

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΦΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ-ΠΜΣ
ΔΙΚΤΥΟΚΕΝΤΡΙΚΑ ΣΥΣΤΗΜΑΤΑ

ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΧΕΔΙΑΣΗΣ ΑΣΥΡΜΑΤΩΝ ΔΙΚΤΥΩΝ
ΚΟΡΜΟΥ

ΣΤΑΜΑΤΟΠΟΥΛΟΣ ΓΙΑΝΝΗΣ
ΜΕ/08103

ΕΠΙΒΛΕΠΩΝ: Κα ΑΛΕΞΙΟΥ ΑΓΚΕΛΙΚΗ

ΠΕΡΙΟΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1 ^ο	4
1.1 ΚΙΝΗΤΡΟ	4
1.2 ΣΥΝΕΙΣΦΟΡΑ	5
1.3 ΔΟΜΗ ΕΡΓΑΣΙΑΣ	5
ΚΕΦΑΛΑΙΟ 2. SINGLE-PATH ROUTING	6
2.1 ΕΙΣΑΓΩΓΗ	6
2.2 ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ MESH	6
2.3 ΑΠΑΙΤΗΣΕΙΣ ΔΙΚΤΥΟΥ (Requirements)	8
2.4 ΔΡΟΜΟΛΟΓΗΣΗ ΣΕ ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ MESH (BACKHAUL ROUTING)	8
2.5 Topology-Based Approach	9
2.5.1 Reactive πρωτόκολλα δρομολόγησης	9
2.5.2 Proactive πρωτόκολλα δρομολόγησης	9
2.6 ΠΡΩΤΟΚΟΛΛΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΜΙΑΣ ΔΙΑΔΡΟΜΗΣ (SINGLE PATH)	10
2.6.1 DSDV	10
2.6.1.1 Μειονεκτήματα (Drawbacks)	11
2.6.2 Bandwidth Routing Protocol	11
2.6.2.1 Μειονεκτήματα (Drawbacks)	12
2.6.3 Ad-Hoc On-Demand Distance Vector Routing Protocol	12
2.6.3.1 Route Discovery	12
2.6.3.2 Route Maintenance	14
2.6.4 Dynamic Source Routing (DSR)	15
2.6.4.1 Route Discovery	15
2.6.4.2 Route Maintenance	17
2.6.5 Optimized Link State Routing (OLSR)	17

2.6.5.1 Neighbor Sensing	18
2.6.5.2 Multipoint Relay Station.....	18
2.6.5.3 MPR Information Declaration	19
2.6.5.4 Routing Table Calculation	19
2.6.6 Zone Routing Protocol.....	19
2.6.6.1 Intra-zone Routing Protocol	21
2.6.6.2 Inter-zone Routing Protocol	21
2.6.7 Hover	22
2.6.7.1 Link Quality Estimation.....	23
2.6.7.2 Αναζήτηση Διαδρομής στον HOVER	24
2.7 ΣΥΓΚΡΙΣΗ ΠΡΩΤΟΚΟΛΛΩΝ SINGLE PATH.....	27
ΚΕΦΑΛΑΙΟ 3 ^ο : ΔΡΟΜΟΛΟΓΗΣΗ ΠΟΛΛΑΠΛΩΝ ΔΙΑΔΡΟΜΩΝ(MULTIPATH ROUTING).....	31
3.1 ΕΙΣΑΓΩΓΗ	31
3.2 Routing Metrics	32
3.3 PROACTIVE MULTIPATH ROUTING	33
3.4 REACTIVE MULTIPATH ROUTING	33
3.5 MULTIPATH ROUTING PROTOCOLS FOR WIRELESS BACKHAUL NETWORKS	35
3.5.1 Proactive Multiroute Algorithms	35
3.5.1.1 MOLSR	35
3.5.1.2 OLSR-based multipath routing.....	36
3.5.1.3 MMESH	36
3.5.2 Reactive Multiroute Algorithms.....	37
3.5.2.1 AODV-DM	37
3.5.2.2 AODV-BR.....	38
3.5.2.3 AOMDV	39

3.5.3 ΖΗΤΗΜΑΤΑ ΣΧΕΔΙΑΣΗΣ MULTI ROUTE ΑΛΓΟΡΙΘΜΩΝ	40
ΚΕΦΑΛΑΙΟ 4Ο : CLUSTERING ΑΛΓΟΡΙΘΜΟΙ.....	40
4.1 LOWEST ID.....	41
4.2 ACE.....	41
4.3 ΣΤΟΧΟΙ MULTI-PATH ROUTING	42
ΚΕΦΑΛΑΙΟ 5 ^ο : ΕΙΣΑΓΩΓΗ ΣΤΟ ΜΟΝΤΕΛΟ ΠΡΟΣΟΜΟΙΩΣΗΣ ΚΑΙ ΑΝΑΛΥΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ	43
5.1 Εισαγωγή στο OPNET Modeller	43
Εικόνα 2: Ιεραρχική δομή μοντελοποίησης στο OPNET.....	44
5.1.1 Μοντέλο Δικτύου.....	44
5.1.2 Μοντέλο Κόμβων.....	47
5.1.3 Μοντέλο Επεξεργασίας.....	49
5.2 Στόχοι της εξομοίωσης στο OPNET Modeller.....	51
6.ΠΑΡΑΤΗΡΗΣΕΙΣ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ	64
6.1 ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ.....	65

ΚΕΦΑΛΑΙΟ 1^ο

1.1 ΚΙΝΗΤΡΟ

Ένα WMN (Wireless Mesh Network) αποτελείται από ένα αριθμό από gateways, mesh routers (δρομολογητές) και mesh clients. Τα gateways καθώς και οι mesh routers που χρησιμοποιούνται αποτελούν την "ραχοκοκαλιά" (backbone) του δικτύου όπου η κινητικότητα των κόμβων (nodes) είναι μειωμένη. Οι δρομολογητές επικοινωνούν με τα άλλα δίκτυα (internet) προωθώντας τα πακέτα που παράγονται από την χρήση του δικτύου από τους clients προς τους gateway κόμβους οι οποίοι είναι συνδεδεμένοι με την ενσύρματη υποδομή (wired infrastructure). Σε ένα WMN δίκτυο κάθε δρομολογητής προωθεί πακέτα και εκ μέρους άλλων κόμβων οι οποίοι μπορεί να μην είναι στην εμβέλεια εκπομπής των κόμβων προορισμού (destination nodes). Επιπλέον οι λειτουργικότητα των gateway κόμβων έχουν καταφέρει την ενσωμάτωση των WMN δικτύων σε άλλες ασύρματες τεχνολογίες όπως είναι το Wi-Fi, κινητές επικοινωνίες και WiMax.

Σε τέτοιου τύπου δίκτυα οι κόμβοι εγκαθιστούν και διατηρούν συνδεσιμότητα μεταξύ τους δημιουργώντας με αυτό τον τρόπο ένα ad hoc δίκτυο. Για αυτό το λόγο ένα WMN δίκτυο θεωρείται ως αυτό-οργανωμένο (self-organized), αυτό-διαμορφώσιμο (self-configured) και redundant (διότι εάν ένας κόμβος πέσει οι υπόλοιποι κόμβοι μπορούν να επικοινωνήσουν μεταξύ τους). Επομένως τέτοιου τύπου δίκτυα παρουσιάζουν και πολλά πλεονεκτήματα όπως:

- Χαμηλό κόστος
- Εύκολη διαχείριση και συντήρηση δικτύου
- Αξιόπιστη κάλυψη υπηρεσιών
- Ενσωμάτωση με ήδη ευρέως χρησιμοποιούμενες ασύρματες τεχνολογίες (hybrid)

Τα WMN δίκτυα αναμένεται να βελτιστοποιήσουν αισθητά την απόδοση και τους περιορισμούς που υπάρχουν στα ad hoc δίκτυα συμπεριλαμβανομένων και των WLANs, WPANs και των ασύρματων μητροπολιτικών δικτύων (WMANs). Τα WMN δίκτυα αναμένεται να προσφέρουν υπηρεσίες σε ένα μεγάλο εύρος σεναρίων διαφορετικής κλίμακας όπως για παράδειγμα σε πανεπιστημιούπολεις (campus), σε τοπικές περιοχές καθώς επίσης και σε μεγαλύτερης κλίμακας περιοχής όπως γίνεται σήμερα με τα MANs.

Δεδομένου ότι τέτοιου είδους δίκτυα είναι κυρίως multi-hop, για να προωθήσουν πακέτα σε άλλους κόμβους πρέπει να κάνουν κάποιες αποφάσεις σύμφωνα με τον πίνακα δρομολόγησης που θα δημιουργηθεί και θα αποθηκευτεί σε κάθε κόμβο. Επομένως οι αλγόριθμοι δρομολόγησης που δημιουργήθηκαν στοχεύουν στην καλύτερη διαδρομή από τον αρχικό κόμβο προς τον gateway κόμβο. Με αυτό τον τρόπο είναι φυσικό συγκεκριμένες συνδέσεις να είναι υπερφορτωμένες ενώ άλλες συνδέσεις να χρησιμοποιούνται σπάνια. Αυτό θα έχει ως αποτέλεσμα να υπάρχει μειωμένη απόδοση του δικτύου (αφού θα υπάρχει μεγάλος φόρτος σε συγκεκριμένες συνδέσεις και πολλά πακέτα θα χάνονται) και στο τέλος τη αποτυχία ολοκλήρωσης της επικοινωνίας. Αυτό το σενάριο είναι πιθανό να αποφευχθεί αν χρησιμοποιηθεί η τεχνική πολλαπλής δρομολόγησης (multi-route). Η τεχνική αυτή έχει

ως στόχο την εξισορρόπηση της κίνησης (balanced traffic) δρομολογώντας πακέτα σε περισσότερες από μία διαδρομές που θα έχουν ως τελικό προορισμό τον gateway κόμβο. Η τεχνική multipath μπορεί να χρησιμοποιηθεί επίσης και για την εξασθένιση των παρεμβολών (αφού θα υπάρχουν πολλές διαδρομές) καθώς επίσης και για την ανθεκτικότητα λαθών.

Στόχος της διπλωματικής εργασίας είναι να αναλυθούν σε μεγαλύτερο βαθμό οι ήδη υπάρχουσες λύσεις για multi routing, να πραγματοποιηθεί μια σύγκριση ώστε να παρουσιαστούν πλεονεκτήματα αλλά και μειονεκτήματα κάθε λύσης καθώς και η εύρεση νέας τεχνικής που θα ενσωματωθεί σε ήδη υπάρχων αλγόριθμο και θα έχει ως στόχο την μεγιστοποίηση του throughput άρα και της συνολικής απόδοσης του δικτύου.

1.2 ΣΥΝΕΙΣΦΟΡΑ

Η εργασία παρουσιάζει και συγκρίνει τεχνικές για την εύρεση και διατήρηση διαδρομών . Ως αποτέλεσμα η εργασία έχει ως στόχο να προτείνει τα παρακάτω:

- Ένα ιεραρχικό μοντέλο αλγορίθμου όπου γειτονικοί κόμβοι συγκεντρώνονται σε ομάδες δημιουργώντας clusters καθώς και τις απαραίτητες προϋποθέσεις που χρειάζεται να εφαρμοστούν.
- Την ενσωμάτωση τεχνικής σε ένα cluster based αλγόριθμο δρομολόγησης που ακολουθεί το μοντέλο των πολλαπλών διαδρομών με στόχο την μεγιστοποίηση του συνολικού throughput του δικτύου.

1.3 ΔΟΜΗ ΕΡΓΑΣΙΑΣ

Η δομή της εργασίας παρουσιάζεται παρακάτω: Το 2ο κεφάλαιο παρουσιάζει μια εισαγωγή στις τεχνικές και στους υπάρχοντες αλγορίθμους που σχετίζονται με το αντικείμενο της εργασίας. Το 3ο κεφάλαιο παρουσιάζει αναλυτικά την τεχνική που προτείνει η εργασία και τα πλεονεκτήματα αυτής. Τέλος στο 4ο κεφάλαιο παρουσιάζονται αποτελέσματα και συμπεράσματα αναφέροντες τις τεχνικές που παρουσιάστηκαν και πιθανές μελλοντικές έρευνες πάνω στο συγκεκριμένο κομμάτι.

ΚΕΦΑΛΑΙΟ 2. SINGLE-PATH ROUTING

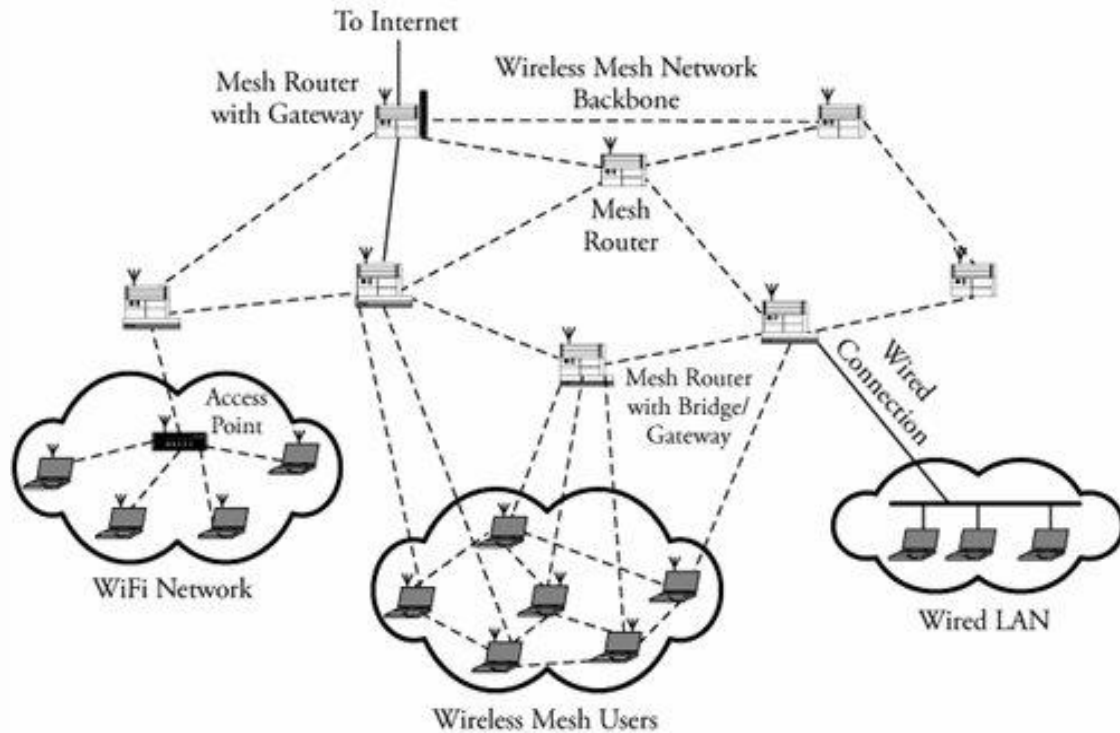
2.1 ΕΙΣΑΓΩΓΗ

Αν και τα δίκτυα τύπου mesh είναι στην ουσία μια καινούργια τεχνολογία, η έννοια της δρομολόγησης μέσω πολλαπλών διαδρομών (multi routing) είναι σχετικά παλιά η οποία εφαρμόστηκε για πρώτη φορά στις τηλεπικοινωνίες. Γνωστή ως εναλλακτική δρομολόγηση (alternate path routing) στα τηλεφωνικά δίκτυα μεταγωγής κυκλώματος, η τεχνική αυτή χρησιμοποιήθηκε για τη μείωση του blocking probability. Στην συγκεκριμένη τεχνική χρησιμοποιείται η βέλτιστη (κοντινότερη) διαδρομή ανάμεσα σε δύο οντότητες έως ότου παρουσιαστεί κάποιο πρόβλημα και οι κλήσεις δρομολογούνται στην εναλλακτική διαδρομή. Η έννοια των πολλαπλών διαδρομών έχει επίσης χρησιμοποιηθεί και στα δίκτυα ATM και πιο συγκεκριμένα στο PNNI σύστημα σηματοδότησης κατά την διαδικασία ανίχνευσης πολλαπλών εναλλακτικών διαδρομών. Άλλα γνωστά πρωτόκολλα όπως το OSPF είχαν χρησιμοποιήσει την έννοια του multipath αλλά με κάποιους περιορισμούς (οι διαδρομές πρέπει να έχουν ίσα κόστη).

Το κεφάλαιο ξεκινάει αναφέροντας τα χαρακτηριστικά των ασύρματων mesh δικτύων και τα πρωτόκολλα δρομολόγησης που χρησιμοποιούνται. Έπειτα αναλύεται η έννοια του multipath routing στο ασύρματο δίκτυο κορμού (backbone) συμπεριλαμβανομένων και των κυριότερων προβλημάτων που παρουσιάζονται καθώς επίσης και την λειτουργία του κάθε πρωτοκόλλου. Τέλος παρουσιάζεται μια αξιολόγηση των πρωτοκόλλων που αναλύθηκαν και παρουσιάζονται κάποια παραδείγματα.

2.2 ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ MESH

Τα δίκτυα αυτού του είδους συνδυάζουν στατικούς δρομολογητές (mesh routers-gateways) οι οποίοι λειτουργούν σε ad-hoc με τους κόμβους-πελάτη (client-nodes). Τα gateways είναι δρομολογητές που επικοινωνούν κυρίως ενσύρματα με υπόλοιπα δίκτυα (Internet) και οι mesh δρομολογητές είναι υπεύθυνοι για την δρομολόγηση των πακέτων στους τελικούς κόμβους (destination nodes). Επίσης προωθούν την κίνηση και στους gateway κόμβους όταν στον πίνακα δρομολόγησης δεν υπάρχει η διεύθυνση του τελικού κόμβου που αναφέρεται στο πακέτο. Το σύνολο αυτών των δρομολογητών σχηματίζει επομένως το backbone του δικτύου όπου η κινητικότητα είναι μειωμένη. Οι mesh clients αποτελούνται από φορητούς και σταθερούς υπολογιστές, κινητά τηλέφωνα ή άλλες ασύρματες συσκευές. Η εργασία επικεντρώνεται στο backbone αυτών των δικτύων.



Εικόνα 1: Αναπαράσταση ενός ασύρματου mesh δικτύου

Ένα mesh δίκτυο μπορεί να κατηγοριοποιηθεί ανάλογα με την αρχιτεκτονική του σε έναν από τους ακόλουθους τύπους:

- Backbone mesh: αποτελείται από έναν αριθμό δρομολογητών που επικοινωνούν μεταξύ τους με ασύρματες συνδέσεις και είναι υπεύθυνη για την δρομολόγηση των πακέτων στους clients. Οι gateway δρομολογητές παρέχουν επικοινωνία με εξωτερικά δίκτυα ή το Ιντερνέτ στους mesh clients.
- Client mesh: αποτελείται από κινητά ad-hoc δίκτυα αφού κάθε client λειτουργεί ως ανεξάρτητος δρομολογητής χωρίς βέβαια την παρουσία κάποιου πρωτοκόλλου δρομολόγησης.
- Hybrid mesh: αποτελεί συνδυασμό των δύο τύπων που αναλύθηκαν παραπάνω.

Η εργασία επικεντρώνεται στο backbone δίκτυο και ειδικότερα στον τρόπο που δρομολογούνται τα πακέτα και γενικότερα όλη η κίνηση προς τους mesh clients ή στους gateway κόμβους.

2.3 ΑΠΑΙΤΗΣΕΙΣ ΔΙΚΤΥΟΥ (Requirements)

Έχοντας υπόψη τα παραπάνω χαρακτηριστικά που αναλύθηκαν, ένα backhaul δίκτυο έχει τις παρακάτω απαιτήσεις

- Υψηλή χωρητικότητα (High Capacity)
- Γρήγορη και επιταχυνόμενη κλιμάκωση (Accelerated and Rapid Scalability)
- 99.999% διαθεσιμότητα (Availability)
- Quality of Service (QoS)
- Μικρό χρόνο καθυστέρησης (Ultra low latency-Jitter)
- Επαρκής κάλυψη (Coverage)

2.4 ΔΡΟΜΟΛΟΓΗΣΗ ΣΕ ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ MESH (BACKHAUL ROUTING)

Η δρομολόγηση(routing) σε ένα ασύρματο Backhaul δίκτυο αποτελεί τον πυρήνα της ορθής λειτουργίας του. Ο αλγόριθμος δρομολόγησης που εισάγεται είναι υπεύθυνος για την σωστή διαχείριση της εισερχόμενης και εξερχόμενης κίνησης στο δίκτυο, για την διαχείριση των πόρων που πρέπει να δεσμευτούν (resource allocation) καθώς επίσης και για την αξιόπιστη και γρήγορη μετάδοση-παραλαβή των πακέτων. Η δρομολόγηση ως όρος αντιπροσωπεύει την συνδεσιμότητα του δικτύου δίνοντας έμφαση στην επικοινωνία και στο status των links του δικτύου. Αυτή η προσέγγιση ονομάζεται Topology-Based Approach.Μια δεύτερη προσέγγιση έχει να κάνει με την τοποθεσία των κόμβων του δικτύου για την εύρεση βέλτιστης διαδρομής και ονομάζεται Location-Based Approach.

Για να επιτευχθεί υψηλή απόδοση σε τέτοιου είδους δίκτυα είναι απαραίτητος ο έλεγχος των παρακάτω παραμέτρων:

- Λάθη μετάδοσης: η αναξιοπιστία του ασύρματου μέσου είναι πολύ πιθανό να προκαλέσει λάθη κατά την μετάδοση του πακέτου.
- Σφάλμα στις συνδέσεις και στους κόμβους: υπάρχει η πιθανότητα οι κόμβοι και οι συνδέσεις του δικτύου να υπολειτουργούν ή να σταματήσουν να λειτουργούν λόγω μηχανικού σφάλματος ή άλλων παραγόντων (π.χ. συνθήκες περιβάλλοντος).
- Λανθασμένες διαδρομές (routes): εξαιτίας κάποιου σφάλματος σε κόμβο ή μετά από λανθασμένο σχεδιασμό στην επέκταση του δικτύου.
- Κορεσμένες διαδρομές: Εξαιτίας της φύσης των πρωτοκόλλων δρομολόγησης αλλά και της τοπολογίας του δικτύου είναι πιθανό ορισμένες διαδρομές να είναι υπερφορτωμένες προκαλώντας έτσι σημαντικές καθυστερήσεις και μείωση στην απόδοση του δικτύου.

Στην συγκεκριμένη διπλωματική εργασία αναλύεται η Topology-Based Approach για το γεγονός ότι η ύπαρξη ενός GPS συστήματος μπορεί να μην είναι διαθέσιμο στα σενάρια που θα αναλυθούν.

2.5 Topology-Based Approach

Στη προσέγγιση αυτή, τα πρωτόκολλα μπορούν να κατηγοριοποιηθούν σε τρεις κατηγορίες.

