

Ατομική Διπλωματική Εργασία

**Αξιολόγηση Αλγορίθμων δρομολόγησης κινητών ad-hoc δικτύων
(MANETS) σε συνθήκες τροχαίων ad-hoc δικτύων (VANETS)**

Γιώργος Χατζηζωρζής

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2010

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Αξιολόγηση Αλγορίθμων δρομολόγησης κινητών ad-hoc δικτύων
(MANETS) σε συνθήκες τροχαίων ad-hoc δικτύων (VANETS)**

Γιώργος Χατζηζωρζής

Επιβλέπων Καθηγητής
Δρ. Αντρέας Πιτσιλίδης

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2010

Ευχαριστίες

Πρώτα από όλα θα ήθελα να ευχαριστήσω την οικογένεια μου για την αμέριστη συμπαράσταση που μου παρείχε στην ολοκλήρωση της διπλωματικής αυτής.

Επίσης θα ήθελα να ευχαριστήσω το κ. Γιάννο Μυλωνά για την πολύτιμη βοήθεια την οποία μου παρείχε ως προς το την κατεύθυνση και την ολοκλήρωση της εργασίας αυτής, καθώς και για τις πάμπολλες φορές που χρειάστηκε να τον ενοχλήσω για απορίες και συμβουλές.

Τέλος θα ήθελα να ευχαριστήσω τον κ. Αντρεα Πιτσιλλιδη, τον υπεύθυνο καθηγητή μου, για την πολύτιμη και περιεκτική βοήθεια που μου πρόσφερε κατά τις συναντήσεις τις οποίες είχαμε.

Περίληψη

Η διπλωματική μου εργασία αφορά την μελέτη και αξιολόγηση διαφόρων πρωτοκόλλων που έχουν δημιουργηθεί για λειτουργία σε δίκτυα MANETS όταν εφαρμοστούν σε δίκτυα με συνθήκες VANETS. Η ολοένα αυξανόμενη τάση για ατυχήματα σε μεγάλους αυτοκινητοδρόμους και η ανάγκη για γρήγορη ανταπόκριση από τις αρμόδιες υπηρεσίες ώστε να παρέχεται πιο γρήγορα βοήθεια και αντιμετώπιση του περιστατικού, δημιουργεί την ανάγκη για την δημιουργία ενός αλγορίθμου ο οποίος θα μπορεί να λειτουργήσει αποδοτικά σε ένα αυτοκινητόδρομο λαμβάνοντας υπόψη τις πολύ ιδιόμορφες συνθήκες που επικρατούν σε τέτοιες περιπτώσεις. Πιο συγκεκριμένα για την υλοποίηση του πιο πάνω θα πρέπει κάθε αυτοκίνητο να λειτουργεί και σαν ένας δρομολογητής ώστε σε περίπτωση οποιουδήποτε ατυχήματος να δρομολογούνται κάποια πακέτα που θα μεταφέρουν κριτικές πληροφορίες προς κάποιο σημείο που θα υπάρχουν υπηρεσίες που θα μπορούν να προσφέρουν άμεση βοήθεια.

Εδώ θα αξιολογήσουμε συγκεκριμένους αλγόριθμους όπως έχουμε προαναφέρει πιο πάνω σε σενάρια με πραγματικές συνθήκες αυτοκινητοδρόμου και θα δούμε ποιοι ανταποκρίνονται πιο καλά, ποιοι όχι, που υστερεί και που πλεονεκτεί ο κάθε αλγόριθμος καθώς και ποιος θεωρείται ο πιο αποδοτικός από όλους. Για την αξιολόγηση αυτή θα χρησιμοποιήσουμε το προσομοιωτή δικτύων OPNET και σενάρια από αμερικανικό αυτοκινητόδρομο που έχουν δημιουργηθεί στο προσομοιωτή κίνησης VISSIM. Η διαδικασία αυτή θα μας δώσει την δυνατότητα μέσα από τις μετρικές που μας παρέχει το εργαλείο OPNET να κάνουμε πολλές και ενδιαφέρουσες παρατηρήσεις με όσο πιο σφαιρικό τρόπο γίνεται. Ακολούθως θα γίνει μία συνολική αξιολόγηση η οποία θα μας δώσει την δυνατότητα να προτείνουμε μελλοντικές ενέργειες που θα κινούνται προς την κατεύθυνση ενός αποδοτικού αλγορίθμου ο οποίος θα μπορεί να εφαρμοστεί σε πραγματικές συνθήκες.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
1.1	Στόχος της Διπλωματικής μου Εργασίας.....	1
1.2	Αλγόριθμοι και εργαλεία που θα χρησιμοποιηθούν.....	2
1.3	Οργάνωση Διπλωματικής Εργασίας.....	3
Κεφάλαιο 2	Προηγούμενη Δουλειά.....	4
2.1	Γενικά.....	4
2.2	Ερευνητικά άρθρα.....	4
2.3	Ανοικτά θέματα.....	5
Κεφάλαιο 3	Ανάλυση Αλγορίθμων και μετρικές Αξιολόγησης	7
3.1	Αλγόριθμοι MANETS.....	7
3.1.1	AODV.....	7
3.1.1.1	Εισαγωγή.....	7
3.1.1.2	Ανάλυση αλγορίθμου.....	8
3.1.1.3	Πλεονεκτήματα/Μειονεκτήματα.....	11
3.1.2	DSR.....	12
3.1.2.1	Εισαγωγή.....	12
3.1.2.2	Ανάλυση αλγορίθμου.....	13
3.1.2.3	Πλεονεκτήματα/Μειονεκτήματα.....	14
3.1.3	TORA.....	16
3.1.3.1	Εισαγωγή.....	16
3.1.3.2	Ανάλυση αλγορίθμου.....	16
3.1.3.3	Πλεονεκτήματα/Μειονεκτήματα.....	19
3.1.4	OLSR.....	20
3.1.4.1	Εισαγωγή.....	20
3.1.4.2	Ανάλυση αλγορίθμου.....	20
3.1.4.3	Πλεονεκτήματα/Μειονεκτήματα.....	22
3.1.5	GRP	23
3.1.5.1	Εισαγωγή.....	23

3.1.5.2 Ανάλυση αλγορίθμου.....	23
3.1.5.3 Πλεονεκτήματα/Μειονεκτήματα.....	25
3.2 Διαφορές MANETS-VANETS.....	26
3.3 Συνθήκες που επικρατούν στα δίκτυα VANETS.....	27
3.4 Μετρικές αξιολόγησης.....	28
Κεφάλαιο 4 Υλοποίηση και Ανάλυση Σεναρίων.....	31
4.1 Καθορισμός Προβλήματος.....	31
4.2 Ο προσομοιωτής OPNET.....	32
4.3 Σενάρια αξιολόγησης.....	33
4.3.1 Γενικά για τα σενάρια.....	33
4.3.2 Σενάριο Light.....	36
4.3.3 Σενάριο Medium.....	38
4.3.4 Σενάριο Heavy.....	40
4.4 Προβλήματα που προέκυψαν στις προσομοιώσεις.....	43
Κεφάλαιο 5 Αποτελέσματα.....	44
5.1 Αποτελέσματα σεναρίου Light.....	44
5.1.1 Global Αποτελέσματα.....	45
5.1.2 Μελέτη σύνδεσης κόμβων 82-116.....	50
5.2 Αποτελέσματα σεναρίου Medium.....	54
5.2.1 Global Αποτελέσματα.....	54
5.2.2 Μελέτη σύνδεσης κόμβων 124-259.....	59
5.3 Αποτελέσματα σεναρίου Heavy.....	63
5.3.1 Global Αποτελέσματα.....	63
5.3.2 Μελέτη σύνδεσης κόμβων 260-978.....	68
5.4 Σύγκριση μεταξύ σεναρίων και Scalar Αποτελέσματα.....	71
5.5 Συνολικά Αποτελέσματα.....	78
Κεφάλαιο 6 Συμπεράσματα.....	80
6.1 Τελικά Συμπεράσματα.....	80
6.1.1 Συμπεράσματα με βάση μετρικές.....	80
6.1.2 Συμπεράσματα με βάση περιβάλλον VANETS.....	82
6.1.3 Τελικά Συμπεράσματα.....	83
6.2 Μελλοντική Δουλειά.....	85

Βιβλιογραφία	87
Παράρτημα Α.....	A-1
Παράρτημα Β.....	B-1
Παράρτημα Γ.....	Γ-1

Κεφάλαιο 1

Εισαγωγή

1.1 Στόχος της Διπλωματικής μου Εργασίας	1
1.2 Αλγόριθμοι και εργαλεία που θα χρησιμοποιηθούν	2
1.3 Οργάνωση Διπλωματικής Εργασίας	3

1.1 Στόχος της Διπλωματικής μου Εργασίας

Από τον καιρό της δημιουργίας τους, τα ασύρματα δίκτυα έχουν συγκεντρώσει το ενδιαφέρον και αποτελούν σήμερα μια τεχνολογία ευρέως διαδεδομένη η οποία προσφέρει πολλές και σημαντικές υπηρεσίες στη βιομηχανία, του υπολογιστές, στην έρευνα κλπ. Μέχρι πριν από λίγα χρόνια μιλούσαμε όμως μόνο για infrastructure-based ασύρματα όπου η ασύρματη σύνδεση αφορά μόνο από το τερματικό σύστημα στο access point. Τα τελευταία χρόνια έχει αναπτυχθεί και η άλλη μεγάλη κατηγορία των ασύρματων δικτύων που ονομάζονται infrastructure less-based δίκτυα. Σε αυτά τα δίκτυα όλοι οι κόμβοι συνδέονται ασύρματα μεταξύ τους είτε αφορούν τερματικούς κόμβους είτε αφορούν ενδιάμεσους κόμβους στο δίκτυο.

Η κατηγορία αυτή των ασύρματων δικτύων εισάγει και μία μεγάλη καινοτομία στη δομή τους η οποία αφορά στην δυνατότητα των κόμβων να κινούνται ενώ ανταλλάζουν δεδομένα. Έτσι έχουν δημιουργηθεί τα λεγόμενα wireless ή mobile AD-HOC δίκτυα. Κάποιες μεγάλες κατηγορίες τέτοιων δικτύων είναι τα MANETS, τα VANETS και τα WSN. Τα δίκτυα αυτά δίνουν την δυνατότητα στους κόμβους ενώ κινούνται να ανταλλάζουν παράλληλα και δεδομένα αυξάνοντας έτσι το πεδίο εφαρμογών τους. Μια από αυτές αφορά στα VANETS(Vehicular Ad-Hoc Networks) όπου στην ουσία αναφερόμαστε σε οχήματα τα οποία αποτελούν τους κόμβους του δικτύου μας και ανταλλάζουν πληροφορίες μεταξύ τους ενημερώνοντας το ένα το άλλο για πιθανούς κινδύνους κατά την διαδρομή. Επίσης σε

περίπτωση ατυχήματος σε κάποιο σημείο της διαδρομής μπορούν να δρομολογηθούν πακέτα προς συγκεκριμένους προορισμούς ώστε να ενημερωθούν περιπολικά, ασθενοφόρα κ.λπ. Εκτός τούτου μπορούν να εφαρμοστούν και διάφορες εφαρμογές μεταξύ των κόμβων.

Συνοπτικά βλέπουμε την χρησιμότητα των δικτύων αυτών και πόσο σημαντική θα είναι η συνεισφορά τους στην προσπάθεια ελέγχου της κίνησης ειδικά σε αυτοκινητόδρομους και στην προσπάθεια μείωσης των ατυχημάτων. Παρόλα αυτά θα πρέπει να γίνει ακόμη αρκετή δουλειά μέχρι τη δημιουργία κάποιου πρωτοκόλλου το οποίο θα είναι αρκετά αποδοτικό ώστε να μπορεί να λειτουργήσει σε πραγματικές συνθήκες όπου έχουμε ένα πολύ ιδιόμορφο και πολύ δυναμικό περιβάλλον.

Στόχος της διπλωματικής μου εργασίας είναι να χρησιμοποιήσω διάφορα πρωτόκολλα δρομολόγησης τα οποία είναι σχεδιασμένα για MANETS(Mobile Ad Hoc Networks) και να τα εφαρμόσω πάνω σε δίκτυα τα οποία δουλεύουν κάτω από συνθήκες VANETS κάνοντας διάφορες προσομοιώσεις. Από τις προσομοιώσεις θα παραχθούν κάποια χρήσιμα αποτελέσματα με βάση συγκεκριμένες μετρικές ώστε να μπορέσει να γίνει μια ενδελεχής αξιολόγηση των πρωτοκόλλων αυτών όσο αφορά την εφαρμογή τους σε VANETS και έτσι θα εξαχθούν χρήσιμα συμπεράσματα.

1.2 Αλγόριθμοι και εργαλεία που θα χρησιμοποιηθούν

Αναλυτικότερα, θα χρησιμοποιηθούν οι εξής αλγόριθμοι για MANETS: AODV, DSR, TORA, OLSR και GRP. Υπάρχουν αρκετές μετρικές στο εργαλείο που θα χρησιμοποιήσουμε και μέσω μιας σωστής μελέτης που θα γίνει πρέπει να επιλεγούν αυτές οι οποίες κατά την άποψη μας παίζουν το μεγαλύτερο ρόλο στην αξιολόγηση ενός τέτοιου πρωτοκόλλου λαμβάνοντας φυσικά υπόψη τις ιδιάζουσες συνθήκες που επικρατούν σε ένα VANET. Στη συνέχεια θα γίνουν οι προσομοιώσεις οι οποίες θα αφορούν σενάριο ή σενάρια τα οποία θα εμπεριέχουν και κάποιες ακραίες περιπτώσεις οι οποίες μπορεί να συμβούν σε ένα δίκτυο VANET.

Τα εργαλεία που θα χρησιμοποιηθεί είναι η OPNET που είναι ένας προσομοιωτής δικτύων (network simulator) με γραφικό περιβάλλον ο οποίος μας δίνει πολλές δυνατότητες. Στην OPNET υπάρχουν υλοποιημένες αρκετές μετρικές και έτσι μπορούν να επιλεγούν κάποιες ή αρκετές από αυτές. Τα σενάρια μπορούν να δημιουργηθούν από το VISSIM, που είναι ένας traffic simulator όπου μπορούμε να δημιουργήσουμε αυτοκινητοδρόμους, να εισάγουμε αυτοκίνητα καθώς και να καθορίσουμε την πορεία που θα ακολουθήσουν. Τέλος θα

χρησιμοποιήσουμε και την Microsoft Excel 2007 ώστε να παράξουμε γραφικές από τα scalar αποτελέσματα.

1.3 Οργάνωση Διπλωματικής Εργασίας

Η οργάνωση της Διπλωματικής μου εργασίας από δω και κάτω είναι ο εξής:

Στο επόμενο κεφάλαιο (Κεφάλαιο 2) θα γίνει μία σύντομη αναφορά στην προηγούμενη δουλειά καθώς και στο τι είδους προσομοιώσεις έγιναν και ποια ήταν τα αποτελέσματα. Στο Κεφάλαιο 3 θα γίνει μία ενδελεχής ανάλυση του κάθε ενός από τους 5 αλγόριθμους που θα αξιολογηθούν, θα αναλυθούν οι μετρικές οι οποίες έχουν επιλεγεί για αξιολόγηση των αλγόριθμων ενώ θα γίνει μία σύντομη αναφορά στις διαφορές μεταξύ των δικτύων MANETS και VANETS όπως και στις ιδιάζουσες συνθήκες που επικρατούν στα δίκτυα VANETS. Το Κεφάλαιο 4 θα αφορά την υλοποίηση των προσομοιώσεων και εκεί θα γίνει μία λεπτομερής αναφορά στο εργαλείο OPNET, στα σενάρια και στην τοπολογία που θα χρησιμοποιηθεί και στο τρόπο που έχει επιλεγεί για αξιολόγηση, στο πώς θα δημιουργηθεί δηλαδή το traffic και ποιοι θα είναι οι κόμβοι οι οποίοι θα στέλνουν και ποιοι θα λαμβάνουν πληροφορίες. Το Κεφάλαιο 5 αφορά στα αποτελέσματα των προσομοιώσεων όπου θα παρουσιαστούν οι γραφικές παραστάσεις για κάθε μετρική αξιολόγησης ανά σενάριο και στο τέλος θα παρουσιαστούν οι συνοπτικές γραφικές παραστάσεις οι οποίες θα αφορούν σε σύγκριση μεταξύ των διαφόρων σεναρίων. Τέλος στο Κεφάλαιο 6 θα παρουσιαστούν τα τελικά συμπεράσματα βάση των αποτελεσμάτων των μετρικών που είδαμε και βάση των συνθηκών που επικρατούν σε τέτοιου είδους δίκτυα και θα αναφερθούν μελλοντικές ενέργειες που μπορούν να γίνουν προς την κατεύθυνση επίλυσης του αρχικού προβλήματος.

Κεφάλαιο 2

Προηγούμενη Δουλειά

2.1 Γενικά	4
2.2 Ερευνητικά άρθρα	4
2.3 Ανοικτά θέματα	5

2.1 Γενικά

Στο κεφάλαιο αυτό θα ασχοληθούμε με την δουλειά που έχει ήδη γίνει όσο αφορά την αξιολόγηση των αλγόριθμων αυτών καθώς και ποια ήταν τα αποτελέσματα που προέκυψαν. Πιο συγκεκριμένα θα αναφερθούμε στους τρόπους αξιολόγησης που ακολουθήθηκαν, τα σενάρια καθώς και τα αποτελέσματα που έχουν προκύψει. Επίσης θα αναφερθούμε στο τι δεν έχει γίνει μέχρι τώρα, στο τι θεωρείται ανοικτό ακόμα και που αποσκοπεί η ανάλυση η οποία θα γίνει στην συγκεκριμένη διπλωματική.

2.2 Ερευνητικά άρθρα

Μέχρι στιγμής έχουν γίνει αρκετές μελέτες για τους αλγόριθμους τους οποίους θα χρησιμοποιήσουμε εμείς σε διάφορα εργαλεία (ns2, OPNET, Glomosim) οι οποίες παρουσιάζουν ένα κοινό χαρακτηριστικό: το ότι δεν έχουν αξιολογήσει τους αλγόριθμους αυτούς με σχετικά μεγάλα σενάρια όπως θα κάνουμε εμείς αλλά χρησιμοποιούν μικρό αριθμό κόμβων οι οποίοι πολλές φορές μπορεί να μην κινούνται καθόλου. Στο [4] βλέπουμε να γίνεται μία αξιολόγηση των αλγόριθμων DSR και TORA στον προσομοιωτή OPNET. Τα πειράματα αφορούν σταθερούς και κινητούς κόμβους ο αριθμός των οποίων δεν ξεπερνά τους 60. Πιο συγκεκριμένα γίνονται πειράματα με 10,25,40 και 60 κόμβους και παρουσιάζονται αποτελέσματα χρησιμοποιώντας γραφικές time average. Στο [5] αξιολογούνται οι αλγόριθμοι AODV, DSR και TORA στον προσομοιωτή OPNET με σενάρια όπου έχουμε μικρό αριθμό κόμβων όπως 10,25 και 40. Για την αξιολόγηση

χρησιμοποιήθηκαν οι κόμβοι σαν σταθεροί ή κινητοί ενώ χρησιμοποιήθηκαν συγκεκριμένες ρυθμίσεις. Στη συνέχεια παρουσιάστηκαν τα αποτελέσματα με τη χρήση βασικών μετρικών που υπάρχουν στο συγκεκριμένο προσομοιωτή. Στο [29] γίνεται μία σύγκριση μεταξύ των αλγόριθμων AODV και DSR στο προσομοιωτή Glomosim με συγκεκριμένες ρυθμίσεις που έχουν επιλέξει οι συγγραφείς και σενάρια με 50 κόμβους οι οποίοι αναπτύσσουν ταχύτητες από 0 μέχρι 10Km/h. Στο [19] γίνεται σύγκριση των αλγορίθμων DSDV, AODV και DSR μέσω του προσομοιωτή Glomosim ενώ στο [34] συγκρίνεται ο αλγόριθμος DSR με κάποιους άλλους αλγόριθμους όπως ο DSDV και ο SEAD σε σενάρια με κόμβους οι οποίοι είναι ομαδοποιημένοι σε group κινούνται με ταχύτητες 5m/s(~18Km/h). Για την έρευνα αυτή χρησιμοποιούνται πάλι βασικές μετρικές για να φανεί ποιος αλγόριθμος αποδίδει καλύτερα σε τέτοιου είδους συνθήκες. Στο [21] συναντούμε την αξιολόγηση των αλγορίθμων AODV, DSR, FSR και TORA με πραγματικές συνθήκες αυτοκινητοδρόμου και με διαφορετικές πυκνότητες κόμβων. Οι ταχύτητες των κόμβων κυμαίνονται μεταξύ 80 και 140Km/h. Εδώ μόνο παρουσιάζεται παρόμοια αξιολόγηση όπως αυτές που θα κάνουμε εμείς.

2.3 Ανοικτά θέματα

Αυτό το οποίο διαφαίνεται με την μελέτη των πιο πάνω άρθρων καθώς και άλλων που αφορούν τους αλγόριθμους αυτούς είναι ότι βρισκόμαστε σε ένα αρχικό σημείο στη δημιουργία ενός αποδοτικού αλγόριθμου ο οποίος θα μπορεί να χρησιμοποιηθεί αποτελεσματικά στις ιδιόμορφες συνθήκες που επικρατούν στους αυτοκινητοδρόμους όσο αφορά κυρίως την συνδεσιμότητα μεταξύ των κόμβων. Παρόλα αυτά η ύπαρξη πολλών αλγόριθμων σε άλλες μεγάλες παρόμοιες κατηγορίες όπως τα MANETS μας δίνει την δυνατότητα να τους αξιολογήσουμε στις καινούργιες συνθήκες ώστε να δούμε ποιες αλλαγές χρειάζονται ώστε να ταιριάζουν με τα δεδομένα της καινούργιας μεγάλης κατηγορίας των VANETS. Επιπλέον οι αλγόριθμοι αυτοί μπορούν να αποτελέσουν την βάση για την δημιουργία του αλγόριθμου ο οποίος θα μπορέσει να λειτουργήσει σωστά αφού μπορούμε να τους αξιολογήσουμε, να δούμε τα μειονεκτήματά τους και τα πλεονεκτήματά τους και να δούμε πως μπορούμε να τους αλλάξουμε ώστε να τους βελτιώσουμε. Τέλος μπορούν τα πλεονεκτήματα κάθε αλγόριθμου να αποτελέσουν την απαρχή για την δημιουργία ενός καινούργιου αλγόριθμου ο οποίος θα συνενώνει όλα τα στοιχεία αυτά.

Κεφάλαιο 3

Ανάλυση Αλγορίθμων και Μετρικές Αξιολόγησης

3.1 Αλγόριθμοι MANETS	7
3.1.1 AODV	7
3.1.1.1 Εισαγωγή	7
3.1.1.2 Ανάλυση αλγορίθμου	8
3.1.1.3 Πλεονεκτήματα/Μειονεκτήματα	11
3.1.2 DSR	12
3.1.2.1 Εισαγωγή	12
3.1.2.2 Ανάλυση αλγορίθμου	13
3.1.2.3 Πλεονεκτήματα/Μειονεκτήματα	14
3.1.3 TORA	16
3.1.3.1 Εισαγωγή	16
3.1.3.2 Ανάλυση αλγορίθμου	16
3.1.3.3 Πλεονεκτήματα/Μειονεκτήματα	19
3.1.4 OLSR	20
3.1.4.1 Εισαγωγή	20
3.1.4.2 Ανάλυση αλγορίθμου	20
3.1.4.3 Πλεονεκτήματα/Μειονεκτήματα	22
3.1.5 GRP	23
3.1.5.1 Εισαγωγή	23
3.1.5.2 Ανάλυση αλγορίθμου	23
3.1.5.3 Πλεονεκτήματα/Μειονεκτήματα	25
3.2 Διαφορές MANETS-VANETS	26
3.3 Συνθήκες που επικρατούν στα δίκτυα VANETS	27
3.4 Μετρικές αξιολόγησης	28

3.1 Αλγόριθμοι MANETS

Εδώ θα γίνει η ανάλυση των 5 αλγορίθμων τους οποίους θα συγκρίνουμε και θα αξιολογήσουμε ώστε να δούμε τα τεχνικά τους χαρακτηριστικά, τις μεθόδους που χρησιμοποιούν και πως προσπαθούν να αντεπεξέλθουν σε κάποια ακραία σενάρια. Επίσης στη συνέχεια θα αναφερθούμε στις κατάλληλες μετρικές με τις οποίες θα πρέπει να γίνει η αξιολόγηση των αλγορίθμων αυτών ώστε να αποκτήσουμε σωστή εικόνα για το πώς δουλεύει ο κάθε ένας σε ένα περιβάλλον VANETS. Πριν επιλεγούν οι μετρικές όμως θα γίνει μία ανάλυση των ιδιάζουσων συνθηκών που επικρατούν στα VANETS καθώς και μια σύγκριση μεταξύ VANETS-MANETS. Αυτό θα μας βοηθήσει να επιλέξουμε τις κατάλληλες μετρικές που θα μας δώσουν τις πληροφορίες και τα στοιχεία που θέλουμε ώστε να δούμε που υπερτερεί το κάθε πρωτόκολλο.

3.1.1 Αλγόριθμος AODV [15] [17] [29] [32] [33]

3.1.1.1 Εισαγωγή

Ο αλγόριθμος δρομολόγησης AODV(Ad-hoc On-demand Distance Vector) ανήκει στην κατηγορία των reactive unicast αλγορίθμων δρομολόγησης. Οι reactive αλγόριθμοι δρομολόγησης είναι αυτοί που δεν διατηρούν συνέχεια κάποιο μονοπάτι από μία πηγή προς ένα προορισμό αλλά δημιουργούν ένα εάν γίνει κάποιο request από κάποιο κόμβο. Έτσι η δημιουργία του μονοπατιού γίνεται δυναμικά. Χρησιμοποιεί την λειτουργία on-demand η οποία αφορά στην δημιουργία ενός μονοπατιού εάν και εφόσον ζητηθεί από κάποιο κόμβο κάτι που έχουμε αναφέρει και πιο πάνω.

Βασίζεται κυρίως στο πρωτόκολλο DSDV και προσπαθεί να βελτιώσει τα ελαττώματα του προκατόχου του. Ο αλγόριθμος αυτός επίσης θεωρείται πρόδρομος του και είναι κατασκευασμένος για δίκτυα με μικρό πληθυσμό. Σαν κύριο σκοπό ο αλγόριθμος έχει την μετάδοση πληροφοριών μόνο εάν είναι αναγκαίο, την διαφοροποίηση των πληροφοριών τοπικού επιπέδου και global επιπέδου και την μετάδοση πληροφοριών σε τοπικό επίπεδο ώστε να γίνεται ενημέρωση των γειτόνων ενός κόμβου που έχει υποστεί αλλαγές. Έχει σχεδιαστεί για χρήση σε δίκτυα τα οποία αποτελούνται από μερικές δεκάδες μέχρι μερικές χιλιάδες κόμβους. Στο δίκτυο αυτό μπορεί να χρησιμοποιηθούν κόμβοι με χαμηλή, μέτρια

και σχετικά ψηλή κινητικότητα. Τέλος αξίζει να αναφέρουμε ότι στον AODV όλα τα μηνύματα αποστέλλονται στο port 654 μέσω του πρωτοκόλλου UDP.

3.1.1.2 Ανάλυση αλγορίθμου

Για να μπορεί να υπάρχει σωστή τοπική συνδεσιμότητα και επικοινωνία, ο αλγόριθμος χρησιμοποιεί μικρά hello messages ώστε να δίνει την δυνατότητα σε κάθε κόμβο να γνωρίζει ποιοι είναι οι γείτονες του. Τα hello messages είναι μικρά πακέτα τα οποία εκπέμπονται από τους διάφορους κόμβους ενός δικτύου για να δηλώσουν την παρουσία τους στο χώρο ώστε να τους αντιληφθούν οι υπόλοιποι κόμβοι που είναι στην εμβέλεια τους. Η διαδικασία αυτή μπορεί να γίνει με 2 τρόπους: είτε στέλνοντας τα μικρά hello messages κατά την διάρκεια ενός hello interval, είτε αν δεν σταλεί κάποιο πακέτο από κάποιο κόμβο προς τους γείτονες του κατά την διάρκεια ενός hello interval, ο κόμβος αυτός κάνει broadcast ένα hello message το οποίο περιέχει την ταυτότητα του και ένα sequence number ώστε να το λάβουν όλοι οι γείτονες του και να ανανεώσουν τις πληροφορίες τους. Το hello interval είναι ο χρόνος τον οποίο περιμένει κάποιος κόμβος πριν στείλει το επόμενο hello message στους γείτονες του. Το πακέτο αυτό έχει TTL (ο χρόνος τον οποίο είναι "ζωντανό" ένα πακέτο-συνήθως μετριέται με hop counts) ίσο με 1 για να μην μεταδοθεί πιο πέρα από τους τοπικούς γείτονες του. Έτσι με τους δύο τρόπους αυτούς διασφαλίζεται ότι ο κάθε κόμβος θα μπορεί ανά πάσα στιγμή να γνωρίζει ποιοι είναι οι γείτονες του.

Κάθε κόμβος έχει ένα πίνακα δρομολόγησης. Είναι στην ουσία ένας ηλεκτρονικός πίνακας ή βάση δεδομένων η οποία υπάρχει σε κάθε κόμβο στην οποία αποθηκεύονται πληροφορίες διαδρομών η και κάποιες άλλες πληροφορίες με σκοπό να βοηθούν τους κόμβους να μεταδίδουν σωστά τα πακέτα πληροφορίας, να φθάνουν δηλαδή στο σωστό προορισμό.

Όσο αφορά την διαχείριση του πίνακα δρομολόγησης κάθε κόμβος διατηρεί κάποια καταχώρηση στο πίνακα αυτό για κάθε προορισμό για τον οποίο έχει επιδειχθεί ενδιαφέρον, για αποστολή πληροφοριών δηλαδή. Σε κάθε καταχώρηση υπάρχουν οι εξής πληροφορίες: ο προορισμός, ο επόμενος κόμβος, ο αριθμός των hops, το sequence number για τον προορισμό, οι ενεργοί κόμβοι για τη διαδρομή και ο χρόνος λήξης για τη καταχώρηση αυτή στο πίνακα. Σε περίπτωση που υπάρχει προσφορά καινούργιας διαδρομής μεταξύ δύο κόμβων τότε γίνεται σύγκριση με την υφιστάμενη διαδρομή και επιλέγεται η διαδρομή με το

μεγαλύτερο sequence number. Εάν τα sequence number είναι τα ίδια τότε επιλέγεται η διαδρομή με τα λιγότερα hops. Επίσης πέρα από αυτές τις καταχωρήσεις υπάρχουν και άλλες διαφορετικού είδους που περιέχουν χρήσιμες πληροφορίες όπως ο route request expiration timer ο οποίος σχετίζεται με τις καταχωρήσεις για reverse path και έχει σκοπό να σβήνει τις καταχωρήσεις αυτές στους κόμβους που δεν συμμετέχουν στη διαδρομή από την πηγή στο προορισμό. Ακόμα υπάρχει και το route caching timeout το οποίο δείχνει μετά από ποιο χρονικό σημείο μια διαδρομή θεωρείται άκυρη. Μέσω αυτών των καταχωρήσεων είτε των διαδρομών είτε των ειδικών αυτών τιμών ο κάθε κόμβος προσπαθεί να διαχειρίζεται σωστά τις διαθέσιμες διαδρομές ώστε να μπορεί να επικοινωνεί αποτελεσματικά με τους διάφορους κόμβους του δικτύου.

Ένας γείτονας θεωρείται ενεργός προς ένα προορισμό εάν στείλει κάποιο πακέτο προς αυτό το προορισμό κατά την διάρκεια του πιο πρόσφατου active timeout period. Οι ενεργοί κόμβοι περιέχονται σε κάθε καταχώρηση του πίνακα δρομολόγησης. Έτσι είναι εύκολο να ειδοποιηθούν όλοι οι ενεργοί κόμβοι εάν σπάσει κάποιο link. Μια καταχώρηση διαδρομής θεωρείται ενεργή εάν χρησιμοποιείται από οποιοδήποτε ενεργό γείτονα. Μια διαδρομή θεωρείται ενεργή εάν μεταφέρονται πακέτα από την πηγή προς το προορισμό με τη χρήση των διαφόρων καταχωρήσεων στους ενδιάμεσους κόμβους.

Ο αλγόριθμος αυτός έχει δύο μεγάλες διαδικασίες: τη δημιουργία καινούργιου μονοπατιού και την διαχείριση υφιστάμενου μονοπατιού. Η δημιουργία καινούργιου μονοπατιού ενεργοποιείται όταν κάποιος κόμβος θέλει να μεταδώσει κάποιες πληροφορίες σε κάποιο άλλο αλλά δεν έχει πληροφορίες για αυτόν. Έτσι ο κόμβος αυτός κάνει broadcast ένα πακέτο RREQ(Route Request) το οποίο περιέχει τα εξής: διεύθυνση πηγής, sequence number πηγής, ταυτότητα broadcast, διεύθυνση προορισμού, sequence number προορισμού και hop count. Οι μοναδικές τιμές είναι το ζεύγος διεύθυνση πηγής, ταυτότητα broadcast. Τα sequence numbers κάθε κόμβου καθώς και η ταυτότητα του broadcast είναι δύο ανεξάρτητοι μετρητές οι οποίοι διαχειρίζονται από κάθε κόμβο ξεχωριστά.

Έτσι οι κόμβοι οι οποίοι λαμβάνουν το RREQ είτε αποστέλλουν ένα RREP (Route Reply) πίσω στο κόμβο που τους έστειλε το request είτε αναμεταδίδουν το RREQ. Για να δημιουργήσει κάποιος κόμβος ένα RREP και να το στείλει πίσω στο προορισμό θα πρέπει είτε να είναι ο ίδιος ο προορισμός είτε να έχει μία έγκυρη διαδρομή προς αυτόν και το sequence number της διαδρομής στο route table να είναι έγκυρο και μεγαλύτερο ή ίσο από το destination sequence number που αναγράφεται στο RREQ. Εάν κάποιος κόμβος λάβει

περισσότερες από μια φορές το ίδιο RREQ τότε διαγράφει το επιπλέον πακέτο και δεν το αναμεταδίδει. Εάν ένας κόμβος δεν στείλει πίσω RREP αλλά μεταδώσει το RREQ διατηρεί κάποια στοιχεία ώστε να μπορέσει να υλοποιήσει το reverse path setup όπως και το forward path setup τα οποία είναι: i.p. address του προορισμού, i.p. διεύθυνση της πηγής, ταυτότητα broadcast, χρόνος λήξης για τη καταχώρηση του reverse path και το sequence number της πηγής.

Το reverse path setup αφορά στην δημιουργία του αντίστροφου μονοπατιού, από τον προορισμό στην πηγή δηλαδή, καθώς ταξιδεύει το RREQ από την πηγή προς τον προορισμό. Αυτό επιτυγχάνεται με το να φυλάει κάθε κόμβος που λαμβάνει το RREQ την διεύθυνση του κόμβου που του έχει στείλει το RREQ δημιουργώντας στην ουσία ένα pointer προς τον κόμβο που του έχει στείλει το RREQ.

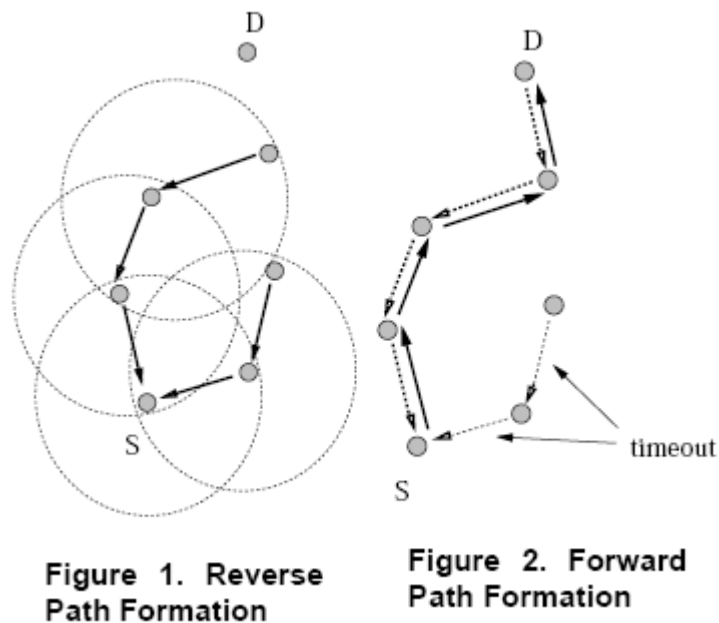
Το forward path setup αφορά στην αντίθετη διαδικασία από το reverse path setup όπου ξεκινά ένα RREP από τον προορισμό προς την πηγή για να απαντήσει στο RREQ και κάθε κόμβος που το μεταδίδει φυλάει τον κόμβο που του έχει στείλει το RREP. Οι κόμβοι που δεν είναι μέσα στο μονοπάτι που έχει καθορίσει το RREP θα έχουν time out και έτσι θα αναγκαστούν να σβήνουν το δείκτη προς τον κόμβο από τον οποίο έλαβαν το RREP. Οι δύο διαδικασίες αυτές φαίνονται στην εικόνα πιο κάτω (Σχήμα 3.1).

Όσο αφορά την διαχείριση των μονοπατιών έχουμε τα εξής: Εάν μετακινηθεί η πηγή και σπάσει κάποιο link τότε μπορεί να ξεκινήσει από την αρχή την διαδικασία ανακάλυψης καινούργιας διαδρομής. Εάν μετακινηθεί κάποιος ενδιάμεσος κόμβος ή ο προορισμός και σπάσει πάλι κάποιο link τότε στέλνονται RRER ώστε να ενημερωθούν όλοι οι κόμβοι μέχρι την πηγή. Έτσι η πηγή μπορεί να ξεκινήσει καινούργια διαδικασία εξεύρεσης μονοπατιού με RREQ που έχουν μεγαλύτερο sequence number από τα προηγούμενα ώστε να αποφύγει να λάβει RREP από κόμβους που θεωρούν την προηγούμενη διαδρομή έγκυρη. Τα RREP αυτά έχουν hop count ίσο με άπειρο αλλά αυτό δεν δημιουργεί ανησυχία καθώς η μετάδοση των RREP τερματίζει λόγω της ιδιότητας του αλγορίθμου να είναι loop free. Είναι η δυνατότητα του αλγόριθμου δηλαδή να αποφεύγει τους κύκλους καθώς μεταδίδονται πακέτα μεταξύ των διαφόρων κόμβων. Θεωρείται ένα από τα πλεονεκτήματα του αλγόριθμου και βοηθά κυρίως στη διαχείριση των μονοπατιών.

3.1.1.3 Πλεονεκτήματα/Μειονεκτήματα

Στα πλεονεκτήματα του αλγορίθμου κατατάσσεται η ιδιότητα που έχει ο αλγόριθμος να αποφεύγει τους κύκλους (loop-free) με την χρήση ενός sequence number για κάθε κόμβο, ο οποίος χρησιμοποιείται μαζί με τα πακέτα που αποστέλλει κάθε κόμβος. Αυτό συμβαίνει και στους κόμβους οι οποίοι αποτελούν είτε την πηγή είτε τον προορισμό μίας σύνδεσης ενώ όταν χρειάζεται να γίνει μία καινούργια σύνδεση πρέπει οι δύο αυτοί τερματικοί κόμβοι να αυξήσουν τα sequence number τους δείχνοντας έτσι ότι η καινούργια σύνδεση είναι πιο φρέσκια από τις προηγούμενες. Η χρήση του sequence number βοηθά και στο να μπορεί να επιλεγεί μεταξύ δύο ή περισσότερων διαδρομών προς τον ίδιο προορισμό η διαδρομή που είναι πιο "φρέσκια", πιο καινούργια. Επιπλέον το πρωτόκολλο δεν απαιτεί περιοδικές global διαφημίσεις αλλά μόνο τοπικές. Επίσης το πρωτόκολλο αυτό έχει αρκετό πλεονέκτημα όσο αφορά το overhead απέναντι σε άλλα πιο απλά πρωτόκολλα που μεταφέρουν όλη τη διαδρομή από το προορισμό στη πηγή στα μηνύματα τους. Σημαντικό μπορεί επίσης να θεωρηθεί το ότι ο αλγόριθμος αυτός αντιδρά γρήγορα σε οποιεσδήποτε τοπολογικές αλλαγές και ενημερώνει μόνο τους κόμβους οι οποίοι πιθανόν να επηρεάστηκαν από την αλλαγή αυτή.

Σαν μειονεκτήματα στο πρωτόκολλο αυτό σημειώνουμε το overhead που μπορεί να υπάρχει στη χρήση των καναλιών κατά την διάρκεια εξεύρεσης ενός μονοπατιού λόγω της διαδικασίας reverse path setup και forward path setup. Επίσης μπορεί να παρατηρηθεί υψηλό latency/delay κατά την διάρκεια εξεύρεσης κάποιου καινούργιου μονοπατιού ειδικά σε ad-hoc networks. Τέλος στη διαχείριση των διαδρομών δεν επαναχρησιμοποιούνται οι πληροφορίες δρομολόγησης αλλά ανακαλύπτονται δυναμικά κάτι που συμβαίνει σε άλλα πρωτόκολλα όπως ο DSR.



Σχήμα 3.1 : Reverse και Forward Path setup στον αλγόριθμο AODV [15]

3.1.2. Αλγόριθμος DSR [1] [8] [15] [22] [29] [33]

3.1.2.1. Εισαγωγή

Ο αλγόριθμος DSR(Dynamic Source Routing) ανήκει και αυτός στους reactive unicast αλγόριθμους δρομολόγησης. Είναι ένα απλό και αποδοτικό πρωτόκολλο και έχει σχεδιαστεί ειδικά για χρήση σε multi-hop wireless δίκτυα για κινητούς κόμβους. Τα δίκτυα αυτά είναι ασύρματα δίκτυα στα οποία η επικοινωνία μπορεί να γίνει μέσω ενός αριθμού από ενδιάμεσους κόμβους δημιουργώντας έτσι μια διαδρομή με πολλά hops.

Η χρήση οποιουδήποτε infrastructure δεν είναι απαραίτητη αφού το πρωτόκολλο ρυθμίζεται από μόνο του.

Ο αλγόριθμος αυτός χρησιμοποιεί source routing. Αυτό σημαίνει ότι το κάθε πακέτο το οποίο διακινείται στο δίκτυο περιέχει όλη τη διαδρομή από την πηγή μέχρι το προορισμό. Η διαδικασία μπορεί να θεωρηθεί κάποιες φορές ως πλεονέκτημα και κάποιες άλλες ως μειονέκτημα. Ως reactive πρωτόκολλο χρησιμοποιεί την on-demand λειτουργία για την

δημιουργία μιας διαδρομής. Ο αλγόριθμος αυτός έχει δύο κύριες λειτουργίες: το route discovery και το route maintenance. Μια άλλη σημαντική λειτουργία την οποία χρησιμοποιεί το πρωτόκολλο είναι το route caching. Τέλος το πρωτόκολλο αυτό δεν χρησιμοποιεί περιοδικά μικρά hello messages (beacons).

3.1.2.2 Ανάλυση αλγορίθμου

Η διαδικασία route discovery (εξεύρεση διαδρομής) ενεργοποιείται ως εξής: Η διαδικασία ξεκινά όταν κάποιος κόμβος χρειάζεται να μεταδώσει κάτι σε κάποιο κόμβο για τον οποίο δεν έχει κάποια διαδρομή στο cache του. Έτσι ξεκινά την διαδικασία αυτή κάνοντας broadcast ένα route request - χρησιμοποιεί flooding δηλαδή. Στο πακέτο αυτό δίνεται ένα sequence number ώστε να μπορούν οι ενδιαμέσοι κόμβοι να το χρησιμοποιήσουν για να αποφύγουν τους διάφορους κύκλους. Έτσι όποιος κόμβος λάβει το πακέτο αυτό για πρώτη φορά(από την δεύτερη φορά και έπειτα δεν το λαμβάνει υπόψη) προσθέτει στην ουσία την διεύθυνση του στη λίστα της διαδρομής, στην λίστα δηλαδή της διαδρομής από την πηγή και αναμεταδίδει το πακέτο. Έτσι σε κάποια στιγμή το πακέτο αυτό θα φτάσει στο προορισμό. Αυτόματα λόγω αυτού που αναφέραμε πιο πάνω υπάρχει και η αντίστροφη διαδρομή από τον προορισμό προς την πηγή μέσα στο πακέτο αυτό. Έτσι με το που λαμβάνει το πακέτο αυτό ο προορισμός στέλνει πίσω ένα reply πακέτο προς την πηγή ώστε να θεωρηθεί ότι έχει επιλεγεί αυτή η διαδρομή. Όλη η διαδικασία αυτή φαίνεται στο σχήμα 3.2.

Έτσι η πηγή φυλάει την διαδρομή στο cache της. Άρα μπορούν να υπάρχουν αρκετές διαδρομές προς διάφορους προορισμούς στο cache ενός κόμβου.

Παράλληλα με την διαδικασία αυτή, οι ενδιαμέσοι κόμβοι έχουν την δυνατότητα να φυλάγουν στο cache τους πληροφορίες δρομολόγησης που λαμβάνονται από το request πακέτο καθώς ακολουθεί την πορεία του προς τον προορισμό καθώς και από τα πακέτα πληροφορίας. Επίσης άλλοι κόμβοι μπορούν να "ακούσουν" το traffic που δεν διακινείται προς αυτούς χρησιμοποιώντας το promiscuous mode. Το promiscuous mode είναι η λειτουργία κατά την οποία κάποιος κόμβος έχει την δυνατότητα να διαβάζει κάθε πακέτο το οποίο βρίσκεται στην εμβέλεια του ώστε να λαμβάνει επιπλέον πληροφορίες για το δίκτυο.

Επιπλέον εάν κάποιος ενδιαμέσος κόμβος έχει την διαδρομή προς τον προορισμό τότε μπορεί να στείλει ένα reply package στη πηγή αντί να μεταδώσει το request πακέτο. Αυτό

δίνει την δυνατότητα στην πηγή να δημιουργεί πολλές διαδρομές προς ένα συγκεκριμένο προορισμό. Επίσης βοηθά στην μείωση του αχρείαστου flooding στο δίκτυο αφού καθώς και στην πιο γρήγορη εξεύρεση διαδρομής.

Η διαδικασία route maintenance είναι μια διαδικασία η οποία παρέχει την δυνατότητα να επιβεβαιωθεί ότι κάποιο link δεν έχει σπάσει. Είναι πολύ σημαντική διαδικασία λόγω της ιδιότητας των κόμβων να είναι κινητοί και αυτό δημιουργεί την ανάγκη να υπάρχει η σιγουριά ανά πάσα στιγμή ότι κάποιο link μπορεί να χρησιμοποιηθεί για την διακίνηση δεδομένων από μια πηγή σε ένα προορισμό.

