Packet Size 512 bytes.

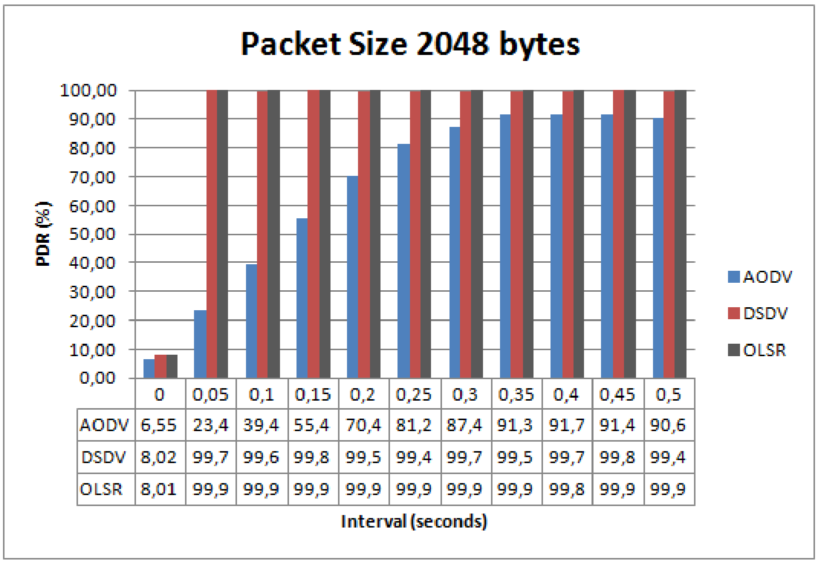
In Figure 71 we can see that routing protocols DSDV and OLSR outperform AODV in every case. DSDV outperforms OLSR only when the interval is 0 seconds.



	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
AODV	6,90	32,3	58,1	81,9	90,9	91,9	92,6	92,3	92,5	92,6	91,2
DSDV	8,02	99,8	99,6	99,9	99,6	99,6	99,8	99,6	99,8	99,8	99,6
OLSR	8,00	99,9	99,9	99,9	99,9	99,9	99,9	99,9	99,8	99,9	99,9

Figure 72 PDR vs. interval, DSSS Rate 11Mbps, Packet Size 1024 bytes.

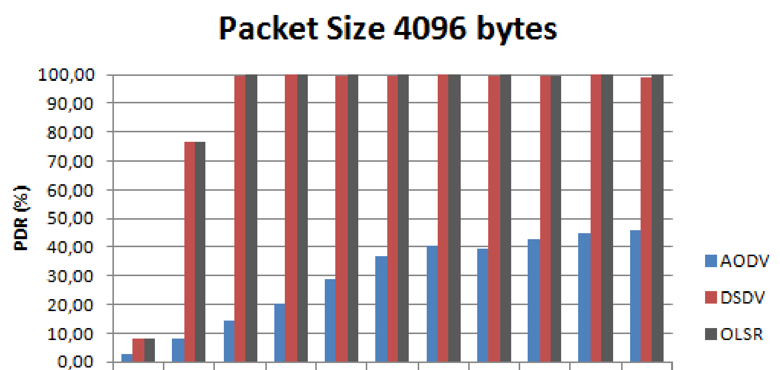
In Figure 72 we can see that routing protocols DSDV and OLSR outperform AODV in every case. DSDV outperforms OLSR only when the interval is 0 seconds.



	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
AODV	6,55	23,4	39,4	55,4	70,4	81,2	87,4	91,3	91,7	91,4	90,6
DSDV	8,02	99,7	99,6	99,8	99,5	99,4	99,7	99,5	99,7	99,8	99,4
OLSR	8,01	99,9	99,9	99,9	99,9	99,9	99,9	99,9	99,8	99,9	99,9

Figure 73 PDR vs. interval, DSSS Rate 11Mbps, Packet Size 2048 bytes.

In Figure 73 we can see that routing protocols DSDV and OLSR outperform AODV in every case. DSDV outperforms OLSR only when the interval is 0 seconds.



	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
AODV	2,73	8,22	14,6	20,3	28,8	36,9	40,5	39,6	42,8	44,9	46,0
DSDV	8,02	76,7	99,6	99,8	99,4	99,3	99,8	99,5	99,6	99,8	99,1
OLSR	8,00	76,9	99,8	99,8	99,8	99,8	99,7	99,8	99,7	99,8	99,8

Interval (seconds)

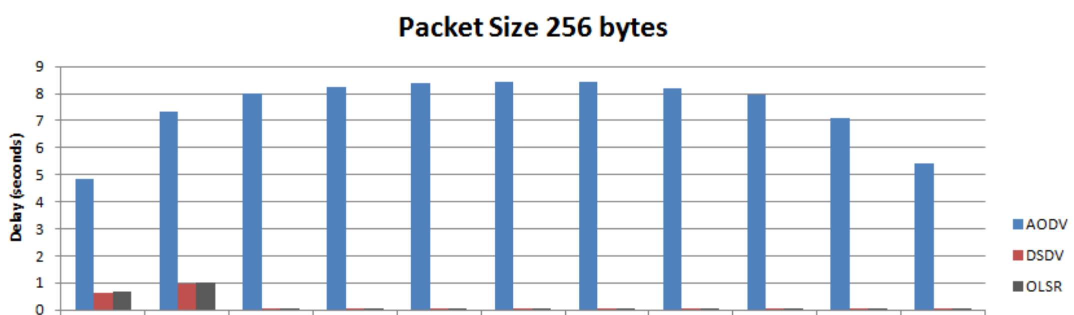
Figure 74 PDR vs. interval, DSSS Rate 11Mbps, Packet Size 4096 bytes.

In Figure 74 we can see that routing protocols DSDV and OLSR outperform AODV in every case. DSDV outperforms OLSR only when the interval is 0 and 0.3 seconds.

__Average Delay__

In Figures 75-94 we can compare the three routing protocols based on the average delay they achieve. The results are categorized based on the different DSSS Rates.

1. DSSS Rate 1Mbps

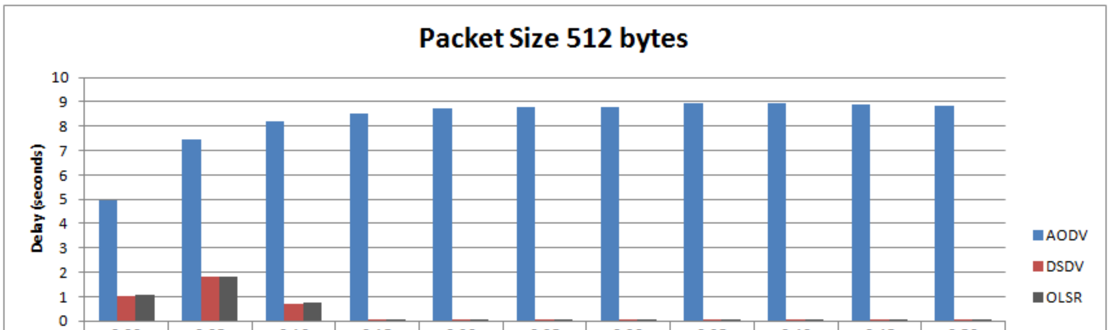


	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	4,82066	7,31461	8,00250	8,26311	8,38339	8,44940	8,43952	8,19078	7,94444	7,09593	5,40885
DSDV	0,6285770	0,9756730	0,0389371	0,0350230	0,0343341	0,0338209	0,0346340	0,0337035	0,0340253	0,0338831	0,0326925
OLSR	0,662201	1,006513	0,033036	0,033060	0,033145	0,033095	0,033041	0,032999	0,033436	0,033091	0,033025

Interval (seconds)

Figure 75 Delay vs. interval, DSSS Rate 1Mbps, Packet Size 256 bytes.

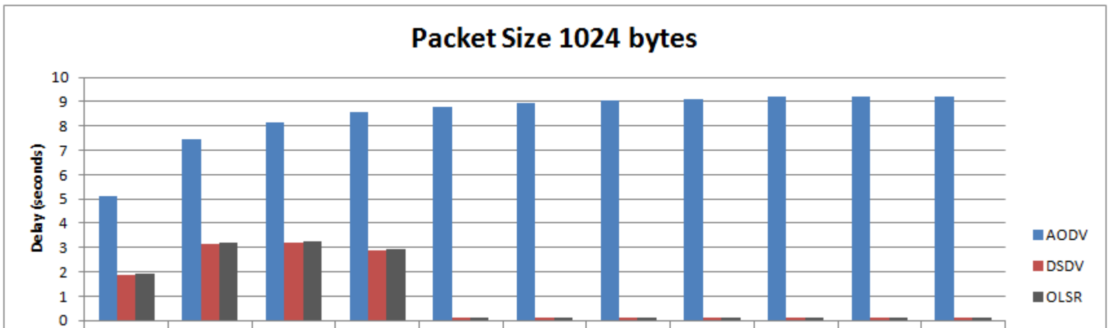
In Figure 75 we can observe that AODV has in every case the highest average delay. When the interval is 0, 0.05 and 0.50 seconds DSDV has the lowest delay and in all the other cases OLSR achieves the lowest delay.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	4,96547	7,43830	8,18503	8,54122	8,70761	8,78696	8,77272	8,92326	8,91783	8,86399	8,82606
DSDV	1,0402260	1,7961230	0,7275273	0,0595254	0,0563966	0,0559436	0,0563850	0,0552761	0,0555336	0,0555537	0,0542592
OLSR	1,077216	1,815013	0,777346	0,054942	0,055139	0,054995	0,054942	0,054906	0,055565	0,055007	0,054895

Figure 76 Delay vs. interval, DSSS Rate 1Mbps, Packet Size 512 bytes.

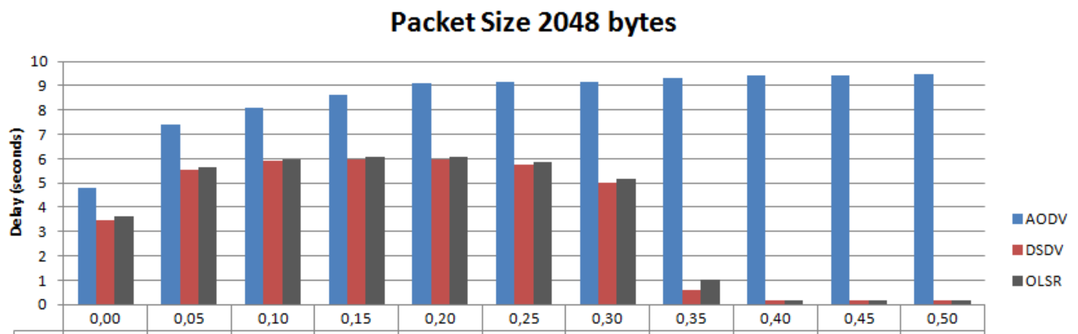
In Figure 76 we can observe that AODV has in every case the highest average delay. When the interval is 0, 0.05, 0.1, 0.4 and 0.50 seconds DSDV has the lowest delay and in all the other cases OLSR achieves the lowest delay.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	5,14005	7,47990	8,16349	8,56497	8,76963	8,94074	9,03611	9,08211	9,19427	9,22300	9,22933
DSDV	1,8635200	3,1341000	3,1850160	2,8814233	0,1115190	0,1006590	0,1003815	0,0988425	0,0986675	0,0982378	0,0971086
OLSR	1,921876	3,179716	3,235440	2,944263	0,100071	0,098776	0,098757	0,098704	0,099525	0,098821	0,098670

Figure 77 Delay vs. interval, DSSS Rate 1Mbps, Packet Size 1024 bytes.

In Figure 77 we can observe that AODV has in every case the highest average delay. When the interval is 0, 0.05, 0.1, 0.15, 0.4, 0.45 and 0.50 seconds DSDV has the lowest delay and in all the other cases OLSR achieves the lowest delay.

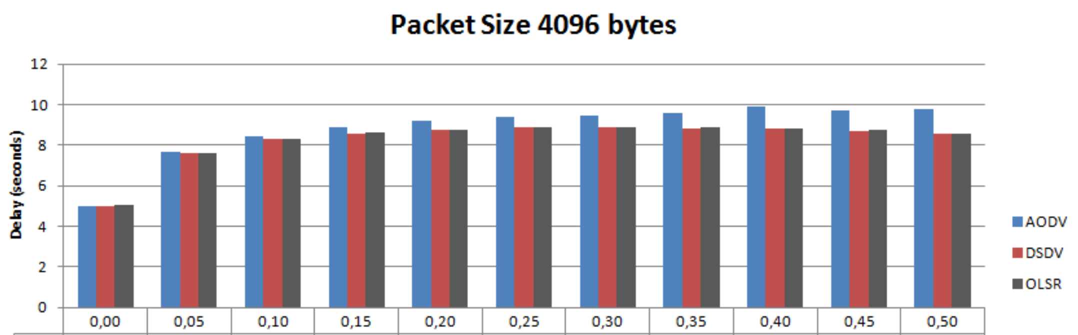


	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	4,78816	7,40678	8,08397	8,63073	9,10342	9,16561	9,17966	9,33158	9,41488	9,39544	9,49039
DSDV	3,4914230	5,5470760	5,9100860	5,9781630	5,9671600	5,7454700	4,9846060	0,5936550	0,1900950	0,1866800	0,1848736
OLSR	3,601233	5,620766	5,989926	6,091363	6,049280	5,846220	5,181620	1,044976	0,186888	0,186234	0,186232

Interval (seconds)

Figure 78 Delay vs. interval, DSSS Rate 1Mbps, Packet Size 2048 bytes.

In Figure 78 we can observe that AODV has in every case the highest average delay. When the interval is 0.4 and 0.45 seconds OLSR has the lowest delay and in all the other cases DSDV achieves the lowest delay.



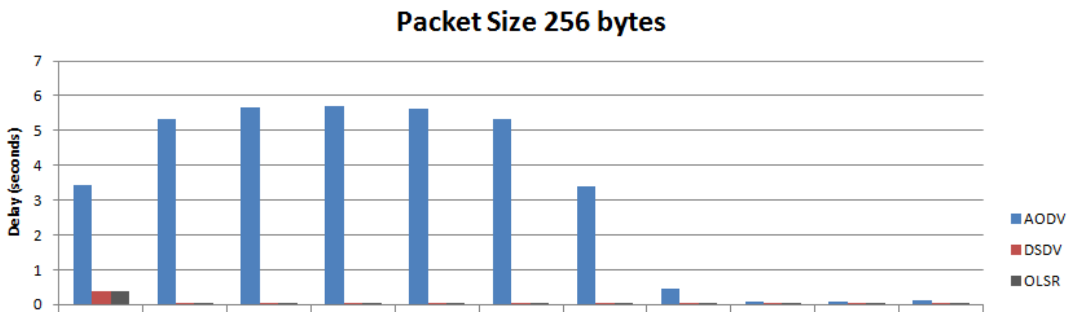
	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	5,00373	7,68344	8,45876	8,87002	9,19095	9,40523	9,44892	9,57675	9,87799	9,70573	9,78486
DSDV	5,0129230	7,5865230	8,3020460	8,5817000	8,7742930	8,8540530	8,8644730	8,8471160	8,8123830	8,6994400	8,5336830
OLSR	5,034976	7,601310	8,311573	8,619240	8,783723	8,858373	8,887383	8,873660	8,827770	8,733660	8,571360

Interval (seconds)

Figure 79 Delay vs. interval, DSSS Rate 1Mbps, Packet Size 4096 bytes.

In Figure 79 we can observe that AODV has the lowest average delay only when the interval is 0 seconds. In all the other cases DSDV achieves the lowest delay.

2. DSSS Rate 2Mbps

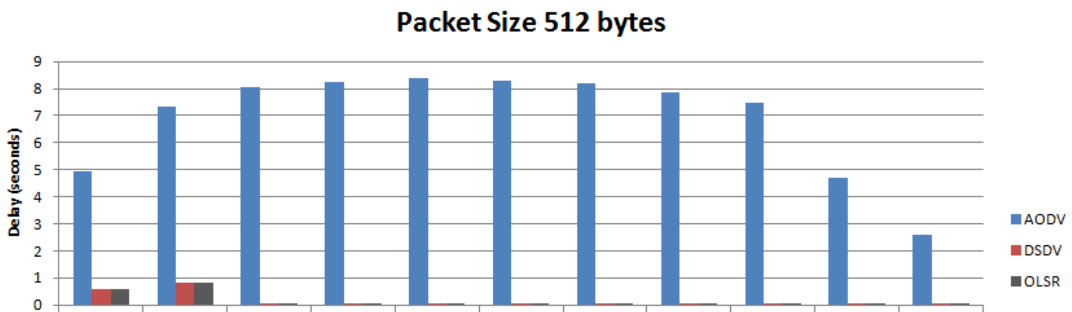


	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	3,41531	5,34710	5,67888	5,71925	5,62295	5,32458	3,40995	0,44458	0,10139	0,09977	0,10303
DSDV	0,3684663	0,0230653	0,0204977	0,0198883	0,0193635	0,0191456	0,0204129	0,0196864	0,0194143	0,0196864	0,0191351
OLSR	0,385490	0,019223	0,019216	0,019186	0,019226	0,019220	0,019136	0,019158	0,019299	0,019185	0,019185

Interval (seconds)

Figure 80 Delay vs. interval, DSSS Rate 2Mbps, Packet Size 256 bytes.

In Figure 80 we can observe that AODV has in every case the highest average delay. When the interval is 0 and 0.5 seconds DSDV has the lowest delay and in all the other cases OLSR achieves the lowest delay.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	4,92490	7,32290	8,06387	8,23724	8,37687	8,26885	8,16883	7,87732	7,45205	4,70065	2,59086
DSDV	0,5742903	0,8079040	0,0323183	0,0307905	0,0302677	0,0299665	0,0314806	0,0305152	0,0301274	0,0303572	0,0300270
OLSR	0,594569	0,813860	0,030068	0,030050	0,030096	0,030077	0,029986	0,030016	0,030226	0,030066	0,030041

Interval (seconds)

Figure 81 Delay vs. interval, DSSS Rate 2Mbps, Packet Size 512 bytes.

In Figure 81 we can observe that AODV has in every case the highest average delay. When the interval is 0, 0.05, 0.25, 0.4 and 0.5 seconds DSDV has the lowest delay and in all the other cases OLSR achieves the lowest delay.

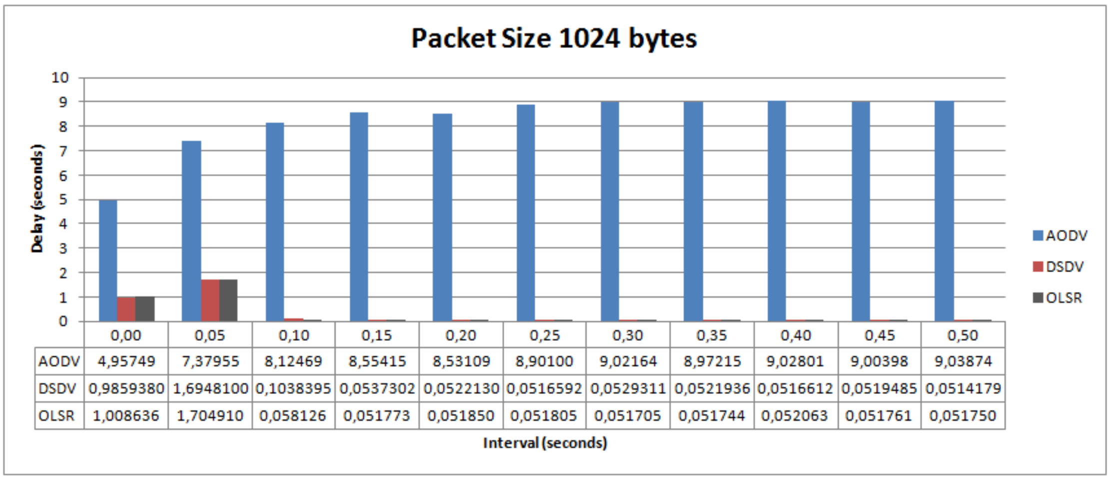


Figure 82 Delay vs. interval, DSSS Rate 2Mbps, Packet Size 1024 bytes.

In Figure 82 we can observe that AODV has in every case the highest average delay. When the interval is 0, 0.05, 0.25, 0.4 and 0.5 seconds DSDV has the lowest delay and in all the other cases OLSR achieves the lowest delay.

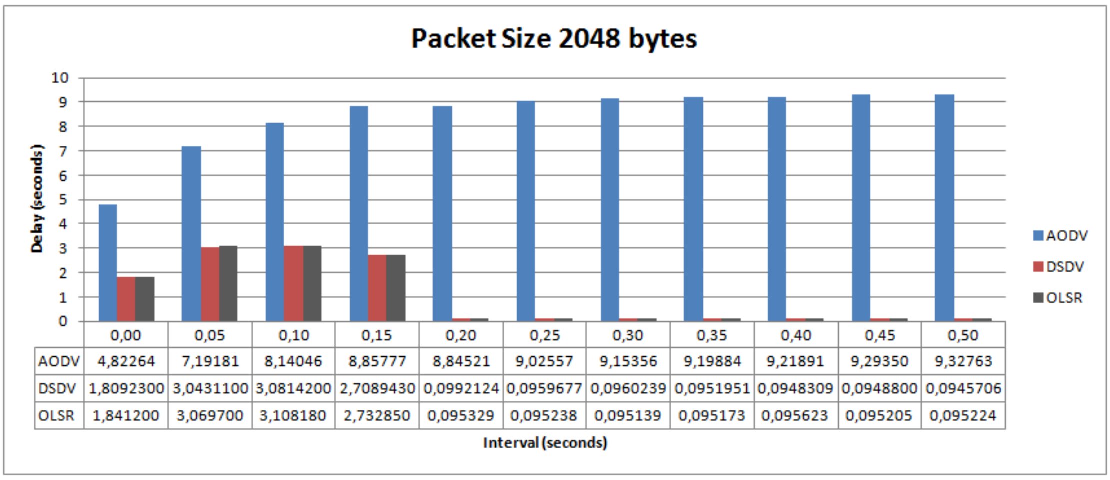


Figure 83 Delay vs. interval, DSSS Rate 2Mbps, Packet Size 2048 bytes.

In Figure 83 we can observe that AODV has in every case the highest average delay. When the interval is 0, 0.05, 0.1, 0.15, 0.4, 0.45 and 0.5 seconds DSDV has the lowest delay and in all the other cases OLSR achieves the lowest delay.

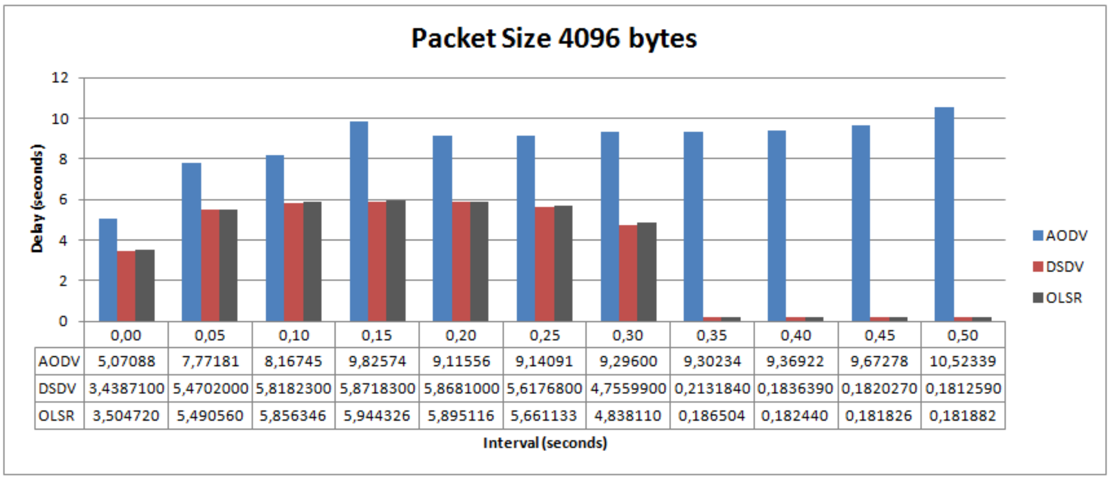


Figure 84 **Delay vs. interval, DSSS Rate 2Mbps, Packet Size 4096 bytes.**

In Figure 84 we can observe that AODV has in every case the highest average delay. When the interval is 0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3 and 0.5 seconds DSDV has the lowest delay and in all the other cases OLSR achieves the lowest delay.

3. DSSS Rate 5.5Mbps

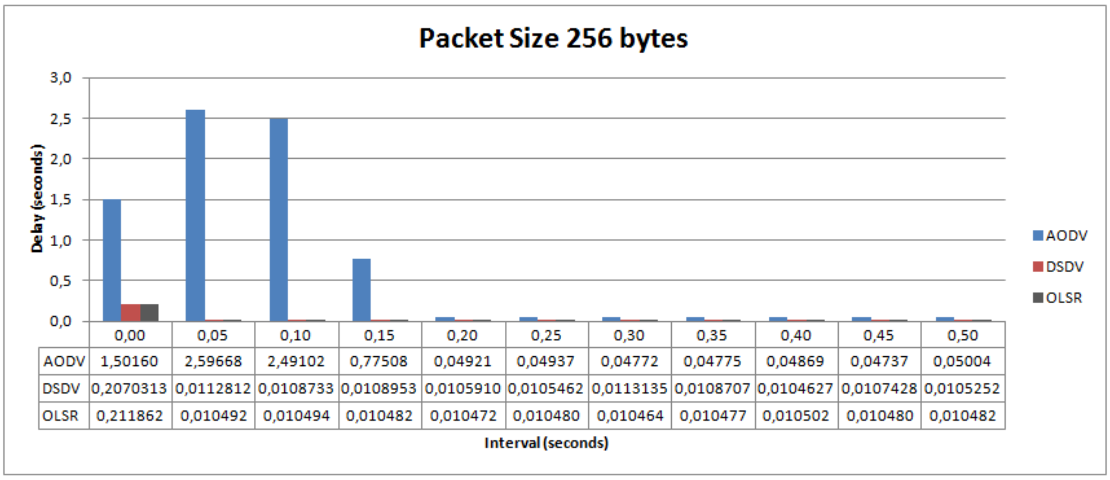


Figure 85 **Delay vs. interval, DSSS Rate 5.5Mbps, Packet Size 256 bytes.**

In Figure 85 we can observe that AODV has in every case the highest average delay. When the interval is 0 and 0.4 seconds DSDV has the lowest delay and in all the other cases OLSR achieves the lowest delay.

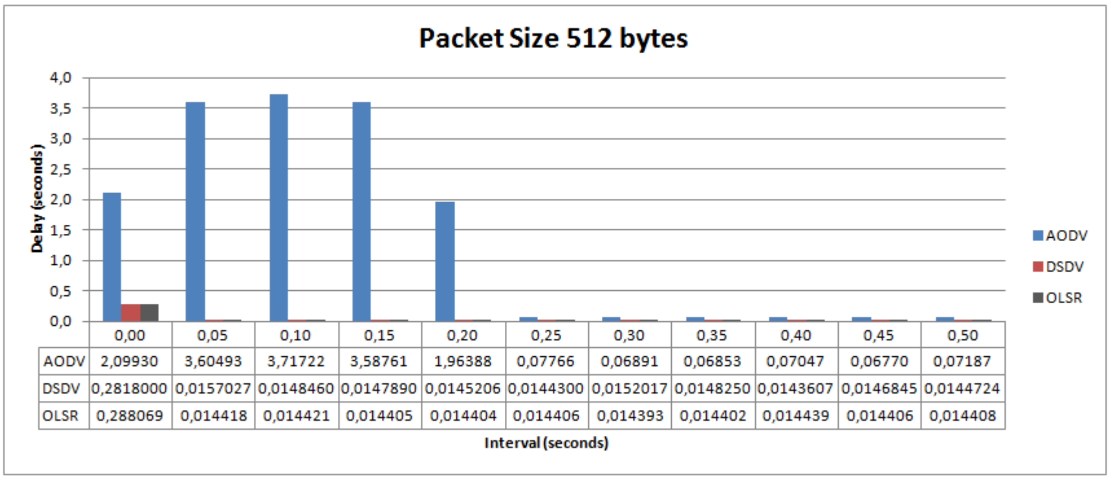


Figure 86 Delay vs. interval, DSSS Rate 5.5Mbps, Packet Size 512 bytes.

In Figure 86 we can observe that AODV has in every case the highest average delay. When the interval is 0 and 0.4 seconds DSDV has the lowest delay and in all the other cases OLSR achieves the lowest delay.

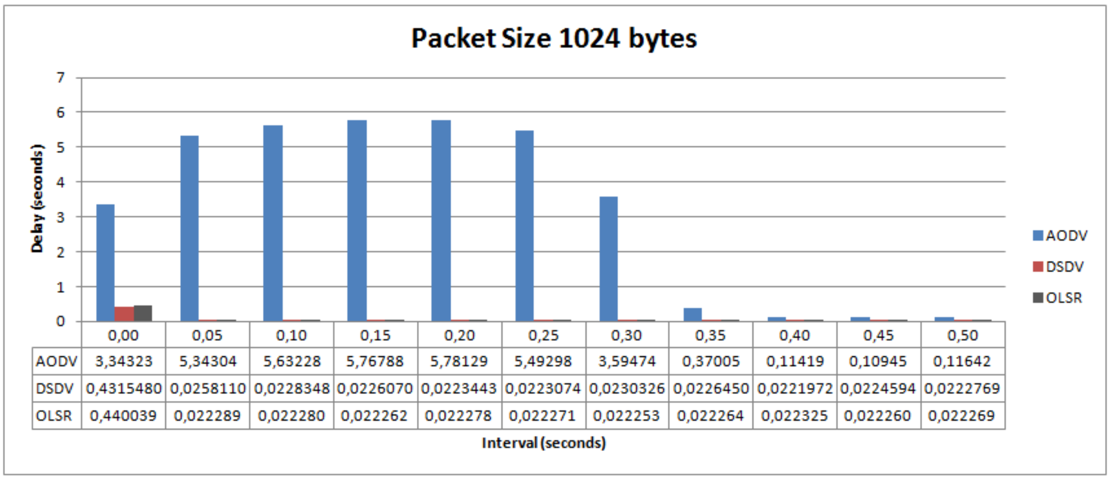
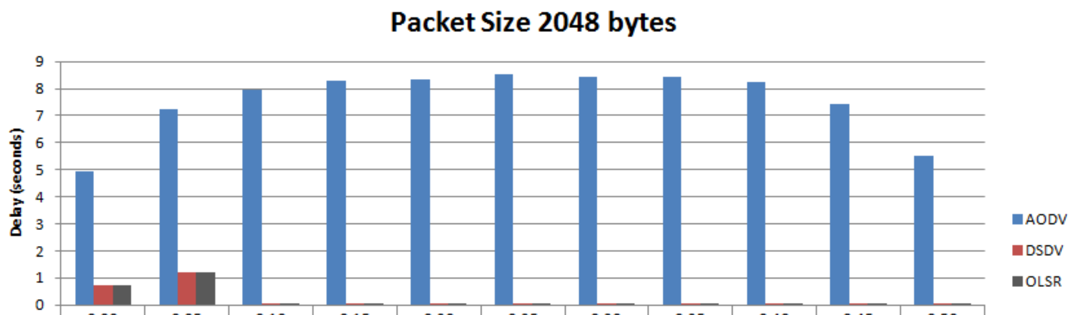


Figure 87 Delay vs. interval, DSSS Rate 5.5Mbps, Packet Size 1024 bytes.

In Figure 87 we can observe that AODV has in every case the highest average delay. When the interval is 0 and 0.4 seconds DSDV has the lowest delay and in all the other cases OLSR achieves the lowest delay.

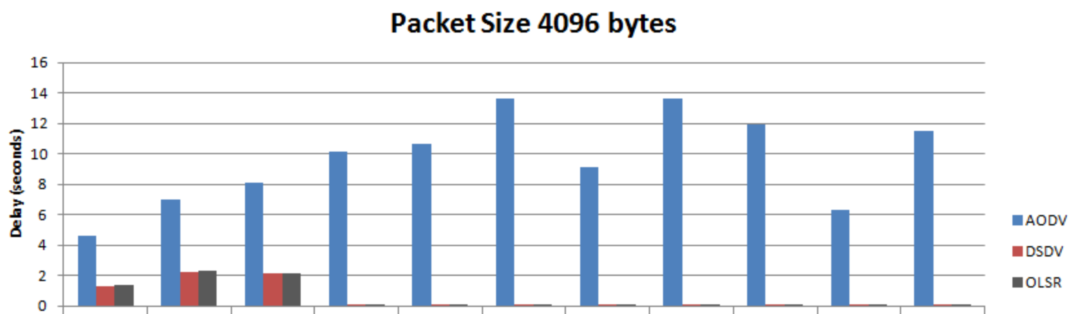


	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	4,95991	7,25142	7,97173	8,28556	8,33360	8,53469	8,44398	8,44863	8,22402	7,41152	5,51238
DSDV	0,7308370	1,2074500	0,0394058	0,0383666	0,0380100	0,0380100	0,0387450	0,0382221	0,0378759	0,0381081	0,0379988
OLSR	0,741245	1,213730	0,037983	0,037972	0,038002	0,037974	0,037976	0,037973	0,038079	0,037975	0,037974

Interval (seconds)

Figure 88 Delay vs. interval, DSSS Rate 5.5Mbps, Packet Size 2048 bytes.

In Figure 88 we can observe that AODV has in every case the highest average delay. When the interval is 0, 0.05, and 0.4 seconds DSDV has the lowest delay and in all the other cases OLSR achieves the lowest delay.



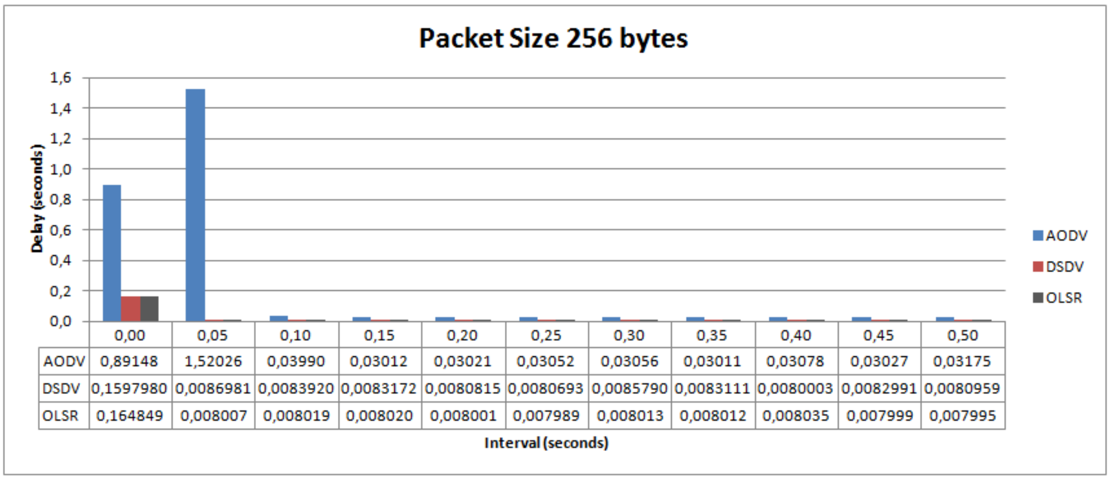
	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	4,59076	6,97764	8,06150	10,14484	10,65508	13,62168	9,15687	13,60900	11,93109	6,30423	11,53588
DSDV	1,3296100	2,2678000	2,1317400	0,0724070	0,0697569	0,0692130	0,0700100	0,0694890	0,0692173	0,0694352	0,0693059
OLSR	1,341710	2,275663	2,134176	0,069400	0,069465	0,069389	0,069403	0,069400	0,069598	0,069396	0,069383

Interval (seconds)

Figure 89 Delay vs. interval, DSSS Rate 5.5Mbps, Packet Size 4096 bytes.

In Figure 89 we can observe that AODV has in every case the highest average delay. When the interval is 0, 0.05, 0.1, 0.25, 0.4 and 0.5 seconds DSDV has the lowest delay and in all the other cases OLSR achieves the lowest delay.

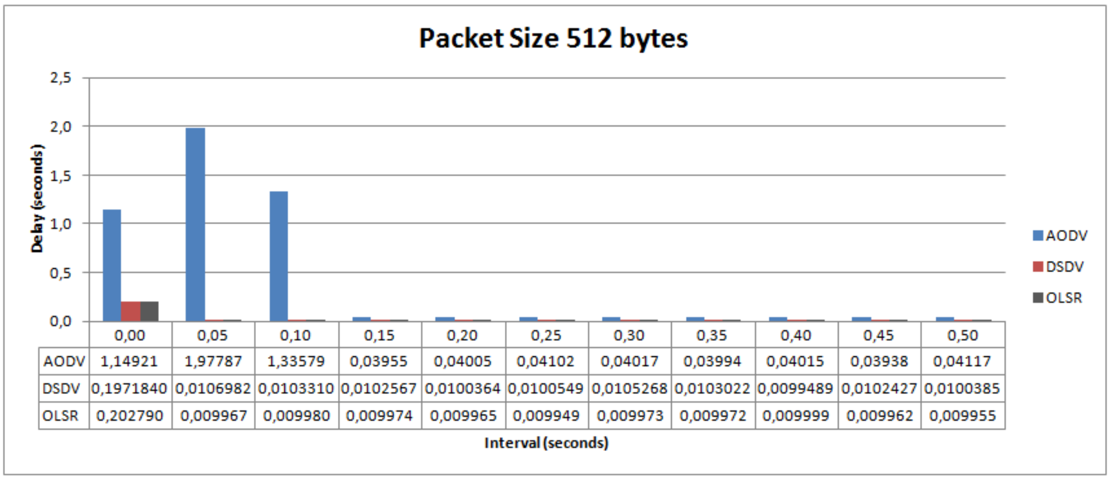
4. DSSS Rate 11Mbps



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	0,89148	1,52026	0,03990	0,03012	0,03021	0,03052	0,03056	0,03011	0,03078	0,03027	0,03175
DSDV	0,1597980	0,0086981	0,0083920	0,0083172	0,0080815	0,0080693	0,0085790	0,0083111	0,0080003	0,0082991	0,0080959
OLSR	0,164849	0,008007	0,008019	0,008020	0,008001	0,007989	0,008013	0,008012	0,008035	0,007999	0,007995

Figure 90 Delay vs. interval, DSSS Rate 11Mbps, Packet Size 256 bytes.

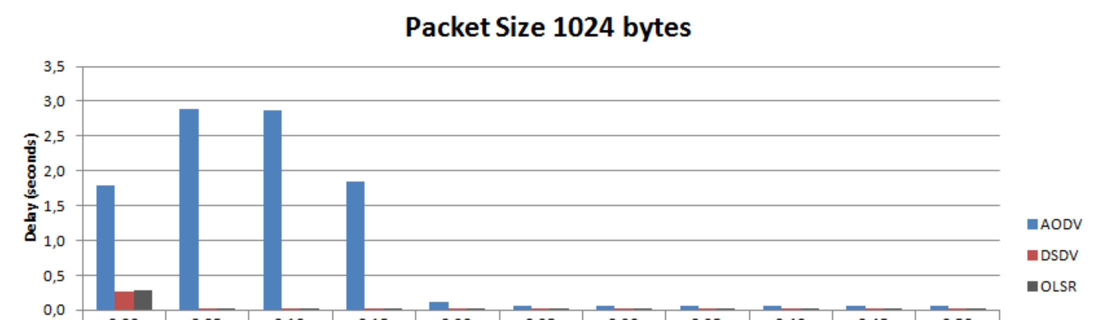
In Figure 90 we can observe that AODV has in every case the highest average delay. When the interval is 0 and 0.4 seconds DSDV has the lowest delay and in all the other cases OLSR achieves the lowest delay.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	1,14921	1,97787	1,33579	0,03955	0,04005	0,04102	0,04017	0,03994	0,04015	0,03938	0,04117
DSDV	0,1971840	0,0106982	0,0103310	0,0102567	0,0100364	0,0100549	0,0105268	0,0103022	0,0099489	0,0102427	0,0100385
OLSR	0,202790	0,009967	0,009980	0,009974	0,009965	0,009949	0,009973	0,009972	0,009999	0,009962	0,009955

Figure 91 Delay vs. interval, DSSS Rate 11Mbps, Packet Size 512 bytes.

In Figure 91 we can observe that AODV has in every case the highest average delay. When the interval is 0 and 0.4 seconds DSDV has the lowest delay and in all the other cases OLSR achieves the lowest delay.

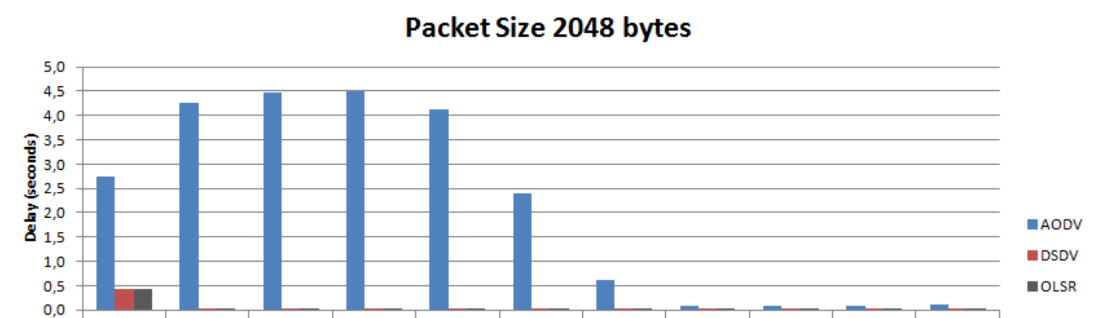


	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	1,78211	2,88761	2,87607	1,84811	0,12370	0,05975	0,05891	0,05868	0,05889	0,05773	0,06080
DSDV	0,2721570	0,0148035	0,0143057	0,0141488	0,0139676	0,0139700	0,0144358	0,0142051	0,0139072	0,0141440	0,0139449
OLSR	0,279226	0,013900	0,013907	0,013908	0,013900	0,013876	0,013903	0,013910	0,013929	0,013895	0,013886

Interval (seconds)

Figure 92 Delay vs. interval, DSSS Rate 11Mbps, Packet Size 1024 bytes.

In Figure 92 we can observe that AODV has in every case the highest average delay. When the interval is 0 and 0.4 seconds DSDV has the lowest delay and in all the other cases OLSR achieves the lowest delay.

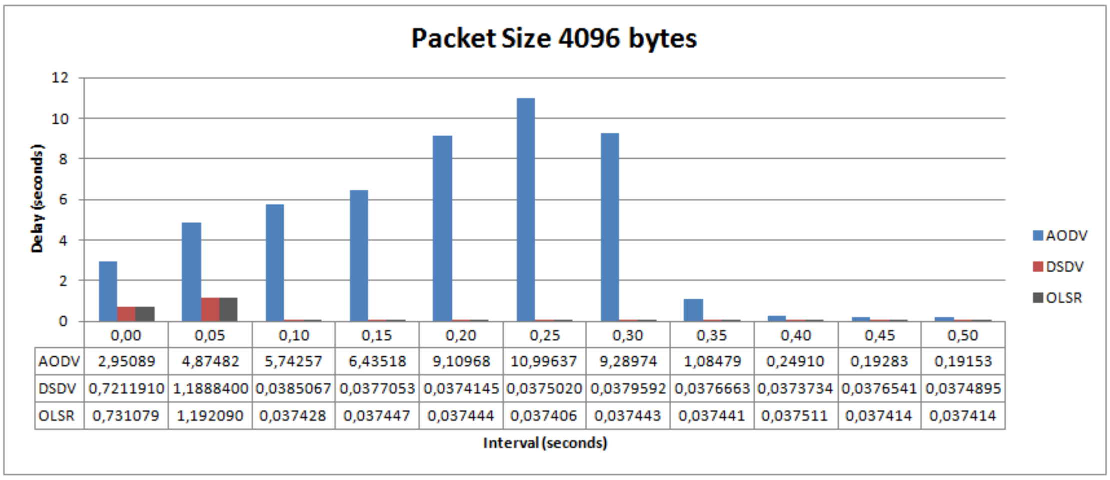


	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	2,73537	4,26280	4,47530	4,49218	4,12339	2,38707	0,61028	0,09763	0,09927	0,09702	0,10007
DSDV	0,4217016	0,0244266	0,0221921	0,0219621	0,0217900	0,0217990	0,0222504	0,0220577	0,0217294	0,0219710	0,0218291
OLSR	0,429649	0,021741	0,021745	0,021750	0,021742	0,021714	0,021740	0,021733	0,021784	0,021735	0,021725

Interval (seconds)

Figure 93 Delay vs. interval, DSSS Rate 11Mbps, Packet Size 2048 bytes.

In Figure 93 we can observe that AODV has in every case the highest average delay. When the interval is 0 and 0.4 seconds DSDV has the lowest delay and in all the other cases OLSR achieves the lowest delay.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	2,95089	4,87482	5,74257	6,43518	9,10968	10,99637	9,28974	1,08479	0,24910	0,19283	0,19153
DSDV	0,7211910	1,1888400	0,0385067	0,0377053	0,0374145	0,0375020	0,0379592	0,0376663	0,0373734	0,0376541	0,0374895
OLSR	0,731079	1,192090	0,037428	0,037447	0,037444	0,037406	0,037443	0,037441	0,037511	0,037414	0,037414

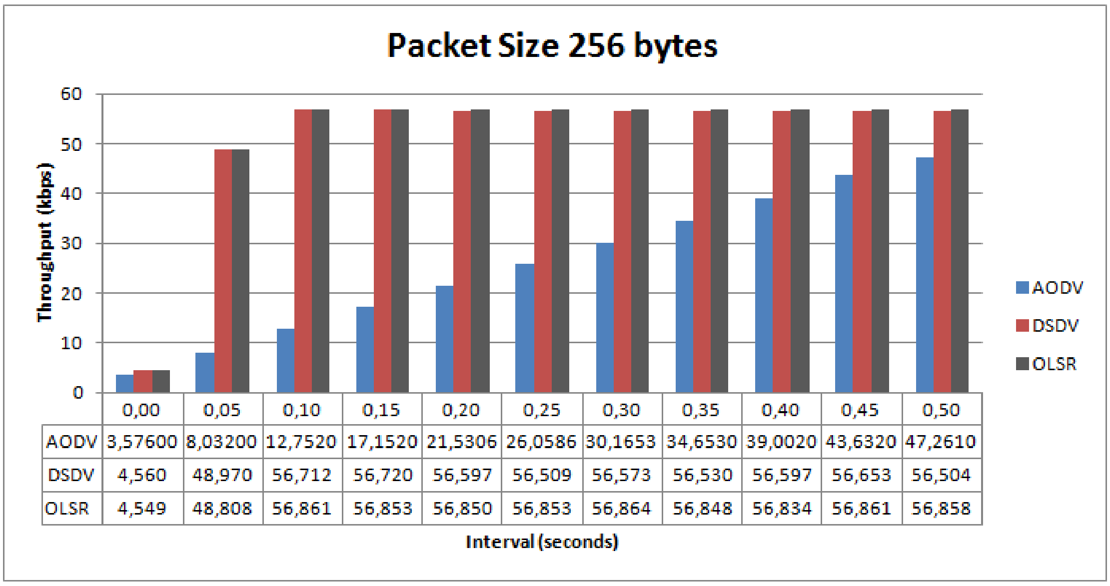
Figure 94 Delay vs. interval, DSSS Rate 11Mbps, Packet Size 4096 bytes.

In Figure 94 we can observe that AODV has in every case the highest average delay. When the interval is 0, 0.05, 0.2 and 0.4 seconds DSDV has the lowest delay and in all the other cases OLSR achieves the lowest delay.

<u>Throughput</u>

In Figures 95-114 we can compare the three routing protocols based on the throughput they achieve. The results are categorized based on the different DSSS Rates.

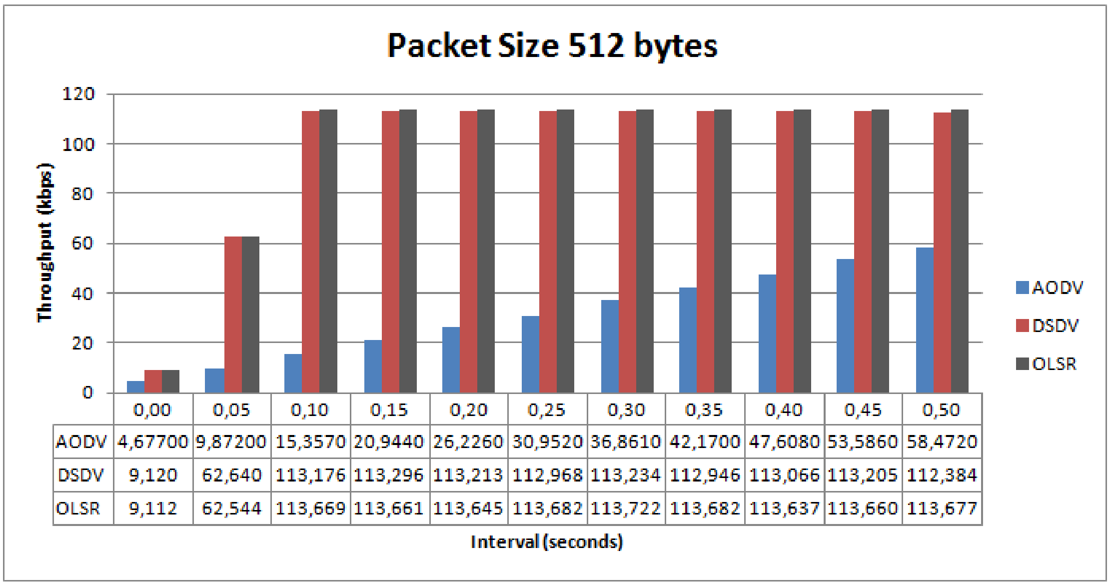
1. *DSSS Rate 1Mbps*



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	3,57600	8,03200	12,7520	17,1520	21,5306	26,0586	30,1653	34,6530	39,0020	43,6320	47,2610
DSDV	4,560	48,970	56,712	56,720	56,597	56,509	56,573	56,530	56,597	56,653	56,504
OLSR	4,549	48,808	56,861	56,853	56,850	56,853	56,864	56,848	56,834	56,861	56,858

Figure 95 Throughput vs. interval, DSSS Rate 1Mbps, Packet Size 256 bytes.