- Proactive (Table-driven)
- Reactive (on Demand)
- Hybrid
- Στον παρακάτω πίνακα γίνεται μια σύγκριση των σημαντικότερων παραμέτρων μεταξύ proactive και reactive αλγορίθμων

Παράμετροι	Proactive	Reactive
Διαθεσιμότητα πληροφοριών δρομολόγησης	Πάντα διαθέσιμες ανεξαρτήτου κατάστασης	Διαθέσιμες όταν χρειάζονται
Περιοδικά route updates	Ναι	Δεν απαιτείται
Υποστήριξη QoS	Κυρίως το shortest path ως παράγοντας QoS	Συγκεκριμένη αλγόριθμοι το υποστηρίζουν

Πίνακας 1: Χαρακτηριστικά proactive και reactive αλγορίθμων

2.5.1 Reactive πρωτόκολλα δρομολόγησης

Τα Reactive πρωτόκολλα αποσκοπούν στην μείωση συμβατικών routing tables στους κόμβους και συνεπώς την ανάγκη για routing table updates ώστε να ανιχνεύονται οι αλλαγές στην τοπολογία του δικτύου. Ως αποτέλεσμα μια on-demand διαδικασία είναι απαραίτητη για την εύρεση βέλτιστης διαδρομής. Δημιουργώντας routing tables μόνο όταν είναι απαραίτητο μειώνεται δραστικά το overhead και το routing table storage. Ωστόσο η διαδικασία που ενεργοποιείται στην αρχή μπορεί να καθυστερήσει αρκετά το route setup. Επιπλέον η ποιότητα του μονοπατιού(bandwidth, delay κ.α.) γνωστοποιούνται μετά την εγκατάσταση του route. Τέλος τα Reactive routing πρωτόκολλα δεν είναι χρήσιμα σε real-time υπηρεσίες.

2.5.2 Proactive πρωτόκολλα δρομολόγησης

Τα σενάρια που εξετάζονται στη συγκεκριμένη διπλωματική έχουν να κάνουν με στατικούς κόμβους στο backhaul δίκτυο. Σε αυτή την περίπτωση είναι αναγκαία η χρήση πρωτοκόλλων δρομολόγησης που τρέχουν σε τακτά χρονικά διαστήματα παρά να ενεργοποιούνται μέσω συγκεκριμένων triggers όπως για παράδειγμα όταν πέσει κάποιος κόμβος. Επιπλέον το QoS (delay, bandwidth κ.λ.π) αποτελεί σημαντικό παράγοντα στα backhaul δίκτυα καθώς όλα οι MSS λειτουργούν σύμφωνα με τις απαιτήσεις των υπηρεσιών. Για το λόγο αυτό η χρήση proactive πρωτοκόλλων δρομολόγησης είναι πιο χρήσιμη για τα σενάρια που θα εξεταστούν.

Για την χρήση proactive δρομολόγησης απαιτείται ότι κάθε κόμβος διατηρεί το δικό του routing table το οποίο διαθέτει όλες τις πιθανές διαδρομές προς όλους τους προορισμούς καθώς και τον αριθμό των hops(κόμβων) προς κάθε πιθανό προορισμό. Ο πίνακας δρομολόγησης είναι κατασκευασμένος έτσι ώστε να υποστηρίζει είτε link state είτε distance vector αλγορίθμους. Ο λόγος που έχουν κατασκευαστεί με αυτό τον τρόπο είναι για υπάρχει διαθεσιμότητα της πληροφορίας

Ένας μεγάλος αριθμών αλγορίθμων έχει δημιουργηθεί ώστε να παρέχει επικοινωνία στο backhaul δίκτυο. Παρακάτω γίνεται μια ανάλυση των σημαντικότερων αλγορίθμων δρομολόγησης καθώς και μια σύγκριση στα πλεονεκτήματα και μειονεκτήματα που παρουσιάζουν.

2.6 ΠΡΩΤΟΚΟΛΛΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΜΙΑΣ ΔΙΑΔΡΟΜΗΣ (SINGLE PATH)

2.6.1 DSDV

Ο συγκεκριμένος αλγόριθμος αποτελείται από τρία χαρακτηριστικά

1. Χρησιμοποιείται μηχανισμός που αυξάνει περιοδικά τα sequence numbers που χρησιμοποιούνται με σκοπό να μένουν τα routes στο προσκήνιο και να μη θεωρούνται expired. Αυτό έχει ως αποτέλεσμα την αποφυγή των routing loops με πιο τρανταχτό παράδειγμα το πρόβλημα “Count to infinity”.
2. Χρησιμοποιεί full route updates τα οποία στέλνονται περιοδικά σε κάθε time interval.
3. Η καθυστέρηση (delay) των route updates για τις διαδρομές που είναι ασταθής όπως για παράδειγμα εκείνων που το update βρίσκεται ένα κόμβο πριν τον προορισμό.

Στον DSDV αλγόριθμο κάθε κόμβος στο δίκτυο κρατάει το δικό του routing table το οποίο περιέχει όλες τις πιθανές διαδρομές προς τον προορισμό καθώς επίσης και τα κόστη(αριθμός των hops) για κάθε προορισμό που έχει καταγραφεί. Κάθε νέα εισαγωγή ενός route έχει ένα μοναδικό σειριακό αριθμό (sequence number). Τα sequence numbers επιτρέπουν σε κάθε κόμβο να ξεχωρίζει τα νέα routes που εισάγονται στους πίνακες δρομολόγησης από τα πιο παλιά αποφεύγοντας έτσι τη δημιουργία routing loops. Ένας μονός αριθμός χρησιμοποιείται για απόσταση ίση με άπειρο πράγμα που δηλώνει ότι ο κόμβος είναι unreachable, ενώ οι ζυγοί αριθμοί δηλώνουν τα updates που γίνονται στους πίνακες δρομολόγησης.

Τα updates που στέλνονται περιοδικά σε όλο το δίκτυο γίνονται ώστε να υπάρχει μια σταθερότητα και αξιοπιστία της πληροφορίας στους πίνακες δρομολόγησης κάθε κόμβου. Όταν ένας κόμβος λάβει κάποιο update σχετικό με route update, ο κόμβος αυτός θα ανανεώσει τον πίνακα δρομολόγησης αν και μόνο αν το sequence number του update που στέλνεται είναι μεγαλύτερο από αυτό που υπάρχει στο routing table. Στην περίπτωση που υπάρχουν δύο ή περισσότερα updates που

έχουν το ίδιο sequence number, θα επιλεγεί η διαδρομή με το μικρότερο metric (αριθμός των hops) επομένως και η πιο σύντομη (shortest path optimization).

Ένα route διαγράφεται από τον πίνακα δρομολόγησης αν ένας κόμβος δεν λάβει κάποιο update σε κάποιο χρονικό περιθώριο (time interval).

Συνοψίζοντας ο DSDV αλγόριθμος ανήκει στην κατηγορία των distance-vector routing πρωτοκόλλων με μικρές παραλλαγές ώστε να προσαρμοστεί καλύτερα σε backhaul δίκτυα. Οι παραλλαγές αυτές έχουν να κάνουν κυρίως με τα update triggers τα οποία στέλνονται όταν γίνει κάποια αλλαγή στην τοπολογία του δικτύου και σε χρονική περίοδο πριν τα scheduled updates. Ο αλγόριθμος περιλαμβάνει δύο στρατηγικές για τη διαχείριση των triggered events.

- DADV-SQ : Ο κόμβος στέλνει ένα triggered update κάθε φορά που λαμβάνει ένα sequence number για κάποιο τελικό κόμβο.
- Για να μειωθεί η κίνηση στο δίκτυο λόγω των περιοδικών updates, τα route updates έχουν δύο τύπους πακέτων
 - Full dump : το οποίο διαθέτει όλη την πληροφορία δρομολόγησης
 - Incremental dump : περιέχουν μόνο την πληροφορία που άλλαξε από το τελευταίο full dump.

2.6.1.1 Μειονεκτήματα (Drawbacks)

Το μεγαλύτερο μειονέκτημα του DSDV είναι που χρησιμοποιεί ως metric τον αριθμό των hops ώστε να βρει τη συντομότερη διαδρομή, ενώ παράλληλα δεν παίρνει ως παράμετρο το φυσικό στρώμα όπως για παράδειγμα τη χωρητικότητα του καναλιού, τις παρεμβολές κ.α αλλά και παραμέτρους από το MAC επίπεδο όπως για παράδειγμα το queue και transmission delay.

Οι παράμετροι αυτοί αποτελούν σημαντικό παράγοντα στην επιλογή βέλτιστης διαδρομής για σε ένα ασύρματο backhaul δίκτυο γιατί το βέλτιστο route δε σημαίνει ότι και η διαδρομή με τους λιγότερους κόμβους (shortest path). Επίσης, στην περίπτωση υπηρεσιών πραγματικού χρόνου το δίκτυο πρέπει να είναι σε θέση να ελαχιστοποιήσει τις παραμέτρους που αναφέρθηκαν παραπάνω.

2.6.2 Bandwidth Routing Protocol

Θεωρεί ένα CDMA-over-TDMA σύστημα. Στον συγκεκριμένο αλγόριθμο κάθε frame χωρίζεται σε δύο φάσεις, στη φάση ελέγχου (control phase) και στην φάση δεδομένων (data phase). Ως QoS παραμέτρους θεωρείται το bandwidth.

Ο BR αποτελείται από τρεις υπό-αλγόριθμους.

- Ο πρώτος υπολογίζει το bandwidth που απαιτείται από την πηγή (source) προς οποιονδήποτε τελικό προορισμό (destination).
- Ο δεύτερος αλγόριθμος εξασφαλίζει έναν επαρκή αριθμό από ελεύθερα slot για το QoS flow.
- Ο τρίτος είναι ένας standby αλγόριθμος ο οποίος υπολογίζει το bandwidth στην περίπτωση που πέσει κάποιο route.

Το bandwidth μεταξύ δύο γειτονικών κόμβων ορίζεται ως ο κοινός αριθμός από ελεύθερα slots μεταξύ τους.

2.6.2.1 Μειονεκτήματα (Drawbacks)

Αν και ο BR αλγόριθμος παρέχει μια ικανοποιητική κατανομή του bandwidth μεταξύ των κόμβων του backhaul δικτύου έχει τα παρακάτω μειονεκτήματα:

- Στη φάση ελέγχου, ένα μοναδικό slot πρέπει να ανατεθεί στατικά σε κάθε κόμβο πριν εισαχθεί στο δίκτυο. Με αυτό τον τρόπο μειώνεται η ευελιξία του δικτύου γιατί δεν επιτρέπει την εισαγωγή νέων κόμβων σε επόμενο χρονικό διάστημα.
- Το επίπεδο ποιότητας σε φυσικό επίπεδο θεωρείται ότι είναι ίδιο για όλους τους κόμβους πράγμα απίθανο σε ένα πραγματικό περιβάλλον.

2.6.3 Ad-Hoc On-Demand Distance Vector Routing Protocol

Ανήκει στην κατηγορία των distance vector αλγορίθμων και βασίζεται κυρίως στη λειτουργία των DSDV και DSR αλγορίθμων. Χρησιμοποιεί τα χαρακτηριστικά του DSR για αναζήτηση κατά απαίτηση (on-demand) επομένως βρίσκει νέες διαδρομές μόνο όποτε χρειάζεται χρησιμοποιώντας μια παρόμοια διαδικασία αναζήτησης. Παρόλα αυτά ο AODV κρατεί μια εγγραφή για κάθε προορισμό στον πίνακα δρομολόγησης και όχι πολλές όπως κάνει ο DSR.

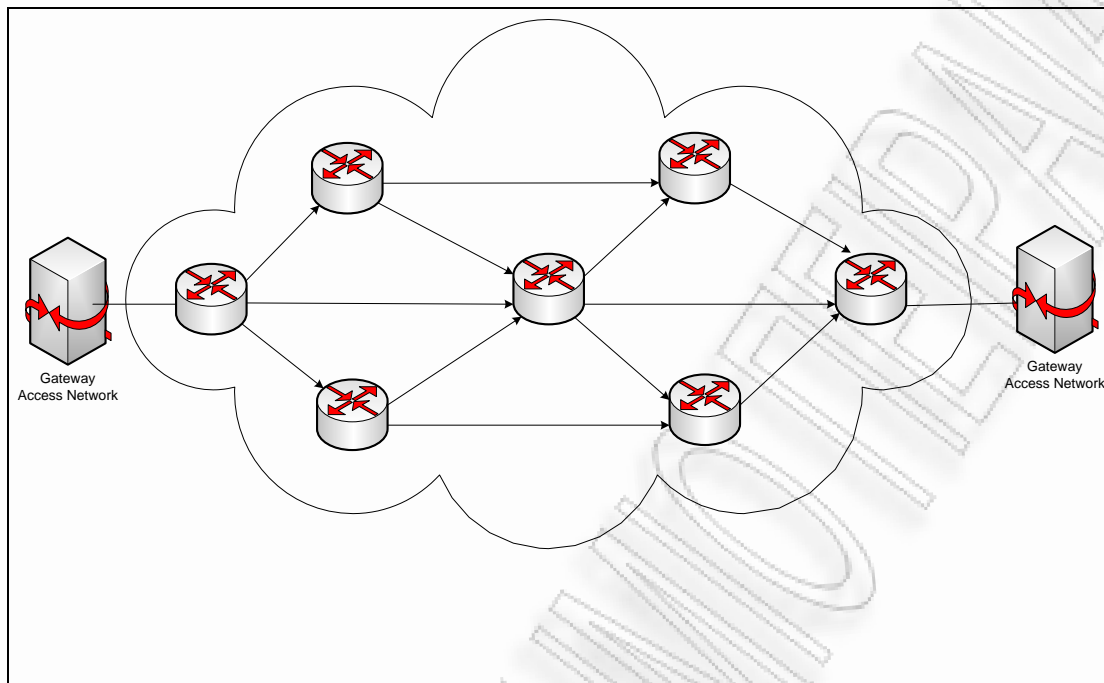
Όπως και ο DSDV αποτρέπει την εμφάνιση routing loops και δεν χρειάζεται περιοδικά route updates τα οποία ανεβάζουν την καθυστέρηση του δικτύου (delay).

Ο AODV έχει επιπλέον τα ακόλουθα χαρακτηριστικά:

- Όταν υπάρχει μια διαθέσιμη διαδρομή από τον αρχικό προς τον τελικό κόμβο δεν προστίθεται κάποιο επιπλέον overhead στα πακέτα μετάδοσης.
- Η διαδικασία αναζήτησης διαδρομής ξεκινάει μόνο όταν τα routes τα οποία υπάρχουν στους πίνακες δρομολόγησης δεν χρησιμοποιούνται ή έχουν λήξει. Αυτή η στρατηγική μειώνει το φαινόμενο stale route καθώς και την διατήρηση διαδρομών που δεν χρησιμοποιούνται.
- Παρέχει unicast, multicast και broadcast επικοινωνία. Ο AODV χρησιμοποιεί έναν broadcast αλγόριθμο για την αναζήτηση των routes και έπειτα ένα unicast μήνυμα απάντηση (route reply message).

2.6.3.1 Route Discovery

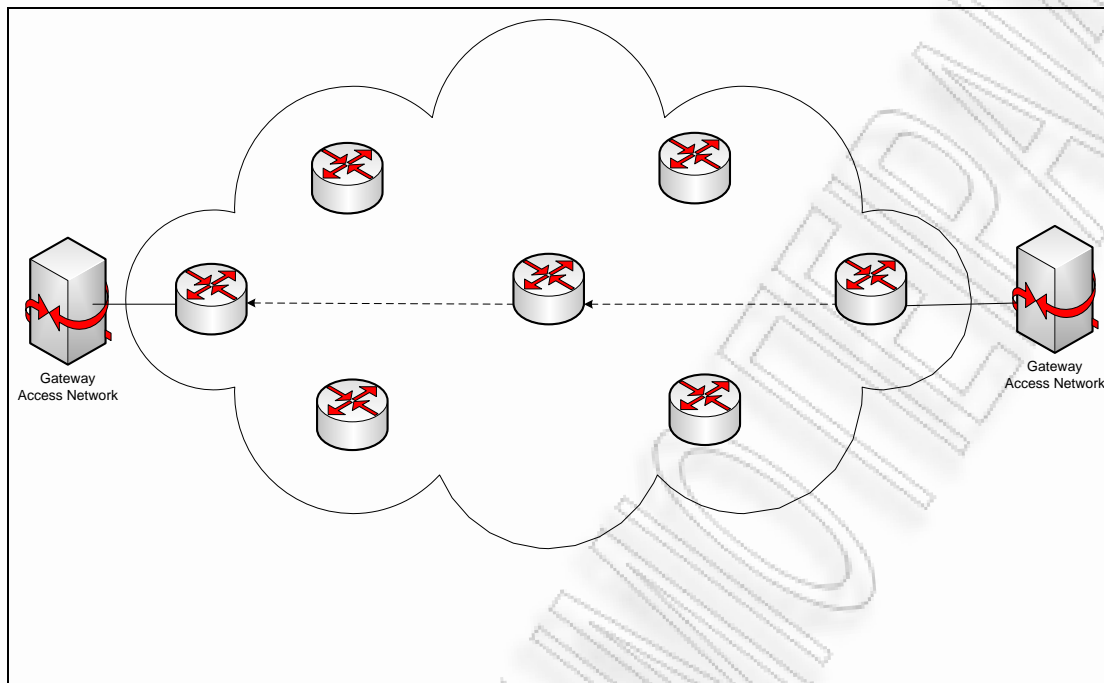
Όταν ένας κόμβος θέλει να στείλει ένα πακέτο σε ένα προορισμό (destination) και δεν υπάρχει κάποια εγγραφή στο πίνακα δρομολόγησης ενεργοποιείται η διαδικασία αναζήτησης. Ο αρχικός κόμβος (source node) ένα broadcast request μήνυμα (RREQ) σε όλους τους γειτονικούς κόμβους οι οποίοι προωθούν το μήνυμα στους δικούς τους γείτονες κ.ο.κ3. Το σχήμα 1 δείχνει την μετάδοση του RREQ μηνύματος στο δίκτυο. Τα βελάκια δηλώνουν την εκπομπή broadcast προς τους γειτονικούς κόμβους.



Σχήμα 1 :Μετάδοση RREQ στο backhaul δίκτυο.

Για να μπορεί να ελεγχθεί η ποσότητα των broadcast μηνυμάτων ώστε να μην υπάρχει συμφόρηση στο δίκτυο (network load) ο αρχικός κόμβος χρησιμοποιεί την τεχνική του δακτυλίου. Σύμφωνα με αυτή την τεχνική ο κόμβος ξεκινάει την αναζήτηση χρησιμοποιώντας ένα Time To Live (TTL). Αν δεν έχει πάρει απάντηση με την λήξη του TTL τότε το TTL αυξάνεται κατά μία μονάδα. Η διαδικασία αυτή θα επαναλαμβάνεται μέχρι το TTL να φτάσει την τιμή κατωφλίου (threshold).

Όταν ένας ενδιάμεσος κόμβος προωθεί το RREQ δημιουργεί μια εγγραφή με την διεύθυνση του γειτονικού κόμβου από τον οποίο παρέλαβε το broadcast μήνυμα, δημιουργώντας έτσι ένα αντίστροφο μονοπάτι (reverse path). Όταν το RREQ λαμβάνεται από έναν κόμβο ο οποίος είτε είναι ο τελικός είτε ένας ενδιάμεσος που περιέχει ένα updated route για τον τελικό, επιστρέφει ένα unicast μήνυμα απάντησης προς τον αρχικό κόμβο. Καθώς το RREQ δρομολογείται προς τον αρχικό κόμβο μέσω του reverse path όλοι οι ενδιάμεσοι κόμβοι εισάγουν στους πίνακες δρομολόγησης την διεύθυνση από όπου έλαβαν το απαντητικό μήνυμα επομένως όταν το RREQ φτάσει στον αρχικό κόμβο θα έχει ανιχνευθεί το route από το αρχικό προς τον τελικό κόμβο. Το σχήμα 2 δείχνει την αντίστροφη διαδικασία από τον τελικό στον αρχικό κόμβο κατά την οποία στέλνεται μήνυμα απάντησης ότι ο τελικός κόμβος έχει βρεθεί. Με την διαδικασία αυτή ανιχνεύεται και η τελική διαδρομή που θα ακολουθήσει το πακέτο.



Σχήμα 2: Μήνυμα απάντησης RRREP από τον τελικό στον αρχικό κόμβο.

2.6.3.2 Route Maintenance

Η διαδρομή που έχει εγκατασταθεί μεταξύ πηγής και προορισμού διατηρείται για όσο χρονικό διάστημα απαιτείται από τον αρχικό κόμβο. Αν ο κόμβος μετακινηθεί κατά της διάρκεια μιας ενεργής σύνδεσης μπορεί να επαναενεργοποιήσει τη διαδικασία αναζήτησης για νέο route. Αν υπάρξει μετακίνηση του τελικού κόμβου ή ενός από τους ενδιάμεσους τότε αφαιρούνται από τους πίνακες δρομολόγησης οι εγγραφές για τον τελικό κόμβο και στέλνεται μήνυμα λάθους (RERR) σε όλους τους κόμβους που μπορεί να επηρεαστούν από την μετακίνηση. Η διαδικασία επαναλαμβάνεται μέχρι το μήνυμα να φτάσει στον αρχικό κόμβο ο οποίος επιλέγει αν θα σταματήσει την κίνηση προς το συγκεκριμένο route ή να ξεκινήσει νέα αναζήτηση στέλνοντας νέο RREQ.

2.6.4 Dynamic Source Routing (DSR)

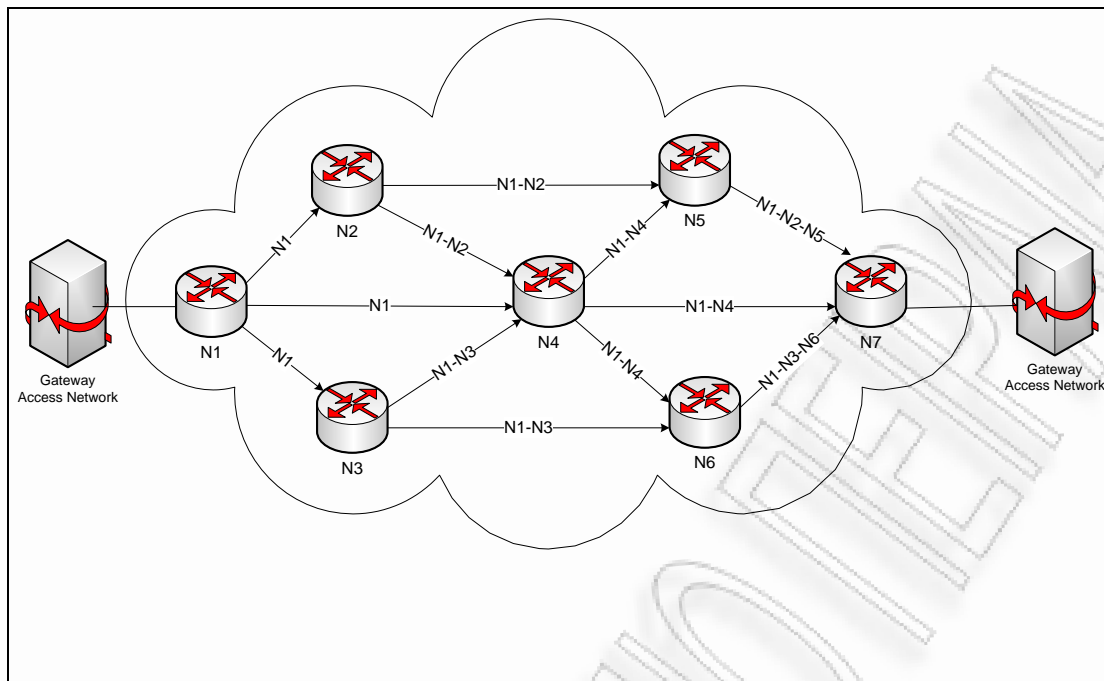
Σχεδιάστηκε ειδικά για multihop δίκτυα. Επιτρέπει στο δίκτυο να είναι αυτό-οργανωμένο (self-organized) και αυτό-διαχειριζόμενο (self-configured). Ο DSR δεν χρησιμοποιεί περιοδικά updates όπως ο AODV επομένως μειώνεται σε σημαντικό βαθμό το overhead, η ποσότητα ενέργειας που καταναλώνεται από τους κόμβους και αποφεύγονται τα μεγάλα και συχνά routing updates. Το χαρακτηριστικό του DSR είναι ότι χρειάζεται και υποστήριξη από το επίπεδο MAC σε περίπτωση που ανιχνευθεί σφάλμα σε κάποιο link.