Το route maintenance στην ουσία είναι μια σειρά από τρεις διαδικασίες που εκτελούνται η μια μετά την άλλη με σκοπό να διαφανεί κατά πόσο κάποια διαδρομή είναι έγκυρη. Πρώτα το DSR ζητά από το link layer(από το MAC πρωτόκολλο και πιο συγκεκριμένα το IEEE 802.11) να εγγυηθεί ότι είναι έγκυρη η διαδρομή. Αν δεν μπορεί να συμβεί αυτό τότε ενεργοποιείται η άλλη διαδικασία η οποία έχει σκοπό να ακούσει κάθε πακέτο στην εμβέλεια του κόμβου ώστε να διαφανεί εάν τα links είναι ακόμα διαθέσιμα. Αυτό ονομάζεται passive acknowledgment. Εάν αποτύχουν οι δυο αυτές διαδικασίες τότε η επιβεβαίωση μπορεί να γίνει με acknowledgments από το network layer. Έτσι στέλνεται ένα πακέτο μέσω της διαδρομής αυτής και εάν σε κάποια στιγμή ένας ενδιάμεσος κόμβος ανακαλύψει ότι το link είναι "σπασμένο " τότε στέλνει ένα σφάλμα διαδρομής(RRER) πίσω στη πηγή. Έτσι η πηγή μπορεί να ψάξει μέσα στο cache της να βρει ένα υπάρχει άλλη διαδρομή. Εάν δεν υπάρχει τότε μπορεί να ξεκινήσει την διαδικασία εξεύρεσης διαδρομής (Route Discovery).

3.1.2.3 Πλεονεκτήματα/Μειονεκτήματα

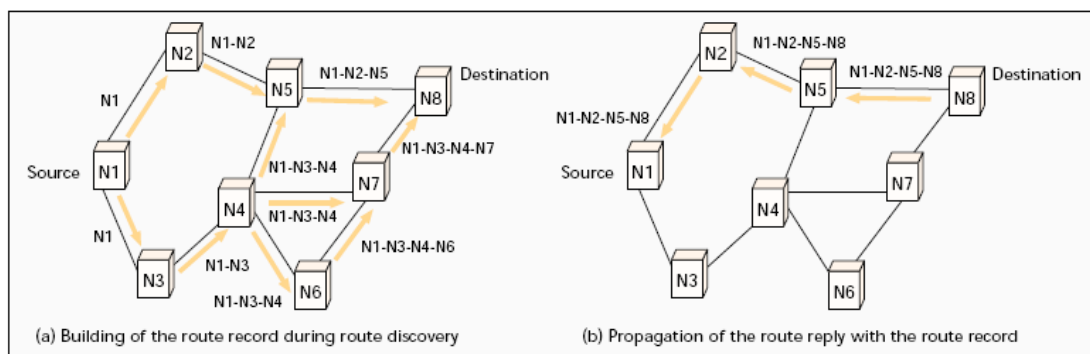
Το πρωτόκολλο DSR έχει αρκετά πλεονεκτήματα σε σύγκριση με τα άλλα πρωτόκολλα και θεωρείται ότι μπορεί να λειτουργήσει καλύτερα σε δίκτυα μικρού και μεσαίου μεγέθους. Σημαντικό πλεονέκτημα θεωρείται το ότι ο αλγόριθμος δεν χρησιμοποιεί περιοδικά hello messages και οι κόμβοι μπορούν να κάνουν οικονομία ενέργειας. Αυτό ισχύει κυρίως στα MANETS όπου οι κόμβοι έχουν περιορισμένα αποθέματα ενέργειας και όχι τόσο στα VANETS όπου τα οχήματα έχουν σαφώς περισσότερα αποθέματα ενέργειας.

Επίσης αυτό ελευθερώνει κάποιο bandwidth από το δίκτυο συνολικά.

Επιπλέον πλεονέκτημα θεωρείται για το DSR το ότι οι κόμβοι μπορούν να φυλάξουν πολλαπλές διαδρομές αφού ο κάθε κόμβος διατηρεί το δικό του cache και έτσι κάποιος

κόμβος πριν ξεκινήσει την διαδικασία για εξεύρεση διαδρομής μπορεί να κοιτάξει στο δικό του cache. Αυτή η χρήση του routing cache βοηθά στο να μειωθεί το routing overhead με τις εξής τρεις λειτουργίες που έχουμε αναφέρει και πιο πάνω: εάν κάποιος ενδιαμέσος κόμβος έχει την διαδρομή προς τον προορισμό τότε μπορεί να στείλει ένα reply πακέτο στη πηγή, με το να μπορεί η πηγή να χρησιμοποιήσει κάποιο άλλο μονοπάτι προς τον ίδιο προορισμό που έχει φυλαγμένο στο cache της εάν το υφιστάμενο που προσπαθεί να χρησιμοποιήσει είναι "σπασμένο" και με το να μπορεί ένας ενδιαμέσος κόμβος ο οποίος έχει ανακαλύψει ότι ένα link είναι broken να χρησιμοποιήσει κάποια άλλη διαδρομή εάν υπάρχει στο cache του. Εννοιολογικά το routing overhead είναι το overhead που αφορά στα πακέτα τα οποία στέλνονται μεταξύ των κόμβων και έχουν ως σκοπό την δημιουργία κάποιας διαδρομής.

Στα μειονεκτήματα του πρωτοκόλλου σημειώνεται η λειτουργία του κατά την οποία όλη η διεύθυνση της διαδρομής μεταφέρεται στο πακέτο γεγονός που αυξάνει το protocol overhead ειδικά σε μεγάλα δίκτυα με πάρα πολλούς κόμβους. Επίσης μεγάλο ζήτημα στο πρωτόκολλο αυτό είναι κατά πόσο οι διαδρομές που βρίσκονται στο cache κάθε κόμβου γίνονται μη έγκυρες στο σωστό χρόνο ή παραμένουν στο cache ακόμα και μετά που δεν ισχύουν με αποτέλεσμα να προκαλούν λάθος δρομολογήσεις.



Σχήμα 3.2: Η διαδικασία εξεύρεσης διαδρομής στο DSR [15]

3.1.3 Αλγόριθμος TORA [16] [36] [37]

3.1.3.1 Εισαγωγή

Ο αλγόριθμος TORA (Temporally Ordered Routing Protocol) είναι ένα reactive πρωτόκολλο το οποίο χρησιμοποιεί την λειτουργία on-demand. Είναι ένας κατανεμημένος αλγόριθμος δρομολόγησης ο οποίος χρησιμοποιεί ένα αλγόριθμο "link reversal" που δίνει την δυνατότητα δημιουργίας κατεύθυνσης στα διάφορα links που υπάρχουν στο δίκτυο και τροποποίησης της ανάλογα με τις αλλαγές στην τοπολογία . Είναι loop-free και παρέχει την δυνατότητα των πολλαπλών διαδρομών μεταξύ δύο συγκεκριμένων κόμβων. Βασίζεται στο πρωτόκολλο LMR (Light-weight Mobile Routing) αφού χρησιμοποιεί παρόμοιο αλγόριθμο "link reversal" και route repair. Ο "link reversal" είναι ο αλγόριθμος ο οποίος προσπαθεί να δημιουργεί ένα γράφημα με τάση να κινείται προς το προορισμό. Όσο αφορά τον αλγόριθμο LMR (Light-Weight Mobile Routing) είναι ένα on-demand πρωτόκολλο το οποίο παρέχει πολλαπλά μονοπάτια προς τον προορισμό με την χρήση του αλγορίθμου link-reversal ώστε να δημιουργήσει ένα DAG. Οι κόμβοι διατηρούν διαδρομές μόνο προς τους γείτονες τους ενώ επίσης link failure θεωρείται ότι συμβαίνει και ανακοινώνεται όταν κάποιος κόμβος χάσει όλα τα downstream links του, όχι μόνο κάποια.

Πολλές φορές αυτό μπορεί να γίνει αλλάζοντας την κατεύθυνση κάποιων links σε περίπτωση αλλαγών στο δίκτυο Υποστηρίζει multicasting αλλά αυτό δεν περιλαμβάνεται στις βασικές του λειτουργίες. Η λειτουργία-κλειδί για το TORA θεωρείται η τοπικοποίηση των control messages στο σημείο που παρατηρείται η αλλαγή στην τοπολογία. Τα control messages είναι μηνύματα ή πακέτα τα οποία αφορούν στην επικοινωνία των κόμβων μεταξύ τους ώστε να δείξουν για παράδειγμα την αποτυχία σύνδεσης με κάποιο κόμβο ή την προσπάθεια δημιουργίας κάποιου μονοπατιού προς κάποιο κόμβο.

Για να μπορέσει να γίνει αυτό πρέπει οι κόμβοι να διατηρούν routing information για τους γειτονικούς τους κόμβους δηλαδή μέχρι 1 hop.

3.1.3.2 Ανάλυση αλγορίθμου

Ο αλγόριθμος χρησιμοποιεί τρεις βασικές λειτουργίες: το route creation, το route maintenance και το route erasure.

Το route creation προκύπτει δυναμικά όταν ένας κόμβος θέλει να επικοινωνήσει με κάποιο άλλο κόμβο και χρειάζεται να δημιουργηθεί μια διαδρομή προς αυτόν. Είναι δηλαδή μια διαδικασία source-initiating. Έτσι η πηγή κάνει broadcast ένα request packet που περιέχει την διεύθυνση του προορισμού. Το πακέτο αυτό μετακινείται μέσα στο δίκτυο μέχρι να φτάσει στο προορισμό ή σε κάποιο κόμβο ο οποίος περιέχει κάποια διαδρομή προς αυτόν. Έτσι ο παραλήπτης του πακέτου κάνει broadcast ένα update packet τοποθετώντας μέσα το "ύψος" του με βάση το προορισμό(το "ύψος" του προορισμού είναι 0). Το "ύψος"(height) είναι μια μετρική η οποία καθορίζει πόσο πιο ψηλά βρίσκεται κάποιο κόμβος από τον προορισμό. Είναι μία πλειάδα η οποία αποτελείται από 5 μεταβλητές και πιο συγκεκριμένα: $H=(\tau,oid,r,\delta,I)$. Το τ είναι το logical time που συμβαίνει ένα link failure, το oid είναι η μοναδική ταυτότητα του κόμβου που καθορίζει ένα καινούργιο reference level(ένα καινούργιο reference level δημιουργείται όταν υπάρχει link failure και ο κόμβος που το ανακαλύπτει αυξάνει το ύψος του), το r είναι το reflection indicator το οποίο δείχνει δύο μοναδικά sub levels, το δ το οποίο δείχνει το επίπεδο στο κοινό reference level(στο οποίο το reference level του προορισμού είναι 0) και το I που είναι η μοναδική ταυτότητα του κόμβου. Ο προορισμός έχει πάντοτε $H_i=(0,0,0, i)$ ενώ όταν κάποιος κόμβος δεν συμμετέχει στο DAG και έχει σαν NULL την τιμή του ύψους του τότε το ύψος του αναπαριστάται με $H_i=(-,-,-,i)$. Από όσο έχει φανεί από τα παραδείγματα οι κόμβοι, καθώς το πακέτο UPD απομακρύνεται από το προορισμό και πάει προς την πηγή, αυξάνουν το ύψος τους σε σχέση με το κόμβο από τον οποίο έλαβαν το πακέτο. Επίσης εάν έχω δύο κόμβους i και j οι οποίοι συνδέονται μεταξύ τους ο τρόπος για να βρω προς τα πού δείχνει το link(αφού είναι κατευθυνόμενα όπως έχουμε προαναφέρει) είναι ο εξής ψευδοκώδικας : Εάν το $H_j==NULL$ τότε είναι unassigned το link, εάν το $H_i==NULL$ τότε είναι downstream, εάν το $H_i>H_j$ τότε είναι downstream(προς τον j) και εάν το $H_j < H_i$ τότε είναι upstream το link.

Έτσι καθώς ταξιδεύει το update πακέτο πίσω προς την πηγή κάθε κόμβος που το λαμβάνει βάζει τιμή για το ύψος του μεγαλύτερη από αυτή που έχει ο κόμβος που έστειλε το update πακέτο προς αυτόν. Παράλληλα δημιουργούνται και κατευθυνόμενα links από τους κόμβους με μεγαλύτερο ύψος προς τους κόμβους με μικρότερο ύψος. Στην ουσία δημιουργείται ένα γράφημα το οποίο ονομάζεται DAG(Directed Acyclic Graph). Αυτό είναι ένα γράφημα στο οποίο δεν υπάρχουν κύκλοι, δεν μπορώ να καταλήξω δηλαδή κάπου χρησιμοποιώντας δύο φορές την ίδια ακμή. Επίσης υπάρχει κάποιος προορισμός στο γράφημα και είναι αυτός ο οποίος δεν έχει κάποιο εξερχόμενο link σε αντίθεση με όλους τους άλλους κόμβους οι οποίοι

έχουν κάποιο link εξερχόμενο από αυτούς. Κάθε ακμή του γραφήματος αυτού έχει κατεύθυνση.

Αυτό φαίνεται στο σχήμα 3.3α.

Η διαδικασία route maintenance προκύπτει όταν κάποιο link σπάσει. Αυτό είναι πιθανό να συμβεί λόγω της υψηλής κινητικότητας των κόμβων στο δίκτυο ειδικά εάν πρόκειται για αυτοκίνητα τα οποία κινούνται με πολύ ψηλές ταχύτητες. Έτσι το DAG σπάζει και ξεκινά η λειτουργία route maintenance ώστε να αποκατασταθεί το πρόβλημα που έχει προκύψει. Ο κόμβος ο οποίος εντοπίζει το σπασμένο link που είναι και ο κοντύτερος στο link αυτό τότε αυξάνει το "ύψος" του ώστε να έχει το πιο ψηλό ύψος τοπικά. Στη συνέχεια οι κόμβοι που δεν έχουν downstream link(συνήθως ο κόμβος που έχει ανακαλύψει το σπασμένο link), δηλαδή link με κατεύθυνση προς το προορισμό(links που να φεύγουν από αυτούς, outgoing links δηλαδή) αλλάζουν την κατεύθυνση των downstream links (στην περίπτωση του αλγορίθμου μας είναι ένα link, μια ακμή η οποία εξέρχεται από κάποιο κόμβο) που έρχονται με κατεύθυνση προς αυτούς και την κάνουν αντίθετη (Σχήμα 3.3β). Ο αλγόριθμος αυτός χρησιμοποιεί μία παραλλαγή του link reversal αλγορίθμου όπου τα links τα οποία αντιστρέφονται είναι αυτά τα οποία δεν έχουν αντιστραφεί ξανά πριν.

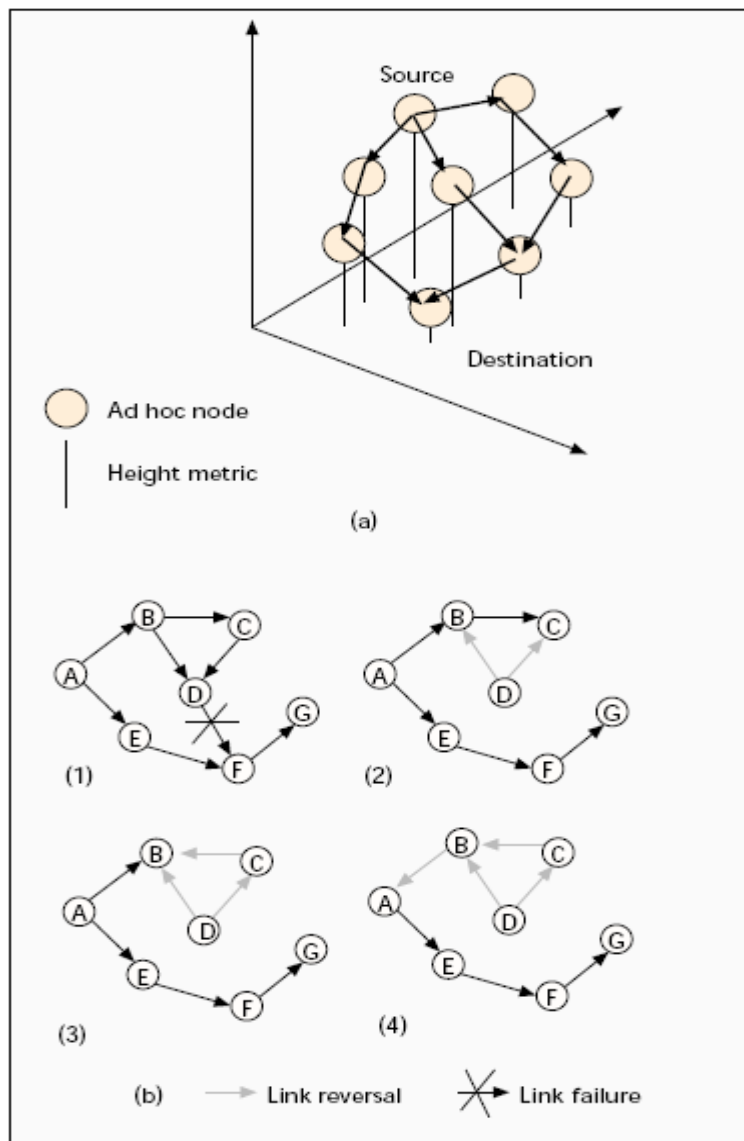
Πολύ σημαντικό στο αλγόριθμο αυτό θεωρείται το timing αφού η μετρική "ύψος" βασίζεται πολύ στη ώρα που συμβαίνει ένα link failure. Το link failure αφορά στην αποτυχία σύνδεσης μεταξύ δύο κόμβων κυρίως λόγω του παύει ο ένας κόμβος να είναι στην εμβέλεια του άλλου. Ο αλγόριθμος θεωρεί ότι όλοι οι κόμβοι έχουν συγχρονισμένα ρολόγια όπως σε ένα GPS.

Η λειτουργία route erasure αφορά στη μετάδοση μέσω flooding ενός clear packet (CLR) μέσα στο δίκτυο για να σβηστούν όλες οι άκυρες διαδρομές. Στη διαδικασία αυτή όλοι οι κόμβοι θέτουν το ύψος τους στο NULL και παύουν τα διάφορα links να έχουν κατεύθυνση. Μπορεί για παράδειγμα να ανακαλυφθεί ένα network partition, δηλαδή ένα κομμάτι του δικτύου να διαχωριστεί από το προορισμό. Τότε θα πρέπει να σταλεί το clear packet αυτό ώστε να αφαιρεθούν όλες οι άκυρες διαδρομές.

Επίσης υπάρχει ένας δευτερεύων μηχανισμός ο οποίος ονομάζεται route optimization κατά τον οποίο οι διάφοροι κόμβοι του δικτύου επαναυπολογίζουν το "ύψος" τους ώστε να βελτιώσουν την δρομολόγηση στο δίκτυο.

3.1.3.3 Πλεονεκτήματα/Μειονεκτήματα

Ο αλγόριθμος αυτός έχει ως πλεονέκτημα του ότι μειώνει το overhead με την δυνατότητα που έχει να τοπικοποιεί τις αντιδράσεις του δικτύου εάν υπάρξει οποιαδήποτε αλλαγή τοπολογίας. Επίσης δημιουργεί περισσότερες από μια διαδρομές προς τους διάφορους προορισμούς του δικτύου. Στα μειονεκτήματα του σημειώνουμε το ότι θεωρείται ένας από τους πιο πολύπλοκους αλγόριθμους για προβλήματα routing ενώ όταν πρόκειται για μεγάλα δίκτυα τα οποία αλλάζουν γρήγορα όπως δεν αποδίδει δεν αποδίδει τόσο καλά όσο άλλα πρωτόκολλα. Επίσης παρατηρείται overhead σε περίπτωση link failure και χρειάζεται να γίνει αναδιάταξη μέσα στο δίκτυο.



Σχήμα 3.3: Η μετρική height στον αλγόριθμο TORA [33]

3.1.4 Αλγόριθμος OLSR [14] [18] [40]

3.1.4.1 Εισαγωγή

Ο αλγόριθμος αυτός (Optimized Link State Routing) ανήκει στην κατηγορία των proactive πρωτόκολλων δρομολόγησης, δηλαδή ανά πάσα στιγμή διατηρεί διαδρομές προς όλους τους γνωστούς προορισμούς. Βασίζεται στο παραδοσιακό link-state αλγόριθμο. Ο link state αλγόριθμος αποτελεί μια από τις δύο μεγάλες κατηγορίες αλγορίθμων με κύρια χαρακτηριστικά την διαφήμιση των link information μεταξύ των κόμβων ώστε να κτίσει ο κάθε ένας τη δική του τοπολογία δικτύου καθώς και την δημιουργία update όταν αλλάζουν οι πληροφορίες των link.

Βασικό χαρακτηριστικό του αλγόριθμου αυτού είναι ότι προσπαθεί να περιορίσει το control message overhead κάθε κόμβου με το να επιλέγει ένα set γειτονικών κόμβων οι οποίοι θα αναμεταδίδουν τα μηνύματα αυτά. Οι κόμβοι αυτοί ονομάζονται multipoint relays(MPRs). Αυτοί οι κόμβοι είναι υπεύθυνοι για να κάνουν προωθούν το control traffic του συγκεκριμένου κόμβου του οποίου αποτελούν τους MPRs χρησιμοποιώντας flooding. Ο αλγόριθμος είναι εντελώς καταναεμημένος και δεν υπάρχει κάποια κεντρική οντότητα στο δίκτυο. Έτσι ο κάθε κόμβος χρησιμοποιεί τις δικές του τοπικές πληροφορίες για να δρομολογήσει τα διάφορα πακέτα.

3.1.4.2.Ανάλυση αλγορίθμου

Ο αλγόριθμος αυτός, όπως έχουμε αναφέρει και πιο πάνω, είναι proactive. Αυτό σημαίνει ότι ο κάθε κόμβος έχει ανά πάσα στιγμή το βέλτιστο δρομολόγιο για κάθε γνωστό προορισμό με το να χρησιμοποιεί τις πληροφορίες τις οποίες έχει φυλαγμένες στο δικό του πίνακα δρομολόγησης. Βασίζεται στο παραδοσιακό link state αλγόριθμο. Για να καταφέρει να έχει το σωστό πίνακα δρομολόγησης κάθε κόμβος χρησιμοποιεί τα μικρά hello messages ώστε να γνωρίζει ποιοι είναι οι γείτονες του ανά πάσα χρονική στιγμή. Επίσης χρησιμοποιεί τα Topology Control(TC) messages τα οποία μεταδίδονται μόνο μέσω των MPRs κάποιου κόμβου και αφορούν σε updates στην τοπολογία ώστε να ενημερωθούν και οι υπόλοιποι κόμβοι του δικτύου και να υπολογίσουν σωστά τους πίνακες δρομολόγησης τους. Τα TC μεταδίδονται ανά τακτά χρονικά διαστήματα που ονομάζονται TC_intervals. Παρόλα αυτά κάποιες φορές μπορεί να σταλούν επιπλέον TC μηνύματα όταν υπάρχει πιθανότητα να έχει

σπάσει κάποιο link ή έχει αλλάξει το set των MPRs χωρίς απαραίτητα να περάσει το TC_interval. Αυτό δίνει την δυνατότητα πιο γρήγορης αντίδρασης του δικτύου σε οποιαδήποτε τοπολογική αλλαγή. Έτσι με τη βοήθεια των μηνυμάτων αυτών κάθε κόμβος μπορεί να διατηρεί οποιαδήποτε στιγμή δρομολόγια για όλους του γνωστούς προορισμούς αφού πρώτα υπολογίσει το δικό του πίνακα δρομολόγησης.

Ειδική αναφορά πρέπει να γίνει στα MultiPoint Relays (MPRs) αφού αυτά αποτελούν μία από τις σημαντικότερες λειτουργίες του συγκεκριμένου αλγόριθμου. Τα MPRs είναι υπεύθυνα για την μετάδοση όλων των control messages (TC) του κόμβου του οποίου αποτελούν τα MPRs ώστε να ενημερωθεί όλο το δίκτυο. Όσοι κόμβοι δεν ανήκουν στο σύνολο των κόμβων αυτών δεν μπορούν να αναμεταδώσουν τα μηνύματα αυτά, μπορούν απλώς μόνο να τα διαβάσουν και να τα επεξεργαστούν. Οι MPRs επιλέγονται με τον εξής τρόπο: Επιλέγονται κάποιοι κόμβοι που είναι ένα hop μόνο μακριά από το βασικό κόμβο ώστε να μπορούν να καλύπτουν όλους τους κόμβους που είναι δύο hops μακριά από το βασικό κόμβο. Αυτό φαίνεται στο σχήμα 3.4. Οι MPRs είναι οι κόμβοι με το γαλάζιο χρώμα. Όσο πιο μικρό είναι το σύνολο αυτό τόσο πιο μικρό θα είναι το control traffic overhead. Το traffic overhead είναι το overhead το οποίο δημιουργείται από το traffic που διακινείται από τους κόμβους του δικτύου.

Έτσι κάθε κόμβος διαφημίζει τα links του μεταξύ αυτού και των MPRs του ώστε να υπάρχουν αρκετές πληροφορίες για να γίνουν δρομολογήσεις και για να αποθηκεύονται οι σωστές πληροφορίες στους πίνακες δρομολόγησης που διατηρεί ο κάθε κόμβος. Έτσι οποιεσδήποτε αλλαγές στα links δημιουργούν την ανάγκη update στους πίνακες δρομολόγησης των κόμβων και γι' αυτό στέλνονται τα control messages(TC) μέσω των MPRs.

Κάθε control message (TC) το οποίο μεταδίδεται έχει το δικό του sequence number και τα πιο καινούργια μηνύματα έχουν μεγαλύτερο sequence number. Έτσι οι κόμβοι μπορούν εύκολα να αναγνωρίσουν πιο μήνυμα είναι πιο "φρέσκο".

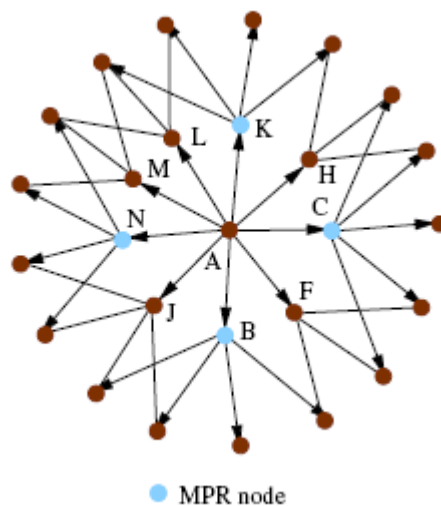
Ο αλγόριθμος αυτός δεν είναι σχεδιασμένος για να λαμβάνει πληροφορίες από το link layer (το layer το οποίο βρίσκεται κάτω από το network layer). Παρόλα αυτά εάν υπάρχουν διαθέσιμες πληροφορίες ειδικά για κάποιο link failure τότε κάποιος κόμβος μπορεί να τις χρησιμοποιήσει. Τα μικρά hello messages τα οποία χρησιμοποιεί ο αλγόριθμος τον βοηθούν ώστε να βρίσκει ποιοι κόμβοι είναι δύο hops μακριά του και να επιλέγει κατάλληλα τους

MPRs. Επιπλέον σε όλες τις διαδρομές που δημιουργούνται από μία πηγή προς ένα προορισμό οι ενδιάμεσοι κόμβοι είναι MPRs.

3.1.4.3 Πλεονεκτήματα/Μειονεκτήματα

Ο αλγόριθμος αυτός μπορεί να δουλέψει καλύτερα σε πυκνά και μεγάλα δίκτυα. Αυτό μπορεί να έχει τα δικά του θετικά και αρνητικά. Θετικό στο πρωτόκολλο αυτό μπορεί να θεωρηθεί η χρήση των MPRs αφού μόνο αυτοί μεταδίδουν τα TC μηνύματα υπό μορφή flooding και όχι όλοι μειώνοντας έτσι το routing overhead καθώς και το traffic που μπορεί να παρατηρηθεί μέσα στο δίκτυο.

Από την άλλη ο αλγόριθμος αυτός μπορεί να παρουσιάσει υψηλό overhead λόγω της ιδιότητας του να είναι proactive και κάθε κόμβος να διατηρεί διαδρομές προς όλους τους κόμβους χωρίς να είναι πάντοτε απαραίτητο. Επίσης στον αλγόριθμο δεν υπάρχει καθόλου υποστήριξη για multicast όπως υπάρχει για τους άλλους αλγόριθμους.



Σχήμα 3.4 : Κόμβοι MPR στον αλγόριθμο OLSR [1]

3.1.5 Αλγόριθμος GRP

3.1.5.1. Εισαγωγή

Το πρωτόκολλο αυτό (Geographic Routing Protocol) είναι το πρωτόκολλο της OPNET για τα MANETS. Θεωρείται σαν ένα proactive πρωτόκολλο όπου η δρομολόγηση γίνεται με βάση την πιο μικρή γεωγραφική απόσταση μεταξύ του κόμβου και της πηγής. Για να επιτευχθεί αυτό κάθε κόμβος έχει δικό του GPS το οποίο το βοηθά να γνωρίζει ανά πάσα στιγμή την τοποθεσία του. Κάποιες φορές μπορεί να χρησιμοποιηθεί flooding για να βοηθηθούν οι διάφορες λειτουργίες του δικτύου ενώ και εδώ γίνεται χρήση των μικρών hello messages για να επικοινωνούν οι γείτονες μεταξύ τους. Βασικό concept του αλγόριθμου αυτού είναι ο διαχωρισμός του δικτύου σε γειτονιές (quadrants) το οποίο μπορεί να ευκολύνει την διαχείριση του δικτύου αφού το χωρίζει σε μικρότερα κομμάτια. Η γειτονιά είναι βασική έννοια του αλγόριθμου αυτού όπου στην ουσία ο χώρος του δικτύου χωρίζεται σε ίσα τετράγωνα με την δημιουργία πολλαπλών επιπέδων όπως έχουμε αναφέρει πιο πάνω. Ετυμολογικά η λέξη quadrant σημαίνει το ένα τέταρτο του κύκλου ή κάτι που έχει σαν σχήμα το ένα τέταρτο του κύκλου.

3.1.5.2 Ανάλυση αλγορίθμου

Βασικό ρόλο στο πρωτόκολλο αυτό παίζουν οι γειτονιές-quadrants όπως έχουμε αναφέρει πιο πάνω. Με το τρόπο αυτό κάποιος κόμβος μπορεί εύκολα να καθορίσει σε ποια γειτονιά βρίσκεται αφού όπως έχουμε ήδη αναφέρει διατηρεί το δικό του GPS το οποίο του επιτρέπει να βρίσκει την θέση του. Ο διαχωρισμός γίνεται σε διάφορα επίπεδα. Αρχικά το δίκτυο χωρίζεται σε δύο μεγάλα ίσα τετράγωνα. Αυτό αποτελεί το πρώτο επίπεδο. Ακολουθώντας το πιο κάτω επίπεδο αποτελεί το διαχωρισμό του πιο πάνω επιπέδου σε άλλα πιο μικρά τετράγωνα. Συγκεκριμένα το κάθε τετράγωνο του πιο πάνω επιπέδου χωρίζεται σε 4 ίσα μικρότερα τετράγωνα. Αυτό αποτελεί το αμέσως πιο κάτω επίπεδο. Αυτό συνεχίζεται μέχρι το επίπεδο όπου τα τετράγωνα θα έχουν το μέγεθος το οποίο έχει επιλέξει ο χρήστης. Έτσι γίνεται ο διαχωρισμός του χώρου του δικτύου και αυτό δίνει την ευκολία στους κόμβους να καθορίσουν σε ποια γειτονιά βρίσκονται ανά πάσα στιγμή.

Το πρωτόκολλο αυτό χρησιμοποιεί επίσης μικρά περιοδικά hello messages. Αυτό γίνεται ώστε ο κάθε κόμβος να γνωρίζει ποιοι είναι οι γείτονες του καθώς και ποια είναι η τοποθεσία του κάθε ενός. Το κάθε hello message περιέχει την τοποθεσία του κόμβου ο οποίος το στέλλει. Έτσι δίνεται η δυνατότητα στους κόμβους αυτούς να γνωρίζουν του γείτονες τους.

Ο κάθε κόμβος διατηρεί στην βάση δεδομένων του δύο πίνακες. Ο πρώτος αφορά σε όλους τους δυνατούς προορισμούς στο δίκτυο και ο δεύτερος αφορά σε όλους τους έγκυρους γείτονες κάθε κόμβου. Όσοι κόμβοι είναι στην ίδια γειτονιά-quadrant με κάποιο κόμβο, τότε αυτός διατηρεί στο πίνακα του την ακριβή τοποθεσία τους. Εάν είναι όμως σε διαφορετική γειτονιά-quadrant τότε διατηρεί τις πληροφορίες τις οποίες έχει από τα ψηλότερα επίπεδα που υπάρχουν όσο αφορά τις γειτονίες.

Στην αρχικοποίηση του δικτύου γίνεται ένα flooding. Αυτό έχει ως σκοπό το να μάθει κάθε κόμβος την τοποθεσία όλων των άλλων κόμβων μέσα στο δίκτυο. Επίσης flooding παρατηρείται όταν κάποιος κόμβος περάσει από μια γειτονία σε άλλη.

Η διαδικασία δρομολόγησης γίνεται ως εξής: Καταρχήν υπάρχουν δύο σενάρια. Αυτά αφορούν στο να είναι η πηγή όπως και ο προορισμός στην ίδια γειτονιά/quadrant και στο να είναι η πηγή και ο προορισμός σε διαφορετικές γειτονίες.

Όταν είναι στην ίδια γειτονία και οι δύο τότε η πηγή στέλνει το πακέτο στο γείτονα της ο οποίος είναι πιο κοντά στο προορισμό. Έτσι ακολουθείται η ίδια διαδικασία έως ότου φτάσει στον προορισμό.

Όταν τώρα οι κόμβοι είναι σε διαφορετικές γειτονίες, ο κόμβος ο οποίος θέλει να στείλει κάποιες πληροφορίες σε κάποιο άλλο κόμβο γνωρίζει από τα ψηλότερα επίπεδα των γειτονιών/quadrants, στα οποία έχουμε αναφερθεί πιο πάνω, πως γειτονεύει με τον προορισμό. Έτσι στέλνει το πακέτο στο γείτονα του ο οποίος είναι πιο κοντά στην γειτονία του προορισμού. Αυτό γίνεται συνεχώς μέχρι να φτάσει το πακέτο στο προορισμό του.

Υπάρχει φυσικά και η περίπτωση όπου κάποιος κόμβος ίσως αδυνατεί να προωθήσει κάποιο πακέτο προς το προορισμό ίσως διότι μπορεί να μην υπάρχει κανένας άλλος κόμβος στην εμβέλεια του. Τότε ξεκινά μια διαδικασία η οποία έχει ως σκοπό να βρει μια εναλλακτική διαδρομή που θα επιτρέψει την μεταφορά του πακέτου προς το προορισμό. Βασικά ο κόμβος ο οποίος δεν μπορεί να μεταδώσει το πακέτο κάνει backtrack το πακέτο πίσω στο προηγούμενο κόμβο από τον οποίο είχε λάβει το πακέτο. Έτσι αναλαμβάνει ο κόμβος αυτός

να βρει μια εναλλακτική διαδρομή προς το προορισμό. Εάν δεν μπορεί να βρει ούτε αυτός μία εναλλακτική διαδρομή τότε το στέλνει ακόμα πιο πίσω στον κόμβο από τον οποίο είχε λάβει αρχικά το πακέτο. Η διαδικασία αυτή συνεχίζεται μέχρι να βρει κάποιος κόμβος μια εναλλακτική διαδρομή, κάποιο άλλο κόμβο που να είναι κοντά στο προορισμό δηλαδή, ή μέχρι να φτάσει πίσω στη πηγή το πακέτο όπου και διαγράφεται.

3.1.5.3.Πλεονεκτήματα/Μειονεκτήματα

Η χρήση του GPS (Geographic Positioning System) στο πρωτόκολλο αυτό είναι που το διαφοροποιεί κατά πολύ από τα άλλα πρωτόκολλα. Του δίνει αρκετές ευκολίες στο να μπορεί να βρίσκει εύκολα τους γείτονες του καθώς και να δρομολογεί πιο εύκολα τα διάφορα πακέτα. Αυτό όμως δεν σημαίνει ότι υπάρχουν μόνο θετικά στη χρήση του GPS. Από την μια υπάρχουν αυτές οι ευκολίες οι οποίες ευκολύνουν το πρωτόκολλο όπως στις διαδικασίες εξεύρεσης διαδρομής ή διαχείρισης διαδρομής. Από την άλλη το GPS σαν σύστημα δεν υποστηρίζεται σε όλες τα σημεία του κόσμου. Άρα απαραίτητη προϋπόθεση για τον αλγόριθμο είναι να υπάρχει υποστήριξη GPS στο σημείο που λαμβάνει χώρα το δίκτυο. Αυτό δίνει στον αλγόριθμο στο μειονέκτημα ότι δεν μπορεί να εφαρμοστεί παντού.

3.2 Διαφορές MANETS-VANETS [25]

Στην ουσία τα VANETS μπορούν να θεωρηθούν σαν μια μεγάλη υποκατηγορία των MANETS αφού έχουν αρκετές ομοιότητες. Παρόλα αυτά υπάρχουν και κάποιες σημαντικές διαφορές οι οποίες κάνουν την μια κατηγορία να ξεχωρίζει από την άλλη. Πιο συγκεκριμένα, κάποιες σημαντικές διαφορές μεταξύ των δύο είναι ότι στα MANETS υπάρχουν περιορισμένα αποθέματα ενέργειας σε κάθε κόμβο κάτι το οποίο δεν ισχύει τόσο στα VANETS όπου οι κόμβοι στην ουσία είναι αυτοκίνητα και έχουν αρκετά αποθέματα ενέργειας αφού ως γνωστό υπάρχει μια μπαταρία στο αυτοκίνητο που του δίνει την δυνατότητα να υποστηρίζει αρκετές υπηρεσίες. Επίσης κάτι πολύ σημαντικό είναι ότι στα MANETS οι κόμβοι κινούνται με αργό ρυθμό ενώ στα VANETS οι κόμβοι κινούνται με γρήγορο ρυθμό(αφού είναι αυτοκίνητα) ειδικά εάν πρόκειται για συνθήκες αυτοκινητόδρομου όπου οι ταχύτητες των κόμβων-αυτοκινήτων μπορούν να φτάσουν μέχρι και 100-120Km/h.

Επιπλέον ο αριθμός των κόμβων διαφέρει: στα MANETS μιλούμε για δίκτυα μέχρι και 1000 κόμβους ενώ στα VANETS δεν υπάρχει κάποιος ιδιαίτερος περιορισμός οι κόμβοι μπορεί να φθάσουν και μέχρι κάποια εκατομμύρια αυτοκίνητα ειδικά σε μεγάλους αυτοκινητοδρόμους που καλύπτουν μεγάλες αποστάσεις. Τέλος στα VANETS είναι τυχαία η θέση του κάθε κόμβου ενώ μπορεί να παρατηρηθούν φαινόμενα όπου οι κόμβοι είναι πολύ αραιοί ή πολύ πυκνοί ή ακόμα να παρατηρηθεί το φαινόμενο του network partitioning όπου μια ομάδα αυτοκινήτων ξεκόβει από το υπόλοιπο δίκτυο και χάνει την επικοινωνία μαζί του. Στα MANETS είναι κάπως πιο συντονισμένη η κίνηση των οχημάτων. Αυτό δημιουργεί μεγάλες προκλήσεις όσο αφορά την πιο αποδοτική μεταφορά δεδομένων ακόμα και στις πιο ακραίες περιπτώσεις που έχουμε δει.

Άρα βλέπουμε ότι υπάρχουν διαφορές μεταξύ των δυο κατηγοριών οι οποίες μπορεί να παίξουν καταλυτικό ρόλο στην απόδοση των πρωτόκολλων όταν εφαρμοστούν σε VANETS. Πιο κάτω προσπαθούμε να καθορίσουμε τα χαρακτηριστικά που πρέπει να έχουν τέτοια δίκτυα όπως τα VANETS ώστε να μπορούν να δουλέψουν αποδοτικά κυρίως σε περιπτώσεις που χρειάζεται να γίνει δρομολόγηση κάποιου πακέτου μέσα στο δίκτυο. Αυτό θα μας βοηθήσει να επιλέξουμε και τις κατάλληλες μετρικές βάση των οποίων θα αξιολογήσουμε τα πρωτόκολλα.

3.3 Χαρακτηριστικά Δικτύων VANETS [12] [24] [26] [28] [30] [31]

Στο σημείο αυτό θα προσπαθήσουμε να μελετήσουμε τα χαρακτηριστικά που θα πρέπει να έχει ένα τέτοιο δίκτυο ώστε να είναι αξιόπιστο και αποδοτικό. Αυτά τα χαρακτηριστικά θα μας βοηθήσουν στο να αποφασίσουμε με ποιες μετρικές θα αξιολογήσουμε τα πρωτόκολλα μας. Θα πρέπει επίσης να επικεντρωθούμε σε μετρικές που αφορούν δρομολόγηση πακέτων μεταξύ των διαφόρων κόμβων του δικτύου. Λαμβάνοντας υπόψη ότι στα VANETS έχουμε σκοπό να χρησιμοποιήσουμε και εφαρμογές ασφαλείας που έχουν να κάνουν με την ασφάλεια των επιβατών και όχι μόνο με εφαρμογές διαφήμισης ή ψυχαγωγίας καταλαβαίνουμε ότι τα δίκτυα αυτά πρέπει να επιτυγχάνουν υψηλή απόδοση. Προχωρούμε λοιπόν με την ανάλυση των χαρακτηριστικών τέτοιων δικτύων.

Καταρχήν πρέπει να είναι αξιόπιστο. Η αξιοπιστία του αυτή είναι που θα μας δώσει την δυνατότητα να εξυπηρετήσουμε αρκετές και διαφόρων ειδών εφαρμογές όπως εφαρμογές ασφαλείας, διαφήμιση, ψυχαγωγία. Επιπλέον η αξιοπιστία αυτή θα μας δώσει την δυνατότητα να υπάρχουν πάντα "φρέσκιες"-καινούργιες πληροφορίες και να αποφευχθεί το φαινόμενο όπου πληροφορίες η οποίες παύουν να είναι έγκυρες να συνεχίζουν να κυκλοφορούν μέσα στο δίκτυο. Πέραν τούτου πρέπει να επιτυγχάνει χαμηλό delay (μικρότερο από 150 ms) των δεδομένων ώστε να υπάρχει σωστή και αποτελεσματική χρήση όλων των εφαρμογών. Τρίτο χαρακτηριστικό είναι το υψηλό throughput. Αυτό είναι απαραίτητο να επιτευχθεί ώστε να μπορούν να μεταδίδονται όλα τα είδη πακέτων που κυκλοφορούν σε ένα δίκτυο, είτε είναι πακέτα πληροφοριών είτε είναι control πακέτα. Ένα άλλο χαρακτηριστικό είναι η ευρωστία του δικτύου και γενικά των αλγορίθμων που θα χρησιμοποιηθούν. Είναι σημαντικό να δουλεύει αποτελεσματικά ακόμα και σε ακραίες περιπτώσεις όπου τα πράγματα γίνονται δύσκολα και οι καταστάσεις είναι πιο απαιτητικές. Τέλος, κάτι πολύ σημαντικό είναι αυτό το οποίο λέμε ποιότητα υπηρεσίας (Qos). Αυτό περιλαμβάνει και πολλά από αυτά που έχουν λεχθεί προηγουμένως καθώς επίσης και άλλα όπως την επιτυχία παράδοσης των πακέτων κ.λπ.

Κάτι αρκετά βασικό είναι ότι στα VANETS η όλη λειτουργία του δικτύου γίνεται on-the-fly, δηλαδή δεν πρέπει να υπάρχει κάποιο infrastructure από το οποίο να εξαρτάται το δίκτυο μου. Τέλος το δίκτυο μου πρέπει να είναι scalable, να μπορεί να ανταποκρίνεται όταν υπάρχει ψηλός αριθμός κόμβων για μεγάλες αποστάσεις. Να μπορεί δηλαδή να δουλεύει σταθερά σε όλο το εύρος του δικτύου.

3.4 Μετρικές Αξιολόγησης

Η σωστή αξιολόγηση των πρωτοκόλλων απαιτεί την επιλογή μετρικών οι οποίες θα φανερώσουν πώς τα πρωτόκολλα αυτά θα ανταποκριθούν στις ιδιάζουσες συνθήκες ενός VANET όπως είδαμε και πιο πάνω. Είναι σημαντικό να επιλεγούν οι σωστές μετρικές ώστε να υπάρχει μια καθαρή εικόνα ως προς το που υπερτερεί το κάθε πρωτόκολλο. Οι μετρικές θα πρέπει να εστιαστούν στη δρομολόγηση αφού αυτό είναι που μας ενδιαφέρει κυρίως. Τα πρωτόκολλα αυτά έχουν δημιουργηθεί ώστε να δουλεύουν κάτω από περιβάλλον MANET. Υπάρχουν όμως κάποιες διαφορές μεταξύ των MANETS και των VANETS οι οποίες μπορούν να επηρεάσουν και την ίδια την απόδοση των πρωτοκόλλων. Αυτό έχει φανεί ήδη και πιο πάνω κατά την σύγκριση των δύο μεγάλων κατηγοριών.

Θα παρουσιαστούν μετρικές οι οποίες θα μας βοηθήσουν να αξιολογήσουμε τα πρωτόκολλα από διάφορες οπτικές γωνιές ώστε να είναι πιο αντικειμενική και εύρωστη η αξιολόγηση μας.

- **Routing overhead [18]**

Ετυμολογικά το overhead στα δίκτυα σημαίνει τα bits τα οποία χρησιμοποιούνται σε ένα πακέτο και τα οποία περιέχουν πληροφορίες για τη δρομολόγηση του πακέτου και όχι καθαρά δεδομένα. Άρα στην περίπτωση αυτή το routing overhead περιλαμβάνει όλα τα bits ενός πακέτου ή ακόμα και ολόκληρα πακέτα τα οποία χρησιμοποιούνται για τη δημιουργία, διαχείριση ή ακόμα και την διαγραφή μιας διαδρομής. Είναι σημαντικό να είναι χαμηλό το routing overhead διότι έτσι δεν θα χρησιμοποιεί μεγάλο bandwidth από τα ήδη περιορισμένα resources που υπάρχουν σε τέτοιου είδους δίκτυα. Στην OPNET η μετρική αυτή αναφέρεται σαν routing traffic.

- **Packet delivery ratio [18] [2]**

Η μετρική αυτή αφορά στο ποσοστό (ratio είναι το πηλίκο μιας διαίρεσης) το οποίο προκύπτει από τα πακέτα τα οποία έφτασαν στο προορισμό τους δια τα πακέτα τα οποία στάλθηκαν από κόμβους που είχαν το ρόλο της πηγής. Είναι σημαντική διότι μας δείχνει το ποσοστό επιτυχίας των πακέτων που έχουμε σκοπό να μεταδώσουμε μέσα στο δίκτυο μας.

Αυτή η μετρική μπορεί να εξαχθεί από την OPNET χρησιμοποιώντας το traffic sent και traffic received. Επίσης μπορεί να χρησιμοποιηθεί και η μετρική traffic dropped.

- **Routing Packet transmission Ratio [2] [34]**

Αυτή η μετρική μας δίνει το ποσοστό των πακέτο που στέλνονται από όλους τους κόμβους για διαχείριση και δημιουργία κάποιας διαδρομής (control packets) σε σχέση με τον αριθμό των πακέτων δεδομένων τα οποία φθάνουν στο προορισμό τους. Είναι σημαντική διότι αυτό μας δείχνει εάν τελικά χρησιμοποιώ πολλά πακέτα για να χτίσω μια διαδρομή σε σύγκριση με τα πακέτα πληροφοριών που έχω να στείλω. Αυτή η μετρική μπορεί να εξαχθεί στην OPNET με την βοήθεια του routing traffic sent και του traffic received για τον κάθε αλγόριθμο.