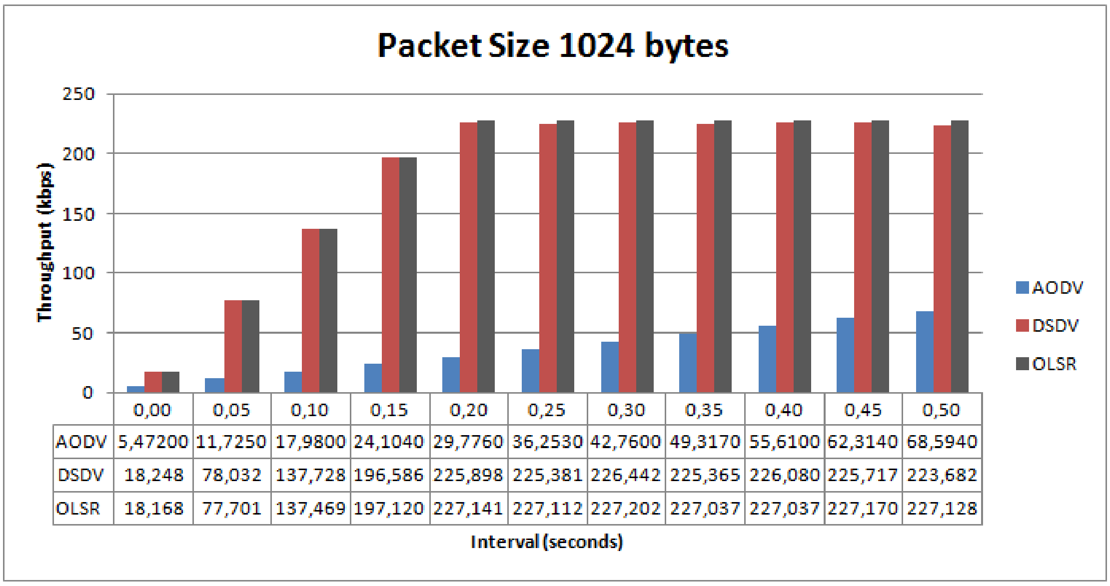
In Figure 95 we can see that AODV has in every case the lowest throughput. When the interval is 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45 and 0.5seconds OLSR has highest throughput than DSDV. In all the other cases DSDV achieves highest throughput than OLSR.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	4,67700	9,87200	15,3570	20,9440	26,2260	30,9520	36,8610	42,1700	47,6080	53,5860	58,4720
DSDV	9,120	62,640	113,176	113,296	113,213	112,968	113,234	112,946	113,066	113,205	112,384
OLSR	9,112	62,544	113,669	113,661	113,645	113,682	113,722	113,682	113,637	113,660	113,677

Figure 96 Throughput vs. interval, DSSS Rate 1Mbps, Packet Size 512 bytes.

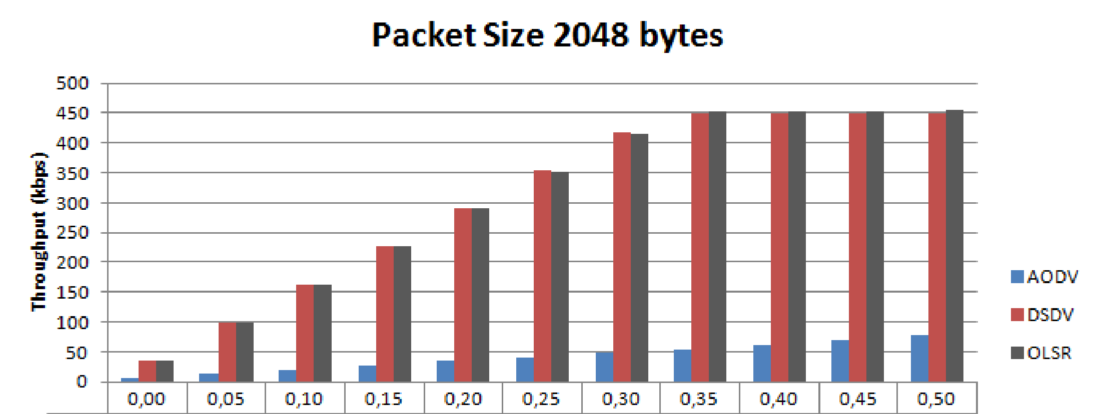
In Figure 96 we can see that AODV has in every case the lowest throughput. When the interval is 0.1, 0.15, 0.2, 0.3, 0.35, 0.4, 0.45 and 0.5seconds OLSR has highest throughput than DSDV. In all the other cases DSDV achieves highest throughput than OLSR.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	5,47200	11,7250	17,9800	24,1040	29,7760	36,2530	42,7600	49,3170	55,6100	62,3140	68,5940
DSDV	18,248	78,032	137,728	196,586	225,898	225,381	226,442	225,365	226,080	225,717	223,682
OLSR	18,168	77,701	137,469	197,120	227,141	227,112	227,202	227,037	227,037	227,170	227,128

Figure 97 Throughput vs. interval, DSSS Rate 1Mbps, Packet Size 1024 bytes.

In Figure 97 we can see that AODV has in every case the lowest throughput. When the interval is 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45 and 0.5seconds OLSR has highest throughput than DSDV. In all the other cases DSDV achieves highest throughput than OLSR.

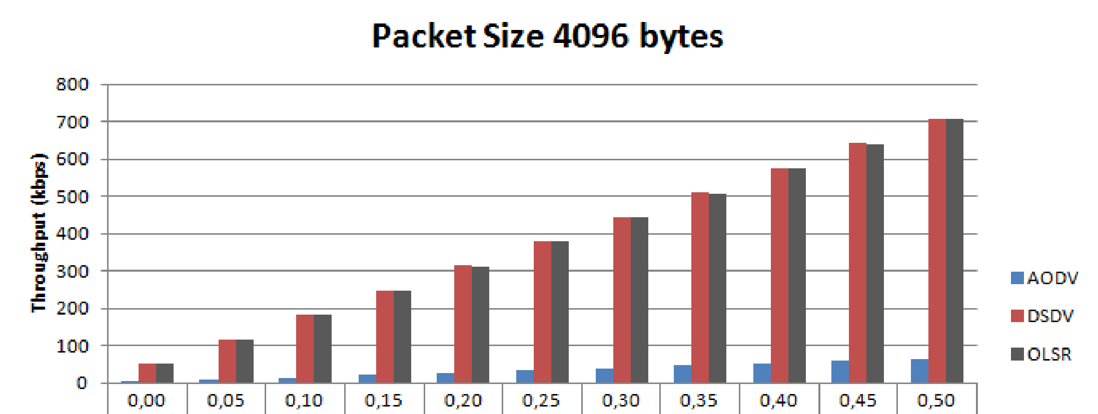


	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	5,63700	13,1040	20,1730	26,8480	34,0690	41,1380	47,4800	54,1250	62,1650	68,7460	77,6370
DSDV	35,525	99,664	162,530	225,669	290,266	353,253	416,696	449,920	450,738	450,768	449,010
OLSR	36,373	99,240	162,440	225,944	289,114	352,645	415,754	453,560	453,709	453,530	454,165

Interval (seconds)

Figure 98 Throughput vs. interval, DSSS Rate 1Mbps, Packet Size 2048 bytes.

In Figure 98 we can see that AODV has in every case the lowest throughput. When the interval is 0, 0.15, 0.35, 0.4, 0.45 and 0.5 seconds OLSR has highest throughput than DSDV. In all the other cases DSDV achieves highest throughput than OLSR.



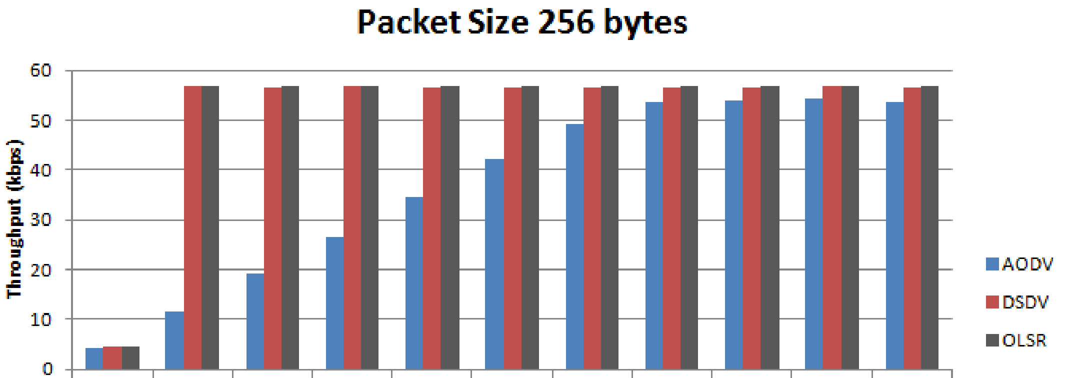
	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	4,91200	10,4340	14,7410	21,1120	26,9970	33,8560	39,9840	45,8100	51,0880	60,0080	65,5330
DSDV	52,000	117,232	183,922	248,546	314,264	379,738	445,157	510,085	576,954	641,400	706,389
OLSR	52,546	117,474	181,920	247,514	312,685	378,282	443,032	508,322	573,376	639,152	704,933

Interval (seconds)

Figure 99 Throughput vs. interval, DSSS Rate 1Mbps, Packet Size 4096 bytes.

In Figure 99 we can see that AODV has in every case the lowest throughput. When the interval is 0 and 0.05 seconds OLSR has highest throughput than DSDV. In all the other cases DSDV achieves highest throughput than OLSR.

2. DSSS Rate 2Mbps

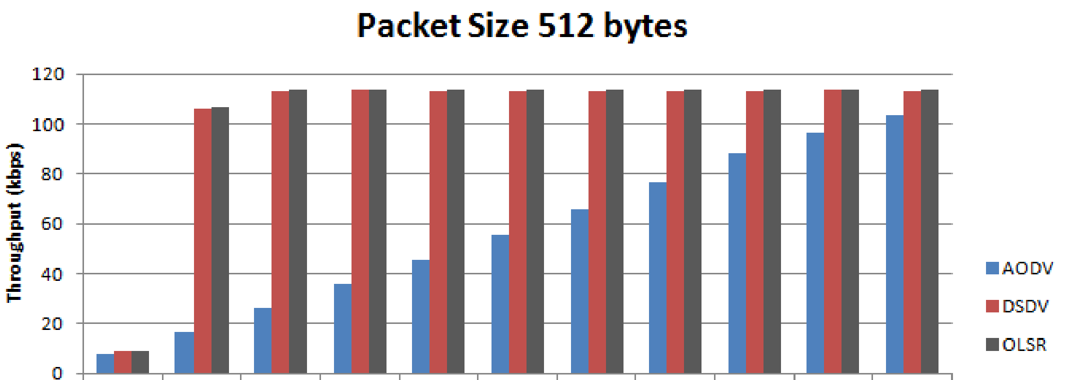


	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	4,13000	11,6340	19,1600	26,6000	34,4340	42,3300	49,3200	53,5940	54,0080	54,1620	53,6820
DSDV	4,560	56,752	56,624	56,765	56,560	56,490	56,570	56,498	56,664	56,762	56,525
OLSR	4,546	56,856	56,866	56,850	56,877	56,834	56,858	56,850	56,829	56,856	56,861

Interval (seconds)

Figure 100 Throughput vs. interval, DSSS Rate 2Mbps, Packet Size 256 bytes.

In Figure 100 we can see that AODV has in every case the lowest throughput. When the interval is 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45 and 0.5seconds OLSR has highest throughput than DSDV. In all the other cases DSDV achieves highest throughput than OLSR.

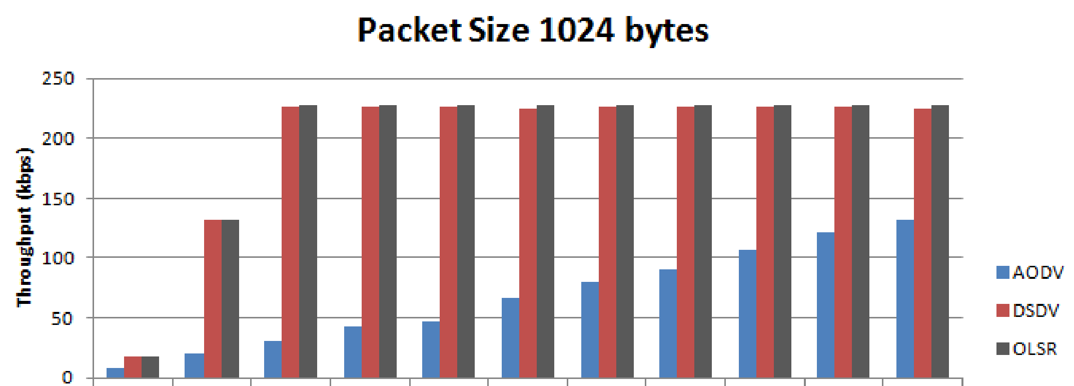


	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	7,60500	16,7160	26,2020	35,7060	45,6740	55,8370	65,8720	76,6720	88,1890	96,2500	103,224
DSDV	9,120	106,218	113,205	113,538	112,970	112,856	113,365	113,069	113,218	113,432	112,744
OLSR	9,112	106,509	113,669	113,706	113,677	113,674	113,690	113,720	113,669	113,690	113,624

Interval (seconds)

Figure 101 Throughput vs. interval, DSSS Rate 2Mbps, Packet Size 512 bytes.

In Figure 101 we can see that AODV has in every case the lowest throughput. When the interval is 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45 and 0.5seconds OLSR has highest throughput than DSDV. In all the other cases DSDV achieves highest throughput than OLSR.

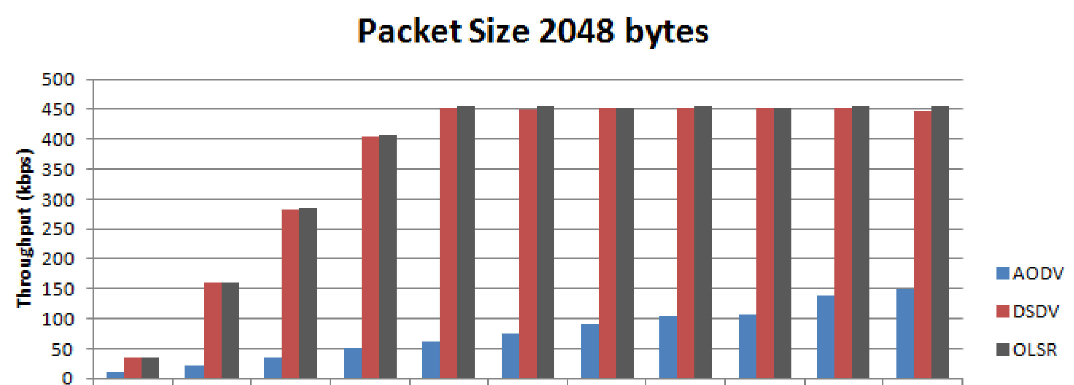


	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	8,91400	19,6580	30,4560	42,5493	47,1460	66,3520	79,6560	90,1840	106,173	120,586	131,296
DSDV	18,248	131,568	226,504	226,792	225,914	225,533	226,669	226,186	225,973	226,565	224,456
OLSR	18,232	131,765	227,277	227,202	227,202	227,202	227,157	227,277	227,050	227,112	227,173

Interval (seconds)

Figure 102 Throughput vs. interval, DSSS Rate 2Mbps, Packet Size 1024 bytes.

In Figure 102 we can see that AODV has in every case the lowest throughput. When the interval is 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45 and 0.5seconds OLSR has highest throughput than DSDV. In all the other cases DSDV achieves highest throughput than OLSR.

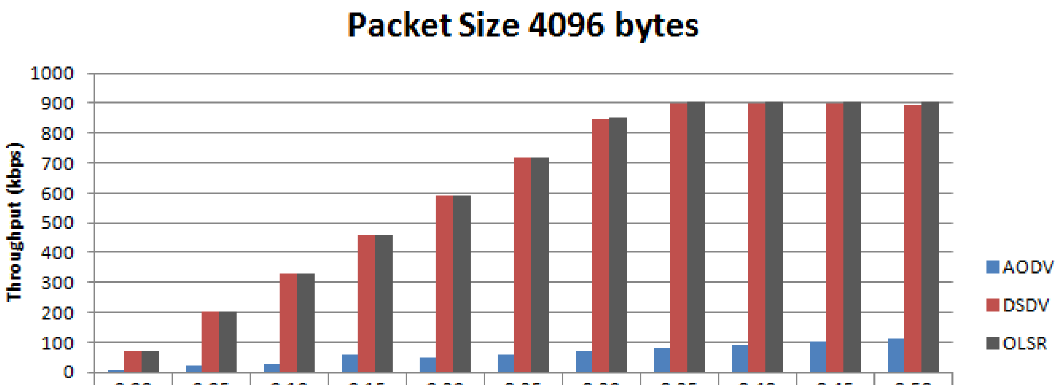


	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	10,0370	22,2340	34,6770	50,1493	62,0720	75,6600	91,7440	104,520	107,354	137,562	150,456
DSDV	36,496	159,890	283,226	405,560	451,858	451,013	453,682	451,194	451,858	452,101	447,642
OLSR	36,408	160,165	283,956	407,776	453,925	454,168	453,589	454,045	453,650	454,045	453,832

Interval (seconds)

Figure 103 Throughput vs. interval, DSSS Rate 2Mbps, Packet Size 2048 bytes.

In Figure 103 we can see that AODV has in every case the lowest throughput. When the interval is 0.05, 0.1, 0.15, 0.2, 0.25, 0.35, 0.4, 0.45 and 0.5seconds OLSR has highest throughput than DSDV. In all the other cases DSDV achieves highest throughput than OLSR.

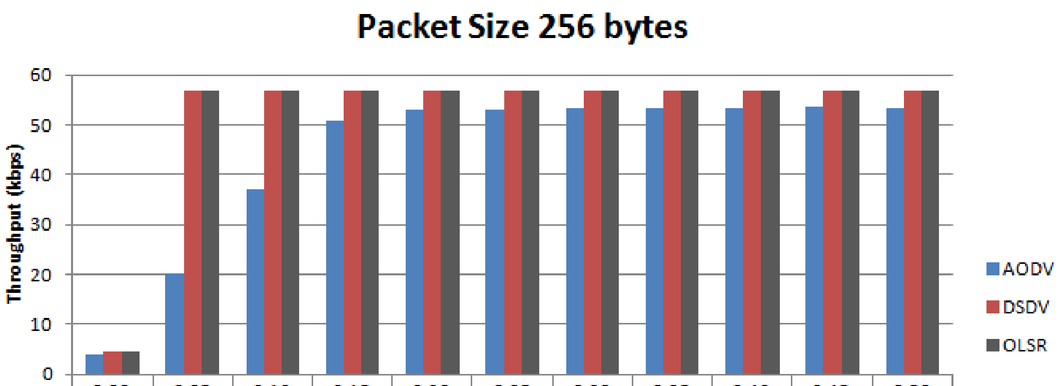


	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	8,91700	20,2640	28,8180	58,2480	49,3300	62,1330	71,5380	80,4610	93,0820	100,660	111,346
DSDV	71,357	201,882	329,741	458,202	589,394	718,528	847,232	901,421	901,845	902,026	896,688
OLSR	72,690	201,640	330,709	460,445	590,672	719,437	849,477	906,029	904,514	905,485	904,816

Figure 104 Throughput vs. interval, DSSS Rate 2Mbps, Packet Size 4096 bytes.

In Figure 104 we can see that AODV has in every case the lowest throughput. When the interval is 0, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45 and 0.5seconds OLSR has highest throughput than DSDV. In all the other cases DSDV achieves highest throughput than OLSR.

3. DSSS Rate 5.5Mbps



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	3,97000	20,2640	37,1280	50,7700	52,9330	53,1010	53,2580	53,3970	53,4770	53,6480	53,2640
DSDV	4,560	56,792	56,704	56,848	56,768	56,749	56,781	56,736	56,850	56,776	56,760
OLSR	4,554	56,882	56,874	56,888	56,861	56,861	56,869	56,880	56,861	56,880	56,872

Figure 105 Throughput vs. interval, DSSS Rate 5.5Mbps, Packet Size 256 bytes.

In Figure 105 we can see that AODV has in every case the lowest throughput. When the interval is 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45 and 0.5seconds OLSR has highest throughput than DSDV. In all the other cases DSDV achieves highest throughput than OLSR.

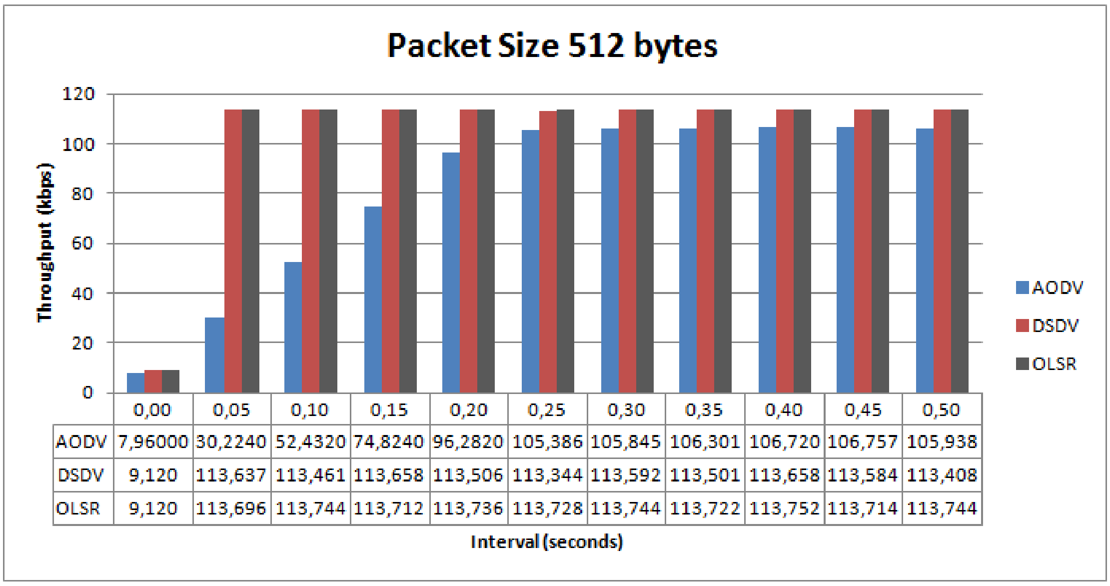


Figure 106 Throughput vs. interval, DSSS Rate 5.5Mbps, Packet Size 512 bytes.

In Figure 106 we can see that AODV has in every case the lowest throughput. OLSR has highest throughput and outperforms DSDV.

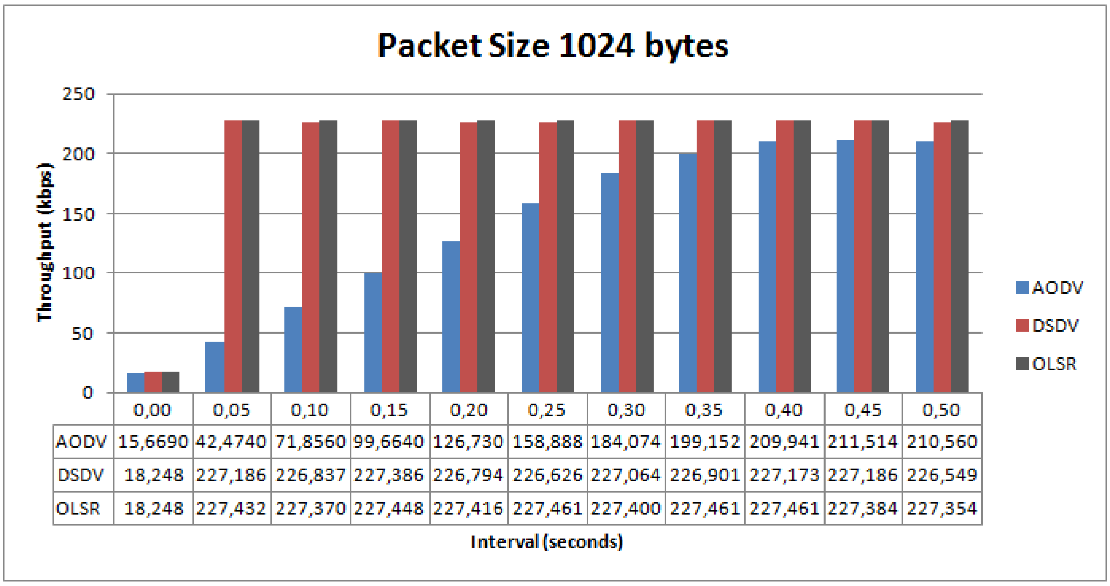
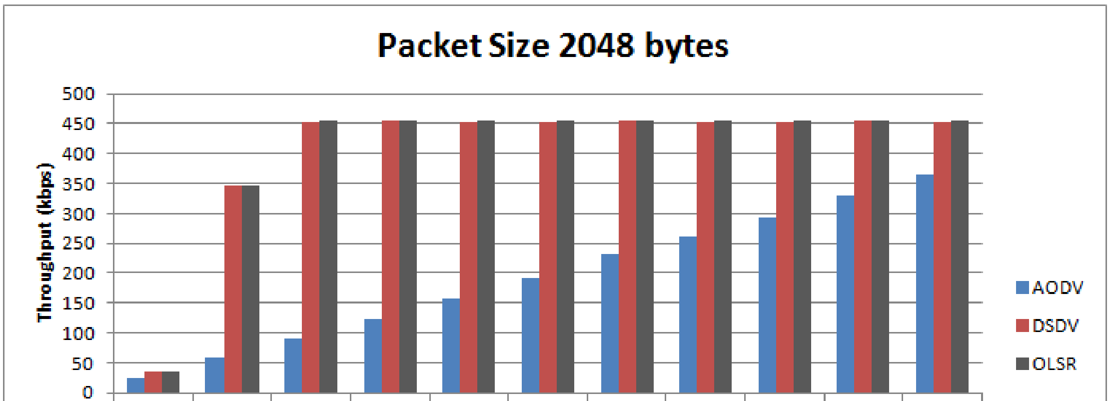


Figure 107 Throughput vs. interval, DSSS Rate 5.5Mbps, Packet Size 1024 bytes.

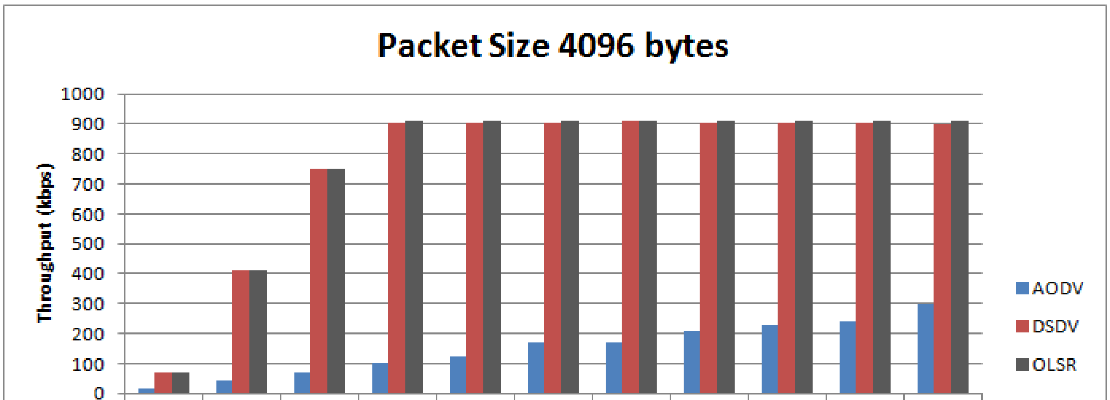
In Figure 107 we can see that AODV has in every case the lowest throughput. OLSR has highest throughput and outperforms DSDV.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	25,4820	59,1920	91,2930	122,418	157,464	193,237	232,648	261,080	291,813	329,498	364,632
DSDV	36,496	344,941	453,744	454,498	453,136	452,586	454,530	453,408	453,560	454,136	452,042
OLSR	36,437	345,640	454,472	454,624	454,592	454,653	454,834	454,621	454,349	454,592	454,592

Figure 108 Throughput vs. interval, DSSS Rate 5.5Mbps, Packet Size 2048 bytes.

In Figure 108 we can see that AODV has in every case the lowest throughput. When the interval is 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45 and 0.5seconds OLSR has highest throughput than DSDV. In all the other cases DSDV achieves highest throughput than OLSR.

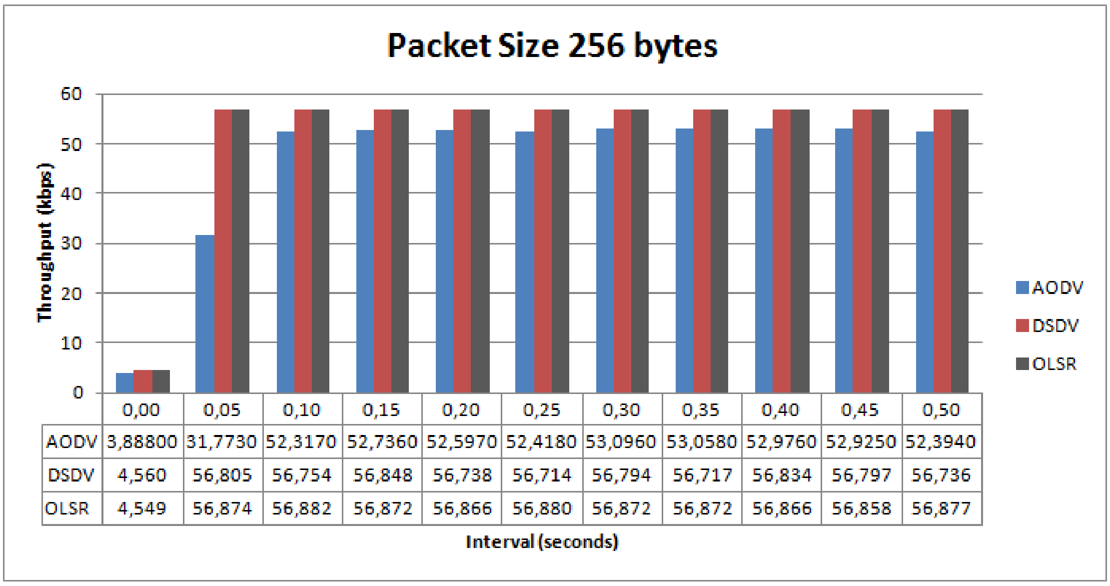


	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	19,7780	44,4130	70,8100	102,061	124,696	170,146	172,210	210,074	227,914	242,784	299,157
DSDV	72,992	411,720	749,714	907,426	905,909	904,269	908,578	905,485	906,637	907,122	901,784
OLSR	72,872	412,205	752,568	908,458	908,701	909,368	908,277	908,698	908,400	908,277	908,034

Figure 109 Throughput vs. interval, DSSS Rate 5.5Mbps, Packet Size 4096 bytes.

In Figure 109 we can see that AODV has in every case the lowest throughput. When the interval is 0.05, 0.1, 0.15, 0.2, 0.25, 0.35, 0.4, 0.45 and 0.5seconds OLSR has highest throughput than DSDV. In all the other cases DSDV achieves highest throughput than OLSR.

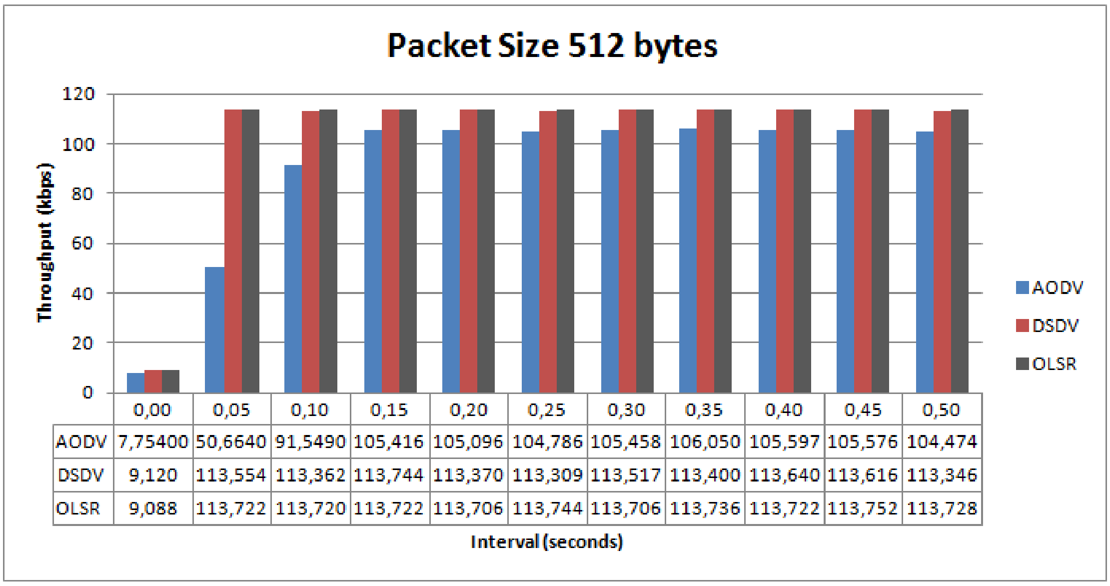
4. DSSS Rate 11Mbps



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	3,88800	31,7730	52,3170	52,7360	52,5970	52,4180	53,0960	53,0580	52,9760	52,9250	52,3940
DSDV	4,560	56,805	56,754	56,848	56,738	56,714	56,794	56,717	56,834	56,797	56,736
OLSR	4,549	56,874	56,882	56,872	56,866	56,880	56,872	56,872	56,866	56,858	56,877

Figure 110 Throughput vs. interval, DSSS Rate 11Mbps, Packet Size 256 bytes.

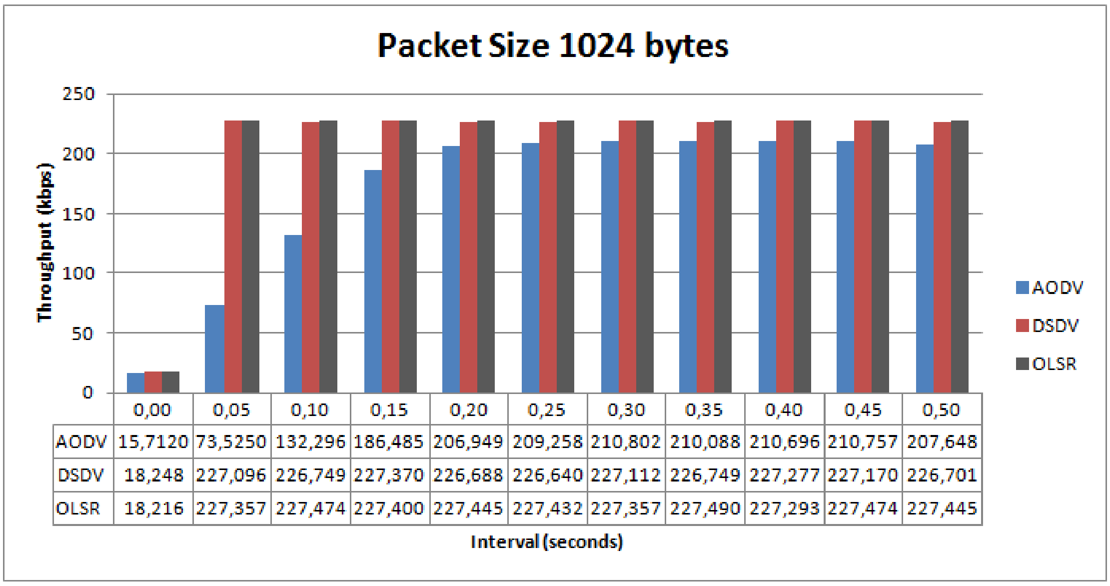
In Figure 110 we can see that AODV has in every case the lowest throughput. When the interval is 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45 and 0.5seconds OLSR has highest throughput than DSDV. In all the other cases DSDV achieves highest throughput than OLSR.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	7,75400	50,6640	91,5490	105,416	105,096	104,786	105,458	106,050	105,597	105,576	104,474
DSDV	9,120	113,554	113,362	113,744	113,370	113,309	113,517	113,400	113,640	113,616	113,346
OLSR	9,088	113,722	113,720	113,722	113,706	113,744	113,706	113,736	113,722	113,752	113,728

Figure 111 Throughput vs. interval, DSSS Rate 11Mbps, Packet Size 512 bytes.

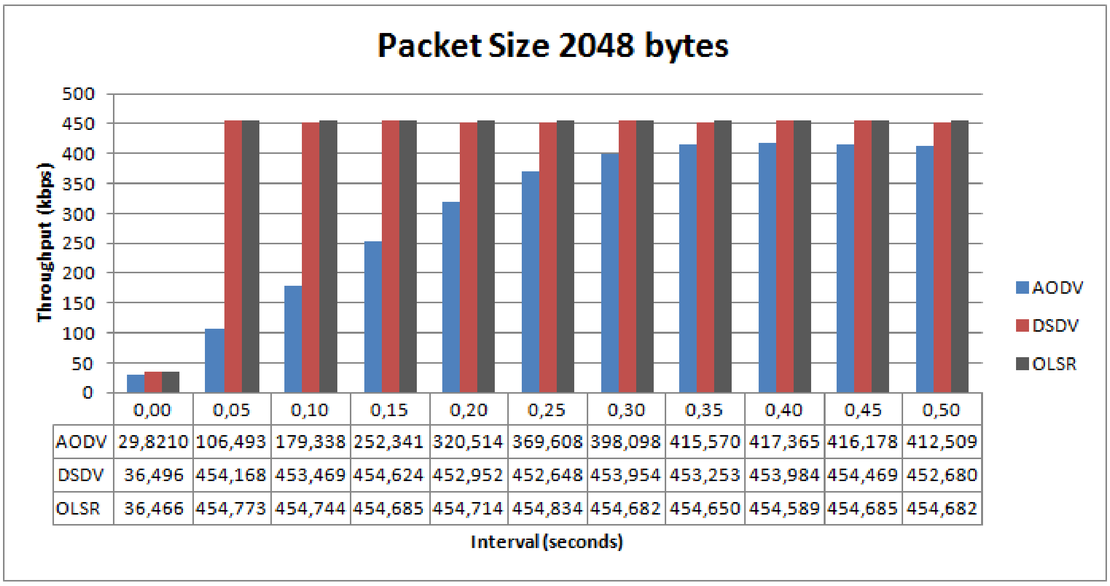
In Figure 111 we can see that AODV has in every case the lowest throughput. When the interval is 0.05, 0.1, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45 and 0.5seconds OLSR has highest throughput than DSDV. In all the other cases DSDV achieves highest throughput than OLSR.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	15,7120	73,5250	132,296	186,485	206,949	209,258	210,802	210,088	210,696	210,757	207,648
DSDV	18,248	227,096	226,749	227,370	226,688	226,640	227,112	226,749	227,277	227,170	226,701
OLSR	18,216	227,357	227,474	227,400	227,445	227,432	227,357	227,490	227,293	227,474	227,445

Figure 112 Throughput vs. interval, DSSS Rate 11Mbps, Packet Size 1024 bytes.

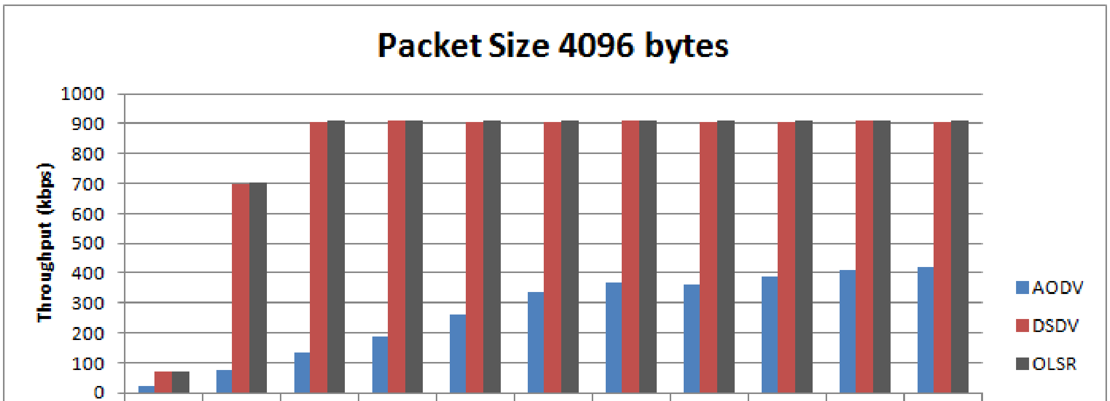
In Figure 112 we can see that AODV has in every case the lowest throughput. When the interval is 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45 and 0.5seconds OLSR has highest throughput than DSDV. In all the other cases DSDV achieves highest throughput than OLSR.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	29,8210	106,493	179,338	252,341	320,514	369,608	398,098	415,570	417,365	416,178	412,509
DSDV	36,496	454,168	453,469	454,624	452,952	452,648	453,954	453,253	453,984	454,469	452,680
OLSR	36,466	454,773	454,744	454,685	454,714	454,834	454,682	454,650	454,589	454,685	454,682

Figure 113 Throughput vs. interval, DSSS Rate 11Mbps, Packet Size 2048 bytes.

In Figure 113 we can see that AODV has in every case the lowest throughput. When the interval is 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45 and 0.5seconds OLSR has highest throughput than DSDV. In all the other cases DSDV achieves highest throughput than OLSR.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	24,8770	74,8773	133,072	185,560	262,930	336,293	368,696	361,293	390,421	408,869	419,122
DSDV	72,992	698,136	907,000	909,002	904,938	904,088	909,066	905,850	907,061	908,882	902,874
OLSR	72,813	700,261	908,760	909,064	908,944	908,944	908,338	908,760	907,850	908,581	908,760

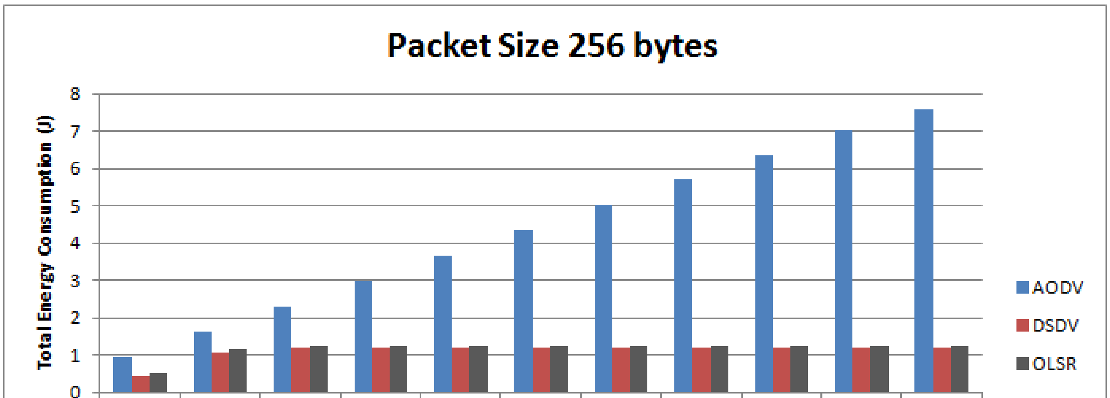
Figure 114 Throughput vs. interval, DSSS Rate 11Mbps, Packet Size 4096 bytes.

In Figure 114 we can see that AODV has in every case the lowest throughput. OLSR has highest throughput and outperforms DSDV.

<u>Total Energy Consumption</u>

In Figures 115-134 we can compare the three routing protocols based on the total energy consumption they achieve. The results are categorized based on the different DSSS Rates.

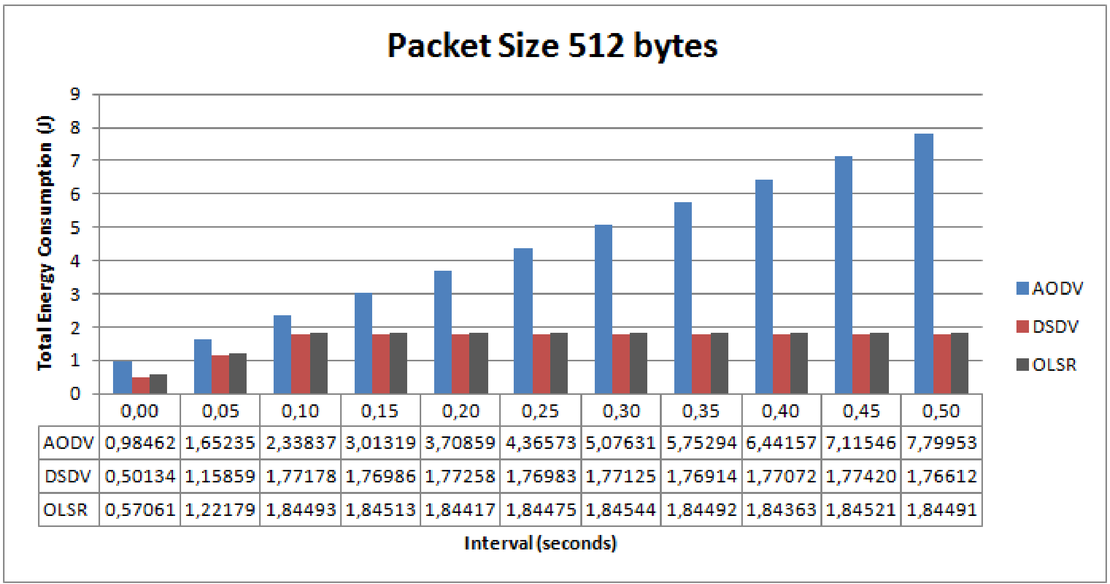
1. DSSS Rate 1Mbps



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	0,95389	1,63711	2,30892	2,98711	3,66364	4,33921	5,01573	5,69212	6,35306	7,02350	7,58186
DSDV	0,45385	1,07140	1,18418	1,18583	1,18280	1,18120	1,18051	1,18392	1,18373	1,18292	1,18327
OLSR	0,52311	1,14168	1,25377	1,25427	1,25394	1,25424	1,25397	1,25363	1,25315	1,25364	1,25408

Figure 115 Total Energy Consumption vs. interval, DSSS Rate 1Mbps, Packet Size 256 bytes.

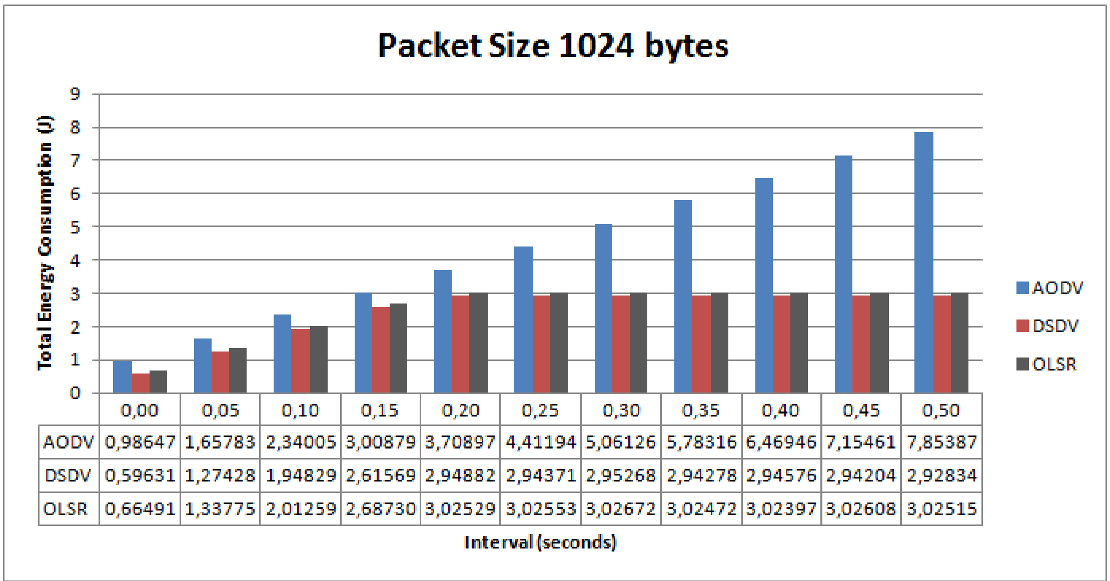
In Figure 115 we can see that AODV has in every case the highest total energy consumption. Also, DSDV achieves lowest total energy consumption than OLSR in all cases.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	0,98462	1,65235	2,33837	3,01319	3,70859	4,36573	5,07631	5,75294	6,44157	7,11546	7,79953
DSDV	0,50134	1,15859	1,77178	1,76986	1,77258	1,76983	1,77125	1,76914	1,77072	1,77420	1,76612
OLSR	0,57061	1,22179	1,84493	1,84513	1,84417	1,84475	1,84544	1,84492	1,84363	1,84521	1,84491

Figure 116 Total Energy Consumption vs. interval, DSSS Rate 1Mbps, Packet Size 512 bytes.

In Figure 116 we can see that AODV has in every case the highest total energy consumption. Also, DSDV achieves lowest total energy consumption than OLSR in all cases.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	0,98647	1,65783	2,34005	3,00879	3,70897	4,41194	5,06126	5,78316	6,46946	7,15461	7,85387
DSDV	0,59631	1,27428	1,94829	2,61569	2,94882	2,94371	2,95268	2,94278	2,94576	2,94204	2,92834
OLSR	0,66491	1,33775	2,01259	2,68730	3,02529	3,02553	3,02672	3,02472	3,02397	3,02608	3,02515

Figure 117 Total Energy Consumption vs. interval, DSSS Rate 1Mbps, Packet Size 1024 bytes.

In Figure 117 we can see that AODV has in every case the highest total energy consumption. Also, DSDV achieves lowest total energy consumption than OLSR in all cases.