Ο DSR αποτελείται από δύο μηχανισμούς Route Discovery και Route Maintenance οι οποίοι λειτουργούν μαζί για να επιτρέπουν στους κόμβους να ανακαλύπτουν και να διατηρούν τις διαδρομές που υπάρχουν στους εκάστοτε τελικούς προορισμούς. Η επόμενη ενότητα εξηγεί με περισσότερες λεπτομέρειες τους μηχανισμούς αυτούς.

2.6.4.1 Route Discovery

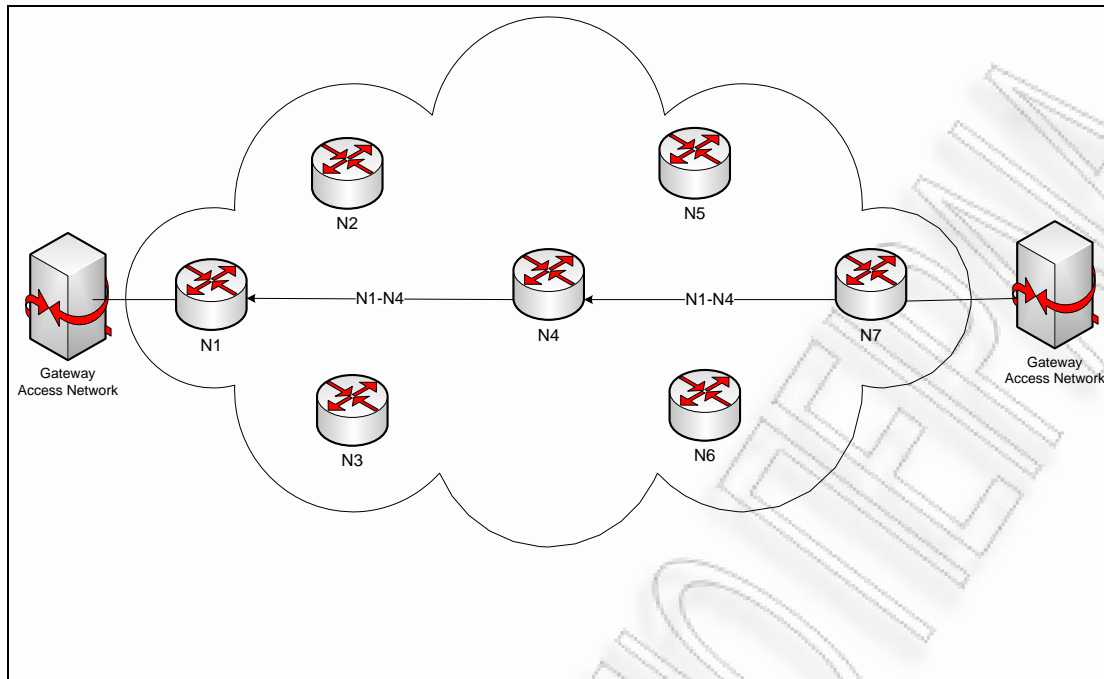
Όταν ένας κόμβος θέλει να στείλει ένα πακέτο σε έναν τελικό προορισμό, ελέγχει πρώτα την route cache για να καθορίσει αν ήδη υπάρχει διαδρομή, προς τον τελικό προορισμό, στον πίνακα δρομολόγησης. Αν υπάρχει εγγραφή ο αλγόριθμος επιλέγει να μεταδώσει το πακέτο προς την συγκεκριμένη διαδρομή. Σε αντίθετη περίπτωση ξεκινάει την διαδικασία αναζήτησης εκπέμποντας ένα συγκεκριμένο broadcast μήνυμα (route request packet).

Κάθε κόμβος που λαμβάνει αυτό το μήνυμα ψάχνει στη δικιά του route cache που να έχει τον τελικό κόμβο. Αν δεν υπάρχει εγγεγραμμένη η διαδρομή γράφει την δικιά του διεύθυνση στον πίνακα δρομολόγησης και προωθεί το μήνυμα στους γειτονικούς κόμβους. Η διαδικασία αυτή επαναλαμβάνεται μέχρι το μήνυμα να φτάσει στον τελικό κόμβο ή να υπάρχει ενδιάμεσος κόμβος που να έχει εγγραφή προς τον τελικό στον πίνακα δρομολόγησης του. Το σχήμα 3 δείχνει τη διαδρομή που ακολουθεί το μήνυμα μέχρι να φτάσει στον τελικό κόμβο.



Σχήμα 3 Route Request Packet

Όταν το broadcast μήνυμα φτάσει τον τελικό κόμβο ή κάποιον ενδιαμέσο που να γνωρίζει τη διαδρομή για τον τελικό στέλνεται ένα unicast μήνυμα απάντησης στον αρχικό κόμβο. Το σχήμα 4 δείχνει την τελική διαδρομή που θα εγγραφεί στους πίνακες δρομολόγησης των κόμβων.



Σχήμα 4 Route Reply Message

2.6.4.2 Route Maintenance

Στον DSR μια διαδρομή μένει εγγεγραμμένη στον πίνακα δρομολόγησης με την αποστολή και λήψη acknowledgements. Όταν ένα πακέτο προωθείται στους επόμενους κόμβους, κάθε κόμβος που εκπέμπει ή προωθεί το πακέτο είναι υπεύθυνος για την επιτυχή λήψη του γειτονικού του κόμβου (next hop). Το πακέτο στέλνεται συνεχώς μέχρι να ληφθεί μήνυμα επιβεβαίωσης από το next hop. Αν οι αποτυχημένες προσπάθειες λήψης του πακέτου περάσουν το threshold που έχει οριστεί ο κόμβος στέλνει ένα μήνυμα λάθους (route error message) στον αρχικό κόμβο που έστειλε το πακέτο. Με την λήψη αυτού του μηνύματος όλοι οι κόμβοι διαγράφουν τον λανθασμένο κόμβο από το πίνακα δρομολόγησης.

2.6.5 Optimized Link State Routing (OLSR)

Ανήκει στην κατηγορία των proactive αλγορίθμων και έχει το πλεονέκτημα εύρεσης διαδρομών μόλις είναι αναγκαίο. Η ιδέα που χρησιμοποιείται σε αυτό το πρωτόκολλο είναι αυτή των multipoint relays (MPR). Τα MPRs είναι στην ουσία οι γειτονικοί κόμβοι οι οποίοι προωθούν broadcast μηνύματα κατά τη διάρκεια αναζήτησης διαδρομής. Ο OLSR μειώνει το μέγεθος των μηνυμάτων αναζήτησης δηλώνοντας έναν αριθμό κόμβων που θα μπορούν να εκπέμπουν τα broadcast μηνύματα μέσω της διαδικασίας των MPR. Ως εκ τούτου ο αλγόριθμος μειώνει την κίνηση στο δίκτυο και δεν το υπερφορτώνει σε περίπτωση κάποιου λάθους ή κάποιας βλάβης σε κόμβο.

2.6.5.1 Neighbor Sensing

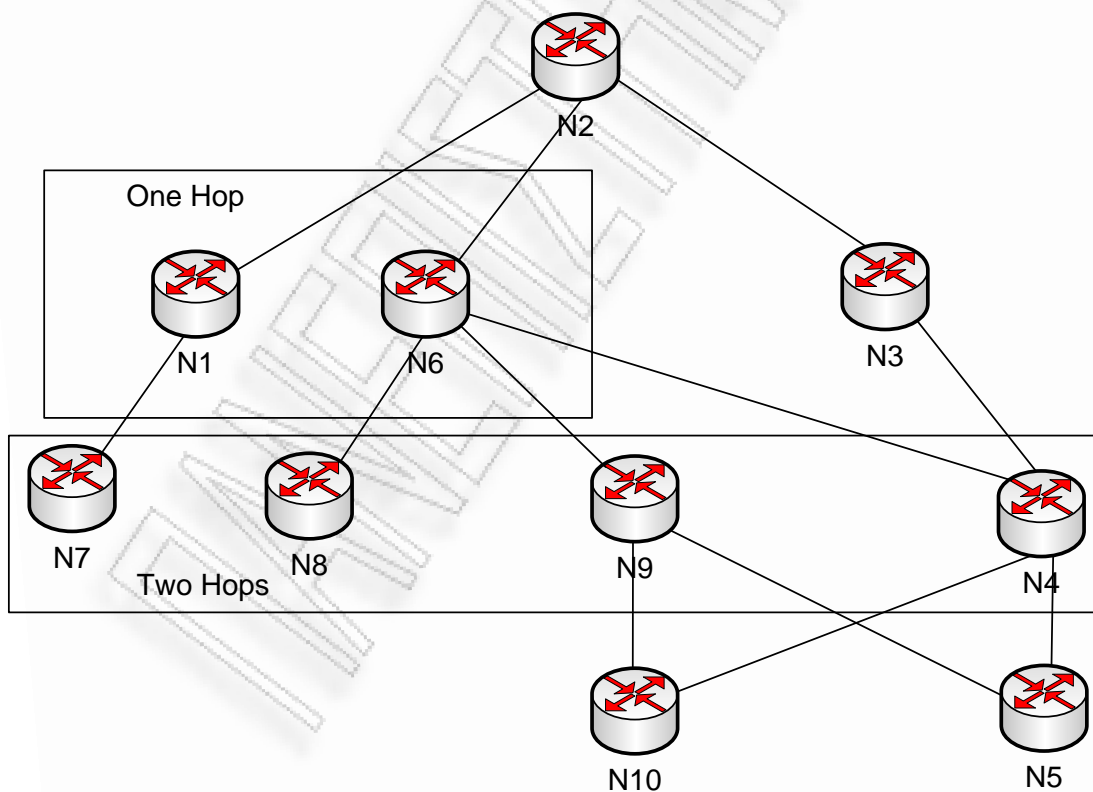
Για την ανίχνευση ενός γείτονα, κάθε κόμβος στέλνει περιοδικά μηνύματα αναγνώρισης (Hello messages) τα οποία περιέχουν πληροφορίες για το status της γραμμής. Ο αλγόριθμος επιλέγει μόνο απευθείας ζεύξεις ώστε να αποφευχθεί το πρόβλημα η μετάδοση των μηνυμάτων και σε κόμβους που δεν είναι γειτονικοί. Τα μηνύματα αναγνώρισης λαμβάνονται από όλους τους γειτονικούς κόμβους αλλά μεταδίδονται περαιτέρω. Με αυτό τον τρόπο κάθε κόμβος περιέχει πληροφορίες για κάθε γειτονικό μέχρι και δύο κόμβους μακριά βοηθώντας έτσι στην γρηγορότερη επιλογή διαδρομής.

2.6.5.2 Multipoint Relay Station

Κάθε κόμβος επιλέγει μόνος του τον αριθμό των MPRs που θα μεταδώσει τα μηνύματα αναγνώρισης. Η διαδικασία επιλογής MPRs έχει ως εξής:

- Ο κόμβος επιλέγει τα MPRs με τέτοιο τρόπο ώστε οι γειτονικοί κόμβοι που απέχουν ένα hop από τον αρχικό κόμβο να συνδέονται με όλους τους κόμβους που απέχουν δύο hops από τον αρχικό.

Για παράδειγμα στο σχήμα 5 ο κόμβος N2 θα επιλέξει τους κόμβους N1 και N6 ως MPR γιατί αυτοί οι δύο κόμβοι καλύπτουν όλους τους κόμβους του δικτύου που είναι δύο κόμβους μακριά από τον N2.



Σχήμα 5: Παράδειγμα επιλογής MPR

Τα MPRs επαναυπολογίζονται μόνο όταν ανιχνευτεί αλλαγή στους γειτονικούς κόμβους ή κάποια αλλαγή στους δύο hop κόμβους αυτών.

2.6.5.3 MPR Information Declaration

Κάθε κόμβος στο δίκτυο μεταδίδει περιοδικά μηνύματα ελέγχου τα οποία ονομάζονται Topology Control (TC). Τα μηνύματα αυτά είναι απαραίτητα ώστε να συγκεντρωθεί όλη η απαραίτητη πληροφορία για την προώθηση των πακέτων δρομολόγησης (routing packets). Ένα TC μήνυμα περιλαμβάνει ένα σειριακό αριθμό ο οποίος αυξάνεται κάθε φορά που αλλάζουν τα MPRs. Οι πληροφορίες από τα TC μηνύματα χρησιμοποιούνται ώστε να δημιουργηθεί ένας πίνακας τοπολογίας στο οποίο θα γίνονται οι εγγραφές για την τοπολογία του δικτύου. Ένας κόμβος παίρνει τις πληροφορίες από τα MPRs των άλλων κόμβων και βάση αυτής την πληροφορίας δημιουργεί τον πίνακα δρομολόγησης.

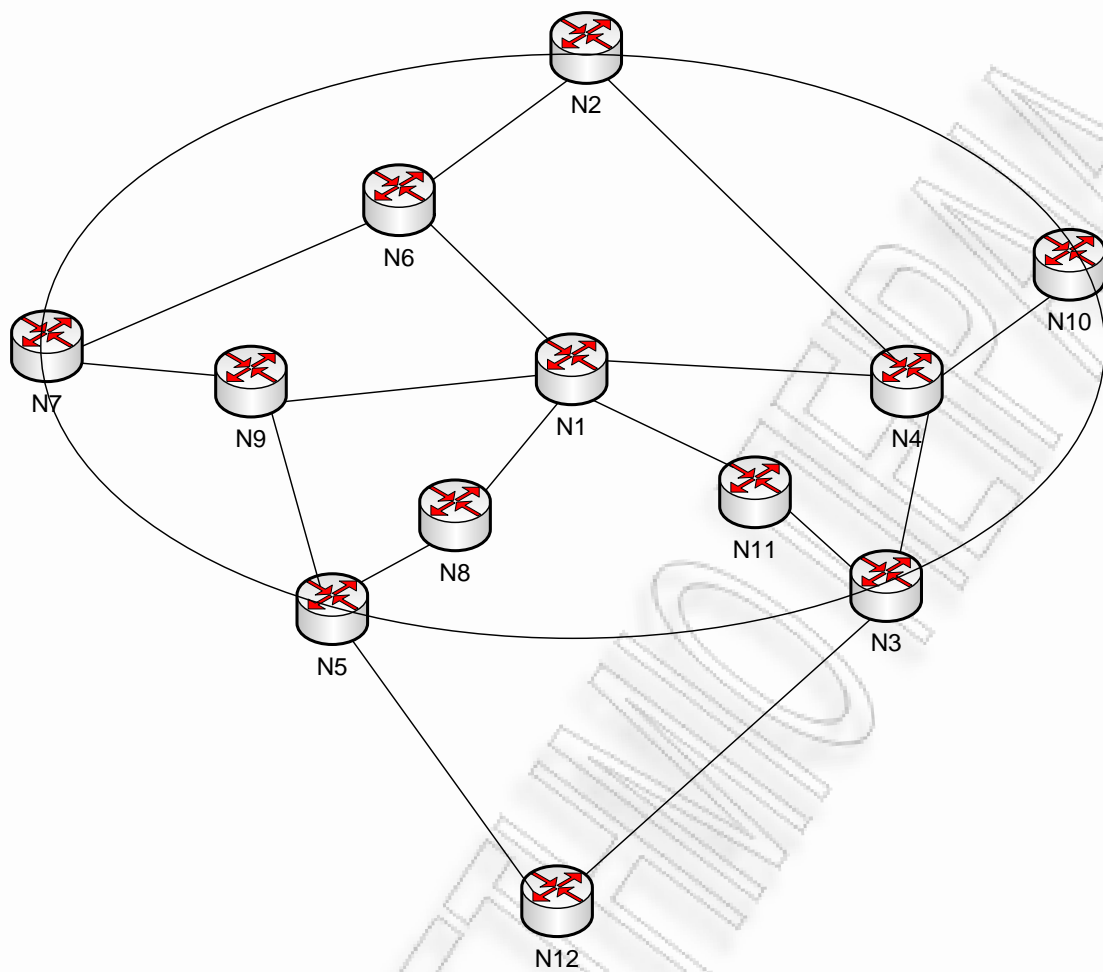
2.6.5.4 Routing Table Calculation

Κάθε κόμβος περιέχει τουλάχιστον μία διαδρομή που του επιτρέπει να μεταδώσει την πληροφορία από τον αρχικό στον τελικό κόμβο. Ο πίνακας δρομολόγησης κάθε κόμβου υπολογίζεται μέσω των πληροφοριών που λαμβάνει από τα TC μηνύματα. Οι εγγραφές στους πίνακες δρομολόγησης περιλαμβάνουν την διεύθυνση προορισμού (destination address), την διεύθυνση του επόμενου κόμβου (next-hop) και μια εκτιμώμενη απόσταση από τον αρχικό στον τελικό κόμβο. Οι πληροφορίες ανανεώνονται μόνο όταν συμβεί κάποια αλλαγή στην τοπολογία των γειτονικών κόμβων ή στην περίπτωση που υπάρχει καλύτερη διαδρομή από αυτή που χρησιμοποιείται.

2.6.6 Zone Routing Protocol

Θεωρείται υβριδικό πρωτόκολλο γιατί συνδυάζει τα πλεονεκτήματα proactive αλλά και reactive αλγορίθμων. Τα proactive πρωτόκολλα χαρακτηρίζονται για την χρήση υψηλού bandwidth για την διατήρηση της πληροφορίας δρομολόγησης ενώ τα reactive πρωτόκολλα υποφέρουν από τις υψηλές καθυστερήσεις για route requests αυξάνοντας έτσι το φόρτο του δικτύου μέχρι την εγκατάσταση διαδρομής. Ο ZRP συνδυάζει τα χαρακτηριστικά και των δύο κατηγοριών ώστε να αντιμετωπίσει όσο πιο αποδοτικά γίνεται όσα προβλήματα προκύψουν.

Κάθε κόμβος στο ZRP διατηρεί διαδρομές προς τους τελικούς κόμβους μέσα σε μια τοπική περιοχή που ονομάζεται routing zone. Το μέγεθος αυτής της περιοχής εξαρτάται από μία παράμετρο που ονομάζεται zone radius. Το σχήμα 6 δείχνει ένα routing zone με αρχικό κόμβο τον N1 με radius ίσο με 2 hops, δηλαδή η απόσταση μέχρι τα όρια της περιοχής θα είναι δύο κόμβοι.



Σχήμα 6: Routing Zone με radius=2

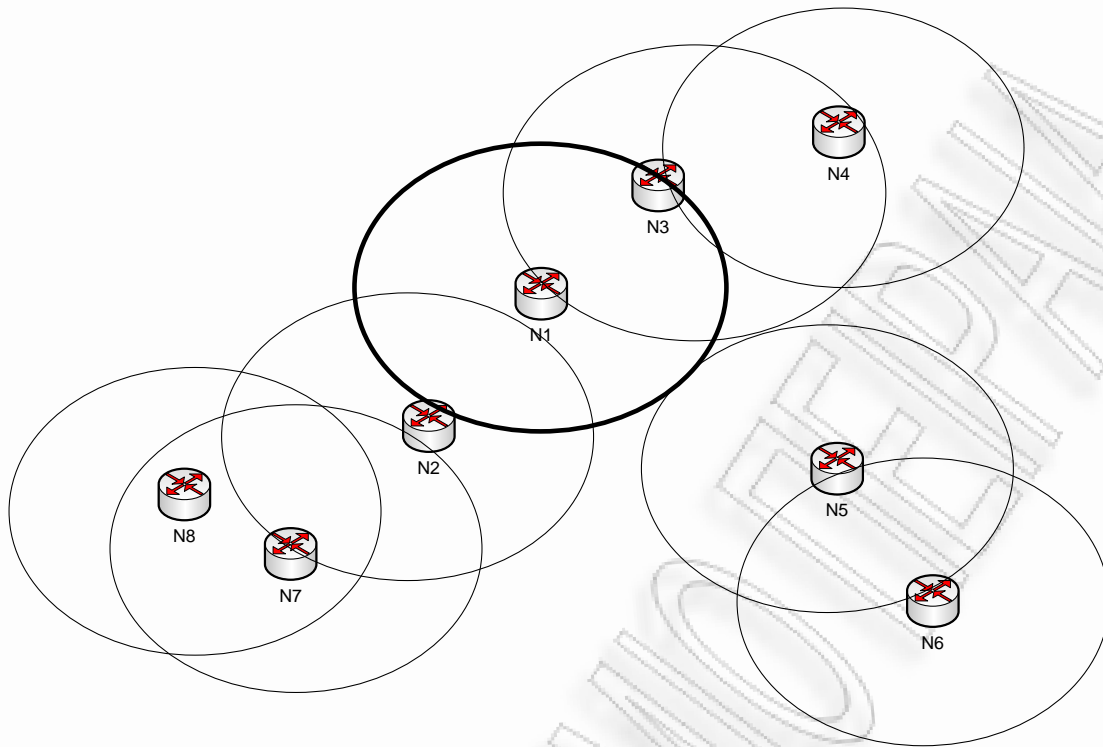
2.6.6.1 Intra-zone Routing Protocol

Στον ZRP κάθε κόμβος διατηρεί όλη την πληροφορία δρομολόγησης μέσα στο routing zone. Οι κόμβοι μαθαίνουν την τοπολογία της περιοχής μέσω του αλγορίθμου IARP. Ο αλγόριθμος αυτός μπορεί να χρησιμοποιήσει οποιαδήποτε κατηγορία αλγορίθμων (link state ή distance vector) για την δημιουργία της τοπολογίας. Κάθε περιοχή μπορεί να χρησιμοποιήσει το δικό της πρωτόκολλο αρκεί να περιορίζεται μόνο στους κόμβους που ανήκουν στο εκάστοτε routing zone. Με αυτόν τον τρόπο μια αλλαγή στην τοπολογία θα επηρεάσει μόνο τους κόμβους που βρίσκονται μέσα στο routing zone και όχι ολόκληρο το δίκτυο.

2.6.6.2 Inter-zone Routing Protocol

Χρησιμοποιείται στην περίπτωση που δεν βρεθεί διαδρομή προς τον τελικό κόμβο μέσα σε κάποια περιοχή. Τα μηνύματα αναζήτησης (route request packets) μεταδίδονται σε όλους τους ακριανούς κόμβους κάθε περιοχής (border nodes) οι οποίοι προωθούν το μήνυμα σε περίπτωση που δεν υπάρχει εγγραφή για τον τελικό προορισμό στον πίνακα δρομολόγησης. Τα μηνύματα που προωθούν οι ακριανοί κόμβοι κάθε περιοχής ονομάζονται bordercasting και παρέχονται από το πρωτόκολλο BRP (Bordercast Resolution Protocol). Στο σχήμα 7 φαίνεται ένα παράδειγμα εύρεσης διαδρομής από το N1 κόμβο στον N8.