- **End-to end delay [7] [37]**

Εδώ στην ουσία μετρούμε το χρόνο τον οποίο χρειάζεται για να πάει ένα πακέτο από την πηγή στο προορισμό. Αυτός ο χρόνος περιλαμβάνει στην ουσία 4 delays: το transmission, το propagation, το processing και το queuing delay. Είναι σημαντικό διότι προσπαθούμε να κατασκευάσουμε αποτελεσματικά δίκτυα τα οποία να προσφέρουν ποιοτικές υπηρεσίες που βασίζονται κυρίως σε real-time καταστάσεις. Άρα είναι σημαντικό τα πακέτα να φθάνουν έγκαιρα στο προορισμό τους ειδικά εάν πρόκειται για safety applications. Εδώ μπορεί να υπολογιστεί και το average end-to-end delay αντί της μετρικής αυτής. Στην OPNET η μετρική αυτή μας δίνεται από την μετρική MANET.Delay.

- **Throughput και Load [21]**

Οι δύο αυτές μετρικές μας δείχνουν πόσο χρησιμοποιούνται τα resources στο δίκτυο και συγκεκριμένα το bandwidth των αυτοκινήτων-κόμβων του δικτύου. Είναι σημαντικό αφού είναι καλύτερα να χρησιμοποιείται όσο το δυνατό περισσότερο bandwidth για να έχει καλύτερα αποτελέσματα το ad-hoc δίκτυο μου. Εδώ σημαντικό επίσης είναι να μελετηθεί κατά πόσο ανταποκρίνεται ικανοποιητικά το ασύρματο περιβάλλον και με πόσο φορτίο μπορεί να επιφορτιστεί το περιβάλλον αυτό. Εδώ μέσω της OPNET μπορούμε να χρησιμοποιήσουμε τις μετρικές που αφορούν το ασύρματο περιβάλλον όπως Load, WirelessLan, Delay και Retransmission Attempts.

- **Route Discovery Time**

Οι μετρική αυτή μας δείχνει πόσο χρόνο χρειάζεται για να ανακαλυφθεί μία διαδρομή από μια πηγή προς ένα προορισμό. Συγκεκριμένα μετρά το χρόνο από την ώρα που στέλνεται ένα broadcast πακέτο από τη πηγή μέχρι την ώρα που φτάνει ένα Reply πακέτο στην πηγή που περιέχει τη διαδρομή. Η μετρική αυτή υπάρχει αυτούσια για τα πρωτόκολλα AODV, DSR. Επίσης για τον αλγόριθμο TORA υπάρχει κάτι αντίστοιχο το οποίο θα μπορούσε να μας βοηθήσει να κάνουμε την σύγκριση και ονομάζεται Route Discovery Delay που στην ουσία μετρά την ώρα που χρειάζεται για να ανακαλυφθεί μία διαδρομή από ένα προορισμό προς μία πηγή. Από την άλλη η μετρική αυτή δεν υπάρχει στους proactive αλγόριθμους αφού θεωρούμε ότι δεν χρειάζεται οποιοσδήποτε χρόνος για να ανακαλυφθεί κάποια διαδρομή λόγω του ότι οι κόμβοι διατηρούν πληροφορίες για όλους τους κόμβους ανά πάσα χρονική στιγμή. Τέλος σημαντική εδώ μπορεί να θεωρηθεί και η μετρική που αφορά στα cached replies που έχουν αποσταλεί και κατά πόσο αυτό βοηθά στην πιο γρήγορη ανεύρεση της διαδρομής σύμφωνα πάντα με τον τρόπο λειτουργίας στον κάθε αλγόριθμο όπως έχει αναλυθεί πιο πάνω.

- **Number of hops**

Σημαντικό ρόλο στην δρομολόγηση παίζει επίσης και ο αριθμός των hops, το σύνολο των κόμβων που μεσολαβούν μεταξύ των δυο άκρων μίας σύνδεσης. Όσο περισσότεροι είναι οι κόμβοι τόσο πιο ψηλό μπορεί να είναι delay. Άρα σημαντικό είναι να παρατηρήσουμε το πόσο σύντομη είναι η διαδρομή και κατά πόσο αυτό βοηθά στην πιο αξιόπιστη μεταφορά των δεδομένων ή είναι καλύτερα να υπάρχουν περισσότερα hops στην διαδρομή ώστε να διατηρείται περισσότερο χρονικό διάστημα έγκυρη και αποδοτική. Πολλές φορές μπορεί να αποδεικνύεται ότι είναι καλύτερα να έχουμε διαδρομές με περισσότερα hops αφού θα είναι πολύ πιο αξιόπιστες παρά να έχουμε διαδρομές με μικρότερα hops οι οποίες ναι μεν θα μας δίνουν πιο σύντομες διαδρομές αλλά θα "σπάζουν" συνεχώς.

Κεφάλαιο 4

Υλοποίηση και Ανάλυση Σεναρίων

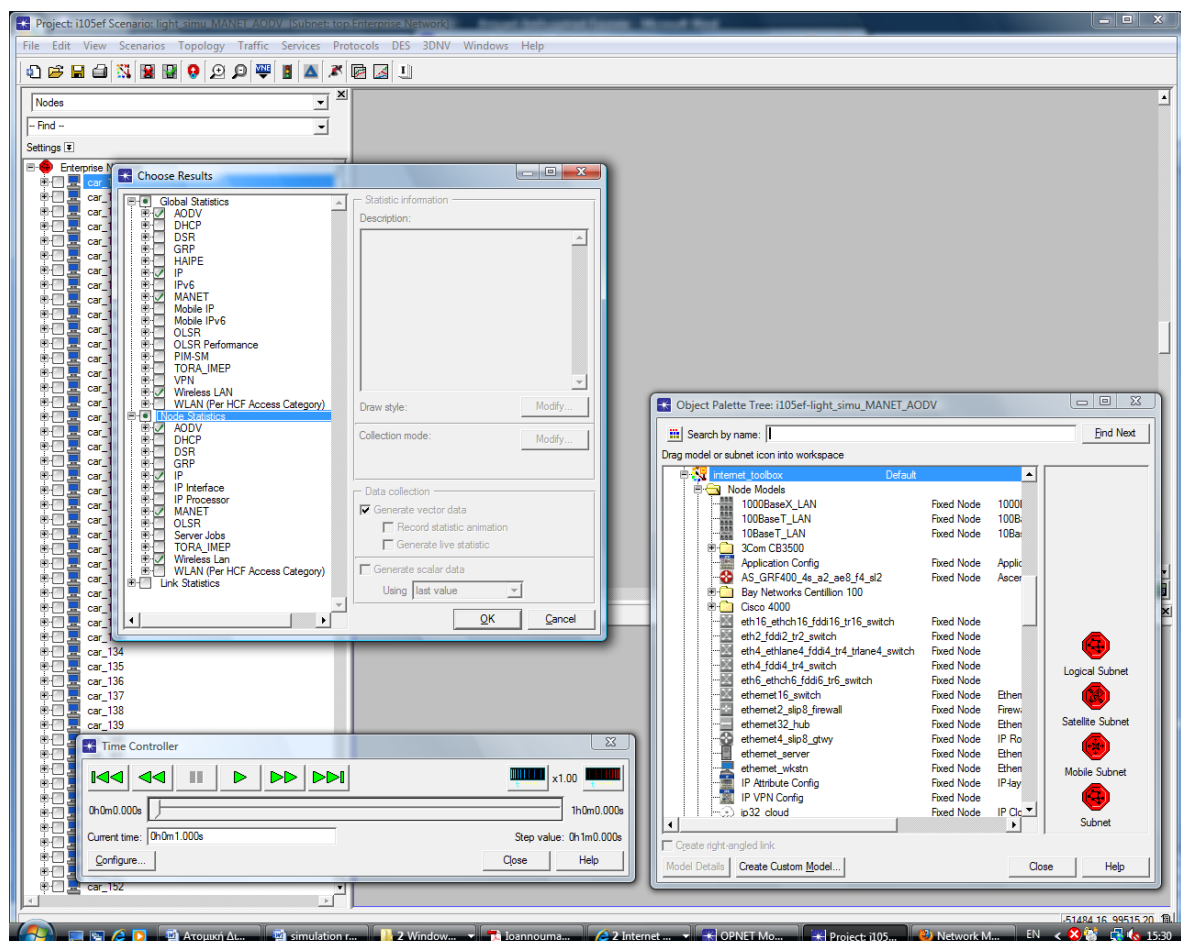
4.1 Καθορισμός Προβλήματος	31
4.2 Ο προσομοιωτής OPNET	32
4.3 Σενάρια αξιολόγησης	33
4.3.1 Γενικά για τα σενάρια	33
4.3.2 Σενάριο Light	36
4.3.3 Σενάριο Medium	38
4.3.4 Σενάριο Heavy	40
4.4 Προβλήματα που προέκυψαν στις προσομοιώσεις	43

4.1 Καθορισμός Προβλήματος

Το πρόβλημα στο οποίο θα βασιστεί η υλοποίηση μου είναι στην ουσία η προσπάθεια δημιουργίας ενός αποδοτικού αλγορίθμου στα VANETS ο οποίος θα μπορεί να χρησιμοποιηθεί σε πραγματικές συνθήκες και θα είναι αποδοτικός, εύρωστος και γρήγορος. Στην προσπάθεια αυτή θα πρέπει να ξεκινήσουμε μελετώντας κάποιους αλγόριθμους οι οποίοι ήδη υπάρχουν και είναι υλοποιημένοι. Έτσι θα μπορούμε να δούμε ποια πλεονεκτήματα και ποια μειονεκτήματα έχουν οι υφιστάμενοι αλγόριθμοι ώστε να γίνουν οι κατάλληλες αλλαγές οι οποίες θα τους κάνουν πιο αποδοτικούς. Αυτό το οποίο θα γίνει εδώ είναι η εφαρμογή των υφιστάμενων αλγόριθμων που υπάρχουν στα MANETS σε δίκτυα τα έχουν δημιουργηθεί και υποστηρίζουν συνθήκες VANETS όπως τις έχουμε αναφέρει πιο πάνω. Έτσι θα προκύψουν συγκεκριμένα αποτελέσματα τα οποία θα φανούν στην συνέχεια.

4.2 Ο προσομοιωτής OPNET

Η OPNET (Optimized Network Engineering Tools) είναι ένας network simulator ο οποίος μας δίνει πολλές δυνατότητες όσο αφορά τη δημιουργία διαφόρων ειδών τοπολογιών όπως ασύρματων, ενσύρματων, κινητών με ή χωρίς infrastructure. Χρησιμοποιεί γραφικό περιβάλλον και μας δίνει την δυνατότητα να επιλέξουμε από μία μεγάλη σειρά εικονιδίων και επιλογών μέσα από το palette το οποίο διαθέτει ώστε να σχεδιάσουμε το δίκτυο το οποίο θέλουμε να σχεδιάσουμε. Ακολούθως μπορούμε να επιλέξουμε τις μετρικές βάση των οποίων θέλουμε να αξιολογήσουμε την απόδοση του δικτύου μας και να δούμε ποια προβλήματα προκύπτουν, που προκαλείται συμφόρηση στο δίκτυο και ποιοι παράγοντες επηρεάζουν την αποδοτικότητα του δικτύου.



Σχήμα 4.1: Ο προσομοιωτής OPNET

Όσο αφορά τις δυνατότητες του προσομοιωτή αυτού, μας βοήθησε πολύ λόγω των πολλών εφαρμογών που έχει όπως η δυνατότητα εισαγωγής σεναρίων τα οποία έχουν δημιουργηθεί από άλλα εργαλεία όπως το VISSIM το οποίο μας δίνει την δυνατότητα να δημιουργήσουμε σενάρια αυτοκινητόδρομων με πραγματικές συνθήκες και να τα εισάγουμε στο προσομοιωτή. Επίσης μας δίνει την δυνατότητα να επιλέξουμε το είδους του κόμβου που θέλουμε να χρησιμοποιήσουμε, να αναθέσουμε IP διευθύνσεις και να προσθέσουμε traffic στο δίκτυο μας. Τέλος μπορούμε να επιλέξουμε γραφικές από τις πάρα πολλές που μας προσφέρει το εργαλείο αυτό ώστε να μπορέσουμε να αξιολογήσουμε όσο το δυνατό καλύτερα την λειτουργία του δικτύου μας. Για τις προσομοιώσεις χρησιμοποιήσαμε την έκδοση 15.0 της OPNET.

4.3 Σενάρια αξιολόγησης

4.3.1 Γενικά για τα σενάρια

Με τις προσομοιώσεις των σεναρίων αυτών προσδοκούμε να κάνουμε μια αναλυτική αξιολόγηση στους αλγόριθμους αυτούς που έχουμε προαναφέρει ώστε η ανάλυση αυτή να μπορέσει να αποτελέσει την απαρχή για την περαιτέρω έρευνα και μελέτη στην κατεύθυνση δημιουργίας καινούργιου ή τροποποίησης κάποιου από τους αλγόριθμους ώστε να μπορεί να είναι συμβατός και αποδοτικός με τις ανάγκες δρομολόγησης οι οποίες προκύπτουν στα δίκτυα VANETS. Καταρχήν να αναφέρουμε ότι τα σενάρια έχουν παρθεί από ένα αυτοκινητόδρομο στην Αμερική, τον i105. Η κίνηση των κόμβων αντιπροσωπεύει την κίνηση των αυτοκινήτων σε πραγματικές συνθήκες με διάφορες ταχύτητες αλλά αφορά κίνηση μόνο μίας κατεύθυνσης.

Να αναφέρουμε επίσης ότι για την αξιολόγηση των αλγόριθμων θα χρησιμοποιήσουμε 3 σενάρια. Το πρώτο είναι το σενάριο Light, το δεύτερο είναι το σενάριο Medium και το τρίτο το σενάριο Heavy. Οι διαφορές των σεναρίων έχουν να κάνουν με τον αριθμό των κόμβων που χρησιμοποιούνται καθώς και με την ταχύτητα με την οποία κινούνται οι κόμβοι. Όπως έχουμε προαναφέρει ο σκοπός μας είναι να αξιολογήσουμε την δρομολόγηση πακέτων(routing) με τον κάθε αλγόριθμο ξεχωριστά. Για το λόγο αυτό δημιουργήσαμε Raw Traffic από συγκεκριμένους αποστολείς σε συγκεκριμένους παραλήπτες. Επιλέξαμε για κάθε σενάριο τρεις διαφορετικές ποσότητες traffic και προσπαθήσαμε να τοποθετήσουμε όσο το δυνατό περισσότερους κόμβους ενδιάμεσα ώστε να κάνουμε πιο δύσκολη τη διαδικασία

εξευρέσεων διαδρομής και αποστολής πακέτων. Οι τρεις διαφορετικές ποσότητες traffic διαφέρουν όσο αφορά την ποσότητα των αποστολών στα σενάρια μας. Συγκεκριμένα έχουμε τρέξει το κάθε σενάριο (Light, Medium και Heavy) με αποστολές 5, 10 και 15 κόμβους. Επίσης ως επί το πλείστον οι αποστολές είναι στην αρχή της κίνησης του σεναρίου και οι παραλήπτες είναι στο τέλος. Εκτός τούτου το traffic ξεκινά να αποστέλλεται από το 50 δευτερόλεπτο μέχρι το τέλος της προσομοίωσης ώστε να μπορούμε να βλέπουμε ανά πάσα χρονική στιγμή πόσο επηρεάζεται η αποστολή των πακέτων λόγω της κίνησης των κόμβων. Τέλος να αναφέρουμε ότι έχουμε αναθέσει σε κάθε κόμβο το δικό του ip και subnet mask ώστε να μπορούμε να κάνουμε δρομολόγηση των πακέτων ενώ το είδος του κόμβου που έχει επιλεγεί είναι το MANET station το οποίο μας δίνει την δυνατότητα εισαγωγής Raw Traffic στο δίκτυο και της δρομολόγησης της βάση του ip κάθε κόμβου. Η ανάθεση της διεύθυνσης ip φαίνεται στο σχήμα 4.2.

IP Host Parameters	(...)
Interface Information	(...)
Name	IF0
Address	128.1.0.2
Subnet Mask	255.255.0.0
MTU (bytes)	WLAN
Compression Information	None

Σχήμα 4.2: Ανάθεση ip και subnet mask σε κάθε κόμβο

Πέρα από την ανάθεση διευθύνσεων ip, προσθέσαμε και Raw Traffic στο δίκτυο όπως φαίνεται στο σχήμα 4.3. Όπως μπορούμε να δούμε το traffic ξεκινά από το 50 δευτερόλεπτο μέχρι το τέλος της προσομοίωσης και κάθε 1 δευτερόλεπτο στέλνουμε ένα πακέτο μεγέθους 1024 bits. Έτσι εάν οποιαδήποτε στιγμή προκύψει οποιοδήποτε πρόβλημα στην παράδοση πακέτων στο προορισμό αυτό θα φανεί στην αξιολόγηση των αποτελεσμάτων.

MANET Traffic Generation Parameters	(...)
Number of Rows	1
50	
Start Time (seconds)	50
Packet Inter-Arrival Time (seconds)	exponential (1)
Packet Size (bits)	exponential (1024)
Destination IP Address	128.1.0.81
Stop Time (seconds)	End of Simulation

Σχήμα 4.3: Δημιουργία Raw Traffic για το κάθε αποστολέα στο δίκτυο.

Επίσης όσο αφορά τις παραμέτρους για το ασύρματο δίκτυο έχουμε επιλέξει το πρωτόκολλο 801.12a(OFDM) με ταχύτητα μεταφοράς τα 12 Mbps. Ο λόγος που έχουμε επιλέξει αυτό το πρωτόκολλο είναι γιατί είναι αυτό από το οποίο έχει δημιουργηθεί το πρωτόκολλο 802.11p το οποίο χρησιμοποιείται σε τέτοιου είδους προσομοιώσεις. Επιπλέον θέσαμε το transmit power στα 0.0039W από 0.005W που είναι το αρχικό, που δίνει σε κάθε κόμβο εμβέλεια περίπου ~300-310m. Οι υπόλοιπες παράμετροι έχουν παραμείνει σταθερές και φαίνονται στο σχήμα 4.4.

[-] Wireless LAN	
Wireless LAN MAC Address	82
[-] Wireless LAN Parameters	(...)
BSS Identifier	Auto Assigned
Access Point Functionality	Disabled
Physical Characteristics	OFDM (802.11a)
Data Rate (bps)	12 Mbps
[+] Channel Settings	Auto Assigned
Transmit Power (W)	0.0039
Packet Reception-Power Threshold...	-95
Rts Threshold (bytes)	None
Fragmentation Threshold (bytes)	2000
CTS-to-self Option	Enabled
Short Retry Limit	7
Long Retry Limit	4
AP Beacon Interval (secs)	0.02
Max Receive Lifetime (secs)	0.5
Buffer Size (bits)	256000
Roaming Capability	Disabled
Large Packet Processing	Drop
[+] PCF Parameters	Disabled
[+] HCF Parameters	Not Supported

Σχήμα 4.4: Ρυθμίσεις ασύρματου δικτύου για κάθε κόμβο

Για κάθε αλγόριθμο χρησιμοποιήσαμε τις αρχικές ρυθμίσεις που μας δίνονται εκτός από κάποιες αλλαγές οι οποίες έγιναν στον αλγόριθμο OLSR στο σενάριο medium και heavy όπου μας περιόρισαν οι αυξημένες ανάγκες του αλγόριθμου σε μνήμη αφού οι υπολογιστές δεν μπορούσαν να αντέξουν το βάρος της προσομοίωσης. Έτσι αναγκαστήκαμε να αλλάξουμε κάποιες ρυθμίσεις, να τις κάνουμε πιο «χαλαρές» ώστε να μπορέσει να τρέξει ο αλγόριθμος χωρίς προβλήματα. Οι αλλαγές που έγιναν είναι οι εξής: στο σενάριο medium αλλάξαμε το hello interval από 2 σε 7 δευτερόλεπτα και το TC interval από 5 σε 9 δευτερόλεπτα ενώ στο σενάριο heavy αλλάξαμε το hello interval από 2 σε 12 δευτερόλεπτα

και το TC interval από 5 σε 14 δευτερόλεπτα. Οι αλλαγές αυτές καθώς και όλες οι ρυθμίσεις των αλγορίθμων αυτών φαίνονται στο παράρτημα Α.

4.3.2 Σενάριο Light

Το σενάριο αυτό αποτελείται από 215 περίπου κόμβους οι οποίοι κινούνται σε ένα αυτοκινητόδρομο. Κάθε κόμβος αντιπροσωπεύει τη κίνηση ενός αυτοκίνητου. Όπως έχουμε προαναφέρει έχουμε δημιουργήσει σενάρια με τρεις διαφορετικές ποσότητες traffic: στο πρώτο σενάριο στέλνουν πληροφορίες μόνο 5 κόμβοι, στο δεύτερο στέλνουν πληροφορίες 10 κόμβοι και στο τρίτο στέλνουν 15 κόμβοι. Το traffic το οποίο δημιουργήσαμε για την αξιολόγηση του κάθε αλγόριθμου στο σενάριο αυτό είναι το εξής:

Για το σενάριο με τους 5 κόμβους ως αποστολές δημιουργήσαμε το traffic που φαίνεται στον πιο κάτω πίνακα.

Πίνακας 4.1 : Traffic για το σενάριο Light

Source No. (car number)	Destination No. (car number-ip)
82	176(128.1.0.81)
115	156(128.1.0.62)
131	168(128.1.0.74)
101	174(128.1.0.79)
143	162(128.1.0.68)

Για το σενάριο με τους 10 κόμβους ως αποστολές διατηρήσαμε το υφιστάμενο traffic και προσθέσαμε επιπλέον το traffic που βλέπουμε πιο κάτω:

Πίνακας 4.2 :Επιπλέον Traffic για το σενάριο Light

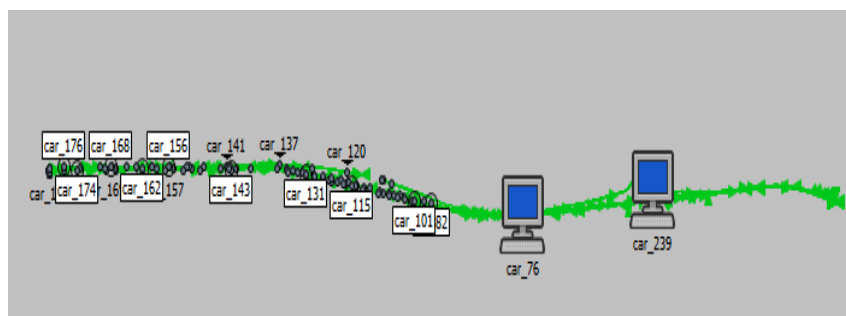
Source No. (car number)	Destination No. (car number-ip)
94	304(128.1.0.208)
102	171(128.1.0.77)
109	170(128.1.0.76)
291	164(128.1.0.70)
103	173(128.1.0.78)

Για το σενάριο με τους 15 κόμβους ως αποστολείς διατηρήσαμε το υφιστάμενο traffic που έχουμε δει πιο πάνω και προσθέσαμε επιπλέον το traffic που βλέπουμε πιο κάτω.

Πίνακας 4.3 : Traffic για το σενάριο Light με 15 κόμβους ως αποστολείς

Source No. (car number)	Destination No. (car number-ip)
177	163(128.1.0.69)
125	165(128.1.0.71)
95	169(128.1.0.75)
96	175(128.1.0.80)
98	178(128.1.0.83)

Το σενάριο Light φαίνεται στο σχήμα πιο κάτω:



Σχήμα 4.5: το σενάριο Light

Στο πιο πάνω σχήμα μπορούμε να δούμε του κόμβους του δικτύου μας καθώς και τις κατευθύνσεις τους (trajectories) οι οποίες φαίνονται με πράσινο χρώμα. Επίσης έχουμε επιλέξει τους κόμβους οι οποίοι αποτελούν είτε τους αποστολείς είτε τους παραλήπτες ώστε να φανεί ότι οι μεν αποστολείς είναι στην αρχή του σεναρίου ενώ οι δε παραλήπτες είναι στο τέλος του σεναρίου. Η κίνηση των κόμβων όπως μπορούμε να δούμε από το time controller της OPNET γίνεται για περίπου 300 δευτερόλεπτα ο οποίος είναι και ο χρόνος της προσομοίωσης μας. Οι ταχύτητες στις οποίες κινούνται οι κόμβοι του δικτύου μου κυμαίνονται σε γενικές γραμμές από 107-120 Km/h. Αυτό συμβαίνει ως επί το πλείστον αφού υπάρχουν και περιπτώσεις όπου οι ταχύτητες μειώνονται στα 70Km/h ή αυξάνονται στα 143Km/h. Όσο αφορά το σενάριο τώρα βλέπουμε ότι έχουμε κίνηση για περίπου 3.5Km. Αυτό θεωρείται ότι είναι και το μήκος του σεναρίου.

4.3.3 Σενάριο Medium

Το σενάριο αυτό αποτελείται από 340 περίπου κόμβους οι οποίοι κινούνται σε ένα highway. Κάθε κόμβος αντιπροσωπεύει τη κίνηση ενός αυτοκινήτου. Όπως έχουμε προαναφέρει και στο σενάριο Light έχουμε δημιουργήσει σενάρια με τρεις διαφορετικές ποσότητες traffic: στο πρώτο σενάριο στέλνουν πληροφορίες μόνο 5 κόμβοι, στο δεύτερο στέλνουν πληροφορίες 10 κόμβοι και στο τρίτο στέλνουν 15 κόμβοι. Το traffic το οποίο δημιουργήσαμε για την αξιολόγηση του κάθε αλγόριθμου στο σενάριο αυτό είναι το εξής:

Για το σενάριο με τους 5 κόμβους ως αποστολείς δημιουργήσαμε το traffic που φαίνεται στον πιο κάτω πίνακα.

Πίνακας 4.4: Traffic για το σενάριο Medium

Source No. (car number)	Destination No. (car number-ip)
124	259(128.1.0.175)
91	247(128.1.0.163)
126	238(128.1.0.154)
156	424(128.1.0.84)
113	218(128.1.0.134)

Για το σενάριο με τους 10 κόμβους ως αποστολείς διατηρήσαμε το υφιστάμενο traffic πιο πάνω και προσθέσαμε επιπλέον το traffic που βλέπουμε πιο κάτω.

Πίνακας 4.5 :Επιπλέον Traffic για το σενάριο Medium

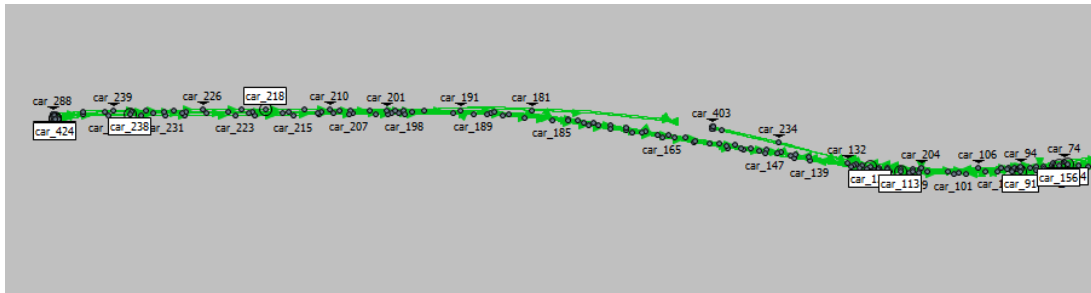
Source No. (car number)	Destination No. (car number-ip)
62	215(128.1.0.131)
80	217(128.1.0.133)
94	221(128.1.0.137)
84	231(128.1.0.147)
204	423(128.1.1.83)

Για το σενάριο με τους 15 κόμβους ως αποστολείς διατηρήσαμε το υφιστάμενο traffic που έχουμε δει πιο πάνω και προσθέσαμε επιπλέον το traffic που βλέπουμε πιο κάτω.

Πίνακας 4.6 : Traffic για το σενάριο Medium με 15 κόμβους ως αποστολείς

Source No. (car number)	Destination No. (car number-ip)
74	219(128.1.0.135)
75	224(128.1.0.140)
77	226(128.1.0.142)
102	232(128.1.0.148)
112	241(128.1.0.157)

Το σενάριο Medium φαίνεται στο σχήμα πιο κάτω:



Σχήμα 4.6 : Το σενάριο Medium

Στο πιο πάνω σχήμα μπορούμε να δούμε του κόμβους του δικτύου μας καθώς και τις κατευθύνσεις τους (trajectories) οι οποίες φαίνονται με πράσινο χρώμα. Επίσης έχουμε επιλέξει τους κόμβους οι οποίοι αποτελούν είτε τους αποστολείς είτε τους παραλήπτες ώστε να φανεί ότι οι μεν αποστολείς είναι στην αρχή του σεναρίου ενώ οι δε παραλήπτες είναι στο τέλος του σεναρίου. Η κίνηση των κόμβων όπως μπορούμε να δούμε από το time controller της OPNET γίνεται για περίπου 300 δευτερόλεπτα ο οποίος είναι και ο χρόνος της προσομοίωσης μας. Οι ταχύτητες στις οποίες κινούνται οι κόμβοι του δικτύου μου κυμαίνονται σε γενικές γραμμές από 107-120 Km/h. Αυτό συμβαίνει ως επί το πλείστον αφού υπάρχουν και περιπτώσεις όπου οι ταχύτητες μειώνονται στα 70Km/h ή αυξάνονται στα 143Km/h. Βλέπουμε ότι συμβαίνει το ίδιο όπως και για το σενάριο Light. Όσο αφορά το σενάριο τώρα βλέπουμε ότι έχουμε κίνηση για περίπου 3.5Km. Αυτό θεωρείται ότι είναι και το μήκος του σεναρίου.

4.3.4 Σενάριο Heavy

Το σενάριο αυτό αποτελείται από 900 περίπου κόμβους οι οποίοι κινούνται σε ένα αυτοκινητόδρομο. Κάθε κόμβος αντιπροσωπεύει τη κίνηση ενός αυτοκίνητου. Οι μόνες αλλαγές που έχουμε κάνει λόγω περιορισμού στην μνήμη των μηχανών που τρέξαμε τα σενάρια είναι το hello interval από 2 σε 12 δευτερόλεπτα και το TC interval από 5 σε 14 δευτερόλεπτα. Όπως έχουμε προαναφέρει και στα προηγούμενα σενάρια έχουμε δημιουργήσει σενάρια με τρεις διαφορετικές ποσότητες traffic: στο πρώτο σενάριο στέλνουν πληροφορίες μόνο 5 κόμβοι, στο δεύτερο στέλνουν πληροφορίες 10 κόμβοι και στο τρίτο στέλνουν 15 κόμβοι. Το traffic το οποίο δημιουργήσαμε για την αξιολόγηση του κάθε αλγόριθμου στο σενάριο αυτό είναι το εξής:

Πίνακας 4.7: Traffic για το σενάριο Heavy

Source No. (car number)	Destination No. (car number-ip)
260	978(128.1.2.78)
357	1158(128.1.3.2)
547	980(128.1.2.80)
326	951(128.1.2.53)
961	912(128.1.2.15)

Για το σενάριο με τους 10 κόμβους ως αποστολείς διατηρήσαμε το υφιστάμενο traffic πιο πάνω και προσθέσαμε επιπλέον το traffic που βλέπουμε πιο κάτω.

Πίνακας 4.8 :Επιπλέον Traffic για το σενάριο Heavy

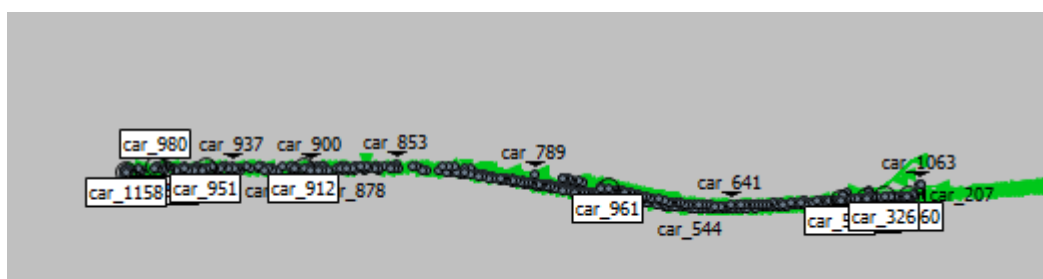
Source No. (car number)	Destination No. (car number-ip)
319	893(128.1.1.254)
521	928(128.1.2.31)
297	963(128.1.2.65)
562	981(128.1.2.81)
598	985(128.1.2.85)

Για το σενάριο με τους 15 κόμβους ως αποστολείς διατηρήσαμε το υφιστάμενο traffic που έχουμε δει πιο πάνω και προσθέσαμε επιπλέον το traffic που βλέπουμε πιο κάτω.

Πίνακας 4.9 : Traffic για το σενάριο Heavy με 15 κόμβους ως αποστολείς

Source No. (car number)	Destination No. (car number-ip)
540	969(128.1.2.71)
686	950(128.1.2.52)
691	955(128.1.2.57)
1150	957(128.1.2.59)
333	959(128.1.2.61)

Το σενάριο Heavy φαίνεται στο σχήμα πιο κάτω:



Σχήμα 4.7: Το σενάριο Heavy

Στο πιο πάνω σχήμα μπορούμε να δούμε του κόμβους του δικτύου μας καθώς και τις κατευθύνσεις τους (trajectories) οι οποίες φαίνονται με πράσινο χρώμα. Επίσης έχουμε επιλέξει τους κόμβους οι οποίοι αποτελούν είτε τους αποστολείς είτε τους παραλήπτες ώστε να φανεί ότι οι μεν αποστολείς είναι στην αρχή του σεναρίου ενώ οι δε παραλήπτες είναι στο τέλος του σεναρίου. Η κίνηση των κόμβων όπως μπορούμε να δούμε από το time controller της OPNET γίνεται για περίπου 800 δευτερόλεπτα ο οποίος είναι και ο χρόνος της προσομοίωσης μας. Οι ταχύτητες στις οποίες κινούνται οι κόμβοι του δικτύου μου κυμαίνονται σε γενικές γραμμές από 0-70 Km/h. Οι πιο συχνές τιμές που εμφανίζονται είναι τα 35-36Km/h . Όσο αφορά το σενάριο τώρα βλέπουμε ότι έχουμε κίνηση για περίπου 3.5Km. Αυτό θεωρείται ότι είναι και το μήκος του σεναρίου.

4.4 Προβλήματα που προέκυψαν στις προσομοιώσεις

Κατά την διάρκεια των προσομοιώσεων που έγιναν προέκυψαν κάποια προβλήματα με συγκεκριμένους αλγόριθμους. Πιο συγκεκριμένα ο αλγόριθμος TORA δεν τρέχει καθόλου με τις πιο πάνω ρυθμίσεις (0 δευτερόλεπτα). Επίσης ο αλγόριθμος GRP δεν καταφέρνει να ολοκληρώσει την προσομοίωση και σταματά στα 2 λεπτά και 30 δευτερόλεπτα στα σενάρια Light και Medium ενώ στο Heavy σταματά στα 10 λεπτά περίπου (600 δευτερόλεπτα). Φυσικά ο αλγόριθμος αυτός έχει παράξει κάποιες γραφικές παραστάσεις, όχι όμως όλες αυτές τις οποίες θα θέλαμε. Γι' αυτό δεν έχουμε προσθέσει καθόλου τον αλγόριθμο αυτό στα αποτελέσματα μας. Στον αλγόριθμο TORA φαίνεται ότι επηρεάζει ο αριθμός των κόμβων του δικτύου καθώς και η ανάθεση διευθύνσεων IP στο δίκτυο. Αυτό διαφάνηκε από τις αλληπάλληλες προσομοιώσεις τις οποίες κάναμε αλλάζοντας κάθε φορά συγκεκριμένες παραμέτρους. Επίσης προσπαθήσαμε κάνοντας αλληπάλληλες προσομοιώσεις, αλλάζοντας κάθε φορά αρκετές από τις παραμέτρους του κάθε αλγόριθμου, παραμέτρους οι οποίες αφορούσαν είτε τις ρυθμίσεις του αλγορίθμου είτε το ασύρματο περιβάλλον είτε το IP layer, να δούμε αν επηρεάζει κάποια παράμετρος και στον αλγόριθμο GRP ώστε να παρατηρήσουμε και να δούμε εάν καταφέρουμε να τρέξουμε κανονικά τα σενάρια αυτά. Δεν προέκυψε κάποια βελτίωση και έτσι απευθυνθήκαμε στην OPNET αποστέλλοντας request στην σελίδα που έχουν για υποστήριξη του προσομοιωτή τους. Έτσι μας ανέφεραν κάποιο τρόπο με τον οποίο μπορούμε να αποστείλουμε τα σενάρια υπό μορφή package και τα αποστείλαμε. Μέχρι στιγμής δεν έχουμε λάβει κάποια απάντηση.

Όπως έχουμε προαναφέρει αντιμετωπίσαμε πρόβλημα και με τον αλγόριθμο OLSR στα σενάρια Medium και Heavy λόγω αυξημένων απαιτήσεων στην μνήμη της μηχανής που τρέχαμε τα σενάρια το οποίο και επιλύσαμε με τις αλλαγές που κάναμε στους δύο αλγορίθμους όπως τις έχουμε αναφέρει στο υποκεφάλαιο που μιλήσαμε γενικά περί σεναρίων. Οι αλλαγές που έγιναν είναι οι εξής: στο σενάριο medium αλλάξαμε το hello interval από 2 σε 7 δευτερόλεπτα και το TC interval από 5 σε 9 δευτερόλεπτα ενώ στο σενάριο heavy αλλάξαμε το hello interval από 2 σε 12 δευτερόλεπτα και το TC interval από 5 σε 14 δευτερόλεπτα. Οι αλλαγές αυτές καθώς και όλες οι ρυθμίσεις των αλγορίθμων αυτών φαίνονται στο παράρτημα Α. Οι αλλαγές αυτές έχουν μειώσει την απόδοση του αλγόριθμου και την έχουν κάνει χειρότερη και αυτό θα φανεί από τα αποτελέσματα τα οποία θα παρουσιαστούν στην συνέχεια.

Κεφάλαιο 5

Αποτελέσματα

5.1 Αποτελέσματα σεναρίου Light	44
5.1.1 Global Αποτελέσματα	45
5.1.2 Μελέτη σύνδεσης κόμβων 82-116	50
5.2 Αποτελέσματα σεναρίου Medium	54
5.2.1 Global Αποτελέσματα	54
5.2.2 Μελέτη σύνδεσης κόμβων 124-259	59
5.3 Αποτελέσματα σεναρίου Heavy	63
5.3.1 Global Αποτελέσματα	63
5.3.2 Μελέτη σύνδεσης κόμβων 260-978	68
5.4 Σύγκριση μεταξύ σεναρίων και Scalar Αποτελέσματα	71
5.5 Συνολικά Αποτελέσματα	78

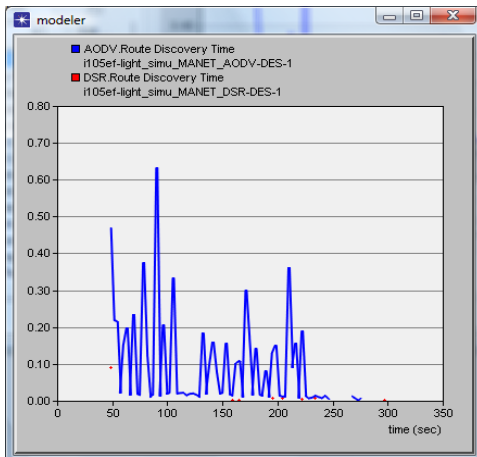
5.1 Αποτελέσματα σεναρίου Light

Να αναφέρουμε ότι τα αποτελέσματα τα οποία λάβαμε χρησιμοποιώντας κάθε φορά διαφορετικό traffic, όπως προαναφέραμε, δεν παρουσίασαν ιδιαίτερες διαφορές μεταξύ τους, γι αυτό και εδώ θα αναφερθούμε μόνο στα αποτελέσματα τα οποία πήραμε όταν το traffic μας δημιουργείτο από 5 κόμβους μόνο. Εάν κάποια από τις υπόλοιπες γραφικές παρουσιάσει κάποιο επιπλέον ενδιαφέρον θα αναφερθεί. Από εκεί και πέρα οι υπόλοιπες γραφικές παραστάσεις θα παρουσιαστούν στο Παράρτημα β.

5.1.1 Global Αποτελέσματα

Μετά την προσομοίωση των σεναρίων και την επιλογή των μετρικών αξιολόγησης, λάβαμε τα αποτελέσματα τα οποία φαίνονται πιο κάτω:

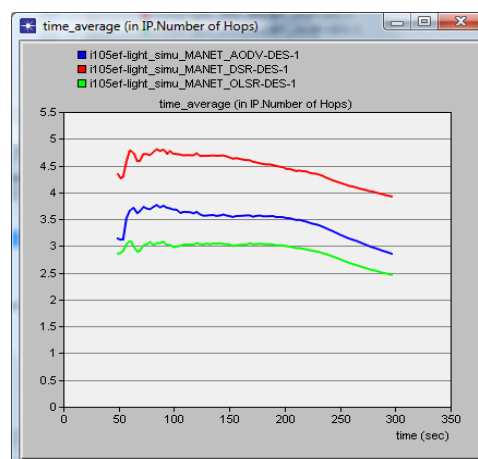
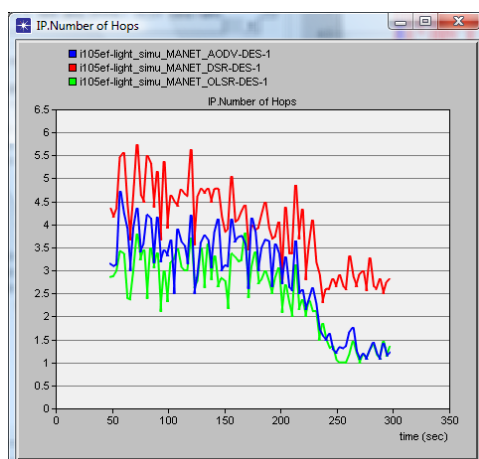
Route Discovery Time (AODV, DSR)



Εδώ μπορούμε να δούμε το χρόνο που χρειάζεται για να ανακαλυφθεί μία διαδρομή προς κάποιο προορισμό. Αυτό περιλαμβάνει και την αποστολή του Reply πακέτου στον αποστολέα. Όπως βλέπουμε αυτή η μετρική μας παρέχεται μόνο για τους αλγόριθμους AODV, DSR. Αυτό που παρατηρούμε είναι ότι ο DSR χρειάζεται πολύ λιγότερο χρόνο για να βρει μία διαδρομή ενώ επίσης αυτό δεν γίνεται συνεχώς όπως φαίνεται να συμβαίνει με τον αλγόριθμο AODV. Αυτό

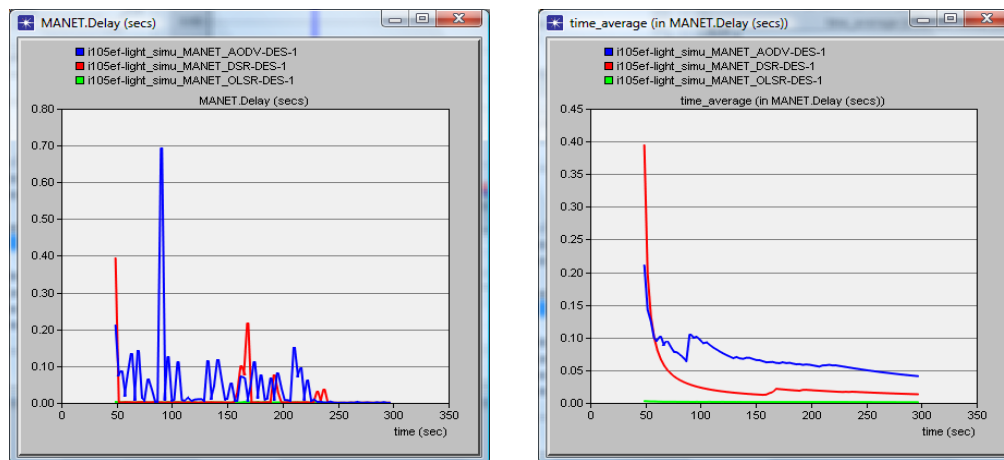
το οποίο φαίνεται να συμβαίνει με τον αλγόριθμο αυτό είναι ότι σπάζουν συνεχώς τα links και χρειάζεται συνεχώς να ανακαλύπτεται καινούργια διαδρομή προς τους προορισμούς αυτούς. Για την μετρική αυτή μπορούμε να πούμε ότι ο επικρατεί ο αλγόριθμος DSR έναντι του AODV. Από την άλλη, όσο αφορά τον αλγόριθμο OLSR, λόγω του ότι ο αλγόριθμος είναι proactive δεν χρειάζεται τέτοια διαδικασία για εξεύρεση διαδρομής αφού έχει ανά πάσα στιγμή διαδρομές προς όλους τους κόμβους του δικτύου. Εκείνο το οποίο έχει ενδιαφέρον στην συνέχεια είναι να δούμε κατά πόσο αυτό επηρεάζει τον αλγόριθμο στην την καθυστέρηση στην μετάδοση των πακέτων.

IP.number of Hops (AODV, DSR, OLSR)



Στις δύο πιο πάνω γραφικές παραστάσεις μπορούμε να δούμε τον αριθμό των κόμβων οι οποίοι μεσολαβούν ώστε να φτάσει κάποιο πακέτο στο προορισμό του. Στα αριστερά είναι η παράσταση As Is ενώ δεξιά βλέπουμε την γραφική παράσταση του time average. Αυτό το οποίο παρατηρούμε είναι ότι ο αλγόριθμος OLSR υπερνικά τους άλλους δύο αλγόριθμους αφού έχει σαν average τα 3 Hops σε αντίθεση με τον DSR που έχει average τα 4.5 Hops και τον AODV ο οποίος κινείται average κοντά στα 3.5 Hops. Ο λόγος που βλέπουμε το αποτέλεσμα αυτό είναι διότι ο OLSR ως αλγόριθμος proactive δίνει την δυνατότητα σε κάθε κόμβο του να διατηρεί πληροφορίες προς όλους τους προορισμούς με αποτέλεσμα να μπορεί να επιλέξει διαδρομές κοντά στις βέλτιστες. Από την άλλη οι reactive αλγόριθμοι μπορεί να μην βρίσκουν διαδρομές κοντά στις βέλτιστες και κυρίως ο DSR ο οποίος χρησιμοποιεί σε μεγάλο βαθμό τα cached replies με αποτέλεσμα να δίνει πολλές φορές απαντήσεις οι οποίες δεν περιέχουν την βέλτιστη διαδρομή. Σημαντικό είναι να δούμε κατά πόσο και εδώ ο αριθμός των Hops επηρεάζει την καθυστέρηση στην μετάδοση των πακέτων.

