Πανεπιστήμιο Θεσσαλίας

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Διπλωματική εργασία

Θέμα:

« Ποιοτική σύγκριση και αξιολόγηση πρωτοκόλλων δρομολόγησης σε ασύρματα δίκτυα πλέγματος »

Βουλγαράκης Δημήτριος

UNIVERSITY OF THESSALY

<u>Επιβλέπων καθηγητής:</u>

Κοράκης Αθανάσιος (Λέκτορας Καθηγητής)

<u>Συνεπιβλέπων καθηγητής:</u>

Αργυρίου Αντώνιος (Λέκτορας Καθηγητής)

Βόλος, 2015

University of Thessaly

Department of Electrical & Computer Engineering

Thesis

Title:

« Evaluation of routing algorithms on top of wireless mesh networks »

Voulgarakis Dimitrios

UNIVERSITY OF THESSALY

Supervisor professor: Korakis Athanasios

Co-supervisor professor: Argyriou Antonios

Volos, 2015

Ευχαριστίες

Η παρούσα Διπλωματική εργασία πραγματοποιήθηκε στο τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών Βόλου. Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή και λέκτορα του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, κ. Κοράκη Αθανάσιο, που μου έδωσε την ευκαιρία να ασχοληθώ με αυτό το θέμα. Επίσης, θα ήθελα να ευχαριστήσω θερμά τον υποψήφιο διδάκτορα, Χούμα Κωνσταντίνο, που καθ' όλη τη διάρκεια της διπλωματικής εργασίας ήταν εκεί και με καθοδηγούσε με χρήσιμες συμβουλές και προτάσεις. Τέλος, ένα μεγάλο ευχαριστώ στην οικογένεια και τους φίλους μου, που με στήριζαν όλο αυτό το διάστημα.

CONTENTS

Ευχαριστίες 3

Περίληψη 6

Abstract 7

1 Introduction 8

2 Dynamic routing protocols 8

  2.1 Proactive protocols 10

  2.2 Reactive protocols 10

3 AODV: Ad hoc On-demand Distance Vector 11

  3.1 Introduction 11

  3.2 Overview 11

  3.3 AODV Operation 12

  3.4 Route Discovery 12

    3.4.1 Generating and Forwarding a RREQ 12

    3.4.2 Generating a RREP 13

    3.4.3 Generating Gratuitous RREP 13

    3.4.4 Receiving and Forwarding RREP 13

  3.5 Route Maintenance 14

  3.6 Hello Messages 14

  3.7 Message Formats 15

    3.7.1 Route Request (RREQ) Message Format 15

    3.7.2 Route Reply (RREP) Message Format 16

    3.7.3 Route Error (RERR) Message Format 17

4 OLSR: Optimized Link State Routing 19

  4.1 Introduction 19

  4.2 Overview 19

  4.3 OLSR Operation 20

  4.4 Link Sensing 20

  4.5 Efficient Control Flooding Using MPRs 21

4.6 Optimal Path Calculation Using Shortest Path Algorithm 21
4.7 Packet Format 22
5 Experimentation 25
5.1 Experimentation in Wireless Testbeds 25
5.2 NITOS Testbed 25
5.3 Outdoor Nodes 26
5.4 Experiment Parameters and Setup 27
5.5 Experimental results on single-path topology 28
5.6 Experimental results on multi-path topology 32
References 36

Περίληψη

Σκοπός της διπλωματικής εργασίας είναι η πειραματική σύγκριση δύο πρωτοκόλλων δρομολόγησης σε πραγματικά δίκτυα πλέγματος. Συγκεκριμένα, αξιολογούμε την απόδοση του OLSR, το οποίο είναι proactive πρωτόκολλο και του AODV το οποίο είναι ένα reactive πρωτόκολλο. Όλα τα πειράματα πραγματοποιήθηκαν στους εξωτερικούς κόμβους του NITOS. Για την αξιολόγηση των πρωτοκόλλων χρησιμοποιήθηκαν δύο ειδών τοπολογίες, μία μονού μονοπατιού και μία πολλαπλών διαδρομών. Οι μετρικές που χρησιμοποιήθηκαν για την αξιολόγηση ήταν: η αναλογία παράδοσης πακέτων (PDR), η μέση καθυστέρηση του κάθε πρωτοκόλλου και το μέσο throughput της μετάδοσης.

Abstract

The purpose of this thesis is to report an experimental comparison, on real ad hoc networks, between two routing protocols. Specifically, we evaluate the performance of Optimized Link State Routing Protocol (OLSR) which is a proactive protocol and Ad hoc On-Demand Distance Vector (AODV) which is a reactive protocol. We compared them in NITOS outdoor testbed nodes. To evaluate routing protocols' performance we used two configurations: i) a single-path string network, ii) a mesh multi-path network. In the performance comparison we analyzed metrics such as Packet Delivery Ratio (PDR), average end-to-end delay and mean throughput.

1 Introduction

Routing is the act of moving information from a source to a destination in a network. Along the way, at least one intermediate node typically is encountered. The topic of routing has been covered in computer science literature for more than two decades, but routing achieved commercial popularity as late as the mid-1980s. The primary reason for this time lag is that networks in the 1970s were simple, homogeneous environments. Only relatively recently has large-scale internetworking become popular. The main tasks of a routing protocol are topology discovery and topology maintenance. The topology discovery phase is usually periodic advertisements about a node's location as well as exchange of certain request-reply messages. However, topology maintenance is detecting link connectivity breaks and fix the broken link, if possible. Routing protocols use several metrics to calculate the best path for routing the packets to its destination. These metrics 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. These route information varies from one routing algorithm to another.
Routing is mainly classified into static routing and dynamic routing. Static routing refers to the routing strategy being stated manually or statically, in the router. Static routing maintains a routing table usually written by a network's administrator. The routing table doesn't depend on the state of the network status. Dynamic routing refers to the routing strategy that is being learnt by a routing protocol. This routing mainly depends on the state of the network i.e., the routing table is affected by the activeness of the destination. The major disadvantage of static routing is that if a new router is added or removed in the network then it is the responsibility of the administrator to make the necessary changes in the routing tables. But this is not the case with dynamic routing as each router announces its presence by flooding the information packet in the network so that every router within the network learn about the newly added or removed router and its entries.

2 Dynamic routing protocols

Dynamic routing protocols are classified depending on what the routers tell each other and how they use the information to form their routing tables. Most of the protocols available in the networks fit into one of the two categories. The two main categories of dynamic routing protocols in packet switching networks are distance vector and link state. Both categories use the shortest path algorithm to find the best next hop neighbor. Link state routing is performed by every switching node in the network. The basic concept of link-state routing is that every node constructs a map of the connectivity to the network, in the form of a graph, showing which nodes are connected to which other nodes. Each node then independently calculates the next best logical path from it to every possible destination in the network. The collection of best paths will then form the node's routing table. In distance vector protocols, each router over the network send the neighboring routers, the information about destination that it knows how to reach. Furthermore the routers send two pieces of information: i) the router tells how far it thinks the destination is ii) it tells in what direction (vector) to use to get to the destination. When the router receives the information from the others, it could then develop a table of destination addresses, distances and associated neighboring routers. From this table then, selects the shortest route to the destination. Using a distance vector protocol, the router simply forwards the packet to the neighboring host (or destination) with the available shortest path in the routing table and assumes that the receiving router will know how to forward the packet beyond that point. Compared to link-state protocols, distance-vector routing protocols have less computational complexity and message overhead. Routing protocols can be classified in many ways depending on routing algorithm and network organization. Depending on the network structure, classification can be flat, hierarchical routing and geographic position assisted, while based on routing strategy used, they can categorized as table driven or on-demand as shown in Figure 2.1.