Πρώτα ο N1 ελέγχει αν ο N8 είναι στο routing zone. Αφού ο N8 βρίσκεται εκτός περιοχής θα κάνει bordercast το μήνυμα προς τους ακριανούς κόμβους N2 και N3. Αφού και ο N2 και ο N3 διαπιστώσουν ότι ο N8 δεν είναι στο δικό τους περιοχή θα προωθήσει το μήνυμα προς τους ακριανούς κόμβους N7 και N4. ο N7 αναγνωρίζει τον N8 ως εσωτερικό κόμβο της περιοχής του και απαντάει στο μήνυμα αναζήτησης στέλνοντας την προτιμώμενη διαδρομή N1-N2-N7-N8.



Σχήμα 7: Τοπολογία ZRP

2.6.7 Hover

Αποτελεί έναν υβριδικό αλγόριθμο ο οποίος συνδυάζει τα στοιχεία του αλγορίθμου AODV. Τα στοιχεία που έχουν προστεθεί και τον διαφοροποιούν από τον AODV είναι:

- Σε ένα ασύρματο δίκτυο κορμού οι γειτονικοί κόμβοι μπορούν να συνδέονται με πολλαπλά links σε ορθογώνια κανάλια. Στον Hover υπάρχει μια διαδικασία επιλογής βέλτιστου link για κάθε κόμβο στο δίκτυο.
- Έχει εισαχθεί μια εκτίμηση ποιότητας γραμμής (link quality estimation) βασισμένη στα μηνύματα αναγνώρισης (HELLO packets) του αλγορίθμου AODV. Επειδή τα μηνύματα αναγνώρισης μεταδίδονται ανά τακτά χρονικά διαστήματα σε όλο το δίκτυο δεν προστίθεται επιπλέον overhead.

Ο αλγόριθμος Hover χρησιμοποιεί την ίδια διαδικασία αναζήτησης διαδρομής με αυτή του AODV. Παρόλα αυτά για να εξασφαλίσει ότι οι διαδρομές έχουν εγκατασταθεί δια μέσω των Mesh Routers αλλά και για την παροχή του βέλτιστου link, ο αλγόριθμος χρησιμοποιεί τους παρακάτω μηχανισμούς:

- Link Quality Estimation
- Optimal Link Selection

2.6.7.1 Link Quality Estimation

Κάθε κόμβος που τρέχει τον αλγόριθμο Hover διατηρεί ένα σει από links με τους γειτονικούς κόμβους. Ο μέγιστος αριθμός από links από των αριθμό των διαθέσιμων διεπαφών (interfaces) που υπάρχουν στους κόμβους. Ο αλγόριθμος επιβλέπει την ποιότητα αυτών των συνδέσεων ακούγοντας τα μηνύματα αναγνώρισης από τους γειτονικούς κόμβους. Όταν ένας κόμβος λάβει ένα Hello μήνυμα από έναν γειτονικό κόμβο, ενεργοποιεί το link από όπου στάλθηκε το μήνυμα και αποθηκεύει την IP διεύθυνση του. Όταν δεν ληφθεί κάποιο μήνυμα αναγνώρισης για συγκεκριμένο χρονικό διάστημα (επιτρεπόμενος αριθμός χαμένων Hellos * Hello interval) το link θεωρείται ως μη ενεργό. Αν το μη ενεργό link αποτελεί μέρος ενός ενεργού κόμβου (Mesh Router), ο αλγόριθμος αλλάζει το route ώστε να χρησιμοποιείται ένα ενεργό link για τον συγκεκριμένο γειτονικό κόμβο. Αυτός ο μηχανισμός είναι εξαιρετικά γρήγορος αφού γίνεται τοπικά και δεν χρειάζεται να γίνει αποστολή μηνυμάτων ελέγχου στο υπόλοιπο δίκτυο διατηρώντας έτσι το μικρό overhead.

Για να μπορεί να επιβλέπει την ποιότητα των συνδέσεων μεταξύ δύο κόμβων, ο αλγόριθμος μετράει τον αριθμό των Hello μηνυμάτων που λαμβάνονται αλλά και που χάνονται σε μια χρονική περίοδο. Ο συγκεκριμένος αλγόριθμος θέτει αυτό το χρονικό διάστημα ως $10 * \text{Hello Interval}$ δευτερόλεπτα. Ξεχωριστοί counters διατηρούνται για κάθε γειτονικό κόμβο και για κάθε διεπαφή. Αυτοί οι counters επιτρέπουν την εύρεση των παραμέτρων df και dr που είναι το ποσοστό λήψης των μηνυμάτων αναγνώρισης κατά τη διαδικασία προώθησης (forward path) αλλά και κατά τη διαδικασία αντίστροφης διαδρομής (reverse path).

Η παράμετρος dr είναι ο αριθμός των Hello μηνυμάτων που λήφθηκαν σε ένα χρονικό παράθυρο (time window) διαιρούμενο με τον αριθμό των αναμενόμενων Hello μηνυμάτων, επομένως δηλώνει την ποιότητα του link για την εισερχόμενη κίνηση. Η παράμετρος df αποτελεί την ποιότητα του link κατά τη διαδικασία προώθησης των μηνυμάτων (forward path).

Με τις δύο αυτές παραμέτρους, ένας κόμβος μπορεί να υπολογίσει την αναμενόμενη διάρκεια μετάδοσης υπό την μορφή count (ETX). Ο τύπος υπολογισμού είναι:

$$ETX = \frac{1}{df * dr}$$

Ο τύπος αυτός χρησιμοποιείται ως metric για την ποιότητα συνδέσμου και ισχύει για όλους τους κόμβους και τους γειτονικούς αυτών. Με την χρήση αυτού του τύπου μπορεί να επιλεγεί από τον αλγόριθμο ο βέλτιστος σύνδεσμος μεταξύ δύο κόμβων κατά την διάρκεια δημιουργίας του route.

2.6.7.2 Αναζήτηση Διαδρομής στον HOVER

Η βασική ιδέα του αλγορίθμου για την αναζήτηση διαδρομής είναι η γρήγορη εγκατάσταση διαδρομής η οποία μπορεί όμως να μην είναι και η βέλτιστη. Με αυτό τον τρόπο ο αλγόριθμος μειώνει σε σημαντικό βαθμό την καθυστέρηση για τη δημιουργία διαδρομής (path creation delay). Η διαδικασία αναζήτησης δεν τερματίζεται σε αυτό το σημείο καθώς ο αλγόριθμος συνεχίζει την αναζήτηση για μια πιο βέλτιστη διαδρομή και σε περίπτωση που βρεθεί το route αλλάζει με την διαδρομή που προϋπήρχε.

Η διαδικασία αυτή αποτελείται από δύο φάσεις:

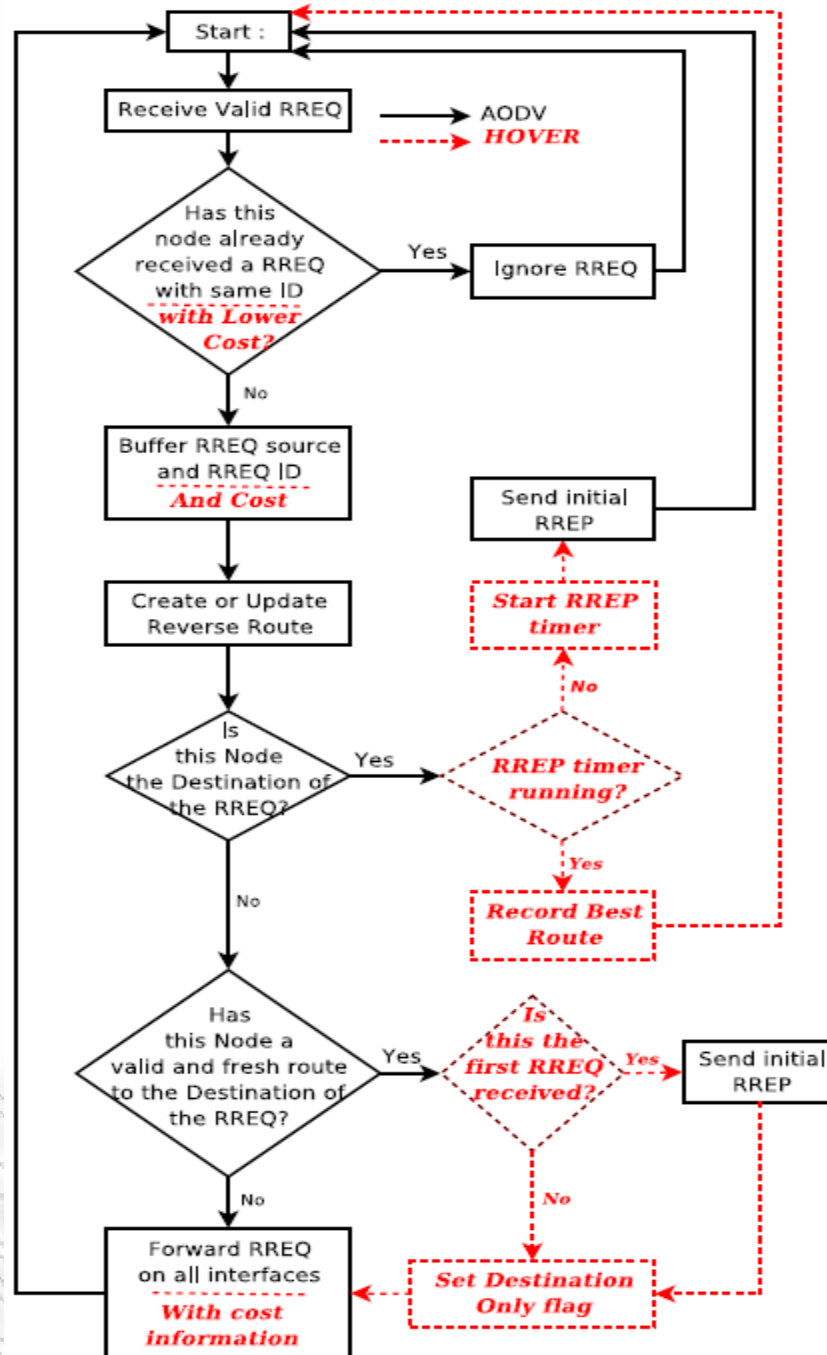
1. Εκτελείται κατά την διαδικασία των RREQs. Κάθε ενδιαμέσος κόμβος που λαμβάνει ένα RREQ μήνυμα ελέγχει πρώτα αν είναι μοναδικό ή διπλότυπο (duplicate). Σε περίπτωση που το μήνυμα είναι μοναδικό, ο κόμβος αποθηκεύει την πληροφορία στον πίνακα δρομολόγησης μαζί με το metric και το ID του μηνύματος. Στη συνέχεια δημιουργείται ένα reverse route προς τον αρχικό κόμβο που έστειλε τα RREQs.

Αν το RREQ είναι διπλότυπο, λαμβάνεται από τον κόμβο μόνο αν έχει μικρότερο κόστος διαδρομής από αυτό που είναι ήδη αποθηκευμένο στον πίνακα δρομολόγησης. Επομένως ένας ενδιαμέσος κόμβος μπορεί να προωθήσει πολλά ίδια RREQs αν το κόστος διαδρομής είναι χαμηλότερο από αυτό που έλαβε.

Αν ένας ενδιαμέσος κόμβος έχει αποθηκευμένη μια διαδρομή για τον τελικό κόμβο μπορεί να στείλει **μόνο ένα RREP μήνυμα** στο πρώτο RREQ με το ίδιο ID. Με αυτό τον τρόπο αποφεύγεται η αποστολή απαντητικών μηνυμάτων για όλα τα RREQs με το ίδιο ID από τους ενδιαμέσους κόμβους. Επομένως μειώνεται σημαντικά το επιπλέον overhead με την ανούσια αποστολή των απαντητικών μηνυμάτων. Για τα υπόλοιπα διπλότυπα RREQs οι ενδιαμέσοι κόμβοι θέτουν ένα **Destination-Only Flag** και προωθούν το μήνυμα.

Για να γίνει η εγκατάσταση μιας αρχικής διαδρομής όσο το δυνατόν γρηγορότερα ο τελικός κόμβος απαντάει αμέσως στο πρώτο RREQ μήνυμα που θα λάβει. Για να επιβεβαιωθεί ότι η διαδρομή που δημιουργείται είναι προσωρινή και κατά πάσα πιθανότητα δεν είναι η βέλτιστη το o-flag στην επικεφαλίδα του RREP θέτεται ως False.

Στον HOVER μόλις σταλθεί ένα αρχικό RREP, ο τελικός κόμβος δέχεται και αποθηκεύει τοπικά (buffer) όλα τα RREQs με το ίδιο μέχρι τη λήξη ενός ορισμένου από αυτόν χρονικού διαστήματος. Με την λήξη του timer ο τελικός κόμβος επιλέγει το βέλτιστο RREQ όπως για παράδειγμα αυτό με το μικρότερο κόστος διαδρομής σύμφωνα με το metric που έχουμε ορίσει και απαντάει με ένα βέλτιστο RREP. Το παρακάτω σχήμα δείχνει το διάγραμμα ροής της πρώτης φάσης



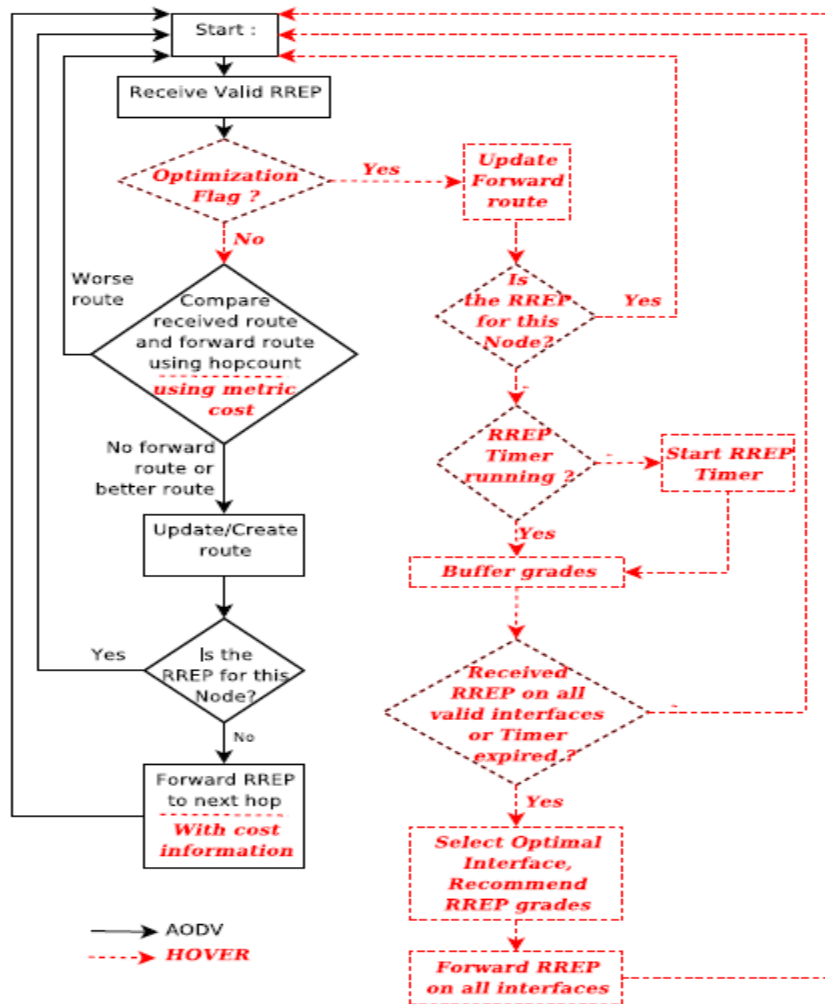
2. αυτόν το τρόπο κάθε κόμβος προτείνει ένα σύνδεσμο που μπορεί να προωθήσει τα μηνύματα στον επόμενο κόμβο.
 Η αποστολή πολλαπλών RREPs για μια συγκεκριμένη διαδρομή δίνει την ευκαιρία στο κόμβο λήψης να επιλέξει τον βέλτιστο σύνδεσμο-link.

Το αρχικό RREP που στέλνεται συμπεριφέρεται ως ένα κανονικό RREP με τη μόνη διαφορά ότι το field που περιέχει το κόστος διαδρομής ανανεώνεται σύμφωνα με τους ίδιους κανόνες που

χρησιμοποιούνται και για τα μηνύματα RREQs. Αν το RREP ληφθεί από έναν ενδιάμεσο κόμβο που έχει ήδη αποθηκευμένη μια διαδρομή, συγκρίνει το κόστος που υπάρχει ήδη στο πίνακα δρομολόγησης με αυτό του RREP μηνύματος και στην περίπτωση που είναι μικρότερος το αποθηκεύει και προωθεί το μήνυμα. Σε αντίθετη περίπτωση το RREP απλά απορρίπτεται.

Όταν λαμβάνεται ένα RREP μήνυμα όλοι οι κόμβοι ελέγχουν πρώτα το status του o-flag . Έπειτα αποθηκεύουν τοπικά τα RREP με το ίδιο ID μέχρι τη λήξη του χρονοδιαστήματος που έχει οριστεί από τον κόμβο.

Αν έχουν ληφθεί περισσότερα από ένα βέλτιστα RREP ο κόμβος επιλέγει ένα από αυτά σύμφωνα με την ποιότητα συνδέσμου-link που έχει υπολογιστεί. Τα παρακάτω σχήμα δείχνει το διάγραμμα ροής της δεύτερης φάσης.



Σχήμα 8: Διάγραμμα ροής δεύτερης φάσης.

2.7 ΣΥΓΚΡΙΣΗ ΠΡΩΤΟΚΟΛΛΩΝ SINGLE PATH

Η ανάλυση των παραπάνω αλγορίθμων έγινε με σκοπό να αναλυθούν αναλυτικά οι τεχνικές δρομολόγησης σε δίκτυα WMN και ειδικότερα στο backhaul δίκτυο. Ο παρακάτω πίνακας που συγκρίνει τα χαρακτηριστικά των πρωτοκόλλων θα μας βοηθήσει να βγάλουμε σημαντικά συμπεράσματα για το λόγο που προτείνεται η τεχνική multipath για την καλύτερη απόδοση του δικτύου. Στα επόμενα κεφάλαια θα γίνει ανάλυση και των multipath πρωτοκόλλων δρομολόγησης τα οποία έχουν ενσωματώσει μεγάλο μέρος από την αρχιτεκτονική που υπάρχει στους single path αλγορίθμους.

Αλγόριθμος	Κατηγορία	Throughput	Delay - QoS	Interference	Limitations
DSDV	Distance-Vector	Υψηλό λόγω χρήσης routing updates για εισαγωγή νέας διαδρομής (+)	Δεν υπολογίζεται ως παράμετρος (-)	Δεν υπολογίζεται ως παράμετρος (-)	Χρησιμοποιεί μόνο hop count για την εύρεση διαδρομής.
QoS Algorithm	Distance-Vector	Δεν υπολογίζει ως παράμετρο το bandwidth κάθε γραμμής.	Υπολογίζει συγκεκριμένες παραμέτρους QoS που πρέπει να υποστηρίζει κάθε κόμβος.	Δεν υπολογίζεται ως παράμετρος (-)	Χρησιμοποιεί ως metric 3 κατηγορίες παραμέτρων ποιότητας ως threshold
Min-Hop Algorithm	Distance-Vector	Υψηλό διότι αποτελεί απλό αλγόριθμο (+)	Υψηλό γιατί αυξάνει το load σε links που υποστηρίζουν το metric αφήνοντας το υπόλοιπο δίκτυο άθικτο. (-)	Δεν υπολογίζεται ως παράμετρος (-)	Χρησιμοποιεί ως metric τα feasible links και στέλνει την κίνηση μόνο προς αυτή τη διαδρομή.
Widest Shortest path Algorithm	Distance-Vector	Εξέλιξη του Min-Hop. Κάνει load balance και διαλέγει τη διαδρομή με το μεγαλύτερο bandwidth. (+)	Με το load balance αυξάνεται ο φόρτος του δικτύου. (-)	Δεν υπολογίζεται ως παράμετρος (-)	Ίδια μειονεκτήματα με αυτά του Min-Hop.
Minimum Interference Routing	Link-state	Δεν υπολογίζει ως παράμετρο το bandwidth κάθε	Δεν υπολογίζεται	Αποτελεί παράμετρο επιλογής	Θέτει βάρη σε κάθε σύνδεσμο με

Algorithm		γραμμής. (-)	ως παράμετρος (-)	διαδρομής. Επιλέγεται η διαδρομή με τις μικρότερες παρεμβολές. (+)	στατικό τρόπο επομένως γίνεται δύσκολα η αναδιανομή βαρών. Δεν υπολογίζει πως μια υπάρχουσα σύνδεση επηρεάζει αρνητικά μελλοντικές συνδέσεις.
Bandwidth Routing Protocol	Distance-Vector	Αποτελείται από δύο φάσεις μία εκ των δύο υπολογίζει το διαθέσιμο bandwidth μεταξύ αρχικού-τελικού κόμβου. (+)	Η δεύτερη φάση υπολογίζει το QoS flow που χρειάζεται κάθε σύνδεση. (+)	Δεν υπολογίζεται ως παράμετρος (-)	Έχει μειωμένη ευελιξία γιατί δεν επιτρέπει την εισαγωγή νέων κόμβων μετα την εγκατάσταση της σύνδεσης.
On- Demand QoS Routing Protocol	Distance-Vector	Υπολογίζει την κατανομή bandwidth και για εναλλακτικές διαδρομές μειώνοντας το χρόνο αναπροσαρμογής. (+)	Βρίσκει αν οι εναλλακτικές διαδρομές ικανοποιούν το QoS. (+)	Δεν υπολογίζεται ως παράμετρος (-)	Αποτελεί εξέλιξη του BR υιοθετώντας έτσι και τα μειονεκτήματά του. Επίσης η on-demand φύση του αλγορίθμου οδηγεί σε μεγαλύτερο χρόνο εγκατάστασης της σύνδεσης.
AODV	Distance-Vector	Λόγω των πολλών μηνυμάτων (broadcast και unicast) αυξάνεται το overhead του δικτύου. (-)	χρειάζεται περιοδικά routing updates (-)	Δεν υπολογίζεται ως παράμετρος (-)	Εξαιτίας των μηνυμάτων που στέλνονται υπάρχει μεγάλο utilization επομένως καταναλώνεται μεγάλος αριθμός πόρων.