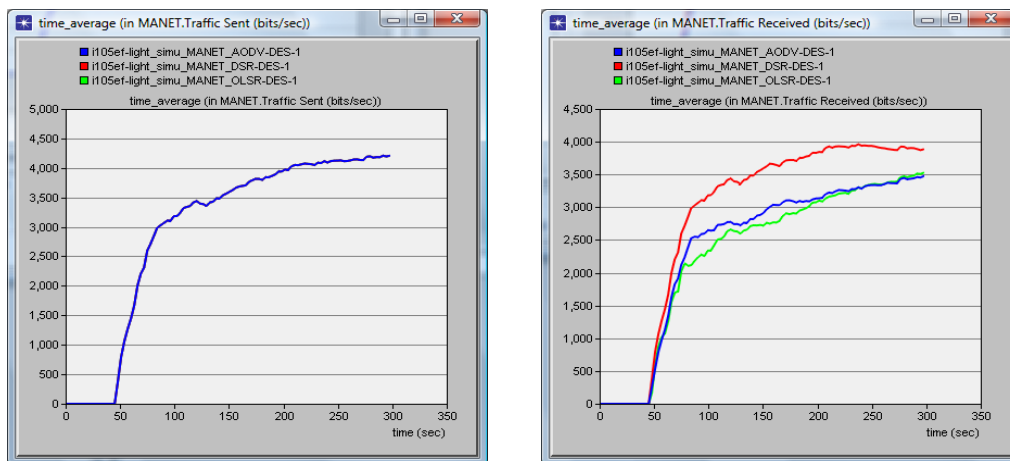
MANET. Delay (AODV, DSR, OLSR)



Πιο πάνω βλέπουμε τις γραφικές παραστάσεις του end-to-end delay για τον κάθε αλγόριθμο. Η γραφική παράσταση στα αριστερά είναι η As Is ενώ στα δεξιά έχουμε την παράσταση του time average. Το end-to-end delay αφορά στον χρόνο ο από την δημιουργία του MANET πακέτου στον αποστολέα μέχρι και την καταστροφή του στον παραλήπτη. Όπως μπορούμε να δούμε και πιο πάνω ο OLSR έχει delay πάρα πολύ κοντά στο μηδέν. Αυτό οφείλεται στην δυνατότητα του αλγορίθμου να διατηρεί ανά πάσα στιγμή διαδρομές προς όλους τους προορισμούς με αποτέλεσμα να είναι όσο το δυνατό πιο μικρός ο χρόνος αυτός. Από την άλλη όσο αφορά τους reactive αλγόριθμους βλέπουμε ότι ο DSR είναι λίγο πιο πάνω από τον OLSR και γύρω στα 0.05 δευτερόλεπτα βρίσκεται ο AODV ο οποίος έχει και το μεγαλύτερο

delay. Φυσικά βλέπουμε να υπάρχουν και κάποιες στιγμές όπου οι δύο reactive αλγόριθμοι παρουσιάζουν απότομη άνοδο όπως φαίνεται στην γραφική παράσταση του As Is. Αυτό συμβαίνει κυρίως στον αλγόριθμο AODV ο οποίος σε κάποιες στιγμές φτάνει στα 0.7 δευτερόλεπτα. Εκείνο το οποίο επιζητούμε από ένα αλγόριθμο ο οποίος θα εφαρμοστεί σε δίκτυα VANETS και θα μπορεί να υποστηρίξει ίσως και κάποιες εφαρμογές είναι να διατηρεί το delay του συνεχώς σε χαμηλά επίπεδα μικρότερα στην χειρότερη περίπτωση από 400msec.

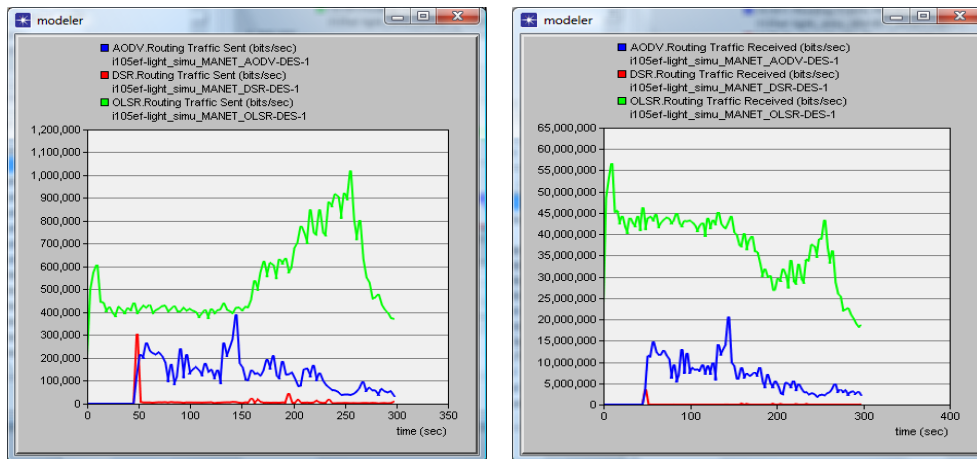
MANET. Traffic Received/Sent (bits/second) (AODV, DSR, OLSR)



Στις δύο γραφικές παραστάσεις πιο πάνω βλέπουμε το traffic sent και received για τους τρεις αλγόριθμους. Οι γραφικές παραστάσεις που χρησιμοποιούμε εδώ αφορούν στο time average λόγω του ότι οι As Is γραφικές δεν μας βοηθούν να ξεχωρίσουμε ιδιαίτερα την συμπεριφορά του κάθε αλγόριθμου ξεχωριστά. Στα αριστερά βλέπουμε το traffic sent το οποίο είναι το ίδιο και για τους τρεις αλγόριθμους αφού έχουμε θέσει το ίδιο raw traffic για κάθε αλγόριθμο. Στα δεξιά βλέπουμε το raw traffic το οποίο καταφθάνει στους προορισμούς. Αυτό το οποίο παρατηρούμε είναι ότι ο αλγόριθμος ο οποίος έχει υψηλότερη αποτελεσματικότητα με τα περισσότερα πακέτα να παραδίδονται στον προορισμό τους είναι ο DSR αφού προσεγγίζει κατά πολύ το traffic το οποίο αποστέλλεται από τους παραλήπτες. Οι υπόλοιποι δύο αλγόριθμοι είναι λίγο πιο κάτω από τον DSR ενώ στο τέλος της προσομοίωσης έχουν το ίδιο traffic received κοντά στα 3500 bits/sec(time average) το οποίο είναι περίπου 500bits/sec(time average) λιγότερα από ότι έχει εκείνη την στιγμή ο αλγόριθμος DSR. Αυτό το οποίο παρατηρούμε είναι ότι εδώ υπερτερεί ο αλγόριθμος DSR. Τέλος να αναφέρουμε ότι εναλλακτικά δοκιμάσαμε αντί να σταματούν οι κόμβοι-πηγές να αποστέλλουν στο τέλος της

προσομοίωσης να σταματούν 50 δευτερόλεπτα πριν ώστε να παρατηρήσουμε εάν αλλάζει κάτι στον αριθμό των πακέτων τα οποία φτάνουν στους προορισμούς. Αυτό το οποίο παρατηρήσαμε είναι ότι δεν παρατηρείται κάποια αλλαγή στα αποτελέσματα και αυτά παραμένουν τα ίδια με την γραφική παράσταση που έχουμε δει στην προηγούμενη σελίδα.

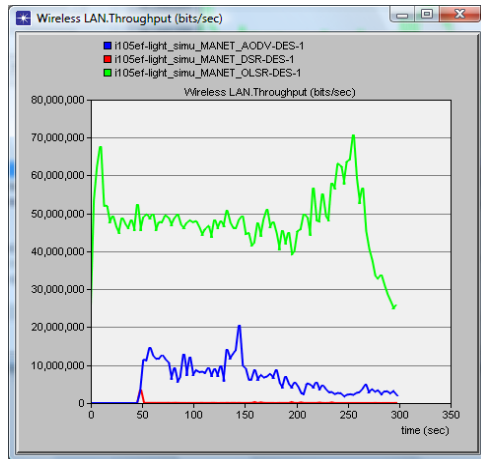
Routing traffic received/Sent (bits/sec) (AODV, DSR, OLSR)



Εδώ μπορούμε να παρατηρήσουμε το routing traffic (overhead στο δίκτυο μας) με την γραφική παράσταση As Is. Στα αριστερά βλέπουμε το routing traffic το οποίο αποστέλλεται από τον κάθε ένα από τους τρεις αλγόριθμους. Στα δεξιά μπορούμε να παρατηρήσουμε το routing traffic που φτάνει στους κόμβους του δικτύου. Κάτι αξιοσημείωτο είναι ότι το traffic received είναι πολύ περισσότερο από το traffic sent. Ο λόγος είναι διότι ένα control πακέτο το οποίο αποστέλλεται με τη μορφή broadcasting στο δίκτυο δημιουργείται μόνο στο κόμβο που το στέλλει. Οι υπόλοιποι κόμβοι οι οποίοι θα το επαναπροωθήσουν απλά το κάνουν duplicate το αποστέλλουν σε όλους τους γείτονες τους. Έτσι ένα control πακέτο το οποίο δημιουργείται και αποστέλλεται με τη μορφή broadcasting φτάνει σε πολλούς παραλήπτες όταν πρόκειται για διαδικασία εξεύρεσης διαδρομής. Αυτός είναι και ο λόγος που έχουμε τόσο μεγάλο traffic received. Από εκεί και πέρα αυτό το οποίο μπορούμε να παρατηρήσουμε είναι ότι ο DSR έχει πολύ χαμηλό overhead, σχεδόν μηδενικό αφού σε αυτό συντείνουν το ότι δεν χρησιμοποιεί hello messages και το ότι χρησιμοποιεί σε πολύ μεγάλο βαθμό τα cached replies. Αυτό σε συνδυασμό με το ότι έχουμε το μεγαλύτερο packet delivery ratio στον DSR δείχνει ένα μεγάλο πλεονέκτημα για τον αλγόριθμο. Ο AODV βρίσκεται αρκετά πιο ψηλά αφού στέλνει συνεχώς μεταξύ 100,000 και 200,000 bits/sec ενώ ο OLSR είναι ο αλγόριθμος ο οποίος παρουσιάζει πολύ μεγάλο overhead με περισσότερα από 400,000

bits/sec να αποστέλλονται ενώ βλέπουμε ότι φτάνουν σε κόμβους κάποιες στιγμές κοντά στα 50,000,000 bits/sec.

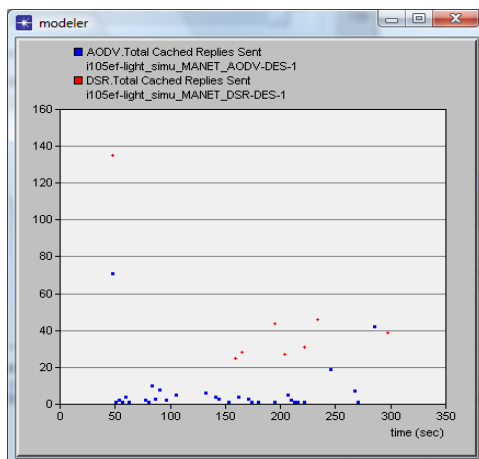
Wireless LAN.Throughput (AODV, DSR, OLSR)



Πιο πάνω μπορούμε να δούμε το throughput γενικά στο δίκτυο μας. Αυτό το οποίο παρατηρούμε είναι ότι ο αλγόριθμος OLSR χρησιμοποιεί πολύ περισσότερο τη γραμμή που του παρέχεται σε αντίθεση με τους δύο άλλους αλγορίθμους όπου ο AODV χρησιμοποιεί αρκετά το link ενώ ο DSR σχεδόν καθόλου. Η υψηλή χρήση της γραμμής από το κάθε κόμβο σίγουρα θα τον περιορίσει σε περίπτωση που χρειαστεί να αυξηθεί η πληροφορία

που θέλουμε να διακινηθεί μέσα στο δίκτυο. Από την άλλη ο DSR με την ελάχιστη χρήση που κάνει έχει την δυνατότητα να μπορεί να αποστέλλει πολύ περισσότερες πληροφορίες λόγω της πολύ χαμηλής χρήσης της γραμμής.

Total Cached Replies Sent (AODV, DSR)

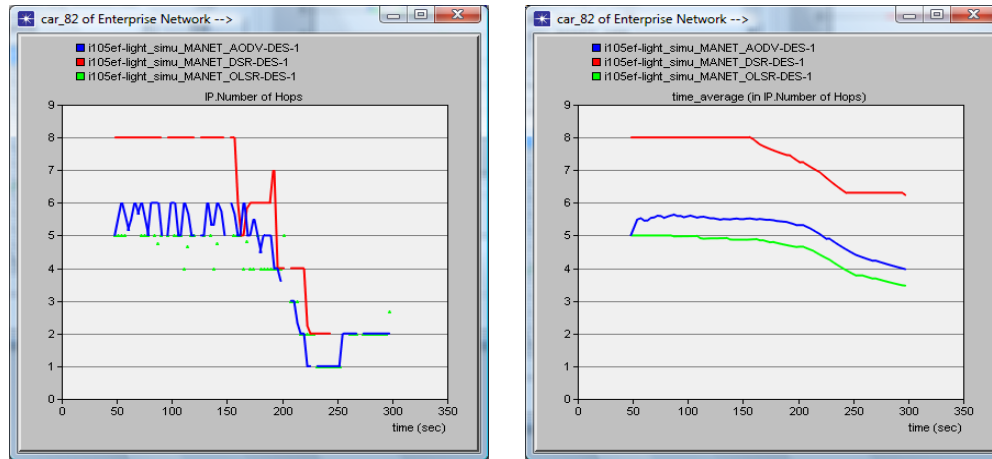


Πιο πάνω μπορούμε να δούμε τα cached replies τα οποία αποστέλλονται από τους δύο αλγόριθμους (AODV, DSR) οι οποίοι χρησιμοποιούν αυτό το χαρακτηριστικό. Βλέπουμε ότι ο αλγόριθμος DSR χρησιμοποιεί ευρέως αυτό το χαρακτηριστικό αφού στην αρχή αποστέλλονται γύρω στις 135 απαντήσεις, ενώ πιο μετά που χρειάζεται να βρει πάλι κάποιες διαδρομές σύμφωνα και με την γραφική του route discovery time ο αλγόριθμος

χρησιμοποιεί και πάλι ευρέως αυτή την δυνατότητα που έχει. Από την άλλη ο AODV χρησιμοποιεί και αυτός την δυνατότητα αυτή αλλά σε μικρότερο βαθμό από ότι ο DSR αφού βλέπουμε τα cached replies να κυμαίνονται μεταξύ 10-20 τις περισσότερες φορές. Όπως είδαμε και στην ανάλυση του αλγορίθμου, ο AODV δεν είναι σχεδιασμένος ώστε να χρησιμοποιεί αυτό το χαρακτηριστικό στο έπακρο.

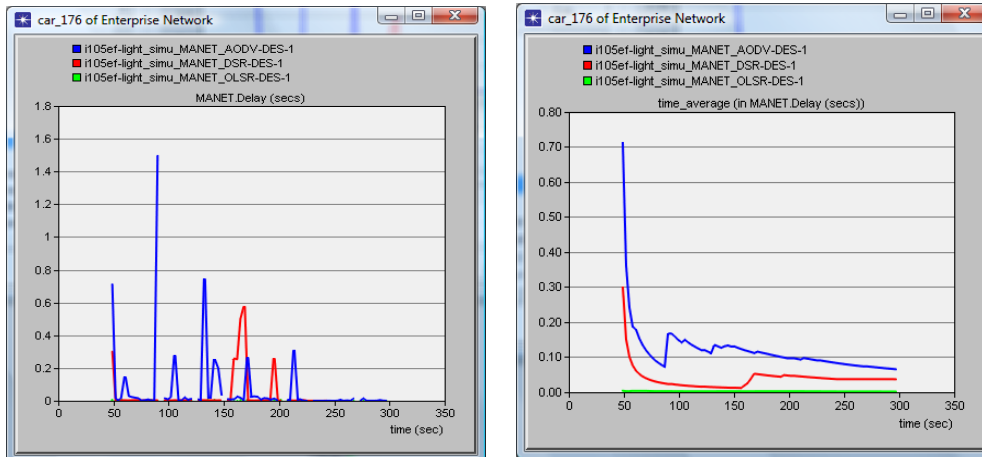
5.1.2 Μελέτη σύνδεσης κόμβων 82-116

IP.Number of Hops



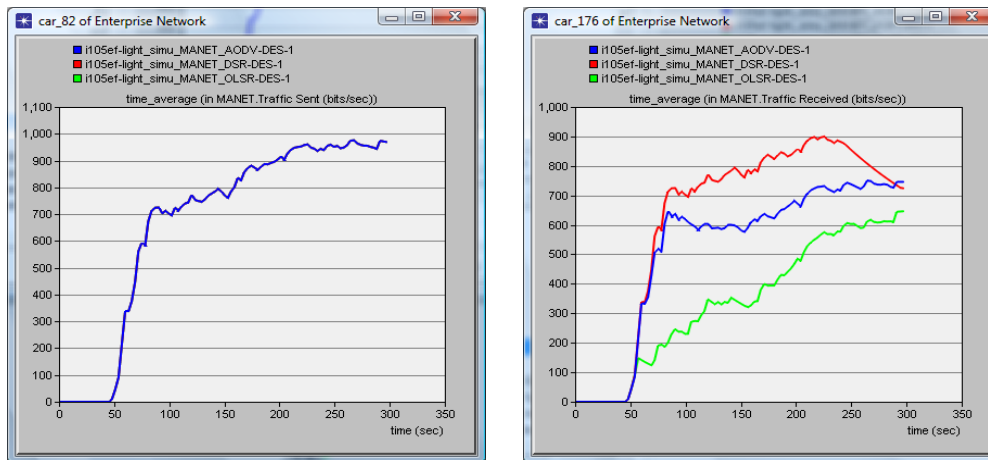
Εδώ μπορούμε να παρατηρήσουμε το αριθμό των κόμβων οι οποίοι παρεμβάλλονται μεταξύ της σύνδεσης η οποία γίνεται μεταξύ των κόμβων 82 και 116. Στα αριστερά έχουμε την γραφική As Is και δεξιά την γραφική παράσταση time average. Αυτό το οποίο μπορούμε να παρατηρήσουμε είναι η ικανότητα του αλγόριθμου OLSR να μπορεί να χρησιμοποιεί διαδρομές με λίγα hops σε σύγκριση με τους υπόλοιπους αλγόριθμους. Πιο συγκεκριμένα ο OLSR κυμαίνεται μεταξύ 4 και 5 hops ενώ ο AODV είναι λίγο πιο πάνω στα 5-6 hops. Ο DSR δείχνει φαίνεται να δημιουργεί μία διαδρομή η οποία περιλαμβάνει περισσότερα hops, γύρω στα 7-8 hops η οποία προσθέτει περισσότερο delay στην παράδοση των πακέτων. Αυτό το οποίο παρατηρούμε εδώ είναι ότι έχουμε περίπου τα ίδια αποτελέσματα με την global γραφική παράσταση όπου ο OLSR έχει τις πιο σύντομες διαδρομές σε αντίθεση με τους άλλους δύο reactive αλγόριθμους δρομολόγησης.

MANET. Delay (AODV, DSR, OLSR)



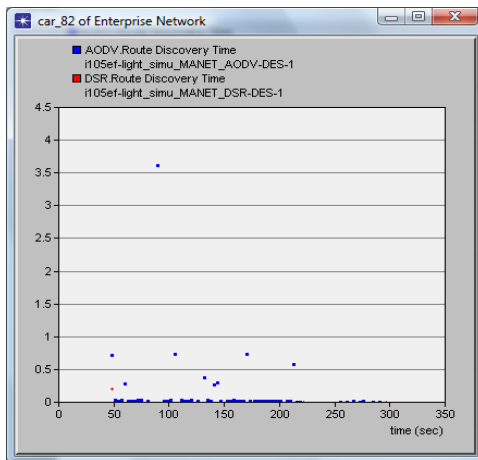
Εδώ μπορούμε να δούμε το end-to-end delay (As Is και Average) για τα πακέτα που αποστέλλονται μεταξύ των δύο κόμβων. Όπως μπορούμε να δούμε ο αλγόριθμος OLSR έχει ελάχιστο delay, έως μηδενικό, το οποίο μπορεί να οφείλεται στην δυνατότητα του αλγόριθμου να δημιουργεί διαδρομές κοντά στις βέλτιστες οι οποίες προκύπτουν από το ότι δεν χρειάζεται να βρει την διαδρομή μέσω της οποίας θα στείλει τα πακέτα προς το προορισμό αφού γνωρίζει την διαδρομή από πριν. Όσο αφορά τους άλλους αλγορίθμους, ο αλγόριθμος DSR έρχεται δεύτερος αφού βλέπουμε σε μέσο όρο να έχει πολύ μικρό delay, μικρότερο από 0.05 δευτερόλεπτα. Εκείνο το οποίο μας δημιουργεί κάποια ανησυχία στον αλγόριθμο αυτό είναι ότι την στιγμή που χρειάζεται ο αλγόριθμος να ανακαλύψει διαδρομή, δηλαδή στο 50 δευτερόλεπτο και μετά προς το 200 δευτερόλεπτο παρουσιάζεται αρκετά μεγάλο delay γύρω στο μισό δευτερόλεπτο. Αυτό δείχνει ότι υπάρχει ένα μειονέκτημα στον αλγόριθμο κυρίως κατά την διαδικασία εξεύρεσης διαδρομής. Ο αλγόριθμος AODV είναι χειρότερος από τον DSR λόγω και του ότι όπως έχουμε δει και πιο πάνω στις γραφικές, σπάζουν συνεχώς τα links και αναγκάζεται ο αλγόριθμος να βρίσκει καινούργιες διαδρομές προς τους προορισμούς. Αυτό προσθέτει συνεχείς και απότομες αυξομειώσεις στην γραφική παράσταση του αλγόριθμου αυτού οι οποίες φέρνουν το delay του έστω και σε κάποιες χρονικές στιγμές σε πολύ ψηλά επίπεδα.

Manet. Traffic received/Sent (AODV, DSR, OLSR)



Εδώ βλέπουμε το traffic που στέλλεται και λαμβάνεται από τους δύο αυτούς κόμβους σε time average γραφική παράσταση. Όπως μπορούμε να παρατηρήσουμε στέλλεται το ίδιο traffic και για τους τρεις αλγόριθμους όπως φαίνεται στην γραφική παράσταση του traffic sent. Όσο αφορά τώρα το traffic το οποίο λαμβάνεται από το κόμβο 116 βλέπουμε ότι ο DSR υπερνικά τους υπόλοιπους δύο αλγόριθμους με το AODV έρχεται δεύτερος και το OLSR να έρχεται τρίτος. Ο OLSR μένει αρκετά πίσω ενώ ο DSR προς το τέλος χάνει όλα τα πακέτα και καταλήγει στο τέλος να βρίσκεται πιο χαμηλά από τον AODV στο time average. Παρόλα αυτά μπορούμε να πούμε ότι ο DSR γενικά έχει ψηλότερο traffic received από τους υπόλοιπους αλγόριθμους.

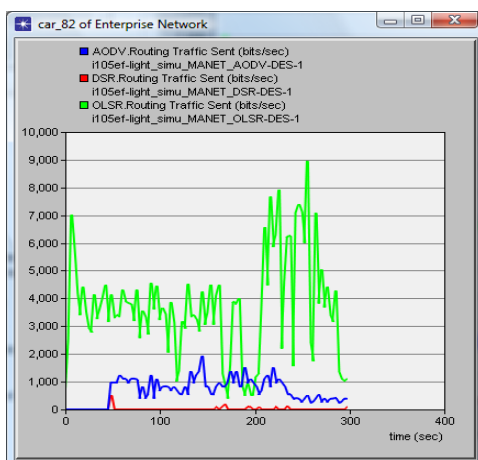
Route Discovery Time (AODV, DSR)



Εδώ μπορούμε να παρατηρήσουμε το χρόνο που χρειάζεται ο συγκεκριμένος κόμβος ώστε να δημιουργήσει μία διαδρομή προς το συγκεκριμένο προορισμό. Βλέπουμε και εδώ όπως και στην global γραφική παράσταση ότι ο DSR γενικά χρειάζεται πολύ πιο λίγες φορές ώστε να ανακαλύψει την διαδρομή προς τον κόμβο 116 ενώ είναι και πιο λίγος ο χρόνος που απαιτείται για την διαδικασία αυτή. Από την άλλη ο AODV, με τις default ρυθμίσεις του,

χρειάζεται συνεχώς να ανακαλύπτει καινούργιες διαδρομές προς τον προορισμό οι οποίες ως επί το πλείστον γίνονται σε ελάχιστο χρόνο. Αυτό δεν παύει όμως να προσθέτει επιπλέον delay στην μετάδοση κάποιων πακέτων.

Routing traffic Sent (bits/sec) (AODV, DSR, OLSR)



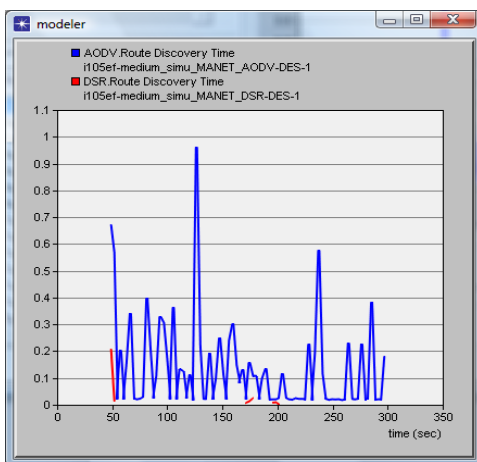
Εδώ μπορούμε να συγκρίνουμε το routing traffic που αποστέλλει ο κόμβος 82 ο οποίος αποτελεί την πηγή της σύνδεσης αυτής. Παρατηρούμε ότι και εδώ ο OLSR χρειάζεται πολύ περισσότερο routing traffic για να διατηρήσει ενημερωμένο τον κόμβο αυτό καθώς και τους γειτονικούς του. Ο AODV κινείται σε αρκετά πιο χαμηλά επίπεδα, κοντά στα 1000 bits/sec, το οποίο είναι αρκετά πιο χαμηλό από αυτό που μας δίνει ο OLSR.

5.2 Αποτελέσματα σεναρίου Medium

Να αναφέρουμε ότι τα αποτελέσματα τα οποία λάβαμε χρησιμοποιώντας κάθε φορά διαφορετικό traffic όπως προαναφέραμε δεν παρουσίασαν ιδιαίτερες διαφορές μεταξύ τους, γι αυτό και εδώ θα αναφερθούμε μόνο στα αποτελέσματα τα οποία πήραμε όταν το traffic μας δημιουργείτο από 5 κόμβους μόνο. Εάν κάποια από τις υπόλοιπες γραφικές παρουσιάσει κάποιο επιπλέον ενδιαφέρον θα αναφερθεί. Από εκεί και πέρα οι υπόλοιπες γραφικές παραστάσεις θα παρουσιαστούν στο Παράρτημα β.

5.2.1 Global Αποτελέσματα

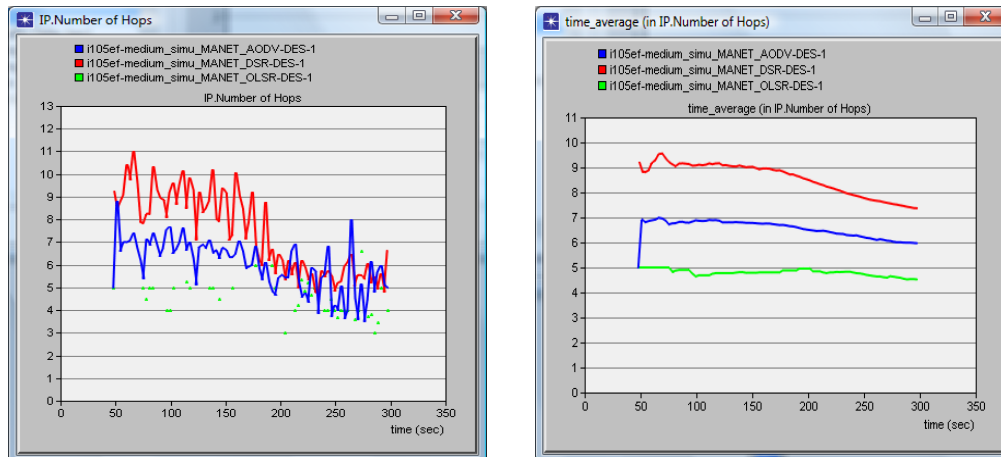
Route Discovery Time (AODV, DSR)



Εδώ μπορούμε να δούμε το χρόνο που χρειάζεται για να ανακαλυφθεί μία διαδρομή προς κάποιο προορισμό για το σενάριο Medium. Αυτό περιλαμβάνει και την αποστολή του Reply στον αποστολέα. Όπως βλέπουμε αυτή η μετρική μας παρέχεται μόνο για τους αλγόριθμους AODV, DSR. Αυτό που παρατηρούμε είναι ότι ο DSR χρειάζεται πολύ λιγότερο χρόνο για να βρει μία διαδρομή ενώ επίσης αυτό δεν γίνεται συνεχώς όπως φαίνεται να συμβαίνει με τον

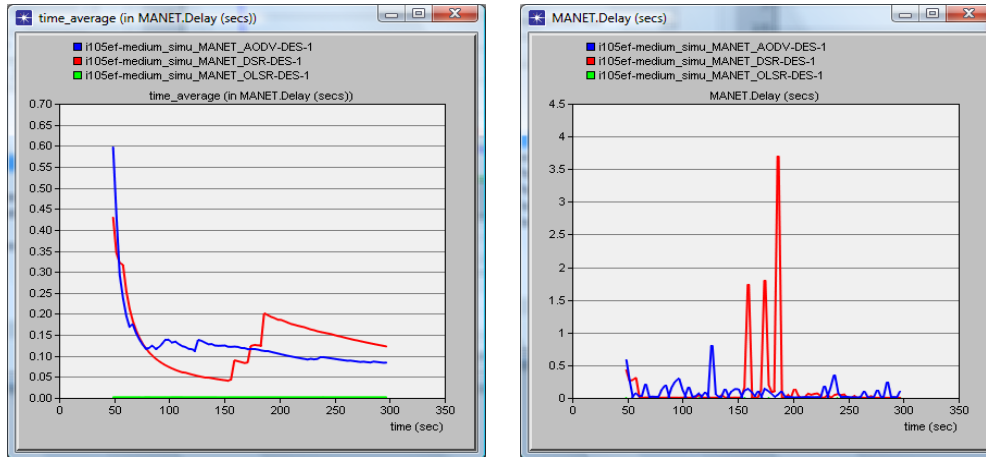
αλγόριθμο AODV. Αυτό το οποίο φαίνεται να συμβαίνει με τον αλγόριθμο αυτό είναι ότι σπάζουν συνεχώς τα links όπως και στο σενάριο Light και χρειάζεται συνεχώς να ανακαλύπτεται καινούργια διαδρομή προς τους προορισμούς αυτούς. Ο DSR χρειάζεται κάποιο χρόνο αρχικά ώστε να βρει όλες τις διαδρομές προς όλους τους προορισμούς και μετά όπως φαίνεται από την γραφική παράσταση χρειάζεται ακόμα δύο φορές να βρει διαδρομές, μεταξύ 150-200 δευτερολέπτου οι οποίες σπάζουν λόγω της κινητικότητας του σεναρίου. Να πούμε εδώ ότι ο αλγόριθμος OLSR δεν περιλαμβάνεται στην γραφική παράσταση διότι ως proactive αλγόριθμος δεν χρειάζεται χρόνο να ανακαλύψει μία διαδρομή αφού διατηρεί ανά πάσα χρονική στιγμή πληροφορίες για κάθε κόμβο και τη διαδρομή προς αυτόν.

IP.number of Hops (AODV, DSR, OLSR)



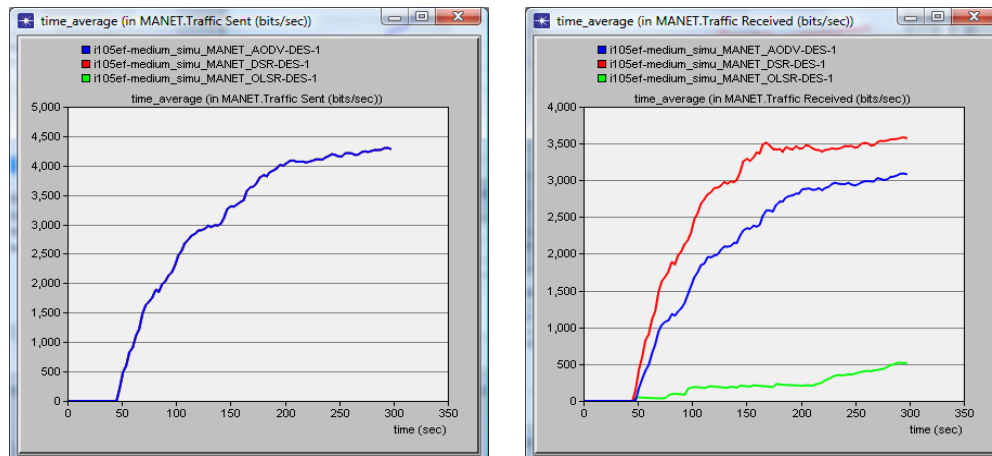
Στις πιο πάνω γραφικές παραστάσεις μπορούμε να δούμε το αριθμό των hops τα οποία μεσολαβούν για την σύνδεση μεταξύ των κόμβων του δικτύου σε global επίπεδο. Αυτό το οποίο παρατηρούμε είναι μία παρόμοια κατάσταση με το σενάριο Light όπου ο OLSR δημιουργεί τις πιο σύντομες διαδρομές, ο AODV έρχεται δεύτερος και ο DSR τρίτος. Όπως φαίνεται και στην γραφική παράσταση του time average ο OLSR κατά μέσο όρο δημιουργεί διαδρομές με περίπου 5 hops, ο AODV δημιουργεί διαδρομές με αριθμό hops μεταξύ 6 και 7 ενώ ο DSR δημιουργεί διαδρομές γύρω στα 9 hops οι οποίες βελτιώνονται όσο περνά ο χρόνος και στο τέλος κυμαίνονται γύρω στα 8 hops. Το ερώτημα το οποίο τίθεται είναι κατά πόσο οι πιο σύντομες διαδρομές τις οποίες δημιουργεί ο OLSR είναι ικανές να δημιουργήσουν συνθήκες οι οποίες να μπορούν να εξασφαλίσουν έγκυρη και έγκαιρη μεταφορά των πληροφοριών μεταξύ του αποστολέα και του παραλήπτη. Πάντως στο σενάριο Light ο αριθμός των hops βοήθησε μόνο στο delay αλλά όχι και στη παράδοση μεγάλου ποσοστού των πακέτων δεδομένων που στάλθηκαν. Αντίθετα ήταν ο DSR ο οποίος πέτυχε μεγαλύτερη παράδοση πακέτων στο προηγούμενο σενάριο.

MANET. Delay (AODV, DSR, OLSR)



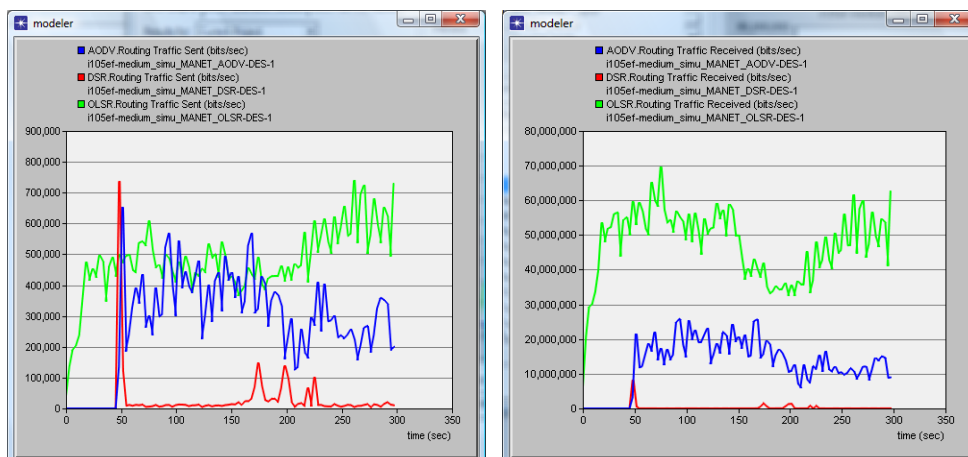
Πιο πάνω βλέπουμε τις γραφικές παραστάσεις του end-to-end delay για τον κάθε αλγόριθμο. Η γραφική παράσταση στα αριστερά είναι η As Is ενώ στα δεξιά έχουμε την παράσταση του time average. Το end-to-end delay αφορά στον χρόνο από την δημιουργία του MANET πακέτου στον αποστολέα μέχρι και την καταστροφή του στον παραλήπτη. Όπως μπορούμε να δούμε και πιο πάνω ο OLSR έχει delay πάρα πολύ κοντά στο μηδέν. Αυτό οφείλεται στην δυνατότητα του αλγορίθμου να διατηρεί ανά πάσα στιγμή διαδρομές προς όλους τους προορισμούς- είναι proactive άλλωστε- με αποτέλεσμα να είναι όσο το δυνατό πιο μικρός ο χρόνος αυτός. Από την άλλη όσο αφορά τους reactive αλγόριθμους βλέπουμε ότι ο DSR είναι λίγο πιο πάνω από τον OLSR ενώ ο AODV είναι ακόμα σε πιο ψηλό σημείο, κάτι το οποίο αλλάζει μετά το 3^ο λεπτό όπου ο DSR παρουσιάζει μεγάλο delay, λίγο πάνω από τα 3.5 δευτερόλεπτα στιγμιαία, αυξάνοντας έτσι και την μέση καθυστέρηση η οποία παρατηρείται και κάνοντας την να είναι πιο μεγάλη ακόμα και από τον αλγόριθμο AODV. Η μεγάλη αυτή καθυστέρηση στο αλγόριθμο DSR φαίνεται να οφείλεται στο μεγάλο χάσιμο πακέτων η οποία φαίνεται να συμβαίνει στο συγκεκριμένο χρονικό διάστημα αφού η γραφική παράσταση του WirelessLan. Data Dropped (Retry Threshold Exceeded) που υποδηλώνει τον αριθμό των πακέτων που χάνονται από τα retransmissions στο MAC layer δείχνει το χάσιμο των πακέτων αυτών. Αυτό δείχνει ότι οι κόμβοι προσπαθούν να στείλουν ανεπιτυχώς τα data πακέτα στο συγκεκριμένο σημείο και προσπαθούν να τα κάνουν retransmit. Γι' αυτό και παρατηρείται αυτή η μεγάλη καθυστέρηση δείχνοντας εδώ ένα σημαντικό μειονέκτημα στον αλγόριθμο DSR όσο αφορά κυρίως το MAC layer.

MANET. Traffic Received/Sent (bits/second) (AODV, DSR, OLSR)



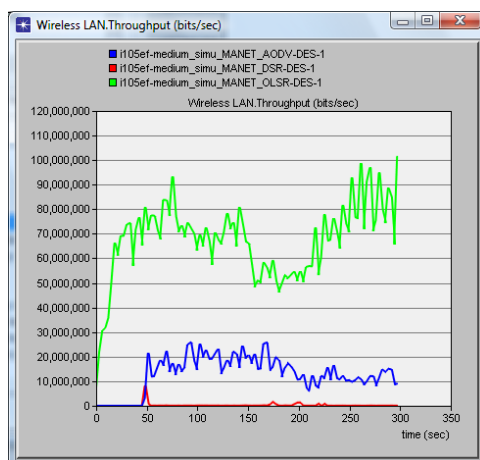
Στις δύο γραφικές παραστάσεις πιο πάνω βλέπουμε το traffic που στέλλεται και λαμβάνεται για τους τρεις αλγόριθμους. Οι γραφικές παραστάσεις που χρησιμοποιούμε εδώ αφορούν στο time average. Στα αριστερά βλέπουμε το traffic sent το οποίο είναι το ίδιο και για τους τρεις αλγόριθμους αφού έχουμε θέσει το ίδιο raw traffic για κάθε αλγόριθμο. Στα δεξιά βλέπουμε το raw traffic το οποίο καταφθάνει στους προορισμούς. Αυτό το οποίο παρατηρούμε είναι ότι ο αλγόριθμος ο οποίος έχει υψηλότερη αποτελεσματικότητα είναι ο DSR αφού προσεγγίζει κατά πολύ το traffic το οποίο αποστέλλεται από τους παραλήπτες. Ο AODV είναι λίγο πιο κάτω από τον DSR και στο τέλος της προσομοίωσης είναι κοντά στα 3000 bits/sec(time average). Ο αλγόριθμος OLSR παρουσιάζει απογοητευτικά αποτελέσματα αφού κυμαίνεται σε πολύ χαμηλά επίπεδα αφού σε καμία στιγμή δεν ξεπερνά τα 500bits/sec(time average). Αυτό το οποίο φαίνεται να έχει επηρεάσει τον OLSR είναι ίσως η αύξηση των κόμβων στο δίκτυο ή το πιο πιθανό οι αλλαγές που κάναμε στις ρυθμίσεις του αλγορίθμου λόγω του περιορισμού στην μνήμη των μηχανών που υπήρχαν στο εργαστήριο των δικτύων.

Routing traffic Received/Sent (bits/sec) (AODV, DSR, OLSR)



Εδώ μπορούμε να παρατηρήσουμε το routing traffic (overhead στο δίκτυο μας) με την γραφική παράσταση As Is. Στα αριστερά βλέπουμε το routing traffic το οποίο αποστέλλεται από τον κάθε ένα από τους τρεις αλγόριθμους. Στα δεξιά μπορούμε να παρατηρήσουμε το routing traffic που φτάνει στους κόμβους του δικτύου. Κάτι αξιοσημείωτο είναι ότι το traffic received είναι πολύ περισσότερο από το traffic sent. Ο λόγος είναι διότι ένα control πακέτο το οποίο αποστέλλεται με τη μορφή broadcasting στο δίκτυο δημιουργείται μόνο στο κόμβο που το στέλλει. Οι υπόλοιποι κόμβοι οι οποίοι θα το επαναπροωθήσουν απλά το κάνουν duplicate το αποστέλλουν σε όλους τους γείτονες τους. Έτσι ένα control πακέτο το οποίο δημιουργείται και αποστέλλεται με τη μορφή broadcasting φτάνει σε πολλούς παραλήπτες. Αυτός είναι και ο λόγος που έχουμε τόσο μεγάλο traffic received. Από εκεί και πέρα αυτό το οποίο μπορούμε να παρατηρήσουμε είναι ότι ο DSR έχει πολύ χαμηλό overhead, σχεδόν μηδενικό αφού σε αυτό συντείνουν το ότι δεν χρησιμοποιεί hello messages και το ότι χρησιμοποιεί cached replies λόγω της δυνατότητας που έχει να διατηρεί cache με τις προδιαγραφές που προαναφέραμε κατά την ανάλυση των αλγορίθμων. Αυτό σε συνδυασμό με το ότι έχουμε το μεγαλύτερο packet delivery ratio στον DSR δείχνει ένα μεγάλο πλεονέκτημα για τον αλγόριθμο. Ο AODV βρίσκεται αρκετά πιο ψηλά αφού στέλνει συνεχώς μεταξύ 200,000 και 500,000 bits/sec ενώ ο OLSR είναι ο αλγόριθμος ο οποίος παρουσιάζει πολύ μεγάλο overhead με περισσότερα από 400,000 bits/sec να αποστέλλονται ενώ βλέπουμε ότι όσο αφορά στα bits που φτάνουν στο προορισμό βλέπουμε ότι σε κάποιες στιγμές είναι κοντά 50,000,000 bits/sec ή ακόμη και περισσότερο.

Wireless LAN.Throughput(AODV,DSR,OLSR)

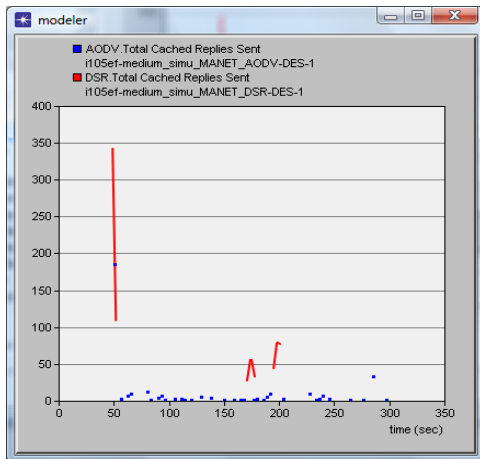


Πιο πάνω μπορούμε να δούμε το throughput γενικά στο δίκτυο μας. Αυτό το οποίο παρατηρούμε είναι ότι ο αλγόριθμος OLSR χρησιμοποιεί πολύ περισσότερο τη γραμμή που του παρέχεται σε αντίθεση με τους δύο άλλους αλγορίθμους όπου ο AODV χρησιμοποιεί αρκετά το link ενώ ο DSR σχεδόν καθόλου. Η υψηλή χρήση της γραμμής από το κάθε κόμβο σίγουρα θα τον περιορίσει σε περίπτωση που χρειαστεί να αυξηθεί η πληροφορία

που θέλουμε να διακινηθεί μέσα στο δίκτυο. Από την άλλη ο DSR με την ελάχιστη χρήση που κάνει έχει την δυνατότητα να μπορεί να αποστέλλει πολύ περισσότερες πληροφορίες

λόγω της πολύ χαμηλής χρήσης της γραμμής. Αυτό έχει φανεί και πιο πάνω όπου βλέπουμε να φτάνουν πολύ λίγες πληροφορίες στο προορισμό σε σχέση με αυτές που αποστέλλονται.

Total Cached Replies Sent (AODV, DSR)

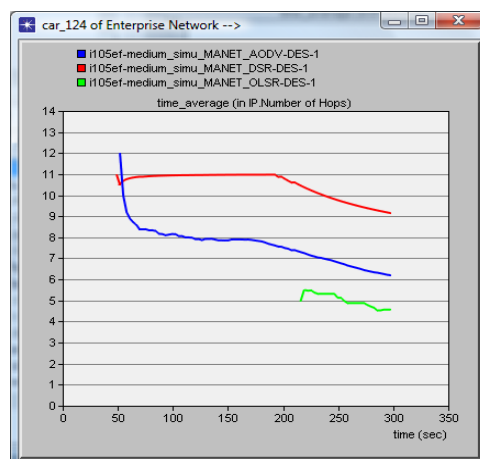
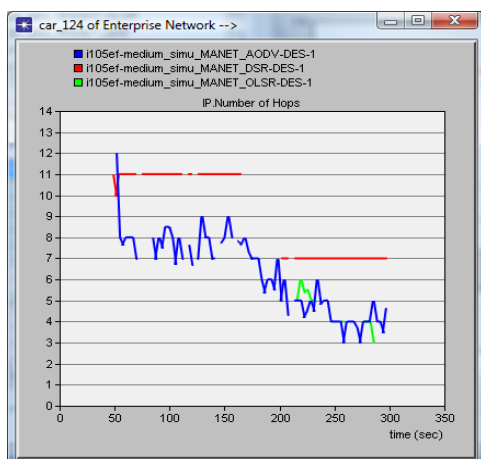


Δίπλα μπορούμε να δούμε τα cached replies τα οποία αποστέλλονται από τους δύο αλγόριθμους (AODV,DSR) οι οποίοι χρησιμοποιούν αυτό το χαρακτηριστικό. Βλέπουμε ότι ο αλγόριθμος DSR χρησιμοποιεί ευρέως αυτό το χαρακτηριστικό αφού στην αρχή αποστέλλονται γύρω στις λίγο περισσότερο από 300 απαντήσεις, ενώ πιο μετά που χρειάζεται να βρει πάλι κάποιες διαδρομές ο αλγόριθμος χρησιμοποιεί και πάλι την αυτή τη

δυνατότητα που έχει φτάνοντας σε περίπου 50 cached replies την πρώτη φορά και 80 περίπου την δεύτερη φορά. Από την άλλη ο AODV χρησιμοποιεί και αυτός την δυνατότητα αυτή αλλά σε μικρότερο βαθμό από ότι τον DSR αφού βλέπουμε τα cached replies να κυμαίνονται μεταξύ 0-20 τις περισσότερες φορές. Βλέπουμε ότι ο αλγόριθμος DSR έχει ένα μεγάλο πλεονέκτημα λόγω της χρήσης αυτής των cached replies οι οποίες μπορούν να μειώσουν το κατά πολύ το overhead στο δίκτυο και να του επιτρέψουν να μειώσει το delay όσο αφορά τον χρόνο κυρίως για την εξεύρεση διαδρομής..