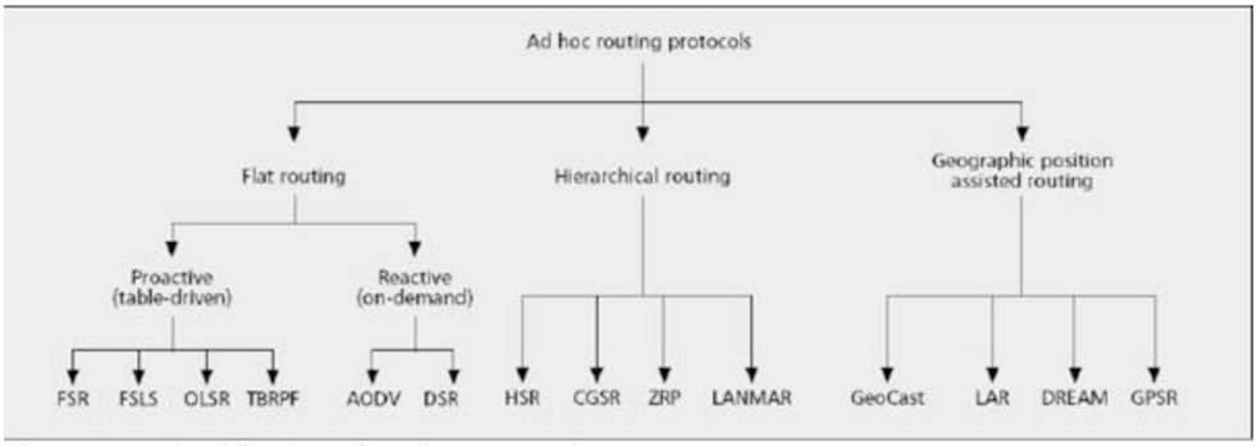


Figure 2.1: Classification of routing protocols

2.1 Proactive protocols

Proactive routing protocols attempt to maintain consistent and updated routing information for every pair of network nodes by proactively propagating route updates at fixed rime intervals (periodically). Proactive routing is also known as table driven routing. The main advantage of such an algorithm is that there is no delay in establishing a communication session and routing table is updated as soon as there is a change in topology. Disadvantages are additional control traffic to keep the routing table up to date irrespective of whether all the routes are used in a session or not. Proactive protocols are not suitable for larger networks, as they need to maintain node entries for each and every node in the routing table of every node. This causes more overhead in the routing table leading to consumption of more bandwidth. Example of proactive algorithm is Optimized Link State Routing (OLSR).

2.2 Reactive protocols

Reactive routing is also called on-demand routing as the routes are established only when needed to forward the data packets. They don't maintain routing information or routing activity at the network nodes if there is no communication. If a node wants to send packets to another node then this protocol searches for the route in an on-demand manner and establishes the connection in order to transmit and receive the packets. The route discovery usually occurs by flooding the route request packets throughout the network. This algorithm has significantly low routing overhead when the traffic is light and network is less dynamic, since there is no need maintain the routes when there is no data traffic. The major disadvantages are longer delay in establishing the routes for forwarding the data and excessive flooding of the control messages that may lead to network block. Example of reactive routing are Ad hoc On-demand Distance Vector (AODV).

3 AODV: Ad hoc On-demand Distance Vector

3.1 Introduction

Ad hoc On-demand Distance Vector (AODV), as the name suggests is an on-demand protocol designed for mobile ad hoc networks. This protocol responds quickly to changing link conditions and link breakages. The nodes mark the routes as invalid whenever there is a link breakage. AODV does not require a node to maintain routes to destinations that are not in active communication. The operation of AODV is loop-free, by using destination sequence numbers. These also allow nodes to use the most recent route to a destination. The routing table information includes the destination address and the next hop address with the number of hops required to reach the destination. Also, the most recent destination sequence number associated with destination and lifetime of the route is stored in the table. If during the lifetime, the route is not used, the routing table entry is discarded.

3.2 Overview

The message types defined by AODV are Route Request (RREQ), Route Reply (RREP) and Route Error (RERR). As long as the endpoints of a communication connection have valid routes to each other, AODV does not play any role. When a route to a new destination is needed, a node broadcasts the RREQ message to find a route. A route is found when the RREQ reaches the destination itself, or an intermediate node that has a "fresh enough" route to the destination. The route is made available by unicasting the RREP message back to the source of the destination. Since, each node that receives the RREQ caches the route back to the source, the RREP can be unicasted to the origination of the RREQ. The link status of active routes is continuously monitored for any link breakage. When a link breaks, RERR message is propagated down the route to notify the affected nodes about the loss of link. The purpose of RERR message is to indicate which destinations are now unreachable because of the link breakage. Each node keeps a "precursor list" that contains the IP address for each of its neighbors that are likely to use it as a next hop towards each destination.

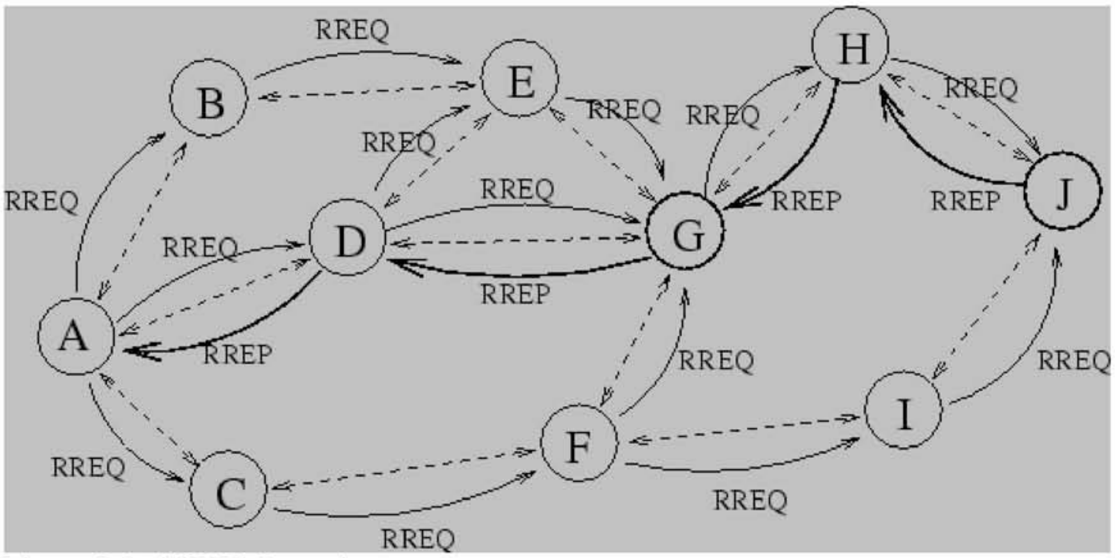


Figure 3.1: AODV Operation

3.3 AODV Operation

The basic operation of AODV can be divided into three phases:

- Route discovery
- Route maintenance
- Hello message

3.4 Route Discovery

3.4.1 Generating and Forwarding a RREQ

When a destination is previously unknown to the node or the route to the destination is no longer valid, the node propagates a RREQ. The Destination Sequence Number (DSN) is the last known DSN for this destination. If no sequence number is known, unknown sequence number flag is set. After broadcasting a RREQ, the node waits for a RREP. If the RREP is not received with NET TRAVERSAL TIME milliseconds, the node rebroadcast the RREQ, up to a maximum of RREQ RETRIES times, which is set to 2. When a node receives a RREQ, it creates or updates the route to the previous hop without valid sequence number and then checks if it has received the RREQ with same originator IP address and RREQ ID. If such a RREQ has been received, the node removes the newly received RREQ. If the RREQ received is not removed, the intermediate node searches for a reverse route to the Originator IP address. If the reverse route already exists, it is updated only if the originator sequence number is higher than the destination sequence number of the originator IP address in the routing table or if the sequence numbers are equal but the hop count in RREQ is smaller than the existing hop count in the routing table. The RREQ is rebroadcasted if the active route does not exist in its routing table or if the existing DSN is smaller than the DSN field of the RREQ. The Time to Live (TTL) in the outgoing RREQ is decremented by one and the hop count field is incremented by one to account for the new hop through the intermediate node.

3.4.2 Generating a RREP

A node generates a RREP if:

- ➢ It is itself the destination.
- ➢ It has an active route to the destination

The RREP is unicast to the next hop towards the originator of the RREQ. As the RREP is propagated, the hop count field in RREP is incremented by one at each hop.