DSR	Distance-Vector	Δεν χρειάζεται περιοδικά routing updates επομένως μειώνει το overhead.	Χρησιμοποιεί stale routes πράγμα που αυξάνει σημαντικά την καθυστέρηση και την απόδοση. (-)	Δεν υπολογίζεται ως παράμετρος (-)	Χρησιμοποιεί και πόρους από το Mac επίπεδο σε περίπτωση που ανιχνευθεί σφάλμα σε κάποιο σύνδεσμο.
ZRP	Link-state	Λόγω της proactive φύσης του αλγορίθμου υπάρχει μεγάλη κίνηση routing πακέτων που στέλνονται για τα μηνύματα αναγνώρισης κόμβων. (-)	Δεν παρουσιάζει υψηλό delay. (+)	Επειδή οι κόμβοι χωρίζονται σε περιοχές αποφεύγονται οι παρεμβολές σε κόμβους που δεν ανήκουν στην ίδια ζώνη. (+)	Δυσκολεύει ο σχεδιασμός του δικτύου με την εισαγωγή routing zones.
OLSR	Link-state	Μέσω των MPR που χρησιμοποιούνται μειώνεται ο φόρτος του δικτύου γιατί τα broadcast μηνύματα περιορίζονται στους γειτονικούς κόμβους. (+)	Λόγω της proactive φύσης του αλγορίθμου δεν παρουσιάζει υψηλό delay. (+)	Με την χρήση των MPR αποφεύγονται οι παρεμβολές. (+)	Έχει το μειονέκτημα ότι αναφέρεται κυρίως σε μικρά δίκτυα αφού τα MPR πρέπει να έχουν πληροφορία μέχρι τον τελικό κόμβο σε απόσταση μέχρι 2 hops.
HOVER	Distance-Vector	Παρουσιάζει υψηλό throughput με την χρήση γρήγορης εγκατάστασης. (+)	Παρουσιάζει χαμηλό delay (+)		Καταναλώνει μεγάλο ποσοστό από πόρους εξαιτίας της γρήγορης εγκατάστασης διαδρομής και έπειτα την αναζήτηση βέλτιστης διαδρομής.

Στον παραπάνω πίνακα παρατηρούμε ότι πολλοί αλγόριθμοι δεν υπολογίζουν μια σημαντική παράμετρος που είναι το throughput αλλά και η χρησιμοποίηση της γραμμής (utilization). Στα ασύρματα δίκτυα το throughput αποτελεί πρωτεύων ζήτημα. Για τον λόγο αυτό αναπτύχθηκε η τεχνική multiroute ως ένας τρόπος για την μείωση των παρεμβολών καθώς επίσης και για την καλύτερη αξιοποίηση του bandwidth αφού με πολλαπλές διαδρομές δεν υπάρχει υπερφόρτωση σε συγκεκριμένες συνδέσεις και κόμβους.

Στα επόμενα κεφάλαια αναλύεται πιο συγκεκριμένα η τεχνική multipath καθώς και αλγόριθμοι που χρησιμοποιούν αυτή την τεχνική.

ΚΕΦΑΛΑΙΟ 3^ο : ΔΡΟΜΟΛΟΓΗΣΗ ΠΟΛΛΑΠΛΩΝ ΔΙΑΔΡΟΜΩΝ(MULTIPATH ROUTING)

3.1 ΕΙΣΑΓΩΓΗ

Όλα τα πρωτόκολλα που αναλύθηκαν παραπάνω είναι συγκεκριμένης διαδρομής που σημαίνει ότι ένα και μόνο ένα βέλτιστο path επιλέγεται μεταξύ αρχικού και τελικού κόμβου. Με την τεχνική των πολλαπλών διαδρομών επιτυγχάνεται η χρησιμοποίηση και άλλων διαθέσιμων paths οι οποίοι μπορεί να μην αποτελούν βέλτιστες διαδρομές. Τα πλεονεκτήματα της τεχνικής multipath παρουσιάζονται παρακάτω:

1. *Fault Tolerance*: εισάγεται η έννοια του πλεονασμού διαδρομών (redundancy) και με αυτό τον τρόπο παρέχονται backup διαδρομές στο δίκτυο σε περίπτωση που σε μερικά links παρουσιαστούν προβλήματα συνδεσιμότητας.
2. *Αύξηση του throughput*: σε ένα mesh δίκτυο κάποιες συνδέσεις μπορεί να έχουν περιορισμένο bandwidth. Η δρομολόγηση μέσω μίας και μόνο διαδρομής μπορεί να αδυνατεί να παρέχει το απαραίτητο εύρος ζώνης για την κάλυψη της ποιότητας υπηρεσίας που απαιτεί ο τελικός χρήστης. Για αυτό το λόγο η δρομολόγηση μέσω πολλαπλών διαδρομών είναι μια καλή τεχνική για την ικανοποίηση του bandwidth ακόμα και των πιο απαιτητικών υπηρεσιών και εφαρμογών. Με την αύξηση του throughput υπάρχει μικρότερη end-to-end καθυστέρηση και βελτιώνεται το QoS.
3. *Load Balancing*: δεδομένου ότι η διανομή της κίνησης (traffic) δεν είναι ίση σε όλες τις συνδέσεις στο δίκτυο, η διάδοση της κίνησης κατά μήκος των πολλαπλάσιων διαδρομών μπορεί να μειώσει αισθητά τη συμφόρηση σε μερικές συνδέσεις.
4. *Ανεκτικότητα λαθών* : Τα πρωτόκολλα πολλαπλών διαδρομών μπορούν να παρέχουν ανεκτικότητα λαθών με το να κάνουν κατανομή της κίνησης στις συνδέσεις με τα μικρότερο ποσοστό στο κώδικα διόρθωσης σφαλμάτων.
5. *Ασφάλεια*: Με την χρήση πολλαπλών διαδρομών γίνεται πιο περίπλοκη η επίθεση από τρίτους αφού ο κακόβουλος χρήστης θα πρέπει να ανακαλύψει όλες τις διαδρομές που ακολουθούν τα πακέτα.

3.2 Routing Metrics

Η φάση εγκαθίδρυσης μιας διαδρομής περιλαμβάνει την επιλογή των κατάλληλων paths μέσα από όλα τα διαθέσιμα. Σε περίπτωση που υπάρχουν πολλές διαθέσιμες διαδρομές επιλέγεται ένας περιορισμένος αριθμός. Η αξιολόγηση των διαθέσιμων διαδρομών αναλύεται παρακάτω έχοντας ως παράγοντα τα διάφορα metrics που χρησιμοποιούνται από πολλούς αλγόριθμους δρομολόγησης.

- Το hop count αποτελεί ένα από τα πιο δημοφιλή metrics αφού επιτρέπει γρήγορη εύρεση διαδρομών. Συνήθως χρησιμοποιείται για να μην αυξηθεί η πολυπλοκότητα του αλγορίθμου και είναι και πιο εύκολο στην υλοποίηση του.
- Η αξιοπιστία της διαδρομής και η ποιότητα της σύνδεσης αποτελούν metrics που χρησιμοποιούνται για την επίτευξη υψηλής επίδοσης σε αλγόριθμους που ως στόχο έχουν να κρατήσουν ψηλά το QoS. Ως αξιοπιστία διαδρομής θεωρούμε την πιθανότητα επιτυχημένης μετάδοσης δεδομένων μεταξύ δύο κόμβων σε συγκεκριμένο χρονικό διάστημα. Οι παράγοντες από τους οποίους αξιολογείται κατά πόσο η διαδρομή είναι αξιόπιστη είναι:
 - ETX (Expected Transmission Count): είναι ο αριθμός που χρειάζεται ένας κόμβος για να μεταδώσει επιτυχώς ένα πακέτο.
 - ML (Minimum Loss): αναφέρεται στη διαδρομή με την μικρότερη πιθανότητα απόρριψης ενός πακέτου.
 - ETT (Expected Transmission Time): θεωρεί την ποιότητα της σύνδεσης ως μια συνάρτηση χρόνου που το πακέτο χρειάζεται για να μεταδοθεί σε όλους τους γειτονικούς κόμβους.
 - mETX (modified ETX): υπολογίζει την πιθανότητα λάθους μιας σύνδεσης. Έχει ως στόχο να λύσει το πρόβλημα της γρήγορης διακύμανσης στην ποιότητα μιας ασύρματης σύνδεσης.
 - ENT (Effective Number of Transmissions): μετράει τον αριθμό των επιτυχημένων αναμεταδόσεων σε μια σύνδεση.
 - iAWARE κάνει μια εκτίμηση για το κατά πόσο είναι κατειλημμένο το μέσο μετάδοσης εξαιτίας των αναμεταδόσεων που γίνονται μεταξύ των γειτονικών κόμβων που παρεμβάλλονται.

Αν και υπάρχουν πολλά metrics που μπορούν να χρησιμοποιηθούν στις τεχνικές multipath τα πιο πολλά πρωτόκολλα δρομολόγησης επιλέγουν τα πιο απλά και εύχρηστα metrics όπως για παράδειγμα το ETX και το ETT.

Άλλες τεχνικές υποστηρίζουν ένα συνδυασμό από metrics όταν έχουν ως στόχο να υποστηρίξουν υψηλό QoS όπου η επιλογή των διαδρομών γίνεται μέσω κάποιων QoS απαιτήσεων που πρέπει να υποστηρίζονται.

Σε ότι αφορά τα πρωτόκολλα δρομολόγησης πολλαπλών διαδρομών οι παρακάτω τεχνική είναι οι πιο κατάλληλη.

Route Coupling: καθορίζει την βαθμό παρεμβολών μεταξύ δύο ή περισσότερων διαδρομών. Αν δύο διαδρομές μοιράζονται κάποια σύνδεση τότε είναι Hcoupled (Highly Coupled) . Επομένως οι διαδρομές που δεν είναι coupled είναι οι καταλληλότερες για μετάδοση πακέτων

πράγμα που σημαίνει ότι αλγόριθμος θα πρέπει να επιλέγει διαδρομές που δεν παρεμβάλλονται μεταξύ τους.

3.3 PROACTIVE MULTIPATH ROUTING

Οι proactive αλγόριθμοι δρομολόγησης πολλαπλών διαδρομών χαρακτηρίζονται από δύο φάσεις:

- Την εγκατάσταση της σύνδεσης (network setup)
- Την διατήρηση της σύνδεσης (network maintenance)

Κατά την διάρκεια της εγκατάστασης δημιουργούνται οι πίνακες δρομολόγησης ώστε να αρχίσει η προώθηση των πακέτων, ενώ κατά την διάρκεια της διατήρησης της σύνδεσης γίνονται οι απαραίτητες αλλαγές στους πίνακες δρομολόγησης σε περίπτωση που γίνει κάποια αλλαγή στην τοπολογία του δικτύου.

3.4 REACTIVE MULTIPATH ROUTING

Η λειτουργία αυτών των πρωτοκόλλων δρομολόγησης μπορεί να διασπαστεί σε τρία μέρη:

1. *Εγκαθίδρυση της σύνδεσης*: Αποτελείται από την εύρεση πολλαπλών διαδρομών από τον αρχικό προς τον τελικό κόμβο. Αυτό πραγματοποιείται με την εκπομπή πολλών RREQ μηνυμάτων με μοναδικό αύξοντα αριθμό. Και για να αποφευχθεί η συμφόρηση οι κόμβοι κάνουν drop τα πακέτα που λαμβάνουν με τον ίδιο αύξοντα αριθμό. Τα μηνύματα θα συνεχίσουν να εκπέμπονται μέχρι να φτάσουν στον τελικό κόμβο όπου και θα σταλεί ένα ή περισσότερα RREPs.

2. *Διατήρησης της σύνδεσης*: Έχει να κάνει με την εύρεση εναλλακτικών διαδρομών σε περίπτωση που μία ή περισσότερες διαδρομές πέσουν. Από τους πολλαπλούς τρόπους που μπορεί να γίνει η διατήρηση της σύνδεσης αναλύουμε τους δύο πιο δημοφιλέστερους τρόπους.

- **Περιοδικά Hello μηνύματα**: κάθε κόμβος εκπέμπει περιοδικά μηνύματα, αν ένας κόμβος δεν λαμβάνει μηνύματα αναγνώρισης από τον γειτονικό του κόμβο για ένα συγκεκριμένο χρονικό διάστημα
- **Mac acknowledgments**: είναι σχεδόν η ίδια διαδικασία με αυτή που περιγράφηκε παραπάνω. Η μόνη διαφορά είναι ότι αν ο γειτονικός κόμβος δεν λάβει κάποιο μήνυμα επιβεβαίωσης από για ένα συγκεκριμένο αριθμό προσπαθειών (retries) τότε θεωρεί ότι ο κόμβος είναι εκτός λειτουργίας και στέλνεται αντίστοιχο μήνυμα λάθους.

Εάν οποιοσδήποτε από τους προηγούμενους μηχανισμούς προτείνει ότι υπάρχει μια αποτυχία διαδρομών, προκειμένου να βρεθεί μια νέα διαδρομή, μια νέα διαδικασία ανακαλύψεων διαδρομών πρέπει να εκτελεσθεί. Στην περίπτωση πολλαπλών διαδρομών, ένα route discovery μπορεί να προκληθεί κάθε φορά που αποτυγχάνει μια από τις διαδρομές ή όταν όλες οι διαδρομές έχει αποτύχει. Το μειονέκτημα που παρουσιάζει όμως αυτή η τεχνική έχει να κάνει με την καθυστέρηση που

δημιουργείται γιατί ο αλγόριθμος αναμένει πρόβλημα κάποιας διαδρομής ώστε να ξεκινήσει το route discovery, διαδικασία η οποία μπορεί να κρατήσει αρκετά και να αυξήσει το overhead. Επιπλέον αν ο αλγόριθμος περιμένει να πέσουν όλες οι διαδρομές μέχρι να ξεκινήσει την διαδικασία του route discovery θα υπάρχει αρκετή αναμονή μέχρι να δημιουργηθούν και να εγκατασταθούν οι νέες διαδρομές. Μια εναλλακτική προσέγγιση θα μπορούσε να ήταν η δημιουργία ενός threshold χαλασμένων διαδρομών μέχρι να ξεκινήσει η διαδικασία route discovery. Αυτή η προσέγγιση θα μπορούσε να δημιουργηθεί για ισοσταθμίσει λίγο τις δύο εναλλακτικές επιλογές που παρουσιάστηκαν παραπάνω.

Για να αποτραπεί η αφαίρεση διαδρομών ως αποτέλεσμα της αδράνειας, το route utilization γίνεται σημαντικό γιατί τα on-demand πρωτόκολλα δρομολόγησης τείνουν να ρίξουν τις διαδρομές εάν δεν έχουν χρησιμοποιηθεί για ένα συγκεκριμένο ποσό του χρόνου.

3. *Traffic Allocation*: Μόλις επιλέξει ο κόμβος πηγής ένα σύνολο διαδρομών στον προορισμό, μπορεί να αρχίσει να στέλνει τα στοιχεία στον προορισμό κατά μήκος αυτών των πορειών. Για να επιλέξει τον τρόπο που το στοιχείο διανέμεται μεταξύ των υπάρχουσών διαδρομών, μια στρατηγική κατανομής απαιτείται. Υπάρχουν δύο τρόποι κατανομής: *granularity* και *scheduling*. Η πρώτη καθορίζει την μικρότερη μονάδα πληροφορίας που διανέμεται σε κάθε διαδρομή:

- Ανά ζευγάρι source-destination: χρησιμοποιεί την ίδια διαδρομή για να προωθήσει την κίνηση που ανήκει σε ένα ζευγάρι.
- Ανά σύνδεση: χρησιμοποιείται για την κατανομή της κίνησης μια σύνδεσης σε μια μονή διαδρομή.
- Ανά πακέτο: χρησιμοποιείται για την διανομή των πακέτων σε πολλαπλούς προορισμούς.
- Ανά segment: χρησιμοποιείται για την διάσπαση του πακέτου σε μικρότερα κομμάτια και την αποστολή αυτών σε διαφορετικές διαδρομές.

Οι διαδρομές που βρίσκονται από την διαδικασία route discovery μπορούν να γίνουν scheduled χρησιμοποιώντας τις παρακάτω στρατηγικές:

- Round Robin: Σε αυτήν την στρατηγική ένας κόμβος στέλνει ακριβώς κάθε πακέτο χρησιμοποιώντας μια διαφορετική διαδρομή. Η στρατηγική αυτή όμως έχει το σημαντικό μειονέκτημα ότι δεν μπορεί να διαχειριστεί πακέτα που έρχονται εκτός σειράς κυρίως σε ασύρματες συνδέσεις. Τα πακέτα εκτός σειράς μπορούν να δημιουργήσουν μεγάλους buffer με αποτέλεσμα την απώλεια σημαντικών πακέτων. Εντούτοις μπορεί να χρησιμοποιηθεί για load balancing γιατί το φορτίο θα κατανεμηθεί ομοιόμορφα.
- Congestion Aware: Αυτό το σχέδιο σχεδιασμού προτείνει να στείλει τη κίνηση σε μη-κορεσμένη συνδέσεις για να αποφύγει τις απώλειες και τις πρόσθετες καθυστερήσεις. Οι κορεσμένες διαδρομές μπορούν να μετρηθούν, παραδείγματος χάριν, από το μέσο μήκος ουρών αναμονής και βελτίωσης της απόδοσης.

- **Backup Path:** Κατά τη διάρκεια της διαδικασίας ανακαλύψεων διαδρομών, μια πηγή μπορεί να καθιερώσει μια αρχική πορεία όπως και διάφορες εφεδρικές πορείες στον επιθυμητό προορισμό. Το Multiroute μπορεί να χρησιμοποιηθεί για να μειώσει την καθυστέρηση στην ανάκτηση από μια αποτυχία, χρησιμοποιώντας κατά συνέπεια τις εφεδρικές πορείες από το να διαγράψει διαδρομές αν πέσει ένας κόμβος ή μια σύνδεση.
- **Unequal cost scheme:** προτείνει να διανείμει την κίνηση βασισμένο σε μια κοινή μέτρηση της απόστασης και ποιότητας συνδέσεων. Η πορεία με τις καλύτερες μετρικές επιλέγεται σύμφωνα με μια διανομή πιθανότητας. Το πλεονέκτημα αυτής της προσέγγισης είναι ότι, καθώς όλες οι διαδρομές χρησιμοποιούνται, δεν αφαιρούνται από τη δρομολόγηση των πινάκων, η οποία αποτρέπει τις διαδικασίες ανακαλύψεων διαδρομών
- **Concurrent Delivery:** αναφέρεται στην αποστολή ταυτόχρονης κίνησης σε περισσότερες από μια διαδρομές.

3.5 MULTIPATH ROUTING PROTOCOLS FOR WIRELESS BACKHAUL NETWORKS

Όπως και στους single route αλγόριθμους, θα αναλύσουμε τις τεχνικές δρομολόγησης και θα τις χωρίσουμε σε δύο κατηγορίες: τους proactive και τους reactive αλγόριθμους.

3.5.1 Proactive Multiroute Algorithms

3.5.1.1 MOLSR

Multipath Optimized Link State Routing protocol (MOLSR) βασίζεται σε μεγάλο βαθμό στην αρχιτεκτονική του OLSR και στοχεύει στην επίτευξη χαμηλότερου delay χρησιμοποιώντας πολλαπλές διαδρομές. Η διαφορά του MOLSR από τον αρχικό αλγόριθμο έχει να κάνει με την φάση εγκατάστασης (network setup). Παρακάτω αναλύονται και συγκρίνονται αυτές οι διαφορές.

- Στον OLSR αλγόριθμο δημιουργούνται δύο πίνακες: ο πίνακας τοπολογίας και ο πίνακας δρομολόγησης. Ο πίνακας τοπολογίας καταγράφει την τοπολογία του συνολικού δικτύου αλλά δεν αποθηκεύει κάποιες link state παραμέτρους καταστρώντας έτσι την δρομολόγηση πολύ δύσκολη με την χρήση μόνο αυτού του πίνακα. Στον MOLSR αλγόριθμο παρέχονται πληροφορίες για την καθυστέρηση και το SNR κάθε σύνδεσης στον πίνακα τοπολογίας επομένως ο πίνακας δρομολόγησης κατασκευάζεται έχοντας ως πρότυπο τον πίνακα τοπολογίας και αποθηκεύει όχι περισσότερες από δύο διαδρομές για κάθε διεύθυνση προορισμού. Αυτές οι διαδρομές αντιπροσωπεύουν τις καλύτερες δυνατές για το συγκεκριμένο χρονικό διάστημα.
Η κύρια διαφορά των δύο αλγόριθμων στην διαδικασία υπολογισμού των διαδρομών και της δημιουργίας του πίνακα δρομολόγησης.

3.5.1.2 OLSR-based multipath routing

Στην συγκεκριμένη παράγραφο αναλύονται τα σημαντικότερα χαρακτηριστικά και άλλων αλγορίθμων που σχετίζονται με τον OLSR. Στον QOLSR οι διαδρομές υπολογίζονται ώστε να ικανοποιούνται όλες οι απαιτήσεις εύρους ζώνης αφού ο αλγόριθμος έχει ως κύριο χαρακτηριστικό το QoS. Ένας άλλος αλγόριθμος ο MP-OLSR έχει ως χαρακτηριστικό τα κόστη κάθε σύνδεσης.

Όλοι οι παραπάνω αλγόριθμοι χρησιμοποιούν τον OLSR ως βάση επομένως οι λειτουργίες του χωρίζονται σε δύο μέρη: δημιουργία ενός πίνακα που θα κρατάει όλη την τοπολογία του δικτύου και την δημιουργία, διατήρηση και ανανέωση ενός πίνακα δρομολόγησης για την προώθηση των πακέτων στους τελικούς προορισμούς.

Για να πάρουν την τοπολογία του δικτύου, οι αλγόριθμοι χρησιμοποιούν τα εξής:

- **Topology Sensing:** Περιλαμβάνει την ανίχνευση γειτονικών κόμβων ώστε να ανακτήσουν πληροφορίες σχετικά με το δίκτυο και την αποστολή και λήψη περιοδικών μηνυμάτων αναγνώρισης (Hello messages).
- **Topology Discovery:** Χρησιμοποιεί τις πληροφορίες που ανακτήθηκαν από τους γειτονικούς κόμβους ώστε να δημιουργηθεί ο πίνακας δρομολόγησης.

Όλες οι πιθανές διαδρομές για έναν τελικό κόμβο υπολογίζονται και αποθηκεύονται στον πίνακα δρομολόγησης. Ο αλγόριθμος MP-OLSR χρησιμοποιεί μια on-demand τεχνική ώστε να αποφευχθεί ο χρονοβόρος υπολογισμός κάθε πιθανής διαδρομής. Αν και αυτή η τεχνική μπορεί να μην επιβαρύνει τα network updates, χρησιμοποιεί πολύπλοκες τεχνικές που επιβαρύνουν την συνολική απόδοση του δικτύου.

3.5.1.3 MMESH

Το πρωτόκολλο αυτό δημιουργήθηκε για να κάνει load balancing χρησιμοποιώντας πολλαπλές διαδρομές. Με τον όρο load balancing εννοούμε την ίση κατανομή της κίνησης σε δύο ή περισσότερες συνδέσεις μεταξύ διεύθυνσης πηγής και διεύθυνσης προορισμού.