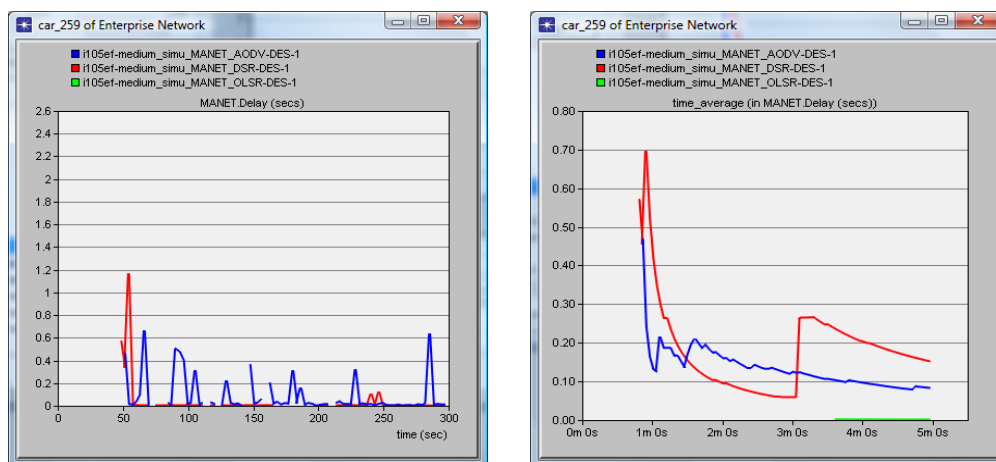
5.2.2 Μελέτη σύνδεσης κόμβων 124-259

IP.number of Hops (AODV, DSR, OLSR)



Εδώ μπορούμε να παρατηρήσουμε το αριθμό των κόμβων οι οποίοι παρεμβάλλονται μεταξύ της σύνδεσης η οποία γίνεται μεταξύ των κόμβων 124 και 259. Στα αριστερά έχουμε την γραφική As Is και δεξιά την γραφική παράσταση time average. Αυτό το οποίο μπορούμε να παρατηρήσουμε είναι η ικανότητα του αλγόριθμου OLSR να μπορεί να χρησιμοποιεί διαδρομές με λίγα hops σε σύγκριση με τους υπόλοιπους αλγόριθμους. Πιο συγκεκριμένα ο OLSR κυμαίνεται μεταξύ 4 και 5 hops ενώ ο AODV είναι λίγο πιο πάνω στα 7 hops. Ο DSR δείχνει να δημιουργεί μία διαδρομή η οποία περιλαμβάνει περισσότερα hops, γύρω στα 10-11 hops η οποία προσθέτει περισσότερο delay στην παράδοση των πακέτων. Φυσικά αργότερα κατά την προσομοίωση τα hops μειώνονται από 11 σε 7 όπως φαίνεται στην γραφική παράσταση στα αριστερά. Ο αλγόριθμος AODV δεν διατηρεί συνεχώς και σταθερά τον ίδιο αριθμό κόμβων αφού όπως βλέπουμε και πάλι από την γραφική παράσταση στα αριστερά συνεχώς αλλάζει ο αριθμός των hops στην συγκεκριμένη διαδρομή. Γενικά όπως συμβαίνει και με τις global γραφικές παραστάσεις ο OLSR φαίνεται να δημιουργεί τις πιο σύντομες διαδρομές παρόλο που όπως φαίνεται για την συγκεκριμένη σύνδεση αργούν να δημιουργηθούν οι διαδρομές. Αυτό φαίνεται να οφείλεται στην αλλαγή την οποία έχουμε κάνει στις παραμέτρους του αλγορίθμου που από ότι φαίνεται οδηγεί τον αλγόριθμο στο να έχει μειωμένη απόδοση και να δημιουργεί διαδρομές μόνο προς το τέλος της προσομοίωσης.

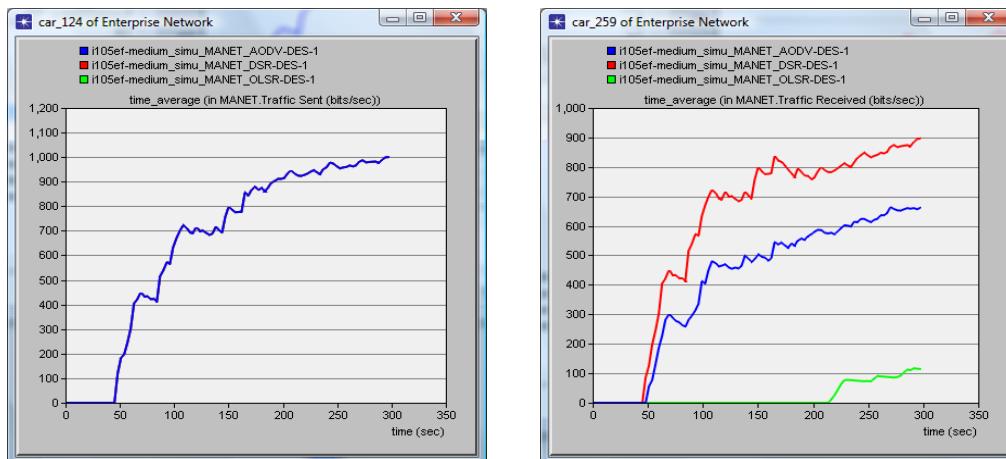
MANET. Delay (AODV, DSR, OLSR)



Όπως φαίνεται και από την γραφική παράσταση και οι τρεις αλγόριθμοι ανταποκρίνονται πολύ καλά όσο αφορά το delay αφού παρουσιάζουν πολύ μικρές τιμές. Εκεί που φαίνεται να δημιουργείται λίγο delay είναι στην αρχή για τους αλγόριθμους AODV και DSR όπου χρειάζεται να βρουν την διαδρομή προς το προορισμό, δηλαδή προς το κόμβο 259. Ο OLSR φαίνεται ότι καταφέρνει να στείλει δεδομένα μόνο προς το τέλος της προσομοίωσης και εκεί

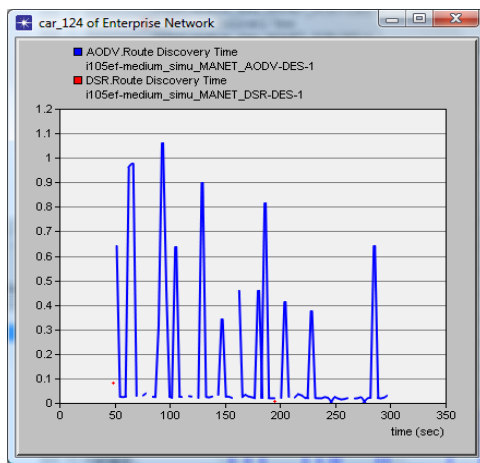
παρατηρούμε αυτό που είδαμε και πιο πριν, ότι δηλαδή έχει πάρα πολύ μικρό delay, σχεδόν μηδενικό. Ο AODV έχει συνεχώς μικρές αυξομειώσεις διότι φαίνεται πως συμβαίνει αυτό που είδαμε στις global γραφικές, όπου σπάζουν συνεχώς τα links και συνεπώς χρειάζεται κάθε φορά να βρίσκεται η διαδρομή προς το προορισμό- τον κόμβο 259. Τα μεγαλύτερα delays του αλγορίθμου αυτού είναι γύρω στα 0.6 δευτερόλεπτα. Για τον DSR αυτό που είναι αξιοσημείωτο είναι ότι στην αρχή έχουμε delay που ξεπερνά το 1 δευτερόλεπτο το οποίο οφείλεται στο γεγονός του ότι χρειάζεται να δημιουργηθεί η διαδρομή προς το κόμβο 259. Από εκεί και πέρα το delay παρουσιάζει είναι πάρα πολύ μικρό, μέσα δηλαδή στα πλαίσια που απαιτούνται για την ομαλή λειτουργία του αλγορίθμου και την σωστή επικοινωνία των κόμβων μεταξύ τους.

MANET. Traffic Sent/Received (AODV, DSR, OLSR)



Εδώ βλέπουμε το traffic που στέλλεται και λαμβάνεται από τους δύο αυτούς κόμβους σε time average γραφικές παραστάσεις. Όπως μπορούμε να παρατηρήσουμε στέλλεται το ίδιο traffic και για τους τρεις αλγόριθμους όπως φαίνεται στην γραφική παράσταση του traffic sent. Όσο αφορά τώρα το traffic το οποίο λαμβάνεται από το κόμβο αυτό βλέπουμε ότι ο DSR υπερνικά τους υπόλοιπους δύο αλγόριθμους με το AODV έρχεται δεύτερος και το OLSR να έρχεται τρίτος. Ο DSR καταφέρνει να παρουσιάζει πολύ καλά αποτελέσματα αφού σχεδόν καταφέρνει να προσεγγίσει τον αριθμό των πακέτων τα οποία στέλλονται από τον κόμβο 259. Στον OLSR ο κόμβος 259 αρχίζει να λαμβάνει πακέτα μετά από 3 λεπτά και 30 δευτερόλεπτα μέχρι το τέλος της προσομοίωσης αλλά πάλι κινείται σε πολύ χαμηλά επίπεδα η αποστολή των πακέτων.

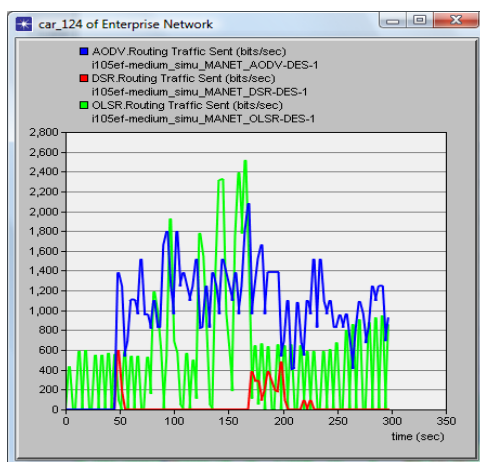
Route Discovery Time (AODV, DSR)



Εδώ μπορούμε να παρατηρήσουμε το χρόνο που χρειάζεται ο κόμβος 124 ώστε να δημιουργήσει την διαδρομή προς τον κόμβο 259. Αυτό το οποίο παρατηρούμε είναι ότι ο αλγόριθμος AODV χρειάζεται συνεχώς να βρίσκει καινούργιες διαδρομές ώστε να μπορεί να αποστέλλει τα πακέτα προς το προορισμό που θέλει. Επίσης φαίνεται ότι του παίρνει και αρκετό χρόνο να ανταποκριθεί ικανοποιητικά αφού κάποιες στιγμές χρειάζεται χρόνο μέχρι και 1 δευτερόλεπτο

ώστε να βρει καινούργια διαδρομή. Από την άλλη ο DSR χρειάζεται πολύ λίγο χρόνο ώστε να βρει την συγκεκριμένη διαδρομή ενώ δεν χρειάζεται να το κάνει αυτό πολλές φορές.

Routing traffic Sent (bits/sec) (AODV, DSR, OLSR)



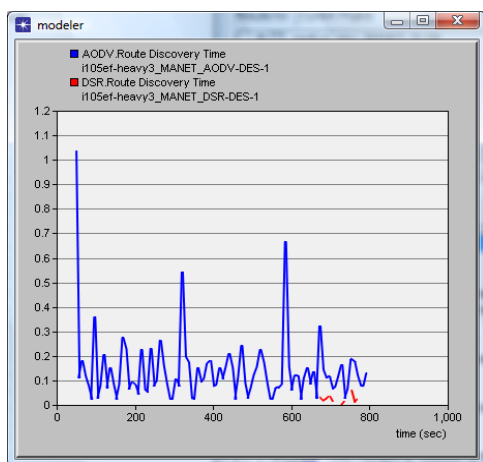
Όσο αφορά το routing traffic που αποστέλλεται από τον κόμβο 124, παρατηρούμε εδώ ότι ο DSR χρειάζεται πολύ λίγο routing traffic να στείλει ώστε να αντεπεξέλθει στην τοπολογία του δικτύου. Όσο αφορά τους άλλους δύο αλγόριθμους, βλέπουμε ότι χρειάζονται πολύ περισσότερο traffic να αποστείλουν το οποί μάλιστα παρουσιάζει και πολλά σκαμπανεβάσματα.

5.3 Αποτελέσματα σεναρίου Heavy

Να αναφέρουμε ότι τα αποτελέσματα τα οποία λάβαμε χρησιμοποιώντας κάθε φορά διαφορετικό traffic όπως προαναφέραμε δεν παρουσίασαν ιδιαίτερες διαφορές μεταξύ τους, γι αυτό και εδώ θα αναφερθούμε μόνο στα αποτελέσματα τα οποία πήραμε όταν το traffic μας δημιουργείτο από 5 κόμβους μόνο. Εάν κάποια από τις υπόλοιπες γραφικές παρουσιάσει κάποιο επιπλέον ενδιαφέρον θα αναφερθεί. Από εκεί και πέρα οι υπόλοιπες γραφικές παραστάσεις θα παρουσιαστούν στο Παράρτημα β.

5.3.1 Global Αποτελέσματα

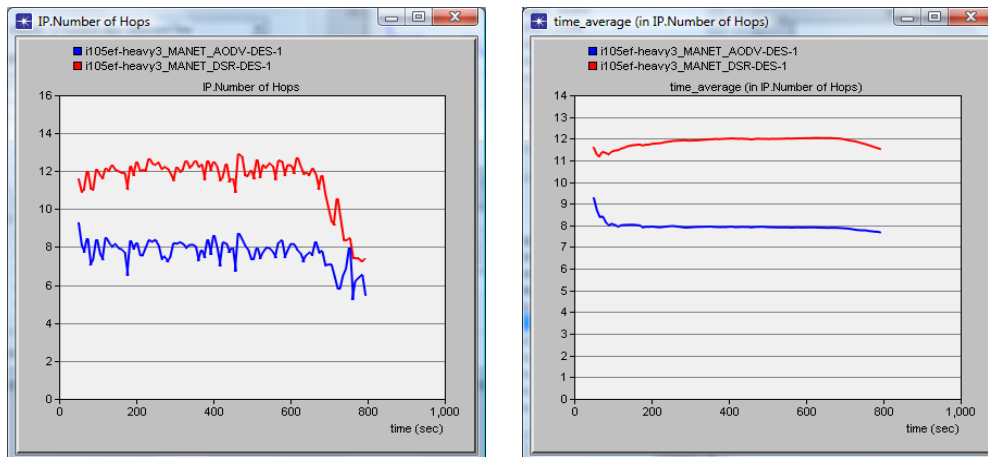
Route Discovery Time (AODV, DSR)



Εδώ μπορούμε να δούμε το χρόνο που χρειάζεται για να ανακαλυφθεί μία διαδρομή προς κάποιο προορισμό για το σενάριο Heavy. Αυτό περιλαμβάνει και την αποστολή του Reply στον αποστολέα. Όπως βλέπουμε αυτή η μετρική μας παρέχεται μόνο για τους αλγόριθμους AODV, DSR. Αυτό που παρατηρούμε είναι ότι ο DSR χρειάζεται πολύ λιγότερο χρόνο να βρει μία διαδρομή ενώ επίσης αυτό δεν γίνεται συνεχώς όπως φαίνεται να συμβαίνει με τον αλγόριθμο

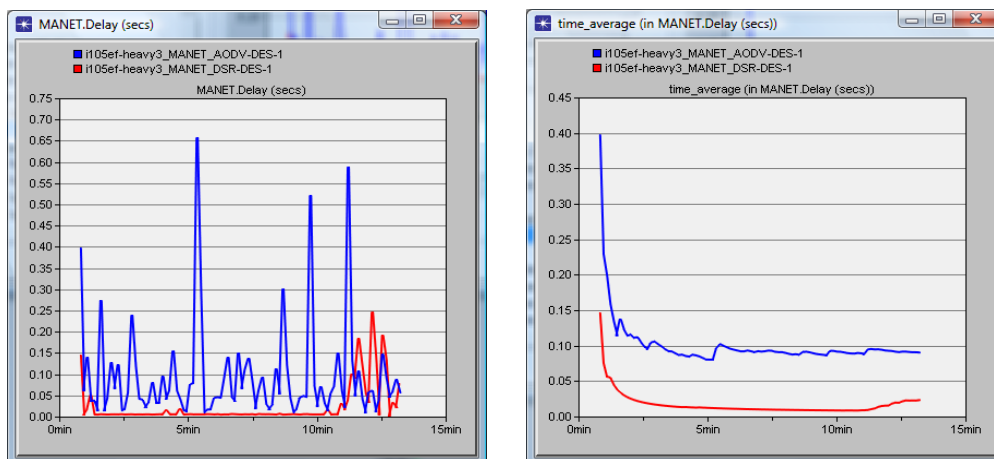
AODV. Αυτό το οποίο φαίνεται να συμβαίνει με τον αλγόριθμο αυτό είναι ότι σπάζουν συνεχώς τα links και χρειάζεται συνεχώς να ανακαλύπτεται καινούργια διαδρομή προς τους προορισμούς αυτούς. Αυτό παρατηρήσαμε και στα δύο προηγούμενα σενάρια, Light και Medium. Ο DSR χρειάζεται πολύ πιο λίγο χρόνο ώστε να μπορέσει να ανακαλύψει κάποια διαδρομή. Να πούμε εδώ ότι ο αλγόριθμος OLSR δεν περιλαμβάνεται στην γραφική παράσταση διότι ως proactive αλγόριθμος δεν χρειάζεται χρόνο να ανακαλύψει μία διαδρομή αφού διατηρεί ανά πάσα χρονική στιγμή πληροφορίες για κάθε κόμβο και τη διαδρομή προς αυτόν.

IP.number of Hops (AODV, DSR, OLSR)



Στις πιο πάνω γραφικές παραστάσεις μπορούμε να δούμε το αριθμό των hops τα οποία μεσολαβούν για την σύνδεση μεταξύ των κόμβων του δικτύου σε global επίπεδο. Αυτό το οποίο παρατηρούμε είναι ότι ο OLSR δεν εμφανίζεται καθόλου στην γραφική παράσταση πράγμα που σημαίνει ότι δεν καταφέρνει να δημιουργήσει καθόλου διαδρομές και να ανταλλάξει καθόλου, ο AODV δημιουργεί διαδρομές με αριθμό hops γύρω στα 8 ενώ ο DSR δημιουργεί διαδρομές γύρω στα 11-12 hops όπως φαίνεται στο σχήμα time average.

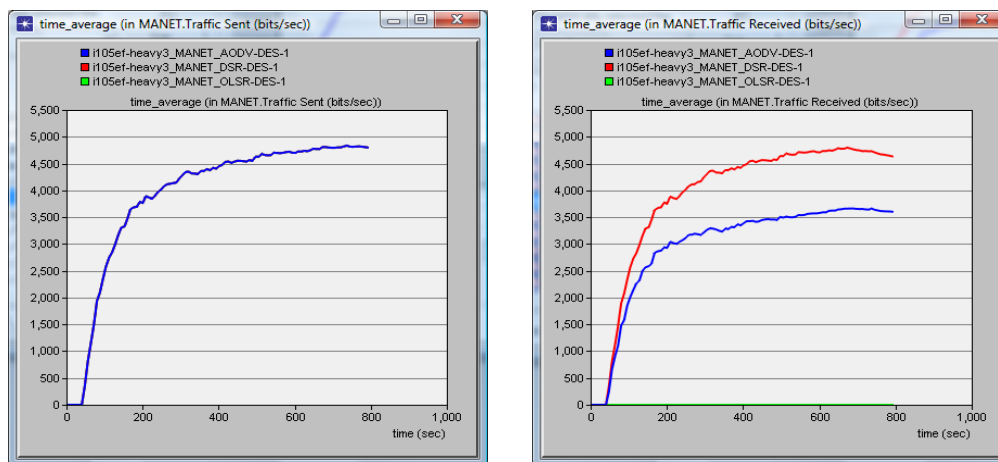
MANET. Delay (AODV, DSR, OLSR)



Πιο πάνω βλέπουμε τις γραφικές παραστάσεις του end-to-end delay για τον κάθε αλγόριθμο για το σενάριο Heavy. Η γραφική παράσταση στα αριστερά είναι η As Is ενώ στα δεξιά έχουμε την παράσταση του time average. Το end-to-end delay αφορά στον χρόνο από την δημιουργία του MANET πακέτου στον αποστολέα μέχρι και την καταστροφή του στον παραλήπτη. Όπως φαίνεται και από την γραφική παράσταση αυτή, ο αλγόριθμος αυτός δεν

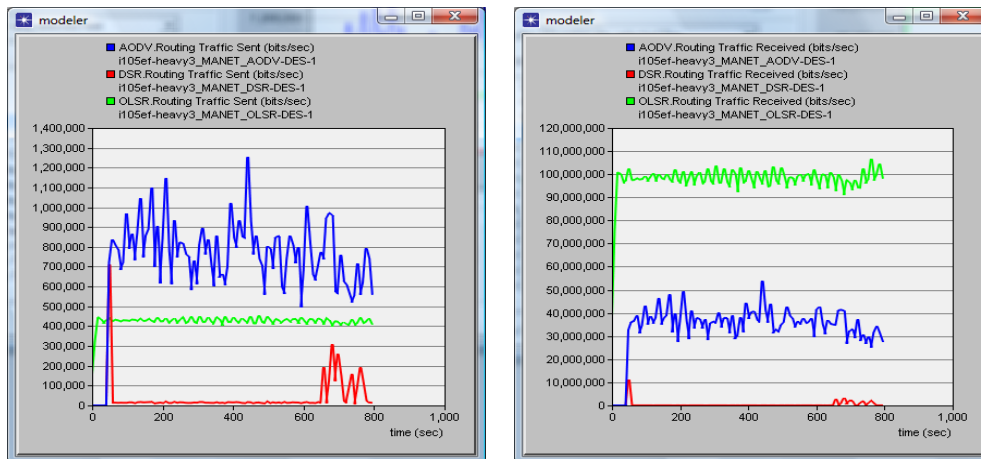
αποδίδει καθόλου στο σενάριο Heavy γι' αυτό και δεν παρουσιάζεται στις γραφικές παραστάσεις. Από την άλλη όσο αφορά τους reactive αλγόριθμους βλέπουμε ότι ο DSR έχει αρκετά καλή απόδοση με πολύ χαμηλό delay, το οποίο αυξάνεται λίγο μόνο προς το τέλος όπου φτάνει μέχρι τα 0.25 δευτερόλεπτα. Ο AODV παρουσιάζει όπως και στις προηγούμενες περιπτώσεις μεγάλες αυξομειώσεις οι οποίες φτάνουν μέχρι και τα 0.65 δευτερόλεπτα.

MANET. Traffic Received/Sent (bits/second) (AODV, DSR, OLSR)



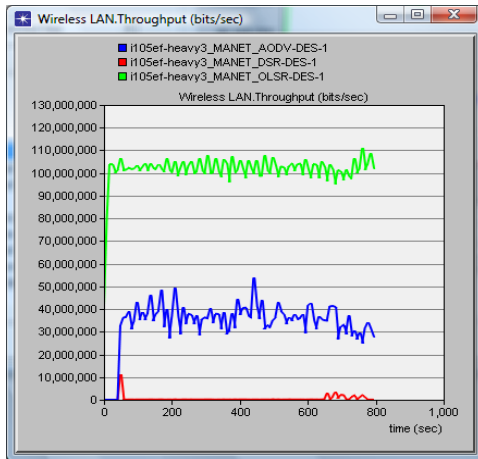
Στις δύο γραφικές παραστάσεις πιο πάνω βλέπουμε το traffic sent και received για τους τρεις αλγόριθμους. Οι γραφικές παραστάσεις που χρησιμοποιούμε εδώ αφορούν στο time average. Στα αριστερά βλέπουμε το traffic sent το οποίο είναι το ίδιο και για τους τρεις αλγόριθμους. Στα δεξιά βλέπουμε το raw traffic το οποίο καταφθάνει στους προορισμούς. Αυτό το οποίο παρατηρούμε είναι ότι ο αλγόριθμος ο οποίος έχει υψηλότερη αποτελεσματικότητα είναι ο DSR αφού προσεγγίζει κατά πολύ το traffic το οποίο αποστέλλεται από τους παραλήπτες. Ο AODV είναι λίγο πιο κάτω από τον DSR και στο τέλος της προσομοίωσης είναι λίγο πιο πάνω από 3500 bits/sec. Ο αλγόριθμος OLSR παρουσιάζει απογοητευτικά αποτελέσματα αφού κυμαίνεται σε πολύ χαμηλά επίπεδα αφού είναι πολύ κοντά στα 0bits/sec. Αυτό το οποίο φαίνεται να έχει επηρεάσει τον OLSR είναι η οι αλλαγές που κάναμε στις ρυθμίσεις του αλγορίθμου λόγω του περιορισμού στην μνήμη των μηχανών ώστε να τον καταφέρουμε να ολοκληρώσει την προσομοίωση.

Routing traffic Received/Sent (bits/sec) (AODV, DSR, OLSR)



Εδώ μπορούμε να παρατηρήσουμε το routing traffic (overhead στο δίκτυο μας) με την γραφική παράσταση As Is. Στα αριστερά βλέπουμε το routing traffic το οποίο αποστέλλεται από τον κάθε ένα από τους τρεις αλγόριθμους. Στα δεξιά μπορούμε να παρατηρήσουμε το routing traffic που φτάνει στους κόμβους του δικτύου. Κάτι αξιοσημείωτο είναι ότι το traffic received είναι πολύ περισσότερο από το traffic sent. Ο λόγος είναι διότι ένα control πακέτο το οποίο αποστέλλεται με τη μορφή broadcasting στο δίκτυο δημιουργείται μόνο στο κόμβο που το στέλλει. Οι υπόλοιποι κόμβοι οι οποίοι θα το επαναπροωθήσουν απλά το κάνουν duplicate το αποστέλλουν σε όλους τους γείτονες τους. Έτσι ένα control πακέτο το οποίο δημιουργείται και αποστέλλεται με τη μορφή broadcasting φτάνει σε πολλούς παραλήπτες. Αυτός είναι και ο λόγος που έχουμε τόσο μεγάλο traffic received. Από εκεί και πέρα αυτό το οποίο μπορούμε να παρατηρήσουμε είναι ότι ο DSR έχει πολύ χαμηλό overhead, σχεδόν μηδενικό, εκτός λίγο προς το τέλος της προσομοίωσης μας που αυξάνεται ελάχιστα, αφού σε αυτό συντείνουν το ότι δεν χρησιμοποιεί hello messages και το ότι χρησιμοποιεί cached replies. Επίσης φαίνεται ότι είναι και πολύ μικρό το traffic το οποίο φτάνει στους κόμβους στον αλγόριθμο DSR. Ο AODV βρίσκεται αρκετά πιο ψηλά αφού στέλνει συνεχώς γύρω στα 700,000 bits/sec και περισσότερα ενώ ο OLSR κινείται και αυτό σταθερά στα 400000 bits/sec. Όσο αφορά τώρα το routing traffic received παρατηρούμε ότι πάλι ο DSR είναι πολύ χαμηλά, ο AODV κινείται κοντά στα 40,000,000 bits/sec ενώ ο OLSR είναι πάρα πολύ ψηλά σε σύγκριση με τους δύο προηγούμενους αλγόριθμους αφού φτάνουν στους κόμβους του δικτύου συνεχώς κοντά στα 100,000,000 bits/sec. Βλέπουμε ότι το routing traffic που παράγει ο OLSR φτάνει σε πολύ περισσότερους κόμβους από ότι με τον αλγόριθμο AODV ο οποίος στέλνει λιγότερα από τον OLSR αλλά λαμβάνει περισσότερα.

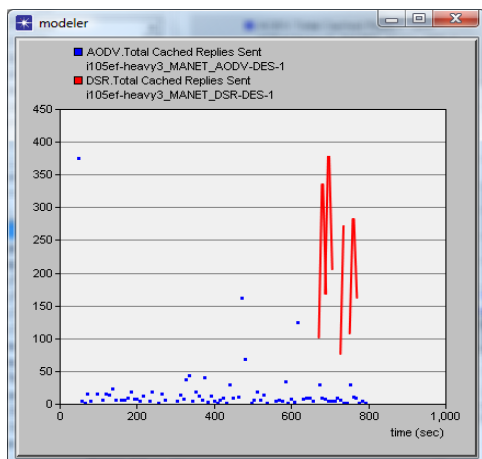
Wireless LAN.Throughput (AODV, DSR, OLSR)



Δίπλα μπορούμε να δούμε το throughput γενικά στο δίκτυο μας. Αυτό το οποίο παρατηρούμε είναι ότι ο αλγόριθμος OLSR χρησιμοποιεί πολύ περισσότερο τη γραμμή που του παρέχεται σε αντίθεση με τους δύο άλλους αλγορίθμους όπου ο AODV χρησιμοποιεί αρκετά το link ενώ ο DSR σχεδόν καθόλου. Η υψηλή χρήση της γραμμής από το κάθε κόμβο σίγουρα θα τον περιορίσει σε περίπτωση που χρειαστεί να αυξηθεί η πληροφορία που θέλουμε να διακινηθεί

μέσα στο δίκτυο. Από την άλλη ο DSR με την ελάχιστη χρήση λόγω του ότι δεν χρησιμοποιεί hello messages, έχει την δυνατότητα να μπορεί να αποστέλλει πολύ περισσότερες πληροφορίες. Αυτό έχει φανεί και πιο πάνω όπου βλέπουμε να φτάνουν πολύ λίγες πληροφορίες στο προορισμό σε σχέση με αυτές που αποστέλλονται.

Total Cached Replies Sent (AODV, DSR)

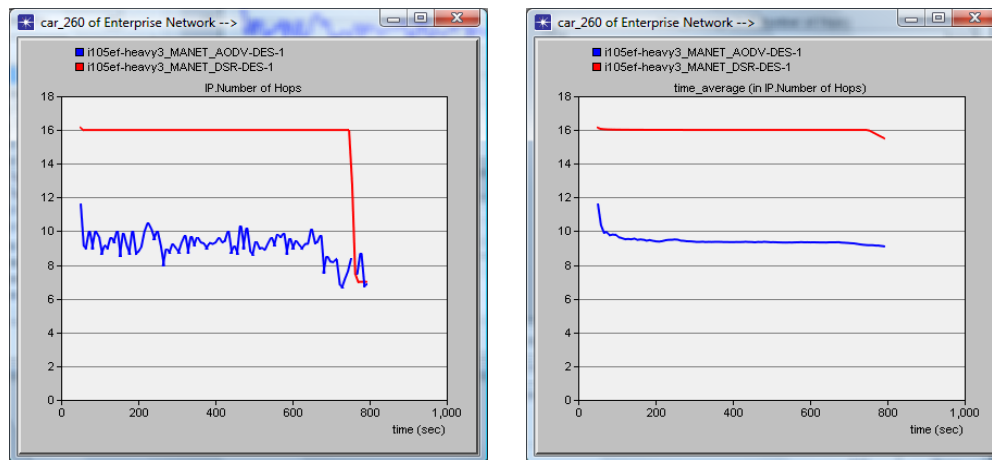


Δίπλα μπορούμε να δούμε τα cached replies τα οποία αποστέλλονται από τους δύο αλγόριθμους (AODV,DSR) οι οποίοι χρησιμοποιούν αυτό το χαρακτηριστικό. Βλέπουμε ότι ο αλγόριθμος DSR χρησιμοποιεί ευρέως αυτό το χαρακτηριστικό κυρίως προς το τέλος αφού βλέπουμε να στέλνει περισσότερες από 300 απαντήσεις, σχεδόν 400. Από την άλλη ο AODV χρησιμοποιεί και αυτός την δυνατότητα αυτή αλλά σε μικρότερο βαθμό από ότι

τον DSR αφού βλέπουμε τα cached replies να κυμαίνονται μεταξύ 0-20 τις περισσότερες φορές. Βλέπουμε ότι ο αλγόριθμος DSR έχει ένα μεγάλο πλεονέκτημα λόγω της χρήσης αυτής των cached replies οι οποίες μπορούν να μειώσουν κατά πολύ το overhead στο δίκτυο και να του επιτρέψουν να μειώσει το delay.

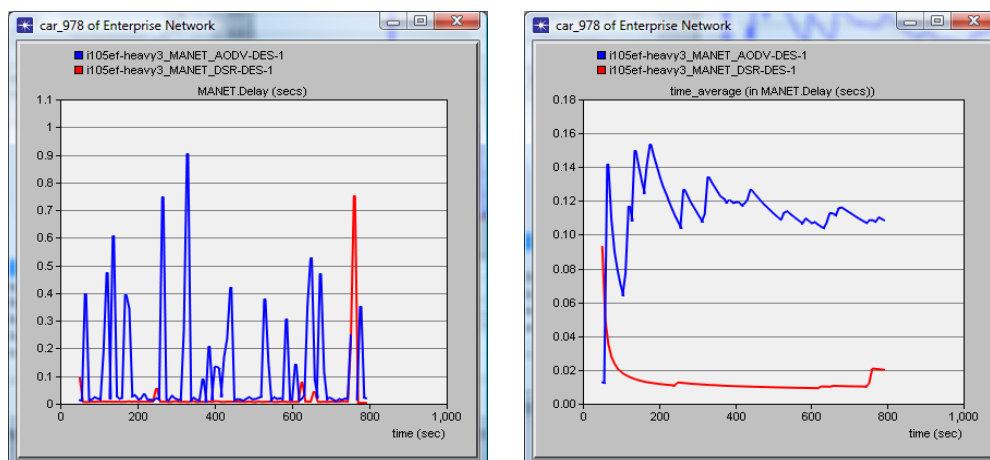
5.3.2 Μελέτη σύνδεσης κόμβων 260-978

IP.number of Hops (AODV, DSR, OLSR)



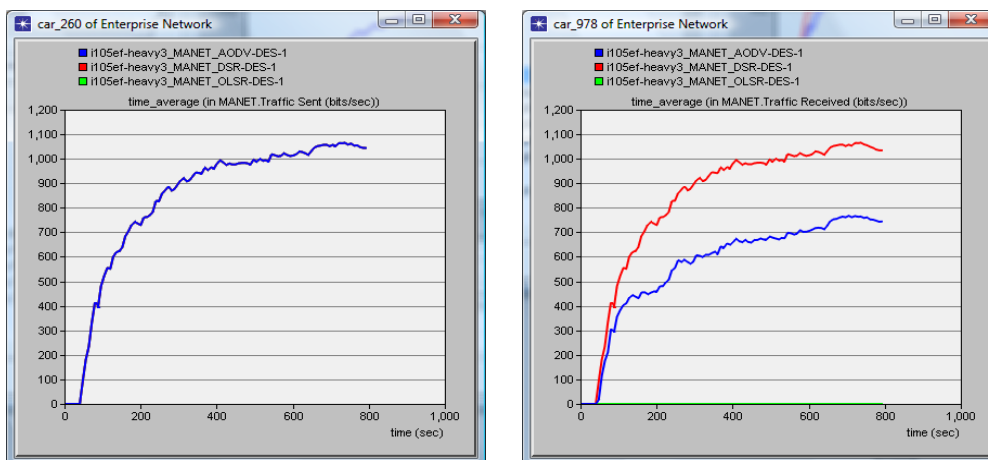
Εδώ μπορούμε να παρατηρήσουμε το αριθμό των κόμβων οι οποίοι παρεμβάλλονται μεταξύ της σύνδεσης η οποία γίνεται μεταξύ των κόμβων 260 και 978. Στα αριστερά έχουμε την γραφική As Is και δεξιά την γραφική παράσταση time average. Αυτό το οποίο μπορούμε να παρατηρήσουμε ότι ο αλγόριθμος OLSR δεν φαίνεται στην συγκεκριμένη διαδρομή πράγμα που σημαίνει ότι δεν φτάνουν καθόλου πακέτα στον προορισμό όπως παρατηρήσαμε και στις global στατιστικές. Από εκεί και πέρα ο AODV κυμαίνεται κοντά στα 10 hops. Ο DSR δείχνει να δημιουργεί μία διαδρομή η οποία περιλαμβάνει περισσότερα hops, γύρω στα 16 hops η οποία προσθέτει περισσότερο delay στην παράδοση των πακέτων.

MANET. Delay (AODV, DSR, OLSR)



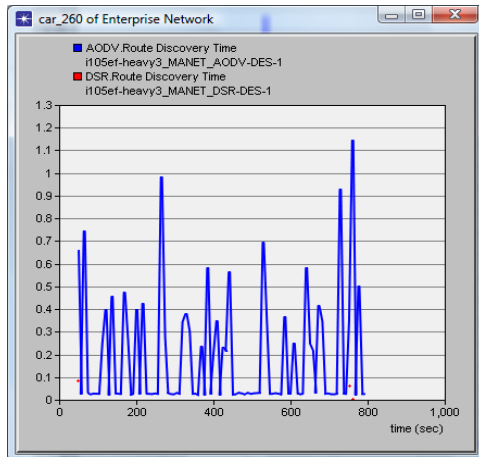
Όπως φαίνεται και από την γραφική παράσταση ο OLSR δεν εμφανίζεται καθόλου λόγω ίσως του ότι δεν φτάνουν καθόλου πληροφορίες προς το προορισμό. Οι δύο λόγοι που φαίνεται να επηρέασαν την απόδοση του αλγόριθμου είναι οι αλλαγές που κάναμε για να μπορέσει να τρέξει ο αλγόριθμος στην μηχανή καθώς και η μεγάλη συγκέντρωση κόμβων σε μικρό εμβαδό πράγμα που όπως έχει αποδειχθεί προκαλεί στον αλγόριθμο πολύ μεγάλο overhead. Από την άλλη το μεγαλύτερο delay φαίνεται να το παρουσιάζει ο AODV με τον DSR να βρίσκεται σε πολύ χαμηλά επίπεδα. Ο AODV έχει πάλι πολλές αυξομειώσεις οι οποίες όμως είναι πιο έντονες αυτή τη φορά λόγω του μεγαλύτερου χρόνου που χρειάζεται ώστε να ανακαλυφθεί μία διαδρομή και φτάνουν μέχρι και τα 0.9 δευτερόλεπτα. Ο DSR προς το τέλος της προσομοίωσης φαίνεται να αυξάνει το delay λόγω της ανάγκης εκείνη την ώρα να βρει καινούργιες διαδρομές προς το προορισμό αφού όπως φαίνεται χάνει τις προηγούμενες. Πιο συγκεκριμένα φτάνει λίγο πιο πάνω από τα 0.7 δευτερόλεπτα. Τις υπόλοιπες στιγμές παραμένει σε πολύ χαμηλά επίπεδα.

MANET. Traffic Received/Sent (bits/second) (AODV, DSR, OLSR)



Εδώ βλέπουμε το traffic που στέλλεται και λαμβάνεται από τους δύο αυτούς κόμβους. Όπως μπορούμε να παρατηρήσουμε στέλλεται το ίδιο traffic και για τους τρεις αλγόριθμους όπως φαίνεται στην γραφική παράσταση του traffic sent. Όσο αφορά τώρα το traffic το οποίο λαμβάνεται από το κόμβο αυτό βλέπουμε ότι ο DSR υπερνικά τους υπόλοιπους δύο αλγόριθμους με το AODV έρχεται δεύτερος και τον OLSR μην εμφανίζεται καθόλου αφού δεν λαμβάνει καθόλου πακέτα ο αλγόριθμος αυτός όπως είδαμε και πριν.

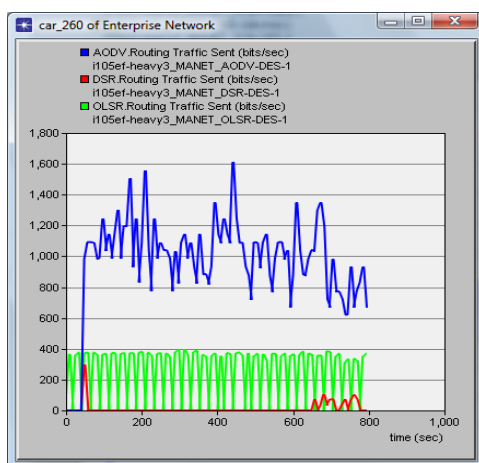
Route Discovery Time (AODV, DSR)



Εδώ μπορούμε να παρατηρήσουμε το χρόνο που χρειάζεται ο κόμβος 260 ώστε να μπορεί να βρει την διαδρομή προς τον κόμβο 978. Βλέπουμε ότι ο AODV χρειάζεται συνεχώς να ανακαλύπτει καινούργιες διαδρομές προς τον προορισμό ενώ υπάρχουν και αρκετά σκαμπανεβάσματα όσο αφορά τον χρόνο. Βλέπουμε γενικά ότι όσο αυξάνεται το μέγεθος του δικτύου, ο AODV δυσκολεύεται να αντεπεξέλθει τόσο εύκολα όπως με πιο μικρά σενάρια. Ο DSR πάλι δεν

εφαρμόζει και πολλές φορές την διαδικασία αυτή ενώ ο χρόνος που χρειάζεται ώστε να βρει κάποια διαδρομή είναι πολύ μικρός.

Routing traffic Sent (bits/sec) (AODV, DSR, OLSR)



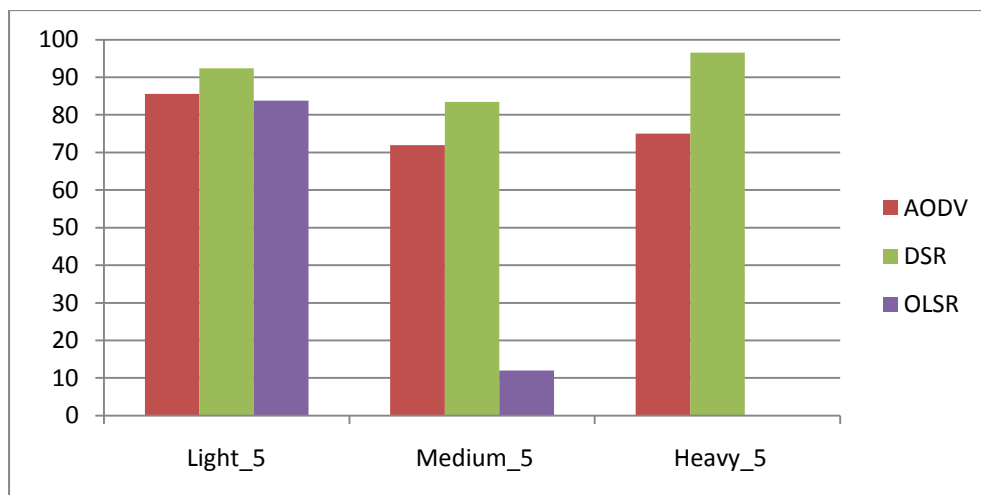
Όσο αφορά το routing traffic το οποίο αποστέλλεται από τον κόμβο 260, βλέπουμε ότι και πάλι ο DSR παρουσιάζει πολύ μικρά ποσοστά. Από την άλλη ο OLSR είναι πιο πάνω αλλά διατηρεί συνεχώς σταθερό το πόσο το οποίο στέλλει ο συγκεκριμένος κόμβος. Τέλος ο AODV είναι πολύ ψηλότερα και από τους δύο αυτούς κόμβους αφού πολλές φορές χρειάζεται να στείλει περισσότερα από 1000 bits/sec.

5.4 Σύγκριση μεταξύ σεναρίων

Μετά την σύγκριση για κάθε σενάριο ξεχωριστά, θα κάνουμε και μία σύγκριση μεταξύ των σεναρίων για να μελετήσουμε και κάποιες παραμέτρους τις οποίες θα πρέπει να πληρούν οι αλγόριθμοι που εφαρμόζονται σε τέτοιου είδους δίκτυα όπως scalability και αξιοπιστία.

Για την αξιολόγηση αυτή έχουμε χρησιμοποιεί ένα άλλο feature το οποίο μας παρέχει η OPNET, τα scalar αποτελέσματα τα οποία μας δίνουν το sample mean average για κάθε μετρική την οποία έχουμε επιλέξει ώστε να μπορούμε να δημιουργήσουμε γραφικές οι οποίες θα μας παρουσιάζουν ποσοστά επί τοις εκατό. Για την εξαγωγή των γραφικών παραστάσεων χρησιμοποιήθηκε το εργαλείο Microsoft Excel 2007. Και εδώ θα παρουσιάσουμε τα αποτελέσματα τα οποία έχουν προκύψει από τα σενάρια όπου το traffic δημιουργείται από 5 κόμβους. Οι υπόλοιπες γραφικές παραστάσεις θα παρουσιαστούν στο Παράρτημα β εκτός και εάν υπάρχει κάποια γραφική παράσταση η οποία μας βοηθά να βγάλουμε περισσότερα συμπεράσματα. Πιο κάτω μπορούμε να δούμε τις γραφικές παραστάσεις:

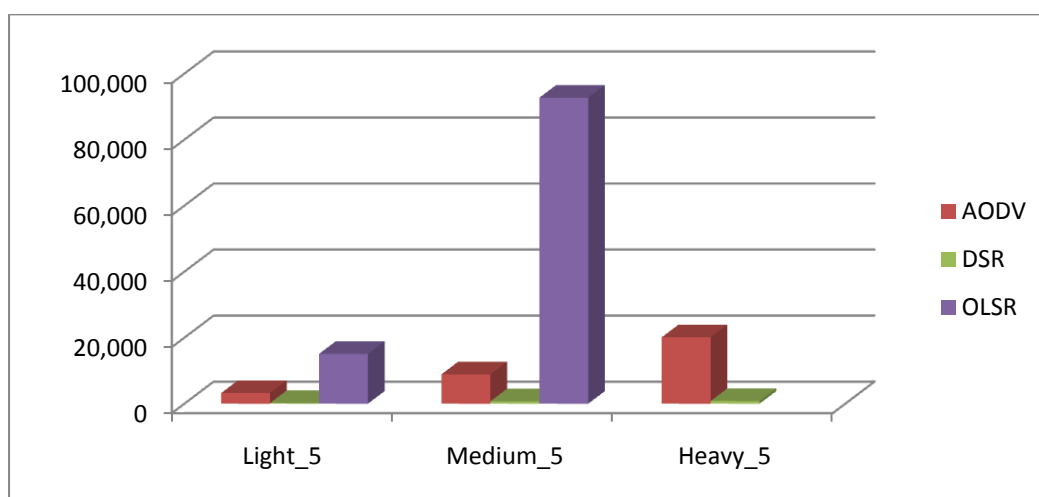
Packet Delivery Ratio (%) (AODV, DSR, OLSR)



Σε αυτή τη γραφική παράσταση μπορούμε να παρατηρήσουμε το packet delivery ratio για κάθε σενάριο για κάθε ένα από τους τρεις αλγόριθμους. Η γραφική αυτή εξάχθηκε από τις μέσες τιμές (mean average) του traffic received και traffic sent που πήραμε μετά που τρέξαμε τα σενάρια. Όπως μπορούμε να παρατηρήσουμε και στα τρία σενάρια ο αλγόριθμος ο οποίος έχει την ψηλότερη τιμή είναι ο DSR αφού συγκεντρώνει στο σενάριο Light 92% περίπου, στο σενάριο Medium 83% και στο σενάριο Heavy 96%. Οι τιμές αυτές θεωρούνται πολύ

καλές αφού ο αλγόριθμος επιτυγχάνει πολύ ψηλές τιμές. Κάτι άλλο που είναι αξιοσημείωτό είναι ότι ο αλγόριθμος αυτός στο σενάριο Heavy έχει το ψηλότερο του ποσοστό του. Αυτό φαίνεται να οφείλεται στην πυκνότητα των κόμβων καθώς και στην πιο χαμηλή κινητικότητα η οποία υπάρχει στο δίκτυο. Όπως φαίνεται η πυκνότητα των κόμβων δεν επηρεάζει τον αλγόριθμο ως προς το overhead καθώς όπως έχουμε δει και κατά την μελέτη των σεναρίων ξεχωριστά, ο αλγόριθμος DSR έχει πάρα πολύ μικρό overhead. Ο αλγόριθμος AODV παρουσιάζει και αυτός αρκετά ψηλά αποτελέσματα αλλά όχι τόσο καλά όσο του αλγόριθμου DSR. Πιο συγκεκριμένα στο σενάριο Light, ο αλγόριθμος AODV, παρουσιάζει ποσοστό επιτυχίας 85.5%, στο σενάριο Medium ποσοστό επιτυχίας 72% και στο σενάριο Heavy ποσοστό επιτυχίας 75%. Και εδώ παρατηρείται στο σενάριο Heavy να έχουμε ψηλότερο ποσοστό επιτυχίας από ότι στο σενάριο medium λόγω της μικρότερης ταχύτητας λόγω της οποίας κινούνται οι κόμβοι στο δίκτυο. Τέλος ο αλγόριθμος OLSR καταφέρνει να επιτύχει καλό ποσοστό μόνο στο σενάριο Light όπου παρουσιάζει 83%. Στα άλλα δύο σενάρια βλέπουμε ότι ο αλγόριθμος παρουσιάζει κατακόρυφη πτώση αφού στο σενάριο medium έχει ποσοστό μόνο 12% ενώ στο σενάριο Heavy 0%. Αυτό το οποίο έχει επηρεάσει είναι η αλλαγή στις ρυθμίσεις όπως έχουμε αναφέρει καθώς και η κατακόρυφη αύξηση του overhead στο αλγόριθμο αυτό αφού λόγω της proactive φύσης του αντιμετωπίζει αυτού του είδους τα προβλήματα όπως έχουμε δει και πιο πριν.