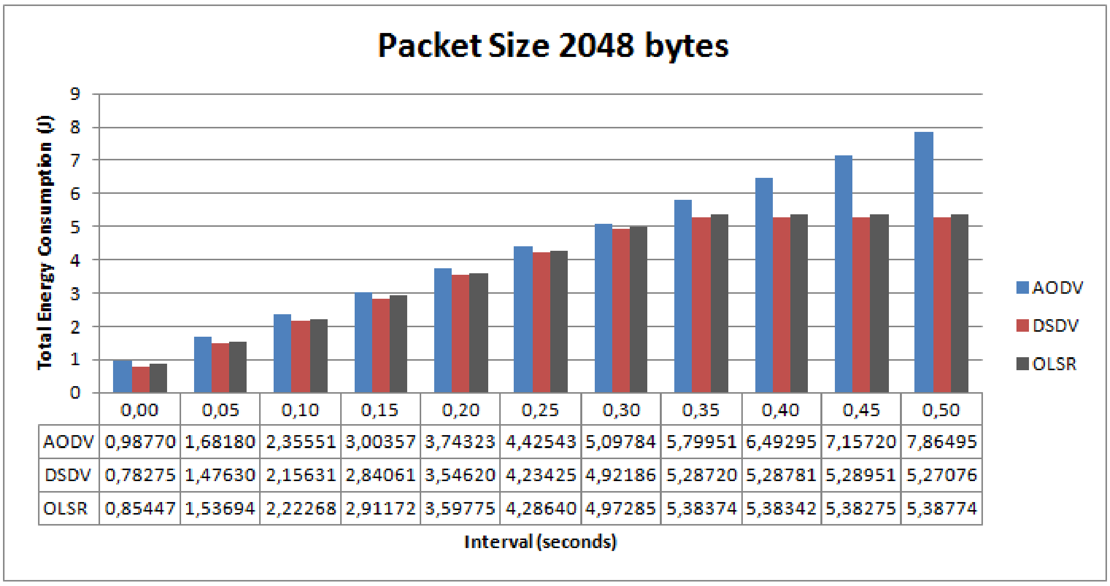


Figure 118 Total Energy Consumption vs. interval, DSSS Rate 1Mbps, Packet Size 2048 bytes.

In Figure 118 we can see that AODV has in every case the highest total energy consumption. Also, DSDV achieves lowest total energy consumption than OLSR in all cases.

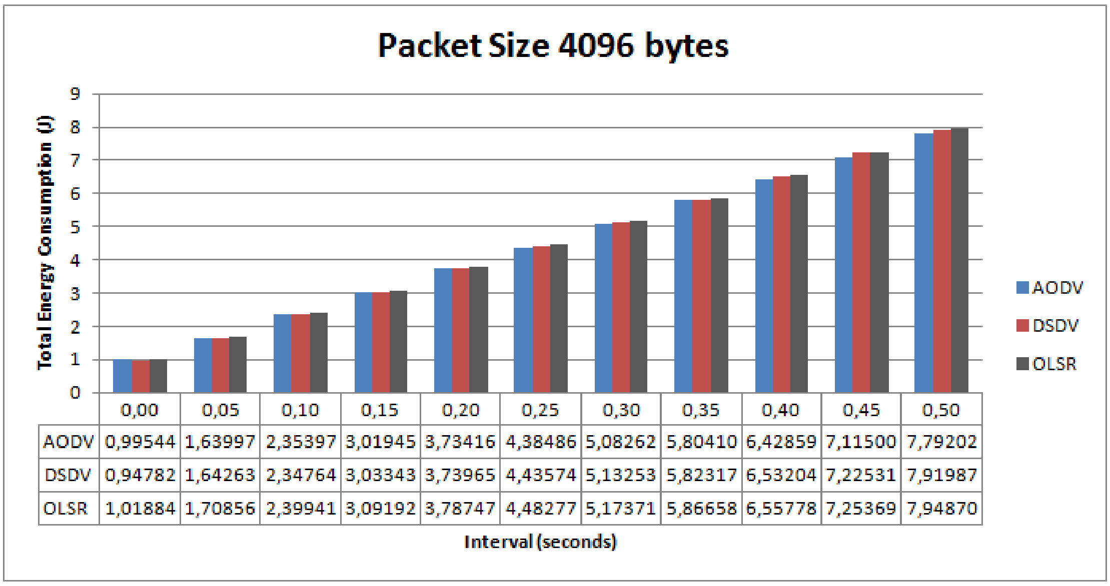
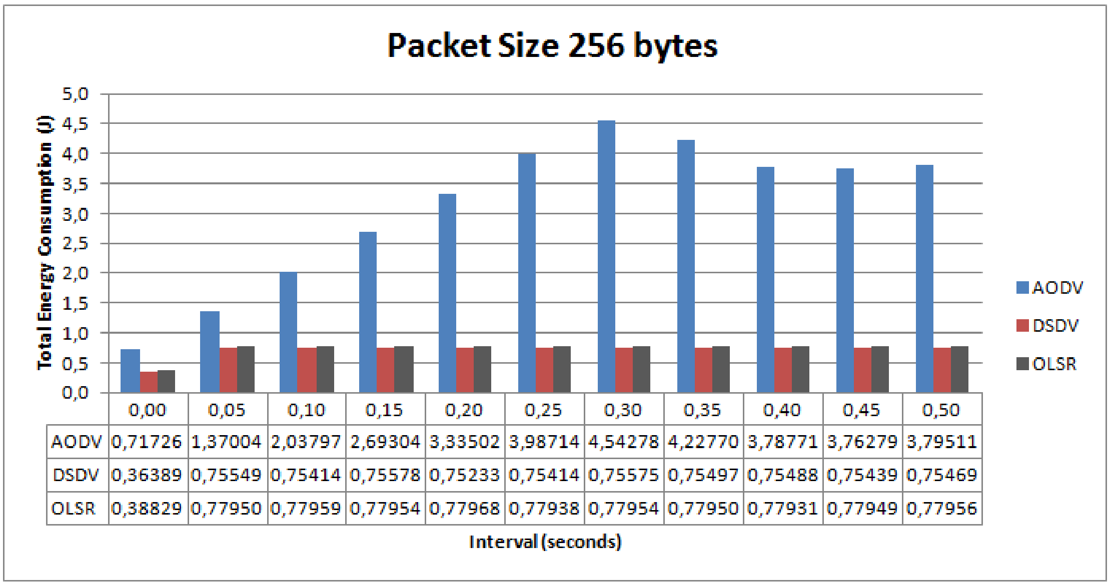


Figure 119 Total Energy Consumption vs. interval, DSSS Rate 1Mbps, Packet Size 4096 bytes.

In Figure 119 we can see that AODV has in every case the highest total energy consumption. Also, DSDV achieves lowest total energy consumption than OLSR in all cases.

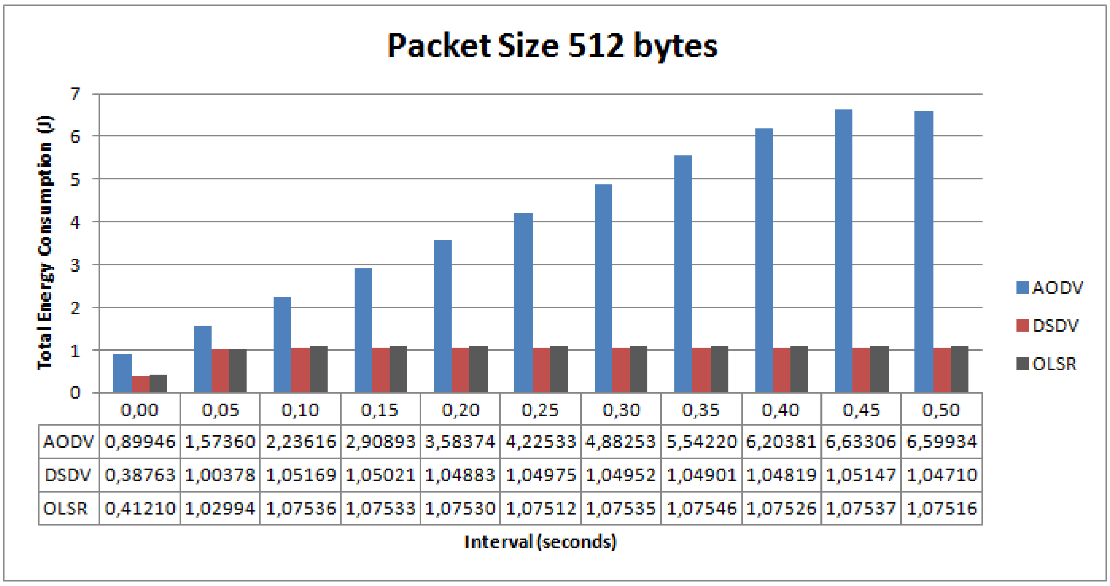
2. DSSS Rate 2Mbps



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	0,71726	1,37004	2,03797	2,69304	3,33502	3,98714	4,54278	4,22770	3,78771	3,76279	3,79511
DSDV	0,36389	0,75549	0,75414	0,75578	0,75233	0,75414	0,75575	0,75497	0,75488	0,75439	0,75469
OLSR	0,38829	0,77950	0,77959	0,77954	0,77968	0,77938	0,77954	0,77950	0,77931	0,77949	0,77956

Figure 120 Total Energy Consumption vs. interval, DSSS Rate 2Mbps, Packet Size 256 bytes.

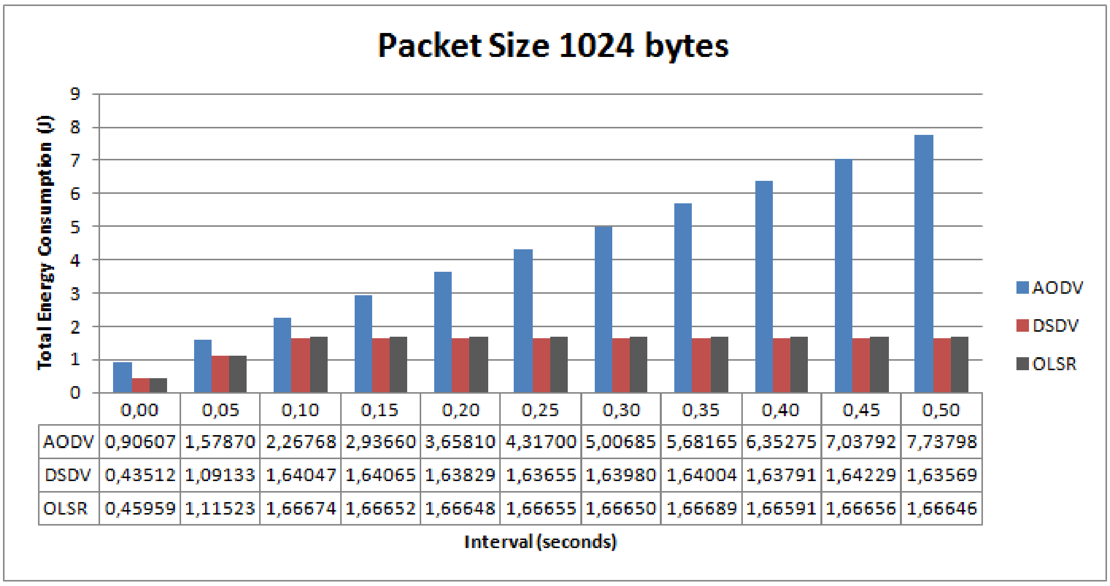
In Figure 120 we can see that AODV has in every case the highest total energy consumption. Also, DSDV achieves lowest total energy consumption than OLSR in all cases.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	0,89946	1,57360	2,23616	2,90893	3,58374	4,22533	4,88253	5,54220	6,20381	6,63306	6,59934
DSDV	0,38763	1,00378	1,05169	1,05021	1,04883	1,04975	1,04952	1,04901	1,04819	1,05147	1,04710
OLSR	0,41210	1,02994	1,07536	1,07533	1,07530	1,07512	1,07535	1,07546	1,07526	1,07537	1,07516

Figure 121 Total Energy Consumption vs. interval, DSSS Rate 2Mbps, Packet Size 512 bytes.

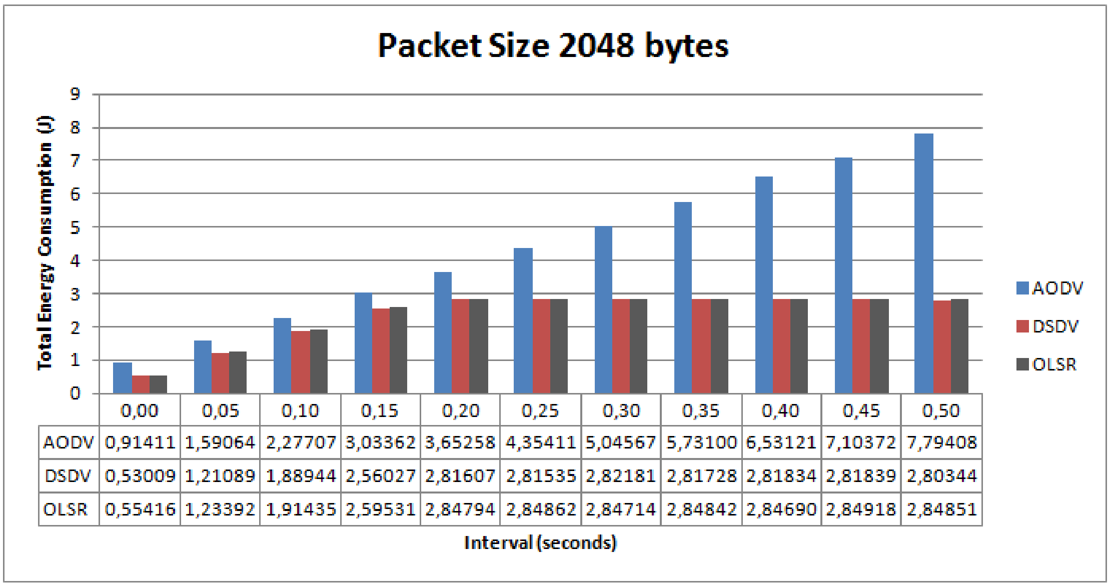
In Figure 121 we can see that AODV has in every case the highest total energy consumption. Also, DSDV achieves lowest total energy consumption than OLSR in all cases.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	0,90607	1,57870	2,26768	2,93660	3,65810	4,31700	5,00685	5,68165	6,35275	7,03792	7,73798
DSDV	0,43512	1,09133	1,64047	1,64065	1,63829	1,63655	1,63980	1,64004	1,63791	1,64229	1,63569
OLSR	0,45959	1,11523	1,66674	1,66652	1,66648	1,66655	1,66650	1,66689	1,66591	1,66656	1,66646

Figure 122 Total Energy Consumption vs. interval, DSSS Rate 2Mbps, Packet Size 1024 bytes.

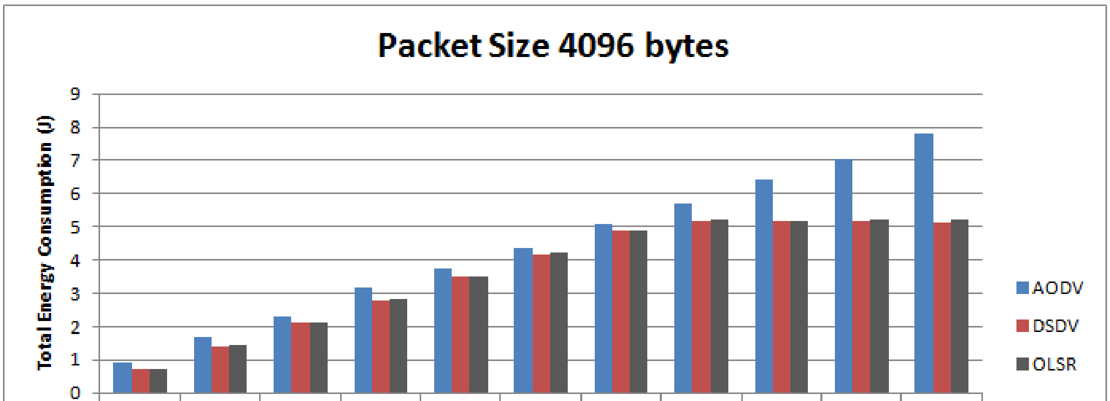
In Figure 122 we can see that AODV has in every case the highest total energy consumption. Also, DSDV achieves lowest total energy consumption than OLSR in all cases.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	0,91411	1,59064	2,27707	3,03362	3,65258	4,35411	5,04567	5,73100	6,53121	7,10372	7,79408
DSDV	0,53009	1,21089	1,88944	2,56027	2,81607	2,81535	2,82181	2,81728	2,81834	2,81839	2,80344
OLSR	0,55416	1,23392	1,91435	2,59531	2,84794	2,84862	2,84714	2,84842	2,84690	2,84918	2,84851

Figure 123 Total Energy Consumption vs. interval, DSSS Rate 2Mbps, Packet Size 2048 bytes.

In Figure 123 we can see that AODV has in every case the highest total energy consumption. Also, DSDV achieves lowest total energy consumption than OLSR in all cases.

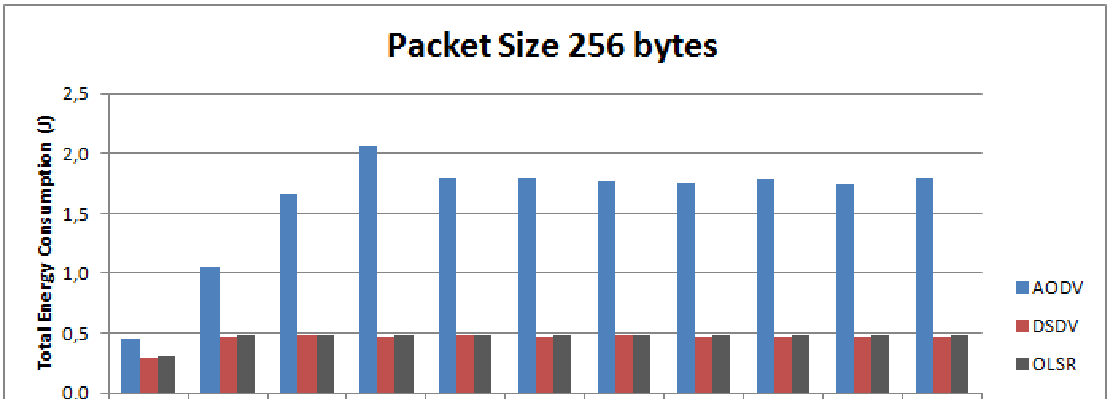


	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	0,94138	1,67234	2,31645	3,15546	3,72148	4,35429	5,07218	5,68769	6,40306	7,05028	7,79563
DSDV	0,71432	1,41250	2,09967	2,78720	3,49552	4,18953	4,88013	5,16885	5,17165	5,17243	5,15180
OLSR	0,74387	1,43438	2,12757	2,82358	3,51849	4,20821	4,90285	5,20677	5,20174	5,20207	5,20386

Figure 124 Total Energy Consumption vs. interval, DSSS Rate 2Mbps, Packet Size 4096 bytes.

In Figure 124 we can see that AODV has in every case the highest total energy consumption. Also, DSDV achieves lowest total energy consumption than OLSR in all cases.

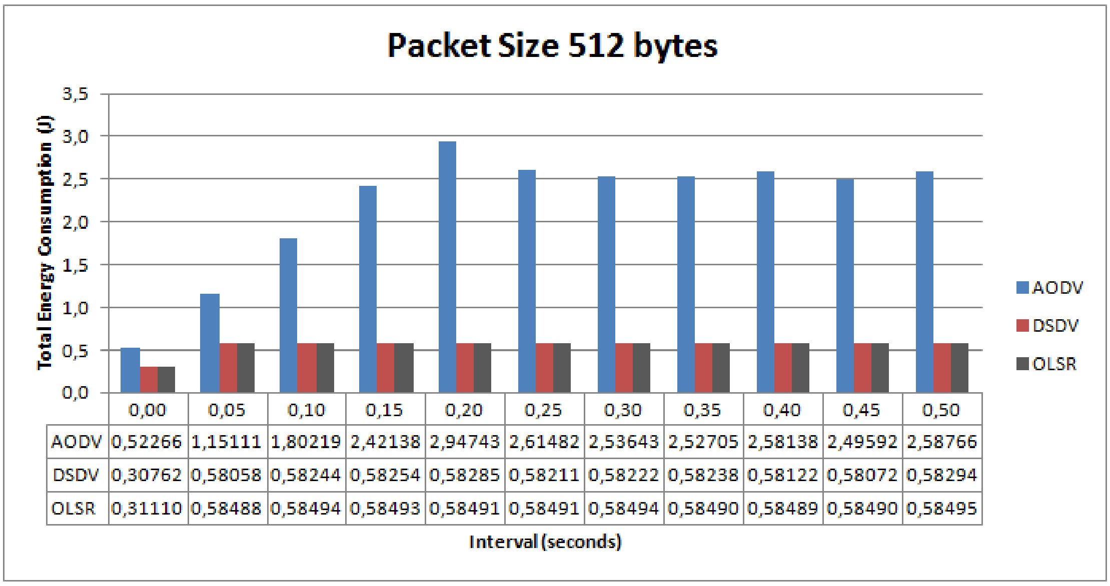
3. DSSS Rate 5.5Mbps



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	0,44967	1,05568	1,66100	2,06539	1,79635	1,79221	1,76802	1,75080	1,78364	1,74637	1,80051
DSDV	0,29900	0,47416	0,47530	0,47484	0,47603	0,47460	0,47532	0,47353	0,47371	0,47309	0,47478
OLSR	0,30249	0,47744	0,47743	0,47746	0,47741	0,47739	0,47741	0,47740	0,47739	0,47744	0,47742

Figure 125 Total Energy Consumption vs. interval, DSSS Rate 5.5Mbps, Packet Size 256 bytes.

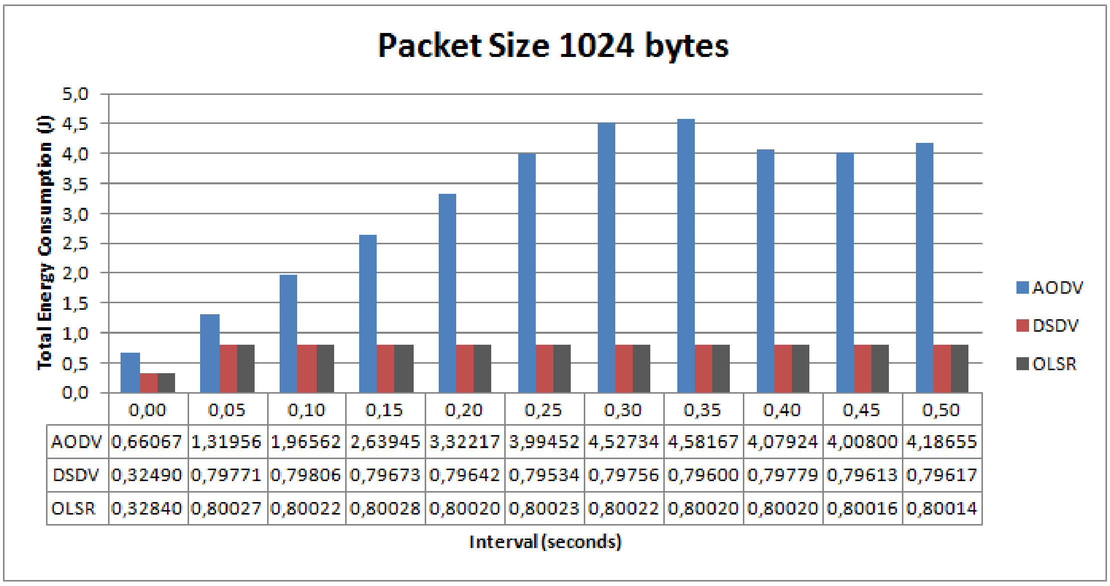
In Figure 125 we can see that AODV has in every case the highest total energy consumption. Also, DSDV achieves lowest total energy consumption than OLSR in all cases.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	0,52266	1,15111	1,80219	2,42138	2,94743	2,61482	2,53643	2,52705	2,58138	2,49592	2,58766
DSDV	0,30762	0,58058	0,58244	0,58254	0,58285	0,58211	0,58222	0,58238	0,58122	0,58072	0,58294
OLSR	0,31110	0,58488	0,58494	0,58493	0,58491	0,58491	0,58494	0,58490	0,58489	0,58490	0,58495

Figure 126 Total Energy Consumption vs. interval, DSSS Rate 5.5Mbps, Packet Size 512 bytes.

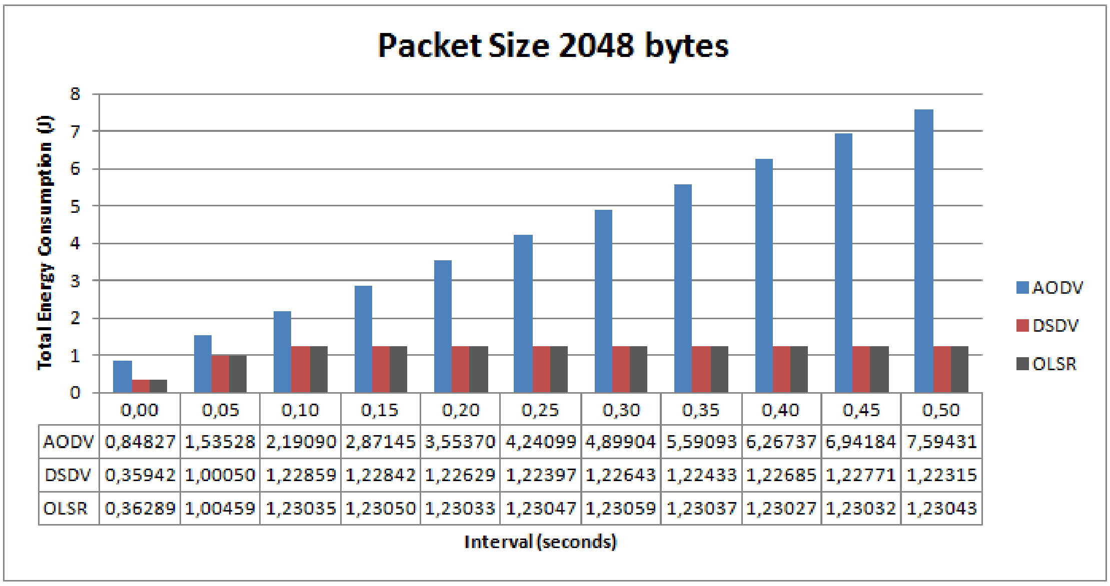
In Figure 126 we can see that AODV has in every case the highest total energy consumption. Also, DSDV achieves lowest total energy consumption than OLSR in all cases.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	0,66067	1,31956	1,96562	2,63945	3,32217	3,99452	4,52734	4,58167	4,07924	4,00800	4,18655
DSDV	0,32490	0,79771	0,79806	0,79673	0,79642	0,79534	0,79756	0,79600	0,79779	0,79613	0,79617
OLSR	0,32840	0,80027	0,80022	0,80028	0,80020	0,80023	0,80022	0,80020	0,80020	0,80016	0,80014

Figure 127 Total Energy Consumption vs. interval, DSSS Rate 5.5Mbps, Packet Size 1024 bytes.

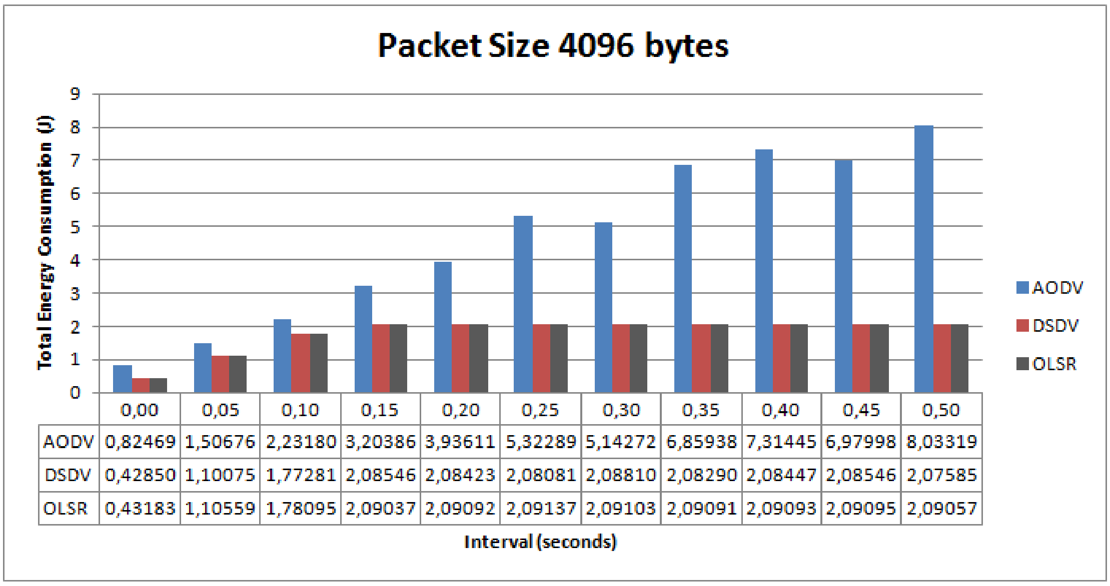
In Figure 127 we can see that AODV has in every case the highest total energy consumption. Also, DSDV achieves lowest total energy consumption than OLSR in all cases.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	0,84827	1,53528	2,19090	2,87145	3,55370	4,24099	4,89904	5,59093	6,26737	6,94184	7,59431
DSDV	0,35942	1,00050	1,22859	1,22842	1,22629	1,22397	1,22643	1,22433	1,22685	1,22771	1,22315
OLSR	0,36289	1,00459	1,23035	1,23050	1,23033	1,23047	1,23059	1,23037	1,23027	1,23032	1,23043

Figure 128 Total Energy Consumption vs. interval, DSSS Rate 5.5Mbps, Packet Size 2048 bytes.

In Figure 128 we can see that AODV has in every case the highest total energy consumption. Also, DSDV achieves lowest total energy consumption than OLSR in all cases.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	0,82469	1,50676	2,23180	3,20386	3,93611	5,32289	5,14272	6,85938	7,31445	6,97998	8,03319
DSDV	0,42850	1,10075	1,77281	2,08546	2,08423	2,08081	2,08810	2,08290	2,08447	2,08546	2,07585
OLSR	0,43183	1,10559	1,78095	2,09037	2,09092	2,09137	2,09103	2,09091	2,09093	2,09095	2,09057

Figure 129 Total Energy Consumption vs. interval, DSSS Rate 5.5Mbps, Packet Size 4096 bytes.

In Figure 129 we can see that AODV has in every case the highest total energy consumption. Also, DSDV achieves lowest total energy consumption than OLSR in all cases.

4. DSSS Rate 11 Mbps

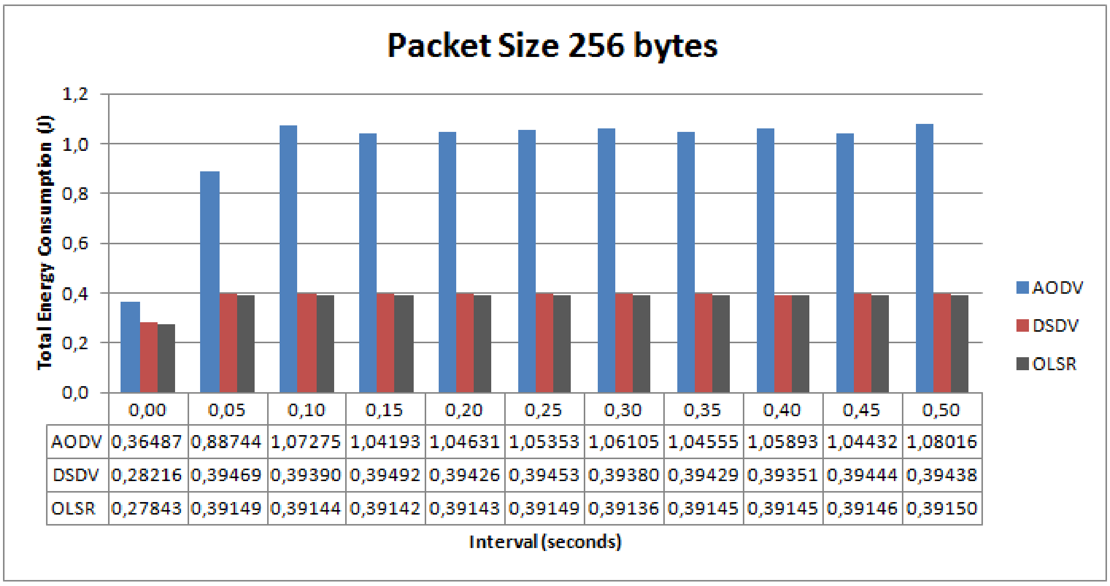


Figure 130 **Total Energy Consumption vs. interval, DSSS Rate 11Mbps, Packet Size 256 bytes.**

In Figure 130 we can see that AODV has in every case the highest total energy consumption. Also, OLSR achieves lowest total energy consumption than DSDV in all cases.

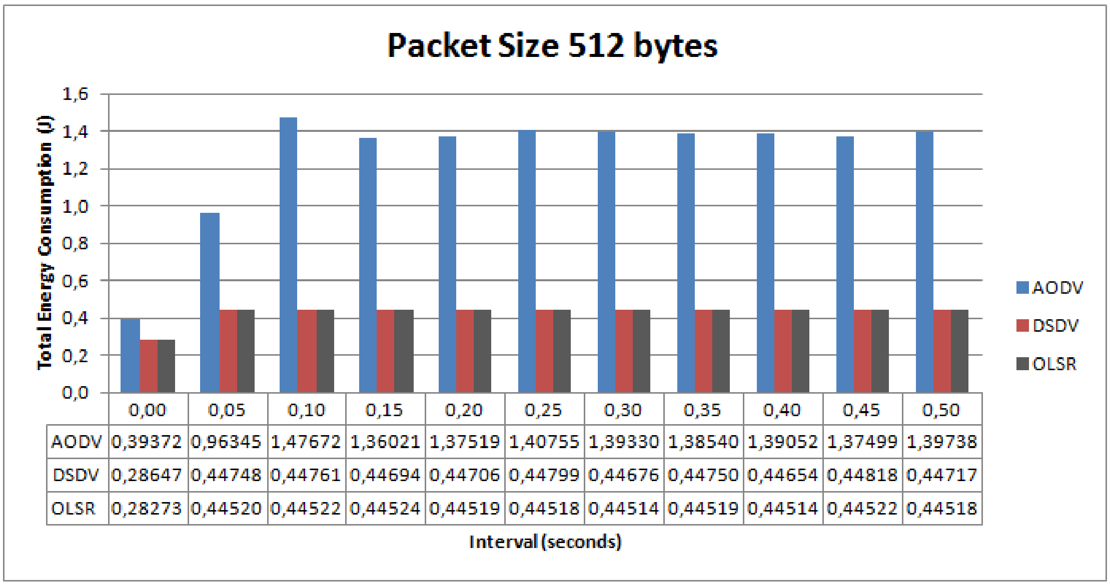
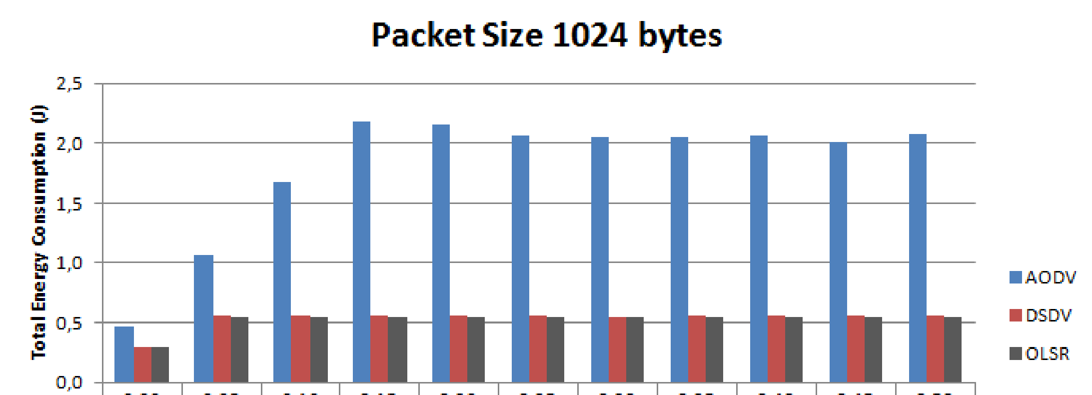


Figure 131 **Total Energy Consumption vs. interval, DSSS Rate 11Mbps, Packet Size 512 bytes.**

In Figure 131 we can see that AODV has in every case the highest total energy consumption. Also, OLSR achieves lowest total energy consumption than DSDV in all cases.

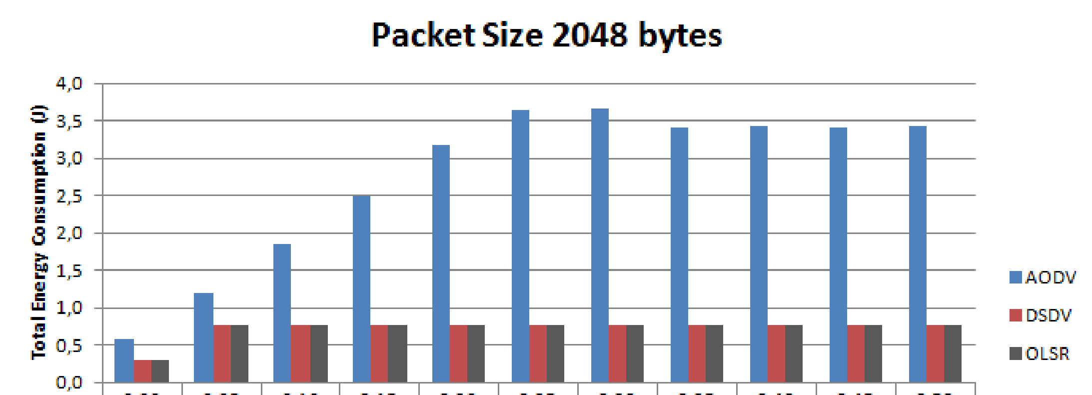


	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	0,46800	1,06682	1,67533	2,18793	2,15019	2,05831	2,04683	2,05396	2,06378	2,01634	2,07018
DSDV	0,29512	0,55535	0,55583	0,55561	0,55556	0,55549	0,55440	0,55522	0,55587	0,55579	0,55509
OLSR	0,29145	0,55295	0,55305	0,55295	0,55301	0,55299	0,55292	0,55305	0,55288	0,55297	0,55293

Interval (seconds)

Figure 132 Total Energy Consumption vs. interval, DSSS Rate 11Mbps, Packet Size 1024 bytes.

In Figure 132 we can see that AODV has in every case the highest total energy consumption. Also, OLSR achieves lowest total energy consumption than DSDV in all cases.

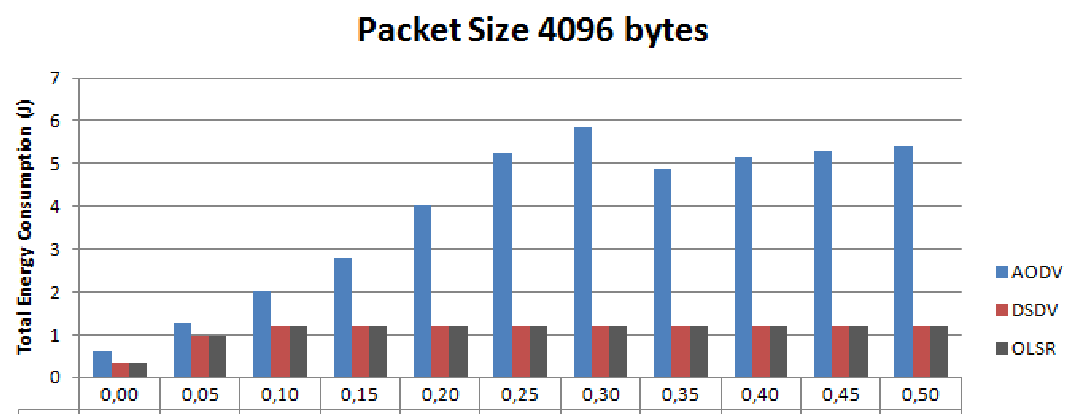


	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	0,57885	1,20615	1,85156	2,48745	3,16594	3,63304	3,65964	3,39724	3,42676	3,40399	3,41914
DSDV	0,31237	0,76987	0,77010	0,77053	0,76934	0,76980	0,76986	0,76909	0,77009	0,77057	0,76945
OLSR	0,30862	0,76804	0,76805	0,76799	0,76805	0,76802	0,76791	0,76783	0,76800	0,76798	0,76798

Interval (seconds)

Figure 133 Total Energy Consumption vs. interval, DSSS Rate 11Mbps, Packet Size 2048 bytes.

In Figure 133 we can see that AODV has in every case the highest total energy consumption. Also, OLSR achieves lowest total energy consumption than DSDV in all cases.



	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
AODV	0,61018	1,28270	2,01099	2,80465	4,03695	5,26564	5,83954	4,89271	5,15107	5,29901	5,39715
DSDV	0,34692	0,98533	1,19959	1,20156	1,19837	1,19713	1,20132	1,19938	1,19944	1,20060	1,19697
OLSR	0,34310	0,98478	1,19816	1,19839	1,19842	1,19837	1,19826	1,19825	1,19794	1,19836	1,19809

Figure 134 Total Energy Consumption vs. interval, DSSS Rate 11Mbps, Packet Size 4096 bytes.

In Figure 134 we can see that AODV has in every case the highest total energy consumption. When the interval is 0, 0.05, 0.1, 0.15, 0.3, 0.35, 0.4 and 0.45 seconds OLSR has lowest total energy consumption than DSDV. In all the other cases DSDV achieves lowest total energy consumption than OLSR.

Conclusions

1. Packet Delivery Ratio (PDR)

In the tables that follow (Tables 12-15) we can see which routing protocol has in every case the best performance considering the PDR and categorized based on the different DSSS Rates.

Table 13 Performance Comparison (DSSS Rate 1Mbps)

		Packet Size				
		256	512	1024	2048	4096
interval	0s	DSDV	DSDV	DSDV	OLSR	OLSR
	0,05s	DSDV	DSDV	DSDV	DSDV	OLSR
	0,1s	OLSR	OLSR	DSDV	DSDV	DSDV
	0,15s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,2s	OLSR	OLSR	OLSR	DSDV	DSDV
	0,25s	OLSR	OLSR	OLSR	DSDV	DSDV
	0,3s	OLSR	OLSR	OLSR	DSDV	DSDV
	0,35s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,4s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,45s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,5s	OLSR	OLSR	OLSR	OLSR	DSDV

Table 14 Performance Comparison (DSSS Rate 2Mbps)

		Packet Size				
		256	512	1024	2048	4096
interval	0s	DSDV	DSDV	DSDV	DSDV	OLSR
	0,05s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,1s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,15s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,2s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,25s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,3s	OLSR	OLSR	OLSR	DSDV	OLSR
	0,35s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,4s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,45s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,5s	OLSR	OLSR	OLSR	OLSR	OLSR

Table 15 Performance Comparison (DSSS Rate 5.5Mbps)

		Packet Size				
		256	512	1024	2048	4096
interval	0s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,05s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,1s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,15s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,2s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,25s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,3s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,35s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,4s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,45s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,5s	OLSR	OLSR	OLSR	OLSR	OLSR

Table 16 Performance Comparison (DSSS Rate 11Mbps)

		Packet Size				
		256	512	1024	2048	4096
interval	0s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,05s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,1s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,15s	OLSR	DSDV	OLSR	OLSR	OLSR
	0,2s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,25s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,3s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,35s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,4s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,45s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,5s	OLSR	OLSR	OLSR	OLSR	OLSR

AODV has never the best PDR performance. The other two protocols always outperform AODV. OLSR has in most of the cases the best performance as it has 84.1% the best PDR. DSDV has only 15.9% PDR and it comes second. Therefore, the best choice for this scenario based on the PDR is OLSR.

2. Average Delay

In the tables that follow (Tables 16-19) we can see which routing protocol has in every case the best performance considering the average delay and categorized based on the different DSSS Rates.

Table 17 Performance Comparison (DSSS Rate 1Mbps)

		Packet Size				
		256	512	1024	2048	4096
interval	0s	DSDV	DSDV	DSDV	DSDV	AODV
	0,05s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,1s	OLSR	DSDV	DSDV	DSDV	DSDV
	0,15s	OLSR	OLSR	DSDV	DSDV	DSDV
	0,2s	OLSR	OLSR	OLSR	DSDV	DSDV
	0,25s	OLSR	OLSR	OLSR	DSDV	DSDV
	0,3s	OLSR	OLSR	OLSR	DSDV	DSDV
	0,35s	OLSR	OLSR	OLSR	DSDV	DSDV
	0,4s	OLSR	DSDV	DSDV	OLSR	DSDV
	0,45s	OLSR	OLSR	DSDV	OLSR	DSDV
	0,5s	DSDV	DSDV	DSDV	DSDV	DSDV

Table 18 Performance Comparison (DSSS Rate 2Mps)

		Packet Size				
		256	512	1024	2048	4096
interval	0s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,05s	OLSR	DSDV	DSDV	DSDV	DSDV
	0,1s	OLSR	OLSR	OLSR	DSDV	DSDV
	0,15s	OLSR	OLSR	OLSR	DSDV	DSDV
	0,2s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,25s	DSDV	DSDV	DSDV	OLSR	DSDV
	0,3s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,35s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,4s	OLSR	DSDV	DSDV	DSDV	OLSR
	0,45s	OLSR	OLSR	OLSR	DSDV	OLSR
	0,5s	DSDV	DSDV	DSDV	DSDV	OLSR

Table 19 Performance Comparison (DSSS Rate 5.5Mbps)

		Packet Size				
		256	512	1024	2048	4096
interval	0s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,05s	OLSR	OLSR	OLSR	DSDV	DSDV
	0,1s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,15s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,2s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,25s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,3s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,35s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,4s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,45s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,5s	OLSR	OLSR	OLSR	OLSR	DSDV

Table 20 Performance Comparison (DSSS Rate 11Mbps)

		Packet Size				
		256	512	1024	2048	4096
interval	0s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,05s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,1s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,15s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,2s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,25s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,3s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,35s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,4s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,45s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,5s	OLSR	OLSR	OLSR	OLSR	OLSR

In only one case AODV has the lowest delay. OLSR and DSDV have quite the same performance, with OLSR being slightly better in most cases (59.9%). DSDV comes second with 40%. For lower DSSS Rates, DSDV has better results. As the DSSS Rates increases, OLSR outperforms DSDV.

3. Throughput

In the tables that follow (Tables 20-23) we can see which routing protocol has in every case the best performance considering the throughput and categorized based on the different DSSS Rates.

Table 21 Performance Comparison (DSSS Rate 1Mbps)

		Packet Size				
		256	512	1024	2048	4096
interval	0s	DSDV	DSDV	DSDV	OLSR	OLSR
	0,05s	DSDV	DSDV	DSDV	DSDV	OLSR
	0,1s	OLSR	OLSR	DSDV	DSDV	DSDV
	0,15s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,2s	OLSR	OLSR	OLSR	DSDV	DSDV
	0,25s	OLSR	OLSR	OLSR	DSDV	DSDV
	0,3s	OLSR	OLSR	OLSR	DSDV	DSDV
	0,35s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,4s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,45s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,5s	OLSR	OLSR	OLSR	OLSR	DSDV

Table 22 Performance Comparison (DSSS Rate 2Mbps)

		Packet Size				
		256	512	1024	2048	4096
interval	0s	DSDV	DSDV	DSDV	DSDV	OLSR
	0,05s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,1s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,15s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,2s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,25s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,3s	OLSR	OLSR	OLSR	DSDV	OLSR
	0,35s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,4s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,45s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,5s	OLSR	OLSR	OLSR	OLSR	OLSR

Table 23 Performance Comparison (DSSS Rate 5.5Mbps)

		Packet Size				
		256	512	1024	2048	4096
interval	0s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,05s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,1s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,15s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,2s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,25s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,3s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,35s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,4s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,45s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,5s	OLSR	OLSR	OLSR	OLSR	OLSR

Table 24 Performance Comparison (DSSS Rate 11Mbps)

		Packet Size				
		256	512	1024	2048	4096
interval	0s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,05s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,1s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,15s	OLSR	DSDV	OLSR	OLSR	OLSR
	0,2s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,25s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,3s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,35s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,4s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,45s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,5s	OLSR	OLSR	OLSR	OLSR	OLSR

AODV has never the highest throughput. The other two protocols always outperform AODV. OLSR has in most of the cases the best performance as it has 83.2% the highest throughput. DSDV has only at 16.8% of the cases the highest throughput. Therefore, the best choice for this scenario based on the PDR is OLSR.

4. Total Energy Consumption

In the tables that follow (Tables 24-27) we can see which routing protocol has in every case the best performance considering the total energy consumption and categorized based on the different DSSS Rates.

Table 25 Performance Comparison (DSSS Rate 1Mbps)

		Packet Size				
		256	512	1024	2048	4096
interval	0s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,05s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,1s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,15s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,2s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,25s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,3s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,35s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,4s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,45s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,5s	DSDV	DSDV	DSDV	DSDV	DSDV

Table 26 Performance Comparison (DSSS Rate 2Mbps)

		Packet Size				
		256	512	1024	2048	4096
interval	0s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,05s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,1s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,15s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,2s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,25s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,3s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,35s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,4s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,45s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,5s	DSDV	DSDV	DSDV	DSDV	DSDV

Table 27 Performance Comparison (DSSS Rate 5.5Mbps)

		Packet Size				
		256	512	1024	2048	4096
interval	0s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,05s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,1s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,15s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,2s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,25s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,3s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,35s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,4s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,45s	DSDV	DSDV	DSDV	DSDV	DSDV
	0,5s	DSDV	DSDV	DSDV	DSDV	DSDV

Table 28 Performance Comparison (DSSS Rate 11Mbps)

		Packet Size				
		256	512	1024	2048	4096
interval	0s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,05s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,1s	DSDV	OLSR	OLSR	OLSR	OLSR
	0,15s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,2s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,25s	OLSR	OLSR	OLSR	OLSR	DSDV
	0,3s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,35s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,4s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,45s	OLSR	OLSR	OLSR	OLSR	OLSR
	0,5s	OLSR	OLSR	OLSR	OLSR	DSDV

AODV has always the highest energy consumption. DSDV one the other hand has the lowest total energy consumption when the DSSS Rate is 1, 2 or 5.5Mbps. When DSSS Rate is 11Mbps OLSR has the lowest total energy consumption with few exceptions. Therefore, the best choice for this scenario from the scope of energy consumption should be based on the DSSS Rate that will be chosen.