- **Network Setup:** Στην αρχική φάση, οι κόμβοι βρίσκουν διαδρομές προς το gateway ακούγοντας τα broadcast και advertised μηνύματα μιας σύνδεσης (π.χ. Internet). Χρησιμοποιώντας τα metrics που υπάρχουν σε αυτά τα μηνύματα ο κάθε router που χρησιμοποιεί αυτόν τον αλγόριθμο εισάγει ένα next-hop στον πίνακα δρομολόγησης. Μετά την εισαγωγή μιας διαδρομής οι κόμβοι στέλνουν update μηνύματα στους γειτονικούς κόμβους ώστε να ενημερωθούν για τα routes που υπάρχουν στον κόμβο που έστειλε τα updates. Για να αποφευχθεί το επιπλέον overhead όταν ολοκληρωθεί η δημιουργία του πίνακα δρομολόγησης εκχωρείται ένας σειριακός αριθμός ο οποίος

δηλώνει την ηλικία του πίνακα δρομολόγησης (όσο πιο μεγάλος ο σειριακός αριθμός τόσο πιο νέος είναι ο πίνακας δρομολόγησης).

- **Network Maintenance:** Κάθε κόμβος ελέγχει την απόδοση των διαδρομών μέχρι να φτάσουν στο gateway και προωθεί τις πληροφορίες που συγκέντρωσε στους γειτονικούς κόμβους. Αναλύοντας αυτές τις αλλαγές μπορεί να προκύψουν νέες διαδρομές ή να σταματήσει την προώθηση πακέτων σε κάποιες διαδρομές επανεκκινώντας την διαδικασία Network Setup. Επομένως σε περίπτωση κάποιου χαλασμένου κόμβου το δίκτυο θα είναι ενήμερο (λόγω των updates και των suspended routes) και ο αλγόριθμος θα κρατάει την κατάσταση του δικτύου πιο σταθερή.

3.5.2 Reactive Multiroute Algorithms

3.5.2.1 AODV-DM

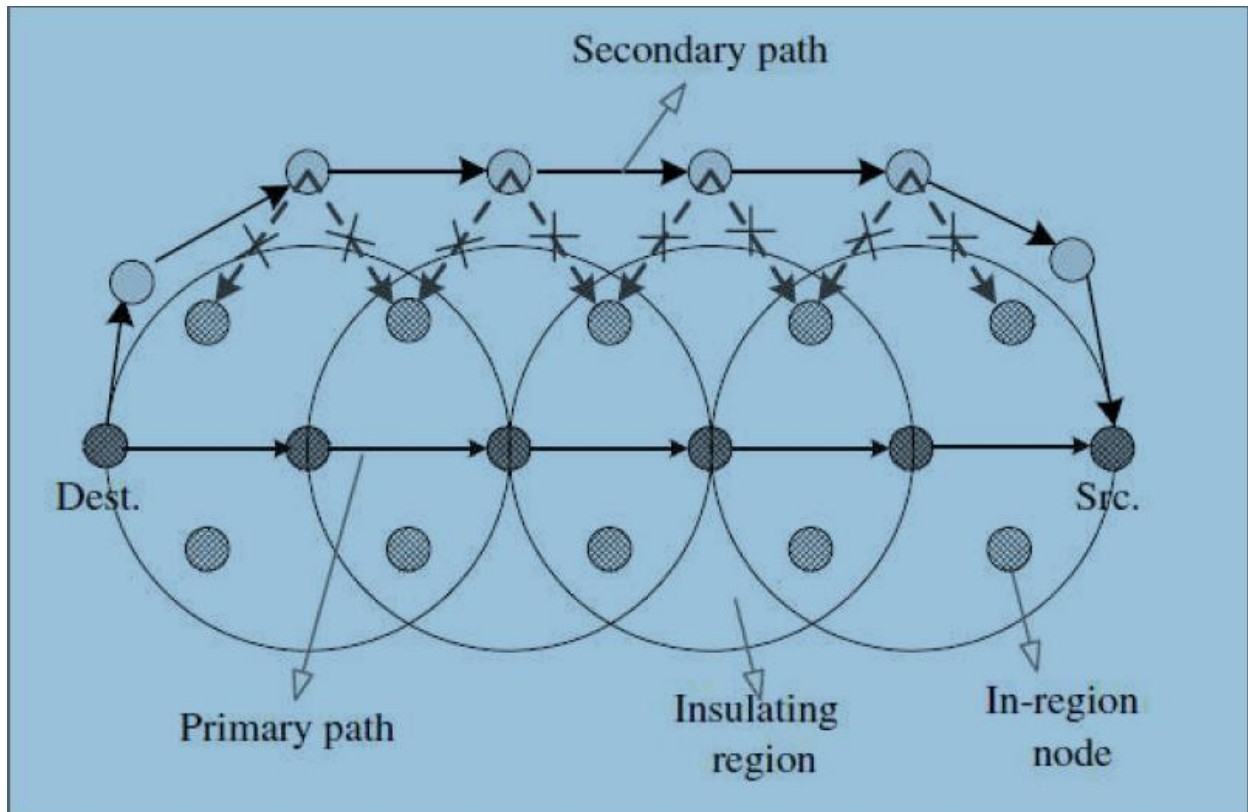
AODV-based Decoupled Multipath προτείνει έναν αλγόριθμο που καθορίζει μια συγκεκριμένη περιοχή (insulate area) γύρω από την πρωτεύων διαδρομή (primary path) για να αποφευχθούν οι παρεμβολές μεταξύ του primary και secondary route.

- **Route Establishment:** Ο primary route δημιουργείται ως εξής: Ένα RREQ μήνυμα διαχέεται σε ολόκληρο το backhaul δίκτυο.

Κάθε κόμβος που λαμβάνει το μήνυμα αυτό αποθηκεύει την πληροφορία στον πίνακα δρομολόγησης του. Είναι πιθανό ένας κόμβος να λάβει RREQ μηνύματα. Σε αυτή την περίπτωση ο κόμβος απαντάει στο μήνυμα που ακολούθησε την μικρότερη διαδρομή (η οποία καλείται ως primary route) στέλνοντας ένα μήνυμα απάντησης primary RREP (pRREP).

Το πακέτο ακολουθεί την κοντύτερη πορεία, οι ενδιαμέσοι κόμβοι κατά μήκος της διαδρομής μεταδίδουν μέσω broadcast μηνυμάτων το πακέτο στους κόμβους γειτόνων, οι οποίοι χαρακτηρίζονται κόμβοι περιοχής (in-region).

Μετά από μια χρονική περίοδο για να επιτρέψει στη περιοχή να καθιερωθεί, ο προορισμός αποκρίνεται σε άλλα Rreqs. Το πακέτο απάντησης καλείται δεύτερη απάντηση διαδρομών (sRrep) και η διαδικασία διάδοσης αυτού του μηνύματος είναι σχεδόν η ίδια με το pRrep. Η εξαίρεση είναι ότι όταν ένας κόμβος που λαμβάνει μια μετάδοση sRrep δεν χαρακτηρίζεται ως in-region κόμβος. Οι ενδιαμέσοι κόμβοι θα μεταδώσουν το πακέτο στους γείτονές τους προκειμένου να αφαιρεθεί από τους πίνακες δρομολόγησης των γειτόνων τους όπως φαίνεται και στο παρακάτω σχήμα.



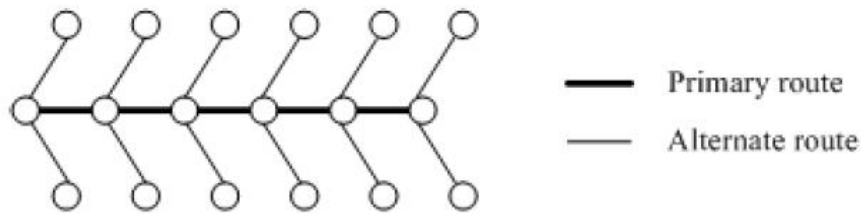
Σχήμα 9: Εικονική απεικόνιση AODV-DM

- Route Maintenance: Επειδή ο αλγόριθμος έχει ως βάση τον AODV η διαδικασία αυτή είναι ακριβώς ίδια με τον AODV.

3.5.2.2 AODV-BR

AODV- Backup Routing προτείνει έναν εφεδρικό μηχανισμό διαδρομών που βελτιώνεται η απόδοση των υπάρχοντων on-demand routes μέσω της διαδικασίας request/reply.

- Route Construction: Η κατασκευή διαδρομών γίνεται σχεδόν με τον ίδιο τρόπο όπως ο AODV, αλλά οι πολλαπλές διαδρομές διαμορφώνονται με τον ακόλουθο τρόπο. Όταν ένας κόμβος λαμβάνει ένα Rreq που δεν κατευθύνεται σε αυτόν από έναν γείτονα, προσθέτει το γείτονα ως next-hop στον πίνακα δρομολόγησής του (εάν λαμβάνει περισσότερους του ενός που επιλέγει το καλύτερο).



Σχήμα 10: Εικονική απεικόνιση AODV-BR

- **Route Maintenance:** Ενεργοποιείται όταν ένας κόμβος αντιληφθεί κάποιο πρόβλημα στην σύνδεση. Σε αυτή την περίπτωση ο κόμβος εκπέμπει ένα μήνυμα λάθους στον επόμενο γειτονικό κόμβο (one hop away) ώστε να δρομολογηθεί το πακέτο σε εναλλακτική διαδρομή. Το μήνυμα φτάνει στον αρχικό κόμβο όπου και αντιλαμβάνεται την προβληματική σύνδεση και ξεκινάει την διαδικασία εύρεσης διαδρομής.
- **Traffic Distribution:** Ένα μεγάλο μειονέκτημα του αλγορίθμου είναι ότι χρησιμοποιείται μια διαδρομή τη φορά (primary route). Όταν παρουσιαστεί σφάλμα στην σύνδεση χρησιμοποιείται το backup route για να δρομολογηθεί η κίνηση

Οι μελέτες έχουν δείξει ότι ο αλγόριθμος λειτουργεί καλά όταν δεν υπάρχει μεγάλη κίνηση στο δίκτυο αλλά δεν είναι αποδοτικός όταν η κίνηση είναι υψηλή λόγω των προβλημάτων collision που μπορεί να δημιουργηθούν στα ασύρματα δίκτυα.

3.5.2.3 AOMDV

Ο Ad-hoc On Demand Multipath Distance Vector είναι ένας αλγόριθμος που βασίζεται στον AODV και έχει ως κύριο σκοπό την μείωση του overhead όταν σε μια διαδρομή παρουσιαστεί σφάλμα.

- **Route Establishment:** Όπως και στον AODV η διαδικασία εύρεσης διαδρομής ενεργοποιείται όταν ένας κόμβος θέλει να επικοινωνήσει με έναν άλλον και δεν υπάρχει διαθέσιμη διαδρομή προς αυτόν. Η διαδικασία που ακολουθείται είναι η ίδια αλλά για την δημιουργία πολλαπλών διαδρομών εξετάζονται όλα τα RREQ μηνύματα που φτάνουν σε έναν ενδιάμεσο κόμβο (duplicate RREQs). Για την δημιουργία της διαδρομής είναι απαραίτητο και το μήνυμα απάντησης RREP. Ο αλγόριθμος για να εξασφαλίσει ότι χρησιμοποιούνται διαφορετικά links σε κάθε διαδρομή στέλνει μήνυμα απάντησης μόνο στους γείτονες που έλαβε μοναδικό (unique) RREQ.
- **Route Maintenance:** Ακολουθείται η ίδια διαδικασία που ακολουθεί ο AODV αλγόριθμος και στην περίπτωση κάποιας προβληματικής σύνδεσης στέλνεται μήνυμα λάθους (RRER).

- Traffic Distribution: Με την διαθεσιμότητα πολλών διαφορετικών διαδρομών η διαδικασία route discovery χρησιμοποιείται μία φορά . Η διαδικασία θα επαναληφθεί μόνο όταν όλες οι διαδρομές παρουσιάσουν κάποιο πρόβλημα.

3.5.3 ΖΗΤΗΜΑΤΑ ΣΧΕΔΙΑΣΗΣ MULTI ROUTE ΑΛΓΟΡΙΘΜΩΝ

Αν και οι παραπάνω αλγόριθμοι που αναλύθηκαν χρησιμοποιούν διαφορετικές διαδρομές για την δρομολόγηση μειονεκτούν σε ένα σημαντικό ζήτημα: πόσο μπορεί να μειωθεί η απόδοση του δικτύου αν δύο ή περισσότερες διαδρομές διασταυρωθούν και αρχίσουν να χρησιμοποιούν τους ίδιους πόρους. Σε αυτό το ζήτημα απαντάει και η συγκεκριμένη εργασία η οποία συνδυάζει στοιχεία από τους παραπάνω αλγόριθμους και εισάγει ένα ιεραρχικό μοντέλο ανάπτυξης του δικτύου χρησιμοποιώντας τεχνικές clustering. Οι παρακάτω ενότητες αναλύουν τις τεχνικές αυτές ώστε να γίνει πιο κατανοητή η όλη αρχιτεκτονική της λύσης που θα παρουσιαστεί στην τελευταία ενότητα.

ΚΕΦΑΛΑΙΟ 40 : CLUSTERING ΑΛΓΟΡΙΘΜΟΙ

Αυτού του είδους οι αλγόριθμοι χρησιμοποιούνται για να βελτιώσουν την απόδοση του δικτύου βελτιώνοντας σημαντικά κάποιες παραμέτρους όπως delay, throughput καθώς επίσης και τις παρεμβολές. Επίσης πολύ σημαντική παράμετρος είναι ότι μειώνουν σημαντικά την **πολυπλοκότητα** του δικτύου. Ένας clustering αλγόριθμος χωρίζει το δίκτυο σε μικρότερες ομάδες με ίδια χαρακτηριστικά σε κάθε ομάδα. Το κάθε cluster είναι ανεξάρτητο από τα υπόλοιπα και ο μόνος τρόπος επικοινωνίας με τα υπόλοιπα clusters είναι μέσω του cluster gateway. Επομένως με την χρήση των cluster κάποιο σφάλμα σε κάποια σύνδεση ή σε ένα κόμβο δεν επηρεάζει ολόκληρη την τοπολογία του δικτύου αλλά μόνο το cluster όπου παρουσιάστηκε η βλάβη.

Για την σωστή χρήση ενός clustering αλγορίθμου είναι απαραίτητος ο υπολογισμός των παρακάτω παραμέτρων στην τοπολογία του δικτύου:

- Κάθε κόμβος μέλος (cluster member) μπορεί να επικοινωνεί απευθείας με το gateway node.
- Κάθε cluster πρέπει να μην επικαλύπτει κάποιο άλλο. Πιο συγκεκριμένα ο σχεδιασμός θα πρέπει να γίνει με τέτοιο τρόπο ώστε να μην υπάρχει κόμβος που να ακούει περισσότερα από ένα gateway nodes.
- Θα πρέπει να χρησιμοποιηθεί κάποιος proactive αλγόριθμος για την επικοινωνία των κόμβων μέσα σε ένα cluster και ένας reactive αλγορίθμος για την επικοινωνία των gateway κόμβων.

Στις επόμενες υποενότητες αναλύονται συνοπτικά οι πιο βασικές τεχνικές clustering αλγορίθμων.

4.1 LOWEST ID

Ο αλγόριθμος χωρίζει του συνολικούς κόμβους του δικτύου σε μη επικαλυπτόμενες ομάδες (clusters). Η λειτουργία του αλγορίθμου είναι η ακόλουθη:

- Σε κάθε κόμβο ανατίθεται από ένα μοναδικό ID. Ανά τακτά χρονικά διαστήματα κάθε κόμβος στέλνει broadcast προς τους γειτονικούς κόμβους σχετικά με τα IDs που έχει λάβει από άλλους κόμβους συμπεριλαμβανομένου και το δικό του. Ένας κόμβος που ακούει IDs μεγαλύτερα από το δικό του τότε έχει οριστεί ως cluster head. Η διαδικασία είναι η παρακάτω:
- Θεωρούμε ότι ένας κόμβος μαζί με τους γειτονικούς one-hop-away κόμβους σχηματίζουν μια ομάδα (NodeSet). Αν ο κόμβος έχει το χαμηλότερο ID γίνεται cluster head και βγαίνει από την διαδικασία αποστολής ID. Έπειτα ο νεοσχηματισμένος cluster head εκπέμπει ένα cluster μήνυμα έχοντας ως id το δικό του μοναδικό αριθμό.
- Μόλις οι κόμβοι που πάρουν το μήνυμα διαπιστώσουν ότι το ID είναι μικρότερο από το δικό τους τότε σημαίνει ότι ανήκουν σε κάποιο cluster.
- Αν κάποιος κόμβος μετά από αυτή την διαδικασία δεν έχει μπει σε κάποιο cluster τότε γίνεται ο ίδιος cluster head και στέλνει το cluster μήνυμα με το δικό του ID. Σχηματίζεται δηλαδή cluster το οποίο περιέχει έναν μόνο κόμβο.

4.2 ACE

Ο αλγόριθμος χωρίζεται σε δύο μέρη: πρώτα χωρίζει τους κόμβους σε ομάδες και τέλος προσαρμόζει τα clusters με τέτοιο τρόπο ώστε να αποφεύγεται όσο το δυνατόν περισσότερο το overallring. Πρόκειται για ένα ασύγχρονο πρωτόκολλο γιατί οι κόμβοι μπορεί να το ενεργοποιήσουν σε τυχαία χρονικά διαστήματα.

Κατά τη διάρκεια εκτέλεσης του αλγορίθμου ένας κόμβος μπορεί να είναι σε τρία στάδια:

1. Unclustered: Δεν έχει εισαχθεί ακόμα σε κάποιο cluster
2. Cluster-head: Η διαδικασία του σχηματισμού έχει ολοκληρωθεί και υπάρχει ένα cluster head σε κάθε cluster.
3. Clustered: Ο κόμβος ανήκει σε ένα cluster.

4.3 ΣΤΟΧΟΙ MULTI-PATH ROUTING

Έχοντας υπόψη τους αλγορίθμους που αναλύθηκαν παραπάνω, θεωρούμε ότι ένα μοντέλο δρομολόγησης πολλαπλών διαδρομών πρέπει να έχει τα ακόλουθα χαρακτηριστικά:

- **Multiple routes:** Πρέπει να χρησιμοποιούνται για να προσφέρουν μια ισορροπία μεταξύ των διαδρομών που χρησιμοποιούνται και τα αρνητικά χαρακτηριστικά των αλγορίθμων όπως οι παρεμβολές και το overhead.
- **Χαμηλό overhead:** Στόχος της τεχνικής που αναπτύξαμε είναι χρήση multi-route αλγόριθμου με το χαμηλότερο δυνατό overhead.
- **Υψηλή επίδοση:** Έχοντας περισσότερες από μία διαδρομές για αποστολή πακέτων το throughput του δικτύου παραμένει σε ψηλά επίπεδα γιατί γίνεται καλύτερη χρήση του bandwidth κάθε σύνδεσης του δικτύου.

Προηγούμενες έρευνες πάνω στο ζήτημα της δρομολόγησης πολλαπλών διαδρομών έχουν δείξει ότι η χρήση ανεξάρτητων διαδρομών αυξάνει σημαντικά το throughput από ότι η χρήση αλγορίθμων με διαδρομές που παρεμβάλλονται (interfering paths). Χρησιμοποιώντας multi-route τεχνικές έχει αποδειχθεί ότι η επίδοση του δικτύου μπορεί να αυξηθεί μέχρι και 50% σε σχέση με τις single-route τεχνικές.

Θεωρώντας αυτές τις έρευνες ως βάση, εισάγουμε στην εργασία μια τεχνική η οποία θα χρησιμοποιεί multi-route αλγόριθμους για την δρομολόγηση αλλά σε επίπεδο clusters. Στα παρακάτω κεφάλαια θα δείξουμε ότι με την multi-cluster routing μαζί με την αποφυγή παρεμβαλλόμενων διαδρομών (route join) η απόδοση του δικτύου αυξάνεται. Επίσης στο κεφάλαιο 5 παρουσιάζονται συνοπτικά τα χαρακτηριστικά του εργαλείου προσομοίωσης που χρησιμοποιήθηκε και τέλος παρουσιάζεται εκτενώς η τεχνική που προτείνουμε στην διπλωματική εργασία.

ΚΕΦΑΛΑΙΟ 5^ο: ΕΙΣΑΓΩΓΗ ΣΤΟ ΜΟΝΤΕΛΟ ΠΡΟΣΟΜΟΙΩΣΗΣ ΚΑΙ ΑΝΑΛΥΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

5.1 Εισαγωγή στο OPNET Modeller

Το OPNET Modeller, αποτελεί ένα εξειδικευμένο εργαλείο στο χώρο των επικοινωνιών, που προσφέρει τη δυνατότητα με τη βοήθεια ενός γραφικού περιβάλλοντος να μοντελοποιηθούν και να προσομοιωθούν διάφορα είδη δικτύων. Στην παρούσα εργασία χρησιμοποιείται το OPNET Modeller 10.5, το οποίο περιέχει έτοιμα μοντέλα δικτύων από το Μάρτη του 2004. Η γλώσσα προγραμματισμού που χρησιμοποιείται (και για την κατασκευή νέων μοντέλων) είναι η Proto C, μια παραλλαγή της γλώσσας C, η οποία μπορεί να χρησιμοποιεί και έτοιμες συναρτήσεις από τον πυρήνα (Kernel) του OPNET.