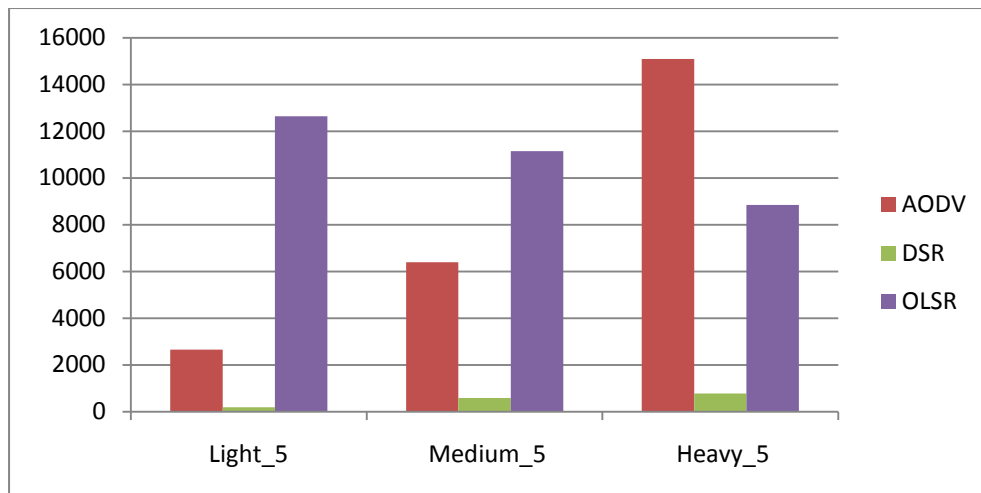
Routing Packet Transmission Ratio (%) (AODV, DSR, OLSR)



Στην πιο πάνω γραφική παράσταση μπορούμε να δούμε το routing packet transmission ratio για κάθε σενάριο για κάθε ένα από τους τρεις αλγόριθμους που δουλεύουν. Το routing packet transmission ratio αφορά το ποσοστό του routing traffic που στέλνεται στο δίκτυο σε

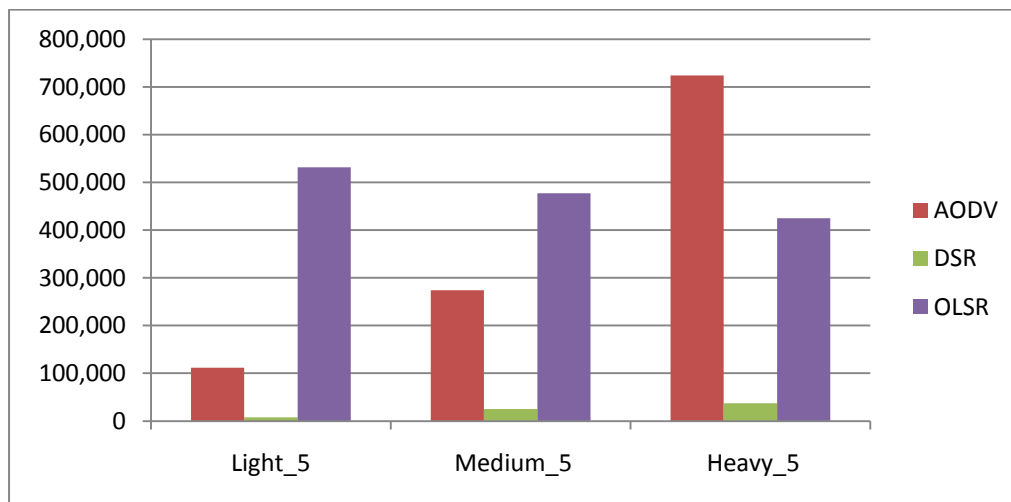
σύγκριση με τα MANET πακέτα τα οποία φτάνουν στους προορισμούς τους. Οι τιμές αφορούν πάντοτε τα scalar δεδομένα που πήραμε από τις προσομοιώσεις και είναι σε ποσοστό επί τοις εκατό. Ο αλγόριθμος ο οποίος θεωρείται καλύτερος είναι αυτός ο οποίος έχει το μικρότερο ποσοστό από τους τρεις αφού αυτό υποδηλώνει ότι για την αποστολή του συγκεκριμένου όγκου δεδομένων χρειάζεται λιγότερο overhead από τους υπόλοιπους. Στην συγκεκριμένη γραφική παράσταση βλέπουμε ότι τα μικρότερα ποσοστά τα παρουσιάζει ο αλγόριθμος DSR αφού στο σενάριο Light παρουσιάζει ποσοστό 206%(σημαίνει ότι για κάθε 100 bits τα οποία αποτελούν καθαρή πληροφορία και φτάνουν στους προορισμούς τους αποστέλλονται στο δίκτυο 206 bits routing traffic), στο σενάριο Medium παρουσιάζει ποσοστό 704% ενώ στο σενάριο Heavy παρουσιάζει ποσοστό 801%. Στην συνέχεια έχουμε τον αλγόριθμο AODV ο οποίος στο σενάριο Light παρουσιάζει ποσοστό 3214%, στο σενάριο Medium παρουσιάζει ποσοστό 8882% ενώ στο σενάριο Heavy παρουσιάζει ποσοστό 20105%. Βλέπουμε ότι χρειάζονται πολύ περισσότερα control bits από ότι τα δεδομένα που φτάνουν προς τους προορισμούς μας ανά πάσα χρονική στιγμή. Ο αλγόριθμος OLSR καταρχήν δεν μας δίνει καθόλου αποτέλεσμα στο σενάριο Heavy αφού δεν φτάνουν δεδομένα προς κανένα από τους προορισμούς. Στο σενάριο Light το ποσοστό που έχουμε είναι 15000% ενώ στο σενάριο Medium είναι γύρω στο 92691%. Παρατηρούμε ότι γενικά και οι τρεις αλγόριθμοι χρησιμοποιούν πολύ περισσότερο control traffic από ότι οι πληροφορίες που στέλνουν. Ο αλγόριθμος ο οποίος έχει τα καλύτερα αποτελέσματα είναι πάλι ο DSR και αν τον συγκρίνουμε με τους υπόλοιπους δύο οι διαφορές είναι γενικά πολύ μεγάλες. Φυσικά εδώ να αναφέρουμε ότι αυτό το οποίο επηρεάζει αυτό το ποσοστό δεν είναι το ίδιο και για τους τρεις αλγόριθμους. Οι αλγόριθμοι AODV και DSR εξαρτώνται και από την αύξηση των κόμβων σε κάθε σενάριο αλλά και από το ότι πρέπει κάθε φορά να δημιουργούμε διαδρομές μεγαλύτερες σε σύγκριση με τα προηγούμενα σενάρια και άρα . Ο αλγόριθμος OLSR εξαρτάται αποκλειστικά και μόνο από τον αριθμό των κόμβων του δικτύου όσο αφορά το ποσοστό αυτό αφού ως proactive αλγόριθμος διατηρεί πληροφορίες για κάθε κόμβο του δικτύου.

Routing Traffic Sent Vs Manet. Traffic Sent Ratio (%) (AODV, DSR, OLSR)



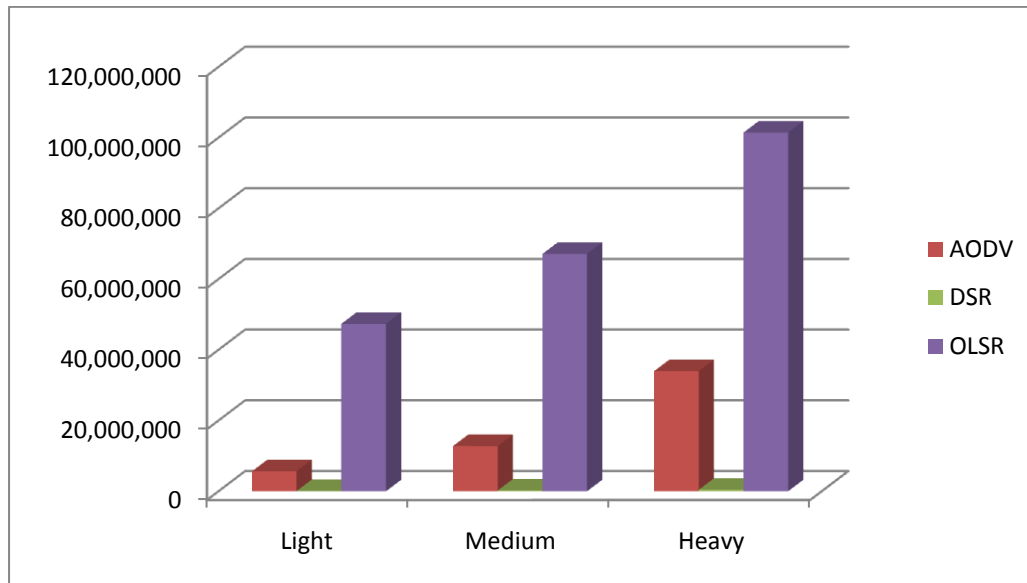
Εδώ βλέπουμε μία γραφική παράσταση παρόμοια με την προηγούμενη αλλά εδώ αντί το routing traffic received, έχουμε το routing traffic sent. Θέλουμε να δούμε δηλαδή από το συνολικό traffic που στέλνουν οι κόμβοι του δικτύου, πόσα bits είναι πληροφορίες και πόσα είναι για routing. Και εδώ αυτό το οποίο βλέπουμε είναι ποσοστό επί τοις εκατό. Αυτό το οποίο παρατηρούμε είναι ότι και πάλι ο αλγόριθμος DSR παρουσιάζει τα καλύτερα αποτελέσματα αφού έχει τα μικρότερα ποσοστά και στα τρία σενάρια. Ο αλγόριθμος AODV καθώς αυξάνει ο αριθμός των κόμβων παρουσιάζει πολύ μεγάλη αύξηση του ποσοστού αυτού ενώ ο αλγόριθμος OLSR μειώνεται σταδιακά μεταξύ των σεναρίων παρόλο που γενικά διατηρείται σε πολύ ψηλά επίπεδα. Πιο συγκεκριμένα ο αλγόριθμος DSR παρουσιάζει στο σενάριο Light ποσοστό 190%, στο σενάριο Medium 588% και στο σενάριο Heavy 774%. Ο αλγόριθμος AODV στο σενάριο Light παρουσιάζει ποσοστό 2653%, στο σενάριο Medium ποσοστό 6394% και στο σενάριο Heavy παρουσιάζει ποσοστό 15085%. Ο OLSR έχει ποσοστά στο Light 12643%, στο Medium 11146% και στο Heavy 8847%. Βλέπουμε ότι οι αλγόριθμοι AODV και OLSR έχουν πολύ ψηλά ποσοστά πράγμα που σημαίνει ότι στέλνουν πολύ περισσότερο control traffic σε σύγκριση με την πληροφορία που στέλνουν στο δίκτυο και αυτό αποτελεί ένα μεγάλο μειονέκτημα για αυτούς σε αντίθεση με τον αλγόριθμο DSR ο οποίος ενώ στέλνει πολύ λιγότερο routing traffic επιτυγχάνει πολύ καλύτερα αποτελέσματα.

Routing traffic Sent (mean bits/sec) (AODV, DSR, OLSR)



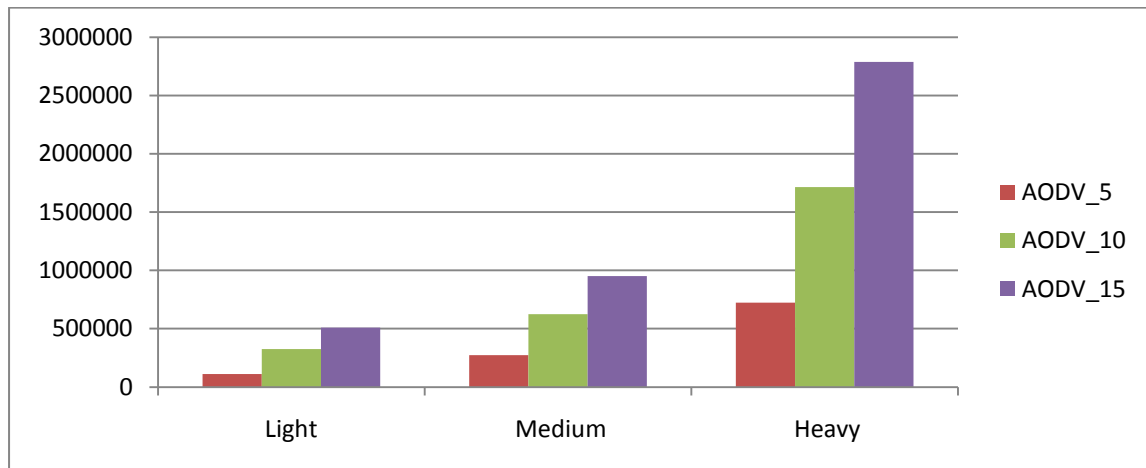
Στην πιο πάνω γραφική παράσταση βλέπουμε το routing traffic σε απόλυτες τιμές πάντα όσο αφορά το mean average. Όπως βλέπουμε η μορφή της γραφικής αυτής παράστασης είναι πολύ παρόμοια με την προηγούμενη αφού το traffic sent είναι το ίδιο για κάθε αλγόριθμο σε κάθε σενάριο. Όπως παρατηρούμε και πάλι έχουμε πολύ μικρό overhead στον αλγόριθμο DSR ενώ στον αλγόριθμο AODV το overhead αυξάνεται γραμμικά καθώς κινούμαστε μεταξύ των τριών διαφορετικών σεναρίων. Ο αλγόριθμος OLSR παρουσιάζει καλύτερα αποτελέσματα όσο πάμε από το σενάριο Light στο σενάριο Heavy αλλά παρόλα αυτά παραμένει σε πολύ υψηλά επίπεδα σε σύγκριση πάντοτε με τον αλγόριθμο DSR. Άρα συνολικά βλέπουμε ότι ο DSR σαν αλγόριθμος παρόλο που έχει το μικρότερο overhead παρουσιάζει το μεγαλύτερο ποσοστό στο packet delivery ratio.

Wireless LAN.Throughput (bits/sec) (AODV, DSR, OLSR)

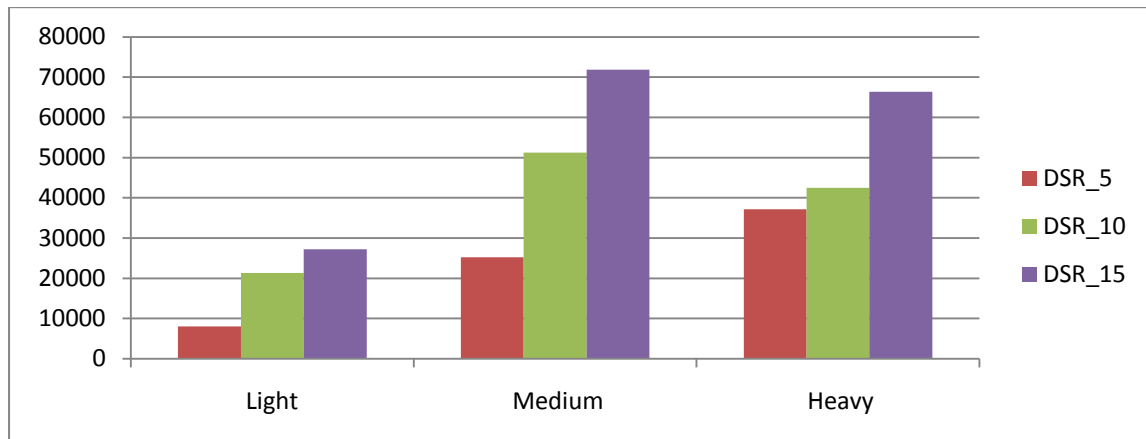


Τέλος στη γραφική παράσταση αυτή μπορούμε να δούμε το sample mean average για το throughput του ασύρματου περιβάλλοντος ολόκληρου του δικτύου μας. Αυτό το οποίο παρατηρούμε είναι ότι ο OLSR χρησιμοποιεί πολύ περισσότερο το ασύρματο περιβάλλον από ότι οι υπόλοιποι αλγόριθμοι. Αυτό το παρατηρήσαμε και στις γραφικές παραστάσεις τις οποίες παράξαμε από τον προσομοιωτή OPNET. Αυτό οφείλεται και στο ότι παρουσιάζει μεγάλο overhead στα σενάρια και έτσι αναγκάζεται να χρησιμοποιήσει πολύ περισσότερο τους πόρους που έχει στη διάθεση του. Ο DSR παραμένει σε πολύ χαμηλά επίπεδα σε σύγκριση πάντοτε με τους άλλους δύο αλγόριθμους. Ο AODV είναι μεταξύ των δύο προηγούμενων αλγόριθμων με το throughput να αυξάνεται καθώς πάμε από το σενάριο Light στο σενάριο Heavy.

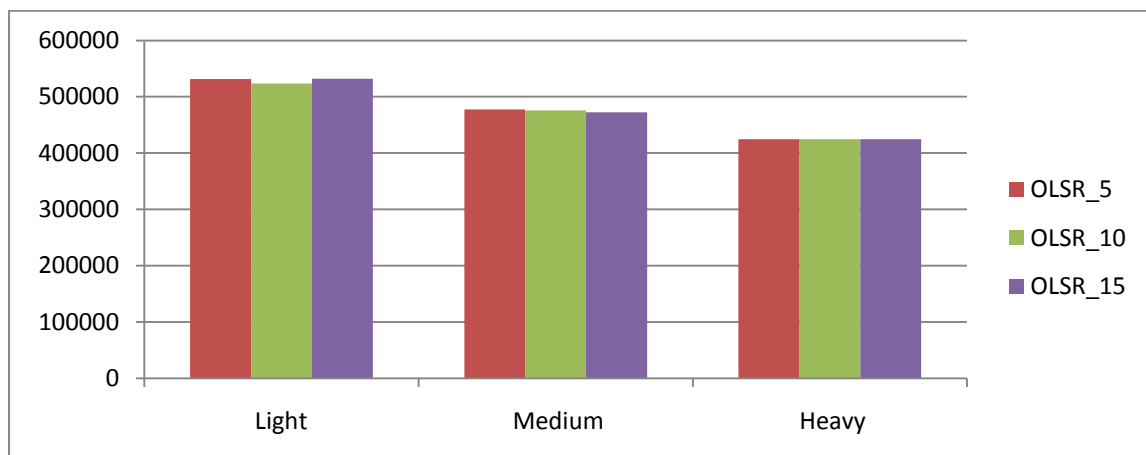
Routing traffic Sent (mean bits/sec) (AODV only)



Routing traffic Sent (mean bits/sec) (DSR only)



Routing traffic Sent (mean bits/sec) (OLSR only)



Στις τρεις πιο πάνω γραφικές παραστάσεις παρατηρούμε για κάθε αλγόριθμο ξεχωριστά το Routing traffic το οποίο στέλλει με τα τρία διαφορετικά traffic με τα οποία τρέξαμε τους τρεις αλγόριθμους.

Κάτι αξιοσημείωτό το οποίο παρατηρούμε εδώ είναι το ότι ο αλγόριθμος OLSR είναι ο μόνος ο οποίος μειώνει το routing traffic το οποίο στέλνει ή τουλάχιστον καταφέρνει να το διατηρεί σε σταθερά επίπεδα. Ο λόγος είναι ότι το overhead στον αλγόριθμο εξαρτάται μόνο από τον αριθμό των κόμβων στο δίκτυο ενώ επίσης όσο πιο πυκνό είναι το δίκτυο τόσο πιο λίγοι είναι οι MPRs. Σε αντίθεση με τον OLSR, το overhead των δύο υπόλοιπων reactive αλγόριθμων εξαρτάται και από τον αριθμό των κόμβων καθώς και από το traffic στο δίκτυο. Περισσότερο traffic στο δίκτυο προκαλεί περισσότερο overhead και αυτό φαίνεται ξεκάθαρα από τις γραφικές παραστάσεις.

5.5 Συνολικά Αποτελέσματα

Μετά από την παραπάνω αξιολόγηση μπορούμε να πούμε ότι τα αποτελέσματα μας συνοψίζονται στα πιο κάτω:

Ο αλγόριθμος AODV, αν τον δούμε συνολικά και στα τρία σενάρια, βλέπουμε ότι όσο αφορά την εξεύρεση διαδρομής δυσκολεύεται να διατηρήσει μία διαδρομή για μεγάλο χρονικό διάστημα αφού συνεχώς σπάζουν τα links μεταξύ των διαδρομών και αναγκάζεται να χρησιμοποιεί συνεχώς την διαδικασία route discovery ώστε να βρίσκει συνεχώς καινούργιες διαδρομές. Όσο αφορά το delay έχουμε παρατηρήσει ότι αυξομειώνεται συνεχώς το πιο πιθανόν και λόγω του ότι έχουμε συνεχώς το σπάσιμο των links και χρειάζεται να βρίσκουμε συνεχώς καινούργια μονοπάτια προς τη διαδρομή. Αρκετές φορές ο αλγόριθμος αυτός παρουσιάζει πολύ υψηλό delay, έστω και στιγμιαία, κάτι το οποίο θεωρείται κριτικής σημασίας για την απόδοση του δικτύου. Επίσης ο αλγόριθμος αυτός παρουσιάζει αρκετά καλό packet delivery ratio το οποίο διατηρείται και στα τρία σενάρια(Light, Medium και Heavy), και με το διαφορετικό traffic το οποίο δοκιμάσαμε ενώ φαίνεται να χρησιμοποιεί αρκετά και τους πόρους τους οποίους έχει.

Ο αλγόριθμος DSR δεν αντιμετωπίζει το πρόβλημα με το συχνό σπάσιμο των links ενώ αντιθέτως άλλη αναγκάζεται να δημιουργήσει πολύ πιο λίγες φορές διαδρομές προς το προορισμό και με πολύ μικρό χρόνο σε σχέση με τον αλγόριθμο AODV. Δημιουργεί

διαδρομές με μεγάλο αριθμό hops σε σύγκριση με τον αριθμό των hops της πιο σύντομης διαδρομής που έχουμε δει στον αλγόριθμο OLSR ενώ όταν χρειάζεται να βρει κάποια διαδρομή βλέπουμε να αυξάνεται το delay του αρκετά σε σύγκριση με τις υπόλοιπες στιγμές της προσομοίωσης. Το μεγάλο του πλεονέκτημα φαίνεται να είναι το πολύ υψηλό packet delivery ratio το οποίο φαίνεται να έχει ενώ παράλληλα παρουσιάζει και πολύ χαμηλό overhead παρόλο που είναι αλγόριθμος ο οποίος χρησιμοποιεί source routing. Εκείνο το οποίο του δίνει μεγάλο πλεονέκτημα είναι η χρήση του cache η οποία του επιτρέπει να μειώνει την περαιτέρω προώθηση των πακέτων όταν πρόκειται για broadcasting για την δημιουργία καινούργιας διαδρομής ενώ από την άλλη φαίνεται ότι δεν προκύπτουν stale caches αφού δεν έχουμε παρατηρήσει το φαινόμενο είτε να χάνονται πολλά πακέτα είτε να χρειάζεται συνεχώς να βρίσκει καινούργιες διαδρομές προς τους προορισμούς. Τέλος φαίνεται ότι χρησιμοποιεί πολύ λίγο τους πόρους του και πιο συγκεκριμένα το ασύρματο περιβάλλον, κάτι το οποίο του δίνει την δυνατότητα να μπορεί να το χρησιμοποιήσει για την αποστολή καθαρών πληροφοριών.

Ο αλγόριθμος OLSR παρουσιάζει πολύ υψηλό overhead λόγω της ιδιότητας του να είναι proactive ενώ από την άλλη δημιουργεί τις πιο σύντομες (βέλτιστες) διαδρομές όσο αφορά τον αριθμό των hops και παρουσιάζει το μικρότερο delay-σχεδόν μηδενικό- για τα συγκεκριμένα σενάρια τα οποία έχουν δουλέψει χωρίς πρόβλημα. Επιπλέον φαίνεται να χρησιμοποιεί σε μεγάλο βαθμό του πόρους που του δίνονται, έχει δηλαδή πάρα πολύ υψηλό throughput στην ασύρματη σύνδεση. Τα σενάρια Medium και Heavy παρουσίασαν πολύ κακά αποτελέσματα αφού λόγω του περιορισμού που είχαμε στην μηχανή κάναμε κάποιες αλλαγές, τις οποίες προαναφέραμε, και οι οποίες ‘χαλάρωσαν’ τον αλγόριθμο κάνοντας τον χειρότερο, σε σημείο όπου έχασε τα λίγα πλεονεκτήματα τα οποία είχε παρουσιάσει.

Κεφάλαιο 6

Συμπεράσματα

6.1 Τελικά Συμπεράσματα	80
6.1.1 Συμπεράσματα με βάση μετρικές	80
6.1.2 Συμπεράσματα με βάση περιβάλλον VANETS	82
6.1.3 Τελικά Συμπεράσματα	83
6.2 Μελλοντική Δουλειά	85

6.1 Τελικά Συμπεράσματα

Με βάση όλα τα αποτελέσματα τα οποία έχουμε αναλύσει στο προηγούμενο κεφάλαιο έχουμε καταλήξει σε πολλά αποτελέσματα τα οποία θα παρουσιάσουμε στο κεφάλαιο αυτό. Καταρχήν θα αναφερθούμε σε κάθε αλγόριθμο ξεχωριστά και θα αναφέρουμε τα απτά πλέον συμπεράσματα στα οποία έχουμε καταλήξει λαμβάνοντας πάντα υπόψη τη σύγκριση του με τους υπόλοιπους αλγόριθμους. Ακολούθως θα δείξουμε με την μορφή πίνακα ποιος αλγόριθμος υπερτερεί και ποιος όχι σε κάθε μετρική την οποία αναλύσαμε. Στη συνέχεια θα επαναλάβουμε το πίνακα αυτό αλλά αυτή τη φορά θα αναφερθούμε στο περιβάλλον VANETS και στο ποιος από τους αλγόριθμους αυτός ανταποκρίνεται σε κάθε ένα από τα χαρακτηριστικά που υπάρχουν σε ένα τέτοιο δίκτυο. Τέλος θα αναφερθούμε στη μελλοντική δουλειά που μπορεί να γίνει για την ολοκλήρωση του τελικού προβλήματος.

6.1.1 Συμπεράσματα με βάση μετρικές

Από τα αποτελέσματα τα οποία προέκυψαν με βάση το προηγούμενο σενάριο βγάζουμε τα εξής συμπεράσματα για τον κάθε αλγόριθμο με βάση τις μετρικές με τις οποίες τον αξιολογήσαμε:

Όπως κάναμε και κατά την συνοπτική επεξήγηση των συνολικών αποτελεσμάτων βλέπουμε ότι ο αλγόριθμος DSR είναι αυτός ο οποίος υπερτερεί γενικότερα, όσο αφορά όλα τα σενάρια δηλαδή, στο μεγαλύτερο ποσοστό παράδοσης πακέτων πληροφοριών, στο μικρότερο overhead όσο αφορά δηλαδή πακέτα δρομολόγησης, στη ελάχιστη χρησιμοποίηση του ασύρματου περιβάλλοντος, καθώς και στην ευρεία χρήση την οποία διατηρεί.

Ο αλγόριθμος AODV είναι περισσότερο δυναμικός και παρουσιάζει αρκετά καλό packet delivery ratio, όχι όμως τόσο καλό όσο του αλγορίθμου DSR. Επιπλέον δεν καταφέρνει να διατηρήσει για πολύ μεγάλο χρονικό διάστημα τις διαδρομές τις οποίες δημιουργεί, ενώ χρησιμοποιεί αρκετά τους πόρους τους οποίους έχει.

Ο αλγόριθμος OLSR, με τα λίγα που μπόρεσε να μας δείξει, φαίνεται ότι έχει πολύ γρήγορη παράδοση δεδομένων - delay - αλλά από την άλλη πολύ υψηλό throughput το οποίο είναι απόρροια της ιδιότητας την οποία έχει να είναι proactive.

Στο πίνακα πιο κάτω παρουσιάζουμε συνοπτικά για κάθε μία από τις αρχικές μετρικές τις οποίες θέσαμε ποιος αλγόριθμος υπερτερεί, ποιος έρχεται δεύτερος και ποιος τρίτος.

Πίνακας 6.1: Συγκριτική συνοπτική επίδοση αλγορίθμων που αξιολογήθηκαν βάση των μετρικών

	AODV	DSR	OLSR
Route Discovery Time	3	2	1
Number of Hops	2	3	1
End-to-end Delay	3	2	1
Packet delivery Ratio	2	1	3
Routing transmission Ratio	2	1	3
Throughput	2	1	3
Routing overhead	2	1	3

Από εδώ οι μετρικές οι οποίες θεωρούμε ότι παρουσιάζουν τη μεγαλύτερη σημασία είναι το end-to-end delay, το packet delivery ratio καθώς και το routing overhead όπου και στις τρεις ο αλγόριθμος που έχει τα καλύτερα αποτελέσματα είναι ο DSR Στο end-to-end delay επιλέξαμε τον αλγόριθμο DSR λόγω του ότι ο αλγόριθμος OLSR που είναι αντικειμενικά

καλύτερος στην μετρική αυτή δεν ανταποκρίθηκε όπως θα έπρεπε στα σενάρια Medium και Heavy.

6.1.2 Συμπεράσματα με βάση περιβάλλον VANETS

Πέρα από την αξιολόγηση βάση των μετρικών σημαντικό ρόλο παίζει και ποιος από τους αλγόριθμους ανταποκρίνεται καλύτερα στις συνθήκες οι οποίες επικρατούν στα δίκτυα VANETS και τις οποίες αναλύσαμε στο Κεφάλαιο 3. Τα αποτελέσματα τα οποία μας βοήθησαν να βγάλουμε τα συγκεκριμένα συμπεράσματα είναι τα συνολικά αποτελέσματα τα οποία είδαμε στο προηγούμενο κεφάλαιο. Αυτό το οποίο προκύπτει είναι ότι και στις τέσσερις μεγάλες κατηγορίες- scalability, αξιοπιστία, robustness και Qos- ο αλγόριθμος DSR αποδίδει καλύτερα από τους υπόλοιπους αλγόριθμους. Άρα θεωρείται ότι μπορεί να αποδώσει καλύτερα σε τέτοιου είδους δίκτυα σε σύγκριση με τους υπόλοιπους αλγόριθμους. Από την άλλη ο AODV έχει δείξει πως έχει τρωτά σημεία τα οποία δεν του δίνουν την δυνατότητα να συναγωνιστεί σε μεγάλο βαθμό τον DSR. Τέλος ο OLSR έχει επιδείξει μεγάλα πλεονεκτήματα αλλά και μεγάλα μειονεκτήματα για όσο μπορέσαμε να κάνουμε προσομοιώσεις με τις αρχικές του ρυθμίσεις. Παρόλα αυτά κανένας από τους τρεις αλγόριθμους δεν ανταποκρίθηκε πλήρως στις απαιτήσεις ενός τέτοιου δικτύου. Στο πιο κάτω πίνακα εμφανίζονται συνοπτικά τα αποτελέσματα για κάθε μία από τις τρεις μεγάλες κατηγορίες.

Πίνακας 6.2: Συγκριτική συνοπτική επίδοση αλγόριθμων που αξιολογήθηκαν βάση συνθηκών στα δίκτυα VANETS

	AODV	DSR	OLSR
Scalability	2	1	3
Αξιοπιστία	2	1	3
Robustness	2	1	3
Qos	3	1	2

6.1.3 Τελικά συμπεράσματα

Στα τελικά συμπεράσματα σημειώνουμε την συνολική πολύ καλή απόδοση του αλγόριθμου DSR συγκριτικά πάντοτε με τους υπόλοιπους δύο και τα χαρακτηριστικά του πάνω στα οποία φαίνεται να οφείλεται η απόδοση αυτή όπως η χρήση source routing, η χρήση cache και η μη χρήση μηνυμάτων hello.

Επίσης σημειώνουμε τα πολύ καλά αποτελέσματα τα οποία μπορεί να δώσει ένας proactive αλγόριθμος στο delay και αριθμό των hops μία διαδρομής αλλά και τα πολύ κακά αποτελέσματα τα οποία δίνει όσο αφορά το routing overhead και το throughput. Στα αποτελέσματα ο OLSR παρουσίασε πάρα πολύ υψηλό routing overhead, πράγμα το οποίο δείχνει ότι σε περίπτωση που το δίκτυο μου είναι κατά πολύ μεγαλύτερο τότε θα προκύψει πολύ πιο μεγάλο routing overhead. Επίσης σε περίπτωση δημιουργίας εφαρμογής η οποία θα έχει σαν σκοπό να ενημερώνει κάποιες υπηρεσίες σε περίπτωση δυστυχήματος τότε θα πρέπει να αποστέλλονται δεδομένα προς ένα συγκεκριμένο προορισμό(π.χ στην αρχή του αυτοκινητόδρομου) και δεν θα χρειάζεται να υπάρχει η γνώση για την ακριβή διαδρομή των υπόλοιπων κόμβων του δικτύου.

Εκτός τούτου, η πολύ υψηλή δυναμικότητα ενός αλγόριθμου μπορεί να προκαλέσει αρνητικά αποτελέσματα στην απόδοση του επηρεάζοντας κυρίως το delay και αυξάνοντας το overhead.

Επιπλέον σημαντικό είναι να αναφερθούμε και στο παράγοντα VANETS και να μελετήσουμε που υστερεί κάθε αλγόριθμος στο περιβάλλον αυτό. Η γενική εικόνα η οποία μας έχει δημιουργηθεί για τέτοιου είδους δίκτυα είναι ότι χρειάζεται ένας αλγόριθμος οποίος να αλλάζει πάρα πολύ γρήγορα σύμφωνα πάντα με τις πάρα πολύ δυναμικές αλλαγές που υπάρχουν στον κάθε δίκτυο. Αυτό δεν φαίνεται να το επιτυγχάνει κανένας από τους τρεις αλγορίθμους αφού και οι τρεις είναι κατασκευασμένοι ώστε να δουλεύουν με δίκτυα τα οποία έχουν αρκετά χαμηλότερη κινητικότητα. Ακόμα και ο OLSR ο οποίος είναι proactive αλγόριθμος και διατηρεί συνεχώς πληροφορίες για όλους τους κόμβους δεν παρουσιάζει τόσο καλά αποτελέσματα όπως οι υπόλοιποι αλγόριθμοι σε συγκεκριμένες μετρικές. Ένα παράδειγμα είναι όταν χρειάζεται να ανακαλυφθούν καινούργιες διαδρομές προς το προορισμό μετά το σπάσιμο κάποιων links. Έτσι καθυστερεί η διαδικασία μεταφοράς

πακέτων μέχρι να εξευρεθεί καινούργια διαδρομή. Αυτό μπορεί να προκαλέσει και την παραμονή κάποιων πακέτων σε ενδιάμεσους κόμβους αφού δεν θα μπορούν να δρομολογηθούν προς το προορισμό και θα πρέπει να ξανασταλούν από την πηγή. Έτσι στα VANETS πολλές φορές χρειάζεται να αντιμετωπιστεί το φαινόμενο αυτό και να βρεθούν λύσεις που θα αντιμετωπίζουν το πρόβλημα του συνεχούς σπασίματος των links. Για το λόγο αυτό γίνεται ευρεία χρήση των γεωγραφικών συντεταγμένων στα δίκτυα VANETS (με χρήση πολλές φορές του GPS) με σκοπό να μπορεί να υπάρχει άμεσα και πιο γρήγορα η εναλλακτική διαδρομή προς κάποιο προορισμό. Η αίσθηση που υπάρχει όσο αφορά τα δίκτυα VANETS είναι ότι θα πρέπει να δημιουργηθεί κάποιος αλγόριθμος ο οποίος να συνενώνει και reactive αλλά και proactive χαρακτηριστικά. Τα reactive χαρακτηριστικά, πολλά από τα οποία είδαμε στους αλγόριθμους AODV και DSR, θα έχουν ως σκοπό την αντιμετώπιση της υψηλής δυναμικότητας του δικτύου ενώ από την άλλη τα proactive χαρακτηριστικά θα έχουν ως σκοπό το να δώσουν στον αλγόριθμο την δυνατότητα να μπορεί να διατηρεί τέτοια στοιχεία ώστε να αντιμετωπίζει τα φαινόμενα τα οποία προκύπτουν από την δυναμικότητα του αλγόριθμου αυτού.

Συνεπώς αναλύοντας κάθε ένα από τους αλγόριθμους που μελετήσαμε μπορούμε να πούμε: Οι αλγόριθμοι AODV και DSR ακολουθούν πολύ καλές τεχνικές για αντιμετώπιση της κινητικότητας αλλά χρειάζονται επιπλέον τεχνικές ώστε να τους δώσουν την δυνατότητα να εφαρμόσουν αυτά που έχουμε προαναφέρει όπως ενσωμάτωση αλγορίθμου που να αφορά χρήση γεωγραφικών συντεταγμένων, χρήση cache το οποίο όμως να έχει πιο μικρό timer για τις διαδρομές από ότι αυτό που υπάρχει ήδη στον DSR, χρήση αλγόριθμου πιθανοτικού flooding, αντί του κλασσικού, με σκοπό την περαιτέρω μείωση του routing overhead και εφαρμογή κάποιων proactive στοιχείων ώστε να έχουμε καλύτερα αποτελέσματα στο end-to-end delay. Για παράδειγμα, σε περίπτωση που έχουμε εφαρμογή για δυστυχήματα και θα χρειάζεται εάν συμβεί κάτι, να αποσταλούν δεδομένα προς ένα συγκεκριμένο προορισμό, θα μπορούσε να γίνει μία εφαρμογή όπου κάθε κόμβος να διατηρεί διαδρομές προς τον συγκεκριμένο προορισμό μόνο και όχι για όλους.

Όσο αφορά τον αλγόριθμο OLSR, χρειάζεται να γίνει το αντίθετο, να προσπαθήσουμε να τον βελτιώσουμε εκεί που υπάρχουν τα μεγαλύτερα μειονεκτήματα, στο routing overhead καθώς και στο throughput. Αυτά τα κακά αποτελέσματα οφείλονται στην proactive ιδιότητα του. Έτσι θα πρέπει να μειώσουμε την ιδιότητα αυτή με την χρήση μεθόδων π.χ. πιθανοτικού

flooding καθώς και κάποιων χαρακτηριστικών από τους αλγόριθμους AODV και DSR ώστε ο OLSR να αποκτήσει και αυτός κάποια reactive χαρακτηριστικά. Φυσικά το μεγάλο στοίχημα στην περίπτωση αυτή θα είναι να μην επηρεαστούν, ή τουλάχιστον να επηρεαστούν πολύ λίγο, τα πολύ καλά αποτελέσματα που έχουμε σε συγκεκριμένες μετρικές οι οποίες είναι πολύ σημαντικές για την απόδοση των δικτύων αυτών.

Φυσικά όπως διαφαίνεται στην πιο πάνω ανάλυση δεν υπάρχει κάποιος από τους αλγόριθμους αυτούς που να έχει τα χαρακτηριστικά εκείνα τα οποία θα του δώσουν την αναμενόμενη απόδοση σε τέτοιου είδους δίκτυα. Γι' αυτό και θα πρέπει να γίνει πολύ ακόμη δουλειά μέχρι να δημιουργηθεί ένας αλγόριθμος ο οποίος να μπορεί να συνδυάζει όλα τα θετικά των πιο πάνω αλγόριθμων ώστε να μπορέσει να ανταποκριθεί στο σκοπό τον οποίο έχουμε θέσει από την αρχή της διπλωματικής αυτής εργασίας.

6.2 Μελλοντική Δουλειά

Η τελευταία παράγραφος του προηγούμενου κεφαλαίου μας δίνει την κατεύθυνση στην οποία θα πρέπει να κινηθούμε ώστε να μπορέσουμε να πετύχουμε το σκοπό τον οποίο θέσαμε εξ αρχής. Θα προτείνουμε εναλλακτικές λύσεις οι οποίες μπορούν να εφαρμοστούν σαν συνέχεια της συγκεκριμένης διπλωματικής ώστε να συνεχίζει η ενδελεχής αξιολόγηση των πρωτοκόλλων και να γίνουν βήματα προς την δημιουργία ενός αλγορίθμου ο οποίος να μπορεί να εφαρμοστεί με επιτυχία. Σαν μελλοντική δουλειά θα μπορούσαν να γίνουν τα εξής:

- Το πρώτο που θα μπορούσε να γίνει είναι η περαιτέρω διερεύνηση των δύο αλγόριθμων που δεν έχουν δουλέψει όπως θα έπρεπε (GRP και TORA) ώστε να μπορέσουν να δουλέψουν και να γίνει σύγκριση με τα αποτελέσματα τα οποία παρουσιάσαμε στην διπλωματική αυτή, ώστε να είναι πιο σφαιρική.
- Επιπλέον θα μπορούσε να μελετηθούν περισσότερο οι λόγοι οι οποίοι έχουν επηρεάσει τον αλγόριθμο OLSR στο να μην μπορεί να αποδώσει όπως θα έπρεπε λόγω της αλλαγής στις ρυθμίσεις του ώστε να φτάσουμε στο σημείο να έχουμε packet delivery ratio 0% στο σενάριο Heavy.

- Το τρίτο που θα μπορούσε να γίνει είναι η προσπάθεια αλλαγής κάποιου από τους τρεις υφιστάμενους αλγόριθμους, ώστε να αποκτήσει κάποια από τα πλεονεκτήματα τα οποία παρουσιάζουν οι υπόλοιποι αλγόριθμοι. Εκείνος ο οποίος θα μπορούσε να αποτελέσει τον αλγόριθμο βάσης για μία τέτοια ιδέα είναι ο DSR. Φυσικά αυτό θα μπορούσε να γίνει και με τον AODV αφού θα μπορούσε να γίνει η προσπάθεια να "αποκτήσει" χαρακτηριστικά του DSR (όπως π.χ. να ενσωματωθεί η χρήση cache στον αλγόριθμο αυτό) καθώς και κάποια άλλα χαρακτηριστικά όπως η χρήση πιθανοτικού flooding με την εφαρμογή του αλγόριθμου SAPF ώστε να μειωθεί το routing overhead. Όσο αφορά τον DSR η προσπάθεια που θα μπορούσε να γίνει αφορά στην ενσωμάτωση στον αλγόριθμο κάποιων proactive χαρακτηριστικών τα οποία θα μπορέσουν να του δώσουν ακόμη καλύτερη απόδοση κυρίως από πλευράς end-to-end delay. Αυτό φυσικά θα μπορούσε να προσθέσει περισσότερο routing overhead αλλά όπως έχουμε δει ο αλγόριθμος αυτός θα μπορούσε να δεχτεί κάποια αύξηση στο routing overhead του αφού παρουσιάζει πολύ μικρά ποσά στην συγκεκριμένη μετρική. Έτσι θα μπορούσε να προκύψει μια καινούργια αξιολόγηση για τον καινούργιο αλγόριθμο και να τεθούν κάποια όρια απόδοσης, ποια είναι δηλαδή τα χειρότερα ανεκτά όρια για κάθε μετρική τα οποία είναι επιτρεπτά για τον κάθε αλγόριθμο. Όσο για τον OLSR λόγω της proactive ιδιότητας του δεν φαίνεται να μπορεί να αντεπεξέλθει σε τόσο δυναμικό περιβάλλον αφού είναι τρομακτικά μεγάλη η χρήση routing overhead η οποία προσθέτει στο δίκτυο πολύ μεγάλο traffic ενώ όπως φαίνεται από τους υπόλοιπους αλγόριθμους η διαδικασία διατήρησης διαδρομών και γενικά διαχείρισης των δικτύων αυτών μπορεί να γίνει με πολύ λιγότερο routing traffic.
- Το πιο τολμηρό εγχείρημα από όλα είναι όμως η προσπάθεια δημιουργίας ενός αλγόριθμου ο οποίος θα μπορεί να "συνενώσει" όλα τα πλεονεκτήματα τα οποία έχουμε δει και στους τρεις αλγόριθμους πιο πάνω και επίσης να προσαρμοστεί στις ανάγκες των δικτύων VANETS ώστε να μπορεί να δουλέψει αποδοτικά στα συγκεκριμένα δίκτυα. Όπως έχουμε προαναφέρει επίσης ο αλγόριθμος αυτός θα πρέπει να μπορεί να συνενώσει με ομαλό τρόπο τόσο reactive όσο και proactive χαρακτηριστικά τα οποία θα του δώσουν την αναμενόμενη απόδοση για τέτοιου είδους δίκτυα.

Βιβλιογραφία

- [1] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks, Ad Hoc Networks", Volume 2, Issue 1, January 2004, Pages 1-22, ISSN 1570-8705, DOI: 10.1016/S1570-87.
- [2] H.AhleHagh, W.R.Michalson, and D.Finkel, "Statistical characteristics of wireless network traffic and its impact on ad hoc network performance", Advanced Simulation Technologies Conference April 2003, Orlando, USA, pp. 66-71.
- [3] M. Akhlaq, M. Jafri, N. Khan, and B. Aslam, " Addressing Security Concerns of Data Exchange in AODV Protocol ", World Academy of Science, Engineering and Technology 2006.
- [4] S. H. Amer, and J. A. Hamilton, "DSR and TORA in Fixed and Mobile Wireless Networks" , in *Proceedings of the 46th Annual Southeast Regional Conference on XX*, March 2008.
- [5] S.H. Amer, and John A. Hamilton, "SOURCE-INITIATED ON-DEMAND ROUTING PROTOCOLS PERFORMANCE EVALUATION USING OPNET SIMULATION" .
- [6] G. Barua, and M. Agarwal, "Caching of Routes in Ad hoc On-Demand Distane Vector Routing for Mobile Ad hoc Networks", in *Proceedings of the 15th international conference on Computer communication*, August 2002.
- [7] M. Boban, G. Misek, and O. Tonguz, "What is the best achievable QoS for unicast routing in VANETs?", in *The 3rd IEEE Workshop on Automotive Networking and Applications (AutoNet)*, 2008, pp. 1–10.