4.3.2 Rescue Operation Scenario

The simulation results for the Rescue Operation Scenario are presented on the four sections that follow. In each section we investigate the behavior of each protocol and we also compare them based on different values of number of nodes and speed. In the end of each section we compare the three routing protocols.

4.3.2.1 Packet Delivery Ratio

In this section we investigate the communication reliability for different group sizes and speed. We measure the PDR. PDR is the percentage from the send messages that was actually delivered. It represents how reliable the communication is. The higher the PDR, the better the communication reliability is. Also, the different group sizes determine how heavy the traffic is because it defines how many the senders are. We use ten different group sizes (10, 20, 30, 40, 50, 60, 70, 80, 90, 100 nodes) and eleven different speed values (5m/s – 55m/s with 5m/s step). The highest the number of nodes, the heavier the traffic is. We will see how the different speed values affect the communication.

- AODV

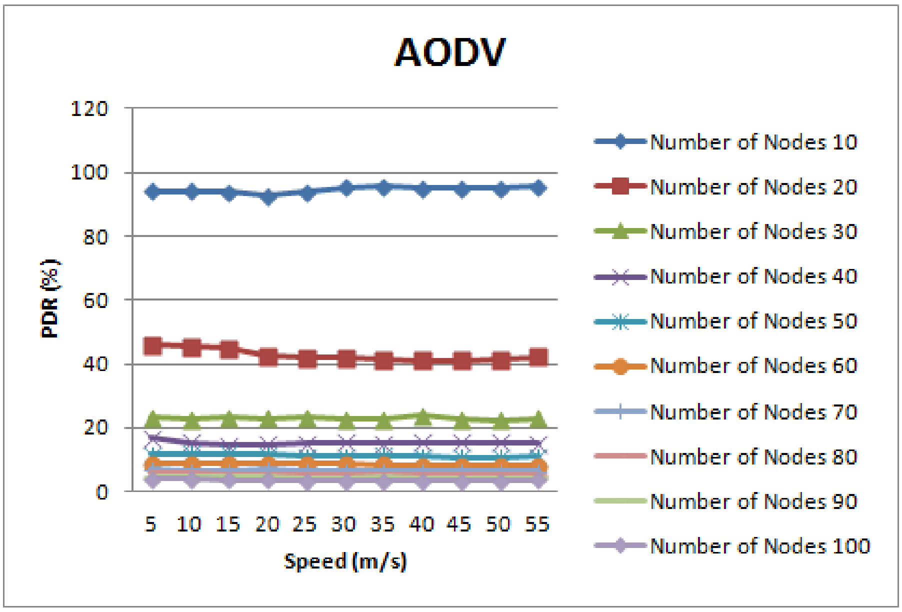


Figure 135 PDR vs. Speed, AODV Routing Protocol

As we can see in Figure 135 the different speed values does not affect the PDR. Only the number of nodes defines the different levels of PDR. More specifically, when the number of nodes is 10 we have in almost every case 94% - 95% PDR, but when the number of nodes doubles to 20 the PDR drops to 41% - 46%, a decrease of almost 50%. As the number of nodes is being decreased, the PDR becomes even lower until it reaches 4% when the number of nodes is 100. Generally we can say that as the number of nodes doubles, the PDR is halved.

- DSDV

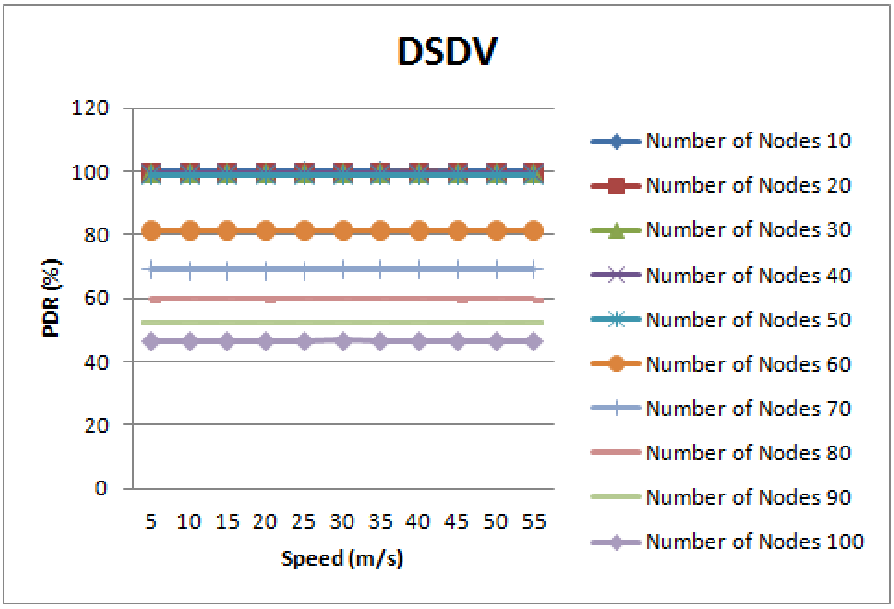


Figure 136 PDR vs. Speed, DSDV Routing Protocol

In contrast with AODV, DSDV has better PDR results. As we can see in Figure 136 DSDV has almost 100% PDR when the number of nodes is 50 or lower. As the number of nodes is increased, the PDR is decreased but in more reasonable levels than AODV. When the number of nodes is 60, the PDR is about 81%, for 70 nodes the PDR is almost 69%, when the number of nodes is 80 the PDR is 59%, for 90 nodes we have PDR 52% and finally for 100 nodes we have PDR 46% which means that almost half of the messages are being transmitted.

- OLSR

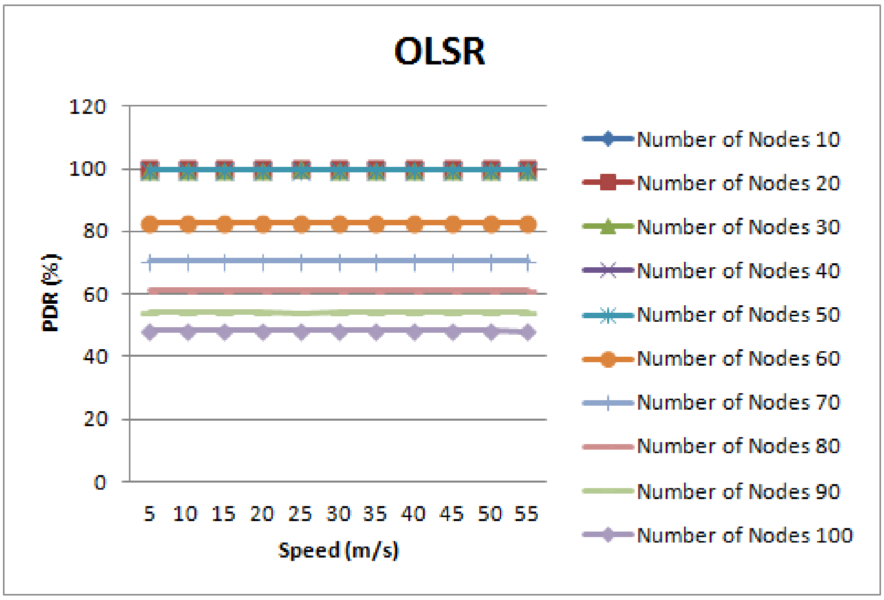


Figure 137 PDR vs. Speed, OLSR Routing Protocol

As we can observe in Figure 137 OLSR, like DSDV, has better PDR results. OLSR has almost 100% PDR when the number of nodes is 50 or lower. As the number of nodes is increased, the PDR is decreased but in more reasonable levels than AODV. When the number of nodes is 60, the PDR is about 83%, for 70 nodes the PDR is almost 70%, when the number of nodes is 80 the PDR is 61%, for 90 nodes we have PDR 54% and finally for 100 nodes we have PDR 48% which means that almost half of the messages are being transmitted.

4.3.2.2 Delay

In this section we investigate the average delay time. The delay affects the quality of the communication and so it is an important factor for the evaluation of the protocols.

- AODV

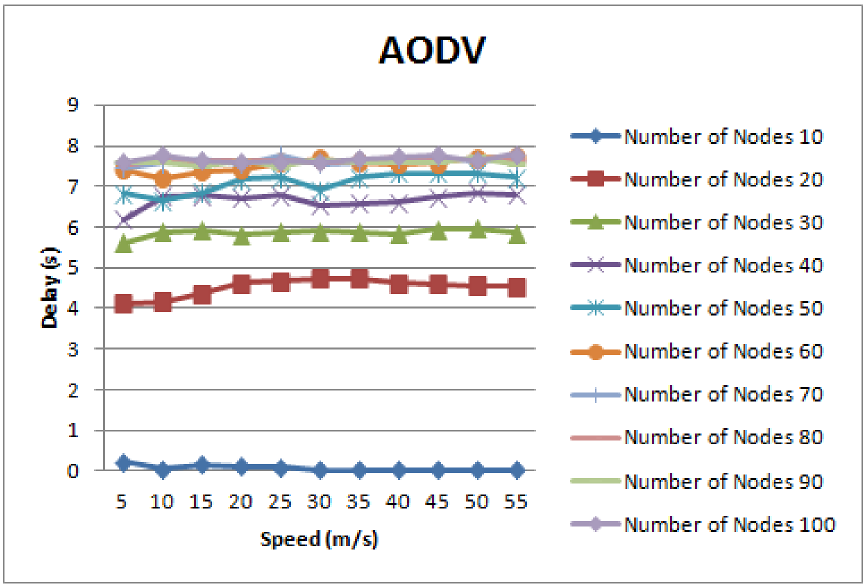


Figure 138 Delay vs. Speed, AODV Routing Protocol

As we can see in Figure 138 the speed does not affect the average delay either. AODV has average delay time almost 0 seconds only when the number of nodes is 10. For 20 nodes and more the delay is from 4.1 to 7.7 seconds.

- DSDV

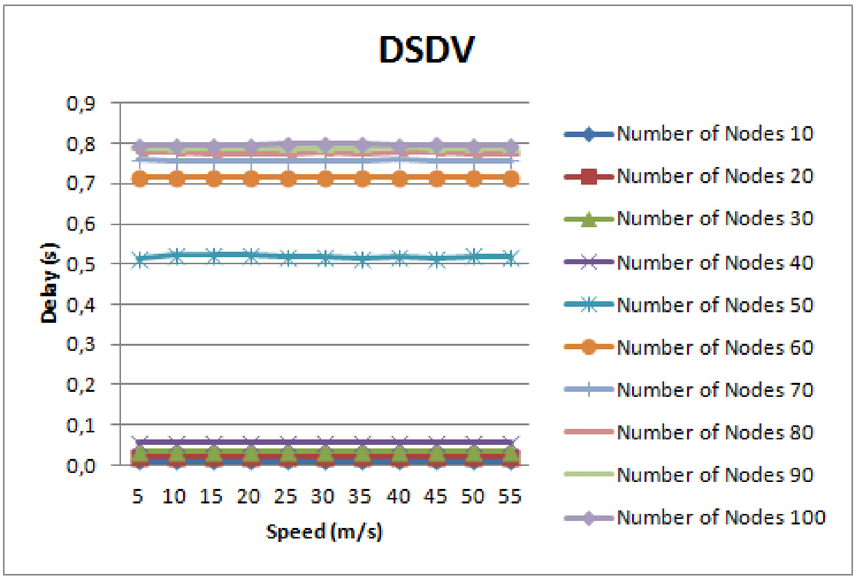


Figure 139 Delay vs. Speed, DSDV Routing Protocol

As we can observe in Figure 139 DSDV has better results than AODV. For all the group sizes, the average delay is below 1 second. If the number of nodes is 40 or lower the delay is below 0.05 seconds and if the nodes are 50 or more the delay is 0.5 to 0.8 seconds.

- OLSR

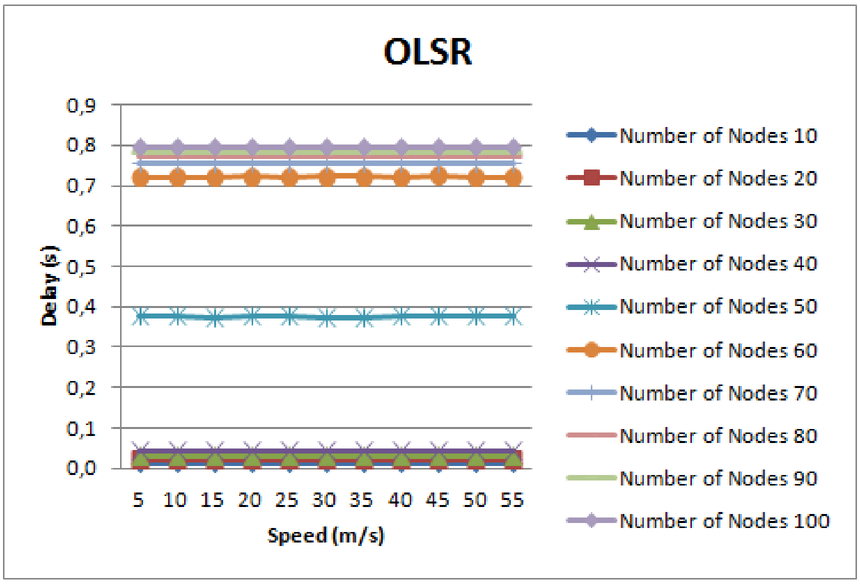


Figure 140 Delay vs. Speed, OLSR Routing Protocol

As we can see in Figure 140, OLSR has almost the same results with DSDV. For all the group sizes, the average delay is below 1 second. If the number of nodes is 40 or lower the delay is below 0.04 seconds and if the nodes are 50 or more the delay is 0.3 to 0.8 seconds.

4.3.2.3 Throughput

In this section we investigate the throughput for different types of traffic and DSSS Rate. The throughput is an important factor as well which affects the communication and the network performance.

- AODV

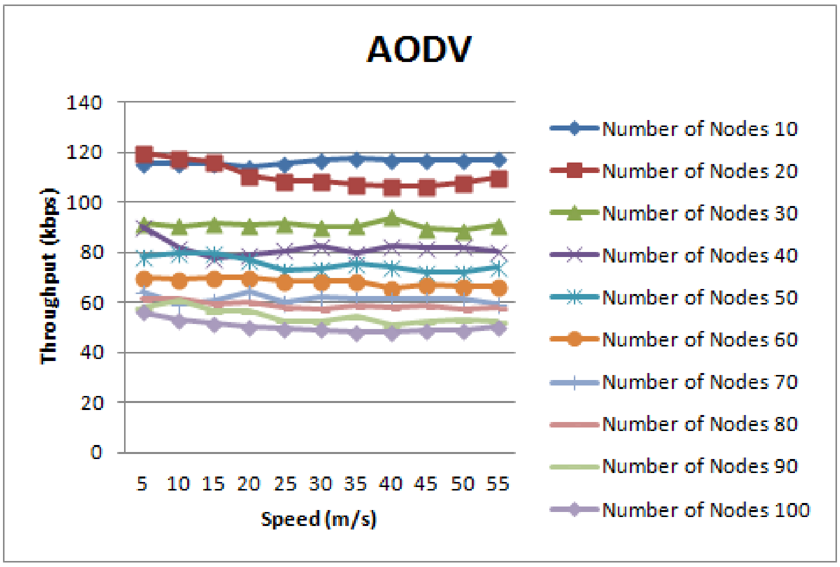


Figure 141 Throughput vs. Speed, AODV Routing Protocol

In Figure 141 we can see that the speed does not affect in a specific pattern the throughput. The variations in throughput because of the speed the nodes have are small. More specifically the throughput varies from 47kbps to 117kbps. As we can observe the throughput is being decreased as the number of nodes is getting bigger. This happens because of the small PDR AODV has for larger groups.

- DSDV

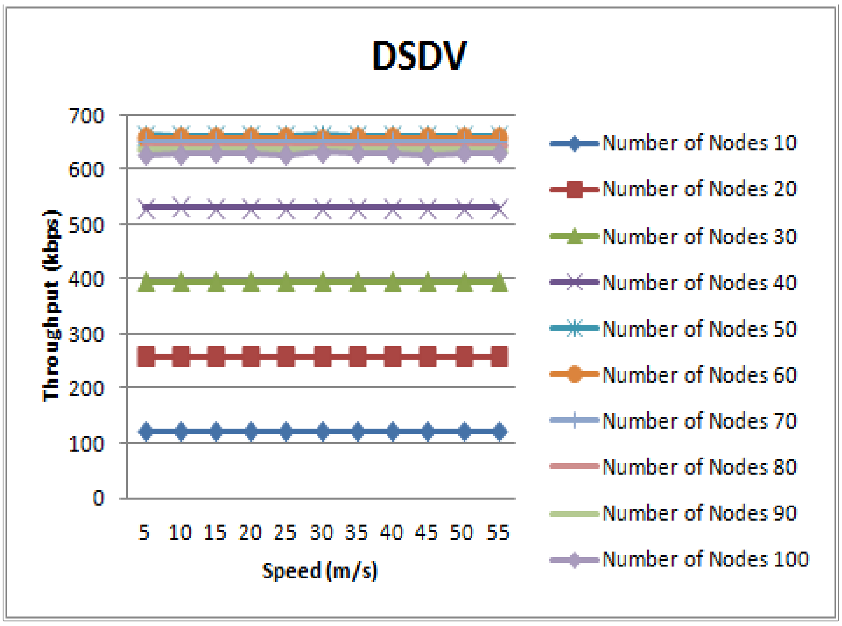


Figure 142 Throughput vs. Speed, DSDV Routing Protocol

In Figure 142 we can see how higher the throughput is for DSDV in contrast with AODV. The speed does not affect in a specific pattern the throughput. We can also observe that the variations of throughput for every number of nodes are even smaller than AODV. The throughput varies from 122kbps to 627 kbps. An interesting observation is that the throughput here is increased as the number of nodes is increased in unlike AODV. This happens because, despite that the PDR for DSDV is being decreased as the number of nodes is increased, the percentage of the successfully delivered packets is still on a very satisfactory level (the PDR is almost 50% for 100 nodes).

- OLSR

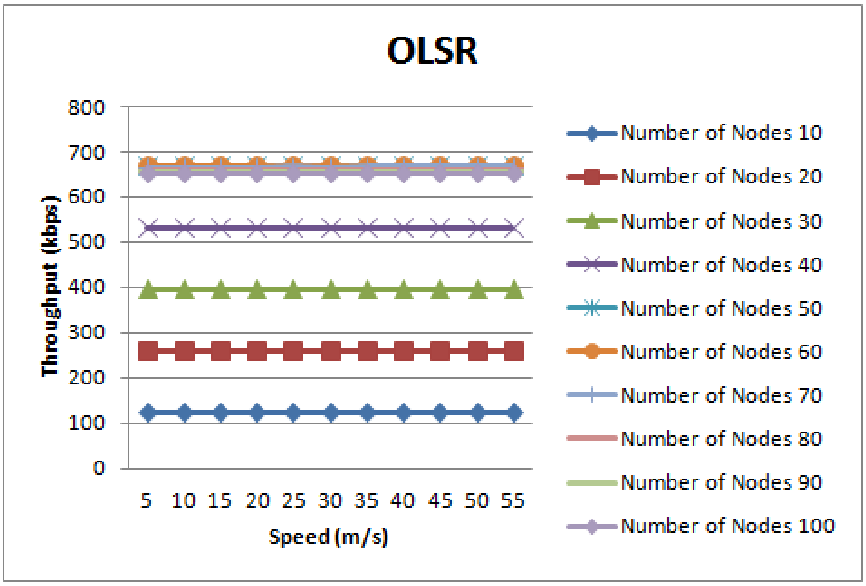


Figure 143 Throughput vs. Speed, OLSR Routing Protocol

Like the previous cases, OLSR has almost the same results as DSDV. In Figure 143 we can see how higher the throughput is in contrast with AODV. The speed does not affect in a specific pattern the throughput when OLSR is used either. We can also observe that the variations of throughput for every number of nodes are even smaller than AODV like DSDV. The throughput varies from 122kbps to 650 kbps. We have the same interesting observation like DSDV. The throughput here is increased as the number of nodes is increased in unlike AODV and this happens for the same reason as before.

4.3.2.4 Total Energy Consumption

The final perfume metric which is being studied is energy. Energy is a very important factor in routing protocols for MANETS, because in our scenario portable devices may not have the chance to be recharged and so the total energy consumption should be reduced as far it can be.

- AODV

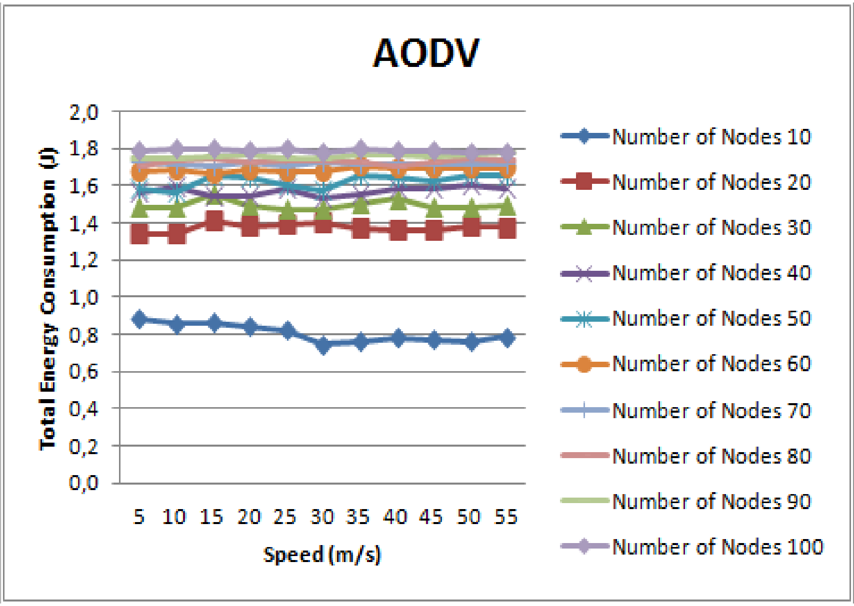


Figure 144 Total Energy Consumption vs. Speed, AODV Routing Protocol

In Figure 144 we can observe the results for the total energy consumption for AODV. As we can see, the best results are when we have few nodes. The speed does not affect in a specific pattern the results. As the number of nodes increases, the total energy consumption is being increased too. The energy consumption varies from 0.87J to 1.79J.

- DSDV

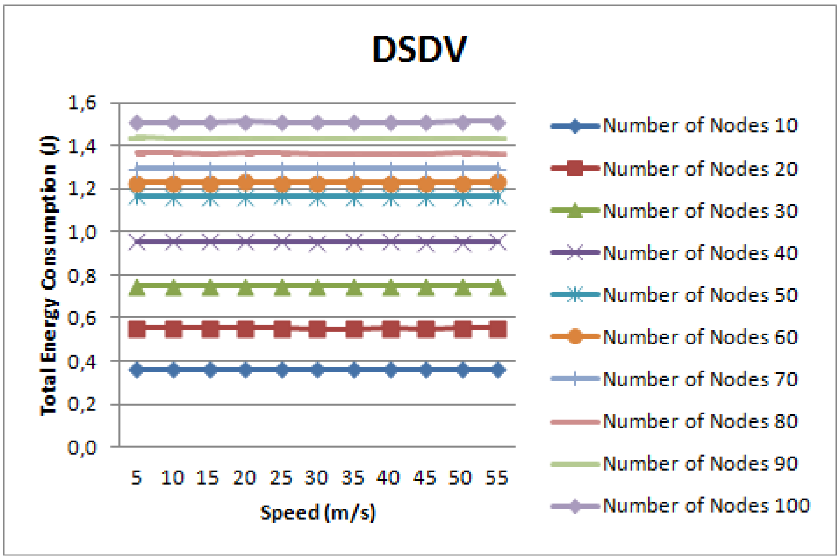


Figure 145 Total Energy Consumption vs. Speed, DSDV Routing Protocol

The total energy consumption as we can see in Figure 145 varies from 0.36J to 1.5J. The consumption is even distributed. For 10 to 50 nodes the energy consumption is increased by 0.2J and for 50 to 100 nodes the consumption is increased by almost 0.1J. The speed does not affect the energy consumption and as we can see when the speed changes the consumption changes for 0.001J or lower.

- OLSR

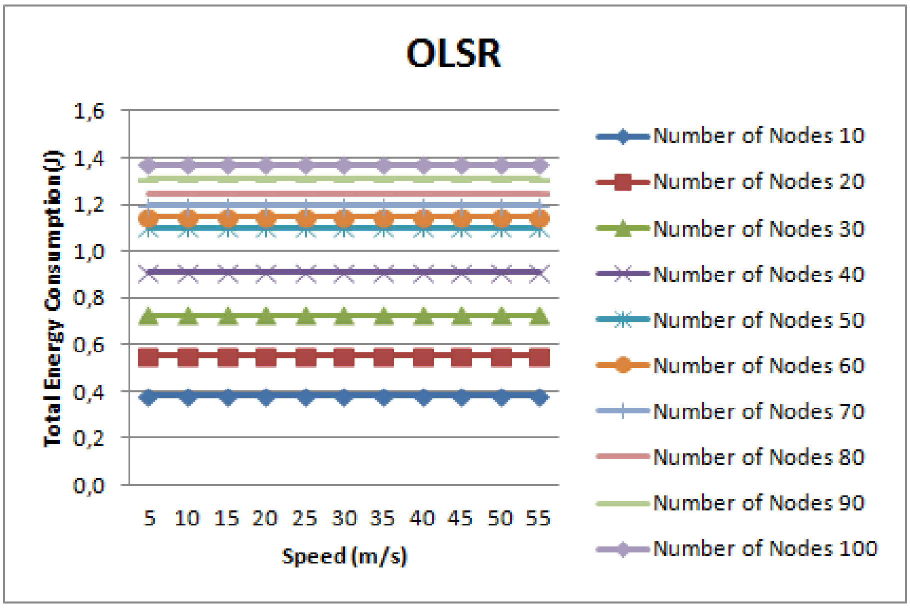


Figure 146 Total Energy Consumption vs. Speed, OLSR Routing Protocol

The results for OLSR as we can observe in Figure 146 are almost the same like DSDV. The consumption is even distributed. For 10 to 50 nodes the energy consumption is increased by 0.2J and for 50 to 70 nodes the consumption is increased by almost 0.04J and for 70 to 100 nodes the consumption is increased by approximately 0.1J. The speed does not affect the energy consumption in this case either and as we can see when the speed changes the consumption changes for 0.001J or lower.

4.3.2.5 Comparison of Routing Protocols

In this section we compare the three routing protocols. The diagrams that follow will provide a better understanding of the results. In the end of this section will be a summary of this comparison.

<u>Packet Delivery Ratio (PDR)</u>

In this section we compare the three routing protocols based on the PDR they achieve. The results are categorized based on the different speeds the nodes have. As we can observe in Figures 147-157 in every case OLSR outperforms DSDV (apart from one case when the number of nodes are 10 and the speed is 35m/s) and they both outperforms AODV.

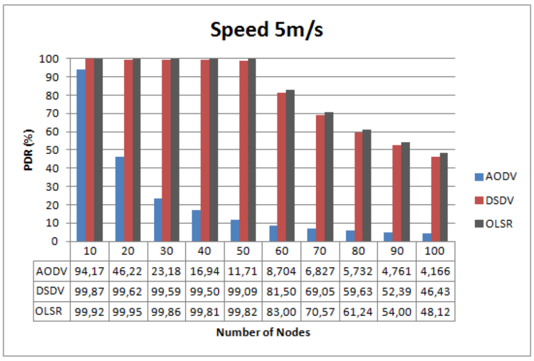


Figure 147 PDR vs. Number of Nodes, Speed 5m/s

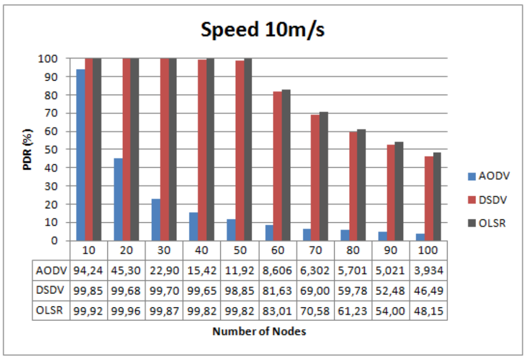


Figure 148 PDR vs. Number of Nodes, Speed 10m/s

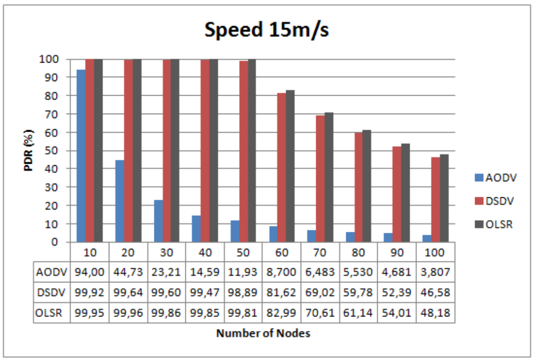


Figure 149 PDR vs. Number of Nodes, Speed 15m/s

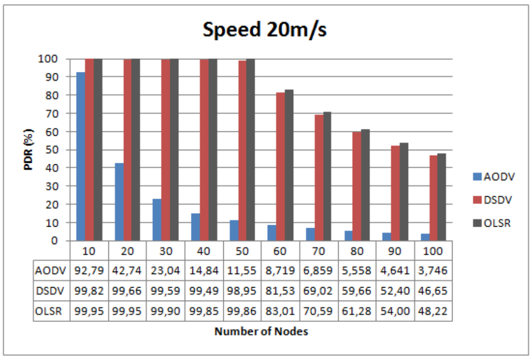


Figure 150 PDR vs. Number of Nodes, Speed 20m/s

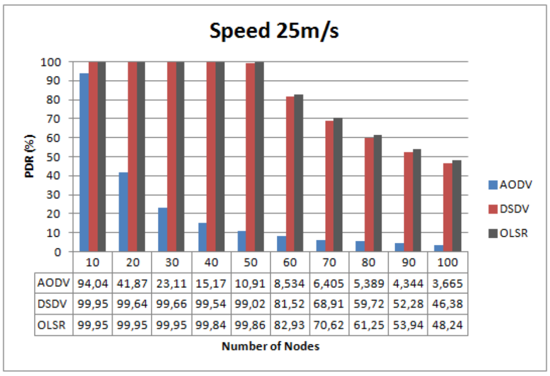


Figure 151 PDR vs. Number of Nodes, Speed 25m/s

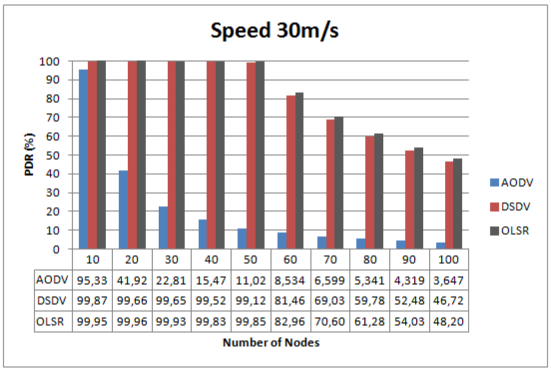


Figure 152 PDR vs. Number of Nodes, Speed 30m/s

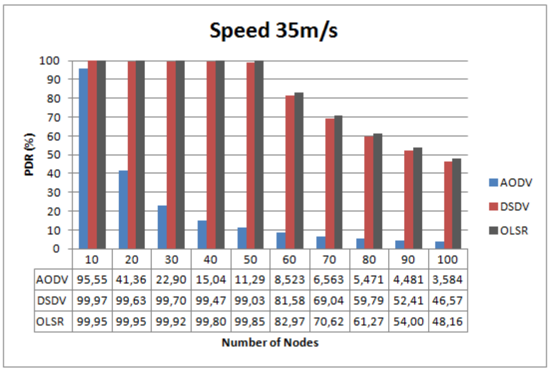


Figure 153 PDR vs. Number of Nodes, Speed 35m/s

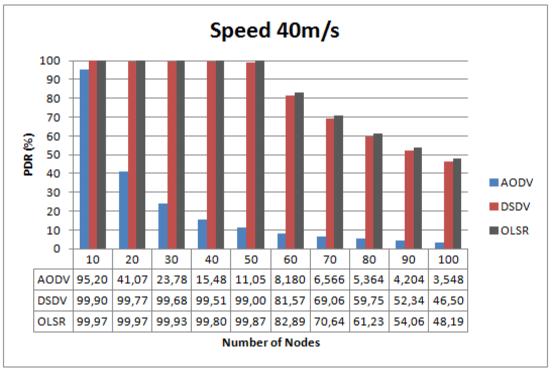


Figure 154 PDR vs. Number of Nodes, Speed 40m/s

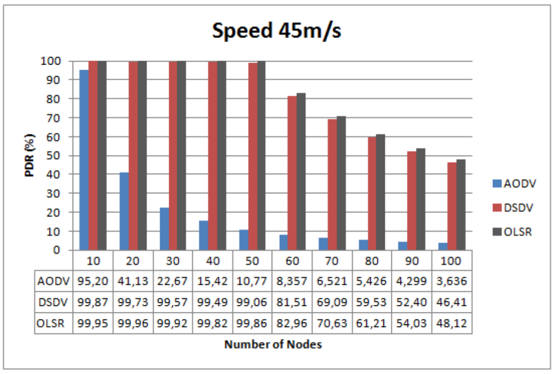


Figure 155 PDR vs. Number of Nodes, Speed 45m/s

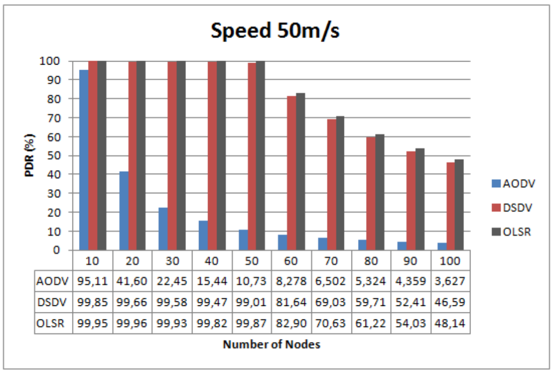


Figure 156 PDR vs. Number of Nodes, Speed 50m/s

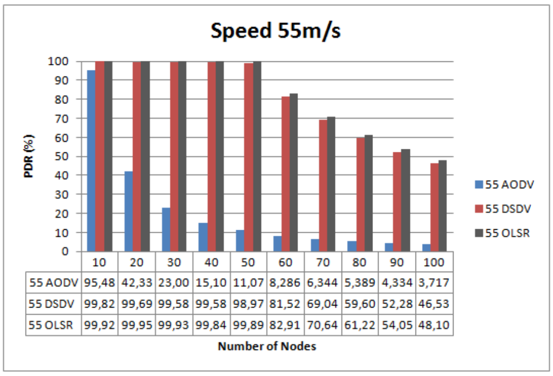


Figure 157 PDR vs. Number of Nodes, Speed 55m/s

The figure that follows (Figure 158) summarizes the results and presents a comparison between the three routing protocols based on the average speed and the number of nodes.

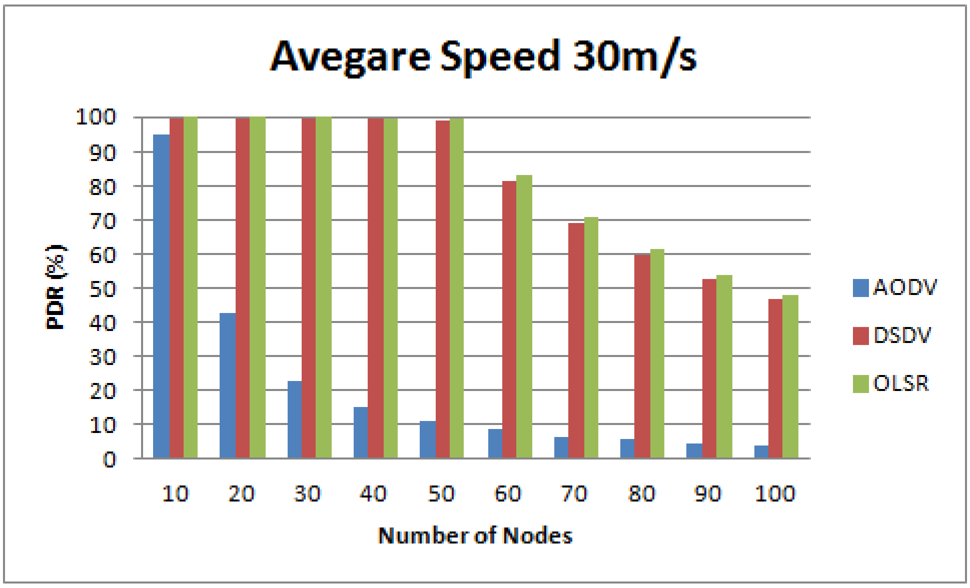
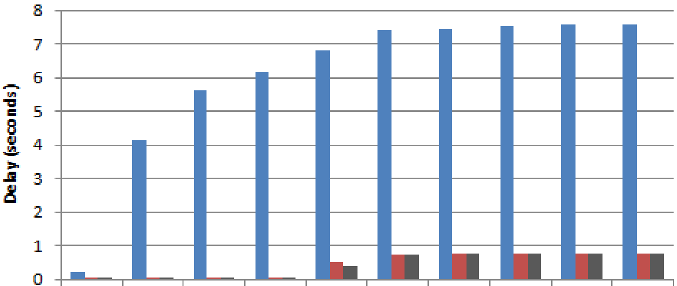


Figure 158 PDR vs. Number of Nodes, Average Speed 30m/s.

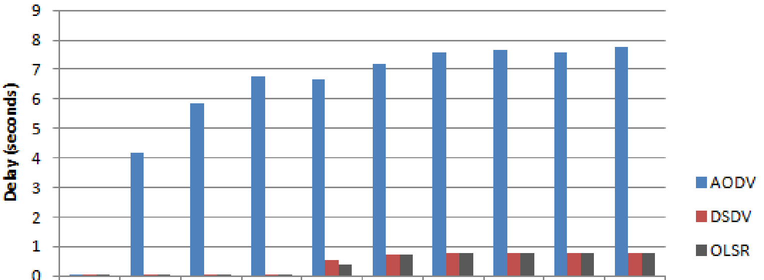
Average Delay

In this section we can compare the three routing protocols based on the average delay they achieve. The results are categorized based on the different speeds the nodes have. As we can observe in Figures 159-169 in every case OLSR outperforms DSDV, apart from one case. When the number of nodes is 60 DSDV always outperforms OLSR (and the number of nodes is 70 and the speed is 50m/s). They both outperform AODV.



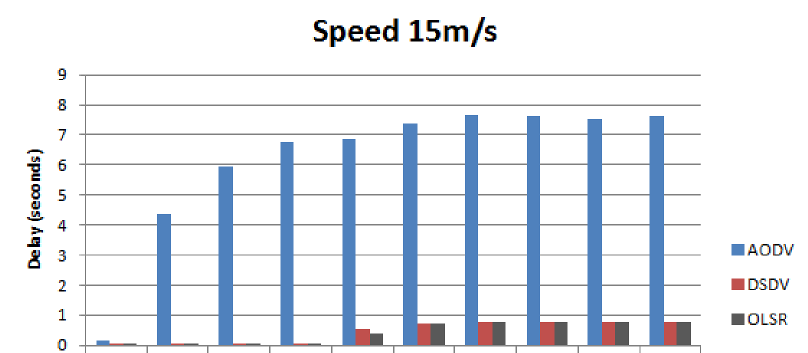
	10	20	30	40	50	60	70	80	90	100
AODV	0,2202	4,1150	5,6218	6,1881	6,8145	7,4137	7,4648	7,5576	7,5756	7,5944
DSDV	0,0101	0,0209	0,0334	0,0555	0,5123	0,7169	0,7585	0,7766	0,7891	0,7949
OLSR	0,0101	0,0206	0,0312	0,0420	0,3750	0,7221	0,7562	0,7742	0,7826	0,7946

Figure 159 Delay vs. Number of Nodes, speed 5m/s.



	10	20	30	40	50	60	70	80	90	100
AODV	0,0470	4,1687	5,8674	6,7615	6,6393	7,2068	7,5624	7,6467	7,5779	7,7764
DSDV	0,0101	0,0209	0,0333	0,0561	0,5242	0,7161	0,7572	0,7760	0,7881	0,7947
OLSR	0,0101	0,0206	0,0313	0,0420	0,3749	0,7219	0,7562	0,7737	0,7829	0,7953

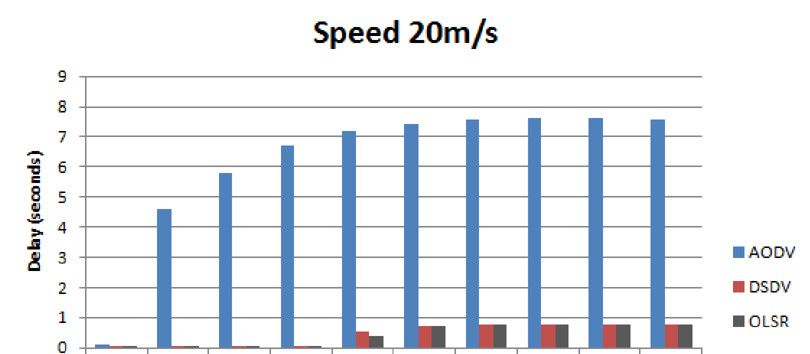
Figure 160 Delay vs. Number of Nodes, speed 10m/s.



	10	20	30	40	50	60	70	80	90	100
AODV	0,1520	4,3588	5,9318	6,7776	6,8296	7,3791	7,6616	7,6203	7,5000	7,6371
DSDV	0,0101	0,0209	0,0334	0,0562	0,5238	0,7165	0,7569	0,7748	0,7888	0,7963
OLSR	0,0101	0,0206	0,0312	0,0420	0,3743	0,7220	0,7564	0,7739	0,783	0,7951

Number of Nodes

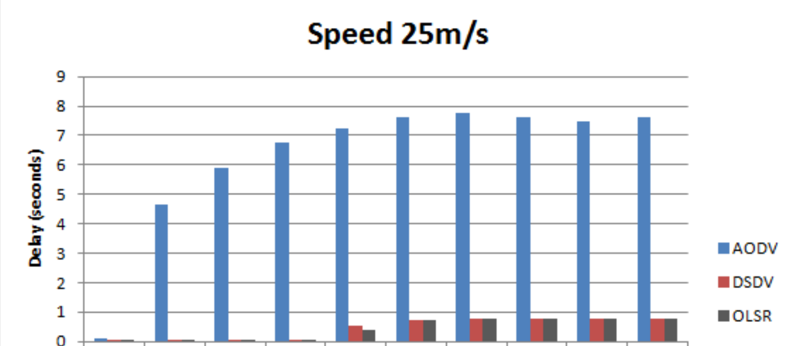
Figure 161 Delay vs. Number of Nodes, speed 15m/s.



	10	20	30	40	50	60	70	80	90	100
AODV	0,1220	4,6214	5,8093	6,6949	7,1733	7,4082	7,5598	7,6412	7,6091	7,5839
DSDV	0,0101	0,0209	0,0333	0,0566	0,5222	0,7175	0,7572	0,7755	0,7888	0,7963
OLSR	0,0101	0,0206	0,0313	0,0420	0,3749	0,7222	0,7562	0,774	0,783	0,7945

Number of Nodes

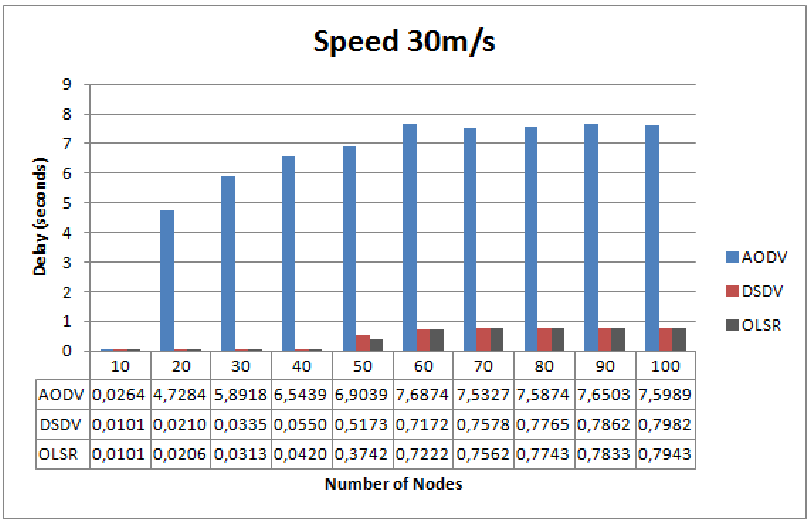
Figure 162 Delay vs. Number of Nodes, speed 20m/s.



	10	20	30	40	50	60	70	80	90	100
AODV	0,0854	4,6662	5,8766	6,7763	7,2118	7,5960	7,7522	7,6246	7,4834	7,6092
DSDV	0,0101	0,0209	0,0336	0,0560	0,5177	0,7169	0,7572	0,7748	0,7871	0,8005
OLSR	0,0101	0,0206	0,0313	0,0420	0,3747	0,7217	0,7565	0,774	0,7823	0,795

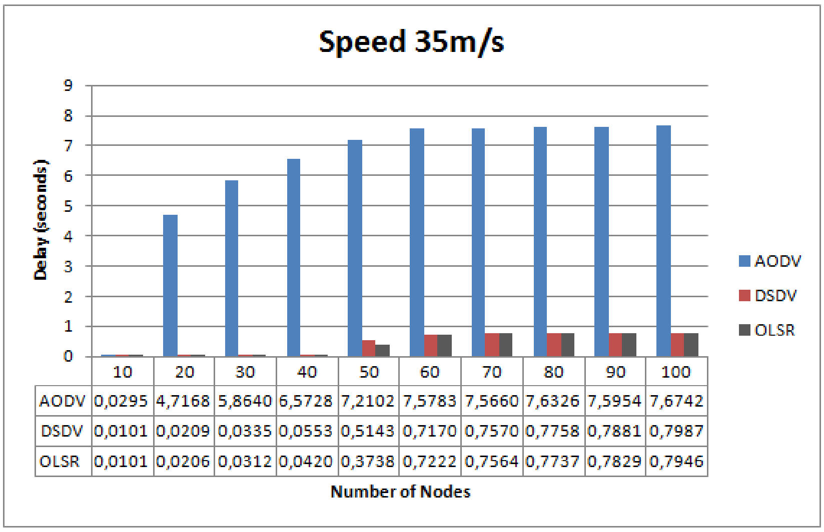
Number of Nodes

Figure 163 Delay vs. Number of Nodes, speed 25m/s.



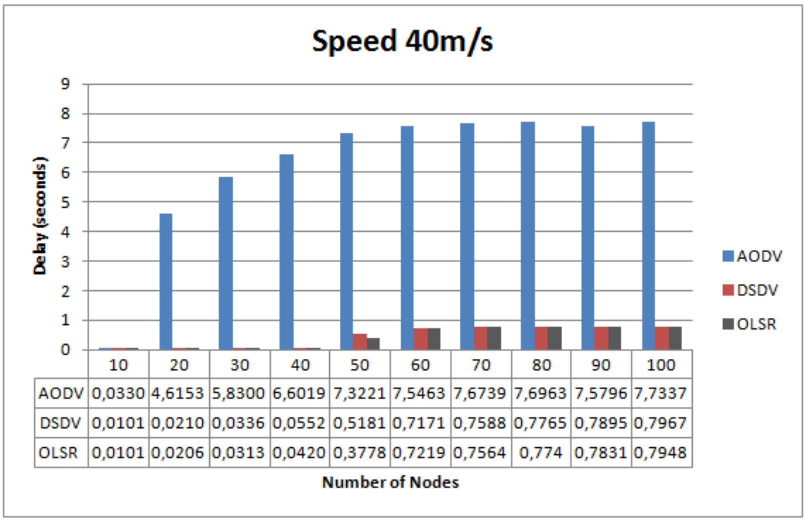
	10	20	30	40	50	60	70	80	90	100
AODV	0,0264	4,7284	5,8918	6,5439	6,9039	7,6874	7,5327	7,5874	7,6503	7,5989
DSDV	0,0101	0,0210	0,0335	0,0550	0,5173	0,7172	0,7578	0,7765	0,7862	0,7982
OLSR	0,0101	0,0206	0,0313	0,0420	0,3742	0,7222	0,7562	0,7743	0,7833	0,7943

Figure 164 Delay vs. Number of Nodes, speed 30m/s.



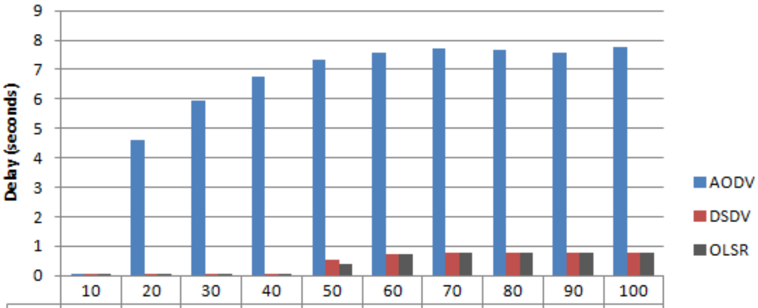
	10	20	30	40	50	60	70	80	90	100
AODV	0,0295	4,7168	5,8640	6,5728	7,2102	7,5783	7,5660	7,6326	7,5954	7,6742
DSDV	0,0101	0,0209	0,0335	0,0553	0,5143	0,7170	0,7570	0,7758	0,7881	0,7987
OLSR	0,0101	0,0206	0,0312	0,0420	0,3738	0,7222	0,7564	0,7737	0,7829	0,7946

Figure 165 Delay vs. Number of Nodes, speed 35m/s.