3.4.3 Generating Gratuitous RREP

When a node receives a RREQ and responds with a RREP, it discards the RREQ. If the RREQ has 'G' flag set, and the intermediate node replies to the RREQ, it unicasts a gratuitous RREP to the destination node.

3.4.4 Receiving and Forwarding RREP

When a node receives a RREP, it searches for a route to the previous hop. Once the route to the destination is created or updated in the route table, the route is marked active and the next hop is assigned to the node from which RREP was received. Therefore, the node can use this route to forward the data to the destination. If the node is not indicated by the Originator IP address in RREP, then the RREP packet is forwarded towards the next hop, selected based on the route table entry. When a node forwards a RREP, the precursor list for the corresponding destination node is updated by adding to it the next hop node to which the RREP is forwarded.

3.5 Route Maintenance

A RERR message is generated in following three scenarios:

1. While transmitting data, if a node detects a link break for the next hop of an active route, the node makes a list of unreachable destinations containing the unreachable neighbors and other destinations that use the unreachable neighbor as next hop.
2. A node receives a packet for the node it does not have an active route in its routing table. For this, there is only one unreachable destination.
3. If a node receives a RERR from a neighbor for one or more active routes. The list consists of destinations in RERR for which there exists a corresponding entry in the local routing table that has the transmitter of the received RERR as the next hop.

AODV has a mechanism called Local Repair by which the upstream node of the broken link attempts to repair the link locally instead of sending RERR. The node initiating the local repair follows the route discovery phase. If the node does not receive a RREP, it then transmits a RERR message for that destination. The process of local repair may result in greater path lengths to the destinations for which local repair was initiated.

3.6 Hello Messages

AODV maintains network connectivity by reception of broadcast hello messages on the active routes. A node that is a part of an active route periodically broadcasts hello messages, which are RREP messages with TTL=1, to announce its presence. If a node does not receive a hello message within a specified interval, then it is assumed that the neighbor node is no longer in transmission range and the connectivity to this node has been lost. Whenever a node receives a hello message from a neighbor, the node checks if it has an active route to the neighbor and if not, it creates one. If the route already exists, the TTL for the route is increased.

3.7 Message Formats

3.7.1 Route Request (RREQ) Message Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |J|R|G|D|U|   Reserved          |   Hop Count   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            RREQ ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination IP Address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Destination Sequence Number                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Originator IP Address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Originator Sequence Number                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format of the Route Request message is illustrated above, and contains the following fields:

Type	1
J	Join flag, reserved for multicast.
R	Repair flag, reserved for multicast.
G	Gratuitous RREP flag, indicates whether a gratuitous RREP should be unicast to the node specified in the Destination IP Address field
D	Destination only flag, indicates only the destination may respond to this RREQ
U	Unknown sequence number; indicates the destination sequence number is unknown
Reserved	Sent as 0, ignored on reception.
Hop Count	The number of hops from the Originator IP Address to the node handling the request.
RREQ ID	A sequence number uniquely identifying the particular RREQ when taken in conjunction with the originating node's IP address.

Destination IP Address
The IP address of the destination for which a route is desired.

Destination Sequence Number
The latest sequence number received in the past by the originator for any route towards the destination.

Originator IP Address
The IP address of the node which originated the Route Request.

Originator Sequence Number
The current sequence number to be used in the route entry pointing towards the originator of the route request.

3.7.2 Route Reply (RREP) Message Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |R|A|    Reserved     |Prefix Sz|   Hop Count   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Destination IP address                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Destination Sequence Number                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Originator IP address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Lifetime                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format of the Route Reply message is illustrated above, and contains the following fields:

Type	2
R	Repair flag, used for multicast.
A	Acknowledgment required
Reserved	Sent as 0, ignored on reception.

Prefix Size If nonzero, the 5-bit Prefix Size specifies that the indicated next hop may be used for any nodes with the same routing prefix (as defined by the Prefix Size) as the requested destination.

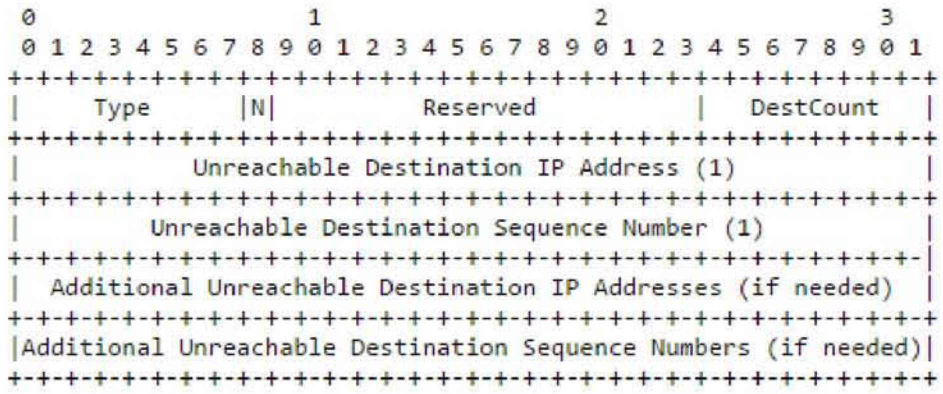
Hop Count The number of hops from the Originator IP Address to the Destination IP Address. For multicast route requests this indicates the number of hops to the multicast tree member sending the RREP.

Destination IP Address
The IP address of the destination for which a route is supplied.

Destination Sequence Number
The destination sequence number associated to the route.

Originator IP Address
The IP address of the node which originated the RREQ for which the route is supplied.
Lifetime The time in milliseconds for which nodes receiving the RREP consider the route to be valid.

3.7.3 Route Error (RERR) Message Format



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |N|          Reserved           |   DestCount   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Unreachable Destination IP Address (1)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Unreachable Destination Sequence Number (1)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
|  Additional Unreachable Destination IP Addresses (if needed)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Additional Unreachable Destination Sequence Numbers (if needed)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format of the Route Error message is illustrated above, and contains the following fields:

Type 3

N No delete flag, set when a node has performed a local repair of a link, and upstream nodes should not delete the route.

Reserved Sent as 0, ignored on reception.

DestCount The number of unreachable destinations included in the message, MUST be at least 1.

Unreachable Destination IP Address
The IP address of the destination that has become unreachable due to a link break.

Unreachable Destination Sequence Number
The sequence number in the route table entry for the destination listed in the previous Unreachable Destination IP Address field.

4 OLSR: Optimized Link State Routing

4.1 Introduction

The Optimized Link State Routing (OLSR) is a proactive table driven protocol for mobile ad hoc networks. It eases efficient flooding of control messages throughout the network by using selected nodes called MultiPoint Relays (MPRs). MPRs are selected by each node and are used to forward control messages resulting in a distributed operation of the protocol. In addition to this, a node continuously maintains routes to all destinations in the network, thus making the protocol suited for traffic pattern that is random and sporadic.

4.2 Overview

The protocol is an optimization of classical link state routing algorithm and uses the concept of MultiPoint Relays (MPRs). The problem of flooding the network with control messages is overcome by the MPR nodes. A node periodically exchanges hello messages to discover its two-hop neighbor. Using this information, each node selects a set of MPRs, which are one-hop neighbors. A node selects MPR such that there exists a path to each of its two-hop neighbors via a node selected as a MPR. The main responsibility of MPR is to forward the control messages throughout the network, thus minimizing the number of transmissions. MPRs periodically broadcast the control information, there by announcing the reachability to the nodes that have selected it as a MPR. A node uses this information to determine next hop destinations for all the nodes in the network using the shortest path algorithm. Because of its proactive nature, routing overhead is generally greater than that of a reactive protocol.