- [8] A. Cavalli, and C.G. Maag, "A Validation Model for the DSR protocol ", in *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops - W7: EC (ICDCSW'04)* - Volume 7, March 2004.
- [9] T. Chen, O. Mehani, and R. Boreli, "Trusted routing for VANET", in *ITST 2009, 9th International Conference on Intelligent Transport Systems Telecommunications*, M. Berbineau, M. Itami, and G. Wen, Eds. Piscataway, NJ, USA: IEEE Computer Society, October 2009, pp. 647–652.
- [10] J. Chennikara-Varghese, Wai Chen, O. Altintas, and Shengwei Cai, "Survey of Routing Protocols for Inter-Vehicle Communications", in *Mobile and Ubiquitous Systems, Annual International Conference on*, pp. 1-5, 2006 3rd Annual International Conference on Mobile and Ubiquitous Systems - Workshops, 2006.
- [11] M. C. Chuah, and F. Fu, "Performance Study of Robust Data Transfer Protocol for VANETs", LNCS - Springer, Vol 4325, pp. 377-39, 2006.
- [12] M. Dikaiakos, A. Florides, T. Nadeem, and L. Iftode, "Location-aware Services over Vehicular Ad-Hoc Networks using Car-to-Car Communication" , IEEE journal on selected areas in communications, vol. 25, no. 8, October 2007.
- [13] V. Park and S. Corson, "Temporally-ordered routing algorithm (tora) version 1," Internet-Draft, draft-ietf-manet-tora-spec-04.txt, July 2001, work in progress.
- [14] T. Clausen, and P. Jaquet, "Optimized Link State Routing Protocol(OLSR)", <http://www.rfc-editor.org/rfc/rfc3626.txt>, October 2007.
- [15] A. Ergawy, "Routing Protocols for Wireless Ad Hoc Wireless Networks: A Review of the On-Demand Routing Protocols".

[16] J.Farooq, "Mobility and Internet Connectivity in Mobile Ad Hoc Networks" , in *Umeå's 9th Students Conference in Computing Science*, June 2005.

[17] Krisha Gorantala, "Routing Protocols in Mobile Ad-hoc Networks", M.S. Thesis in Computing Science at CS-UmU, June 2006.

[18] J. Haerri, F. Filali, C. Bonnet, "Performance Comparison of AODV and OLSR in VANETs Urban Environments under Realistic Mobility Patterns" , in *5th IFIP Mediterranean Ad-Hoc Networking Workshop*, June 14-17, 2006, Lipari, Italy.

[19] M.Hoda, U.Singh, Milind,R. Kumar,"Simulation of AODV, DSR and DSDV Routing Protocols for Mobile Ad-hoc Networks", in *Proceedings of the 2nd National Conference NDIACom-2008*.

[20]N.Islam, Z.A. Shaikh, S Talpur, "Towards a Grid-based approach to Traffic Routing in VANET" , Institute of Industrial and Electronics Engineering, Karachi, Pakistan, E-Indus 2008, 29th April, 2008.

[21] S. Jaap, M. Bechler, and L. Wolf, "Evaluation of Routing Protocols for Vehicular Ad Hoc Networks in Typical Road Traffic Scenarios", in *Proceedongs of the 11th EUNICE Open European Summer Schoolon Networked Applications*, Colmenarejo, Spain, July 2005.

[22] D. Johnson, Y. Hu, and D.Maltz, "The Dynamic Source Routing Protocol (DSR)", <http://www.rfc-editor.org/rfc/rfc4728.txt>, February 2007.

[23] M. Kafsi, O. Dousse, P. Papadimitratos, T. Alpcan, and J.-P. Hubaux. VANET Connectivity Analysis. Technical report, EPFL/T-Labs, 2008.

[24] Z. Zhou, "A survey on routing protocols in MANETs", Technical report, Department of Computer Sciences, Michigan State University, MSUCSE-03-8, 2003.

[25] J.Li, and C. Chigan, "Achieving Robust Message Dissemination in VANET: Challenges and Solutions ".

[26] T. Little, and A. Agarwal, "An information Propagation Scheme for VANETs", in *Proceedings 8th International IEEE Conference on Intelligent Transportation Systems (ITSC2005)*, Vienna, Austria, September, 2005.

[27] A. Mahajan, N. Potnis, K. Gopalan, and A-I. Wang, "Urban Mobility Models for VANETs" , in *Proceedings of 2nd Workshop on Next Generation Wireless Networks*, 2006.

[28] M. Zhang, and R. Wolff, "Routing Protocols for Vehicular Ad Hoc Networks in Rural Areas", *IEEE Communications Magazine*, 2008, 46(11):126-131.

[29] R. Misra, and C. R. Mandal, "Performance Comparison of AODV/DSR On-demand Routing Protocols for Ad Hoc Networks in Constrained Situation", in *Proceedings of 7th IEEE International Conference on Personal Wireless Communications (ICPWC-2005)*, New Delhi, India, pp 86-89, January 23-25, 2005.

[30] T. Nadeem, P. Shankar, and L. Iftode, "A Comparative Study of Data Dissemination Models for VANETs", University of Maryland Technical Report, CS-TR-4810, 2006.

[31] V. Nundoll, G.S. Blair, P.Grace, "A Component-based Approach For (Re)-Configurable Routing in Vanets", in *8th International Workshop on Adaptive and Reflective Middleware (ARM 2009)*, Illinois, USA, December 2009.

[32] Charles E. Perkins, and Elizabeth M. Royer, "Ad-hoc On-Demand Distance Vector Routing," *Mobile Computing Systems and Applications*, IEEE Workshop on, p. 90, Second IEEE Workshop on Mobile Computer Systems and Applications, 1999.

- [33] Elizabeth M. Royer, and Chai -Keong Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks", IEEE Personal Communications, Vol. 6, No. 2, pp. 46-55, April 1999.
- [34] K.Sadasivam, V. Changrani, and T.A. Yang, "SCENARIO BASED PERFORMANCE EVALUATION OF SECURE ROUTING IN MANETs" .
- [35] S. Sesay, Z. Yang, Z. B. Qi, H. Jianhua, "Simulation Comparison of Four Wireless Ad hoc Routing Protocols ", Information Technology Journal 3 (3): 219-226, 2004, ISSN 1682-6027.
- [36] S.R.Thampuran, "Routing Protocols for Ad Hoc Networks of Mobile Nodes" .
- [37] X.Wei, and L. Qing-Quan, "PERFORMANCE EVALUATION OF DATA DISSEMINATIONS FOR VEHICULAR AD HOC NETWORKS IN HIGHWAY SCENARIOS" .
- [38] S-A. Yang, and J.S Baras, "TORA, Verification, Proofs and Model Checking" , March 2003.
- [39] S. Yousefi, and M. Fathy, "METRICS FOR PERFORMANCE EVALUATION OF SAFETY APPLICATIONS IN VEHICULAR AD HOC NETWORKS" , Transport 22(4): 284–289, October, 2008.
- [40] A. Zaballos, A. Vallejo, G. Corral, and J. Abella, "Ad Hoc routing performance study using OPNET Modeler," presented at the OPNETWORK, Washington DC (United States) August, 2006.

Παράρτημα Α

Στο παράρτημα Α θα αναφέρουμε και θα παρουσιάσουμε τις ρυθμίσεις με τις οποίες έτρεξε ο κάθε αλγόριθμος σε κάθε σενάριο.

Σενάριο Light

Αλγόριθμος AODV

[-] AD-HOC Routing Parameters	
... AD-HOC Routing Protocol	AODV
[-] AODV Parameters	(...)
[-] Route Discovery Parameters	(...)
... Route Request Retries	5
... Route Request Rate Limit (pkts/sec)	10
... Gratuitous Route Reply Flag	Disabled
... Destination Only Flag	Disabled
... Acknowledgement Required	Disabled
... Active Route Timeout (seconds)	3
... Hello Interval (seconds)	uniform (1, 1.1)
... Allowed Hello Loss	2
... Net Diameter	35
... Node Traversal Time (seconds)	0.04
... Route Error Rate Limit (pkts/sec)	10
... Timeout Buffer	2
[-] TTL Parameters	(...)
... TTL Start	1
... TTL Increment	2
... TTL Threshold	7
... Local Add TTL	2
... Packet Queue Size (packets)	Infinity
... Local Repair	Enabled
... Addressing Mode	IPv4

Αλγόριθμος DSR

[-] AD-HOC Routing Parameters	
[-] AD-HOC Routing Protocol	DSR
[+] AODV Parameters	Default
[-] DSR Parameters	(...)
[-] Route Cache Parameters	(...)
[-] Max Cached Routes	Infinity
[-] Route Expiry Timer (seconds)	300
[-] Route Cache Export	(...)
[-] Status	Do Not Export
[-] Export Time(s) Specification	(...)
[-] Number of Rows	1
[-] End of simulation	
[-] Time (seconds)	End of simulation
[-] Send Buffer Parameters	(...)
[-] Max Buffer Size (packets)	Infinity
[-] Expiry Timer (seconds)	30
[-] Route Discovery Parameters	(...)
[-] Request Table Size (nodes)	64
[-] Maximum Request Table Identifiers (identifiers)	16
[-] Maximum Request Retransmissions (retransmissions)	16
[-] Maximum Request Period (seconds)	10
[-] Initial Request Period (seconds)	0.5
[-] Non Propagating Request Timer (seconds)	0.03
[-] Gratuitous Route Reply Timer (seconds)	1
[-] Route Maintenance Parameters	(...)
[-] Maximum Buffer Size (packets)	50
[-] Maintenance Holdoff Time (seconds)	0.25
[-] Maximum Maintenance Retransmissions (retransmissions)	2
[-] Maintenance Acknowledgement Timer (seconds)	0.5
[-] DSR Routes Export	Do Not Export
[-] Route Replies using Cached Routes	Enabled
[-] Packet Salvaging	Enabled
[-] Non Propagating Request	Disabled
[-] Broadcast Jitter (seconds)	uniform (0, 0.01)

Αλγόριθμος TORA

[-] TORA/IMEP Parameters	(...)
Router ID	Auto Assigned
[-] TORA Parameters	(...)
Mode of Operation	On-Demand
OPT Transmit Interval (seconds)	300
IP Packet Discard Timeout (seco...	10
[-] IMEP Parameters	(...)
Beacon Period (seconds)	20
Max Beacon Timer (seconds)	60
Max Retries (number of attempts)	3
Max IMEP Packet Length (bytes)	1,500
Route Injection	Disabled

Αλγόριθμος OLSR

[-] OLSR Parameters	(...)
Willingness	Willingness Default
Hello Interval (seconds)	2.0
TC Interval (seconds)	5.0
Neighbor Hold Time (seconds)	6.0
Topology Hold Time (seconds)	15.0
Duplicate Message Hold Time (sec...	30.0
Addressing Mode	IPv4
[+] TORA/IMEP Parameters	Default

Αλγόριθμος GRP

[-] GRP Parameters	(...)
Hello Interval (seconds)	uniform (4.9, 5.0)
Neighbor Expiry Time (seconds)	constant (10)
[-] Position Update Parameters	(...)
Distance moved (metres)	1000
Position Request Timer (seconds)	5.0
Backtrack Option	Enabled
Routes Export	Disabled
Number of Initial Floods	1

Σενάριο Medium

Στο σενάριο medium χρησιμοποιήσαμε τις ίδιες παραμέτρους εκτός από τον αλγόριθμο OLSR όπου χρησιμοποιήσαμε τις εξής παραμέτρους:

[-] OLSR Parameters	(...)
Willingness	Willingness Default
Hello Interval (seconds)	7.0
TC Interval (seconds)	9.0
Neighbor Hold Time (seconds)	6.0
Topology Hold Time (seconds)	15.0
Duplicate Message Hold Time (sec...	30.0
Addressing Mode	IPv4

Σενάριο Heavy

Στο σενάριο heavy χρησιμοποιήσαμε τις ίδιες παραμέτρους εκτός από τον αλγόριθμο OLSR όπου χρησιμοποιήσαμε τις εξής παραμέτρους:

[-] AD-HOC Routing Parameters	
AD-HOC Routing Protocol	OLSR
[+] AODV Parameters	Default
[+] DSR Parameters	Default
[+] GRP Parameters	Default
[-] OLSR Parameters	(...)
Willingness	Willingness Default
Hello Interval (seconds)	12
TC Interval (seconds)	14
Neighbor Hold Time (seconds)	6.0
Topology Hold Time (seconds)	15.0
Duplicate Message Hold Time (sec...	30.0
Addressing Mode	IPv4

Παράρτημα Β

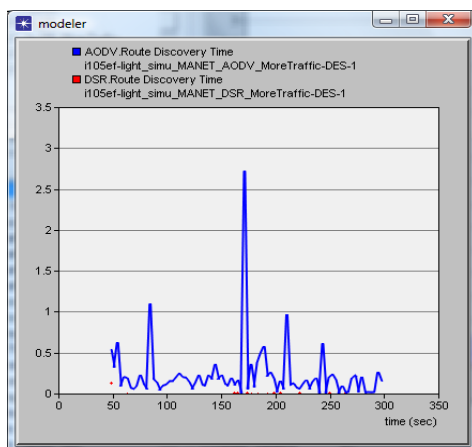
Στο παράρτημα Β παρουσιάζουμε τις γραφικές παραστάσεις τις οποίες επιλέξαμε να μην συμπεριλάβουμε στο κυρίως μέρος της ΑΔΕ. Θα παρουσιάσουμε τα αποτελέσματα τα οποία πήραμε χρησιμοποιώντας περισσότερο traffic (10 και 15 κόμβους να στέλνουν συνεχή πληροφορία). Τέλος θα παρουσιάσουμε την σύγκριση μεταξύ των σεναρίων όπου θα δούμε γραφικές παραστάσεις οι οποίες αφορούν σενάρια με όλων των ειδών τα traffic τα οποία τρέξαμε.

1.Σενάριο με περισσότερο traffic (10 κόμβοι ως συνεχείς αποστολές)

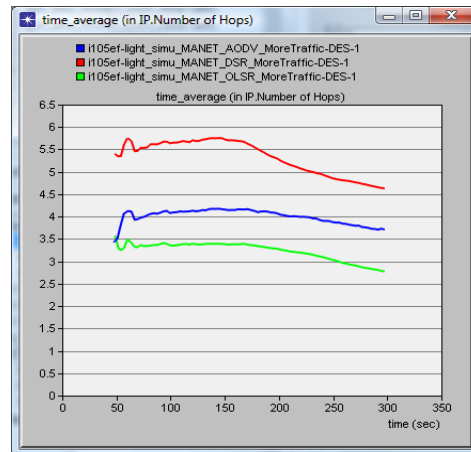
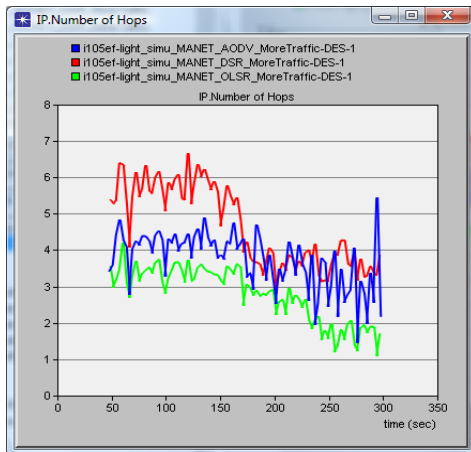
Στο σημείο αυτό θα παρουσιάσουμε τις γραφικές παραστάσεις τις οποίες έχουμε λάβει από τα σενάρια τα οποία τρέξαμε ώστε να φανεί και η ομοιότητα των αποτελεσμάτων τα οποία πήραμε σε σχέση με αυτά τα οποία παρουσιάσαμε στο κύριο μέρος της ΑΔΕ.

Σενάριο Light

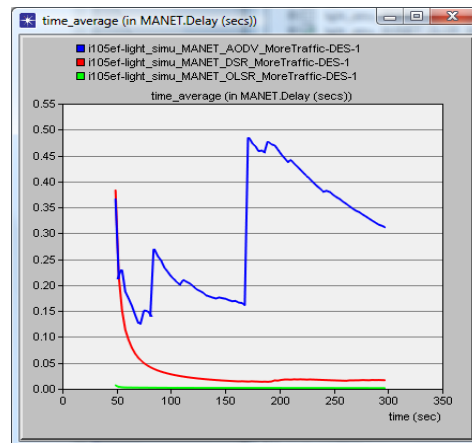
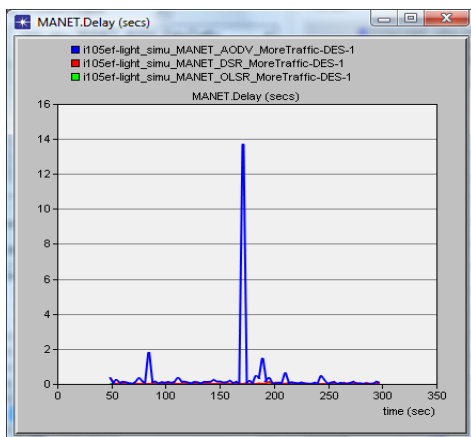
Route Discovery Time (AODV, DSR)



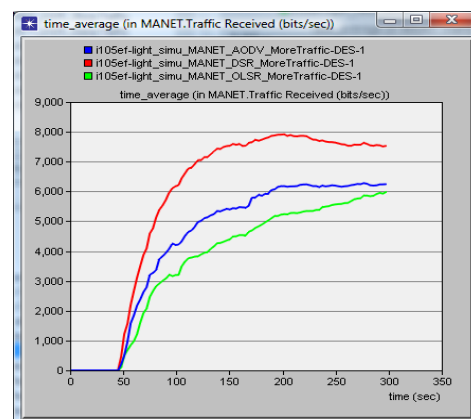
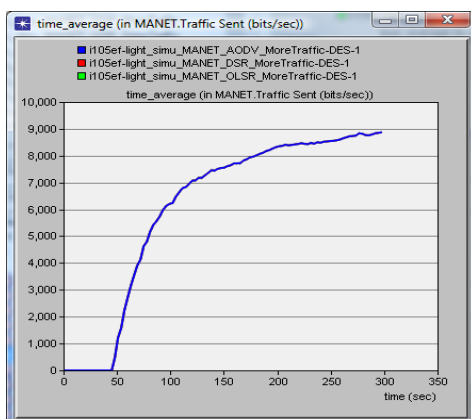
IP.number of Hops (AODV, DSR, OLSR)



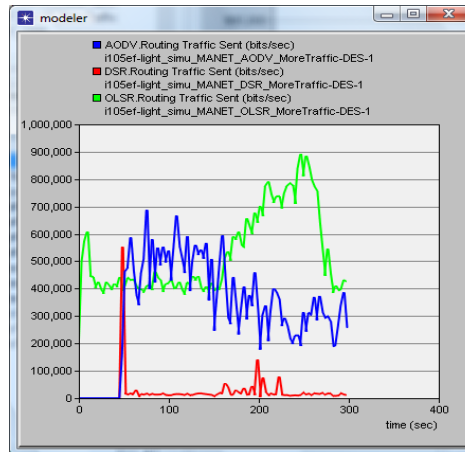
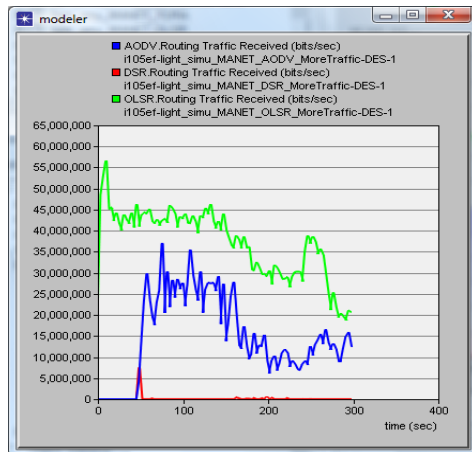
MANET. Delay (AODV, DSR, OLSR)



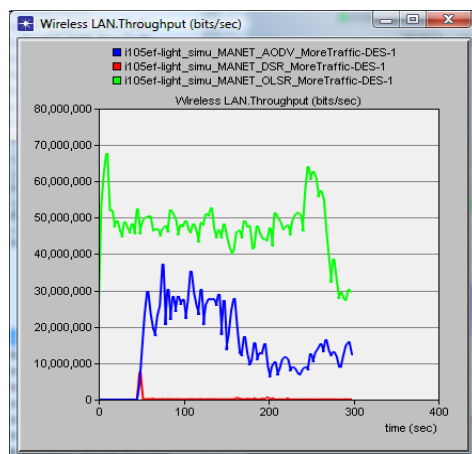
MANET. Traffic Received/Sent bits/second) (time average) (AODV, DSR, OLSR)



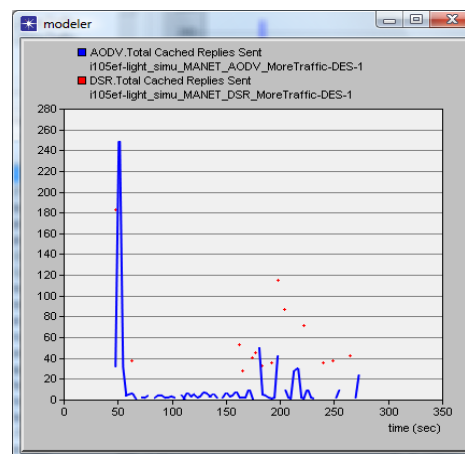
Routing traffic Received/Sent (bits/sec) (AODV, DSR, OLSR)



Wireless LAN Throughput (AODV, DSR, OLSR)

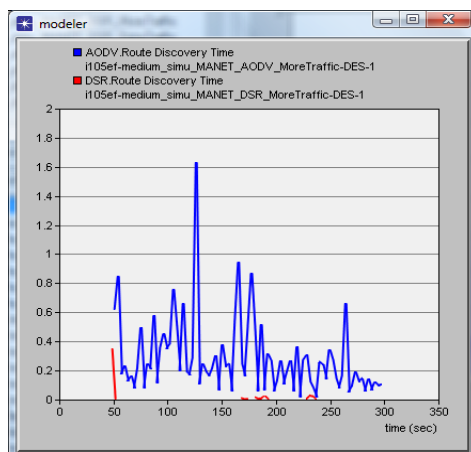


Total Cached Replies Sent (AODV, DSR)

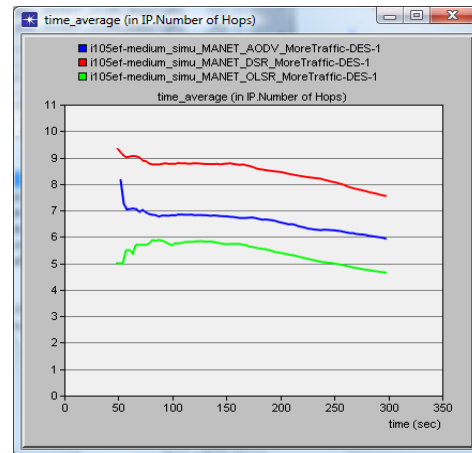
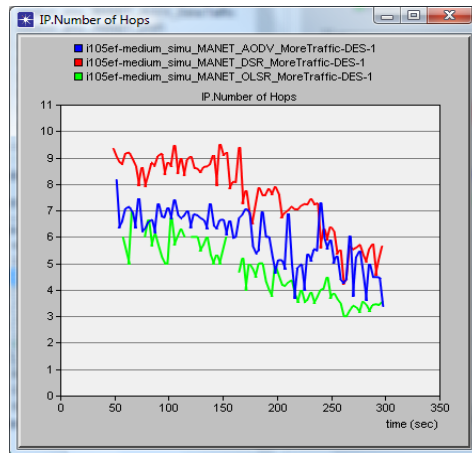


Σενάριο Medium

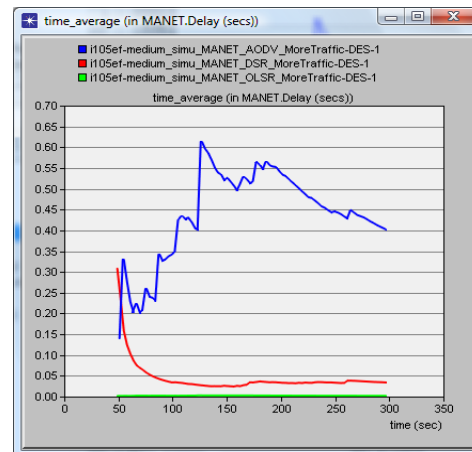
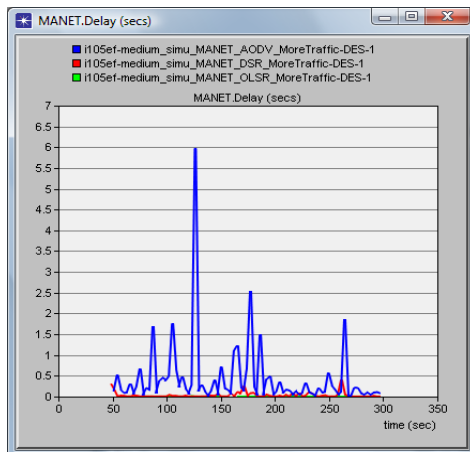
Route Discovery Time (AODV, DSR)



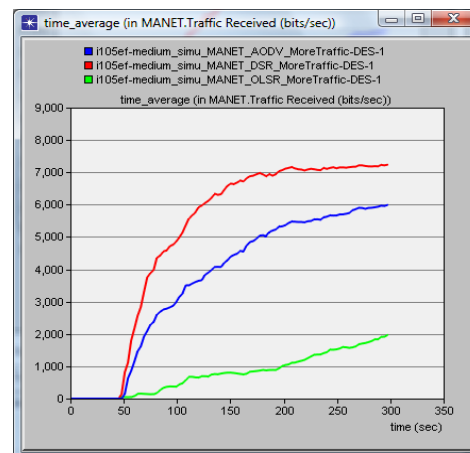
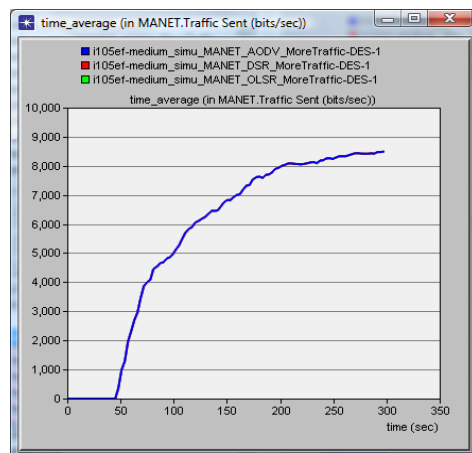
IP.number of Hops (AODV, DSR, OLSR)



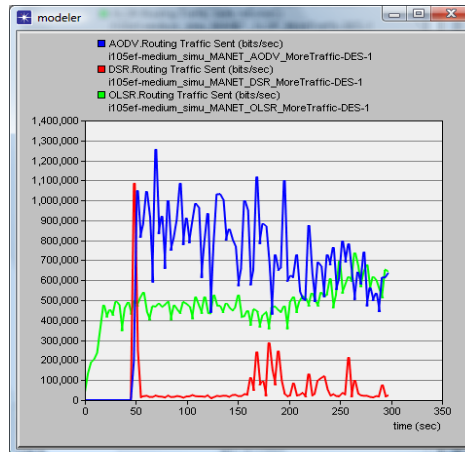
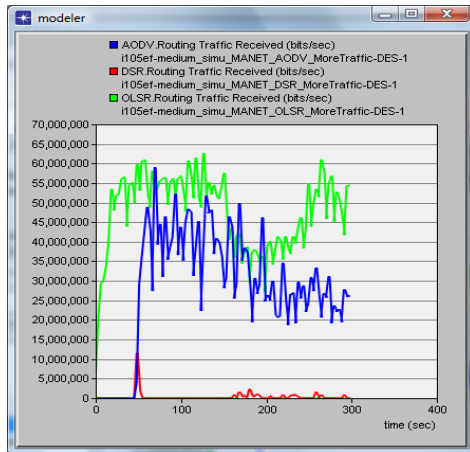
MANET. Delay (AODV, DSR, OLSR)



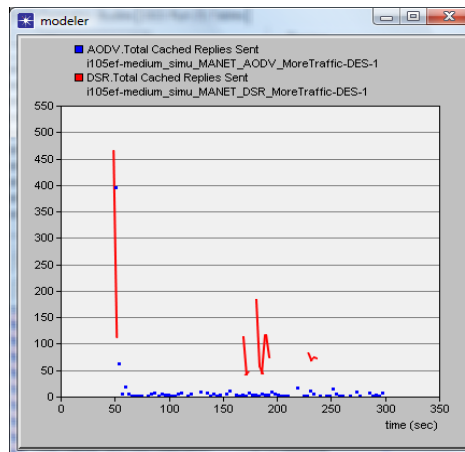
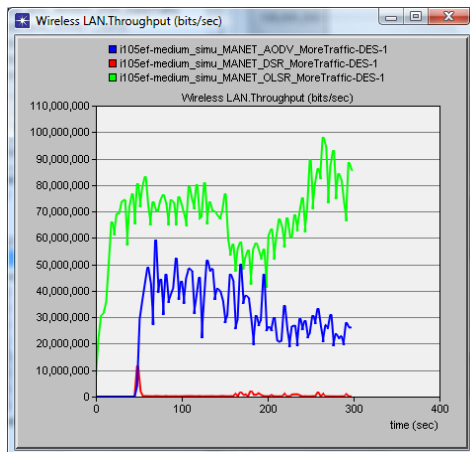
MANET. Traffic Received/Sent (bits/second) (time average) (AODV, DSR, OLSR)



Routing traffic Received/Sent (bits/sec) (AODV, DSR, OLSR)

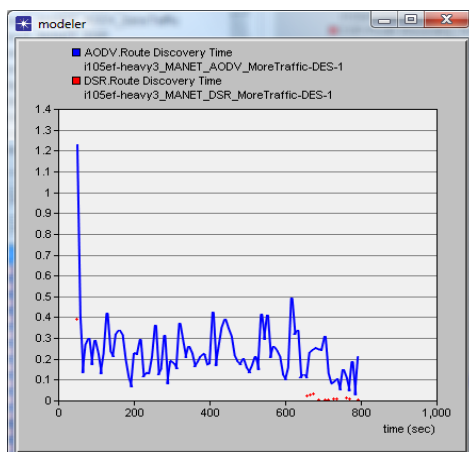


Wireless LAN.Throughput (AODV,DSR,OLSR)

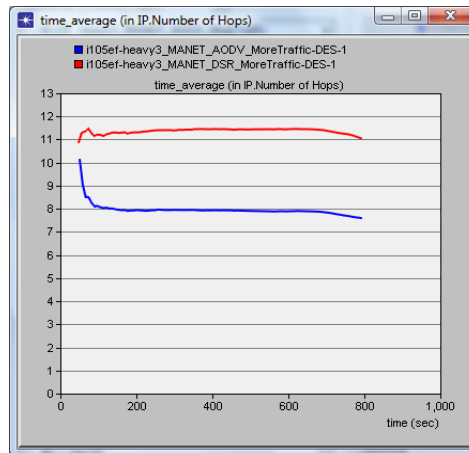
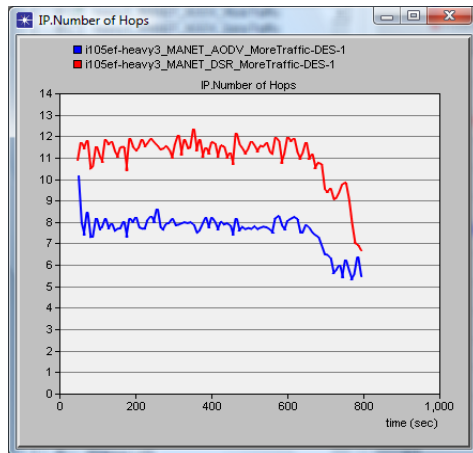


Σενόπιο Heavy

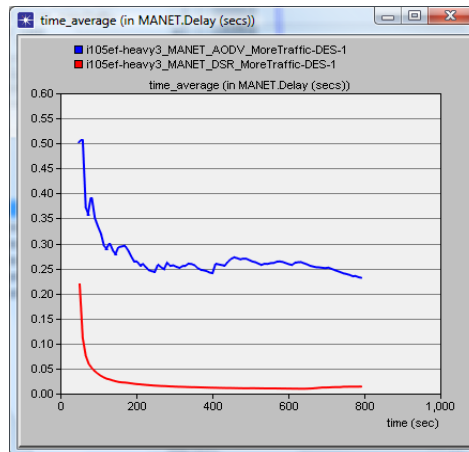
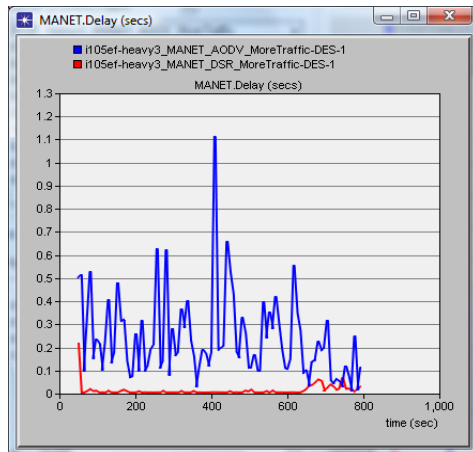
Route Discovery Time (AODV, DSR)



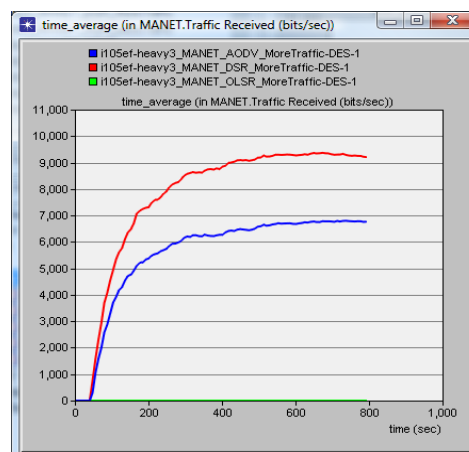
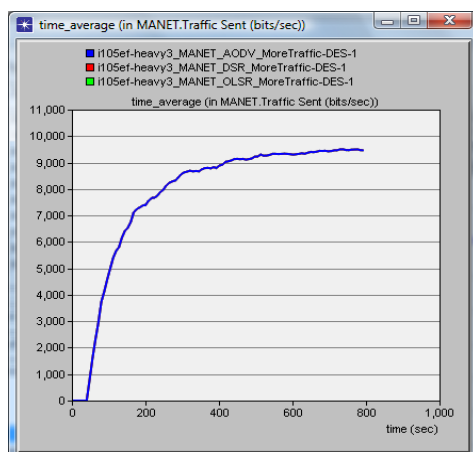
IP.number of Hops (AODV, DSR, OLSR)



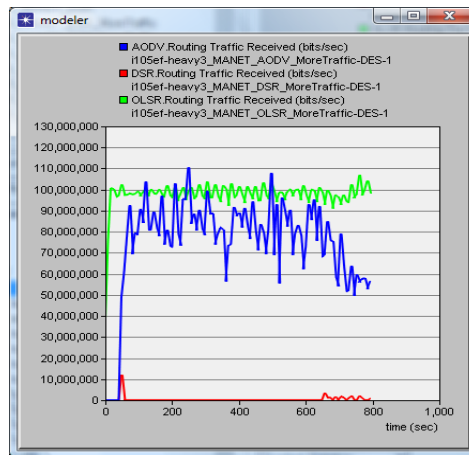
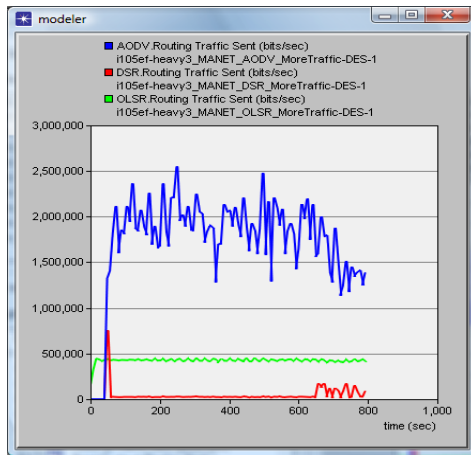
MANET. Delay (AODV, DSR, OLSR)



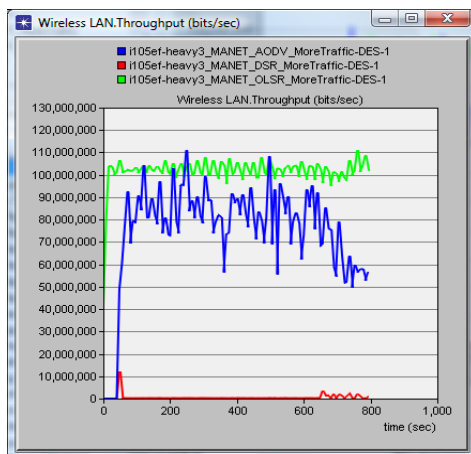
MANET. Traffic Received/Sent (bits/second) (time average) (AODV, DSR, OLSR)



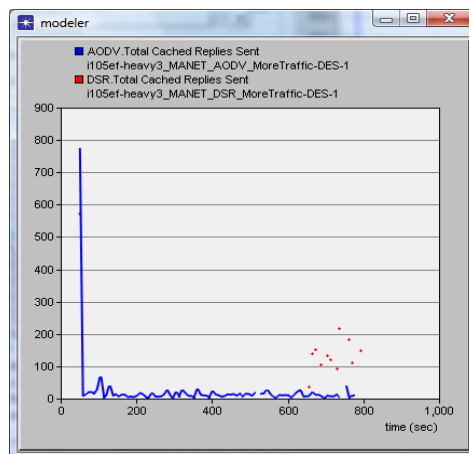
Routing traffic Received/Sent (bits/sec) (AODV, DSR, OLSR)



Wireless LAN Throughput (AODV, DSR, OLSR)



Total Cached Replies Sent (AODV, DSR)

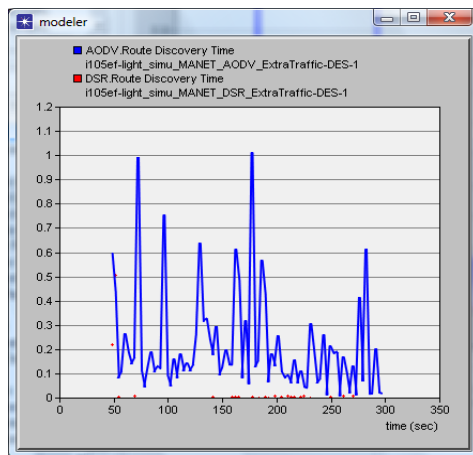


2. Σενάρια με ακόμη περισσότερο traffic (15 κόμβοι ως αποστολείς)

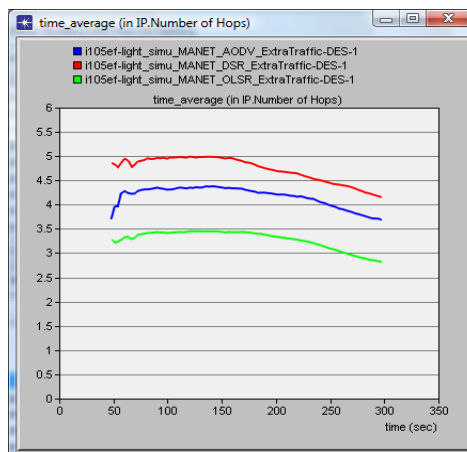
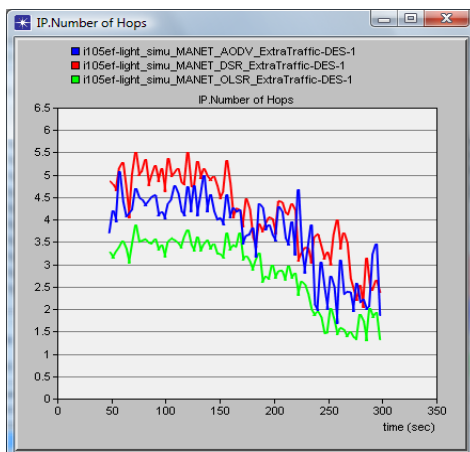
Εδώ θα παρουσιάσουμε τα αποτελέσματα τα οποία λάβαμε όταν τρέξαμε τους αλγόριθμους με ακόμη περισσότερο traffic ώστε να φανεί η ομοιότητα με τα αποτελέσματα τα οποία παρουσιάσαμε στο κυρίως μέρος της ΑΔΕ.

Light Σενάριο

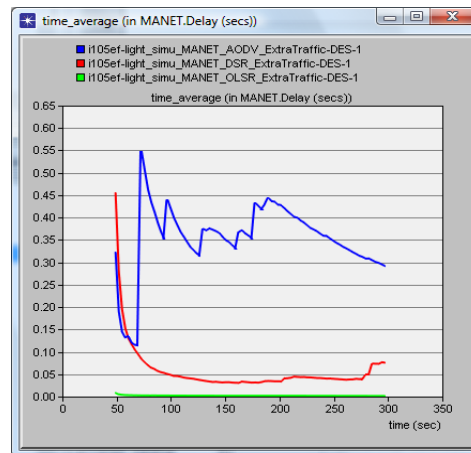
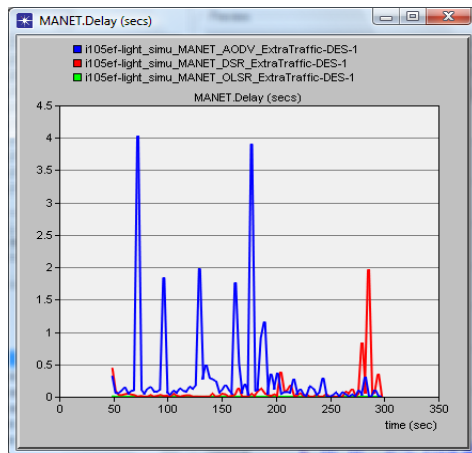
Route Discovery Time (AODV, DSR)



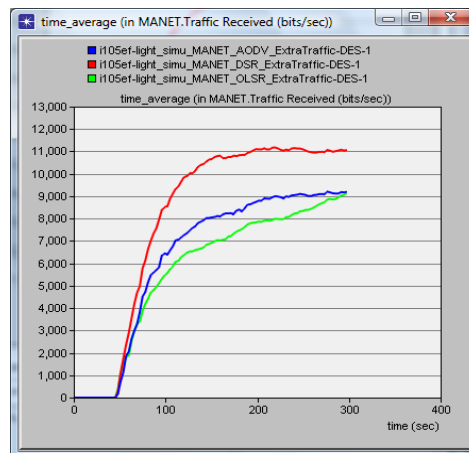
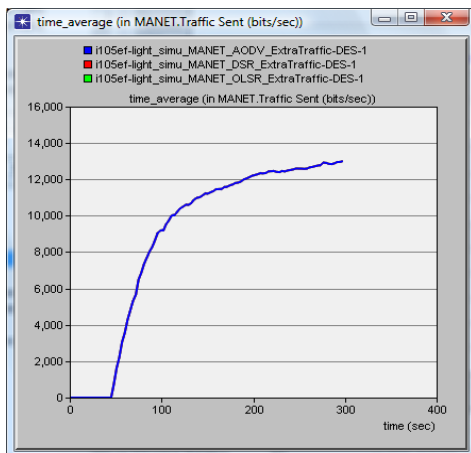
IP.number of Hops (AODV, DSR, OLSR)



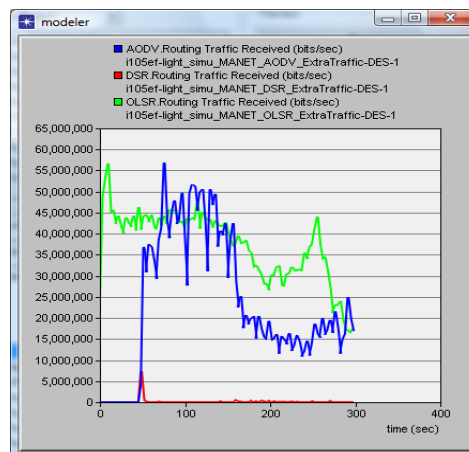
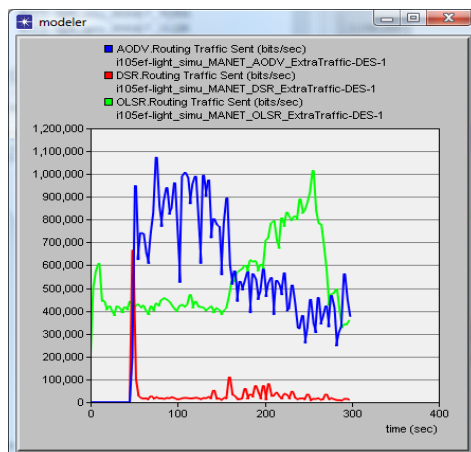
MANET. Delay (AODV, DSR, OLSR)



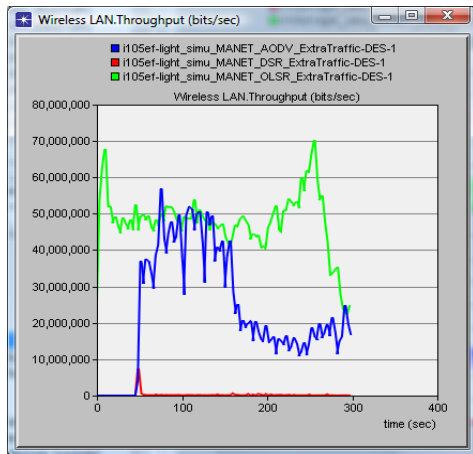
MANET. Traffic Received/Sent (bits/second) (time average) (AODV, DSR, OLSR)



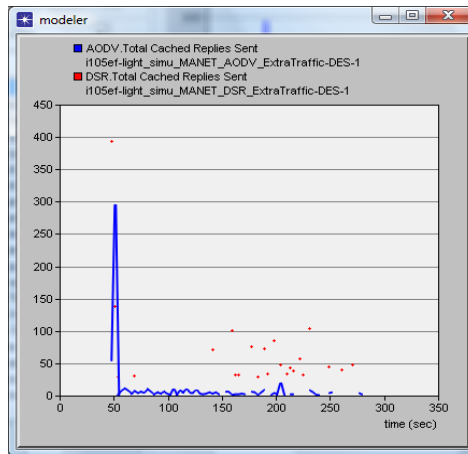
Routing traffic Received/Sent (bits/sec) (AODV, DSR, OLSR)



Wireless LAN Throughput (AODV, DSR, OLSR)

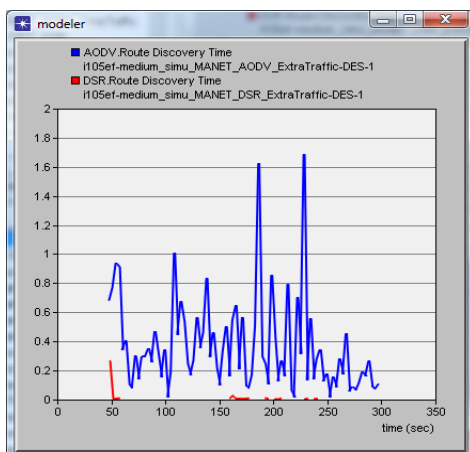


Total Cached Replies Sent (AODV, DSR)

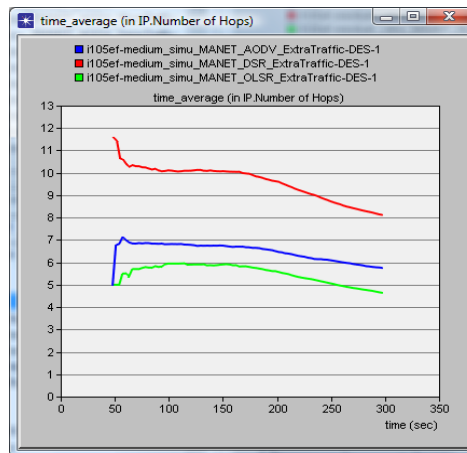
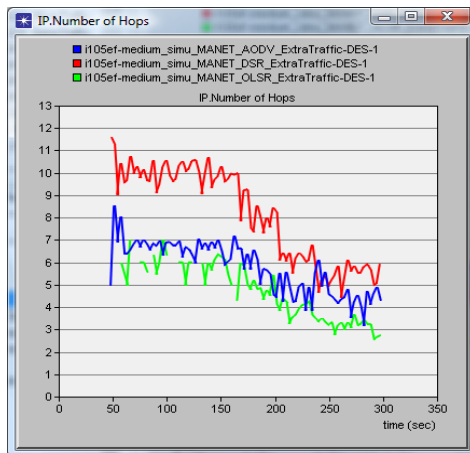