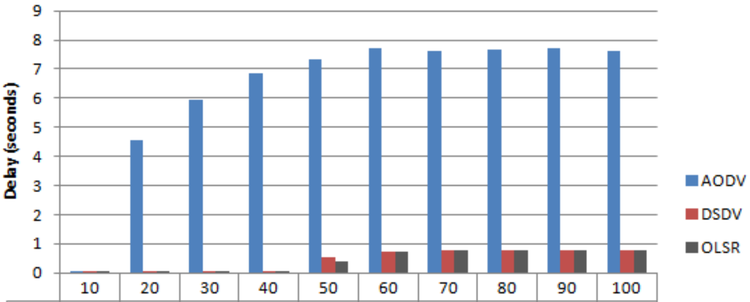
	10	20	30	40	50	60	70	80	90	100
AODV	0,0330	4,6153	5,8300	6,6019	7,3221	7,5463	7,6739	7,6963	7,5796	7,7337
DSDV	0,0101	0,0210	0,0336	0,0552	0,5181	0,7171	0,7588	0,7765	0,7895	0,7967
OLSR	0,0101	0,0206	0,0313	0,0420	0,3778	0,7219	0,7564	0,774	0,7831	0,7948

Figure 166 Delay vs. Number of Nodes, speed 40m/s.



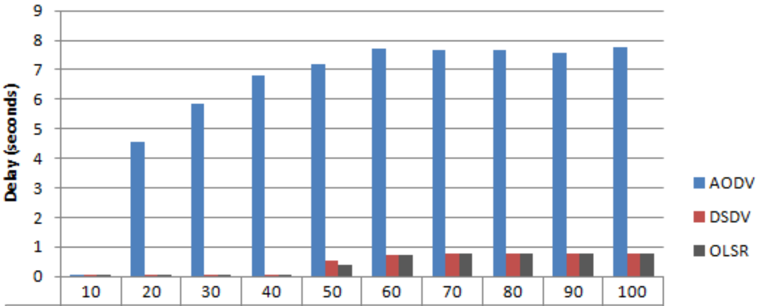
	10	20	30	40	50	60	70	80	90	100
AODV	0,0300	4,5901	5,9349	6,7331	7,3174	7,5661	7,7098	7,6844	7,5890	7,7722
DSDV	0,0101	0,0209	0,0334	0,0552	0,5148	0,7173	0,7582	0,7762	0,7874	0,7979
OLSR	0,0101	0,0206	0,0312	0,0420	0,3778	0,7223	0,7564	0,7738	0,7826	0,7943

Figure 167 Delay vs. Number of Nodes, speed 45m/s.



	10	20	30	40	50	60	70	80	90	100
AODV	0,0273	4,5566	5,9527	6,8287	7,3222	7,7024	7,6332	7,6496	7,7309	7,6196
DSDV	0,0101	0,0210	0,0335	0,0553	0,5199	0,7175	0,7560	0,7749	0,7862	0,7958
OLSR	0,0101	0,0206	0,0312	0,0420	0,3776	0,7220	0,7564	0,7736	0,783	0,795

Figure 168 Delay vs. Number of Nodes, speed 50m/s.



	10	20	30	40	50	60	70	80	90	100
AODV	0,0292	4,5377	5,8465	6,8037	7,2089	7,7086	7,6504	7,6587	7,5601	7,7823
DSDV	0,0101	0,0209	0,0335	0,0565	0,5183	0,7166	0,7579	0,7754	0,7868	0,7960
OLSR	0,0101	0,0206	0,0313	0,0420	0,3783	0,7220	0,7563	0,7742	0,7832	0,7949

Figure 169 Delay vs. Number of Nodes, speed 55m/s.

The figure that follows (Figure 170) summarizes the results and presents a comparison between the three routing protocols based on the average speed and the number of nodes.

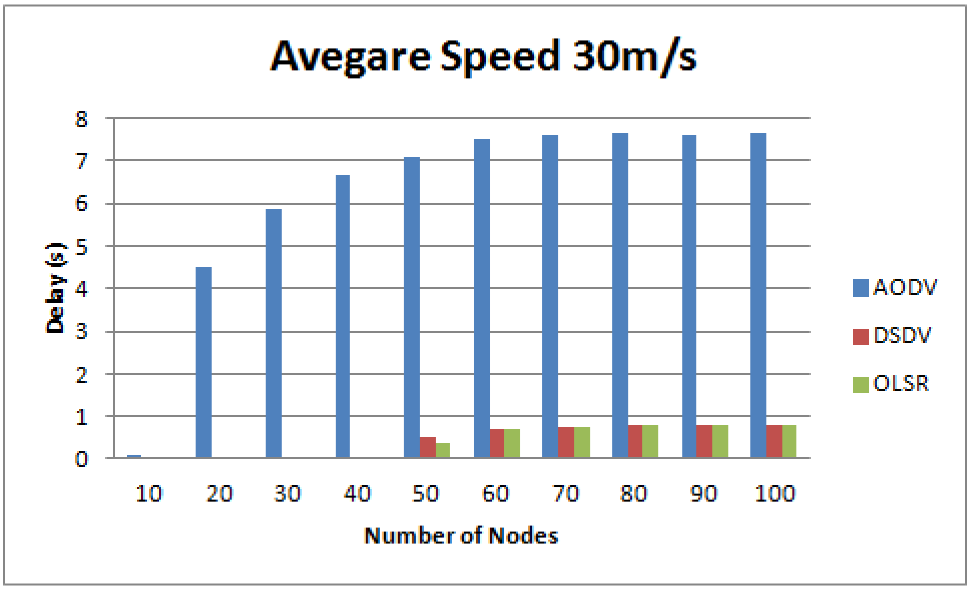
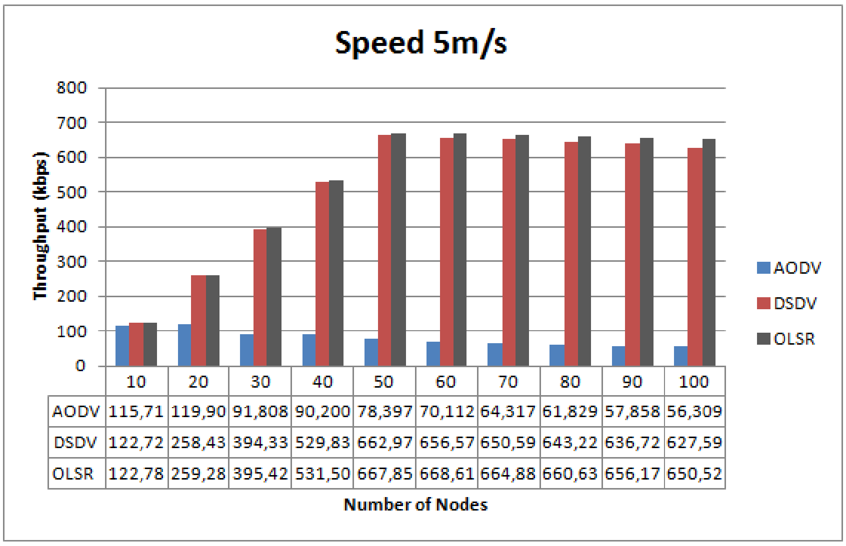


Figure 170 Delay vs. Number of Nodes, Average Speed 30m/s.

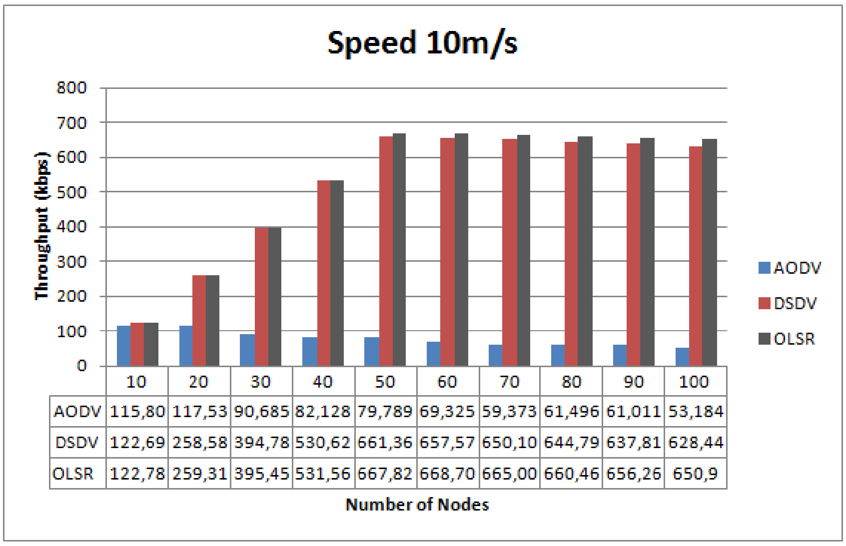
Throughput

In this section we can compare the three routing protocols based on the throughput they achieve. The results are categorized based on the different speeds the nodes have. As we can observe in Figures 171-181 in every case OLSR outperforms DSDV, apart from one case (when the number of nodes is 10 and the speed is 35m/s). They both outperform AODV.



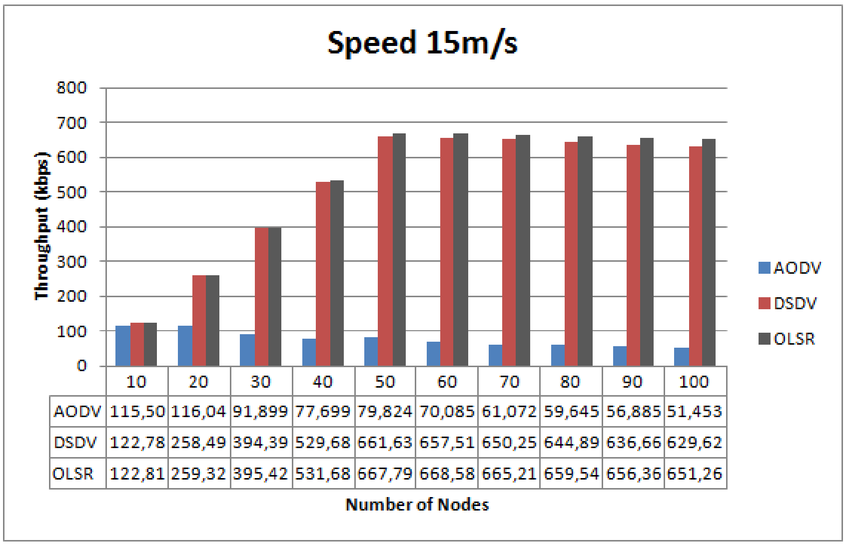
	10	20	30	40	50	60	70	80	90	100
AODV	115,71	119,90	91,808	90,200	78,397	70,112	64,317	61,829	57,858	56,309
DSDV	122,72	258,43	394,33	529,83	662,97	656,57	650,59	643,22	636,72	627,59
OLSR	122,78	259,28	395,42	531,50	667,85	668,61	664,88	660,63	656,17	650,52

Figure 171 Throughput vs. Number of Nodes, speed 5m/s.



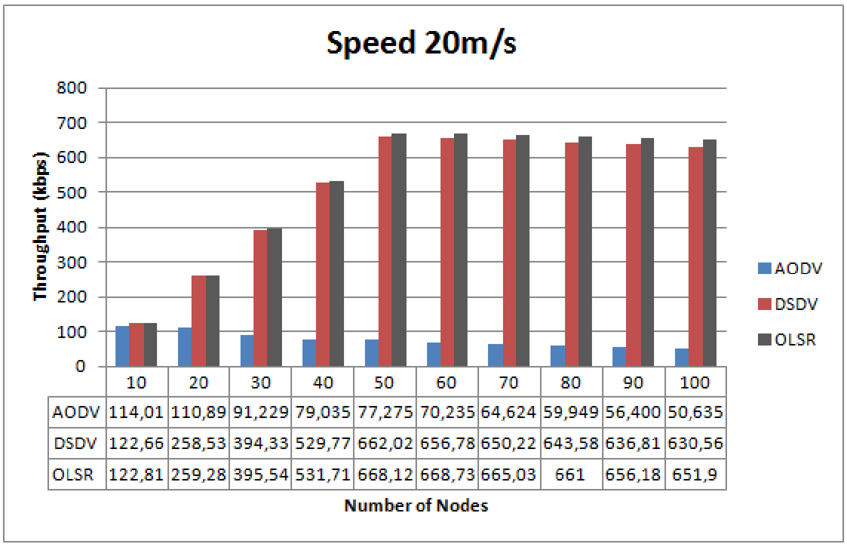
	10	20	30	40	50	60	70	80	90	100
AODV	115,80	117,53	90,685	82,128	79,789	69,325	59,373	61,496	61,011	53,184
DSDV	122,69	258,58	394,78	530,62	661,36	657,57	650,10	644,79	637,81	628,44
OLSR	122,78	259,31	395,45	531,56	667,82	668,70	665,00	660,46	656,26	650,9

Figure 172 Throughput vs. Number of Nodes, speed 10m/s.



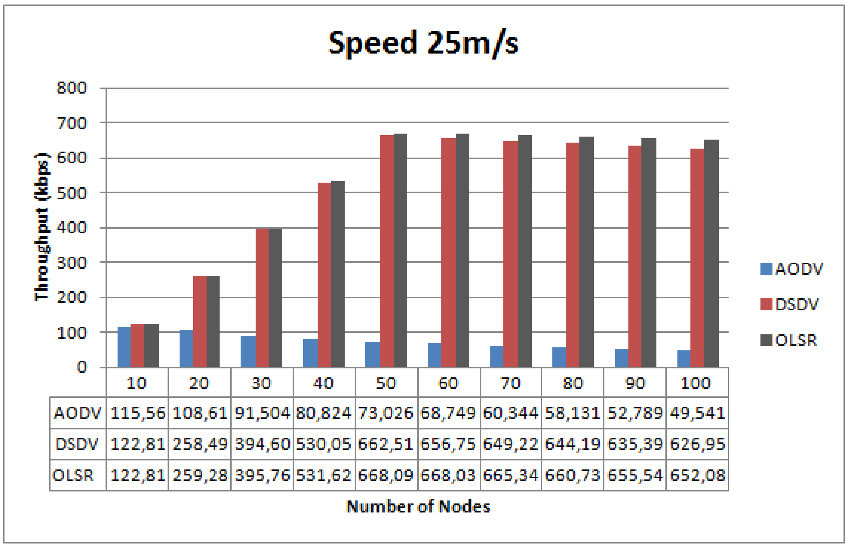
	10	20	30	40	50	60	70	80	90	100
AODV	115,50	116,04	91,899	77,699	79,824	70,085	61,072	59,645	56,885	51,453
DSDV	122,78	258,49	394,39	529,68	661,63	657,51	650,25	644,89	636,66	629,62
OLSR	122,81	259,32	395,42	531,68	667,79	668,58	665,21	659,54	656,36	651,26

Figure 173 Throughput vs. Number of Nodes, speed 15m/s.



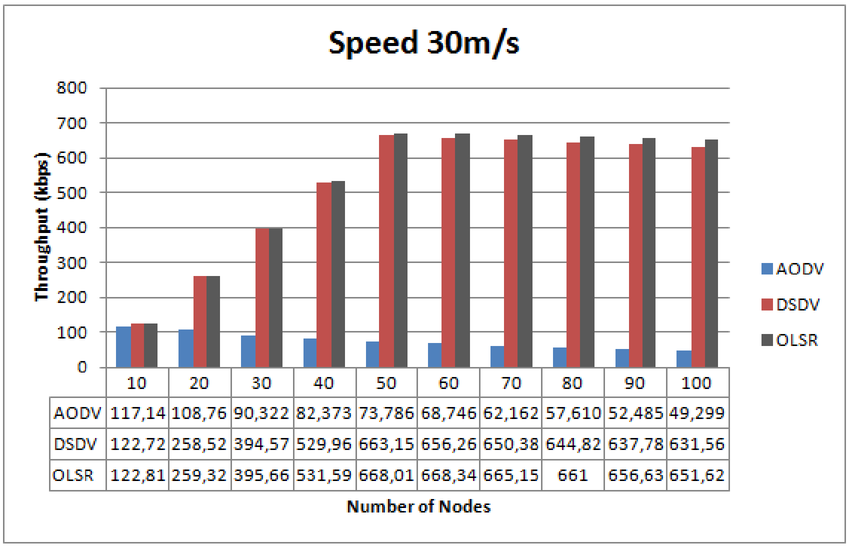
	10	20	30	40	50	60	70	80	90	100
AODV	114,01	110,89	91,229	79,035	77,275	70,235	64,624	59,949	56,400	50,635
DSDV	122,66	258,53	394,33	529,77	662,02	656,78	650,22	643,58	636,81	630,56
OLSR	122,81	259,28	395,54	531,71	668,12	668,73	665,03	661	656,18	651,9

Figure 174 Throughput vs. Number of Nodes, speed 20m/s.



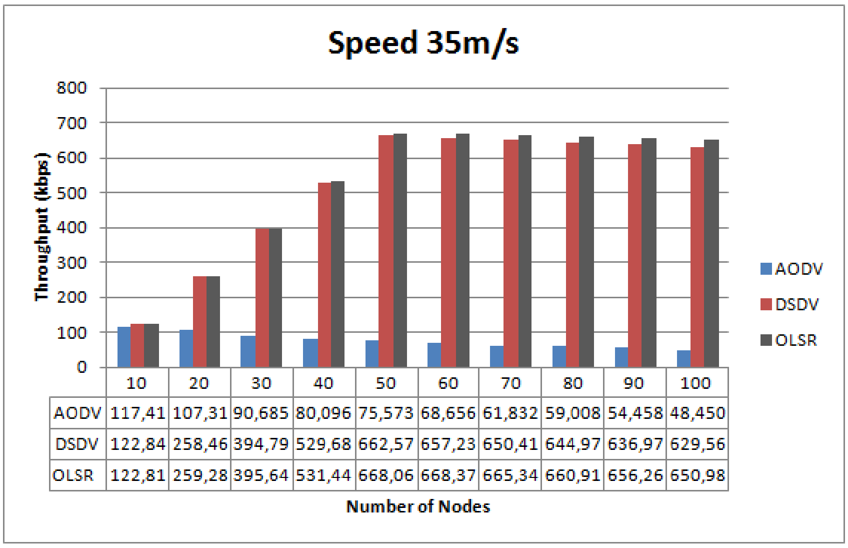
	10	20	30	40	50	60	70	80	90	100
AODV	115,56	108,61	91,504	80,824	73,026	68,749	60,344	58,131	52,789	49,541
DSDV	122,81	258,49	394,60	530,05	662,51	656,75	649,22	644,19	635,39	626,95
OLSR	122,81	259,28	395,76	531,62	668,09	668,03	665,34	660,73	655,54	652,08

Figure 175 Throughput vs. Number of Nodes, speed 25m/s.



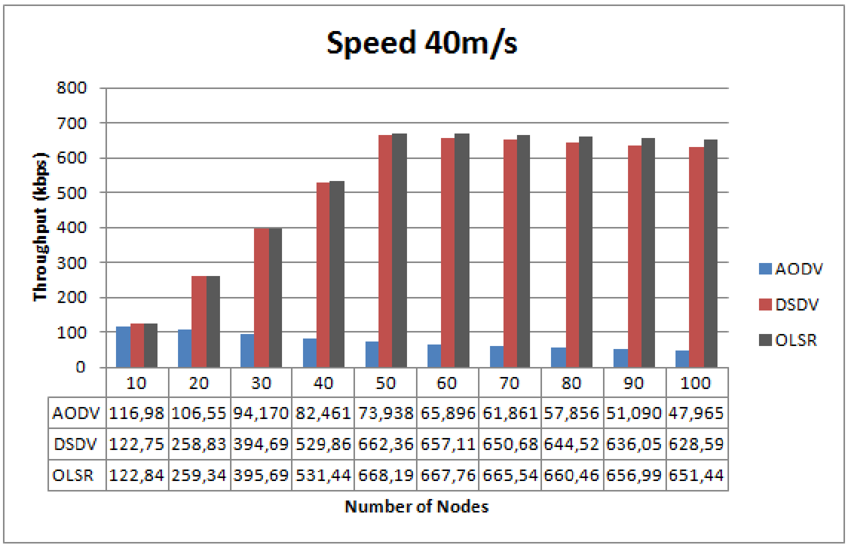
	10	20	30	40	50	60	70	80	90	100
AODV	117,14	108,76	90,322	82,373	73,786	68,746	62,162	57,610	52,485	49,299
DSDV	122,72	258,52	394,57	529,96	663,15	656,26	650,38	644,82	637,78	631,56
OLSR	122,81	259,32	395,66	531,59	668,01	668,34	665,15	661	656,63	651,62

Figure 176 Throughput vs. Number of Nodes, speed 30m/s.



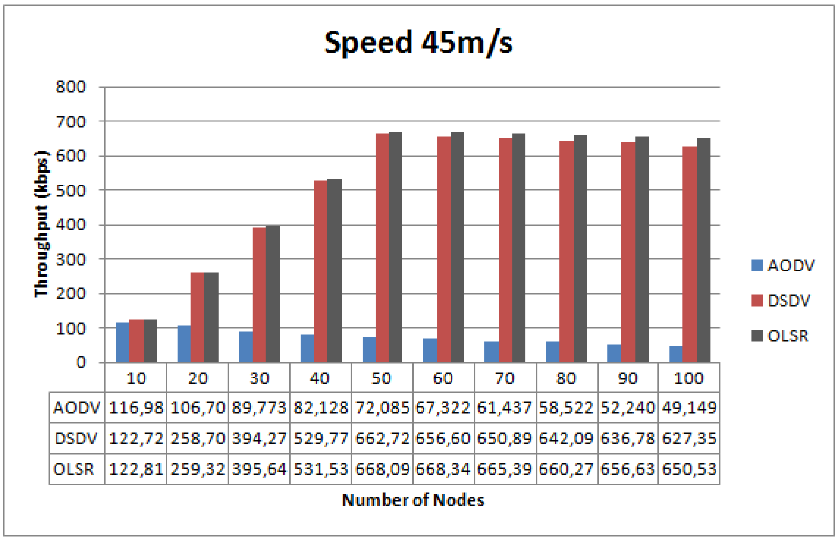
	10	20	30	40	50	60	70	80	90	100
AODV	117,41	107,31	90,685	80,096	75,573	68,656	61,832	59,008	54,458	48,450
DSDV	122,84	258,46	394,79	529,68	662,57	657,23	650,41	644,97	636,97	629,56
OLSR	122,81	259,28	395,64	531,44	668,06	668,37	665,34	660,91	656,26	650,98

Figure 177 Throughput vs. Number of Nodes, speed 35m/s.



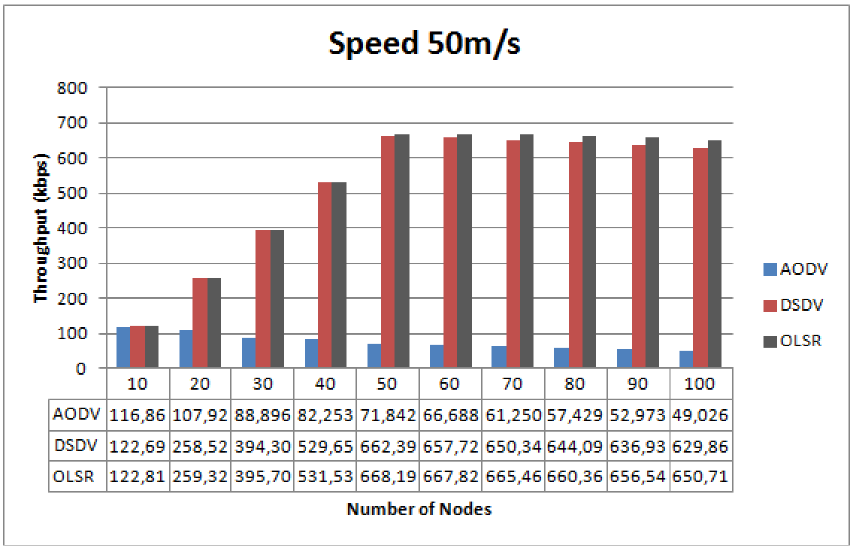
	10	20	30	40	50	60	70	80	90	100
AODV	116,98	106,55	94,170	82,461	73,938	65,896	61,861	57,856	51,090	47,965
DSDV	122,75	258,83	394,69	529,86	662,36	657,11	650,68	644,52	636,05	628,59
OLSR	122,84	259,34	395,69	531,44	668,19	667,76	665,54	660,46	656,99	651,44

Figure 178 Throughput vs. Number of Nodes, speed 40m/s.



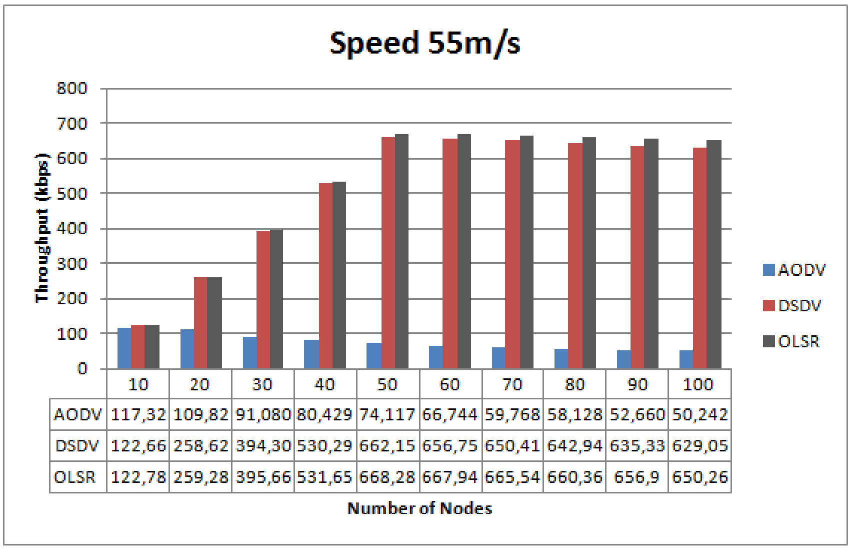
	10	20	30	40	50	60	70	80	90	100
AODV	116,98	106,70	89,773	82,128	72,085	67,322	61,437	58,522	52,240	49,149
DSDV	122,72	258,70	394,27	529,77	662,72	656,60	650,89	642,09	636,78	627,35
OLSR	122,81	259,32	395,64	531,53	668,09	668,34	665,39	660,27	656,63	650,53

Figure 179 Throughput vs. Number of Nodes, speed 45m/s.



	10	20	30	40	50	60	70	80	90	100
AODV	116,86	107,92	88,896	82,253	71,842	66,688	61,250	57,429	52,973	49,026
DSDV	122,69	258,52	394,30	529,65	662,39	657,72	650,34	644,09	636,93	629,86
OLSR	122,81	259,32	395,70	531,53	668,19	667,82	665,46	660,36	656,54	650,71

Figure 180 Throughput vs. Number of Nodes, speed 50m/s.



	10	20	30	40	50	60	70	80	90	100
AODV	117,32	109,82	91,080	80,429	74,117	66,744	59,768	58,128	52,660	50,242
DSDV	122,66	258,62	394,30	530,29	662,15	656,75	650,41	642,94	635,33	629,05
OLSR	122,78	259,28	395,66	531,65	668,28	667,94	665,54	660,36	656,9	650,26

Figure 181 Throughput vs. Number of Nodes, speed 55m/s.

The figure that follows (Figure 182) summarizes the results and presents a comparison between the three routing protocols based on the average speed and the number of nodes.

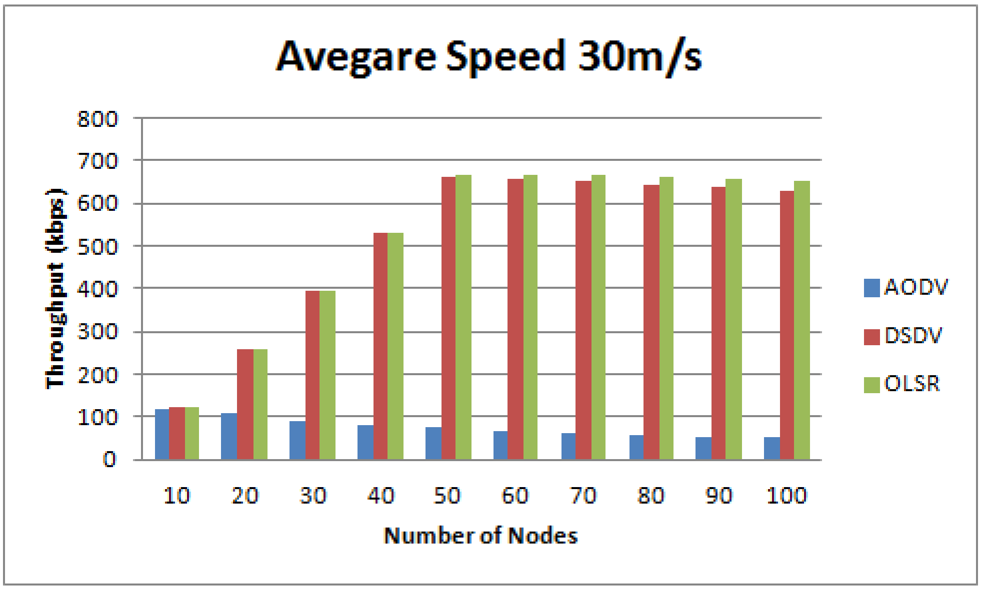
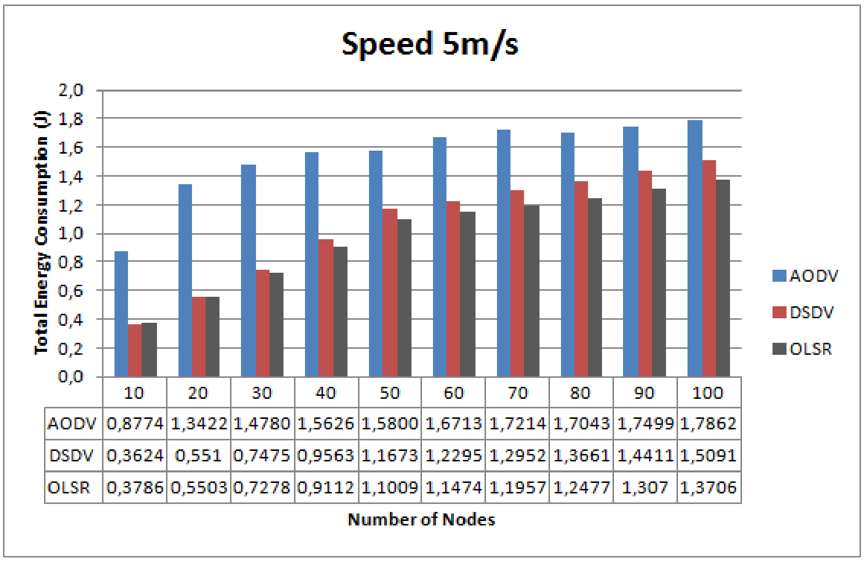


Figure 182 Throughput vs. Number of Nodes, Average Speed 30m/s.

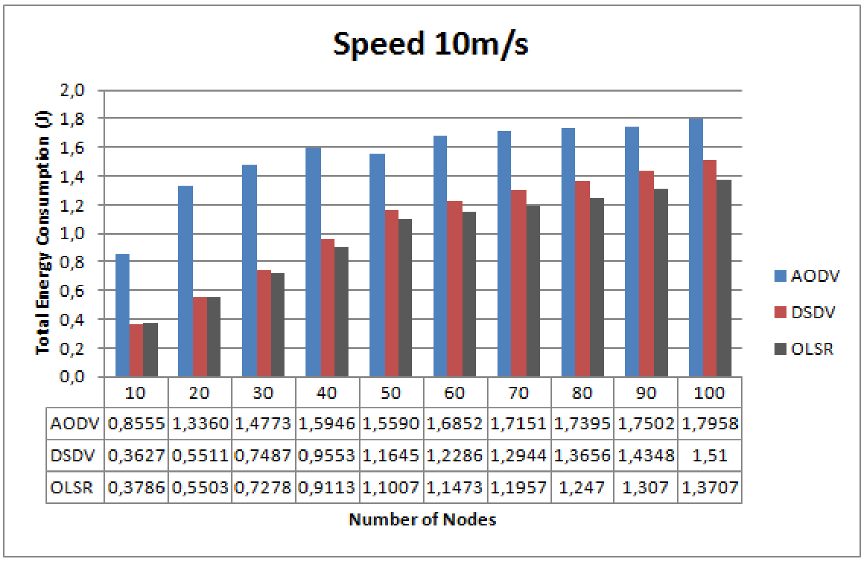
Total Energy Consumption

In this section we can compare the three routing protocols based on the total energy consumption they achieve. The results are categorized based on the different speeds the nodes have. As we can observe in Figures 183-193 in every case OLSR outperforms DSDV, apart from one case where the number of nodes is 10 and when the number of nodes is 20 and the speed is 30m/s and 35m/s. They both outperform AODV.



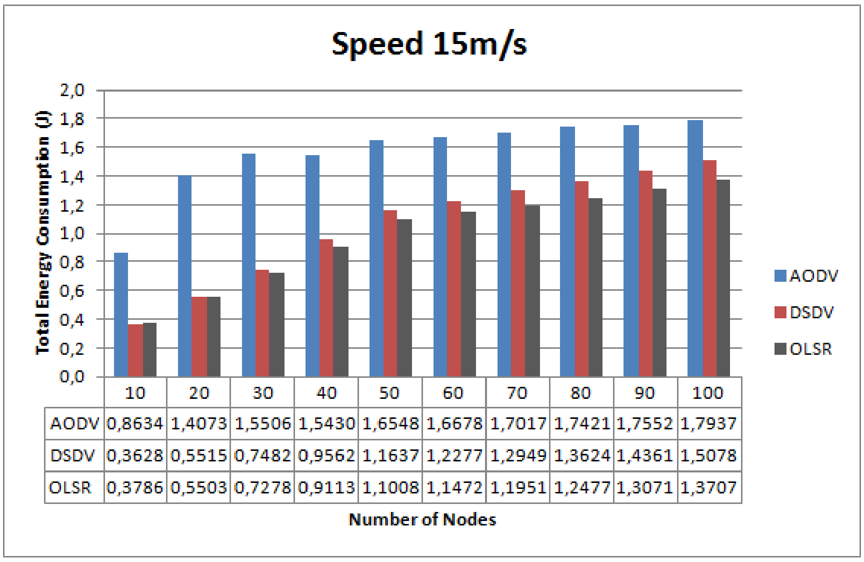
	10	20	30	40	50	60	70	80	90	100
AODV	0,8774	1,3422	1,4780	1,5626	1,5800	1,6713	1,7214	1,7043	1,7499	1,7862
DSDV	0,3624	0,551	0,7475	0,9563	1,1673	1,2295	1,2952	1,3661	1,4411	1,5091
OLSR	0,3786	0,5503	0,7278	0,9112	1,1009	1,1474	1,1957	1,2477	1,307	1,3706

Figure 183 Total Energy Consumption vs. Number of Nodes, speed 5m/s.



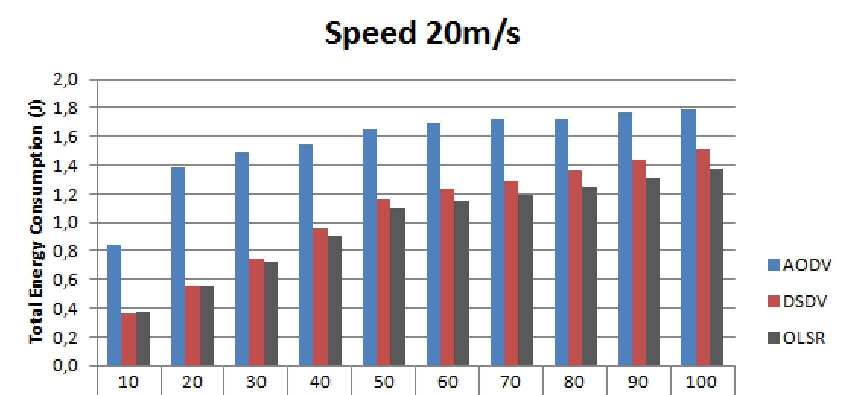
	10	20	30	40	50	60	70	80	90	100
AODV	0,8555	1,3360	1,4773	1,5946	1,5590	1,6852	1,7151	1,7395	1,7502	1,7958
DSDV	0,3627	0,5511	0,7487	0,9553	1,1645	1,2286	1,2944	1,3656	1,4348	1,51
OLSR	0,3786	0,5503	0,7278	0,9113	1,1007	1,1473	1,1957	1,247	1,307	1,3707

Figure 184 Total Energy Consumption vs. Number of Nodes, speed 10m/s.



	10	20	30	40	50	60	70	80	90	100
AODV	0,8634	1,4073	1,5506	1,5430	1,6548	1,6678	1,7017	1,7421	1,7552	1,7937
DSDV	0,3628	0,5515	0,7482	0,9562	1,1637	1,2277	1,2949	1,3624	1,4361	1,5078
OLSR	0,3786	0,5503	0,7278	0,9113	1,1008	1,1472	1,1951	1,2477	1,3071	1,3707

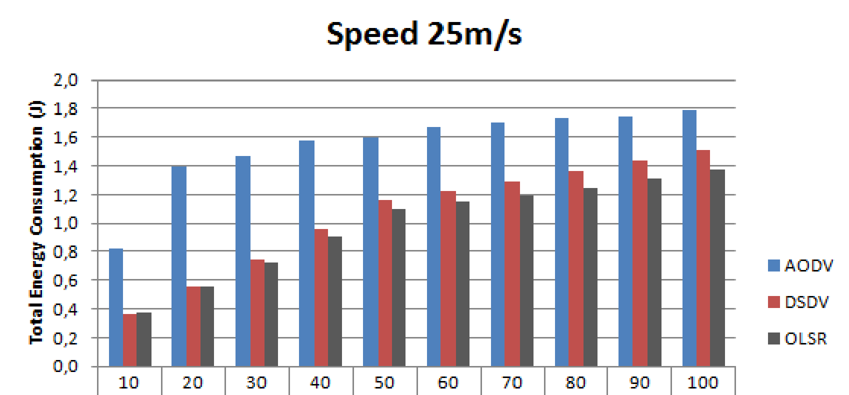
Figure 185 Total Energy Consumption vs. Number of Nodes, speed 15m/s.



	10	20	30	40	50	60	70	80	90	100
AODV	0,8424	1,3817	1,4941	1,5441	1,6455	1,6896	1,7212	1,7216	1,7654	1,7858
DSDV	0,3626	0,5514	0,7469	0,9568	1,1647	1,2307	1,2932	1,365	1,4378	1,5138
OLSR	0,3786	0,5503	0,7278	0,9113	1,1009	1,1473	1,1959	1,247	1,307	1,3707

Number of Nodes

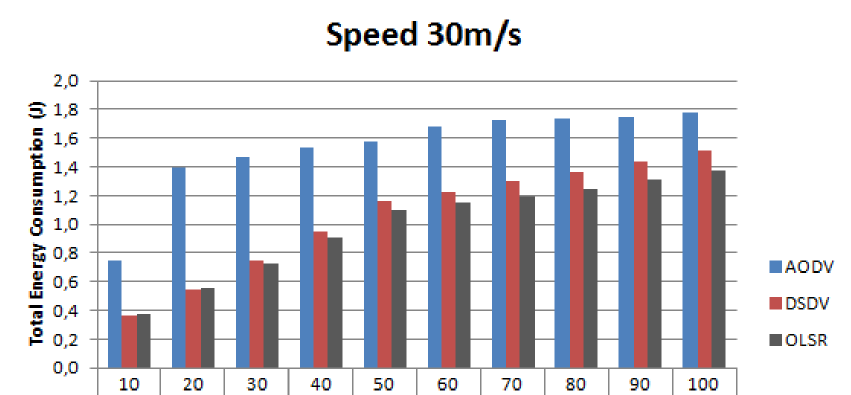
Figure 186 Total Energy Consumption vs. Number of Nodes, speed 20m/s.



	10	20	30	40	50	60	70	80	90	100
AODV	0,8184	1,3911	1,4698	1,5811	1,6006	1,6765	1,7023	1,7357	1,7451	1,7932
DSDV	0,363	0,5508	0,7491	0,9562	1,1665	1,2295	1,2916	1,3658	1,4343	1,5108
OLSR	0,3786	0,5503	0,7279	0,9112	1,1008	1,1471	1,1951	1,2479	1,3073	1,3708

Number of Nodes

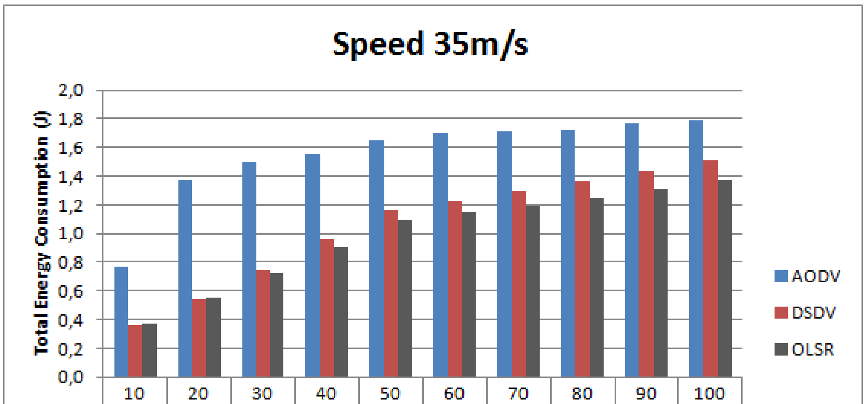
Figure 187 Total Energy Consumption vs. Number of Nodes, speed 25m/s.



	10	20	30	40	50	60	70	80	90	100
AODV	0,7439	1,3960	1,4748	1,5357	1,5771	1,6784	1,7251	1,7382	1,7507	1,7776
DSDV	0,3629	0,55	0,7482	0,9517	1,1645	1,2284	1,2945	1,3621	1,4376	1,5082
OLSR	0,3786	0,5503	0,7278	0,9113	1,1009	1,1472	1,1950	1,2471	1,3074	1,3704

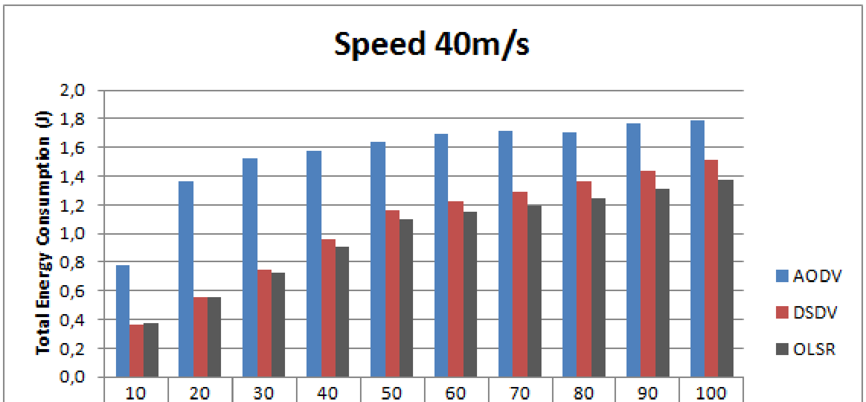
Number of Nodes

Figure 188 Total Energy Consumption vs. Number of Nodes, speed 30m/s.



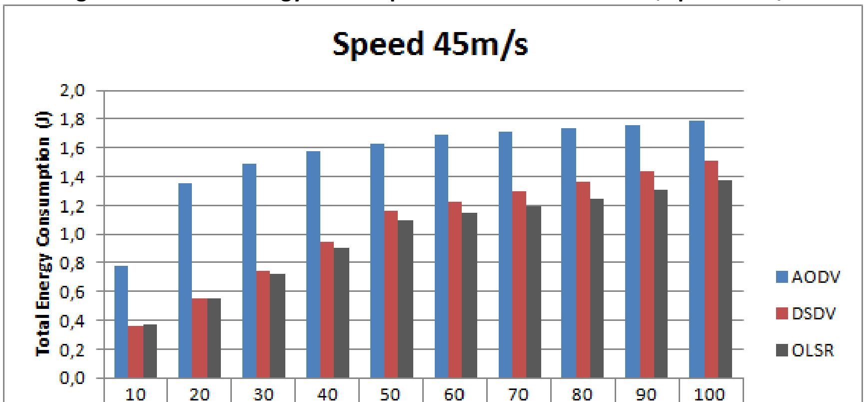
	10	20	30	40	50	60	70	80	90	100
AODV	0,7627	1,3690	1,5040	1,5529	1,6512	1,7014	1,7191	1,7248	1,7678	1,7934
DSDV	0,3629	0,5499	0,7495	0,9548	1,1645	1,2284	1,2954	1,3634	1,4354	1,5086
OLSR	0,3786	0,5503	0,7278	0,9112	1,1007	1,1471	1,1950	1,2479	1,3074	1,3705

Figure 189 Total Energy Consumption vs. Number of Nodes, speed 35m/s.



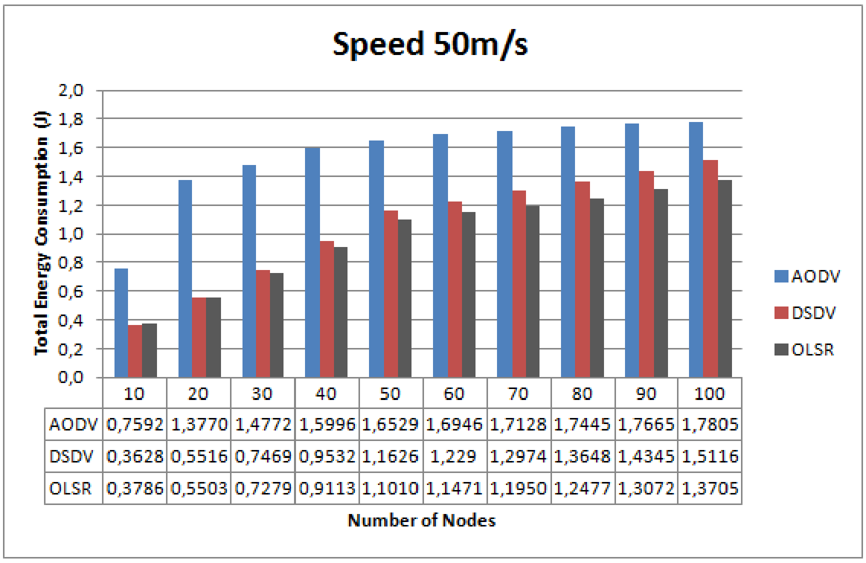
	10	20	30	40	50	60	70	80	90	100
AODV	0,7746	1,3632	1,5277	1,5800	1,6407	1,6956	1,7134	1,7067	1,7637	1,7885
DSDV	0,3629	0,5516	0,7496	0,9561	1,165	1,2268	1,293	1,3636	1,4367	1,5089
OLSR	0,3786	0,5503	0,7279	0,9112	1,1009	1,1471	1,1950	1,2471	1,3075	1,3707

Figure 190 Total Energy Consumption vs. Number of Nodes, speed 40m/s.



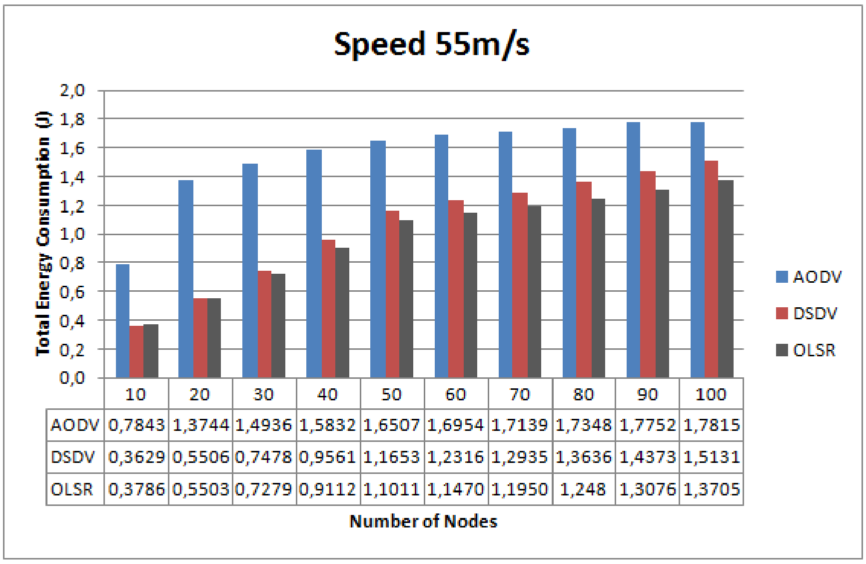
	10	20	30	40	50	60	70	80	90	100
AODV	0,7732	1,3553	1,4862	1,5790	1,6280	1,6931	1,7155	1,7301	1,7545	1,7871
DSDV	0,3629	0,5504	0,7473	0,9533	1,1645	1,229	1,2944	1,3617	1,4377	1,5086
OLSR	0,3786	0,5503	0,7278	0,9112	1,1007	1,1470	1,1951	1,247	1,3071	1,3706

Figure 191 Total Energy Consumption vs. Number of Nodes, speed 45m/s.



	10	20	30	40	50	60	70	80	90	100
AODV	0,7592	1,3770	1,4772	1,5996	1,6529	1,6946	1,7128	1,7445	1,7665	1,7805
DSDV	0,3628	0,5516	0,7469	0,9532	1,1626	1,229	1,2974	1,3648	1,4345	1,5116
OLSR	0,3786	0,5503	0,7279	0,9113	1,1010	1,1471	1,1950	1,2477	1,3072	1,3705

Figure 192 Total Energy Consumption vs. Number of Nodes, speed 50m/s.



	10	20	30	40	50	60	70	80	90	100
AODV	0,7843	1,3744	1,4936	1,5832	1,6507	1,6954	1,7139	1,7348	1,7752	1,7815
DSDV	0,3629	0,5506	0,7478	0,9561	1,1653	1,2316	1,2935	1,3636	1,4373	1,5131
OLSR	0,3786	0,5503	0,7279	0,9112	1,1011	1,1470	1,1950	1,248	1,3076	1,3705

Figure 193 Total Energy Consumption vs. Number of Nodes, speed 55m/s.

The figure that follows (Figure 194) summarizes the results and presents a comparison between the three routing protocols based on the average speed and the number of nodes.