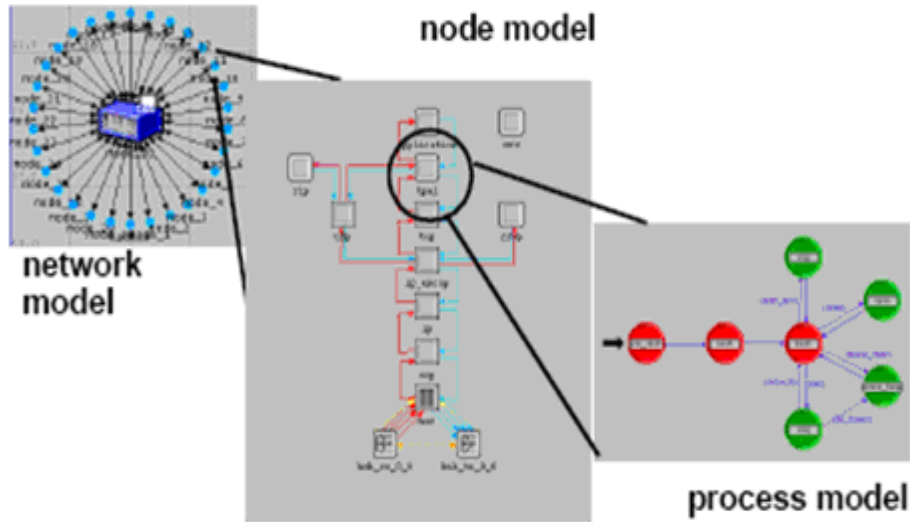
Για την κατασκευή της δομής ενός δικτύου, ακολουθείται αυστηρά η παρακάτω ιεραρχία τριών επιπέδων:

Μοντέλο Δικτύου (Project Editor)

Μοντέλο Κόμβων (Node Editor)

Μοντέλο Επεξεργασίας (Process Editor)

Τα αντικείμενα που ανήκουν στο μοντέλο δικτύου περιγράφονται από αντικείμενα που ανήκουν στο μοντέλο κόμβων, ενώ αυτά που ανήκουν στο μοντέλο κόμβων περιγράφονται από αντικείμενα που ανήκουν στο μοντέλο επεξεργασίας. Ένα παράδειγμα αυτής της ιεραρχίας φαίνεται παρακάτω:

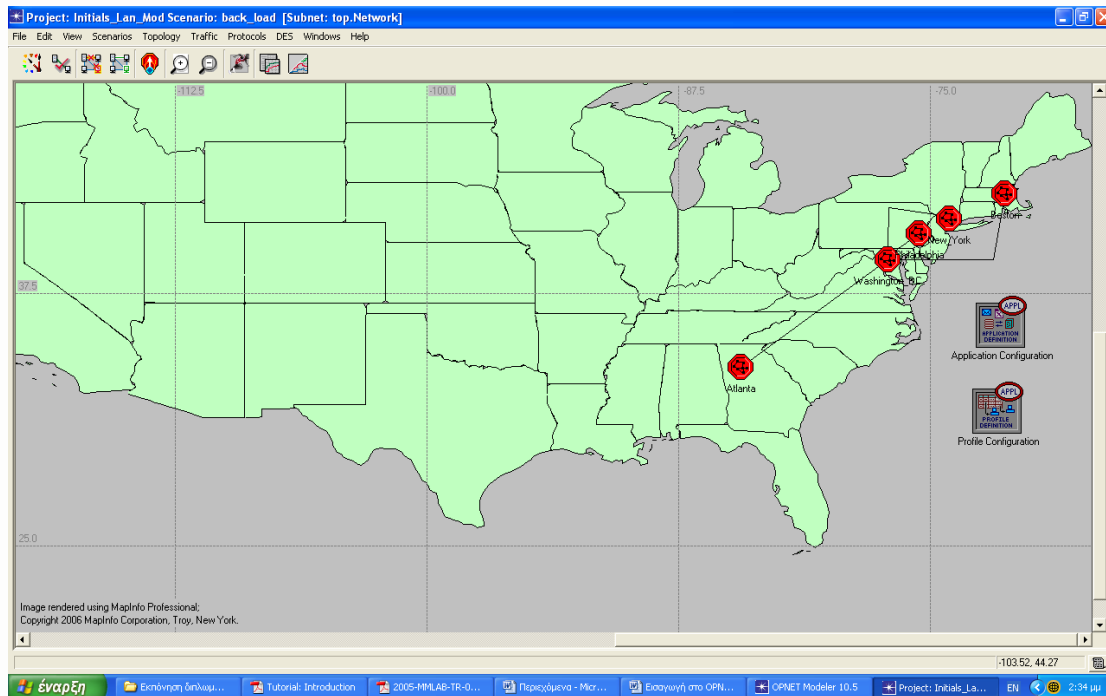


Εικόνα 2: Ιεραρχική δομή μοντελοποίησης στο OPNET

5.1.1 Μοντέλο Δικτύου

Το Μοντέλο Δικτύου (Project Editor), αποτελεί την κύρια πλατφόρμα εργασίας για την κατασκευή και προσομοίωση ενός δικτύου. Από εδώ μπορούμε να κτίσουμε ένα μοντέλο δικτύου, χρησιμοποιώντας τα έτοιμα μοντέλα (π.χ. ένα Τοπικό Δίκτυο (LAN), ένα Μητροπολιτικό Δίκτυο (MAN) ή ένα δίκτυο με δύο υπολογιστές κ.τ.λ.) που υπάρχουν στη βιβλιοθήκη του OPNET, να επιλέξουμε στατιστικά στοιχεία για το δίκτυο, να τρέξουμε μια προσομοίωση και να δούμε τα αποτελέσματα. Επίσης μπορούμε να φτιάξουμε τα μοντέλα δικτύου και επεξεργασίας, να κατασκευάσουμε μοντέλα για πακέτα που στέλνονται, να κατασκευάσουμε φίλτρα και παραμέτρους στα οποία μπορούμε να έχουμε πρόσβαση από το μοντέλο επεξεργασίας.^[50]

Πιο κάτω φαίνεται το προφίλ ενός μοντέλου δικτύου, όπου έχουν δημιουργηθεί 5 Τοπικά Δίκτυα (κόκκινα σημεία), που επικοινωνούν μεταξύ τους:



Εικόνα 3: Μοντέλο Δικτύου

Στη μπάρα εργαλείων (toolbar), που βρίσκεται στο πάνω μέρος του παραθύρου, υπάρχουν οι πιο κάτω επιλογές, τις οποίες επεξηγούμε στη συνέχεια:



Εικόνα 4: Μπάρα εργαλείων του Μοντέλου Δικτύου

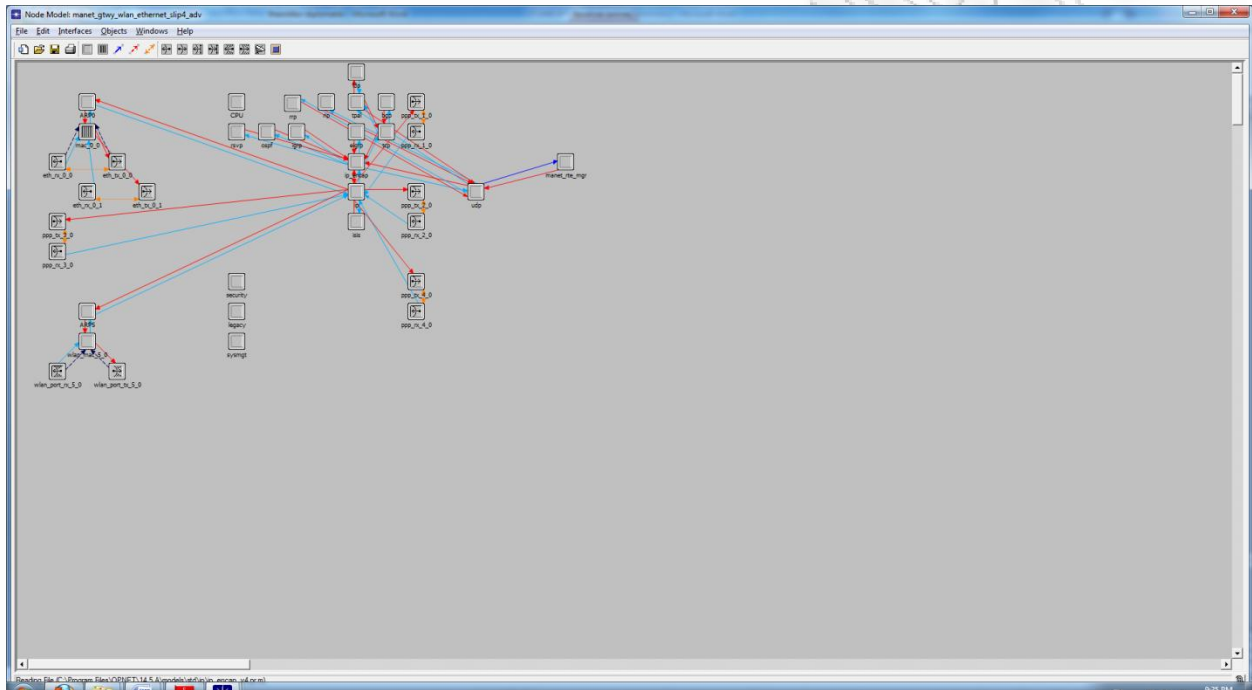
1. Παλέτα αντικειμένων (object palette): Ανοίγοντας την παλέτα αντικειμένων, μπορούμε να ανασύρουμε κάποιο έτοιμο μοντέλο αντικειμένου του OPNET, π.χ. ένα server, ένα δίκτυο υπολογιστών, μια σύνδεση καλωδίου μεταξύ δύο υποδικτύων. Μπορούμε επίσης με την επιλογή “διαμόρφωση παλέτας” (configure palette), να δημιουργήσουμε μια παλέτα που θα περιέχει μόνο τα αντικείμενα που θα χρειαστούμε ή ακόμα και να τροποποιήσουμε κάποια από αυτά έτσι ώστε να έχουν τα επιθυμητά χαρακτηριστικά.

2. Επαλήθευση συνδέσεων (verify links): Με αυτή την επιλογή, γίνεται έλεγχος αν οι συνδέσεις που υπάρχουν ανάμεσα στα διάφορα αντικείμενα λειτουργούν σωστά. Αν κάποια σύνδεση δε λειτουργεί, εμφανίζεται σχετικό μήνυμα για να μπορέσουμε να τη διορθώσουμε.

3. Κατάργηση αντικειμένων (fail selected objects): Επιλέγοντας κάποια αντικείμενα που υπάρχουν στην επιφάνεια εργασίας και με την επιλογή αυτή, μπορούμε να καταργήσουμε τη λειτουργία τους, δηλαδή η προσομοίωση θα “τρέξει” σαν αυτά να μην υπάρχουν καθόλου.
4. Ανάκτηση αντικειμένων (recover selected objects): Επαναφέρει αντικείμενα που είχαν καταργηθεί προηγουμένως.
5. Επαναφορά στο υποδίκτυο του προηγούμενου επιπέδου (go to parent subnet): Αν θελήσουμε να δούμε τα επιμέρους αντικείμενα ενός υποδικτύου, μπορούμε με διπλή επιλογή του (double click), να εισέλθουμε στο ακριβώς πιο κάτω επίπεδο. Για παράδειγμα για ένα από τα τοπικά δίκτυα του σχήματος 3.1.2, με διπλή επιλογή βλέπουμε τα αντικείμενα που το αποτελούν (router και office_lan). Στη συνέχεια με την επιλογή “επαναφορά στο υποδίκτυο του προηγούμενου επιπέδου”, επανερχόμαστε στο προηγούμενο επίπεδο. Για να μπούμε πιο βαθιά στην ιεραρχική δομή, με διπλή επιλογή κάποιου από τα αντικείμενα του δευτέρου επιπέδου στο μοντέλο δικτύου (π.χ. router), μπαίνουμε στο μοντέλο κόμβων και από εκεί στο μοντέλο επεξεργασίας. Αυτά όμως λειτουργούν σε ξεχωριστά παράθυρα και η επιλογή “επαναφορά στο υποδίκτυο του προηγούμενου επιπέδου”, δεν ισχύει.
6. Εστίαση σε ορθογώνιο (zoom to rectangle): Με την επιλογή αυτή, η επιφάνεια εργασίας, εστιάζει σε ένα ορθογώνιο κομμάτι που επιλέγουμε.
7. Εστίαση στο προηγούμενο (zoom to previous): Η εστίαση επαναφέρεται.
8. Διαμόρφωση/τρέξιμο προσομοίωσης ξεχωριστών γεγονότων (configure/run discrete event simulation, DES): Με την επιλογή αυτή, ανοίγουμε ένα παράθυρο, όπου επιλέγουμε τις παραμέτρους της προσομοίωσης, όπως είναι η διάρκεια, τα στατιστικά στοιχεία που θα συλλέξουμε και ο σπόρος (seed) που αφορά την τυχαιότητα κάποιων συναρτήσεων. Από εδώ μπορούμε επίσης να ορίσουμε τα είδη των εξερχόμενων στατιστικών (π.χ. report, animation) και άλλα χαρακτηριστικά της προσομοίωσης. Με την επιλογή “τρέξιμο” (run), η προσομοίωση αρχίζει.
9. Παρατήρηση αποτελεσμάτων (view results): Εδώ μπορούμε να ορίσουμε τα χαρακτηριστικά που μας ενδιαφέρει να δούμε με το πέρας της προσομοίωσης, καθώς και σε ποιες συνδέσεις (π.χ. στο σχήμα 3.1.2 μπορεί να μας ενδιαφέρουν τα χαρακτηριστικά μόνο μιας εκ των τεσσάρων συνδέσεων μεταξύ των τοπικών δικτύων). Έτσι βλέπουμε γραφικές παραστάσεις που αφορούν τα επιλεχθέντα χαρακτηριστικά. Επειδή μια προσομοίωση μπορεί να τρέξει παράλληλα για διαφορετικά σενάρια, από εδώ επιλέγουμε για ποια από αυτά θέλουμε να δούμε τα αποτελέσματα καθώς μπορεί να γίνει και σύγκρισή τους στην ίδια γραφική παράσταση.
10. Απόκρυψη/παρατήρηση γραφικών παραστάσεων (hide/show graph panels): Με το πέρας της προσομοίωσης, αν ανοίξουμε μερικά παράθυρα γραφικών παραστάσεων για να παρατηρήσουμε κάποια χαρακτηριστικά, με την επιλογή αυτή μπορούμε να τα αποκρύψουμε και στη συνέχεια να τα επαναφέρουμε.

5.1.2 Μοντέλο Κόμβων

Τα αντικείμενα του Μοντέλου Κόμβων (Node Editor), περιγράφουν την συμπεριφορά των αντίστοιχων κόμβων δικτύου και μοντελοποιούν τις εσωτερικές τους λειτουργίες όπως δημιουργία δεδομένων, αποθήκευση κ.τ.λ. Η συμπεριφορά κάθε αντικειμένου του δικτύου περιγράφεται από διάφορα “στοιχεία” (modules), τα οποία συνδέονται μεταξύ τους με “συνδέσεις πακέτων” (packet streams) ή “στατιστικά καλώδια” (statistic wires). Όλα αυτά φαίνονται στο σχήμα που ακολουθεί:^[50]



Εικόνα 5: Μοντέλο Κόμβων

Στη μπάρα εργαλείων (toolbar), που βρίσκεται στο πάνω μέρος του παραθύρου, όπως και στο μοντέλο δικτύου, υπάρχουν οι πιο κάτω επιλογές, τις οποίες επεξηγούμε στη συνέχεια:



Εικόνα 6: Μπάρα εργαλείων του Μοντέλου Κόμβων

1. Δημιουργία επεξεργαστή (create processor): Με αυτή την επιλογή μπορούμε να δημιουργήσουμε ένα επεξεργαστή στην επιφάνεια εργασίας. Ο επεξεργαστής αποτελεί ένα “στοιχείο”, που μπορεί να

επιτελέσει διάφορους σκοπούς όπως για παράδειγμα δημιουργία και καταστροφή πακέτων, επεξεργασία πακέτων κ.λ.π..

2. Δημιουργία “στοιχείου” ουράς (create queue): Δημιουργούμε ένα “στοιχείο” ουράς, όπου μπορεί να αποθηκευτεί ένας αριθμός πακέτων και να προγραμματιστεί η μετάδοσή τους. Επίσης μπορούν να οριστούν τύποι πακέτων που το στοιχείο αυτό μπορεί να ανακτήσει, ενώ τα υπόλοιπα πακέτα απορρίπτονται.

3. Δημιουργία σύνδεσης πακέτων (create packet stream): Με την επιλογή αυτή δημιουργούμε μια σύνδεση πακέτου μεταξύ δύο “στοιχείων”. Η σύνδεση πακέτου αποτελεί τη διαδρομή που ακολουθεί το κάθε πακέτο από “στοιχείο” σε “στοιχείο”, μέσα σε ένα αντικείμενο δικτύου, για επεξεργασία.

4. Δημιουργία στατιστικού καλωδίου (create statistic wire): Ένα στατιστικό καλώδιο ακολουθεί την ίδια διαδρομή με μια σύνδεση πακέτων αλλά κατά την αντίθετη κατεύθυνση. Ουσιαστικά, υλοποιεί την αμφίδρομη επικοινωνία που υπάρχει σε ένα αντικείμενο δικτύου καθώς “τρέχει” τη στοίβα πρωτοκόλλων από πάνω προς τα κάτω και αντίστροφα.

5. Δημιουργία λογικού συνδέσμου πομπού/δέκτη (create logical Tx/Rx association): Με αυτή την επιλογή, δημιουργούμε μια λογική σύνδεση μεταξύ δύο στοιχείων πομπού (transmitter, Tx) και δέκτη (receiver, Rx). Κατά τη διάρκεια της προσομοίωσης, πακέτα τα οποία στέλνονται σε κάποιο αντικείμενο δικτύου, παραλαμβάνονται από το “στοιχείο” του δέκτη. Στη συνέχεια στέλνονται προς τα πάνω, στα υπόλοιπα στοιχεία για επεξεργασία (στοίβα πρωτοκόλλων) και ακολούθως, στην αντίθετη διαδρομή προς τα κάτω καταλήγοντας στο “στοιχείο” του πομπού.^[50] Αυτή η λογική σύνδεση μεταξύ πομπού και δέκτη, αναπαρίσταται με ένα λογικό σύνδεσμο Tx/Rx.

6. Δημιουργία δέκτη σημείου σε σημείο (create point-to-point receiver): Δημιουργούμε ένα “στοιχείο” δέκτη, για συνδέσεις σημείο σε σημείο.

7. Δημιουργία πομπού σημείου σε σημείο (create point-to-point transmitter): Δημιουργούμε ένα “στοιχείο” πομπού, για συνδέσεις σημείο σε σημείο.

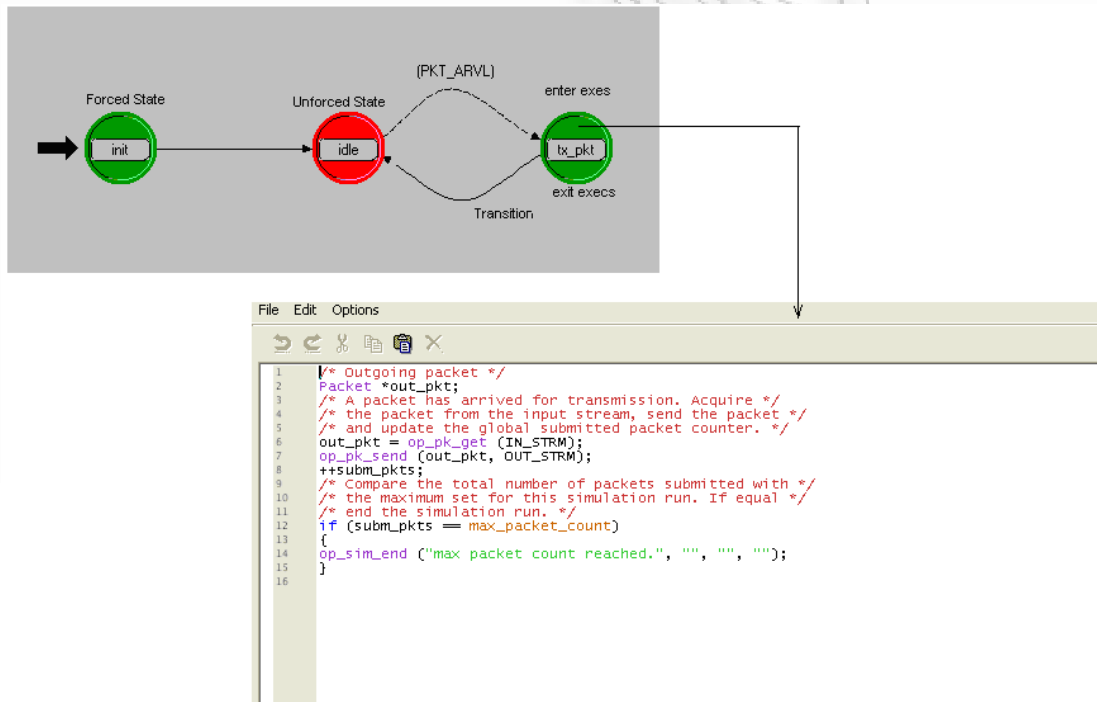
8. Δημιουργία δέκτη για πολλαπλές συνδέσεις (create bus receiver): Δημιουργούμε ένα “στοιχείο” δέκτη, για πολλαπλές συνδέσεις.

9. Δημιουργία πομπού για πολλαπλές συνδέσεις (create bus transmitter): Δημιουργούμε ένα “στοιχείο” πομπού, για πολλαπλές συνδέσεις.

10. Δημιουργία “στοιχείου” εξωτερικού συστήματος (create external system module): Με την επιλογή αυτή, μπορούμε να δημιουργήσουμε ένα “στοιχείο”, το οποίο δεν ανήκει κατ’ ανάγκη στο συγκεκριμένο αντικείμενο δικτύου και μπορεί να λειτουργεί ξεχωριστά.

5.1.3 Μοντέλο Επεξεργασίας

Το Μοντέλο Επεξεργασίας (Process Editor), αναπαρίσταται με διαγράμματα πεπερασμένων καταστάσεων (finite state machines - FSMs) που ελέγχουν την εσωτερική λειτουργικότητα των αντικειμένων στο μοντέλο κόμβων. Τα διαγράμματα αυτά δημιουργούνται από εικόνες που αντιπροσωπεύουν καταστάσεις (states) και γραμμές (transitions) που αντιπροσωπεύουν μεταβάσεις μεταξύ αυτών των καταστάσεων. Κάθε κατάσταση χαρακτηρίζεται από τις λειτουργίες εισόδου (enter execution) – οι οποίες περιγράφονται σε κώδικα της γλώσσας C και γράφονται στο πάνω μισό κομμάτι της εικόνας μιας κατάστασης- και τις λειτουργίες εξόδου (exit execution) – οι οποίες γράφονται στο κάτω μισό κομμάτι. “Αναδύμενα” παράθυρα εμφανίζονται με διπλή επιλογή για κάθε κομμάτι, όπου μπορούμε να γράψουμε τον κώδικα. Οι καταστάσεις επίσης χαρακτηρίζονται σε “εξαναγκασμένες” (forced, πράσινο χρώμα) και “μη-εξαναγκασμένες” (unforced, κόκκινο χρώμα), ανάλογα με το αν θέλουμε μια κατάσταση να μεταβαίνει στην επόμενη χωρίς κανένα περιορισμό ή όχι αντίστοιχα. Συνθήκες μετάβασης μπορούν επίσης να οριστούν στις γραμμές μετάβασης για να ορίσουν περιορισμούς για τη μετάβαση από την μια κατάσταση στην άλλη. Αυτά φαίνονται στο σχήμα που ακολουθεί.^[50]



Εικόνα 7: Μοντέλο Επεξεργασίας

Όπως και στα προηγούμενα μοντέλα, στη μπάρα εργαλείων (toolbar), που βρίσκεται στο πάνω μέρος του παραθύρου, υπάρχουν οι πιο κάτω επιλογές, τις οποίες επεξηγούμε στη συνέχεια:



Εικόνα 8: Μπάρα εργαλείων του Μοντέλου Επεξεργασίας

1. Δημιουργία κατάστασης (create state): Με την επιλογή αυτή, μπορούμε να δημιουργήσουμε μια κατάσταση. Η κατάσταση αυτή, αποτελεί ένα κομμάτι της όλης διαδικασίας που εκτελείται σε ένα “στοιχείο” του Μοντέλου Κόμβων. Με επιλογή του δεξιού κουμπιού του “mouse” του υπολογιστή, εμφανίζεται η επιλογή για να μετατρέψουμε μια κατάσταση από “εξαναγκασμένη” σε “μη-εξαναγκασμένη” και το αντίθετο. Στο πάνω μισό κομμάτι της κατάστασης, όπως αναφέραμε και προηγουμένως, γράφονται οι συναρτήσεις που θα εκτελεστούν κατά την είσοδο στην κατάσταση ενώ στο κάτω μισό κομμάτι γράφονται οι συναρτήσεις που θα εκτελεστούν κατά την έξοδο από την κατάσταση. Οι συναρτήσεις αυτές μπορεί να γράφονται εξ’ ολοκλήρου μέσα στην κατάσταση ή να καλούνται με συγκεκριμένες παραμέτρους, ανατρέχοντας στο “κομμάτι συναρτήσεων” (function block), για να βρουν τον κώδικα της συγκεκριμένης συνάρτησης.
2. Δημιουργία μετάβασης (create transition): Από εδώ μπορούμε να δημιουργήσουμε μια γραμμή μετάβασης από μια κατάσταση σε μια άλλη. Οι μεταβάσεις ακολουθούν συγκεκριμένη κατεύθυνση (δεν είναι αμφίδρομες). Επίσης ορίζονται συνθήκες μετάβασης για το πότε και κάτω από ποιες προϋποθέσεις μια συγκεκριμένη μετάβαση θα συμβεί.
3. Καθορισμός αρχικής κατάστασης (set initial state): Κάθε διαδικασία που περιγράφεται στο μοντέλο επεξεργασίας, πρέπει να έχει μια αρχική κατάσταση, δηλαδή μια κατάσταση που θα εκτελεστεί πρώτη κατά την έναρξη της διαδικασίας. Στην αρχική κατάσταση, αρχικοποιούνται παράμετροι, δίνονται αρχικές τιμές σε μεταβλητές που χρειάζεται να έχουν συγκεκριμένη τιμή σε κάθε εκτέλεση της διαδικασίας καθώς ανακτώνται και παράμετροι από το προηγούμενο επίπεδο (Μοντέλο Κόμβων), που θα χρησιμοποιηθούν.
4. Μορφοποίηση μεταβλητών της διαδικασίας (edit state variables): Στο παράθυρο που “αναδύεται” με αυτή την επιλογή, δηλώνονται όλες οι μεταβλητές που χρησιμοποιούνται στον κώδικα που περιγράφει τις διάφορες συναρτήσεις της διαδικασίας. Για κάθε μεταβλητή, επιλέγεται ένας συγκεκριμένος τύπος (π.χ. int, double, float, char) καθώς επίσης μπορούν να γραφούν και σχόλια για το τι αντιπροσωπεύει η κάθε μία από αυτές. Οι μεταβλητές που δηλώνονται εδώ κρατούν την τιμή τους κάθε φορά που “τρέχει” η διαδικασία.
5. Μορφοποίηση προσωρινών μεταβλητών (edit temporary variables): Εδώ δηλώνονται βοηθητικές μεταβλητές, οι οποίες χάνουν την τιμή που έχουν μετά το τέλος της διαδικασίας και την επόμενη φορά που αυτή θα “τρέξει” χρειάζεται να αρχικοποιηθούν ξανά.
6. Μορφοποίηση κομματιού επικεφαλίδας (edit header block): Στο παράθυρο που “αναδύεται” με αυτή την επιλογή, δηλώνονται όλα τα εξωτερικά αρχεία που θα χρησιμοποιηθούν καθώς και οι έτοιμες

βιβλιοθήκες της C που χρειάζονται στον κώδικα (με include). Επίσης μπορούν να δηλωθούν μεταβλητές με σταθερές τιμές καθώς και αναγνωριστικά διακοπών (interrupts) που θα συμβούν στη διαδικασία (με define).