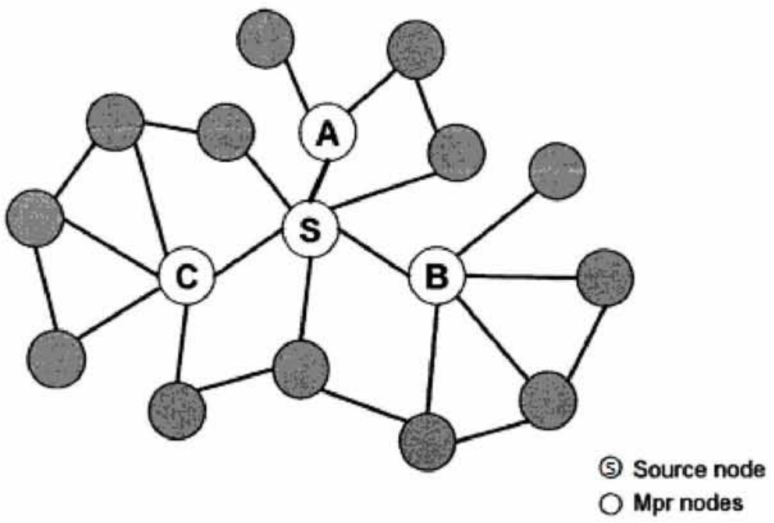


Figure 4.1: Multipoint relays selection

4.3 OLSR Operation

The core functioning of OLSR can be divided into three processes namely:

1. Link Sensing
2. Efficient control flooding using MPR
3. Optimal path calculation using shortest path algorithm.

4.4 Link Sensing

Nodes periodically exchanges HELLO message with each other. A hello message mainly consists of link information and neighborhood information. The three important tasks performed by hello message exchange are: i) link sensing ii) neighbor detection iii) MPR selection signaling. For neighbor and link sensing, a hello message typically comprised of list of links and list of one-hop symmetric neighbors. A hello message is broadcasted by a node to its neighbors and is never forwarded by other nodes. On reception of a hello message, a node performs a link sensing, neighbor detection and MPR selection set population. Each node in the network selects its MPR set. MPR set is elected based on the rule: MPR (N) is selected such that all two-hop neighbors of N are covered by (one-hop neighbors) of MPR (N). Smaller the MPR set, minimum is its protocol overhead.

4.5 Efficient Control Flooding Using MPRs

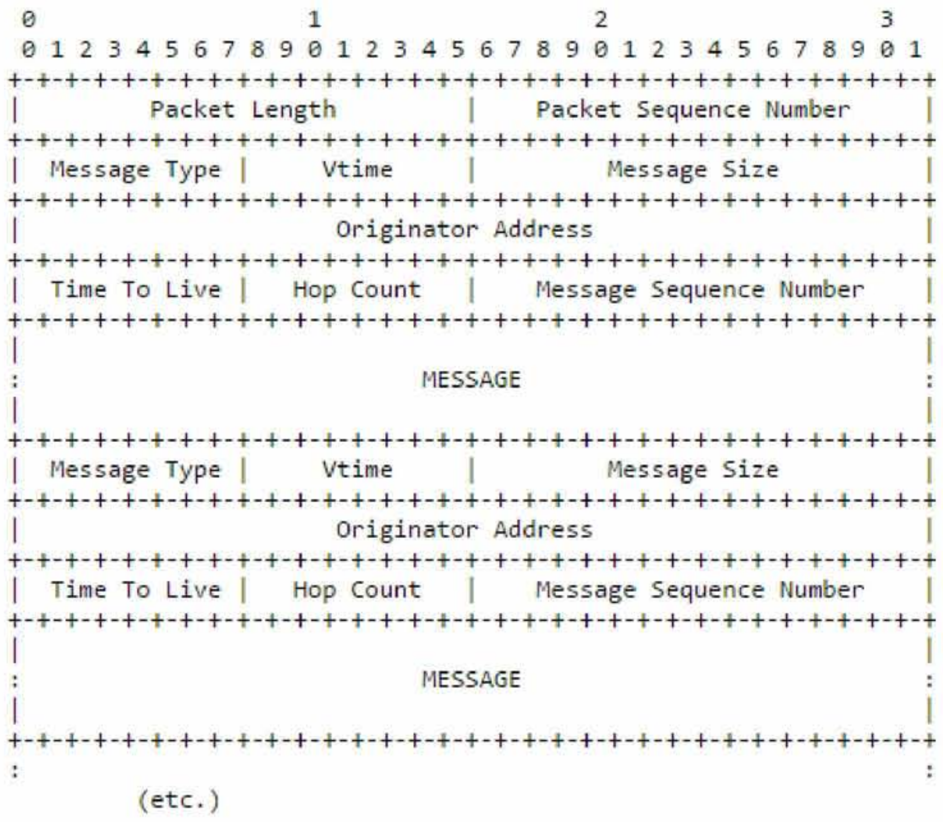
Due to the proactive nature of OLSR, each node maintains the partial topology graph of the network. This information is extracted from Topology Control (TC) messages and is then used for calculating the shortest paths to destinations. A MPR node broadcasts a TC message periodically that is propagated across the network using the other MPR nodes. A TC message contains MPR selector set of the source of the message and is forwarded by MPR if and only if it received the message for the first time by that node and it is in the MPR set of the previous hop node. This controlled flooding results in minimized retransmissions.

4.6 Optimal Path Calculation Using Shortest Path Algorithm

A routing table is maintained by every node, which is updated whenever a change in the topology is detected. To populate a routing table, shortest path algorithm is used on the partial topology graph obtained from TC messages. It is important to note that OLSR is not involved in forwarding of data packets.

4.7 Packet Format

The basic layout of any packet in OLSR is as follows (omitting IP and UDP headers):