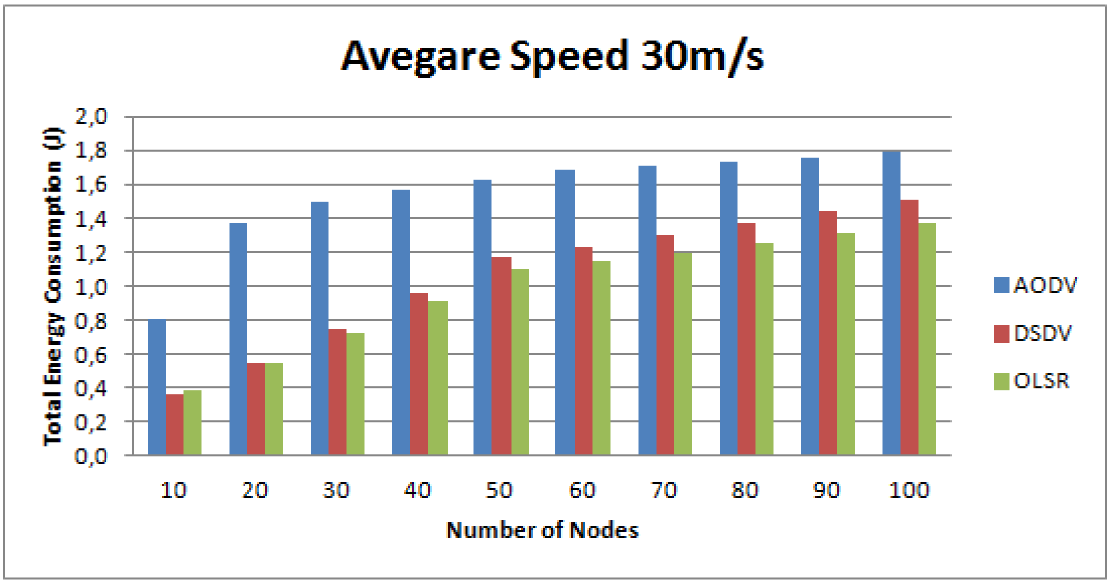


Figure 194 Total Energy Consumption vs. Number of Nodes, Average Speed 30m/s.

Conclusions

1. Packet Delivery Ratio (PDR)

In the table that follows (Tables 28) we can see which routing protocol has in every case the best performance considering the PDR.

Table 29 Performance Comparison

		Speed										
		5	10	15	20	25	30	35	40	45	50	55
Number of Nodes	10	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	DSDV	OLSR	OLSR	OLSR	OLSR
	20	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	30	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	40	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	50	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	60	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	70	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	80	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	90	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	100	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR

AODV has never the best PDR performance. The other two protocols always outperform AODV. OLSR almost always has the best performance as it has 99.09% the best PDR. DSDV has only in one case the best PDR performance (0.9%) and it comes second. Therefore, the best choice for this scenario based on the PDR is OLSR.

2. Average Delay

In the table that follows (Tables 29) we can see which routing protocol has in every case the best performance considering the average delay.

Table 30 Performance Comparison

		Speed										
		5	10	15	20	25	30	35	40	45	50	55
Number of Nodes	10	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	20	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	30	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	40	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	50	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	60	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV
	70	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	DSDV	OLSR
	80	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	90	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	100	OLSR	DSDV	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR

AODV achieves always the highest delay which is why AODV is not preferable compared to the other two protocols. We have concluded that in most of the cases OLSR outperforms DSDV with 89.09% and therefore, the best choice for this scenario based on the average delay is OLSR. Only when the number of nodes is 60 the best choice is DSDV.

3. Throughput

In the table that follows (Tables 30) we can see which routing protocol has in every case the best performance considering the throughput.

Table 31 Performance Comparison

		Speed										
		5	10	15	20	25	30	35	40	45	50	55
Number of Nodes	10	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	DSDV	OLSR	OLSR	OLSR	OLSR
	20	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	30	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	40	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	50	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	60	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	70	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	80	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	90	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	100	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR

AODV has never the highest throughput. The other two protocols always outperform AODV. OLSR almost always has the best performance as it has 99.09% the best throughput. DSDV has only in one case the best throughput performance (0.9%) and it comes second. Therefore, the best choice for this scenario based on the throughput is OLSR.

4. Total Energy Consumption

In the table that follows (Tables 31) we can see which routing protocol has in every case the best performance considering the total energy consumption.

Table 32 Performance Comparison

		Speed										
		5	10	15	20	25	30	35	40	45	50	55
Number of Nodes	10	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV
	20	OLSR	OLSR	OLSR	OLSR	OLSR	DSDV	DSDV	DSDV	OLSR	OLSR	OLSR
	30	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	40	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	50	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	60	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	70	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	80	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	90	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	100	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR

In most of the cases OLSR outperforms DSDV with 88.18%, only when the number of nodes is 10 the best choice is DSDV as it has the lowest total energy consumption (only 11.8% of the cases). They both outperform AODV. Therefore, the best choice for this scenario based on the total energy consumption is OLSR.

4.3.3 Archaeological Site Scenario

The simulation results for the Archaeological Site Scenario are presented on the four sections that follow. In each section we investigate the behavior of each protocol and we also compare them based on different values of number of nodes and number of packets sent by each node. In the end of each section we compare the three routing protocols.

4.3.3.1 Packet Delivery Ratio

In this section we investigate the communication reliability for different group sizes and number of packets sent by the sender. We measure the PDR. PDR is the percentage from the send messages that was actually delivered. It represents how reliable the communication is. The higher the PDR, the better the communication reliability is. We use six different group sizes (5, 10, 15, 20, 25, 30 nodes) and ten different number of packets (50 – 500 packets with 50 packets step). The highest the number of packets being sent, the heavier the traffic is.

- AODV

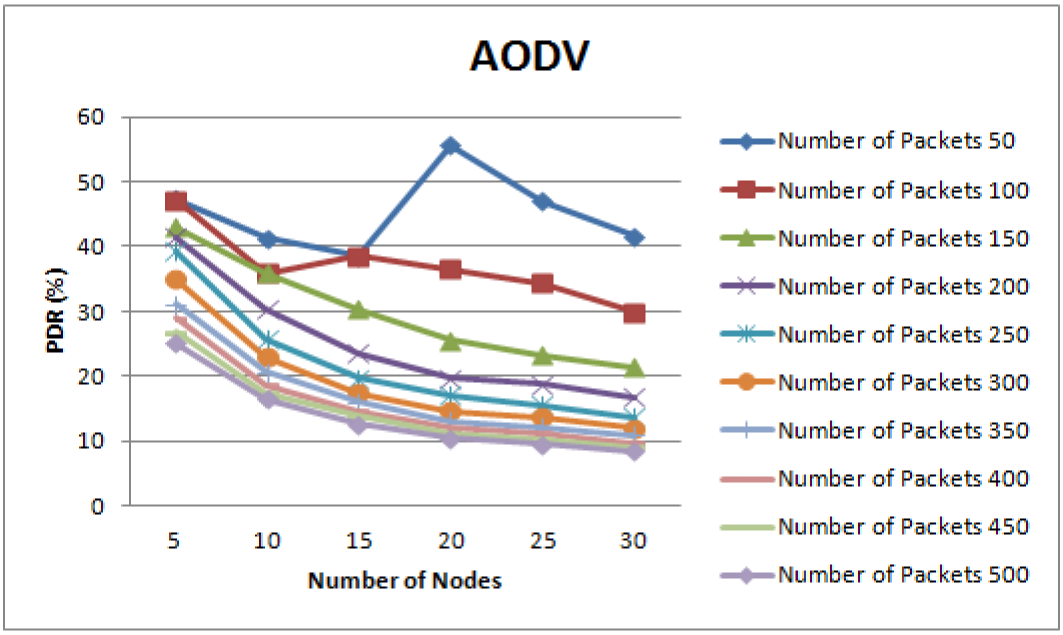


Figure 195 PDR vs. Number of Nodes, AODV Routing Protocol

As we can observe in Figure 195 PDR is affected by the number of nodes and the by the number of packets being sent. As expected, the heavier the traffic, the lower the PDR is. Also, as the number of nodes is being increased, the PDR is decreased. As we can see, even for the best possible circumstances (number of nodes 5 and number of packets 50) the PDR does not overcome 60%. The packets loss is not acceptable considering the fact that in our scenario the tour guide sends video to the tourists and with these levels of PDR the outcome will not be acceptable.

- DSDV

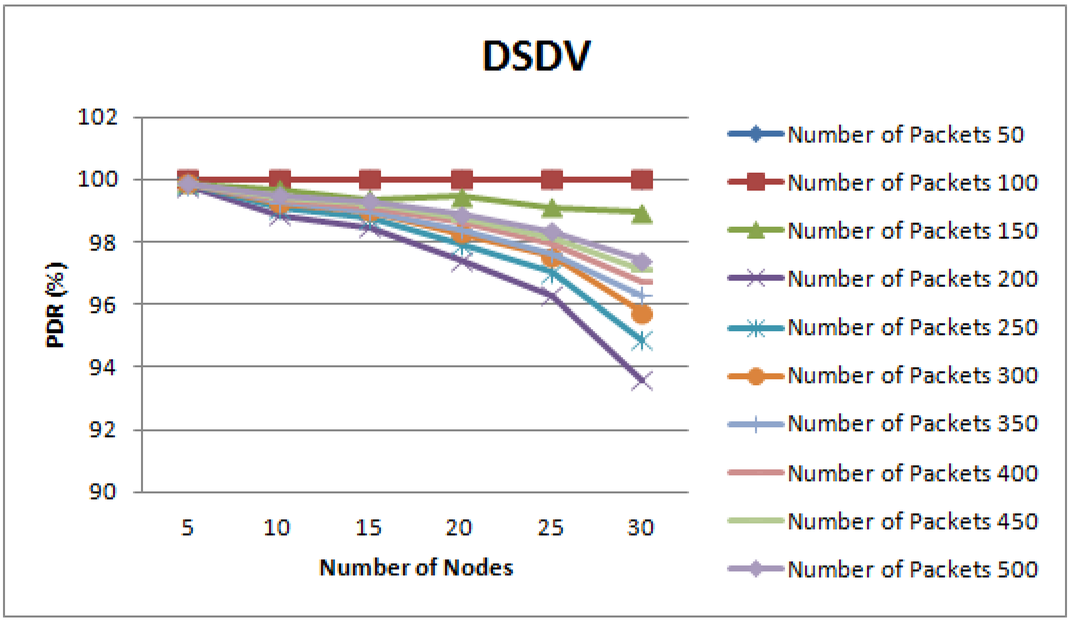


Figure 196 PDR vs. Number of Nodes, DSDV Routing Protocol

In Figure 196 we can see that DSDV offers very good results. We can also observe a certain pattern. More specifically, when the number of packets is 50 or 100, the PDR is 100%. After 100 packets we have a decrease which seems to reach its peak when we have 200 packets. This is clearer in Figure 197 that follows. After that the PDR increases without reaching though 100%. As expected, the highest the number of nodes, the lowest the PDR is, but in the worst case it reaches 93.6%.

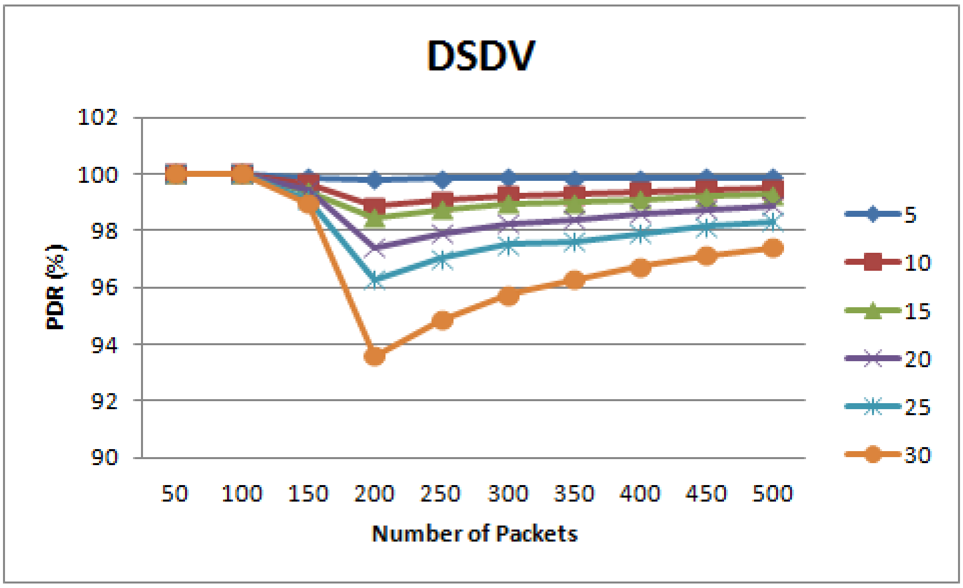


Figure 197 PDR vs. Number of Packets, DSDV Routing Protocol

- OLSR

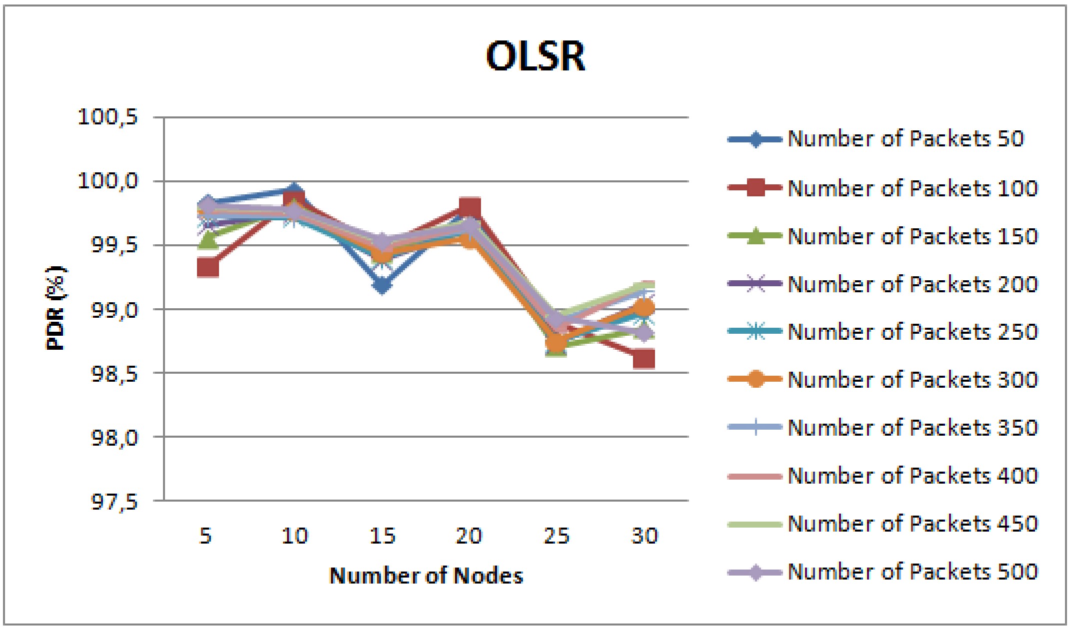


Figure 198 PDR vs. Number of Nodes, OLSR Routing Protocol

OLSR does not have like the previous scenarios the same results with DSDV. As we can see in Figure 198 it has PDR more than 98.5% in every case and also we cannot distinguish a specific pattern in the results. It is not certain that for lower number of nodes we will have better PDR. Nevertheless, for small groups (number of nodes 15 or less) OLSR achieves PDR more than 99%.

4.3.3.2 Average Delay

In this section we investigate the average delay time. The delay affects the quality of the communication and so it is an important factor for the evaluation of the protocols.

- AODV

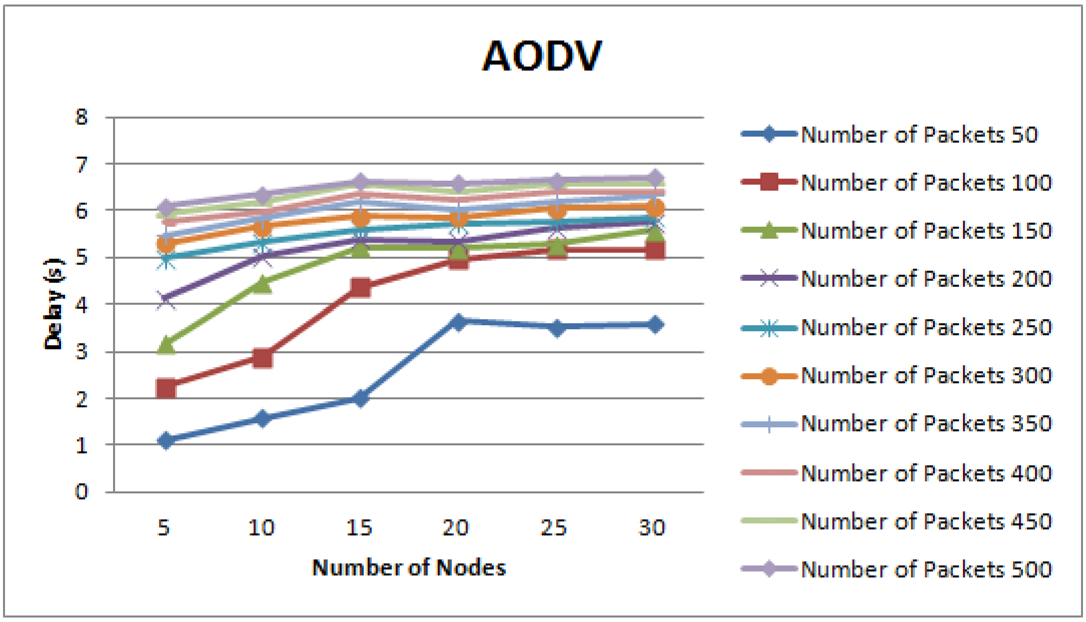


Figure 199 Delay vs. Number of Nodes, AODV Routing Protocol

As we can observe in Figure 199 the higher the number of packets, the higher the average delay is. The delay is also affected by the number of nodes. As we can see, until the number of nodes become 20 the delay is increased with a higher rate than after the nodes are 20 (this is more intense for small number of packets). The delay in any case does not overcome the 7 seconds, and for the best possible circumstances is always higher than 1 second.

- DSDV

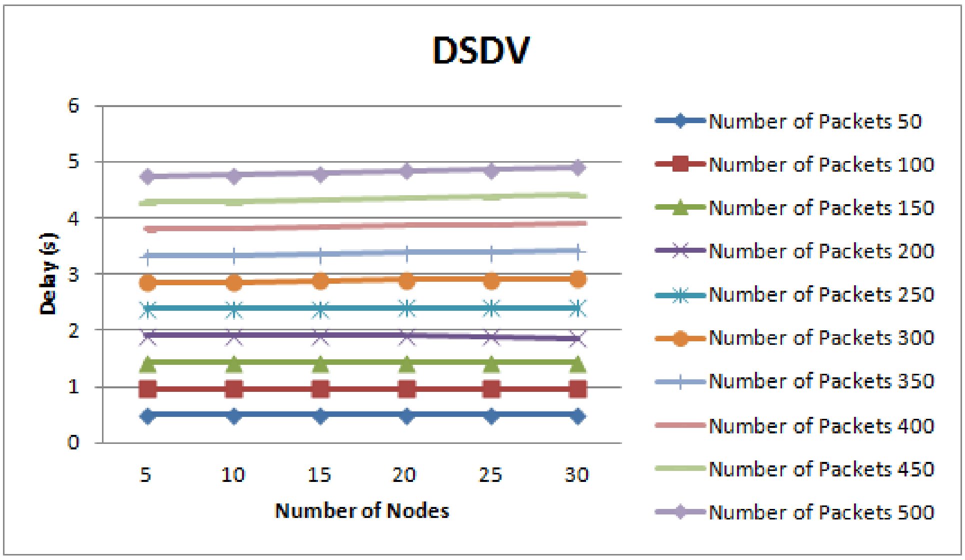


Figure 200 Delay vs. Number of Nodes, DSDV Routing Protocol

As we can observe in Figure 200 the average delay is not affected by the number of nodes. DSDV has a certain delay for a given number of packets no matter how many the nodes are. This is a very interesting result and also very good for the implementation of DSDV. As expected, the higher the number of packets, the higher the delay is.

- OLSR

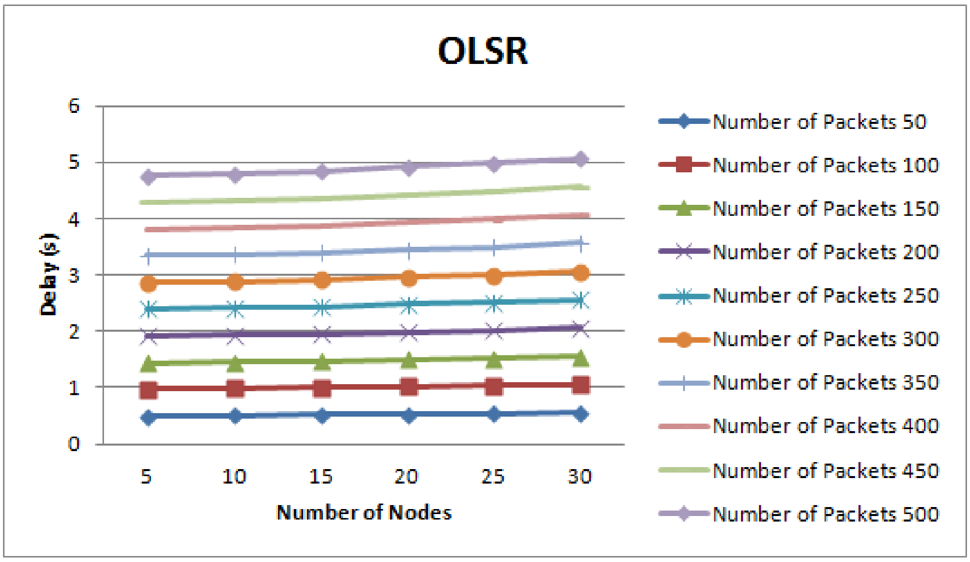


Figure 201 Delay vs. Number of Nodes, OLSR Routing Protocol

As we can observe in Figure 201 the average delay is not affected dramatically by the number of nodes. OLSR has a certain delay for a given number of packets that it is slightly increased as the number of nodes is increased. This is more obvious for higher number of packets. As expected, the higher the number of packets, the higher the delay is.

4.3.3.3 Throughput

In this section we investigate the throughput for different number of packets and number of nodes. The throughput is an important factor as well which affects the communication and the network performance.

- AODV

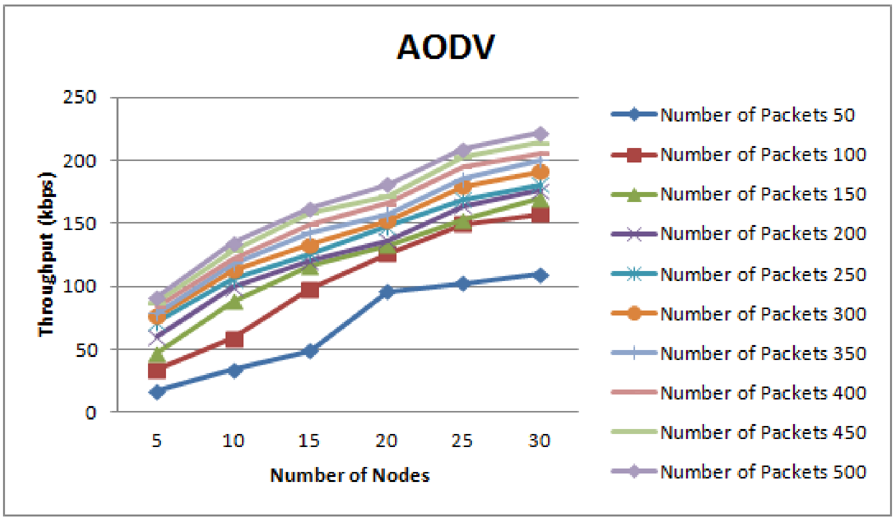


Figure 202 Throughput vs. Number of Nodes, AODV Routing Protocol

As expected (Figure 202), the higher the number of packets, the higher the throughput is. The throughput is also affected by the number of nodes. The throughput does not reach high levels due to the low levels of PDR.

- DSDV

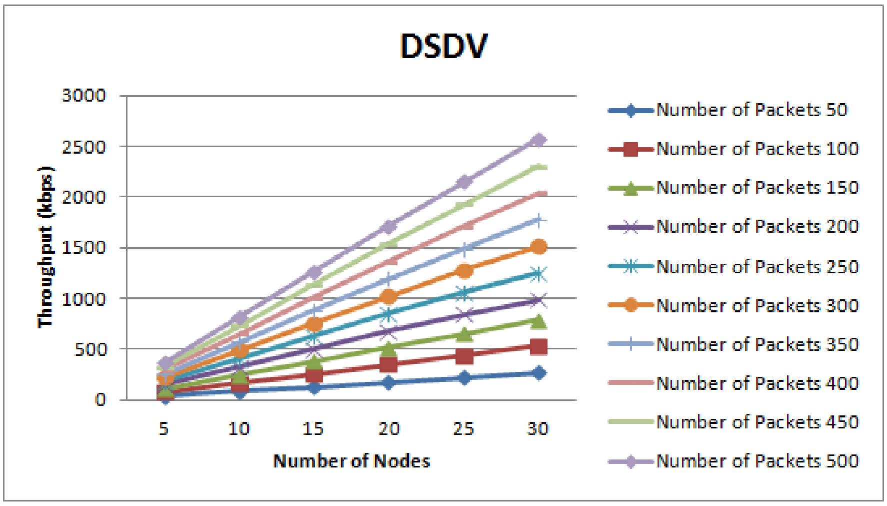


Figure 203 Throughput vs. Number of Nodes, DSDV Routing Protocol

As we can see in Figure 203 the throughput is increased linearly as the number of nodes is increased. The higher the number of packets, the higher the throughput is.

- OLSR

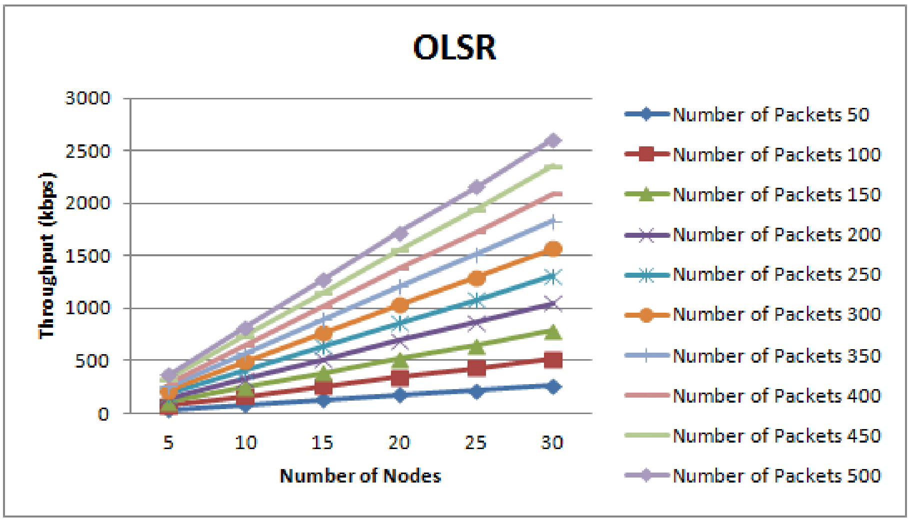


Figure 204 Throughput vs. Number of Nodes, OLSR Routing Protocol

In Figure 204 we can observe that OLSR has similar results with DSDV. In this protocol the throughput is increased linearly too. The higher the number of packets, the higher the throughput is.

4.3.3.4 Total Energy Consumption

The final perfume metric which is being studied is energy. Energy is a very important factor in routing protocols for MANETS, because in our scenario portable devices may not have the chance to be recharged and so the total energy consumption should be reduced as far it can be.

- AODV

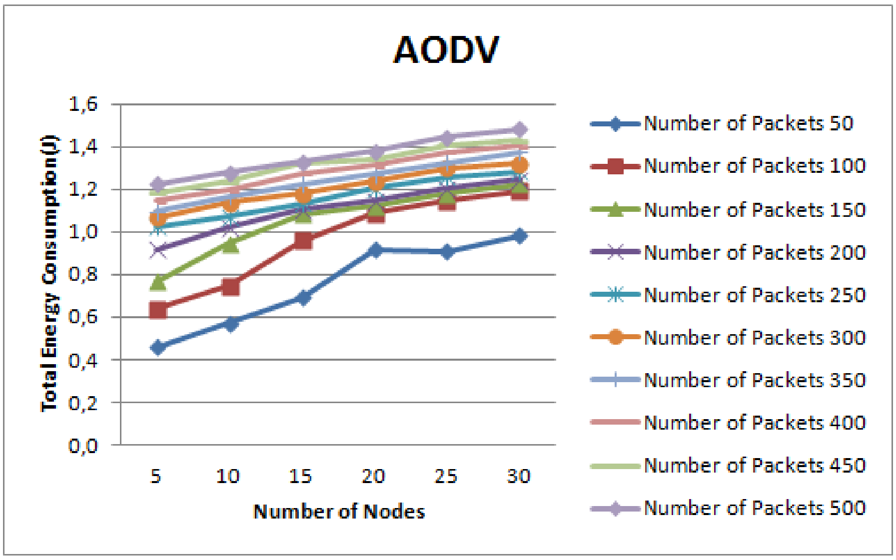


Figure 205 Total Energy Consumption vs. Number of Nodes, AODV Routing Protocol

As we can see in Figure 205, the higher the number of packets and nodes, the higher the total energy consumption is. Total energy consumption overcomes 0.4J under the best circumstances and the highest energy consumption is lower than 1.6J.

- DSDV

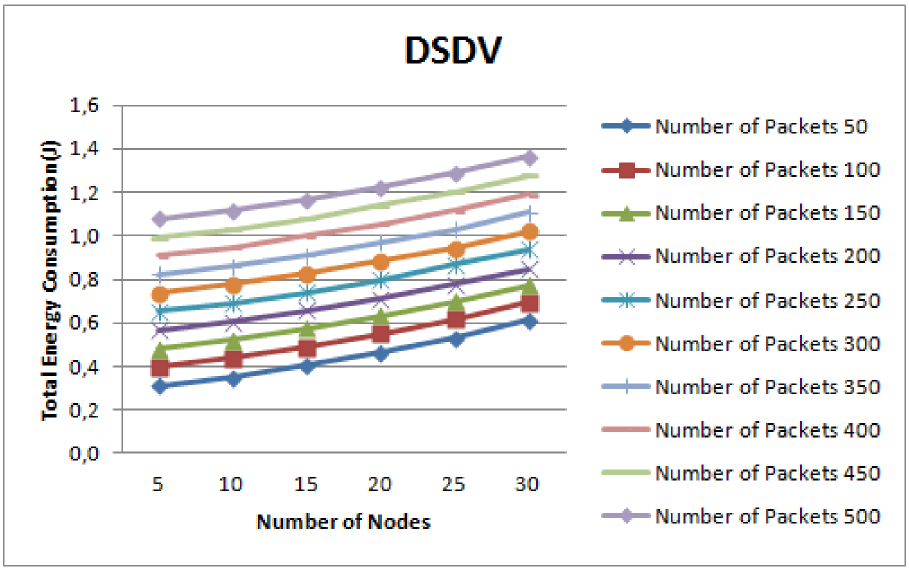


Figure 206 Total Energy Consumption vs. Number of Nodes, DSDV Routing Protocol

In Figure 206 we can observe that the total energy consumption is increased relatively linearly as the number of nodes increased and as expected, the higher the number of packets, the higher total energy consumption is. Total energy consumption overcomes 0.2J under the best circumstances and the highest energy consumption is lower than 1.4J.

- OLSR

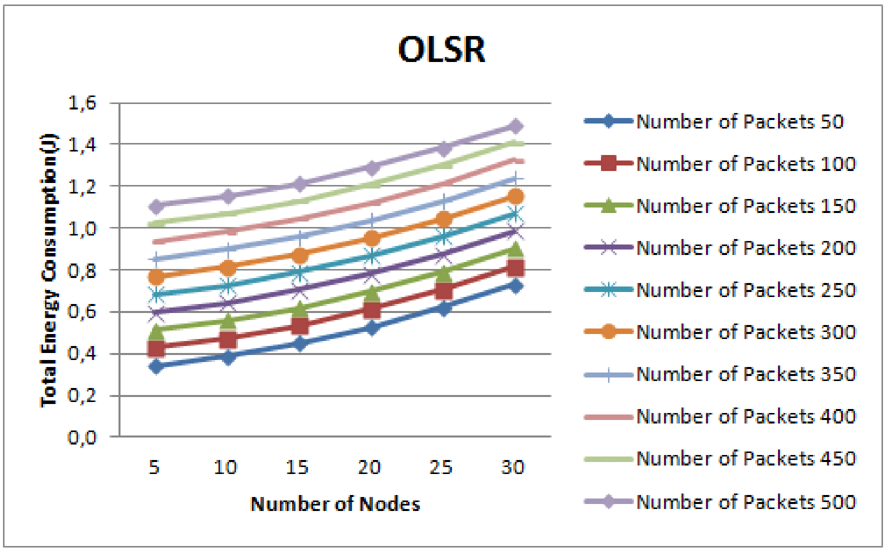


Figure 207 Total Energy Consumption vs. Number of Nodes, OLSR Routing Protocol

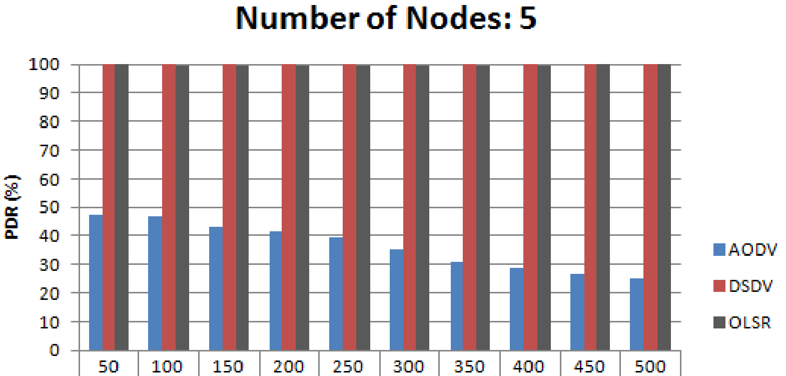
In Figure 207 we can observe that OLSR results are similar to DSDV. The total energy consumption is increased relatively linearly as the number of nodes increased and as expected, the higher the number of packets, the higher total energy consumption is. Total energy consumption overcomes 0.2J under the best circumstances and the highest energy consumption is lower than 1.6J.

4.3.3.5 Comparison of Routing Protocols

In this section we compare the three routing protocols. The diagrams that follow will provide a better understanding of the results. In the end of this section will be a summary of this comparison.

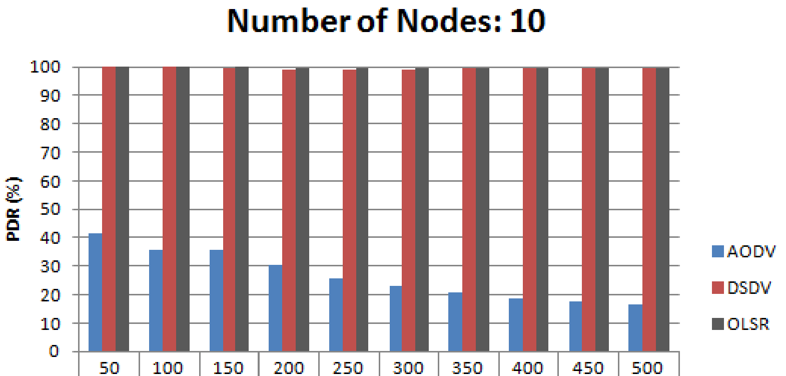
Packet Delivery Ratio (PDR)

In this section we compare the three routing protocols based on the PDR they achieve. The results are categorized based on the different number of nodes. As we can observe in Figures 208-213 in every case DSDV achieves better results for small number of nodes or for small number of packets. More specifically, DSDV outperforms OLSR and AODV when the number of nodes is 5, when the number of packets is 50 or 100 and when the number of packets is 150 and the number of nodes is 25 or 35 at the same time (in total 36.6% of all cases). In every other case (63.4%) OLSR outperforms DSDV. They both outperform AODV.



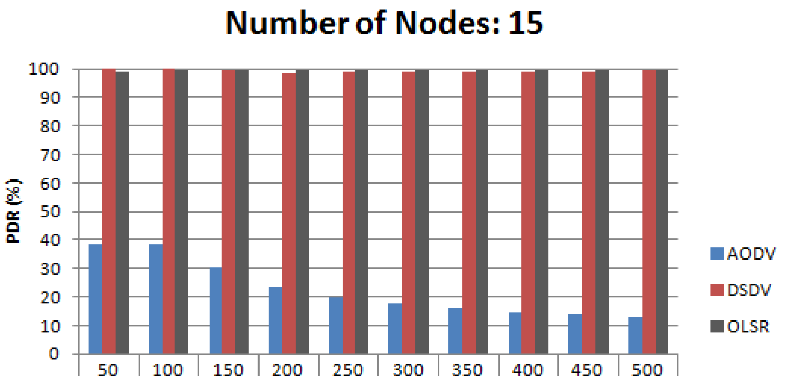
	50	100	150	200	250	300	350	400	450	500
AODV	47,33	47,16	43,06	41,38	39,33	34,94	31,14	28,96	26,81	25,15
DSDV	100,0	100,0	99,88	99,79	99,83	99,86	99,83	99,85	99,87	99,88
OLSR	99,83	99,33	99,56	99,66	99,73	99,78	99,74	99,77	99,80	99,81

Figure 208 PDR vs. Number of Packets, Number of Nodes: 5



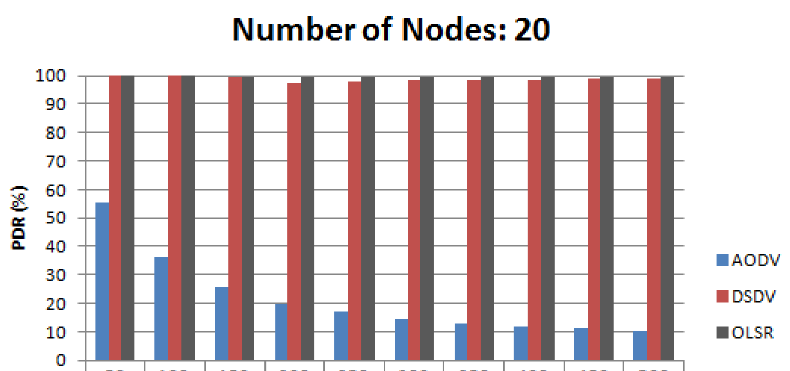
	50	100	150	200	250	300	350	400	450	500
AODV	41,33	35,93	35,95	30,30	25,75	23,01	20,63	18,66	17,44	16,41
DSDV	100,0	100,0	99,68	98,85	99,08	99,23	99,28	99,37	99,44	99,50
OLSR	99,93	99,85	99,80	99,76	99,72	99,77	99,72	99,74	99,77	99,77

Figure 209 PDR vs. Number of Packets, Number of Nodes: 10



	50	100	150	200	250	300	350	400	450	500
AODV	38,48	38,45	30,33	23,58	19,75	17,43	16,02	14,68	13,80	12,68
DSDV	100,0	100,0	99,37	98,45	98,76	98,97	98,98	99,11	99,21	99,29
OLSR	99,19	99,48	99,44	99,38	99,40	99,44	99,52	99,48	99,53	99,54

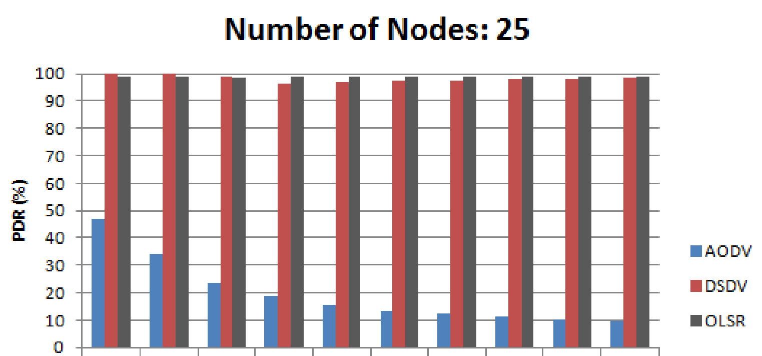
Figure 210 PDR vs. Number of Packets, Number of Nodes: 15



	50	100	150	200	250	300	350	400	450	500
AODV	55,61	36,47	25,49	19,70	17,07	14,64	13,00	11,99	11,04	10,45
DSDV	100,0	100,0	99,46	97,39	97,92	98,26	98,41	98,61	98,76	98,89
OLSR	99,79	99,81	99,66	99,64	99,62	99,56	99,62	99,66	99,68	99,66

Number of Packets

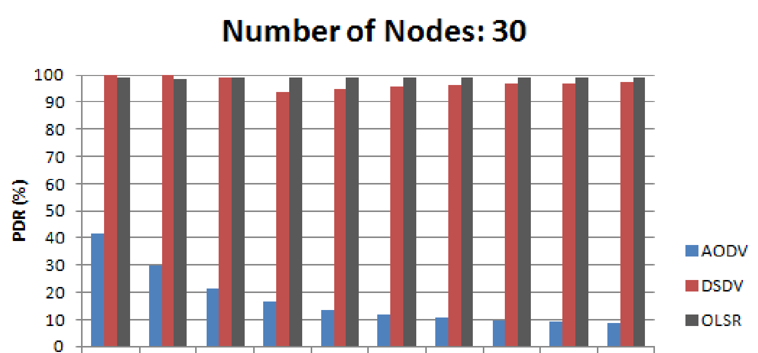
Figure 211 PDR vs. Number of Packets, Number of Nodes: 20



	50	100	150	200	250	300	350	400	450	500
AODV	47,00	34,28	23,31	18,74	15,48	13,68	12,12	11,18	10,32	9,56
DSDV	100,0	100,0	99,09	96,29	97,03	97,53	97,62	97,92	98,15	98,33
OLSR	98,75	98,89	98,70	98,73	98,74	98,75	98,90	98,84	98,95	98,93

Number of Packets

Figure 212 PDR vs. Number of Packets, Number of Nodes: 25



	50	100	150	200	250	300	350	400	450	500
AODV	41,59	29,83	21,44	16,70	13,68	12,05	10,85	9,74	9,02	8,41
DSDV	100,0	100,0	98,94	93,60	94,88	95,74	96,28	96,75	97,11	97,40
OLSR	98,99	98,62	98,84	99,05	98,97	99,02	99,14	99,20	99,19	98,81

Number of Packets

Figure 213 PDR vs. Number of Packets, Number of Nodes: 30

The figure that follows (Figure 214) summarizes the results and presents a comparison between the three routing protocols based on the average number of nodes and the number of packets.

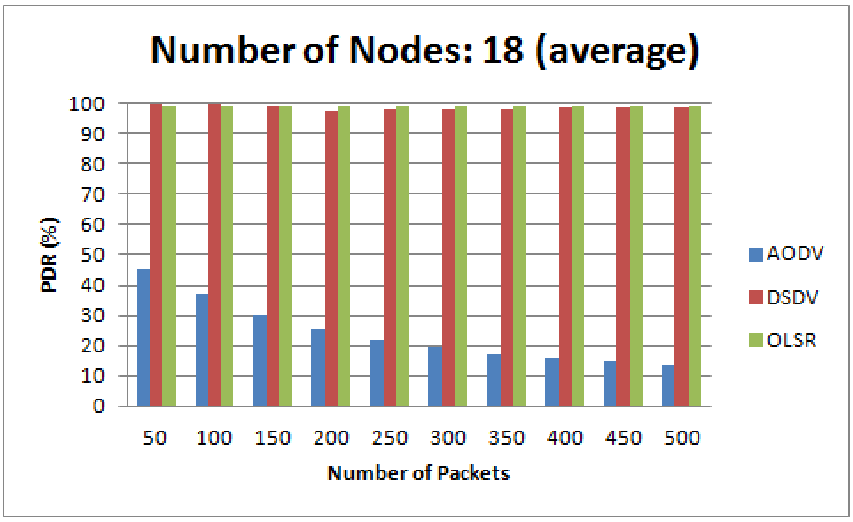
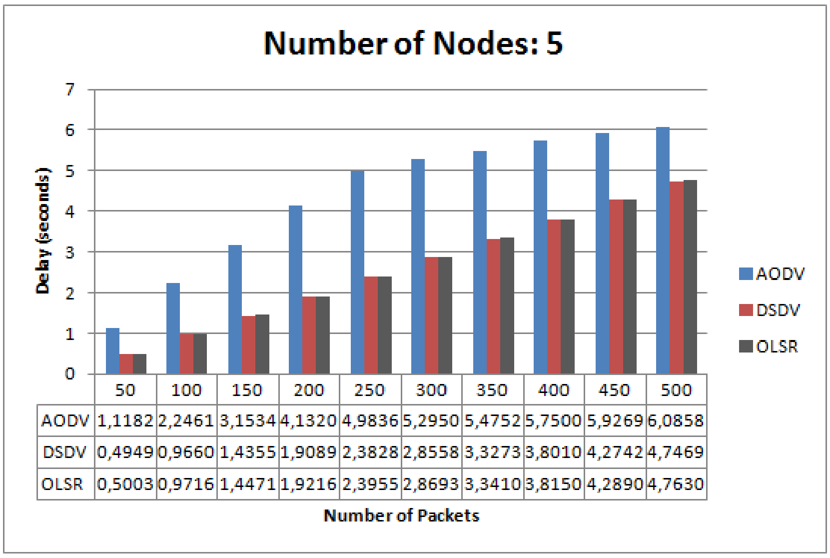


Figure 214 PDR vs. Number of Packets, Number of Nodes: 18 (average)

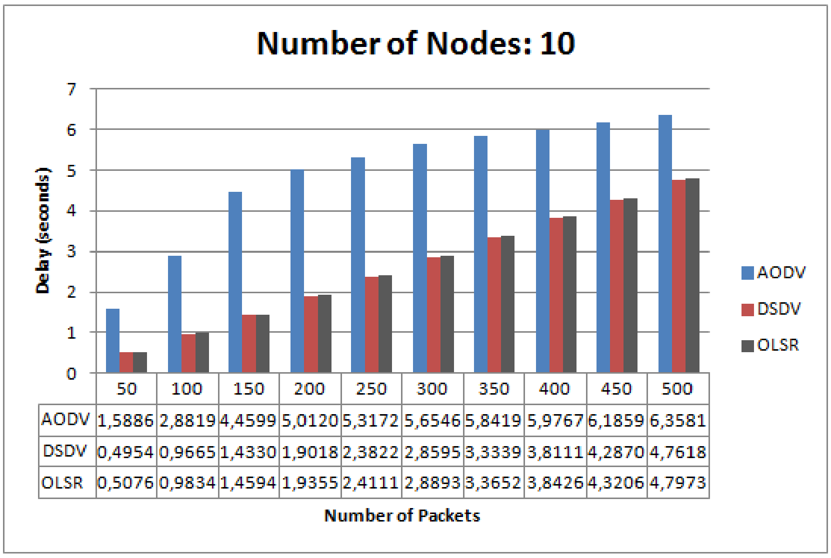
Average Delay

In this section we compare the three routing protocols based on the average delay they achieve. The results are categorized based on the different number of nodes. As we can observe in Figures 215-220 AODV achieves always the highest delay which is why AODV is not preferable compared to the other two protocols. We can also conclude that DSDV outperforms OLSR in every case. DSDV achieves lower delay. As the number of nodes is increased, the difference in the delay between DSDV and OLSR is increased.