Medium Σενάριο

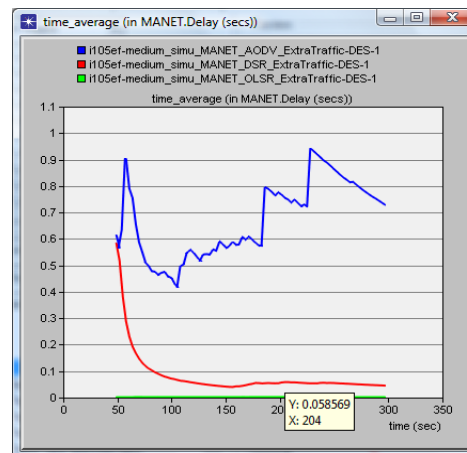
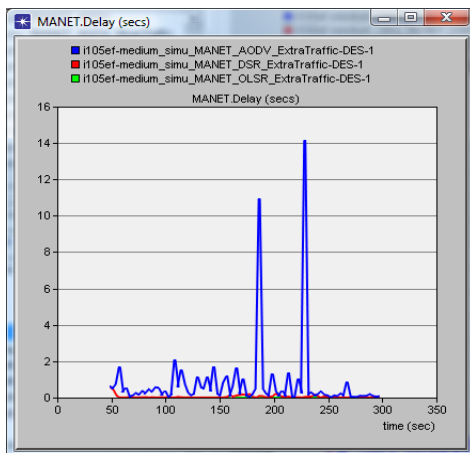
Route Discovery Time (AODV, DSR)



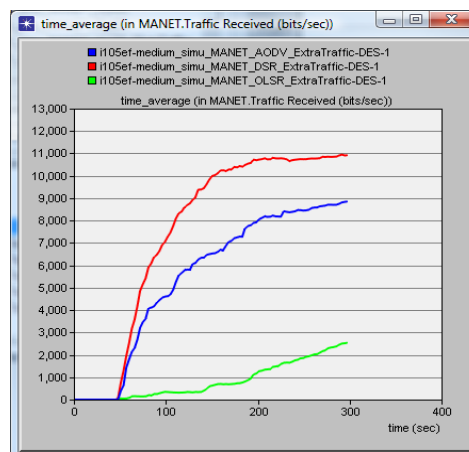
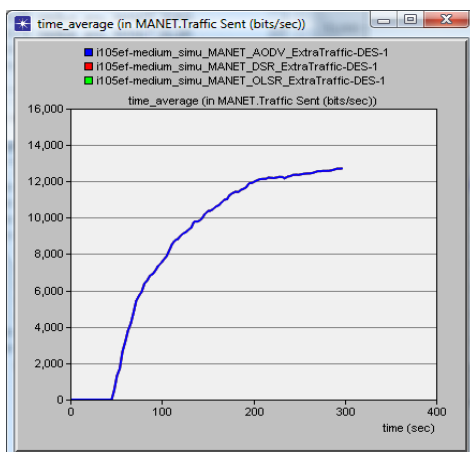
IP.number of Hops (AODV, DSR, OLSR)



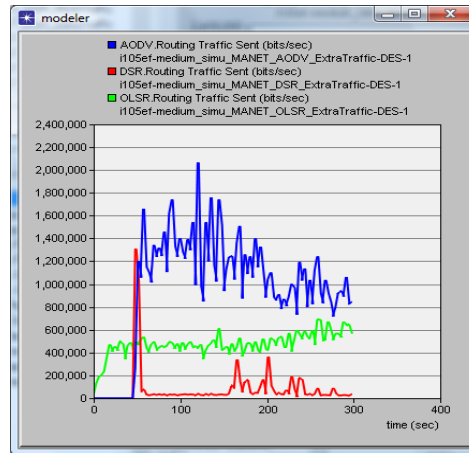
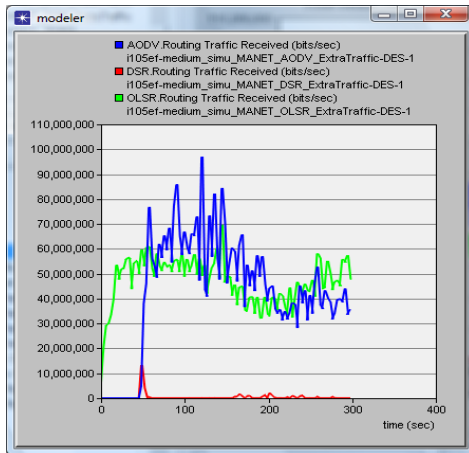
MANET. Delay (AODV, DSR, OLSR)



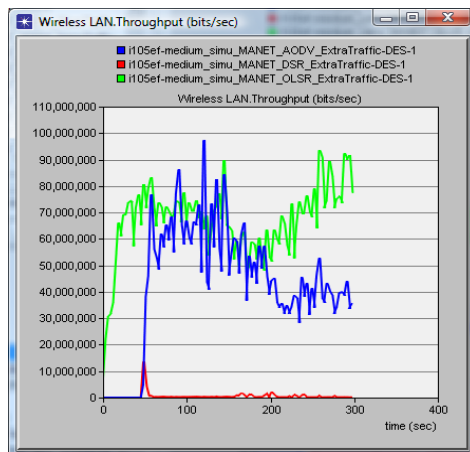
MANET. Traffic Received/Sent (bits/second) (time average) (AODV, DSR, OLSR)



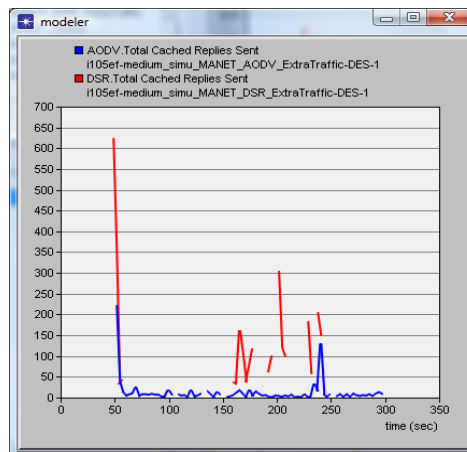
Routing traffic Received/Sent (bits/sec) (AODV, DSR, OLSR)



Wireless LAN.Throughput (AODV,DSR,OLSR)

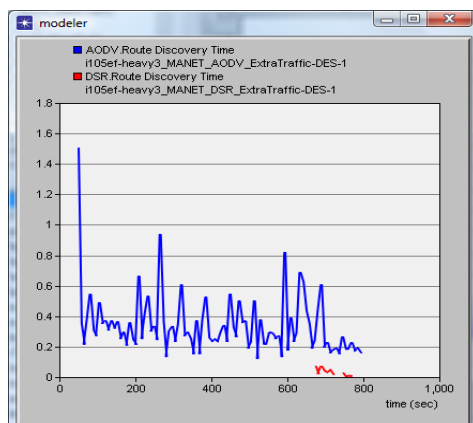


Total Cached Replies Sent (AODV, DSR)

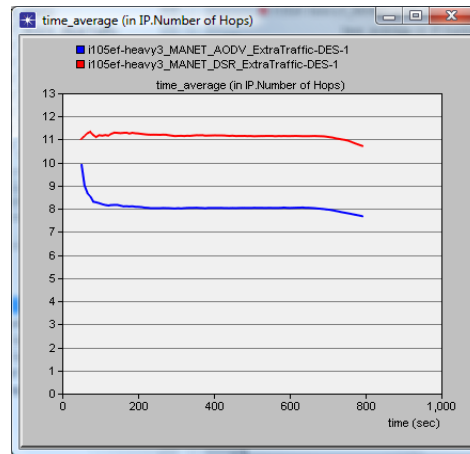


Σενάριο Heavy

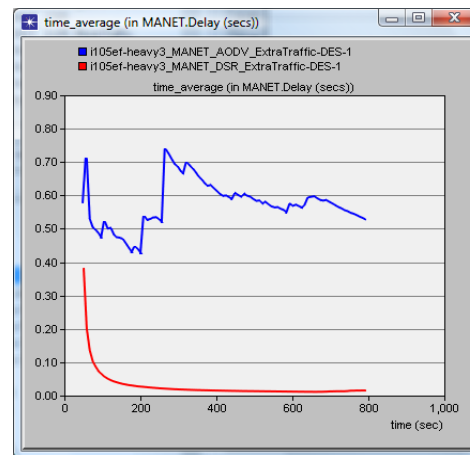
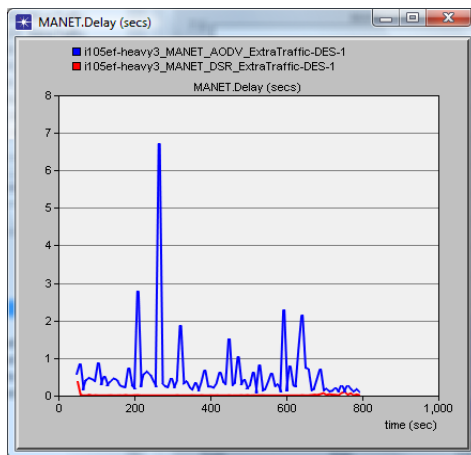
Route Discovery Time (AODV, DSR)



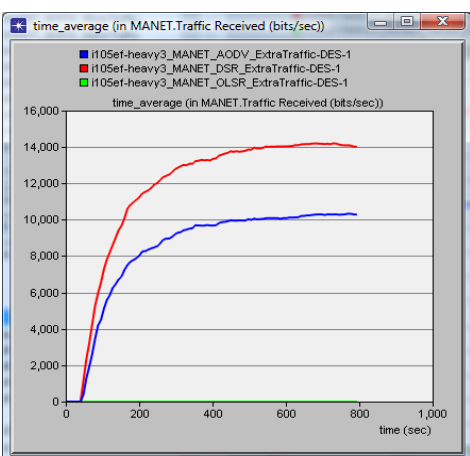
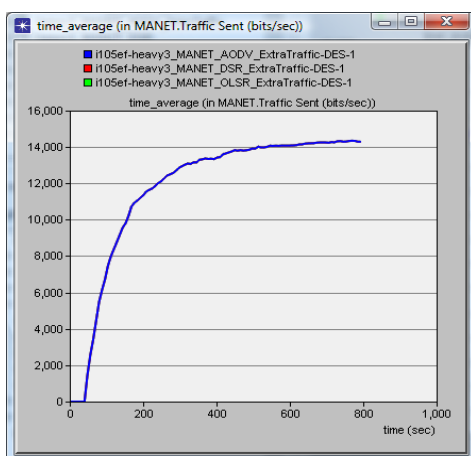
IP.number of Hops (AODV, DSR, OLSR)



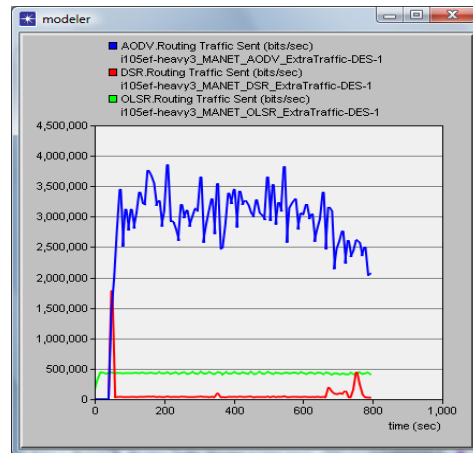
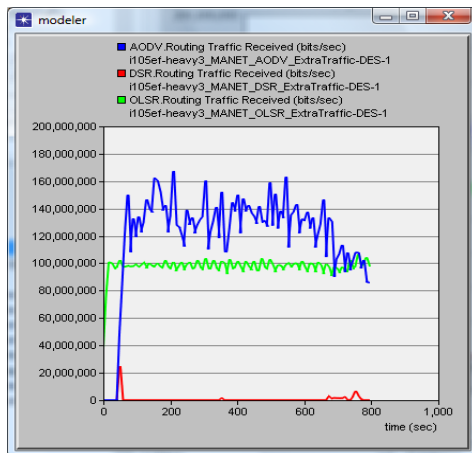
MANET. Delay (AODV, DSR, OLSR)



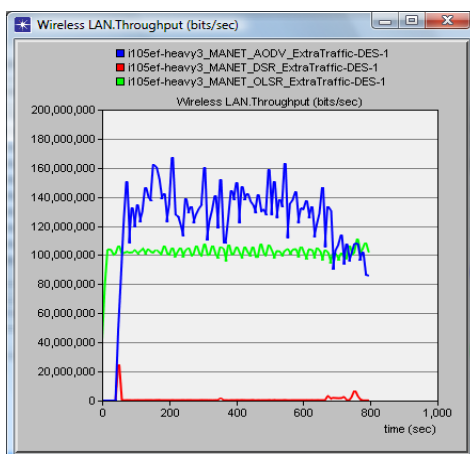
MANET. Traffic Received/Sent (bits/second) (time average) (AODV, DSR, OLSR)



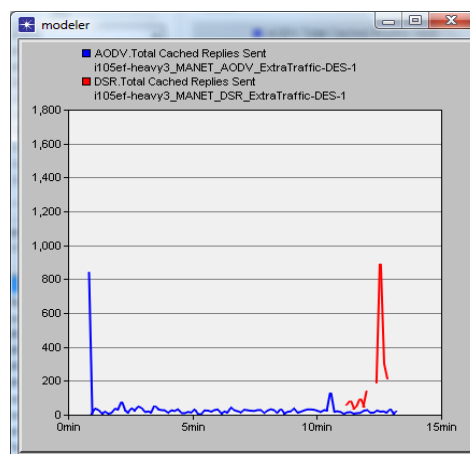
Routing traffic Received/Sent (bits/sec) (AODV, DSR, OLSR)



Wireless LAN.Throughput (AODV,DSR,OLSR)



Total Cached Replies Sent (AODV, DSR)

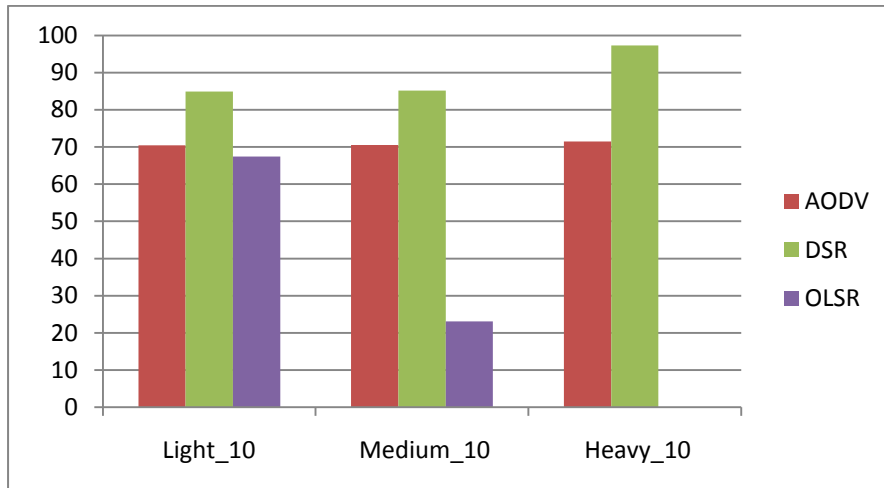


4.Σύγκριση μεταξύ σεναρίων

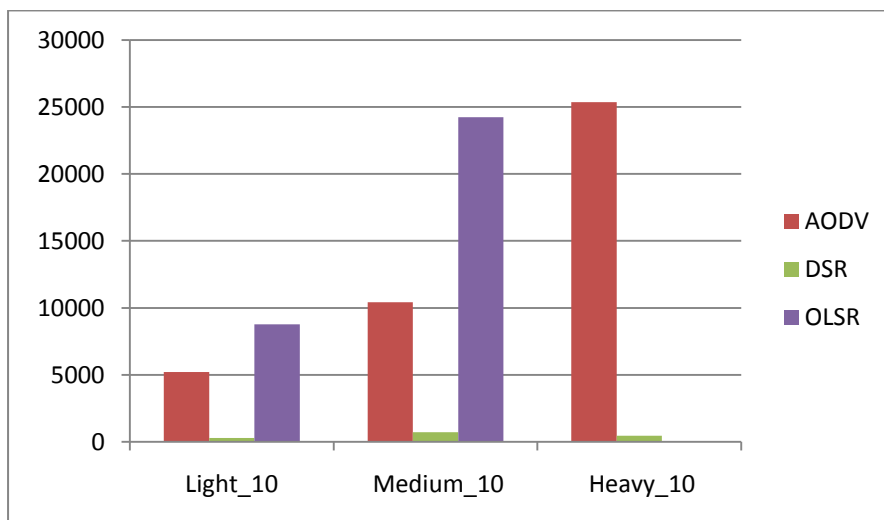
Στο κομμάτι αυτό του παραρτήματος θα παρουσιάσουμε τα αποτελέσματα τα οποία προέκυψαν από την εξαγωγή των scalar δεδομένων στην για τα σενάρια με περισσότερο traffic από αυτό το οποίο αναλύθηκε στο κυρίως μέρος της ΑΔΕ. Επίσης θα δούμε και κάποιες γραφικές παραστάσεις οι οποίες αναλύουν την απόδοση του κάθε αλγόριθμου όταν χρησιμοποιείται περισσότερο traffic.

Σενάριο με 10 κόμβους ως αποστολείς

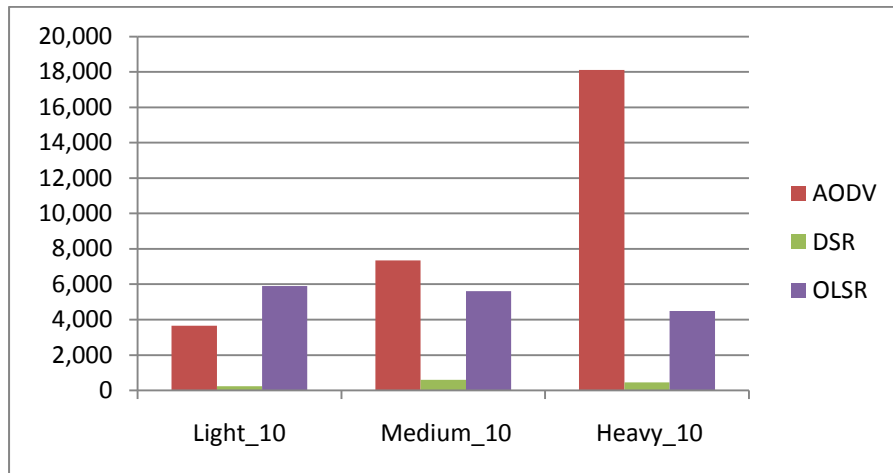
Packet Delivery Ratio (%) (AODV, DSR, OLSR)



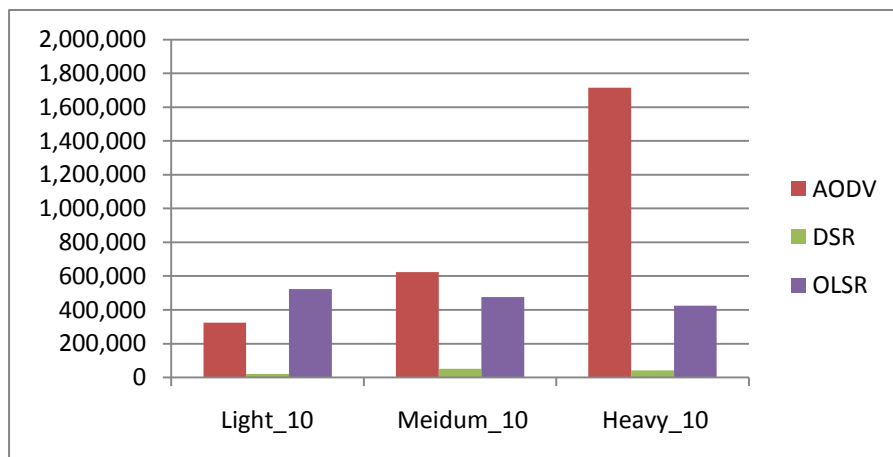
Routing Packet Transmission Ratio (%) (AODV, DSR, OLSR)



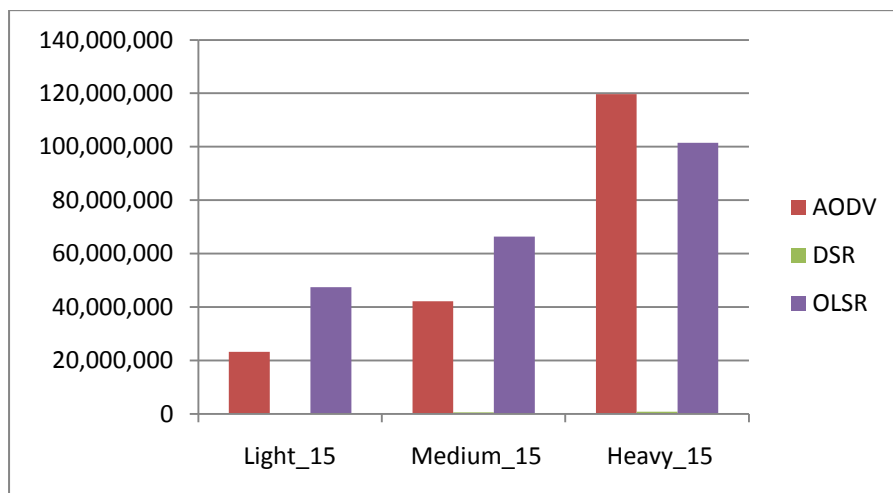
Routing Traffic Sent Vs Manet. Traffic Sent Ratio (%) (AODV, DSR, OLSR)



Routing traffic Sent (mean bits/sec) (AODV, DSR, OLSR)

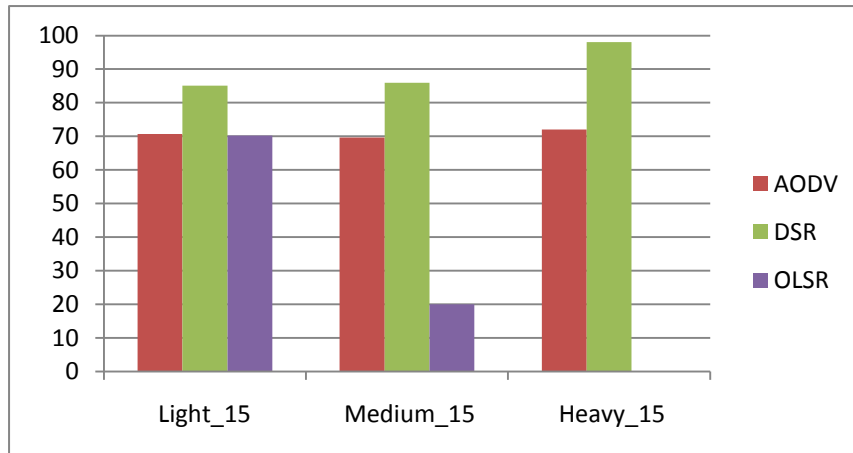


Wireless LAN.Throughput (bits/sec) (AODV, DSR, OLSR)

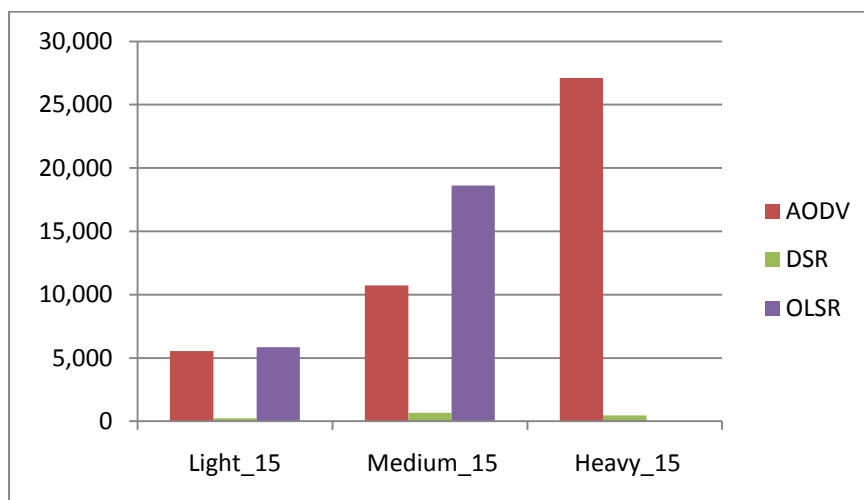


Σενάριο με 15 κόμβους ως αποστολείς

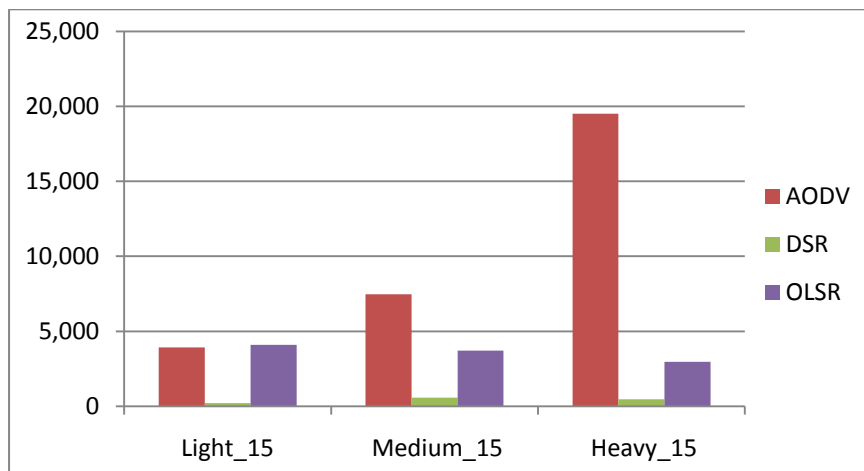
Packet Delivery Ratio (%) (AODV, DSR, OLSR)



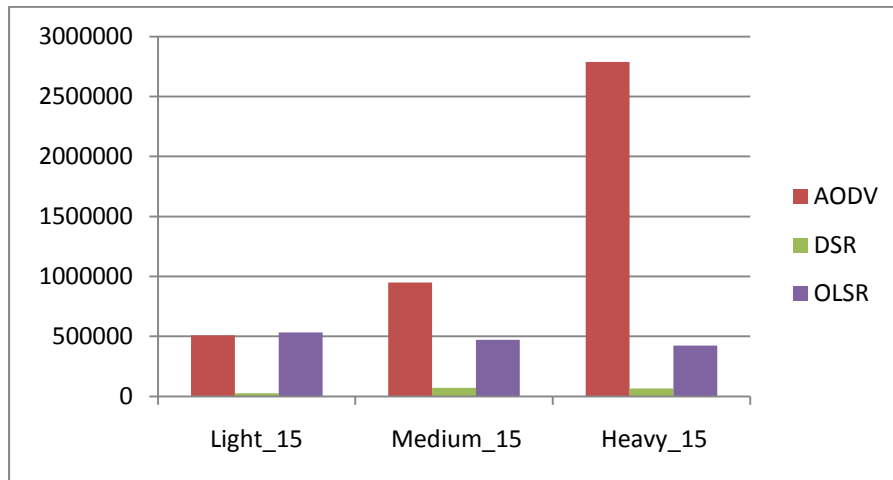
Routing Packet Transmission Ratio (%) (AODV, DSR, OLSR)



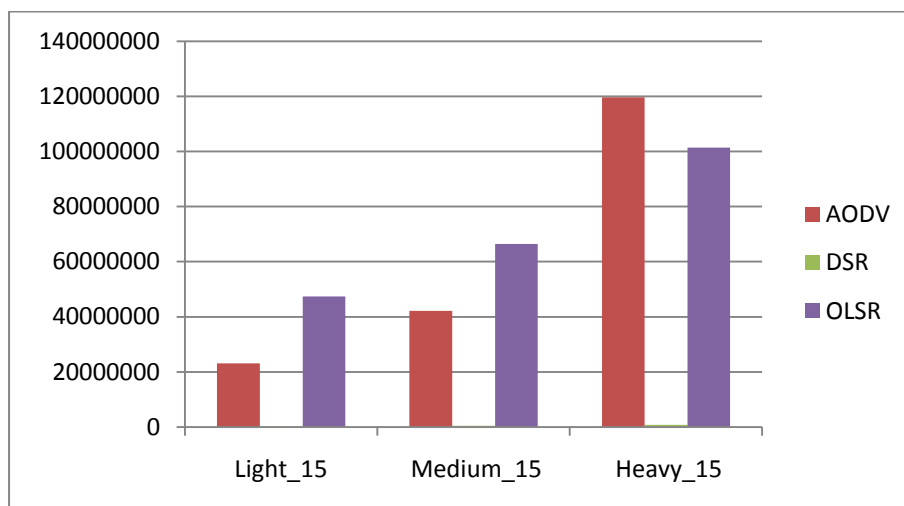
Routing Traffic Sent Vs Manet. Traffic Sent Ratio (%) (AODV, DSR, OLSR)



Routing traffic Sent (mean bits/sec) (AODV, DSR, OLSR)

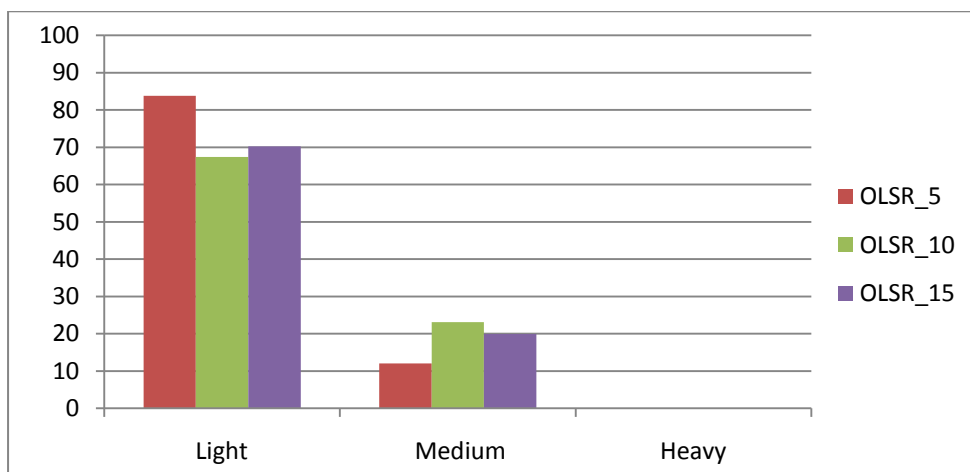
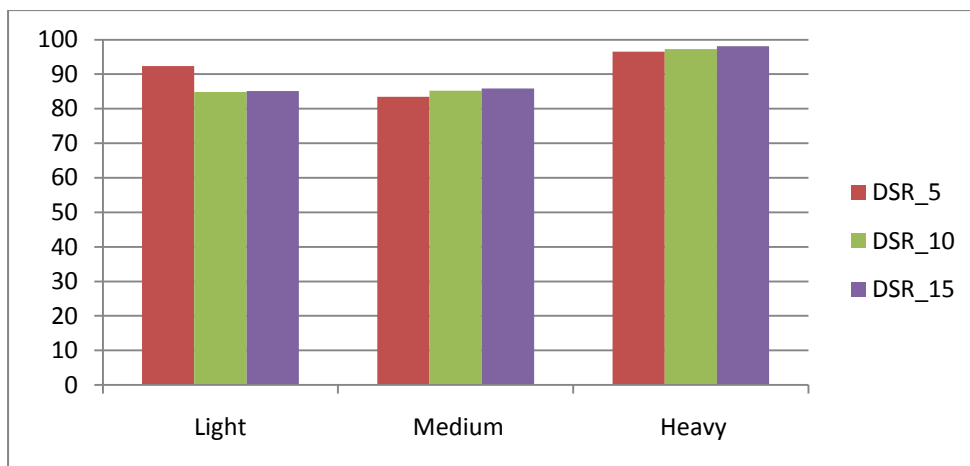
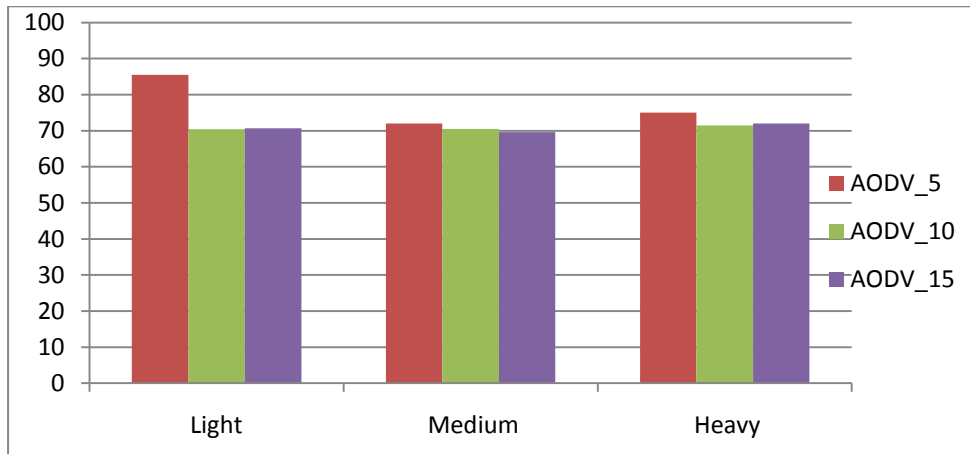


Wireless LAN.Throughput (bits/sec) (AODV, DSR, OLSR)



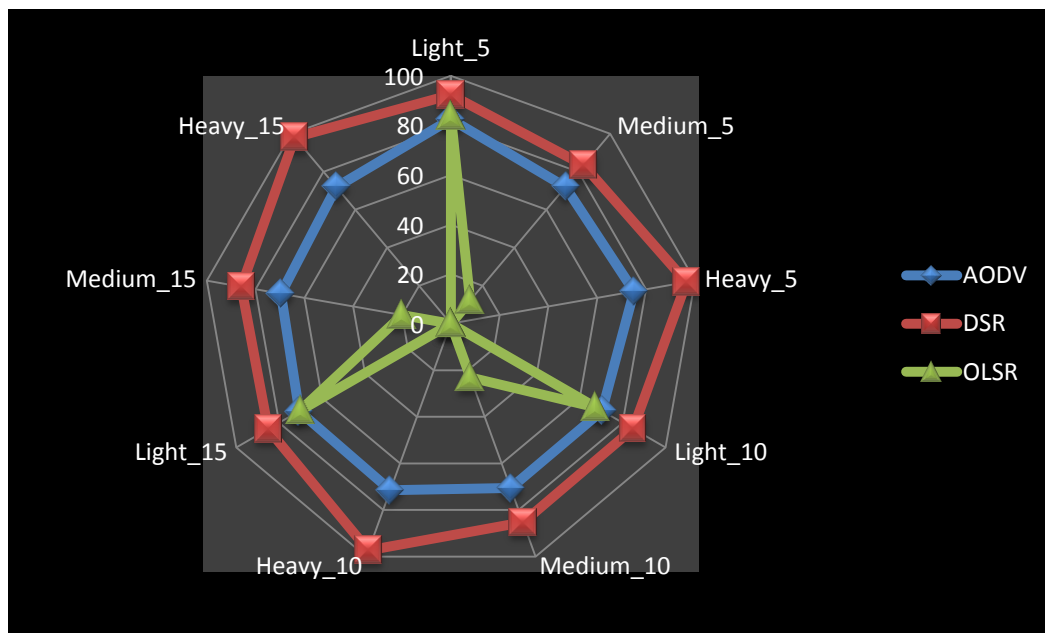
Συγκρίσεις μεταξύ ίδιων αλγορίθμων

Packet Delivery Ratio



Radar Diagram για το Packet Delivery Ratio

Πέρα από τις συνηθισμένες γραφικές παραστάσεις μελετήσαμε και την δυνατότητα να δημιουργήσουμε radar diagrams δείχνοντας συγκριτικά την απόδοση κάθε αλγόριθμου σε κάθε μετρική. Αυτό δεν καταστεί δυνατό λόγω του ότι οι διάφορες μετρικές ήταν πολύ διαφορετικές μεταξύ τους ενώ δεν υπήρχε συγκεκριμένο πεδίο τιμών για κάθε μεταβλητή, πράγμα που έκανε την γραφική παράσταση να μην παρουσιάζει ενδιαφέρον ή να μην είναι ευκολονόητη. Παρόλα αυτά μπορέσαμε να δημιουργήσουμε το radar diagram για το packet delivery ratio για όλα τα σενάρια που έχουμε τρέξει για τον κάθε αλγόριθμο. Έτσι φαίνεται συνολικά η απόδοση του κάθε αλγόριθμου για την μετρική αυτή. Το διάγραμμα φαίνεται πιο κάτω:



Παράρτημα Γ

Στο παράρτημα αυτό θα αναφερθούμε στα διάφορα ακρόνυμα τα οποία εμφανίστηκαν στην διπλωματική εργασία καθώς και σε όποιες λέξεις χρησιμοποιήθηκαν στα αγγλικά και δεν εξηγήθηκαν μέσα στο κύριο μέρος της εργασίας αυτής.

ΑΚΡΩΝΥΜΑ

AODV (Ad-Hoc on-Demand Distance Vector): Ο αλγόριθμος τον οποίο αναλύσαμε στο κεφάλαιο 3.

DAG (Directed Acyclic Graph): Κατευθυνόμενο γράφημα το οποίο είναι με τέτοιο τρόπο δημιουργημένο ώστε να μην επιτρέπει την δημιουργία κύκλου μεταξύ των κόμβων του.

DSR (Dynamic Source Routing): Ο αλγόριθμος τον οποίο αναλύσαμε στο κεφάλαιο 3.

GPS (Global Positioning System): Σύστημα το οποίο προσφέρει πληροφορίες που αφορούν τοποθεσία και χρόνο ώστε να ευκολύνει την κίνηση μίας οντότητας.

GRP (Geographic Routing Protocol): Ο αλγόριθμος τον οποίο αναλύσαμε στο κεφάλαιο 3.

OPNET (Optimized Network Engineering Tools): Ο προσομοιωτής τον οποίο χρησιμοποιήσαμε ώστε να αξιολογήσουμε τα διάφορα πρωτόκολλα.

MAC (Medium Access Control): Το δεύτερο layer στην στοίβα των δικτύων κατά το οποίο έχουμε την επικοινωνία μεταξύ δύο διαδοχικών κόμβων στο δίκτυο μας. Στο συγκριμένο layer έχουμε την χρήση των MAC addresses.

MANETS (Mobile Ad-Hoc Networks): Δίκτυα τα οποία αποτελούνται από κόμβους οι οποίοι κινούνται με χαμηλές ταχύτητες και έχουν κάποια συντονισμένη κίνηση. Ad-Hoc σημαίνει ότι δημιουργούνται χωρίς να υπάρχει κάτι προγραμματισμένο από πριν, δεν καθορίζεται από πριν η ακριβής τοπολογία καθώς και ποιοι κόμβοι θα επικοινωνούν μεταξύ τους. Η φράση Ad-Hoc είναι λατινική και στα αγγλικά σημαίνει for this.

MPR (Multi Point Relays): Κόμβοι στον αλγόριθμο OLSR οι οποίοι είναι υπεύθυνοι για αναμετάδοση των control μηνυμάτων στον αλγόριθμο αυτό.

OLSR (Optimized Link State Routing): Ο αλγόριθμος τον οποίο αναλύσαμε στο κεφάλαιο 3.

RREQ (Route Request): πακέτο το οποίο περιέχει το request για δημιουργία καινούργιας διαδρομής και το οποίο αποστέλλεται από τον κόμβο ο οποίος θέλει να στείλει πληροφορίες προς κάποιο προορισμό.

RREP (Route Reply): πακέτο το οποίο αποτελεί την απάντηση του προορισμού το οποίο αποστέλλεται προς την πηγή που έκανε το request.

RERR (Route Error): πακέτο το οποίο αποστέλλεται προς την πηγή από κάποιο κόμβο ο οποίος έχει ανακαλύψει κάποιο σφάλμα στην υφιστάμενη διαδρομή

SAPF (Speed Adaptive Probabilistic Flooding): Αλγόριθμος πιθανοτικού Flooding

TORA (Temporally Ordered Routing Algorithm): Ο αλγόριθμος τον οποίο αναλύσαμε στο κεφάλαιο 3.

TTL (Time to Live): Χρόνος τον οποίο "ζει" ένα πακέτο. Μετρίεται σε αριθμό hops.

VANETS (Vehicular Ad-Hoc Network): Δίκτυα τα οποία αποτελούνται από κόμβους οι οποίοι κινούνται με χαμηλές ταχύτητες χωρίς να υπάρχει ιδιαίτερα συντονισμένη κίνηση μεταξύ τους. Οι κόμβοι ως επί το πλείστον αποτελούνται από οχήματα. Ad-Hoc σημαίνει ότι δημιουργούνται χωρίς να υπάρχει κάτι προγραμματισμένο από πριν, δεν καθορίζεται από πριν η ακριβής τοπολογία καθώς και ποιοι κόμβοι θα επικοινωνούν μεταξύ τους. Η φράση Ad-Hoc είναι λατινική και στα αγγλικά σημαίνει for this.

WSN (Wireless Sensor Networks): Δίκτυα τα οποία αποτελούνται από sensors οι οποίοι συνεργάζονται μεταξύ τους ώστε να παρέχουν διάφορες πληροφορίες όπως θερμοκρασίες κ.λπ.

ΕΠΕΞΗΓΗΣΗ ΑΓΓΛΙΚΩΝ ΟΡΟΛΟΓΙΩΝ

Access Point: ο δρομολογητής μέσω του οποίου κάποιο τερματικό να μπορεί να αποκτήσει πρόσβαση στο δίκτυο.

Active timeout period: Ο χρόνος για τον οποίο θεωρείται ενεργός ένας κόμβος προς ένα προορισμό (εάν έχει στείλει πακέτα προς αυτόν δηλαδή).

Backtrack: Μέθοδος που εφαρμόζεται στον αλγόριθμο GRP ώστε εάν ένας κόμβος δεν μπορεί να επαναπροωθήσει κάποιο πακέτο, το αποστέλλει πίσω από όπου το έλαβε ώστε να βρεθεί εναλλακτική διαδρομή.

Cache: Η χρήση cache αφορά πιο συγκεκριμένα στην ύπαρξη μίας βάσης δεδομένων σε κάθε κόμβο όπου φυλάγονται οι διάφορες διαδρομές τις οποίες ανακαλύπτει κάθε κόμβος, ασχέτως αν δεν τον αφορούν. Ευρεία χρήση του cache κάνουν αλγόριθμοι όπως ο DSR.

Flooding: Η μέθοδος της πλημύρας όπου μία πηγή αποστέλλει κάποιο πακέτο request προς το δίκτυο. Κάθε κόμβος που το παραλαμβάνει, το αποστέλλει σε όλους τους γείτονες του. Αυτό συνεχίζεται μέχρι να φτάσει το πακέτο στο προορισμό του.

Forward Path Setup: Λειτουργία του αλγόριθμου AODV η οποία έχει επεξηγηθεί στο κεφάλαιο 3.

Glomosim, NS-2: Δύο προσομοιωτές δικτύων οι οποίοι χρησιμοποιούνται για προσομοιώσεις δικτύων VANETS, MANETS και WSN.

Hello interval: Χρονικό διάστημα μεταξύ της αποστολής δύο Hello μηνυμάτων.

Infrastructure-based ασύρματα δίκτυα: Μία από τις δύο μεγάλες κατηγορίες ασυρμάτων δικτύων. Στην ουσία αφορά δίκτυα τα οποία αποτελούνται από κάποια υποδομή (σύνδεση με καλώδιο δηλαδή) και το μόνο κομμάτι της σύνδεσης το οποίο είναι ασύρματο είναι από το τερματικό στο access point του δικτύου.

Infrastructure less-based ασύρματα δίκτυα: Η άλλη μεγάλη κατηγορία των ασυρμάτων δικτύων όπου δεν έχουμε καθόλου υποδομή και όλοι οι κόμβοι του δικτύου μας (τερματικοί ή μη) συνδέονται μεταξύ τους ασύρματα.

Link failure: Συμβαίνει όταν δύο κόμβοι οι οποίοι είναι συνδεδεμένοι μεταξύ τους, ο ένας είναι στην εμβέλεια του άλλου και ανταλλάζουν δεδομένα, απομακρύνονται ο ένας από τον άλλο και έτσι "σπάζει" η σύνδεση μεταξύ τους.

Link: Σύνδεσμος μεταξύ δύο κόμβων.

Multicasting: Διαδικασία όπου κάποιο κόμβος στέλνει δεδομένα προς μερικούς από τους κόμβους του δικτύου. Όταν αποστέλλει σε όλους του κόμβους του δικτύου, τότε αυτό ονομάζεται broadcasting, όταν αποστέλλεται σε μόνο ένα κόμβο τότε ονομάζεται unicasting.

On-Demand Reactive πρωτόκολλο: Ο όρος αυτός αναφέρεται σε πρωτόκολλα τα οποία δεν διατηρούν ανά πάσα στιγμή διαδρομές προς όλους τους προορισμούς αλλά βρίσκουν μία διαδρομή μόνο όταν υπάρχει ζήτηση για ανταλλαγή πληροφοριών προς κάποιο προορισμό.

Passive acknowledgment: Διαδικασία που γίνεται στον DSR κατά την οποία κάποιος κόμβος ακούει κάθε πακέτο που στέλνεται και είναι στην εμβέλεια του ώστε να δει εάν τα links τα οποία χρησιμοποιεί είναι έγκυρα.

Proactive πρωτόκολλο: Τα proactive πρωτόκολλα είναι αυτά τα οποία διατηρούν ανά πάσα στιγμή πληροφορίες προς όλους τους κόμβους του δικτύου.

Promiscuous mode: Είναι η λειτουργία κατά την οποία κάποιος κόμβος έχει την δυνατότητα να διαβάζει κάθε πακέτο το οποίο βρίσκεται στην εμβέλεια του.

Quadrant: Η έννοια της γειτονιάς την οποία έχουμε αναλύσει κατά την μελέτη του αλγόριθμου GRP.

Reverse Path Setup: Λειτουργία του αλγόριθμου AODV η οποία έχει επεξηγηθεί στο κεφάλαιο 3.

Retransmission: Η διαδικασία κατά την οποία ένα πακέτο αποστέλλεται ξανά από την πηγή λόγω του ότι δεν έχει ληφθεί η επιβεβαίωση από το προορισμό ότι λήφθηκε το συγκεκριμένο πακέτο.

Scalar δεδομένα: Είναι τα δεδομένα τα οποία μας παρέχει η OPNET και αφορούν την εξαγωγή του μέσου όρου συγκεκριμένων μεταβλητών.

Sequence number: Η χρήση του sequence number χρησιμοποιείται ευρέως από τους αλγόριθμους για την αποφυγή της δημιουργίας κύκλων κατά την αποστολή πακέτων, ειδικά εάν γίνεται broadcast.

Source Routing: Λειτουργία η οποία υπάρχει στον αλγόριθμο DSR όπου η πηγή γνωρίζει από την αρχή αποστολής κάποιου πακέτου ποια θα είναι η ακριβής διαδρομή προς τον προορισμό.

TC_Interval: Χρονικό διάστημα μεταξύ της αποστολής δύο control μηνυμάτων.

Trajectories: Οι τροχιές που έχουν οι κόμβοι στα σενάρια τα οποία αναλύσαμε.