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Packet Length         |    Packet Sequence Number     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Message Type |     Vtime     |         Message Size          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Originator Address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time To Live |   Hop Count   |    Message Sequence Number    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                            MESSAGE                            :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Message Type |     Vtime     |         Message Size          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Originator Address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time To Live |   Hop Count   |    Message Sequence Number    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                            MESSAGE                            :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
         (etc.)
```

Packet Header

Packet Length

The length (in bytes) of the packet

Packet Sequence Number

The Packet Sequence Number (PSN) MUST be incremented by one each time a new OLSR packet is transmitted. "Wrap-around" is handled as described in section 19. A separate Packet Sequence Number is maintained for each interface such that packets transmitted over an interface are sequentially enumerated.

The IP address of the interface over which a packet was transmitted is obtainable from the IP header of the packet. If the packet contains no messages (i.e., the Packet Length is less than or equal to the size of the packet header), the packet MUST silently be discarded.

For IPv4 addresses, this implies that packets, where the Packet Length < 16 MUST silently be discarded.

Message Header

Message Type

This field indicates which type of message is to be found in the "MESSAGE" part. Message types in the range of 0-127 are reserved for messages in this document and in possible extensions.

Vtime

This field indicates for how long time after reception a node MUST consider the information contained in the message as valid, unless a more recent update to the information is received. The validity time is represented by its mantissa (four highest bits of Vtime field) and by its exponent (four lowest bits of Vtime field). In other words:

$$\text{validity time} = C*(1+a/16)* 2^b \quad [\text{in seconds}]$$

where a is the integer represented by the four highest bits of Vtime field and b the integer represented by the four lowest bits of Vtime field. The proposed value of the scaling factor C is specified in section 18.

Message Size

This gives the size of this message, counted in bytes and measured from the beginning of the "Message Type" field and until the beginning of the next "Message Type" field (or - if there are no following messages - until the end of the packet).

Originator Address

This field contains the main address of the node, which has originally generated this message. This field SHOULD NOT be confused with the source address from the IP header, which is changed each time to the address of the intermediate interface which is re-transmitting this message. The Originator Address field MUST *NEVER* be changed in retransmissions.

Time To Live

This field contains the maximum number of hops a message will be transmitted. Before a message is retransmitted, the Time To Live MUST be decremented by 1. When a node receives a message with a Time to Live equal to 0 or 1, the message MUST NOT be retransmitted under any circumstances. Normally, a node would not receive a message with a TTL of zero.

Thus, by setting this field, the originator of a message can limit the flooding radius.

Hop Count

This field contains the number of hops a message has attained. Before a message is retransmitted, the Hop Count MUST be incremented by 1.

Initially, this is set to '0' by the originator of the message.

Message Sequence Number