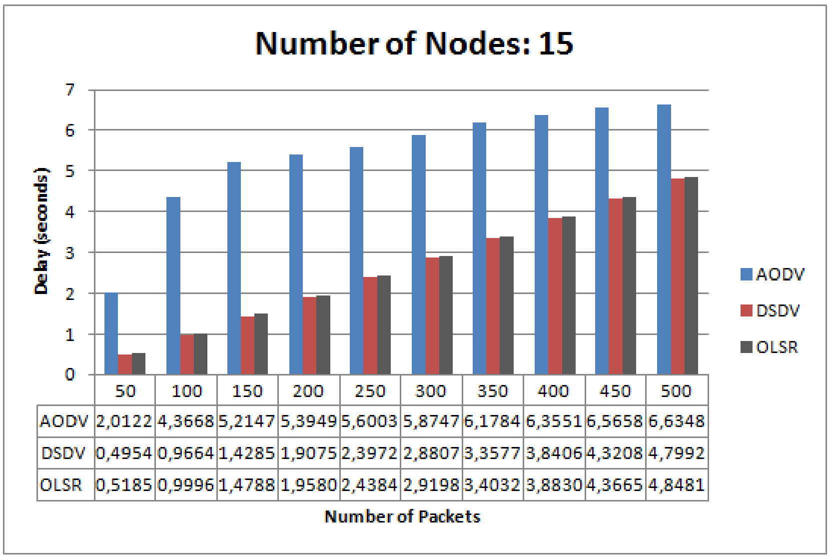
	50	100	150	200	250	300	350	400	450	500
AODV	1,1182	2,2461	3,1534	4,1320	4,9836	5,2950	5,4752	5,7500	5,9269	6,0858
DSDV	0,4949	0,9660	1,4355	1,9089	2,3828	2,8558	3,3273	3,8010	4,2742	4,7469
OLSR	0,5003	0,9716	1,4471	1,9216	2,3955	2,8693	3,3410	3,8150	4,2890	4,7630

Figure 215 Delay vs. Number of Packets, Number of Nodes: 5



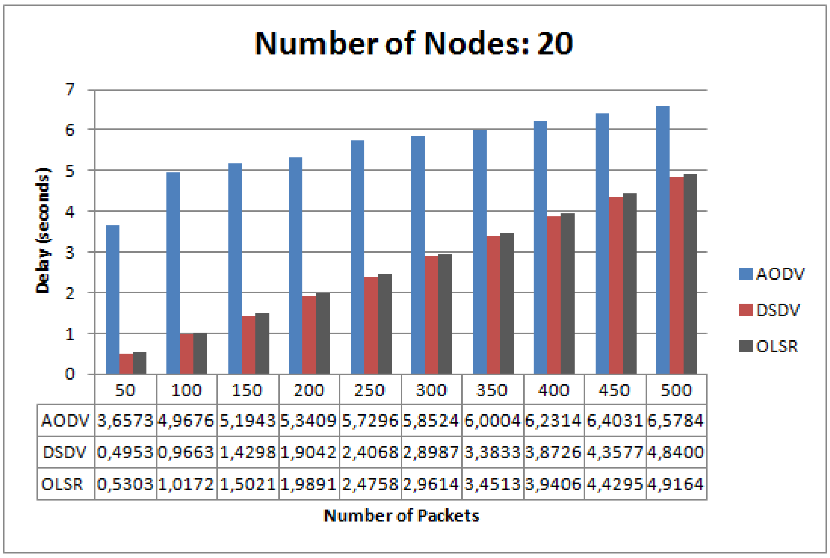
	50	100	150	200	250	300	350	400	450	500
AODV	1,5886	2,8819	4,4599	5,0120	5,3172	5,6546	5,8419	5,9767	6,1859	6,3581
DSDV	0,4954	0,9665	1,4330	1,9018	2,3822	2,8595	3,3339	3,8111	4,2870	4,7618
OLSR	0,5076	0,9834	1,4594	1,9355	2,4111	2,8893	3,3652	3,8426	4,3206	4,7973

Figure 216 Delay vs. Number of Packets, Number of Nodes: 10



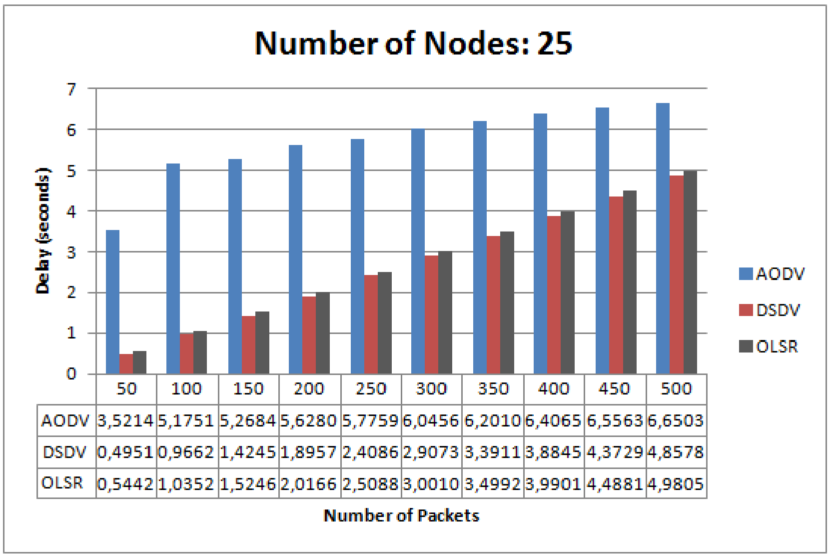
	50	100	150	200	250	300	350	400	450	500
AODV	2,0122	4,3668	5,2147	5,3949	5,6003	5,8747	6,1784	6,3551	6,5658	6,6348
DSDV	0,4954	0,9664	1,4285	1,9075	2,3972	2,8807	3,3577	3,8406	4,3208	4,7992
OLSR	0,5185	0,9996	1,4788	1,9580	2,4384	2,9198	3,4032	3,8830	4,3665	4,8481

Figure 217 Delay vs. Number of Packets, Number of Nodes: 15



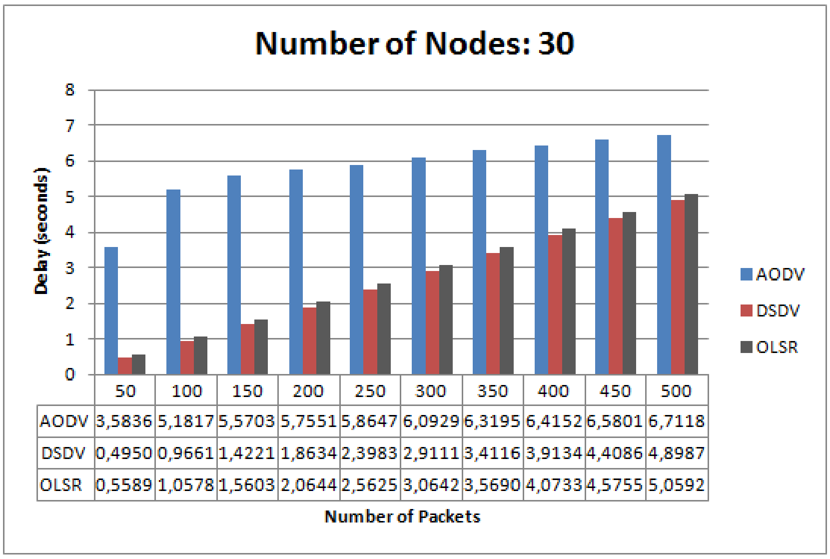
	50	100	150	200	250	300	350	400	450	500
AODV	3,6573	4,9676	5,1943	5,3409	5,7296	5,8524	6,0004	6,2314	6,4031	6,5784
DSDV	0,4953	0,9663	1,4298	1,9042	2,4068	2,8987	3,3833	3,8726	4,3577	4,8400
OLSR	0,5303	1,0172	1,5021	1,9891	2,4758	2,9614	3,4513	3,9406	4,4295	4,9164

Figure 218 Delay vs. Number of Packets, Number of Nodes: 20



	50	100	150	200	250	300	350	400	450	500
AODV	3,5214	5,1751	5,2684	5,6280	5,7759	6,0456	6,2010	6,4065	6,5563	6,6503
DSDV	0,4951	0,9662	1,4245	1,8957	2,4086	2,9073	3,3911	3,8845	4,3729	4,8578
OLSR	0,5442	1,0352	1,5246	2,0166	2,5088	3,0010	3,4992	3,9901	4,4881	4,9805

Figure 219 Delay vs. Number of Packets, Number of Nodes: 25



	50	100	150	200	250	300	350	400	450	500
AODV	3,5836	5,1817	5,5703	5,7551	5,8647	6,0929	6,3195	6,4152	6,5801	6,7118
DSDV	0,4950	0,9661	1,4221	1,8634	2,3983	2,9111	3,4116	3,9134	4,4086	4,8987
OLSR	0,5589	1,0578	1,5603	2,0644	2,5625	3,0642	3,5690	4,0733	4,5755	5,0592

Figure 220 Delay vs. Number of Packets, Number of Nodes: 30

The figure that follows (Figure 221) summarizes the results and presents a comparison between the three routing protocols based on the average number of nodes and the number of packets.

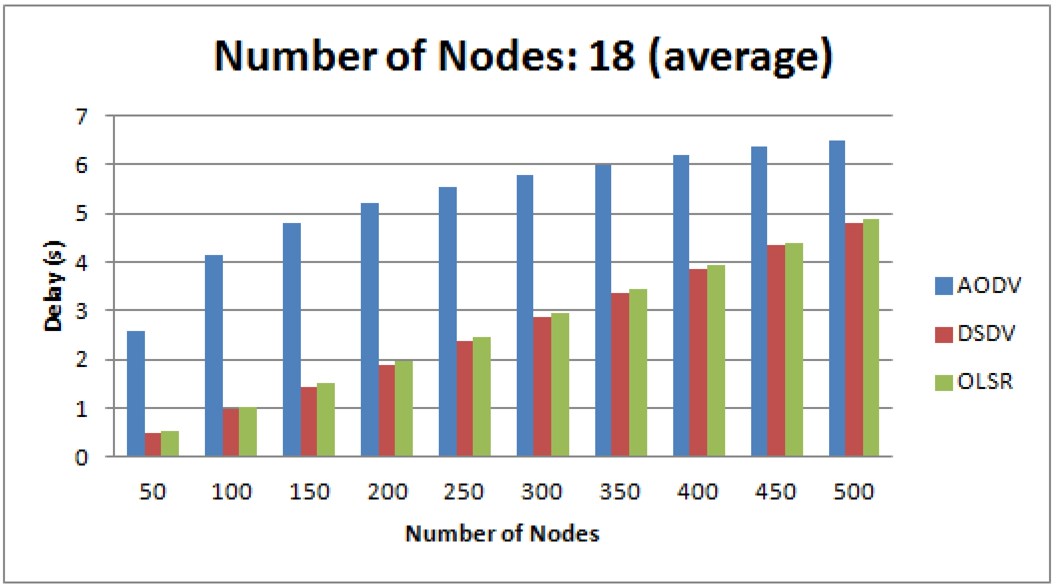
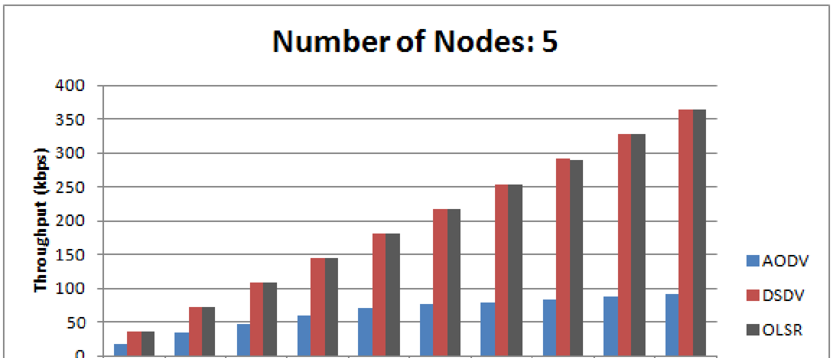


Figure 221 Delay vs. Number of Packets, Number of Nodes: 18 (average)

Throughput

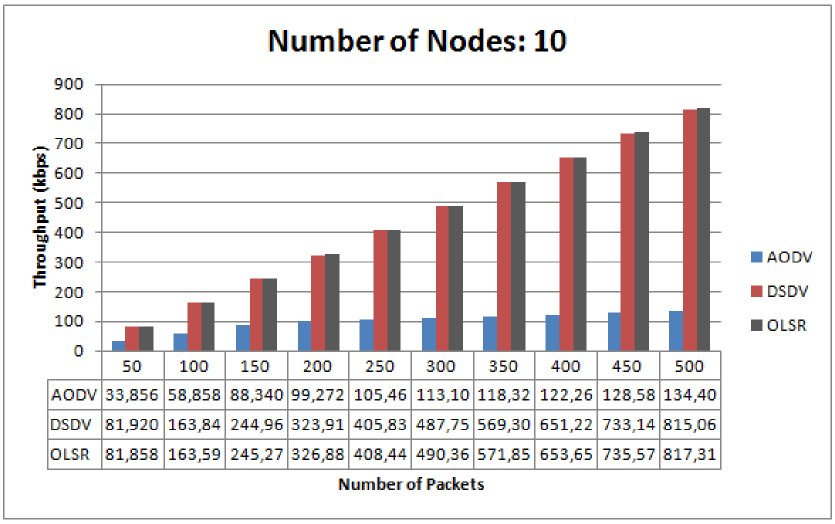
In this section we compare the three routing protocols based on the throughput they achieve. The results are categorized based on the different number of nodes in Figures 222-227. As expected, the performance of the three protocols is the same as with the PDR results. In every case DSDV achieves better results for small number of nodes or for small number of packets. More specifically, DSDV outperforms OLSR and AODV when the number of nodes is 5, when the number of packets is 50 or 100 and when the number of packets is 150 and the number of nodes is 25 or 35 at the same time (in total 36.6% of all cases). In every other case (63.4%) OLSR outperforms DSDV. They both outperform AODV.



	50	100	150	200	250	300	350	400	450	500
AODV	17,229	34,344	47,020	60,250	71,600	76,330	79,368	84,344	87,864	91,565
DSDV	36,408	72,816	109,10	145,32	181,73	218,14	254,43	290,84	327,25	363,66
OLSR	36,346	72,328	108,73	145,14	181,55	217,96	254,18	290,60	327,01	363,41

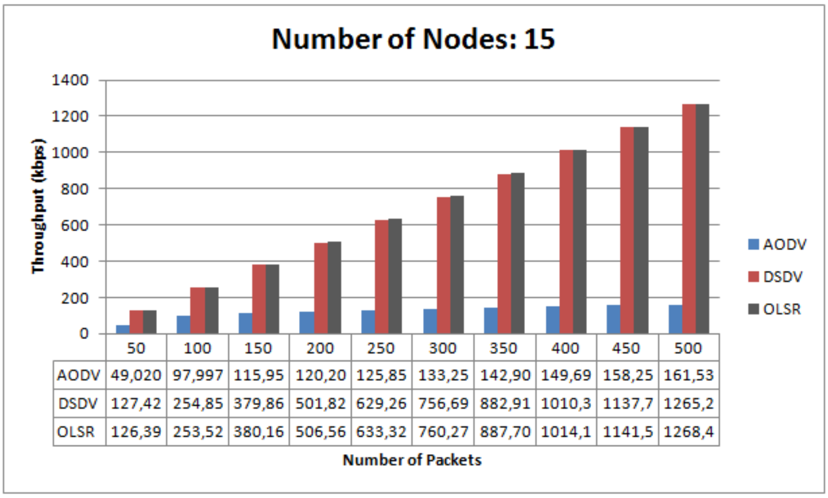
Number of Packets

Figure 222 Throughput vs. Number of Packets, Number of Nodes: 5



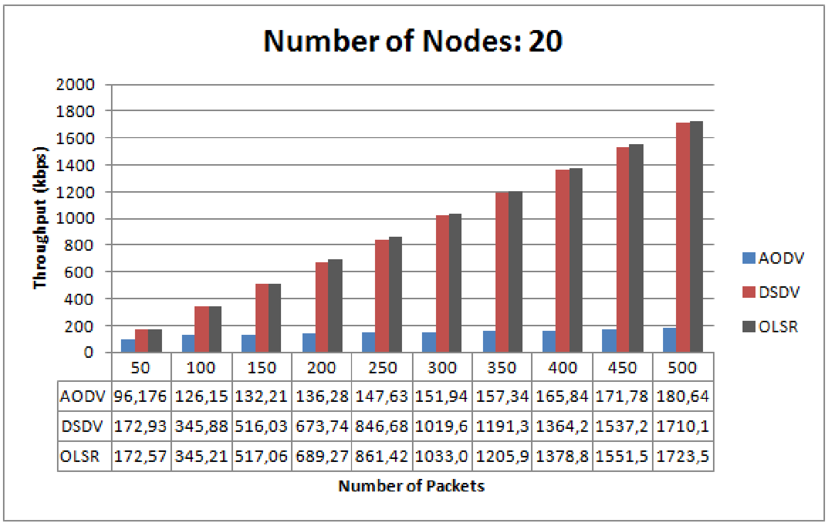
	50	100	150	200	250	300	350	400	450	500
AODV	33,856	58,858	88,340	99,272	105,46	113,10	118,32	122,26	128,58	134,40
DSDV	81,920	163,84	244,96	323,91	405,83	487,75	569,30	651,22	733,14	815,06
OLSR	81,858	163,59	245,27	326,88	408,44	490,36	571,85	653,65	735,57	817,31

Figure 223 Throughput vs. Number of Packets, Number of Nodes: 10



	50	100	150	200	250	300	350	400	450	500
AODV	49,020	97,997	115,95	120,20	125,85	133,25	142,90	149,69	158,25	161,53
DSDV	127,42	254,85	379,86	501,82	629,26	756,69	882,91	1010,3	1137,7	1265,2
OLSR	126,39	253,52	380,16	506,56	633,32	760,27	887,70	1014,1	1141,5	1268,4

Figure 224 Throughput vs. Number of Packets, Number of Nodes: 15



	50	100	150	200	250	300	350	400	450	500
AODV	96,176	126,15	132,21	136,28	147,63	151,94	157,34	165,84	171,78	180,64
DSDV	172,93	345,88	516,03	673,74	846,68	1019,6	1191,3	1364,2	1537,2	1710,1
OLSR	172,57	345,21	517,06	689,27	861,42	1033,0	1205,9	1378,8	1551,5	1723,5

Figure 225 Throughput vs. Number of Packets, Number of Nodes: 20

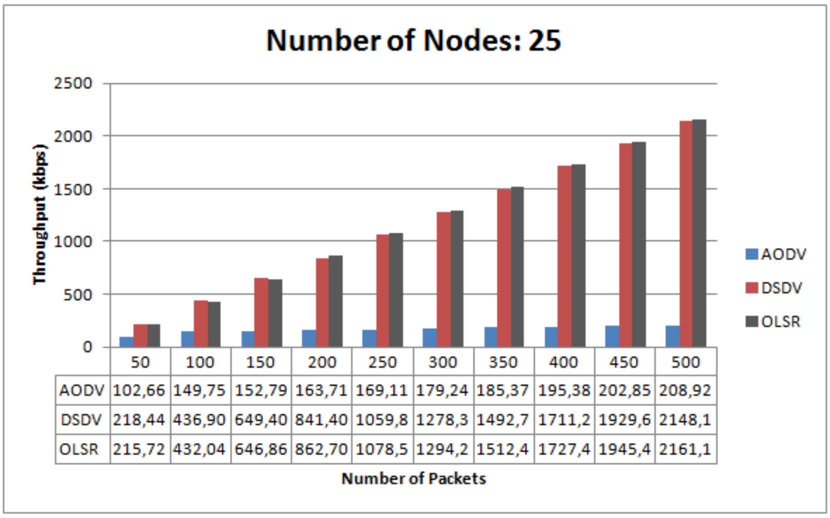


Figure 226 Throughput vs. Number of Packets, Number of Nodes: 25

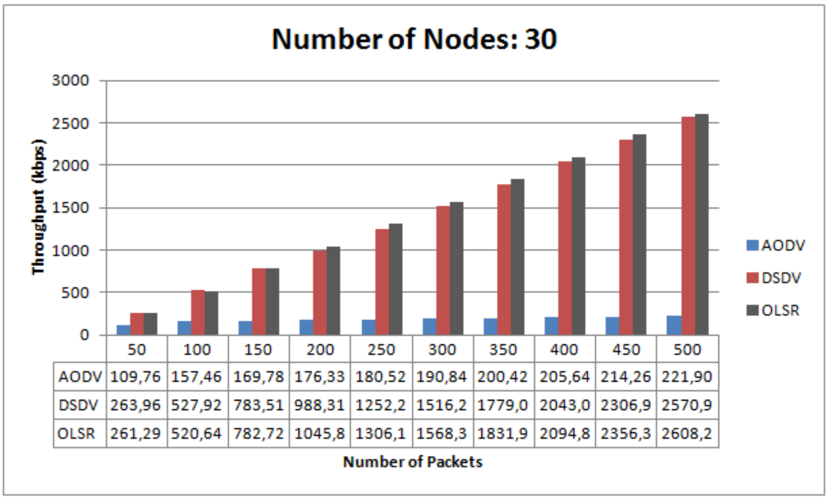


Figure 227 Throughput vs. Number of Packets, Number of Nodes: 30

The figure that follows (Figure 228) summarizes the results and presents a comparison between the three routing protocols based on the average number of nodes and the number of packets.

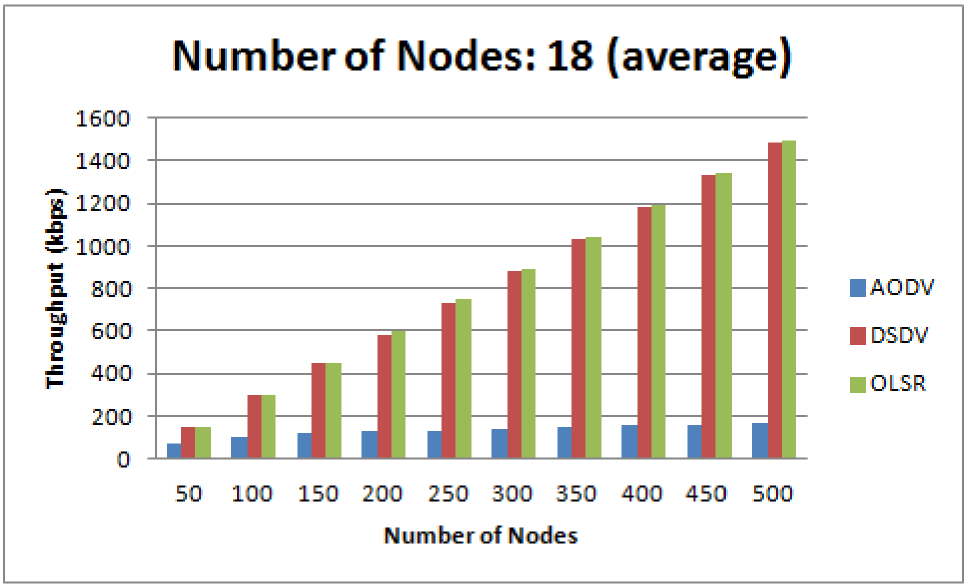
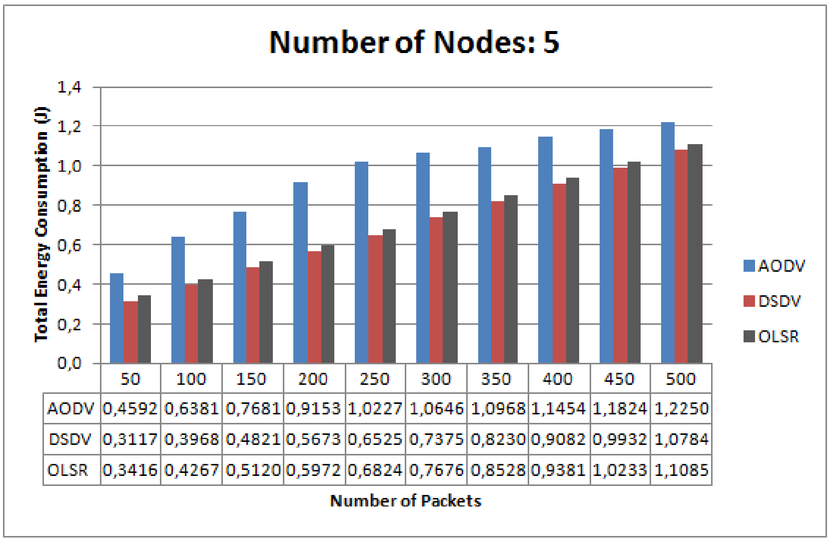


Figure 228 Throughput vs. Number of Packets, Number of Nodes: 18 (average)

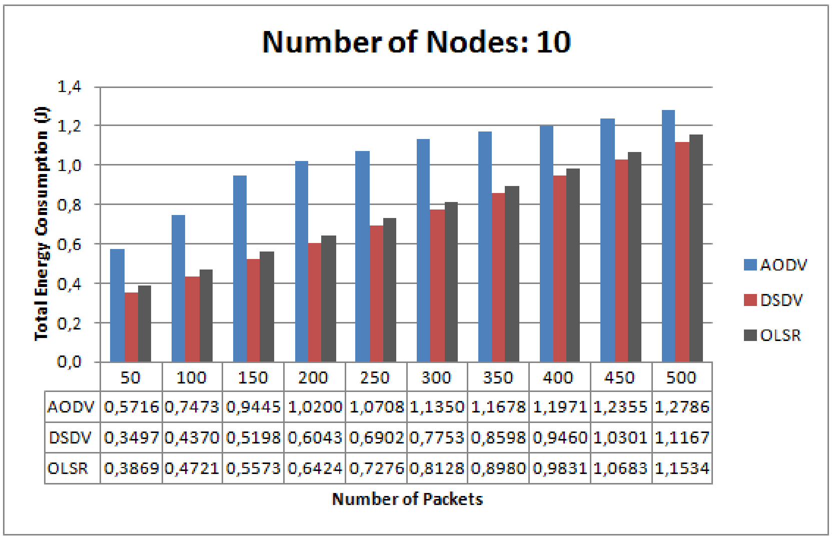
Total Energy Consumption

In this section we compare the three routing protocols based on the total energy consumption they achieve. The results are categorized based on the different number of nodes. As we can observe for Figures 229-234 OLSR outperforms AODV and DSDV outperforms both OLSR and AODV in every case. Therefore, DSDV is the most preferable routing protocol for this scenario considering the energy consumption.



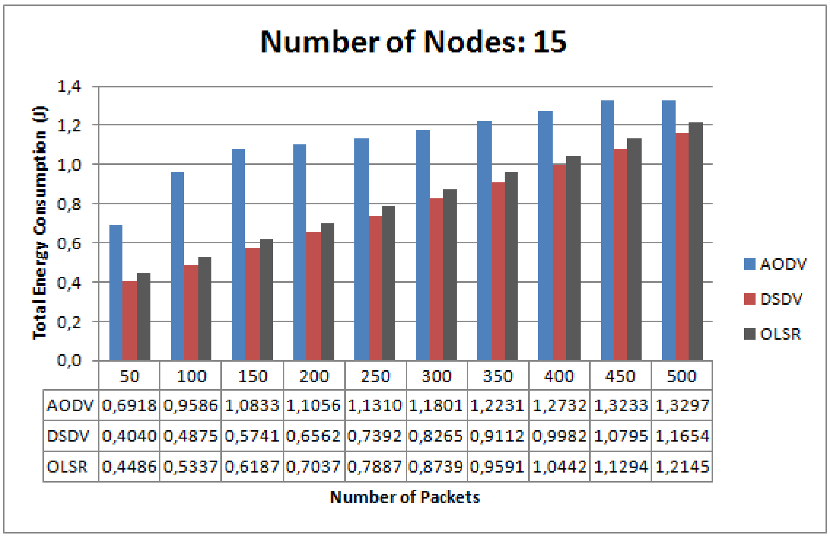
	50	100	150	200	250	300	350	400	450	500
AODV	0,4592	0,6381	0,7681	0,9153	1,0227	1,0646	1,0968	1,1454	1,1824	1,2250
DSDV	0,3117	0,3968	0,4821	0,5673	0,6525	0,7375	0,8230	0,9082	0,9932	1,0784
OLSR	0,3416	0,4267	0,5120	0,5972	0,6824	0,7676	0,8528	0,9381	1,0233	1,1085

Figure 229 Total Energy Consumption vs. Number of Packets, Number of Nodes: 5



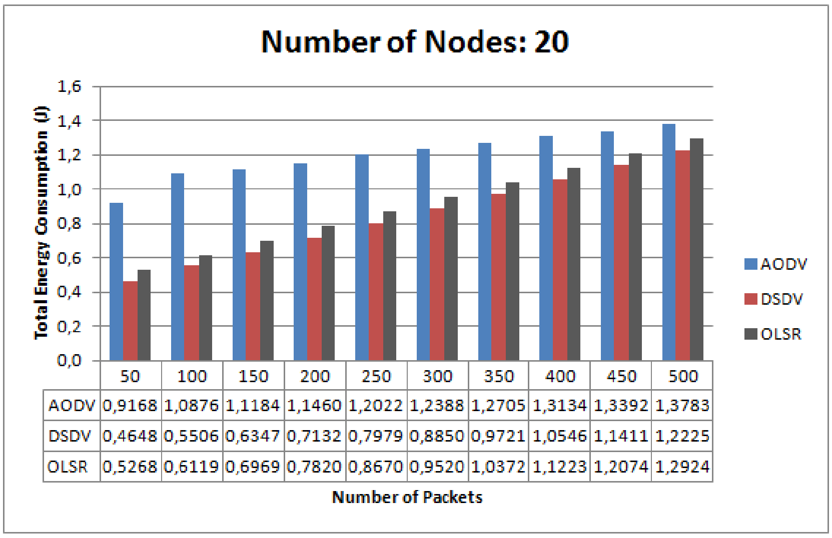
	50	100	150	200	250	300	350	400	450	500
AODV	0,5716	0,7473	0,9445	1,0200	1,0708	1,1350	1,1678	1,1971	1,2355	1,2786
DSDV	0,3497	0,4370	0,5198	0,6043	0,6902	0,7753	0,8598	0,9460	1,0301	1,1167
OLSR	0,3869	0,4721	0,5573	0,6424	0,7276	0,8128	0,8980	0,9831	1,0683	1,1534

Figure 230 Total Energy Consumption vs. Number of Packets, Number of Nodes: 10



	50	100	150	200	250	300	350	400	450	500
AODV	0,6918	0,9586	1,0833	1,1056	1,1310	1,1801	1,2231	1,2732	1,3233	1,3297
DSDV	0,4040	0,4875	0,5741	0,6562	0,7392	0,8265	0,9112	0,9982	1,0795	1,1654
OLSR	0,4486	0,5337	0,6187	0,7037	0,7887	0,8739	0,9591	1,0442	1,1294	1,2145

Figure 231 Total Energy Consumption vs. Number of Packets, Number of Nodes: 15



	50	100	150	200	250	300	350	400	450	500
AODV	0,9168	1,0876	1,1184	1,1460	1,2022	1,2388	1,2705	1,3134	1,3392	1,3783
DSDV	0,4648	0,5506	0,6347	0,7132	0,7979	0,8850	0,9721	1,0546	1,1411	1,2225
OLSR	0,5268	0,6119	0,6969	0,7820	0,8670	0,9520	1,0372	1,1223	1,2074	1,2924

Figure 232 Total Energy Consumption vs. Number of Packets, Number of Nodes: 20

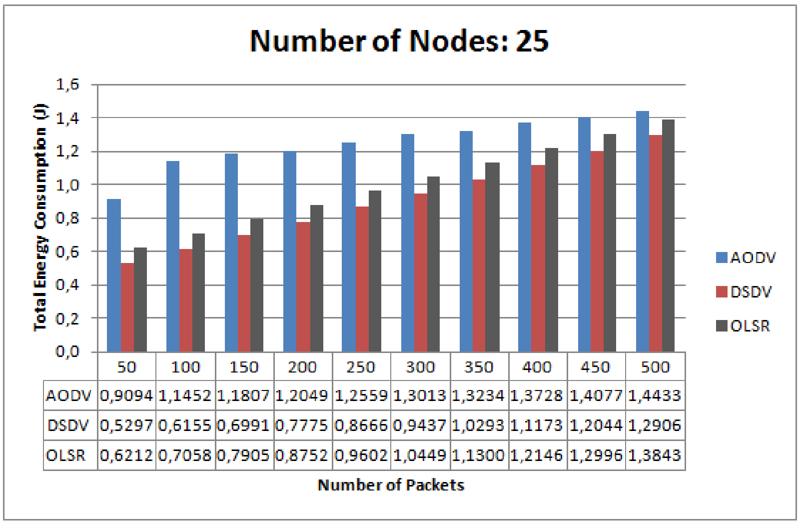


Figure 233 **Total Energy Consumption vs. Number of Packets, Number of Nodes: 25**

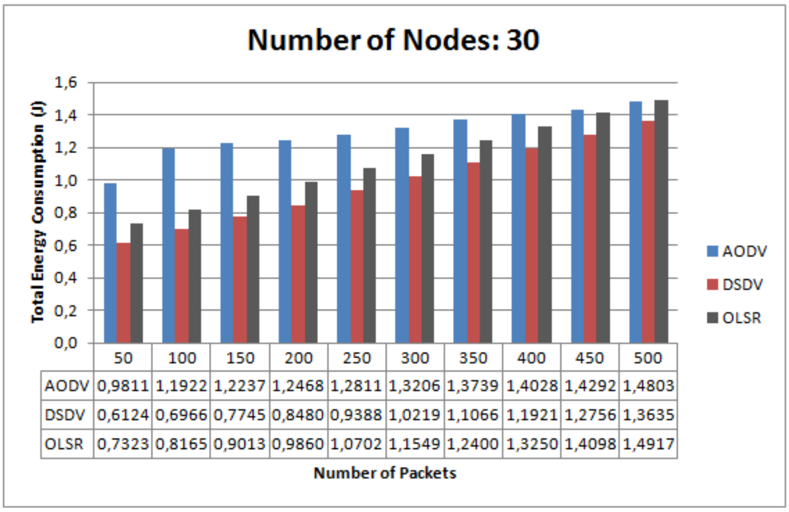


Figure 234 **Total Energy Consumption vs. Number of Packets, Number of Nodes: 30**

The figure that follows (Figure 235) summarizes the results and presents a comparison between the three routing protocols based on the average number of nodes and the number of packets.

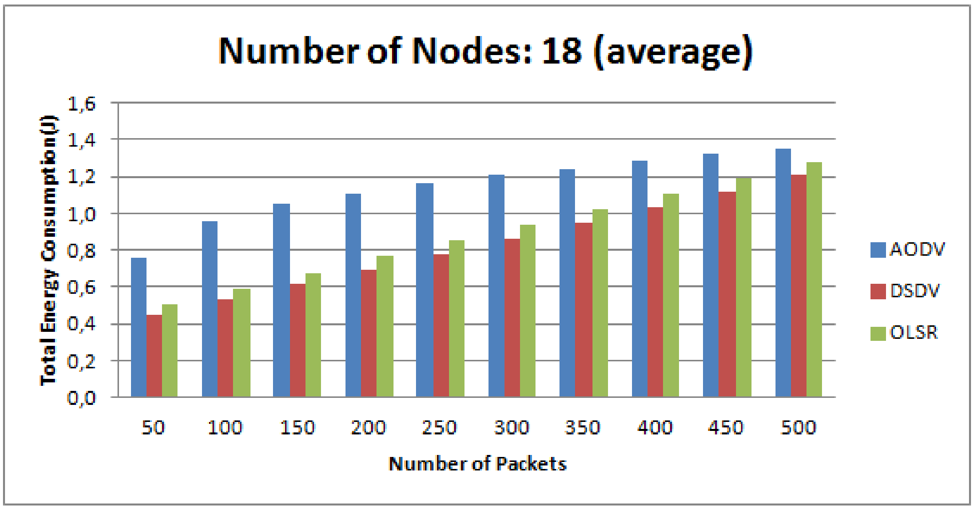


Figure 235 **Total Energy Consumption vs. Number of Packets, Number of Nodes: 18 (average)**

Conclusions

1. Packet Delivery Ratio (PDR)

In the table that follows (Tables 32) we can see which routing protocol has in every case the best performance considering the PDR.

Table 33 Performance Comparison

		Number of Packets									
		50	100	150	200	250	300	350	400	450	500
Number of Nodes	5	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV
	10	DSDV	DSDV	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	15	DSDV	DSDV	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	20	DSDV	DSDV	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	25	DSDV	DSDV	DSDV	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	30	DSDV	DSDV	DSDV	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR

AODV has never the best PDR performance. The other two protocols always outperform AODV. OLSR has in most of the cases the best performance as it has 63.4% the best PDR. DSDV has 36.6% PDR and it comes second (DSDV achieves better results for small number of nodes or for small number of packets). Therefore, the best choice for this scenario based on the PDR is OLSR for high number of packets. DSDV is the best choice for small number of packets or for very small number of nodes.

2. Average Delay

In the table that follows (Tables 33) we can see which routing protocol has in every case the best performance considering the average delay.

Table 34 Performance Comparison

		Number of Packets									
		50	100	150	200	250	300	350	400	450	500
Number of Nodes	5	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV
	10	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV
	15	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV
	20	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV
	25	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV
	30	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV

AODV achieves always the highest delay which is why AODV is not preferable compared to the other two protocols. We have concluded that DSDV outperforms OLSR in every case, as DSDV achieves lower delay. Therefore, the best choice for this scenario based on the average delay is DSDV.

3. Throughput

In the table that follows (Tables 34) we can see which routing protocol has in every case the best performance considering the throughput.

Table 35 Performance Comparison

		Number of Packets									
		50	100	150	200	250	300	350	400	450	500
Number of Nodes	5	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV
	10	DSDV	DSDV	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	15	DSDV	DSDV	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	20	DSDV	DSDV	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	25	DSDV	DSDV	DSDV	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR
	30	DSDV	DSDV	DSDV	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR	OLSR

AODV has never the highest throughput. The other two protocols always outperform AODV. OLSR has in most of the cases the best performance as it has 63.4% the best throughput. DSDV has in the 36.6% of the cases the highest throughput and it comes second. Therefore, the best choice for this scenario based on the throughput is OLSR for high number of packets. DSDV is the best choice for small number of packets or for very small number of nodes.

4. Total Energy Consumption

In the table that follows (Tables 35) we can see which routing protocol has in every case the best performance considering the total energy consumption.

Table 36 Performance Comparison

		Number of Packets									
		50	100	150	200	250	300	350	400	450	500
Number of Nodes	5	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV
	10	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV
	15	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV
	20	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV
	25	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV
	30	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV	DSDV

OLSR outperforms AODV and DSDV outperforms both OLSR and AODV in every case. Therefore, the best choice for this scenario based on the total energy consumption is DSDV.

5 Conclusions

5.1 Results

We have examined the performance of AODV, DSDV and OLSR in three real life scenarios by varying different simulation parameters in each scenario and measuring the four performance metrics (Packet Delivery Ratio (PDR), Average Delay, Throughput, and Total Energy Consumption). From the comparison it is clear that no protocol is absolute winner.

In the first scenario (School Filed Trip) we have a relatively small group (21 nodes) of students that move with the average walking speed (2 m/s) in a certain direction (Gauss-Markov Mobility Model). We have examined the performance of the three routing protocols for different DSSS Rates, packet sizes and interval. With those simulation parameters we can see how the DSSS Rate and the data rate affect the performance of the routing protocols.

Considering the PDR, OLSR has almost always the best results when the DSSS Rate is 2Mbps or higher. When the DSSS Rate is 1Mbps, OLSR has the best performance when the packet size remains small. As the packet size is being increased, DSDV outperforms OLSR. In general OLSR has in 84.1% of the cases the best results. Considering average delay, OLSR outperforms DSDV in most of the cases (59.9%). DSDV has better results mostly when the packet size is being increased. OLSR on the other hand has better results for higher DSSS Rates. The results for throughput show that OLSR outperforms DSDV in most of the cases (83.2%). Only when the DSSS Rate is 1Mbps DSDV outperforms OLSR when the packet size is increased. Finally, considering the total energy consumption, DSDV outperforms OLSR when the DSSS Rate is 1, 2 or 5.5Mbps, but OLSR outperforms DSDV when the DSSS Rate is 11Mbps. We can conclude that OLSR has in most cases the best performance. DSDV has better results for lower DSSS Rates and for bigger packet sizes. AODV never outperforms DSDV or OLSR.

In the second scenario (Rescue Operation) we change the number of nodes and their speed. More specifically we have from 10 to 100 nodes (different networks sizes, small, average and large) and 5 to 55m/s speed in order to see how those simulation parameters affects the performance of the routing protocols. The DSSS Rate is 11Mbps and the data rate is set to 164kbps. The mobility Model is Random Direction. As we have observed, the speed does not affect the performance of OLSR and DSDV. AODV is affected by the speed but not in a certain pattern and not particularly. Only the number of nodes affects the performance. When the number of nodes is increased, the PDR and the throughput are decreased and the average delay and total energy consumption are increased. OLSR has for every simulation metric (PDR (99.09%), Average Delay (89.09%), Throughput (99.09%) and Total Energy Consumption (88.18%) the best performance. In this scenario too AODV never outperforms DSDV or OLSR.

In the third scenario (Archaeological Site), we change the number of nodes in the network (5 to 30 nodes) and we also change the number of packets being send (50 to 500 packets). The DSSS Rate is 1Mbps and the data rate is set to 2200kbps. The mobility Model is Gauss-Markov. With those simulation parameters we can see how the network size (average size networks) and the traffic affects the performance of the routing protocols. From the simulation results we can see that OLSR has better PDR (63.4%) and throughput results (63.4%), but DSDV has in every case better average delay (100%) and total energy consumption (100%) results. In this scenario too AODV never outperforms DSDV or OLSR.

Considering all the aforementioned, we can conclude in the following results:

1. Packet Delivery Ratio (PDR)
    - In general, regardless the size of the network, OLSR performs better.
    - DSDV has better results for bigger packets sizes.
    - DSDV has better results for lower DSSS Rates.
2. Average Delay
    - Average delay is variable in AODV (Reactive Protocol) but remains constant in OLSR and DSDV (Proactive Protocols).
    - DSDV performs better for lower DSSS Rates.
    - For small networks, OLSR has better results. As the size of the network increases, DSDV has better results.
3. Throughput
    - In general, regardless the size of the network, OLSR performs better.
    - DSDV has better results for bigger packets sizes.
    - DSDV has better results for lower DSSS Rates.
4. Total Energy Consumption
    - DSDV performs better for bigger packets sizes.
    - OLSR has better results for higher DSSS Rates, when the size of the network is small (less than 40 nodes).
5. General Results
    - AODV under no circumstances has the best results. DSDV and OLSR always outperform AODV.
    - DSDV and OLSR have in most of the cases similar results due to their nature (proactive routing protocols)
    - Speed does not affect the performance of the protocols on small and average size networks.

Many are the advantages and disadvantages that should be considered before choosing a routing protocol. Proactive routing protocols like OLSR and DSDV continuously try to maintain up-to-date routing information on every node in the network. This has as advantage that connection times are fast, because routing information is already available when the first packet is sent. A disadvantage of proactive protocols is that they continuously use resources to communicate routing information, even when there is no traffic.

On-demand routing protocols do not try to keep their routing tables up-to-date. Instead, a node tries to find a route only when it wants to send a packet. This reduces the traffic needed for routing, but introduces a delay when the first packet is sent to a host.

Considering though the simulation results of our research, for the specific simulation metrics, OLSR and DSDV, therefore the proactive routing protocols have the best performance in the types of networks that we examined. Many other studies that evaluate some of the present simulation metrics and had similar simulation parameters confirm the results of the present study. As an example we can mention (Gurleen Kaur & Charanjit, 2011), (Subramanya, Shwetha, & Devaraju, 2011), (Khan, Zaman, & Reddy, 2008), (Gowrishankar, Basavaraju, Singh, & Kumar Sarkar, 2007), (Sajjad & Asad, 2009). Of course there are studies in favor of reactive routing protocols which is natural due to the different network setup, different simulation parameters and most significant of all, due to the different simulation metrics (for example routing load and number of packets generated).

5.2 Further Study

Ad –hoc networking is a rather hot concept in computer communications. This means that there is much research going on and many issues that remains to be solved. In the present thesis, we have focused on performance evaluation of three routing protocols, AODV, DSDV and OLSR. However there are many issues that could be subject to further studies.

First of all, it is important the practical implementation of those routing protocols, in real applications based on the scenarios that were studied. Also, the simulator environment could be improved. These are just some of the improvements that could be made:

- More routing protocols, for instance TORA, ZRP and DSR.
- Measurement of computing complexity.

Moreover, there are many issues related to ad-hoc networks that could be subject to further studies:

- Simulations which take unidirectional links into consideration.
- An analysis on whether many small control messages are more costly to send in terms of resourced than fewer large control messages.
- Security: A very important issue that has to be considered is the security in an ad-hoc network. Routing protocols are prime targets for impersonation attacks. Because ad-hoc networks are forms without centralized control, security must be handled in a distributed fashion. This will probably mean that IP-Sec (Kent & Atkinson, 1998) authentication headers will be deployed, as well as the necessary key management to distribute keys to the members of the ad-hoc network.
- Quality of Service (QoS): The needs for Quality of Service in an ad-hoc network and how those are related to what the network actually will be used for is an important issue.
- Hand-over of real-time traffic between nodes. How should real-time traffic smoothly be handed over to another node when a route goes down and also how should the flooding be used before a route is found are subjects of study.
- Multicast: We have looked at unicast routing. Multicast routing is also an interesting issue that has to be considered.
- Connecting ad-hoc networks to the Internet through access points: How an ad-hoc network should be connected to the Internet.
- Mobile IP: Integration of mobile IP into ad-hoc networks.

6 References

3ComCorporation. (2000). IEEE 802.11b Wireless LANs, Techical Paper.

Abolhasan, M., Wysocki, T., & Dutkiewicz, E. (2004). A review of routing protocols for mobile ad hoc networks. pp. 1-22.

Adam, G., Bouras, C., Gkamas, A., Kapoulas, V., Kioumourtzis, G., & Tavoularis, N. (2011, Octobe 26-28). Performance Evaluation of Routing Protocols for multimedia transmission over Mobile Ad hoc Networks. *4th Joint IFIP Wireless and Mobile Networks Conference, Toulouse, France* .

Ade, S. A., & Tijare, P. (2010, July-December). Performance Comparison of AODV, DSDV, OLSR and DSR Routing Protocols in Mobile Ad Hoc Networks. *International Journal of Information Technology and Knowledge Management* , pp. 545-548.

Ahvar, E., & Fathy, M. (2007). Performance Evaluation of Routing Protocols For High Density Ad Hoc Networks based on Energy Consumption by GlomoSim Simulator. *World Academy of Science, Engineering and Technology* .

Al-Maashri, A., & Ould-Khaoua, M. (2006, November 14-16). Performance Analysis of MANET Routing Protocols in the Presence of Self-Similar Traffic. *Proceedings of the 31st IEEE Conference on Local Computer Networks* , pp. 801-807.

Amnai, M., Fakhri, Y., & Abouchabaka, J. (2011, June). QoS Routing and Performance Evaluation for Mobile Ad Hoc Networks using OLSR protocol. *International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC)* .

Annamalai, P. (2005). *Comparative Performance Study of Standardized Ad-Hoc Routing Protocols and OSPF-MCDS.* Blacksburg, Virginia: Virginia Polytechnic Institute and State University.

Anton, J. F. (2000, May). Ad Hoc Networks – design and performance issues.

Bai, F., & Helmy, A. (2006). A Survey of Mobility Models in Wireless Adhoc Networks.

Boukerche, A. (2001, April). Performance comparison and analysis of ad hoc routingalgorithms. *IEEE International Conference on Performance, Computing and Communications* , pp. 171-178.

Broch, J., Maltz, D. A., Johnson, D. B., Hu, Y.-C., & Jetcheva, J. (1998, October). A performance comparison of multi-hop wireless ad hoc network routing protocols. *Mobicom* , pp. 85-97.

Broyles, D., Jabbar, A., & Sterbenz, J. P. (2011, October). Design and Analysis of a 3-D Gauss-Markov Mobility Model for Highly Dynamic Airborne Networks. *Proceedings of the international telemetering conference (ITC)* .

Camp, T., Boleng, J., & Davies, V. (2002). A Survey of Mobility Models for Ad Hoc Network Research. *Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications* , pp. 483-502.

Cano, J. C., & Manzoni, P. (2000). A Performance Comparison of Energy Consumption for Mobile Ad Hoc Networks Routing Protocols. *MASCOTS '00 Proceedings of the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems* .

Chang, J.-H., & Tassiulas, L. (2000). Energy conserving routing in wireless ad-hoc networks. *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)* , pp. 22 –31.

Cheng, X., Huang, X., & Du, D. (2006). Ad Hoc Wireless Networking. Kluwer Academic Publishers.

Chlamtac, I., Conti, M., & Liu, J. J.-N. (2003). Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks 1* , pp. 13-64.

Cisco. (2002). Cisco Internetworking. *Cisco Press* .

Clausen, T. H., Hansen, G., Christensen, L., & Behrmann, G. (2001, September). The Optimized Link State Routing Protocol, Evaluation Through Experiments and Simulation. *Proceedings of IEEE Symposium on Wireless Personal Mobile Communications* , pp. 841-846.

Clausen, T., & Jacquet, P. (2003, October). Optimized Link State Routing Protocol (OLSR). RFC 3626. *IETF Network Working Group* .

Clausen, T., Hansen, G., Christensen, L., & Behrmann, G. (2001, September). The Optimized Link State Routing Protocol, Evaluation through Experiments and Simulation. *IEEE Symposium on "Wireless Personal Mobile Communications* .

Corson, M., & Macker, J. (1999, January). Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. *Request For Comments 2501, Internet Engineering Task Force* .

Das, S., Castaneda, R., Yan, J., & Sengupta, R. (1998, October). Comparative performanceevaluation of routing protocols for mobile ad hoc networks. *7th Int. Conf.on Computer Communications and Networks (IC3N)* , pp. 153–161.

Das, S., Perkins, C., & Royer, E. (2000). Performance Comparison of TwoOn-Demand Routing Protocols for Ad Hoc Networks. *INFOCOM* , pp. 3 - 12.

Djenouri, D., Derhab, A., & Badache, N. (2006, April). Ad Hoc Routing Protocols and Mobility. *The International Arab Journal of Information Technology* .

Dr. Roth, U. (2004). *Highly dynamic destination-sequenced distance-vector routing.* http://wiki.uni.lu/secan-lab/Highly+Dynamic+Destination-Sequenced+Distance-Vector+Routing.html(visited 01-01-2012).

Elsevier. http://www.elsevier.com/wps/find/homepage.cws_home.

European Journal of Scientific Research. http://www.eurojournals.com/ejsr.htm.

Feeney, L. M. (1999). *A taxonomy for routing protocols in mobile ad hoc networks. Technical report.* Sweden: Swedish Institute of Computer Science.

Ferro, E., & Potorti, F. (2004, June 30). Bluetooth and Wi-Fi Wireless Protocols: A Survey and a Comparison. *IEEE Wireless Communications magazine* .

Freebersyser, J. A., & Leiner, B. (2001). A DoD perspective on mobile ad hoc networks. *Perkins, C. (Ed.) Ad Hoc Networking* , pp. 29-51.

Ge, Y., Kunz, T., & Lamont, L. (2003). Quality of Service Routing in Ad-Hoc Networks Using OLSR. *Proceeding of the 36th Hawaii International Conference on System Science* .

Gmez, J., Campbell, A. T., Naghshineh, M., & Bisdikian, C. (2001, November). PARO: Conserving Transmission Power in Wireless ad hoc Networks. *IEEE 9th International Conference on Network Protocols (ICNP'01), Riverside, California* .

Goldsmith, A. (2005). *Wireless Communications.* New York, NY, USA: Cambridge University Press.

Gorantala, K. (2006). *Routing Protocols in Mobile Ad-hoc Networks.* Sweden: Umea University, Department of Computing Science.

Gowrishankar, S., Basavaraju, T., Singh, M., & Kumar Sarkar, S. (2007, November 18-19). Scenario based Performance Analysis of AODV and OLSR in Mobile Ad hoc Networks. *Proceedings of the 24th South East Asia Regional Computer Conference* .

Gurleen Kaur, W., & Charanjit, S. (2011). Simulation based Performance Evaluation and Comparison of Proactive and Reactive Routing Protocols in Mobile Ad-hoc Networks. *International Journal of Computer Science and Information Technologies* , pp. 1235 -1239.

Han, L. (2004, October 8). Wireless Ad-hoc Networks.

Hassan, Y. K., Abd El-Aziz, M. H., & Abd El-Radi, A. S. (2010, November). Performance Evaluation of Mobility Speed over MANET Routing Protocols. *International Journal of Network Security* , pp. 128-138.

He, G. (2007). *Destination-sequenced distance vector (DSDV) protocol. Technical report.* Helsinki University of Technology, Finland.

Hoebeke, J., Moerman, I., Dhoedt, B., & Demeester, P. (2004, July). An Overview of Mobile Ad Hoc Networks: Applications and Challenges. *Journal of the Communications Network* , pp. 60-66.

Hogie, L. (2007, April 18). Mobile Ad Hoc Networks: Modelling, Simulation and Broadcast-based Applications.

Hong, X., Xu, K., & Gerla, M. (2002). Scalable routing protocols for mobile ad hoc networks.

Hu, Y.-C., & Johnson, D. B. (2000, August). Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks. *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom 2000)* .

Huhtonen, A. (2004, April 26-27). Comparing AODV and OLSR Routing Protocols.

Huishan, K., Huimin, C., & Nam, K. Y. (n.d.). Routing Protocols in Ad hocWireless Networks. *National University of Singapore* .

IETF Datatracker. http://datatracker.ietf.org/.

IETF MANET Working Group. (n.d.). *http://www.ietf.org/html.charters/manetcharter.charter.html* .

IETF MANET Working Group. http://www.ietf.org/html.charters/manetcharter.charter.html. (n.d.).

Ismail, Z., & Hassan, R. (2011). Effects of Packet Size on AODV Routing Protocol Implementation in Homogeneous and Heterogeneous MANET. *Third International Conference on Computational Intelligence, Modelling & Simulation* .

Jacquet, P., & Viennot, L. (June 2000). *Overhead in mobile ad-hoc network protocols. Technical Report 3965.* Cedex France: Institut National de Recherche en Informatique et en Automatique, Domaine de Voluceau, Rocquencourt.

Jacquet, P., Laouiti, A., Minet, P., & Viennot, L. (2001, September). Performance Analysis of OLSR Multipoint Relay Flooding in Two Ad Hoc Wireless Network Models. Research Report-4260. *RSRCP journal special issue on Mobility and Internet* .

Jacquet, P., Laouiti, A., Minet, P., & Viennot, L. (2002). *Performance of multipoint relaying in ad hoc mobile routing protocols.* Pise, Italy.

Jacquet, P., Mühlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., & Viennot, L. (2001). *Optimized Link State Protocol for Ad Hoc Networks.* Pakistan.

Jayakumar, G., & Ganapathy, G. (2007, November). Performance Comparison of Mobile Ad-hoc Network Routing Protocol. *IJCSNS International Journal of Computer Science and Network Security* .

Jörg, D. O. (2003). Performance Comparison Of MANET Routing Protocols In Different Network Sizes. *Institute of Computer Science and Applied Mathematics Computer Networks and Distributed Systems* .

Kanakaris, V., Ndzi, D., & Azzi, D. (2010). Ad-hoc Networks Energy Consumption: A review of the Ad-Hoc Routing Protocols. *Journal of Engineering Science and Technology* , pp. 162-167.

Karthikeyan, M., Angayarkanni, K., & Sujatha, D. (2010, March 17-19). Throughput Enhancement in Scalable MANETs using Proactive and Reactive Routing Protocols. *Proceedings of the International MultiConference of Engineers and Computer Scientists* .

Kent, S., & Atkinson, R. (1998, July). Security Architecture for the Internet Protocol.

Khan, K. U., Zaman, R. U., & Reddy, A. V. (2008). Performance Comparison of On-Demand and Table Driven Ad Hoc Routing Protocols using NCTUns. *Tenth International Conference on Computer Modeling and Simulation* .

Kurkowski, S., Camp, T., & Colagrosso, M. (2005). MANET Simulation Studies: The Current State and New Simulation Tools. *Sciences New York* , pp. 50-61.

Lakshmikanth, G., Patel, S., & Gaiwak, A. (2009, November). Performance Evaluation of Multicast Routing Protocols in MANET. *International Journal of Recent Trends in Engineering* .

Lam, D., Cox, D. C., & Widom, J. (1999, October). Teletraffic modeling for personal communication services. *IEEE Communications Magazine* , pp. 79-87.

Lang, D. (2006). On the Evaluation and Classification of Routing Protocols for Mobile Ad Hoc Networks.

Laouiti, A., Qayyum, A., & Viennot, L. (2002). Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks. *35th Annual Hawaii International Conference on System Sciences* .

Laouti, A., Mühlethaler, P., Najid, A., & Plakoo, E. (2002). *Simulation Results of the OLSR Routing Protocol for Wireless Network.* Sardegna, Italy: 1st Mediterranean Ad-Hoc Networks workshop (Med-Hoc-Net).

Liang, B., & Haas, Z. J. (1999, April). Predictive Distance-Based Mobility Management for PCS Networks. *Proceedings of IEEE Information Communications Conference (INFOCOM 1999)* .

Lin, T. (2004, March 19). Mobile Ad Hoc Network Routing Protocols: Methodologies and Applications.

Liu, Y., Ning, P., Dai, H., & Liu, A. (2010). Randomized Differential DSSS: Jamming-Resistant Wireless Broadcast Communication. *IEEE INFOCOM 2010 proceedings* .

Lundberg., J. (2000). *Routing Security in Ad Hoc Networks.* Helsinki University of Technology.

Manickam, P., Guru Baskar, T., Girija, M., & Manimegalai, D. (2011, February). Performance Comparisons of Routing Protocols in Mobile Ad Hoc Networks. *International Journal of Wireless & Mobile Networks (IJWMN)* .

Markdoulidakis, J. G., Lyberopoulos, G. L., Tsirkas, D. F., & Sykas, E. D. (1997, August). Mobility modeling in third-generation mobile telecommunication systems. *IEEE Personal Communications* , pp. 41-56.

Mohseni, S., Hassan, R., Patel, A., & Razali, R. (2010). Comparative Review Study of Reactive and Proactive Routing Protocols in MANETs. *4th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2010)* .

Monks, J. P., Ebert, J.-P., Wolisz, A., & Hwu, W. W. (2001, November). A Study of the Energy Saving and Capacity Improvement Potential of Power Control in Multi-hop Wireless Networks. *Proc. of Workshop on Wireless Local Networks, Tampa, Florida, USA* .

Narayanaswamy, S., Kawadia, V., Sreenivas, R. S., & Kumar, P. R. (2001, May 16). The COMPOW protocol for power control in ad hoc networks: Theory, architecture, algorithm, implementation, and experimentation.

Navarro, M. (2001). Optimal routing of multicast traffic in ad hoc networks. *Master's Thesis of Networking Laboratory. Helsinki University of University* .

Ning, P., & Sun, K. (2003). *How to misuse aodv: a case study of insider attacks against mobile ad-hoc routing protocols. Technical report.* Raleigh, NC, USA: Comput. Sci. Dept., North Carolina State Univ.

NS-3. http://www.nsnam.org/.

Patel, D. D. (2000). Energy in ad-hoc networking for the pico radio. Technical report.

Perkins, C. E. (1998, November 17). Mobile Ad Hoc Networking Terminology. *Internet Draft, draft-ietf-manet-term-01.txt* .

Perkins, C. E., & Bhagwat, P. (1994). *Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers.* London England UK: SIGCOMM 94-8/94.

Perkins, C., Belding-Royer, E., & Das, S. (2003, July). Ad hoc On-Demand Distance Vector (AODV) Routing. *IETF NetworkWorking Group* .

Ramachandran, K. (2006). *Aodv. Technical report.* Santa Barbara, USA: University of California, http://moment.cs.ucsb.edu/AODV/aodv.html(visited 01-01-2012).

Royer, E. M., & Toh, C.-K. (1999). *A review of current routing protocols for ad hoc mobile wireless networks. Technical report.* USA: University of California and Georgia Institute of Technology.

Royer, E., Melliar-Smith, P., & Moser, L. (2000). An Analysis of the Optimum Node Density for Ad hoc Mobile Networks.

Ryu, J.-h., & Cho, D.-H. (2001, February). A new routing scheme concerning energy conservation in wireless home AD-HOC networks. *IEEE Transactions on Consumer Electronics* , pp. 1-5.

Saad, M. I., & Zukarnain, Z. A. (2009). Performance Analysis of Random-Based Mobility Models in MANET Routing Protocol. *European Journal of Scientific Research* , pp. 444-454.

Sadasivam, K., Changrani, V., & Yang, T. A. (2005). Scenario Based Performance Evaluation of SecureRouting in MANETs. *In Proceedings of ICWN* .

Sajjad, A., & Asad, A. (2009). *Performance Analysis of AODV, DSR and OLSR in MANET.* Sweden: Department of Electrical Engineering with emphasis on Telecommunication Blekinge Institute of Technology.

Schiller, J. (2000). *Mobile Communications.* Addison-Wesley.

Schwingenschlogl, C., & Kosch, T. (2002). *Geocast enhancements of aodv for vehicular networks. Technical report.* Munich, Germany: Institute of Communication Networks, Munich University of Technology. and BMW Research.

Subramanya, B., Shwetha, D., & Devaraju, J. (2011, August). A Performance Study of Proactive, Reactive and Hybrid Routing Protocols using Qualnet Simulator. *International Journal of Computer Applications (0975 – 8887)* .

Swarnapriyaa, U., Vinodhini, V., Anthoniraj, S., & Anand, R. (2011, February 17-18). Auto Configuration in Mobile Ad Hoc Networks. *Proceedings of the National Conference on Innovations in Emerging Technology* , pp. 61-66.

The Mobile Ad Hoc Networks Study Group (GEDOC). http://www.siam.dcc.ufmg.br/gedoc.

Toh, C. (2001, june). Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks. *IEEE Communications* .

Tuteja, A., Gujral, R., & Thalia, S. (2010, June 20-21). Comparative Performance Analysis of DSDV, AODV and DSR Routing Protocols in MANET Using NS2.

Vanhala, A. (2000). *Security in Ad-hoc Networks.* Helsinki University of Technology.

Vasiliou, A., & Economides, A. A. (2005). Evaluation of Multicasting Algorithms in Manets. *Proceedings of the World Academy of Science, Engineering and Technology (PWASET), Vol 5, April, pp. 94-97. 3rd World Enformatika Conference (WEC'05)- International Conference on Telecommunications and Electronic Commerce (TEC 2005)* .

Vasiliou, A., & Economides, A. A. (2007). Mobile collaborative learning using multicast MANETs. *International Journal of Mobile Communications* , pp. 423-444.

Vasiliou, A., & Economides, A. A. (2008, April 16-18). MANET-based outdoor collaborative learning. *IMCL2008 Conference* .

Vasiliou, A., & Economides, A. (2007, July 24-26). Game-based learning using MANETs. *Proceedings of the 4th WSEAS/ASME International Conference on Engineering Education (EE'07)* , pp. 154-159.

Vasiliou, A., & Economides, A. (2006, September 3-6). MANETs for environmental monitoring. *Proceedings of the ITS2006- International Telecommunications Symposium* , pp. 813-818.

Vasiliou, A., & Economides, A. (2005). Multicast groups in MANETs. *WSEAS Transactions on Communications* , pp. 686-693.

Vasiliou, A., & Economides, A. (2005, July 11-16). Partitioning multicast MANETs. *Proceedings of the 9th International Conference on Communications* , pp. 497-537.

Vasiliou, A., & Economides, A. (2005). Performance of multicast groups moving towards the same direction. *Proceedings of the Annual Conference on Telecommunications & Multimedia (TEMU 2005)* , pp. 1-6.

Vasiliou, A., & Economides, A. (2006, July). Rescue operations using coordinated multicast MANET. *Proceedings of the Fifth International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP'06)* .

Vasiliou, A., & Economides, A. (2006, July 5-7). Scalable multiple team cooperation using MANET multicasting. *Proceedings of the International Conference on Telecommunications & Multimedia (TEMU 2006)* .

Weniger, K., & Zitterbart, M. (2002, February). IPv6 Autoconfiguration in Large Scale Mobile Ad-Hoc Networks. *Proceedings of European Wireless 2002, Florence Italy* .

Wikipedia. *http://www.wikipedia.org/.*

Xu, Y., Heidemann, J., & Estrin, D. (2000, October 12). Adaptive Energy-Conserving Routing for Multihop Ad-hoc Networks. *USC-ISI Research Report 527* .

7 Appendix A – Terminology

This appendix contains some terminology (Perkins C. E., Mobile Ad Hoc Networking Terminology, 1998) that is related to ad hoc networks.

7.1 General Terms

Bandwidth: Total link capacity of a link to carry information (typically bits).

Channel: The physical medium is divided into logical channel, allowing possibly shared uses of the medium. Channels may be made available by subdividing the medium into distinct time slots, distinct spectral bands, or decorrelated coding sequences.

Flooding: The process of delivering data or control messages to every node within any data network.

Host: Any node that is not a router.

Interface: A nodes attachment to a link.

Link: A communication facility or medium over which nodes can communicate at the link layer.

Loop free: A path taken by a packet never transits the same intermediate node twice before arrival at the destination.

MAC-layer address: An address (sometimes called the link address) associated with the link interface of a node on a physical link.

Next hop: A neighbor, which has been designated to forward packets along the way to a particular destination.

Neighbor: A node that is within transmitter range from another node on the same channel.

Node: A device that implements IP.

Router: A node that forwards IP packets not explicitly addressed to itself. In case of ad-hoc networks, all nodes are at least unicast routers.

Routing table: The table where the routing protocols keep routing information for various destinations. This information can include nex thop and the number of hop to the destination.

Scalability: A protocol is scalable if it is applicable to large as well as small populations.

Throughput: The amount of data from a source to a destination processed by the protocol for witch throughput is to be measured for instance, IP, TCP, or the MAC protocol.

7.2 Ad-hoc Related Terms

Ad-hoc: “For this special or temporary purpose” or “a special case without generic support”.

AODV: Ad-Hoc On-Demand Distance Vector. Routing protocol for wireless ad-hoc networks.

Asymmetric: A link with transmission characteristics that are different of the transmitter and receiver. For instance, the range of one transmitter may be much higher than the range of another transmitter on the same medium. The transmission between the two hosts will therefore not work equally well in both directions. See also symmetric.

Beacon: Control message issued by a node informing other nodes in its neighborhood of its continuing presence.

CBRP: Cluster Based Routing Protocol. Routing protocol for wireless ad-hoc networks.

Cluster: A group of nodes typically in range of each other, where one of the nodes is elected as the cluster head. The cluster head ID identifies the cluster. Each node in the network knows it’s corresponding cluster head(s) and therefore knows which cluster(s) it belongs to.

DSDV: Dynamic Source Routing. Routing protocol for wireless Ad Hoc networks.

DSR: Dynamic Source Routing. Routing protocol for wireless Ad Hoc networks.

Proactive: Tries to maintain the routing map for the whole network all the time.

Reactive: Calculates route only upon receiving a specific request.

RREQ: Routing Request. A message used by AODV for the purpose of discovering new routes to a destination node.

RREP: Route Reply. A message used by AODV to reply to route requests.

TORA: Temporally Ordered Routing Algorithm. Routing protocol for wireless ad-hoc networks.

ZRP: Zone Routing Protocol. Routing protocol for wireless ad-hoc networks.

8 Appendix B – Simulation Scripts

This appendix contains the simulation Scripts for the three ad-hoc networks that was implemented.

8.1 School Field Trip Scenario

```
/*1st Simulation Scenario - School Field Trip*/
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/mobility-module.h"
#include "ns3/config-store-module.h"
#include "ns3/wifi-module.h"
#include "ns3/energy-module.h"
#include "ns3/internet-module.h"
#include "ns3/aodv-helper.h"
#include "ns3/aodv-routing-protocol.h"
#include "ns3/delay-jitter-estimation.h"

#include <iostream>
#include <fstream>

NS_LOG_COMPONENT_DEFINE ("1st - School Field Trip");

using namespace ns3;
using namespace std;

uint32_t received_packets = 0;
long double total_delay = 0;
long double total_energy_consumption = 0;

void ReceivePacket (Ptr<Socket> socket)
{
 ofstream outfile;
 outfile.open("1st_Simulation_general.txt", ios::app);
 Ptr<Packet> packet;

 Address from;
 while (packet = socket->RecvFrom (from))
 {
   if (packet->GetSize () > 0)
   {
     received_packets++;
     DelayJitterEstimation delay;
     delay.RecordRx(packet);
     total_delay += delay.GetLastDelay().GetSeconds();
     InetSocketAddress iaddr = InetSocketAddress::ConvertFrom (from);

       NS_LOG_UNCOND ("--\nReceived one packet! Number of Packets until now:
       " << received_packets << "  From: " << iaddr.GetIpv4() << " Delay: "
       << delay.GetLastDelay().GetSeconds()  << " Packetsize "<< packet ->
       GetSize() << " port: " << iaddr.GetPort () << " at time = " <<
       Simulator::Now ().GetSeconds () << "\n--");

       outfile << "Received one packet! Socket: "<< iaddr.GetIpv4 ()<< "
       port: " << iaddr.GetPort () << " at time = " << Simulator::Now
       ().GetSeconds () << "\n";

       (void) iaddr;
   }
}
 outfile.close();
}
```

```
static void GenerateTraffic (Ptr<Socket> socket, uint32_t pktSize, Ptr<Node>
n, uint32_t pktCount, Time pktInterval)
{
 Ptr<Packet> p = Create<Packet> (pktSize);
 DelayJitterEstimation delay;
 delay.PrepareTx(p);
 if (pktCount > 0)
 {
       socket->Send (p);
       Simulator::Schedule (pktInterval, &GenerateTraffic, socket, pktSize,
n, pktCount - 1, pktInterval);
 }
 else
 {
       socket->Close ();
 }
}

void TotalEnergy (double oldValue, double totalEnergy)
{
 ofstream outfile;
 outfile.open("1st_Simulation_general.txt", ios::app);

 NS_LOG_UNCOND ( Simulator::Now().GetSeconds() <<"s Total energy   consumed
 by radio = " << totalEnergy << "J");

 outfile << Simulator::Now().GetSeconds() << "s Total energy consumed by
 radio = " << totalEnergy << "J\n";

 outfile.close();
 total_energy_consumption = totalEnergy;
}


int main (int argc, char *argv[])
{
 SeedManager::SetSeed(142245);
 bool verbose = false;
 double startTime = 10.0;
 uint32_t numNodes = 21;  // number of nodes (20 students + teacher)
 uint32_t numPackets = 250; // number of packets to send
 std::string phyMode ("DsssRate1Mbps");//DSSS Rates: 1, 2, 5.5, 11Mbps

 uint32_t packetSize = atoi(argv[1]); //Packet Size: 256, 512,   1024, 2048,
                                              4096 bytes
 long double interval = atof(argv[2]); //Interval: 0, 0.05, 0.1, 0.15, 0.2,
                                              0.25, 0.3, 0.35, 0.4, 0.45,
                                              0.5

 ofstream outfile;
 ofstream outfile2;

 outfile.open("1st_Simulation_general.txt", ios::app);
 outfile << "Simulation Parameters" << endl
         << "Routing Protocl: AODV" << endl
         << "Number of Nodes:      " << numNodes << endl
         << "Numbet of Packets sent from each node:   " << numPackets <<
endl
         << "Packet Size:          " << packetSize << endl
         << "DsssRate:             " << phyMode << endl
         << "Interval:             " << interval << endl;

 outfile2.open("1st_Simulation_results.txt", ios::app);
 outfile2 << "----Simulation Parameters----" << endl
          << "Routing Protocl: AODV" << endl
          << "Number of Nodes:      " << numNodes << endl
```

```
          << "Numbet of Packets sent from each node:   " << numPackets <<
endl
          << "Packet Size:          " << packetSize << endl
          << "DsssRate:             " << phyMode << endl
          << "Interval:             " << interval << endl;

CommandLine cmd;
cmd.AddValue ("phyMode", "Wifi Phy mode", phyMode);
cmd.AddValue ("PpacketSize", "size of application packet sent", packetSize);
cmd.AddValue ("numPackets", "Total number of packets to send", numPackets);
cmd.AddValue ("startTime", "Simulation start time", startTime);
cmd.AddValue ("verbose", "Turn on all device log components", verbose);
cmd.Parse (argc, argv);

// Convert to time object
Time interPacketInterval = Seconds (interval);
// disable fragmentation for frames below 2200 bytes
Config::SetDefault ("ns3::WifiRemoteStationManager::FragmentationThreshold",
StringValue ("2200"));
// turn off RTS/CTS for frames below 2200 bytes
Config::SetDefault ("ns3::WifiRemoteStationManager::RtsCtsThreshold",
StringValue ("2200"));
// Fix non-unicast data rate to be the same as that of unicast
Config::SetDefault ("ns3::WifiRemoteStationManager::NonUnicastMode",
StringValue (phyMode));

NodeContainer networkNodes;
networkNodes.Create (numNodes);

// The below set of helpers will help us to put together the wifi NICs we
want
WifiHelper wifi;
if (verbose)
{
 wifi.EnableLogComponents ();
}

wifi.SetStandard (WIFI_PHY_STANDARD_80211b);

/** Wifi PHY **/

/***************************************************************************/

YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default ();
wifiPhy.Set ("RxGain", DoubleValue (1));
wifiPhy.Set ("TxGain", DoubleValue (1));
wifiPhy.Set ("CcaMode1Threshold", DoubleValue (0.0));

/***************************************************************************/

/** wifi channel **/
YansWifiChannelHelper wifiChannel;
wifiChannel.SetPropagationDelay ("ns3::ConstantSpeedPropagationDelayModel");
wifiChannel.AddPropagationLoss ("ns3::FriisPropagationLossModel");

// create wifi channel
Ptr<YansWifiChannel> wifiChannelPtr = wifiChannel.Create ();
wifiPhy.SetChannel (wifiChannelPtr);

/** MAC layer **/

// Add a non-QoS upper MAC, and disable rate control
NqosWifiMacHelper wifiMac = NqosWifiMacHelper::Default ();
wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager",
                              "DataMode", StringValue (phyMode),
                              "ControlMode", StringValue (phyMode));

// Set it to ad-hoc mode
```

```
wifiMac.SetType ("ns3::AdhocWifiMac");

/** install PHY + MAC **/
NetDeviceContainer devices = wifi.Install (wifiPhy, wifiMac, networkNodes);

/** mobility **/
MobilityHelper mobility;

mobility.SetMobilityModel ("ns3::GaussMarkovMobilityModel",
   "Bounds", BoxValue (Box (0, 1000, 0, 500, 0, 500)),
   "TimeStep", TimeValue (Seconds (0.5)),
   "Alpha", DoubleValue (0.95),
   "MeanVelocity", RandomVariableValue (UniformVariable (0, 2)),
   "MeanDirection", RandomVariableValue (UniformVariable (0, 6.283185307)),
   "MeanPitch", RandomVariableValue (UniformVariable (0.05, 0.05)),
   "NormalVelocity", RandomVariableValue (NormalVariable (0.0, 1.0, 10.0)),
   "NormalDirection", RandomVariableValue (NormalVariable (0.0, 0.2, 0.4)),
   "NormalPitch", RandomVariableValue (NormalVariable (0.0, 0.02, 0.04)));

mobility.SetPositionAllocator ("ns3::RandomBoxPositionAllocator",
   "X", RandomVariableValue (UniformVariable (0, 1000)),
   "Y", RandomVariableValue (UniformVariable (0, 500)),
   "Z", RandomVariableValue (UniformVariable (0, 500)));

mobility.Install (networkNodes);

/* Energy Model*/
BasicEnergySourceHelper basicSourceHelper;
basicSourceHelper.Set("BasicEnergySourceInitialEnergyJ", DoubleValue(0.1));
EnergySourceContainer sources = basicSourceHelper.Install(networkNodes);
WifiRadioEnergyModelHelper radioEnergyHelper;
radioEnergyHelper.Set("TxCurrentA", DoubleValue(0.0174));
DeviceEnergyModelContainer deviceModels = radioEnergyHelper.Install(devices,
sources);

/** Enable Routing Protocol (AODV) **/
AodvHelper aodv;

/** Internet stack **/
InternetStackHelper internet;
internet.SetRoutingHelper(aodv);
internet.Install (networkNodes);

/** Assign Network Adresses **/
Ipv4AddressHelper ipv4;
NS_LOG_INFO ("Assign IP Addresses.");
ipv4.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer i = ipv4.Assign (devices);

/** Energy **/
// all sources are connected to node 0 (energy source)
Ptr<BasicEnergySource> basicSourcePtr = DynamicCast<BasicEnergySource>
(sources.Get(0));
Ptr<DeviceEnergyModel> basicRadioModelPtr = basicSourcePtr ->
FindDeviceEnergyModels ("ns3::WifiRadioEnergyModel").Get(0);
NS_ASSERT (basicRadioModelPtr != NULL);
basicRadioModelPtr -> TraceConnectWithoutContext("TotalEnergyConsumption",
MakeCallback(&TotalEnergy));

TypeId tid = TypeId::LookupByName ("ns3::UdpSocketFactory");

Ptr<Socket> recvSink = Socket::CreateSocket (networkNodes.Get (0), tid);
InetSocketAddress local = InetSocketAddress (Ipv4Address::GetAny (), 80);
recvSink->Bind (local);
recvSink->SetRecvCallback (MakeCallback (&ReceivePacket));

Ptr<Socket> source;
for(uint32_t count1 = 1; count1 < numNodes; count1++)
```

```
{
 source = Socket::CreateSocket (networkNodes.Get (count1), tid);
 InetSocketAddress remote = InetSocketAddress (Ipv4Address::GetBroadcast (),
80);
 source->SetAllowBroadcast (true);
 source->Connect (remote);
}

for(uint32_t count2 = 1; count2 < numNodes; count2++){
 Simulator::Schedule (Seconds(startTime), &GenerateTraffic, source,
packetSize, networkNodes.Get(count2), numPackets, interPacketInterval);
}

 Simulator::Stop (Seconds (180.0));
 Simulator::Run ();
 Simulator::Destroy();

NS_LOG_UNCOND ("Number of packets sent: " << numPackets*(numNodes - 1) <<
              " packets. \nNumber of packets received: " << received_packets
              << " packets. \nPackets received: " <<
              double(received_packets*100)/(numPackets*(numNodes - 1)) <<
              "%. \nAverage Delay: " << total_delay/received_packets <<
              " seconds. \nData Rate: " << ((packetSize/interval)*8)/1000 <<
              " kbps. \nThroughput: " <<
              double((received_packets*packetSize)/180)*8/1000 <<
              " kbps. \nTotal energy consumption: " <<
              total_energy_consumption << " J.\nEnd of Simulation");

outfile2 << "----Simulation Results----" << endl <<
              "Number of packets sent: " << numPackets*(numNodes - 1) << endl
              << "Number of packets received: " << received_packets << "
              packets. " << endl << "Packets received: " <<
              double(received_packets*100)/(numPackets*(numNodes - 1)) <<
              "%." << endl << "Average Delay: " <<
              total_delay/received_packets << " seconds." << endl <<
              "Data Rate: " << ((packetSize/interval)*8)/1000 << " kbps." <<
              endl << "Throughput: " <<
              double((received_packets*packetSize)/180)*8/1000 << endl <<
              "Total energy consumption: " << total_energy_consumption << "
              J."<< endl;

outfile << "End of Simulation" << endl << endl;
outfile2 << "End of Simulation" << endl << endl <<
       "----------------------------------------------------------" << endl <<
endl;

outfile.close();
outfile2.close();

received_packets = 0;
total_delay = 0;
total_energy_consumption = 0;

return 0;
}
```

8.2 Rescue Operation Scenario

```
/*2nd - Rescue Operation Scenario */
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/mobility-module.h"
#include "ns3/config-store-module.h"
#include "ns3/wifi-module.h"
#include "ns3/energy-module.h"
#include "ns3/internet-module.h"
#include "ns3/aodv-helper.h"
#include "ns3/aodv-routing-protocol.h"
#include "ns3/delay-jitter-estimation.h"

#include <iostream>
#include <fstream>

NS_LOG_COMPONENT_DEFINE ("2nd - Rescue Operation Scenario");

using namespace ns3;
using namespace std;

uint32_t received_packets = 0;
long double total_delay = 0;
long double total_energy_consumption = 0;

void ReceivePacket (Ptr<Socket> socket)
{
 ofstream outfile;
 outfile.open("2nd_Simulation_general.txt", ios::app);
 Ptr<Packet> packet;

 Address from;
 while (packet = socket->RecvFrom (from))
 {
       if (packet->GetSize () > 0)
       {
         received_packets++;
         DelayJitterEstimation delay;
         delay.RecordRx(packet);
         total_delay += delay.GetLastDelay().GetSeconds();
         InetSocketAddress iaddr = InetSocketAddress::ConvertFrom (from);

         NS_LOG_UNCOND ("--\nReceived one packet! Number of Packets until
         now: " << received_packets << "  From: " << iaddr.GetIpv4() << "
         Delay: " << delay.GetLastDelay().GetSeconds()  << " Packetsize "<<
         packet -> GetSize() << " port: " << iaddr.GetPort () << " at time =
         " << Simulator::Now ().GetSeconds () << "\n--");

         outfile << "Received one packet! Socket: "<< iaddr.GetIpv4 ()<< "
         port: " << iaddr.GetPort () << " at time = " << Simulator::Now
         ().GetSeconds () << "\n";

       (void) iaddr;
       }
}
 outfile.close();
}

static void GenerateTraffic (Ptr<Socket> socket, uint32_t pktSize, Ptr<Node>
n, uint32_t pktCount, Time pktInterval)
{
 Ptr<Packet> p = Create<Packet> (pktSize);
 DelayJitterEstimation delay;
 delay.PrepareTx(p);
 if (pktCount > 0)
 {
```

```
       socket->Send (p);
       Simulator::Schedule (pktInterval, &GenerateTraffic, socket, pktSize,
       n, pktCount - 1, pktInterval);
 }
 else
 {
       socket->Close ();
 }
}

void TotalEnergy (double oldValue, double totalEnergy)
{
 ofstream outfile;
 outfile.open("2nd_Simulation_general.txt", ios::app);

 NS_LOG_UNCOND ( Simulator::Now().GetSeconds() <<"s Total energy consumed by
radio = " << totalEnergy << "J");

 outfile << Simulator::Now().GetSeconds() << "s Total energy consumed by
radio = " << totalEnergy << "J\n";

outfile.close();
 total_energy_consumption = totalEnergy;
}


int main (int argc, char *argv[])
{
 SeedManager::SetSeed(147852);
 bool verbose = false;
 double startTime = 10.0;
 uint32_t numPackets = 150;  // number of packets to send
 std::string phyMode ("DsssRate11Mbps");
 uint32_t packetSize = 2048;
 long double interval = 0.1;

 uint32_t numNodes = atoi(argv[1]);      //Number of Nodes: 10, 20, 30, 40,
                                          50, 60, 70, 80, 90, 100
 uint32_t speed = atof(argv[2]);   //Speed: 5, 10, 15, 20, 25, 30, 35, 40,
                                    45, 50, 55

 ofstream outfile;
 ofstream outfile2;

 outfile.open("2nd_Simulation_general.txt", ios::app);
 outfile << "Simulation Parameters" << endl
         << "Routing Protocl: AODV" << endl
         << "Number of Nodes:     " << numNodes << endl
        << "Speed: " << speed << " m/s" << endl
         << "Numbet of Packets sent from each node:   " << numPackets << endl
         << "Data rate: " << ((packetSize/interval)*8)/1000  << " Mbps" <<
         endl;

 outfile2.open("2nd_Simulation_results.txt", ios::app);
 outfile2 << "----Simulation Parameters----" << endl
          << "Routing Protocl: AODV" << endl
          << "Number of Nodes:     " << numNodes << endl
          << "Speed: " << speed << " m/s" << endl
         << "Numbet of Packets sent from each node:   " << numPackets <<endl
         << "Data rate: " << ((packetSize/interval)*8)/1000  << " Mbps" <<
       endl;

CommandLine cmd;
cmd.AddValue ("phyMode", "Wifi Phy mode", phyMode);
cmd.AddValue ("PpacketSize", "size of application packet sent", packetSize);
cmd.AddValue ("numPackets", "Total number of packets to send", numPackets);
cmd.AddValue ("startTime", "Simulation start time", startTime);
cmd.AddValue ("verbose", "Turn on all device log components", verbose);
```

```
cmd.Parse (argc, argv);

// Convert to time object
Time interPacketInterval = Seconds (interval);
// disable fragmentation for frames below 2200 bytes
Config::SetDefault ("ns3::WifiRemoteStationManager::FragmentationThreshold",
StringValue ("2200"));
// turn off RTS/CTS for frames below 2200 bytes
Config::SetDefault ("ns3::WifiRemoteStationManager::RtsCtsThreshold",
StringValue ("2200"));
// Fix non-unicast data rate to be the same as that of unicast
Config::SetDefault ("ns3::WifiRemoteStationManager::NonUnicastMode",
StringValue (phyMode));

NodeContainer networkNodes;
networkNodes.Create (numNodes);

// The below set of helpers will help us to put together the wifi NICs we
want
WifiHelper wifi;
if (verbose)
{
 wifi.EnableLogComponents ();
}

wifi.SetStandard (WIFI_PHY_STANDARD_80211b);

/** Wifi PHY **/

/***************************************************************************/

YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default ();
wifiPhy.Set ("RxGain", DoubleValue (5));
wifiPhy.Set ("TxGain", DoubleValue (5));
wifiPhy.Set ("CcaMode1Threshold", DoubleValue (0.0));

/***************************************************************************/

/** wifi channel **/
YansWifiChannelHelper wifiChannel;
wifiChannel.SetPropagationDelay ("ns3::ConstantSpeedPropagationDelayModel");
wifiChannel.AddPropagationLoss ("ns3::FriisPropagationLossModel");

// create wifi channel
Ptr<YansWifiChannel> wifiChannelPtr = wifiChannel.Create ();
wifiPhy.SetChannel (wifiChannelPtr);

/** MAC layer **/

// Add a non-QoS upper MAC, and disable rate control
NqosWifiMacHelper wifiMac = NqosWifiMacHelper::Default ();
wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager",
                             "DataMode", StringValue (phyMode),
                             "ControlMode", StringValue (phyMode));

// Set it to ad-hoc mode

wifiMac.SetType ("ns3::AdhocWifiMac");

/** install PHY + MAC **/
NetDeviceContainer devices = wifi.Install (wifiPhy, wifiMac, networkNodes);

/** mobility **/
MobilityHelper mobility;

mobility.SetMobilityModel ("ns3::RandomDirection2dMobilityModel",
             "Bounds", RectangleValue (Rectangle(0, 500, 0, 500)),
             "Speed", RandomVariableValue (ConstantVariable(speed)),
```

```
              "Pause", RandomVariableValue (ConstantVariable(0.2)));

 mobility.SetPositionAllocator("ns3::RandomRectanglePositionAllocator",
       "X", RandomVariableValue (UniformVariable (0, 500)),
       "Y", RandomVariableValue (UniformVariable (0, 500)));

 mobility.Install (networkNodes);

/* Energy Model*/
BasicEnergySourceHelper basicSourceHelper;
basicSourceHelper.Set("BasicEnergySourceInitialEnergyJ", DoubleValue(0.1));
EnergySourceContainer sources = basicSourceHelper.Install(networkNodes);
WifiRadioEnergyModelHelper radioEnergyHelper;
radioEnergyHelper.Set("TxCurrentA", DoubleValue(0.0174));
DeviceEnergyModelContainer deviceModels = radioEnergyHelper.Install(devices,
sources);

/** Enable Routing Protocol (AODV) **/
AodvHelper aodv;

/** Internet stack **/
InternetStackHelper internet;
internet.SetRoutingHelper(aodv);
internet.Install (networkNodes);

/** Assign Network Adresses **/
Ipv4AddressHelper ipv4;
NS_LOG_INFO ("Assign IP Addresses.");
ipv4.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer i = ipv4.Assign (devices);

/** Energy **/
// all sources are connected to node 0 (energy source)
Ptr<BasicEnergySource> basicSourcePtr = DynamicCast<BasicEnergySource>
(sources.Get(0));
Ptr<DeviceEnergyModel> basicRadioModelPtr = basicSourcePtr ->
FindDeviceEnergyModels ("ns3::WifiRadioEnergyModel").Get(0);
NS_ASSERT (basicRadioModelPtr != NULL);
basicRadioModelPtr -> TraceConnectWithoutContext("TotalEnergyConsumption",
MakeCallback(&TotalEnergy));

TypeId tid = TypeId::LookupByName ("ns3::UdpSocketFactory");

Ptr<Socket> recvSink;
 recvSink = Socket::CreateSocket (networkNodes.Get (0), tid);
 InetSocketAddress local = InetSocketAddress (Ipv4Address::GetAny (), 80);
 recvSink->Bind (local);
 recvSink->SetRecvCallback (MakeCallback (&ReceivePacket));

Ptr<Socket> source;
for(uint32_t count1 = 1; count1 < numNodes; count1++)
{

 source = Socket::CreateSocket (networkNodes.Get (count1), tid);
 InetSocketAddress remote = InetSocketAddress (Ipv4Address::GetBroadcast (),
80);
 source->SetAllowBroadcast (true);
 source->Connect (remote);
}

for(uint32_t count2 = 1; count2 < numNodes; count2++)
{
 Simulator::Schedule (Seconds(startTime), &GenerateTraffic, source,
packetSize, networkNodes.Get(count2), numPackets, interPacketInterval);
}

 Simulator::Stop (Seconds (180.0));
 Simulator::Run ();
```

```
 Simulator::Destroy();

NS_LOG_UNCOND ("Number of packets sent: " << numPackets*(numNodes - 1) <<
             " packets. \nNumber of packets received: " << received_packets
             << " packets. \nPackets received: " <<
             double(received_packets*100)/(numPackets*(numNodes - 1)) <<
             "%. \nAverage Delay: " << total_delay/received_packets << "
             seconds. \nThroughput: " <<
             double((received_packets*packetSize)/180)*8/1000 <<
             " kbps. \nTotal energy consumption: " <<
             total_energy_consumption << " J.\nEnd of Simulation");

outfile2 << "----Simulation Results----" << endl <<
             "Number of packets sent: " << numPackets*(numNodes - 1) << endl
             << "Number of packets received: " << received_packets << "
             packets. " << endl << "Packets received: " <<
             double(received_packets*100)/(numPackets*(numNodes - 1)) <<
             "%." << endl << "Average Delay: " <<
             total_delay/received_packets << " seconds." << endl <<
             "Throughput: " <<
             double((received_packets*packetSize)/180)*8/1000 << endl <<
             "Total energy consumption: " << total_energy_consumption << "
             J."<< endl;

outfile << "End of Simulation" << endl << endl;
outfile2 << "End of Simulation" << endl << endl <<
       "-----------------------------------------------------------" << endl <<
endl;

outfile.close();
outfile2.close();

received_packets = 0;
total_delay = 0;
total_energy_consumption = 0;

return 0;
}
```

8.3 Archaeological Site Scenario

```
/*3rd - Archaeological Site Scenario*/
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/mobility-module.h"
#include "ns3/config-store-module.h"
#include "ns3/wifi-module.h"
#include "ns3/energy-module.h"
#include "ns3/internet-module.h"
#include "ns3/aodv-helper.h"
#include "ns3/aodv-routing-protocol.h"
#include "ns3/delay-jitter-estimation.h"

#include <iostream>
#include <fstream>

NS_LOG_COMPONENT_DEFINE ("3rd - Archaeological Site Scenario");

using namespace ns3;
using namespace std;

uint32_t received_packets = 0;
long double total_delay = 0;
long double total_energy_consumption = 0;

void ReceivePacket (Ptr<Socket> socket)
{
 ofstream outfile;
 outfile.open("3rd_general.txt", ios::app);
 Ptr<Packet> packet;

 Address from;
 while (packet = socket->RecvFrom (from))
 {
       if (packet->GetSize () > 0)
       {
         received_packets++;
         DelayJitterEstimation delay;
         delay.RecordRx(packet);
         total_delay += delay.GetLastDelay().GetSeconds();
         InetSocketAddress iaddr = InetSocketAddress::ConvertFrom (from);

         NS_LOG_UNCOND ("--\nReceived one packet! Number of Packets until
         now: " << received_packets << "  From: " << iaddr.GetIpv4() << "
         Delay: " << delay.GetLastDelay().GetSeconds()  << " Packetsize "<<
         packet -> GetSize() << " port: " << iaddr.GetPort () << " at time =
         " << Simulator::Now ().GetSeconds () << "\n--");

         outfile << "Received one packet! Socket: "<< iaddr.GetIpv4 ()<< "
         port: " << iaddr.GetPort () << " at time = " << Simulator::Now
         ().GetSeconds () << "\n";

       (void) iaddr;
       }
}
 outfile.close();
}

static void GenerateTraffic (Ptr<Socket> socket, uint32_t pktSize, Ptr<Node>
n, uint32_t pktCount, Time pktInterval)
{
 Ptr<Packet> p = Create<Packet> (pktSize);
 DelayJitterEstimation delay;
 delay.PrepareTx(p);
 if (pktCount > 0)
 {
```

```
        socket->Send (p);
        Simulator::Schedule (pktInterval, &GenerateTraffic, socket, pktSize,
        n, pktCount - 1, pktInterval);
  }
  else
  {
        socket->Close ();
  }
}

void TotalEnergy (double oldValue, double totalEnergy)
{
 ofstream outfile;
 outfile.open("3rd_general.txt", ios::app);

NS_LOG_UNCOND ( Simulator::Now().GetSeconds() <<"s Total energy consumed by
radio = " << totalEnergy << "J");

outfile << Simulator::Now().GetSeconds() << "s Total energy consumed by
radio = " << totalEnergy << "J\n";

 outfile.close();
 total_energy_consumption = totalEnergy;
}


int main (int argc, char *argv[])
{
 SeedManager::SetSeed(4478562);
 bool verbose = false;
 double startTime = 10.0;
 uint32_t packetSize = 4096;
 long double interval = 0.015;
 std::string phyMode ("DsssRate1Mbps");

 uint32_t numNodes = atoi(argv[1]);      //Number of Nodes: 5, 10, 15, 20,
                                         25, 30
 uint32_t numPackets = atof(argv[2]);    //Number of Packets: 50, 100, 150,
                                         200, 250, 300, 350, 400, 450, 500

 ofstream outfile;
 ofstream outfile2;

 outfile.open("3rd_general.txt", ios::app);
 outfile << "Simulation Parameters" << endl
         << "Routing Protocl: AODV" << endl
         << "Number of Nodes:     " << numNodes << endl
        << "Numbet of Packets to sent from each node:   " << numPackets <<
      endl
         << "Packet Size:         " << packetSize << endl
         << "DsssRate:            " << phyMode << endl
         << "Interval:            " << interval << endl
        << "Data Rate:           " << ((packetSize/interval)*8)/1000 <<
        endl;

 outfile2.open("3rd_results.txt", ios::app);
 outfile2 << "----Simulation Parameters----" << endl
          << "Routing Protocl: AODV" << endl
          << "Number of Nodes:     " << numNodes << endl
          << "Numbet of Packets to sent from each node:   " << numPackets <<
endl
          << "Packet Size:         " << packetSize << endl
          << "DsssRate:            " << phyMode << endl
          << "Interval:            " << interval << endl
        << "Data Rate:           " << ((packetSize/interval)*8)/1000 <<
endl;

CommandLine cmd;
```

```
cmd.AddValue ("phyMode", "Wifi Phy mode", phyMode);
cmd.AddValue ("PpacketSize", "size of application packet sent", packetSize);
cmd.AddValue ("numPackets", "Total number of packets to send", numPackets);
cmd.AddValue ("startTime", "Simulation start time", startTime);
cmd.AddValue ("verbose", "Turn on all device log components", verbose);
cmd.Parse (argc, argv);

// Convert to time object
Time interPacketInterval = Seconds (interval);
// disable fragmentation for frames below 2200 bytes
Config::SetDefault ("ns3::WifiRemoteStationManager::FragmentationThreshold",
StringValue ("2200"));
// turn off RTS/CTS for frames below 2200 bytes
Config::SetDefault ("ns3::WifiRemoteStationManager::RtsCtsThreshold",
StringValue ("2200"));
// Fix non-unicast data rate to be the same as that of unicast
Config::SetDefault ("ns3::WifiRemoteStationManager::NonUnicastMode",
StringValue (phyMode));

NodeContainer networkNodes;
networkNodes.Create (numNodes);

// The below set of helpers will help us to put together the wifi NICs we
want
WifiHelper wifi;
if (verbose)
{
 wifi.EnableLogComponents ();
}

wifi.SetStandard (WIFI_PHY_STANDARD_80211b);

/** Wifi PHY **/

/***************************************************************************/

YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default ();
wifiPhy.Set ("RxGain", DoubleValue (1));
wifiPhy.Set ("TxGain", DoubleValue (1));
wifiPhy.Set ("CcaMode1Threshold", DoubleValue (0.0));

/***************************************************************************/

/** wifi channel **/
YansWifiChannelHelper wifiChannel;
wifiChannel.SetPropagationDelay ("ns3::ConstantSpeedPropagationDelayModel");
wifiChannel.AddPropagationLoss ("ns3::FriisPropagationLossModel");

// create wifi channel
Ptr<YansWifiChannel> wifiChannelPtr = wifiChannel.Create ();
wifiPhy.SetChannel (wifiChannelPtr);

/** MAC layer **/

// Add a non-QoS upper MAC, and disable rate control
NqosWifiMacHelper wifiMac = NqosWifiMacHelper::Default ();
wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager",
                             "DataMode", StringValue (phyMode),
                             "ControlMode", StringValue (phyMode));

// Set it to ad-hoc mode

wifiMac.SetType ("ns3::AdhocWifiMac");

/** install PHY + MAC **/
NetDeviceContainer devices = wifi.Install (wifiPhy, wifiMac, networkNodes);

/** mobility **/
```

```
MobilityHelper mobility;

mobility.SetMobilityModel ("ns3::GaussMarkovMobilityModel",
   "Bounds", BoxValue (Box (0, 1000, 0, 1000, 0, 500)),
   "TimeStep", TimeValue (Seconds (0.5)),
   "Alpha", DoubleValue (0.95),
   "MeanVelocity", RandomVariableValue (UniformVariable (0, 2)),
   "MeanDirection", RandomVariableValue (UniformVariable (0, 6.283185307)),
   "MeanPitch", RandomVariableValue (UniformVariable (0.05, 0.05)),
   "NormalVelocity", RandomVariableValue (NormalVariable (0.0, 1.0, 10.0)),
   "NormalDirection", RandomVariableValue (NormalVariable (0.0, 0.2, 0.4)),
   "NormalPitch", RandomVariableValue (NormalVariable (0.0, 0.02, 0.04)));

mobility.SetPositionAllocator ("ns3::RandomBoxPositionAllocator",
   "X", RandomVariableValue (UniformVariable (0, 1000)),
   "Y", RandomVariableValue (UniformVariable (0, 1000)),
   "Z", RandomVariableValue (UniformVariable (0, 500)));

mobility.Install (networkNodes);

/* Energy Model*/
BasicEnergySourceHelper basicSourceHelper;
basicSourceHelper.Set("BasicEnergySourceInitialEnergyJ", DoubleValue(0.1));
EnergySourceContainer sources = basicSourceHelper.Install(networkNodes);
WifiRadioEnergyModelHelper radioEnergyHelper;
radioEnergyHelper.Set("TxCurrentA", DoubleValue(0.0174));
DeviceEnergyModelContainer deviceModels = radioEnergyHelper.Install(devices,
sources);

/** Enable Routing Protocol (AODV) **/
AodvHelper aodv;

/** Internet stack **/
InternetStackHelper internet;
internet.SetRoutingHelper(aodv);
internet.Install (networkNodes);

/** Assign Network Adresses **/
Ipv4AddressHelper ipv4;
NS_LOG_INFO ("Assign IP Addresses.");
ipv4.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer i = ipv4.Assign (devices);

/** Energy **/
// all sources are connected to node 0 (energy source)
Ptr<BasicEnergySource> basicSourcePtr = DynamicCast<BasicEnergySource>
(sources.Get(0));
Ptr<DeviceEnergyModel> basicRadioModelPtr = basicSourcePtr ->
FindDeviceEnergyModels ("ns3::WifiRadioEnergyModel").Get(0);
NS_ASSERT (basicRadioModelPtr != NULL);
basicRadioModelPtr -> TraceConnectWithoutContext("TotalEnergyConsumption",
MakeCallback(&TotalEnergy));

TypeId tid = TypeId::LookupByName ("ns3::UdpSocketFactory");

Ptr<Socket> recvSink;
for(uint32_t count1 = 1; count1 < numNodes; count1++)
{
 recvSink = Socket::CreateSocket (networkNodes.Get (count1), tid);
 InetSocketAddress local = InetSocketAddress (Ipv4Address::GetAny (), 80);
 recvSink->Bind (local);
 recvSink->SetRecvCallback (MakeCallback (&ReceivePacket));
}

 Ptr<Socket> source = Socket::CreateSocket (networkNodes.Get (0), tid);
 InetSocketAddress remote = InetSocketAddress (Ipv4Address::GetBroadcast (),
80);
 source->SetAllowBroadcast (true);
```

```
 source->Connect (remote);

 Simulator::Schedule (Seconds(startTime), &GenerateTraffic, source,
 packetSize, networkNodes.Get(0), numPackets, interPacketInterval);

 Simulator::Stop (Seconds (180.0));
 Simulator::Run ();
 Simulator::Destroy();

NS_LOG_UNCOND ("Number of packets sent: " << numPackets*(numNodes - 1) <<
             " packets. \nNumber of packets received: " << received_packets
             <<" packets. \nPackets received: " <<
             double(received_packets*100)/(numPackets*(numNodes - 1)) <<
             "%. \nAverage Delay: " << total_delay/received_packets <<
             " seconds. \nThroughput: " <<
             double((received_packets*packetSize)/180)*8/1000 <<
             " kbps. \nTotal energy consumption: " <<
             total_energy_consumption <<
             " J.\nEnd of Simulation");

outfile2 << "----Simulation Results----" << endl <<
             "Number of packets sent: " << numPackets*(numNodes - 1) << endl
             << "Number of packets received: " << received_packets << "
             packets. " << endl <<
             "Packets received: " <<
             double(received_packets*100)/(numPackets*(numNodes - 1)) <<
             "%." << endl <<
             "Average Delay: " << total_delay/received_packets << "
             seconds." << endl << "Throughput: " <<
             double((received_packets*packetSize)/180)*8/1000 << endl <<
             "Total energy consumption: " << total_energy_consumption << "
             J."<< endl;

outfile << "End of Simulation" << endl << endl;
outfile2 << "End of Simulation" << endl << endl <<
       "----------------------------------------------------------" << endl <<
endl;

outfile.close();
outfile2.close();

received_packets = 0;
total_delay = 0;
total_energy_consumption = 0;

return 0;
}
```