While generating a message, the "originator" node will assign a unique identification number to each message. This number is inserted into the Sequence Number field of the message. The sequence number is increased by 1 (one) for each message originating from the node. "Wrap-around" is handled as described in section 19. Message sequence numbers are used to ensure that a given message is not retransmitted more than once by any node.

5 Experimentation

5.1 Experimentation in Wireless Testbeds

A testbed is a platform for experimentation of large development projects. A wireless network testbed consists of the hardware and software needed, in order to give researchers a wide range of environments in which to develop, debug, and evaluate their wireless network systems and protocols, in realistic conditions.

5.2 NITOS Testbed

NITOS (Network Implementation Testbed using Open Source code) is a testbed created by the Network Implementation Testbed Laboratory of the Computer and Communication Engineering Department at University of Thessaly, in collaboration with the Centre for Research and Technology Hellas (CERTH). The NITOS platform is open to any researchers who would like to test their protocols in a real-time wireless network. They are given the opportunity to implement their protocols and study their behavior in a custom environment. The testbed's capabilities are constantly extended. Up to now it consists of 45 nodes of different types. More specifically there are 10 Orbit nodes, 15 diskless nodes and 20 custom made outdoor nodes. For the purpose of this thesis we used the outdoor nodes.

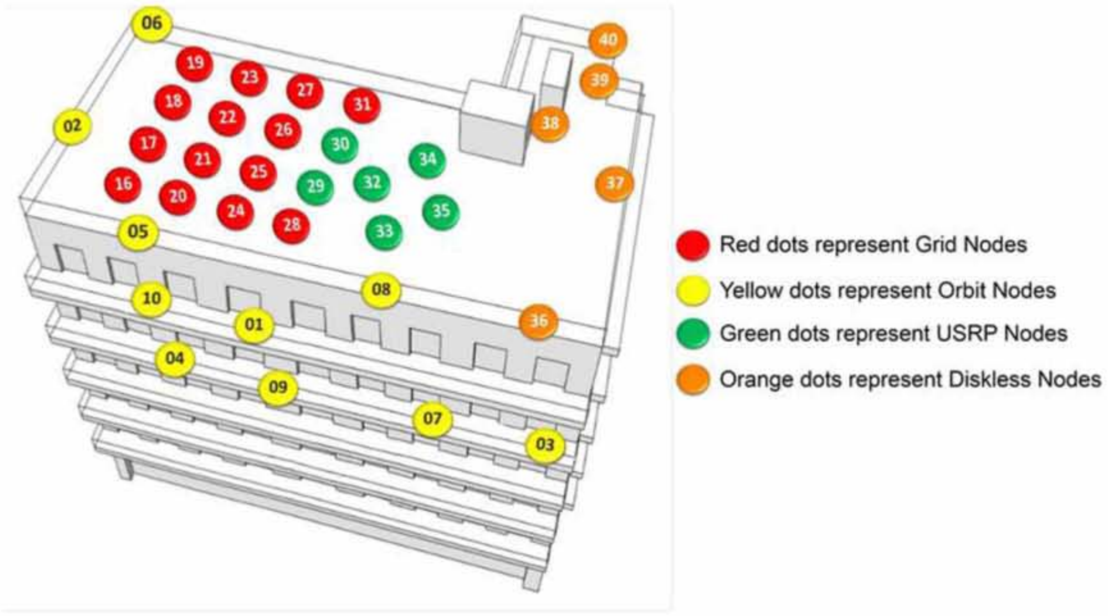

06
19
23
27
31
40
39
38
18
22
26
30
34
02
17
21
25
29
32
35
37
16
20
24
28
33
05
08
10
01
36
04
09
07
03
Red dots represent Grid Nodes
Yellow dots represent Orbit Nodes
Green dots represent USRP Nodes
Orange dots represent Diskless Nodes

Figure 5.1: NITOS nodes

5.3 Outdoor Nodes

The outdoor nodes have been developed by UTH in order to enable experimentation that requires high processing power. They feature 2-core Intel CPU, a new generation solid state drive, a USB web camera and are also equipped with 802.11 a/b/g and 802.11 a/b/g/n wireless interfaces. Furthermore, they are equipped with a Chassis Manager (CM) card in order to enable remote control of their operational status. Another feature of the attached CM card is the support of real time sensor measurements, gathered through their integrated light, temperature and humidity sensors. Detailed specifications of the Outdoor nodes can be found in the following table.

Table 1: Outdoor nodes specifications

Motherboard	Features 2 Gigabit network interfaces and supports 2 wireless interfaces
CPU	Intel Core 2 Duo P8400 2.26 Ghz
RAM	2G DDR3
Wireless interfaces	Atheros 802/11 a/b/g & Atheros 802.11 a/b/g/n (MIMO)
Chassis Manager Card	UTH's CM card
Storage	Solid state drive
Power supply	350 Watt mini-ATX
Antennas	Multi-band 5dbi, operates both on 2,4Ghz & 5Ghz
Pigtails	High quality pigtails (UFL to RP-SMA)

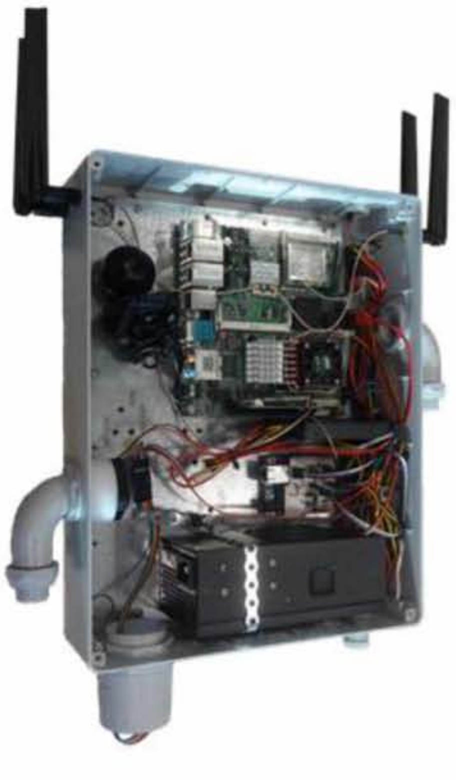


Figure 5.2: NITOS outdoor nodes

5.4 Experiment Parameters and Setup

AODV implementation developed by Uppsala University called AODV-UU version number 0.9.6 was used. For OLSR, OLSRd implementation version number 0.6.6.2 was used. Both these implementations are freely available on the internet. To evaluate routing protocols performance we mainly used two configurations: i) a mesh multi-path network, ii) a single-path string network. To have a meaningful comparison between them, we introduced some traffic at the application layer using the ping utility. In the performance comparison we analyzed i) Packet Delivery Ratio (PDR) at the application level which is the ratio of application packets delivered to the destination and the total number of generated packets ii) Average end-lo-end delay introduced in data transfer measured at application level. Also IPERF traffic generator used to generate UDP traffic flows. IPERF is a network performance tool that measures TCP and UDP bandwidth performances. The UDP traffic was used to capture the mean throughput depending on the packet size. Measurements during the experiments were collected and evaluated using Wireshark.

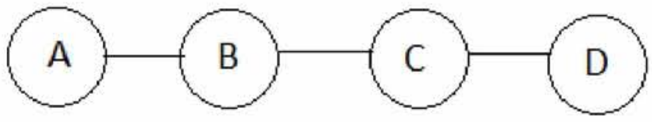


Figure 5.3: Single-path topology

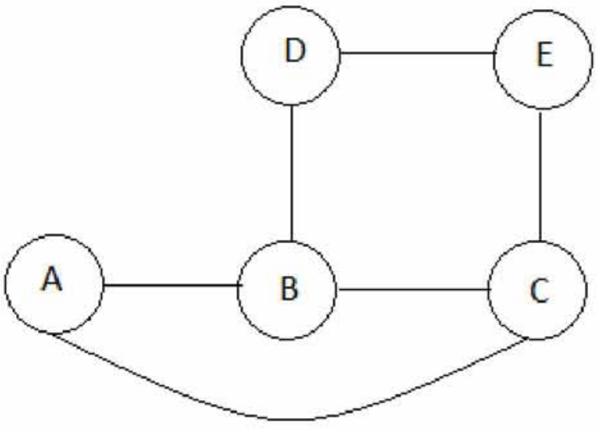


Figure 5.4: Multi-path topology

5.5 Experimental results on single-path topology

To create this single path topology we used /sbin/iptables -A INPUT -m mac --mac-source mac_address -j DROP command to block traffic from specific MAC Addresses. The first experiment was conducted using the same two-hop (A→B→C) and three-hop (A→B→C→D) topology, as shown in figure 5.3 and measuring the throughput while varying the data packet sizes from 256, 512 and 1024 bytes. IPERF was used to generate UDP traffic flows with –l option for various data packets. The main objective behind this was to understand if there was a correlation between packet size and observed throughput. Figure 5.5 and Figure 5.6 show the mean throughput over an experimental duration of 30 seconds. As the size of data packet is increased, mean throughput also increases. It can be seen from the results that throughput scales down as the number of nodes increases. Adding more hops increases protocol's overhead. Note that the throughput for AODV decreases at a faster rate compared to OLSR, as hop count increases. Due to AODV's higher overhead.

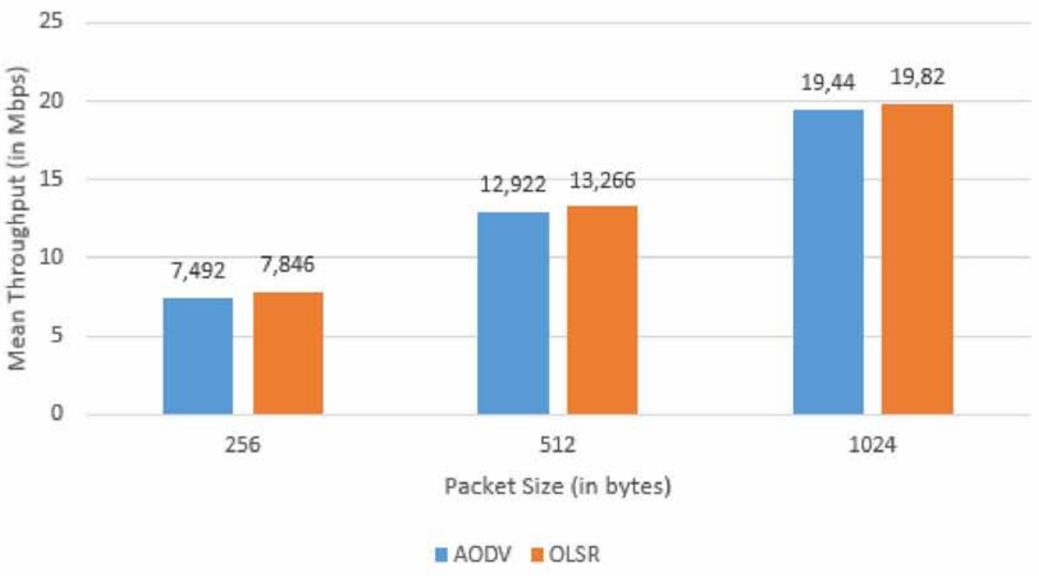


Figure 5.5: Performance-Varying Packet size in two-hop topology

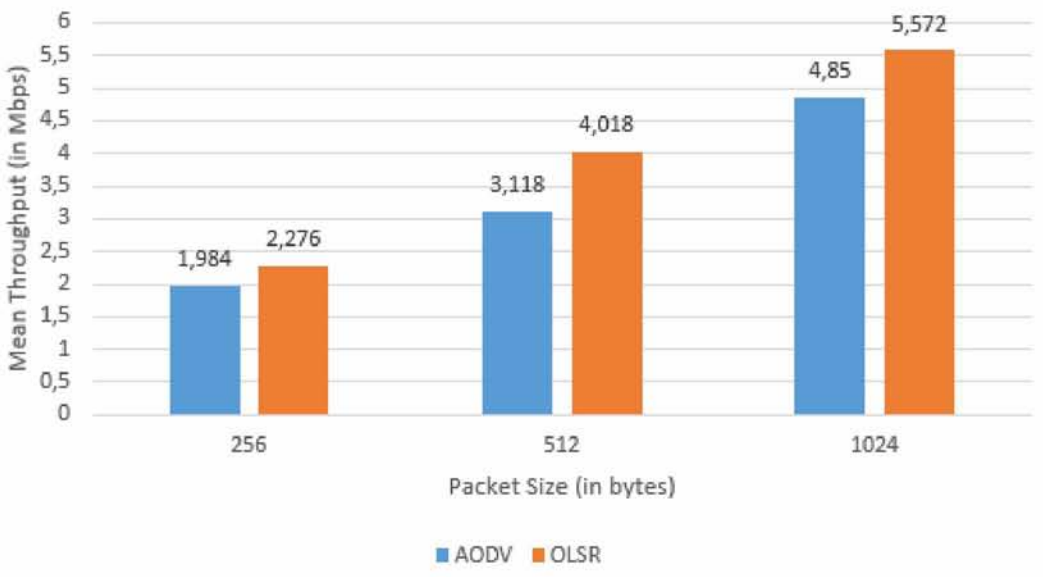


Figure 5.6: Performance-Varying Packet size in three-hop topology

In second experiment we used the single-path topology varying the hop count from one to three, to observe packet delivery ratio for both protocols. For this experiment ping utility was used to create traffic at the application layer between the sender A and the other nodes in the string topology using the sequence B, C, D. Specifically, the duration of each ping operation is variable depending on the distance from destination (1 minute for node B at 1-hop, 2 minutes for C at 2-hops, and 3 minutes for C at 3-hops). Considering the testbed's results, note that OLSR is able to deliver all sent packets for 1-hop destination, while AODV introduces a 16.4% packet loss. For 2-hop destination both protocols introduce a 13.9% packet loss. For 3-hop distance OLSR introduces a 9.9% packet loss, while AODV introduces 38.9%. So, OLSR is more reliable than AODV for static single-path topologies.

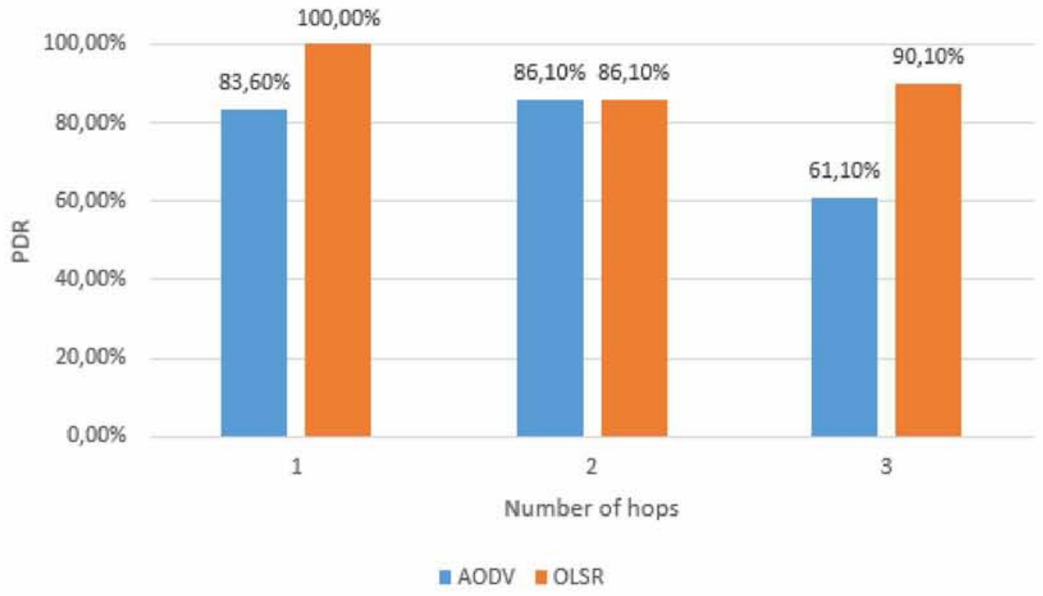


Figure 5.7: Packet Delivery Ratio in single-path topology

Another experiment was conducted to measure the delay each protocol generates in single-path topology varying the hop count from one to three. Ping utility also used for this experiment. For AODV in order to find its delay we subtracted the time of the first RREQ from the time of first ICMP request in the node which creates the UDP traffic. In the case of OLSR, the delay of the protocol is the subtraction of first and last OLSR packets before the first ICMP request was sent. As a proactive protocol, this is the delay to specify the network's topology. As we see from the diagrams, as the number of hops increases, the delay of AODV protocol remains constant around 15 seconds. But in the case of OLSR, protocol's delay goes up as the complexity of network increases.

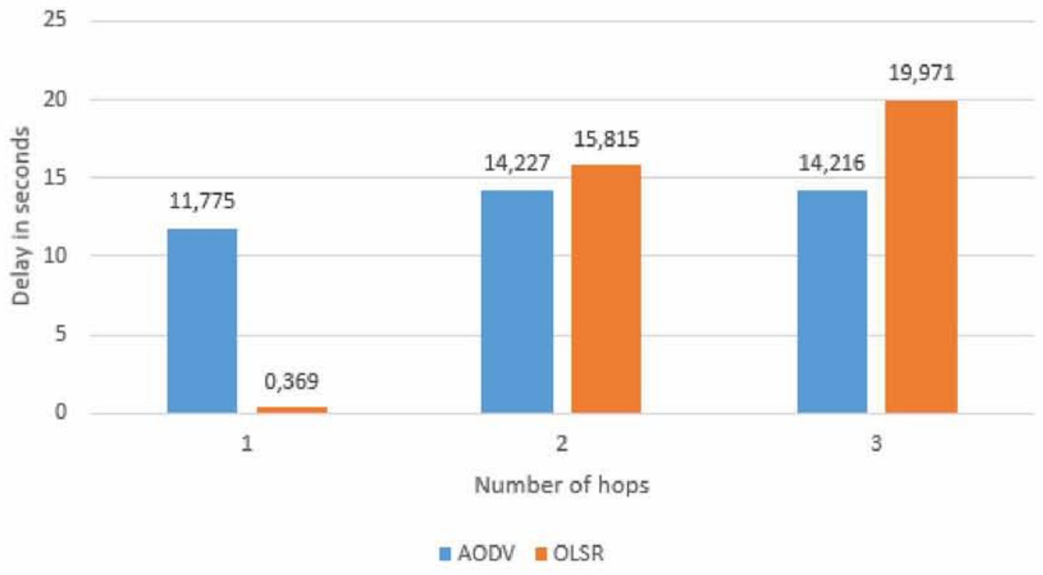


Figure 5.8: Delay in single-path topology

5.6 Experimental results on multi-path topology

We also studied protocols' behavior on multi-path topology as in figure 5.4. To create this multi-path topology we used /sbin/iptables -A INPUT -m mac --mac-source mac_address -j DROP command to block traffic from specific MAC Addresses. For this experiment ping utility was used to create UDP traffic at the application layer between the sender node A and node D. Both protocols followed the shortest path to the destination. Considering the testbed's results, note that OLSR introduces a 16% packet loss, while AODV introduces 19%. So, OLSR is more reliable than AODV for static multi-path topologies. Packet Delivery Ratio for 100 second transmission from node A to D are presented below:

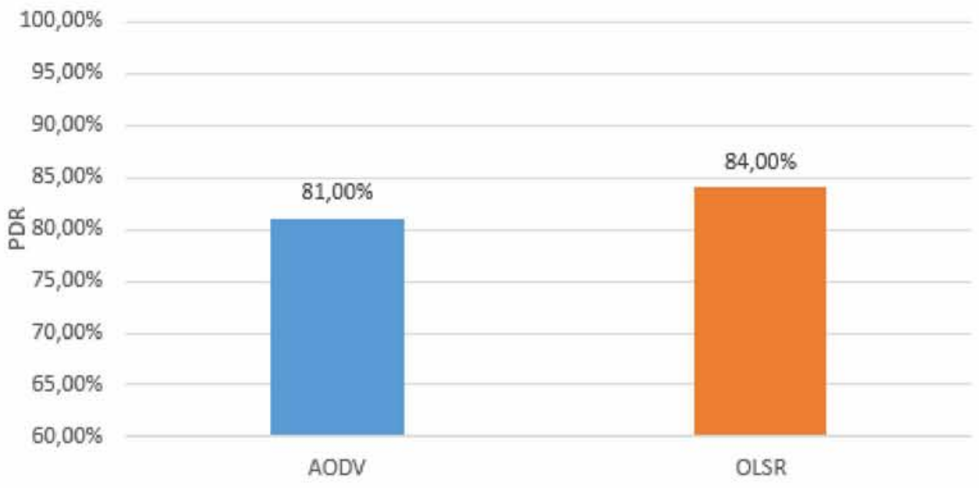


Figure 5.9: Packet Delivery Ratio in multi-path topology

For multi-path topology, we also performed two types of experiments. In the first one, node A starts pinging continuously the same destination D as in figure 5.4, and after a while, node B which is included in the shortest path disconnects itself from the network. OLSR showed a correct behavior. Routing tables were correctly filled in with the shortest path to each destination. In addition the routes were quickly updated with an alternative path upon the node B removal. In case of AODV, each sender was always able to discover the correct path towards each destination, even though the discovery procedure was more time-expensive. In addition, the node was able to discover a new route towards the destination, after the disconnection event, including more delay than OLSR protocol. OLSR is also more reliable, as we see in the chart below.

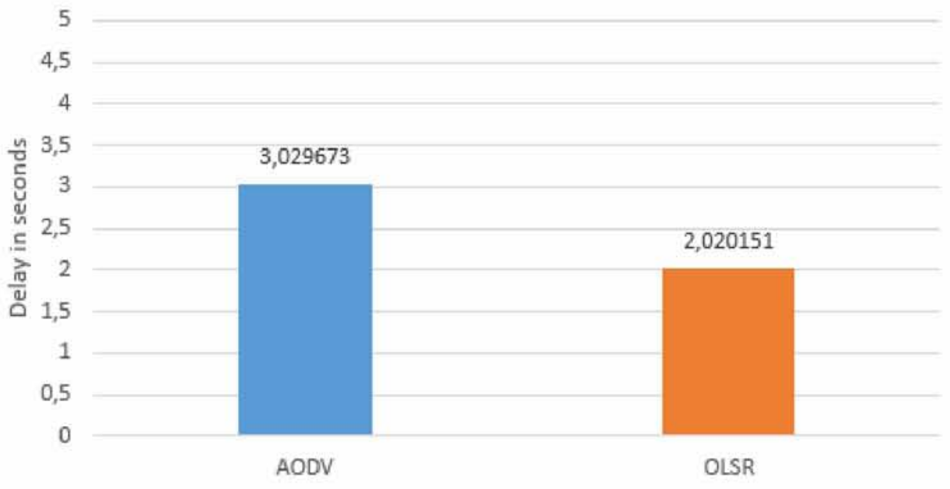


Figure 5.10: Protocol delay in multi-path topology with disconnection event

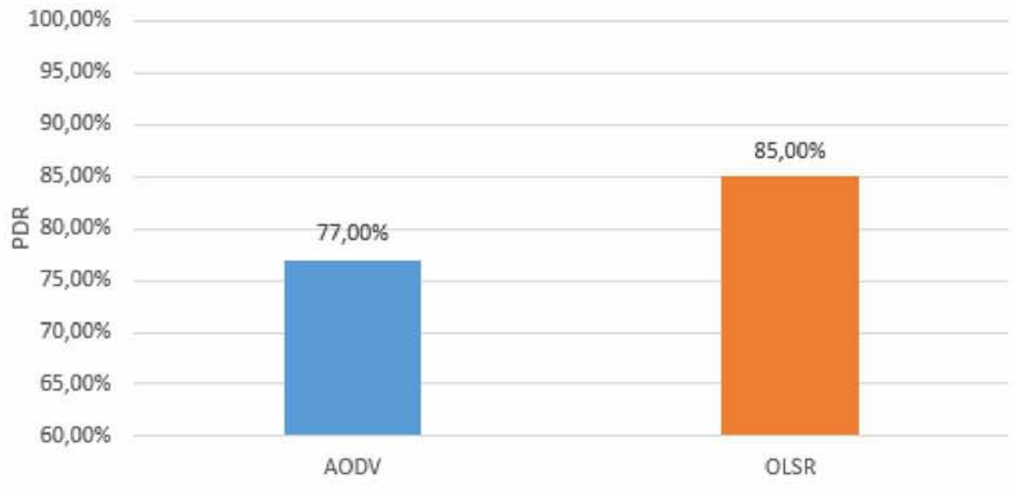


Figure 5.11: Packet Delivery Ratio in multi-path topology with disconnection event

In the second experiment, we studied how fast each protocol switched to an optimal route (shortest path) if a neighboring node in transmission range comes out in the middle of communication. The initial topology used was a three-hop linear topology, as shown in figure 5.12. After a while, node E appears, so that a 2-hop path to the destination can be followed. In case of OLSR, protocol used the 2-hop path after 37 seconds from the time that node E appeared. But AODV finds the shortest path in 21 seconds. So AODV adapts much faster than OLSR when connection events than lead to shortest path encountered. On the other hand, OLSR is again more reliable than AODV protocol, it is able to deliver all sent packets to node D, while AODV introduces 11.7% packet loss.

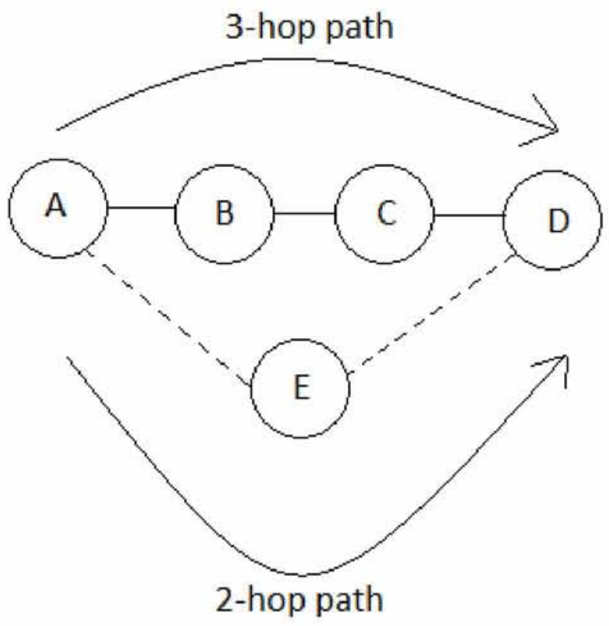


Figure 5.12: Connection event

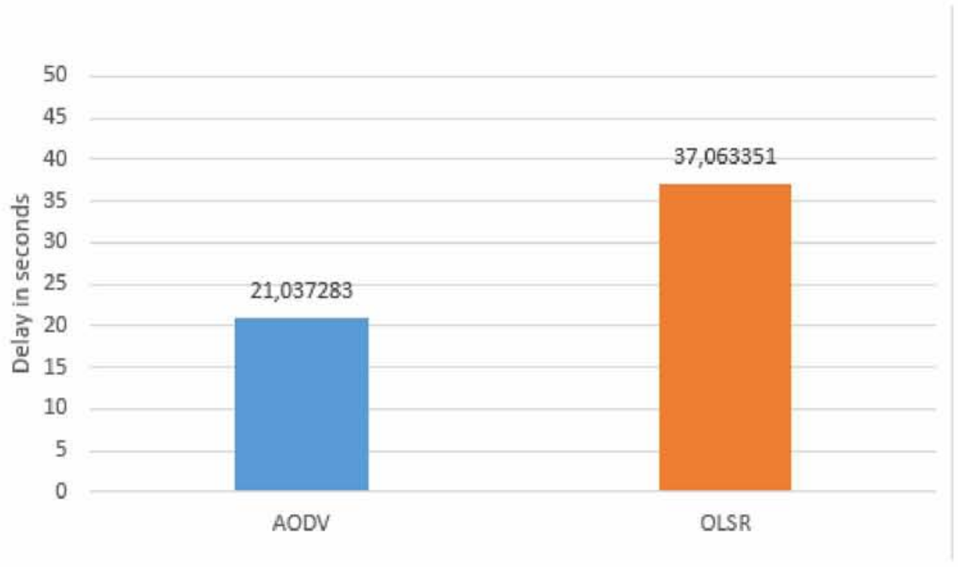


Figure 5.13: Protocol delay in multi-path topology with connection event

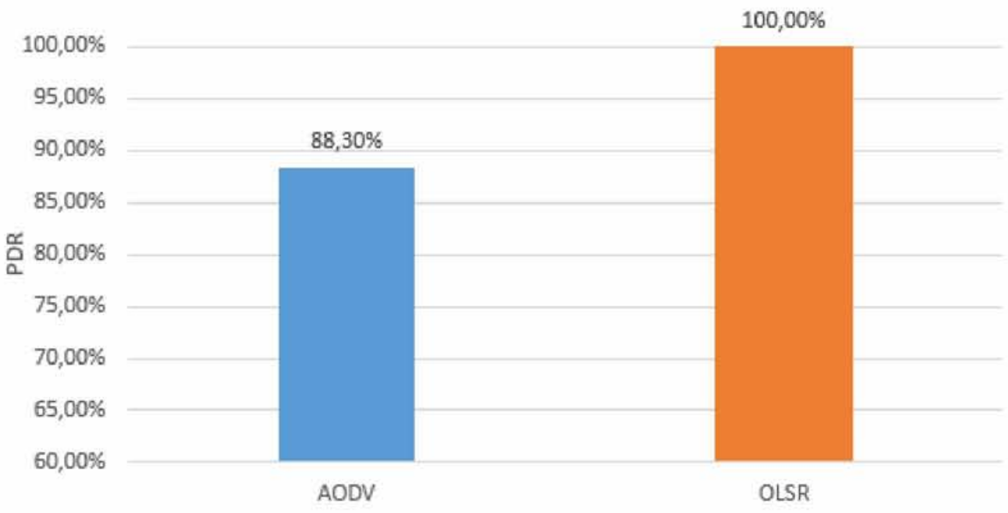


Figure 5.14: Packet Delivery Ratio in multi-path topology with connection event

References

[1] NITOS Wireless Testbed: http://nitlab.inf.uth.gr/NITlab/index.php/testbed

[2] Outdoor Testbed: http://nitlab.inf.uth.gr/NITlab/index.php/testbed/wi-fi-experimentation/outdoor-testbed

[3] Outdoor Nodes' specifications: http://nitlab.inf.uth.gr/NITlab/index.php/hardware/wireless-nodes/outdoor-nodes

[4] Ad-Hoc Networks: http://ciemcal.org/manet-and-routing-techniques/

[5] Routing Basics: http://docwiki.cisco.com/wiki/Routing_Basics

[6] Routing Protocols' Classification: http://www.ciscopress.com/articles/article.asp?p=2180210&seqNum=7

[7] Reactive Protocols: http://www.olsr.org/docs/report_html/node16.html

[8] Proactive Protocols: http://www.olsr.org/docs/report_html/node17.html

[9] Routing Protocols' Classification: http://www.ehow.com/info_8115653_classification-routing-protocols.html

[10] Ad hoc On-Demand Distance Vector Documentation: https://www.ietf.org/rfc/rfc3561.txt

[11] Optimized Link State Routing Protocol Documentation: https://www.ietf.org/rfc/rfc3626.txt

[12] IPERF command: https://iperf.fr/

[13] Wireshark tool: https://www.wireshark.org/

[14] AODV protocol daemon: http://sourceforge.net/projects/aodvuu/files/

[15] OLSR protocol daemon: http://www.olsr.org/?q=download

[16] PING command: http://linux.about.com/od/commands/l/blcmdl8_ping.htm

[17] Iptables MAC Address Filtering: http://www.cyberciti.biz/tips/iptables-mac-address-filtering.html