7. Μορφοποίηση κομματιού συναρτήσεων (edit function block): Εδώ γράφονται όλες οι συναρτήσεις που θα χρησιμοποιηθούν στην εκτέλεση της διαδικασίας. Οι μεταβλητές που αυτές χρησιμοποιούν, δηλώνονται είτε στην αρχή κάθε συνάρτησης είτε στις μεταβλητές της διαδικασίας (state variables). Ο κώδικας που γράφεται είναι σε γλώσσα C αλλά χρησιμοποιούνται και έτοιμες συναρτήσεις που προσφέρει ο πυρήνας του OPNET. Αυτές οι συναρτήσεις αφορούν κυρίως ανάκτηση και μεταφορά παραμέτρων ανάμεσα στα διάφορα επίπεδα, συλλογή στατιστικών στοιχείων, διακοπές κ.ά..

8. Μορφοποίηση διαγνωστικού κομματιού (edit diagnostic block): Εδώ συνήθως γράφονται συναρτήσεις που αφορούν την επικοινωνία της προσομοίωσης με τον έξω κόσμο κατά τη διάρκεια που αυτή εκτελείται. Για παράδειγμα οι συναρτήσεις αυτές τυπώνουν μηνύματα και τιμές παραμέτρων και μεταβλητών.

9. Μορφοποίηση κομματιού τερματισμού (edit termination block): Εδώ συνήθως γράφονται συναρτήσεις που αφορούν τον τερματισμό της διαδικασίας.

10. Μετατροπή κώδικα μοντέλου επεξεργασίας (compile process model): Με την επιλογή αυτή γίνεται το compilation του μοντέλου επεξεργασίας. Συντακτικά λάθη που εμφανίζονται στον κώδικα, λανθασμένη χρησιμοποίηση ονομάτων μεταβλητών ή μη σωστή ανάκληση μνήμης, οδηγούν σε σφάλμα κατά την μετατροπή του κώδικα.

5.2 Στόχοι της εξομίωσης στο OPNET Modeller

Ξεκινώντας αυτή την εργασία, τέθηκαν κάποιοι στόχοι, για το κομμάτι που αφορά την εξομίωση των δικτύων. Κατά τη διαδικασία της εκπόνησής της, έγινε η προσπάθεια να επιτευχθούν οι στόχοι αυτοί κατά το μέγιστο δυνατό βαθμό. Επίσης η τριβή με το πρόγραμμα OPNET, οδήγησε στην αναθεώρηση κάποιων αρχικών στόχων ή ακόμα και τη δημιουργία καινούριων. Ωστόσο, τα μοντέλα που δημιουργήθηκαν, σίγουρα μπορούν να δεχθούν βελτιώσεις και να αξιοποιηθούν σε μεγαλύτερο βαθμό τα εργαλεία που προσφέρει το OPNET. Παρόλα αυτά όμως, οι στόχοι της παρούσας διπλωματικής εργασίας, έχουν προσεγγιστεί σε αρκετά μεγάλο βαθμό και τα όσα έχουν φτιαχτεί, μπορούν να αποτελέσουν τη βάση για μελλοντικές εργασίες.

Ως αρχικός στόχος ήταν η δημιουργία ενός μοντέλου δρομολόγησης το οποίο θα μεγιστοποιούσε το διαθέσιμο throughput και θα χρησιμοποιούσε ένα μεγαλύτερο εύρος από τις διαθέσιμες συνδέσεις που θα προέκυπταν από τον αρχικό προς τον τελικό κόμβο. Αυτός ο στόχος θα μπορούσε να επιτευχθεί αν εισαχθεί στο μοντέλο η έννοια των πολλαπλών διαδρομών όπου κάθε κόμβος κρατάει παραπάνω από μια διαδρομές για να στείλει πακέτα στον επόμενο κόμβο. Επίσης στην πορεία αποφασίστηκε να δημιουργηθεί ένα πιο ιεραρχικό μοντέλο που θα εξασφάλιζε μια ανεξαρτησία των κόμβων που θα άνηκαν σε συγκεκριμένες ομάδες. Χρησιμοποιώντας ένα ιεραρχικό μοντέλο κρατάμε χαμηλή την πολυπλοκότητα του δικτύου αφού η επικοινωνία θα γίνεται μόνο μέσω των gateways άρα θα έχουμε

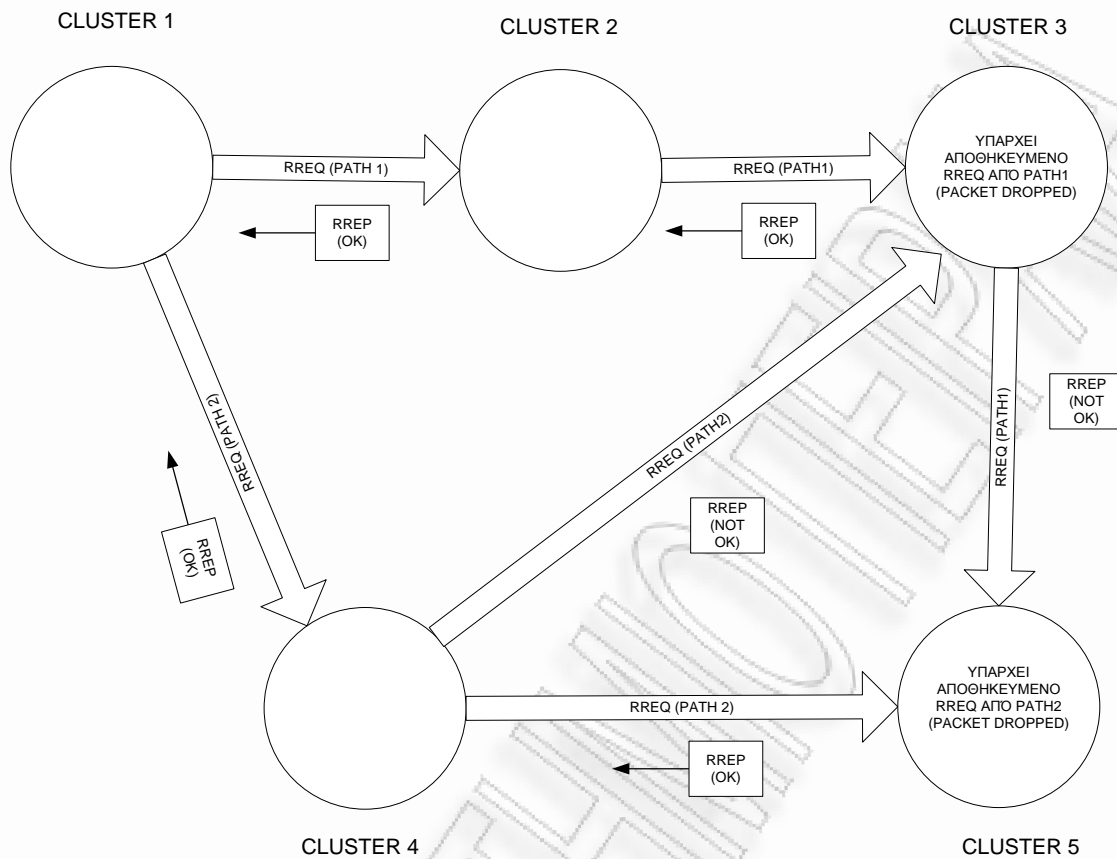
λιγότερες διαδρομές και μικρότερα routing tables. Με αυτό τον τρόπο κάποιο σφάλμα σε κάποιο κόμβο ή σύνδεση θα απέτρεπε την μεγάλη πτώση του bandwidth αφού η κάθε ομάδα κόμβων δεν γνωρίζει την τοπολογία της άλλης. Εισάγοντας λοιπόν το ιεραρχικό μοντέλο μέσα στην προσομοίωση προέκυψαν δύο καινούργιες έννοιες:

- Inter-Cluster Routing: Για διαδρομές από ένα cluster σε ένα άλλο (αποτελεί και το βασικό κομμάτι της προσομοίωσης)
- Intra-Cluster Routing: Για διαδρομές εσωτερικά κάθε cluster.

Αν και η τεχνική multicluster routing περιορίζει σε σημαντικό βαθμό τις παρεμβολές είναι πολύ πιθανόν να εμφανιστούν προβλήματα τύπου route joint. Ο λόγος αυτού του προβλήματος οφείλεται στην επιλογή cluster-by-cluster συνδέσεων (links) και αποτελεί αρνητικό παράγοντα στην επίδοση του δικτύου γιατί μειώνεται αισθητά το throughput.

Για να αποφευχθεί αυτό το πρόβλημα, προτείνεται η παρακάτω λύση η οποία μπορεί να εφαρμοστεί κατά τη διάρκεια εύρεσης των πολλαπλών διαδρομών:

- Κάθε κόμβος κρατάει ένα table στο οποίο αποθηκεύει την διαδρομή από όπου έλαβε το πακέτο (RREQ).
- Αποθηκεύει τα απαραίτητα στοιχεία που είναι απαραίτητα για την δρομολόγηση και στέλνει πίσω στον προορισμό το μήνυμα (RREP)
- Σε περίπτωση που ένας κόμβος λάβει μήνυμα RREQ από άλλο route (όπως φαίνετε και στο σχήμα) ελέγχει τη διεύθυνση πηγής (source address). Αν η διεύθυνση είναι ίδια με αυτή που έχει ήδη αποθηκεύσει και απαντήσει στο πρώτο RREQ τότε το δεύτερο μήνυμα που θα λάβει θα γίνει dropped για την αποφυγή route join κόμβων.
- Ο κόμβος προορισμού επιβεβαιώνει ότι δεν υπάρχει route join στο multiroute και στέλνει μήνυμα επιβεβαίωσης (RREP) ώστε να ξεκινήσει η δρομολόγηση.



Σχήμα 11: Multi-route with route join

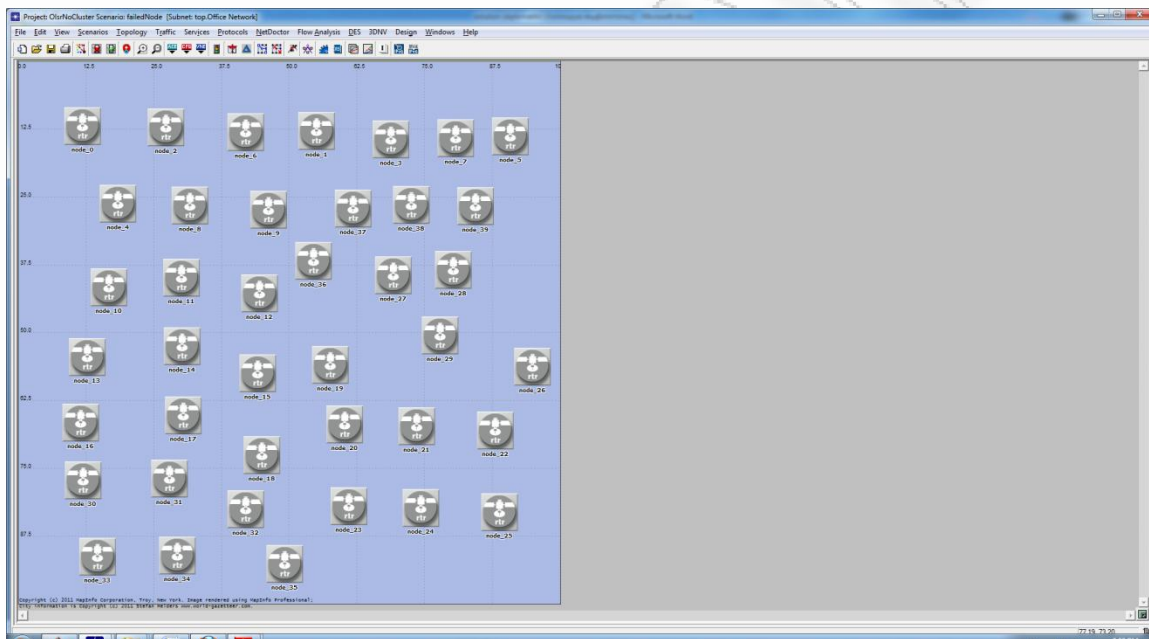
Πιο αναλυτικά:

- Για Inter-cluster Routing
 - Ο gateway κόμβος που ξεκινάει το route discovery και δημιουργεί RREQ μηνύματα.
 - Ο gateway κόμβος σε άλλο cluster λαμβάνει το RREQ μήνυμα το οποίο περιέχει και την αρχική διεύθυνση (source address) . Η διεύθυνση πηγής αποθηκεύεται σε μια μεταβλητή buffer. Σε περίπτωση που η διεύθυνση δεν είναι αποθηκευμένη στο routing table κάνει την εγγραφή. Αν υπάρχει ήδη εγγραφή με το ίδιο source address τότε το πακέτο γίνεται dropped
- Για Intra-cluster Routing
 - Δεν υπάρχει κάποια εισαγωγή flag ή κάποια άλλη διαφοροποίηση στους κόμβους.

- Ο cluster head-gateway λαμβάνει το πακέτο και το προωθεί στον κόμβο προορισμού με την χρήση εσωτερικού πρωτοκόλλου δρομολόγησης

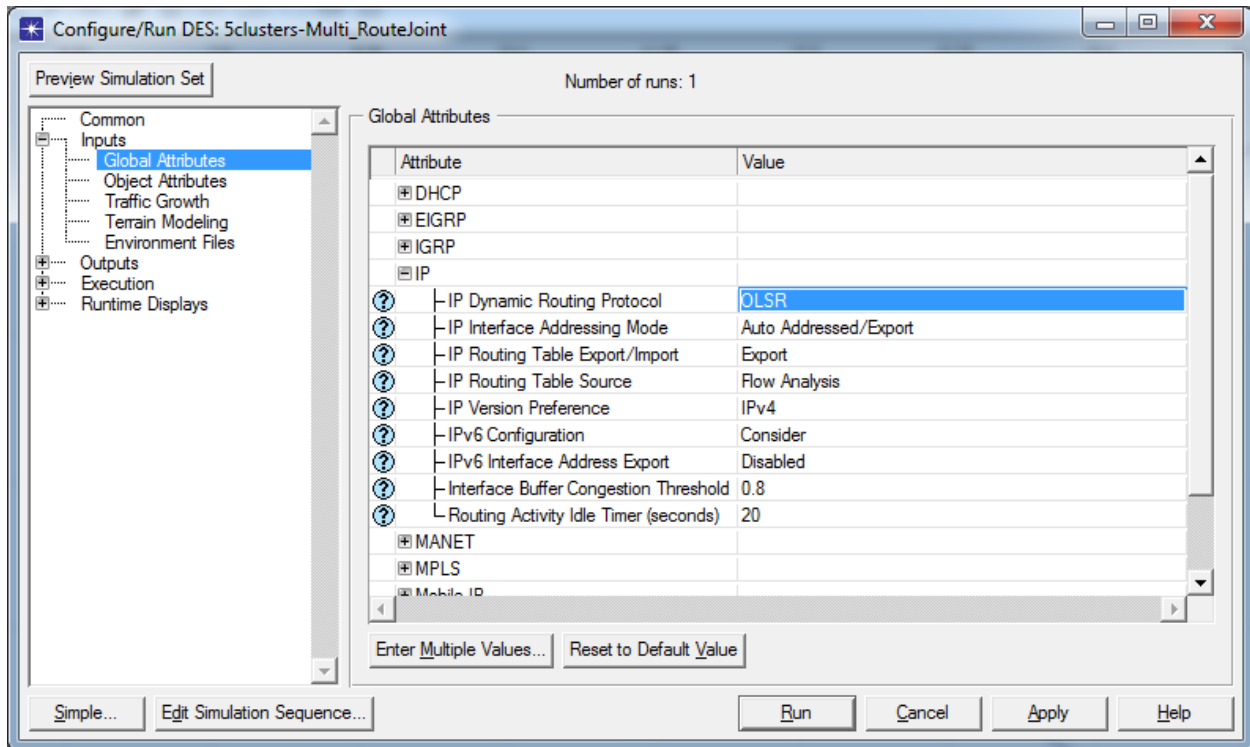
ΠΡΟΣΩΜΟΙΩΣΗ ΣΤΟ ΟΡΝΕΤ

Αρχικά στην προσομοίωση θέλουμε να δείξουμε ότι αν οργανώσουμε το δίκτυο με την εισαγωγή clusters αυξάνεται η απόδοση του. Επομένως υποθέτουμε ότι στο δίκτυο υπάρχουν 50 κόμβοι οι οποίοι θα εισάγονται σε ένα cluster. Θα εκτελέσουμε την προσομοίωση για $C=2$, $C=5$, $C=10$ (όπου C ο αριθμός των cluster). Αρχικά η τοπολογία του δικτύου φαίνεται στην εικόνα 1.



Εικόνα 9: Τοπολογία προσομοίωσης για $C=0$

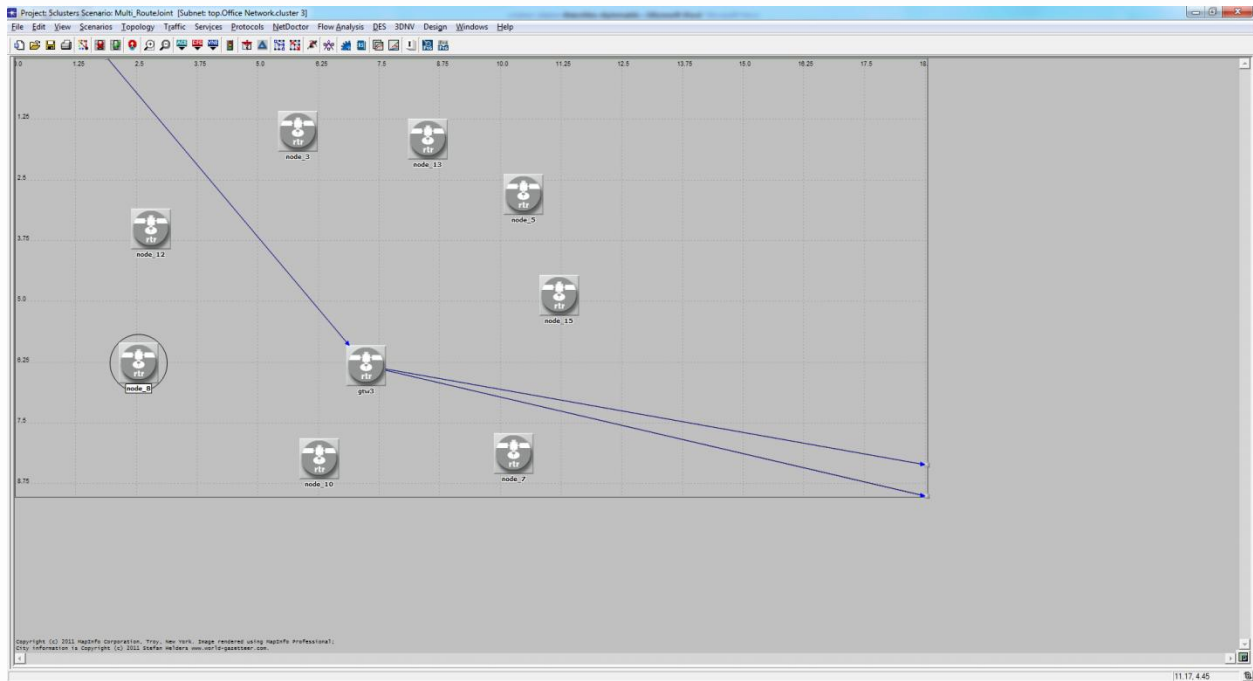
Στην συγκεκριμένη τοπολογία δεν υπάρχουν κάποια clusters επομένως οποιαδήποτε failure σε κάποιον κόμβο θα επηρεάσει όλη την τοπολογία. Θεωρούμε ότι ο αλγόριθμος που χρησιμοποιείται είναι ο OLSR σύμφωνα με τον οποίο όλοι οι κόμβοι στην τοπολογία γνωρίζουν τις πιθανές διαδρομές που μπορεί να ακολουθήσει το πακέτο προς όλους τους υπόλοιπους κόμβους. Το ornet έχει εισάγει τον αλγόριθμο OLSR μέσα στις ρυθμίσεις επομένως τα βήματα που ακολουθήσαμε (εικόνα 2) ήταν να τρέξουμε την προσομοίωση με αυτό τον αλγόριθμο και να δούμε τα αποτελέσματα που θα βγουν για το throughput και το delay.



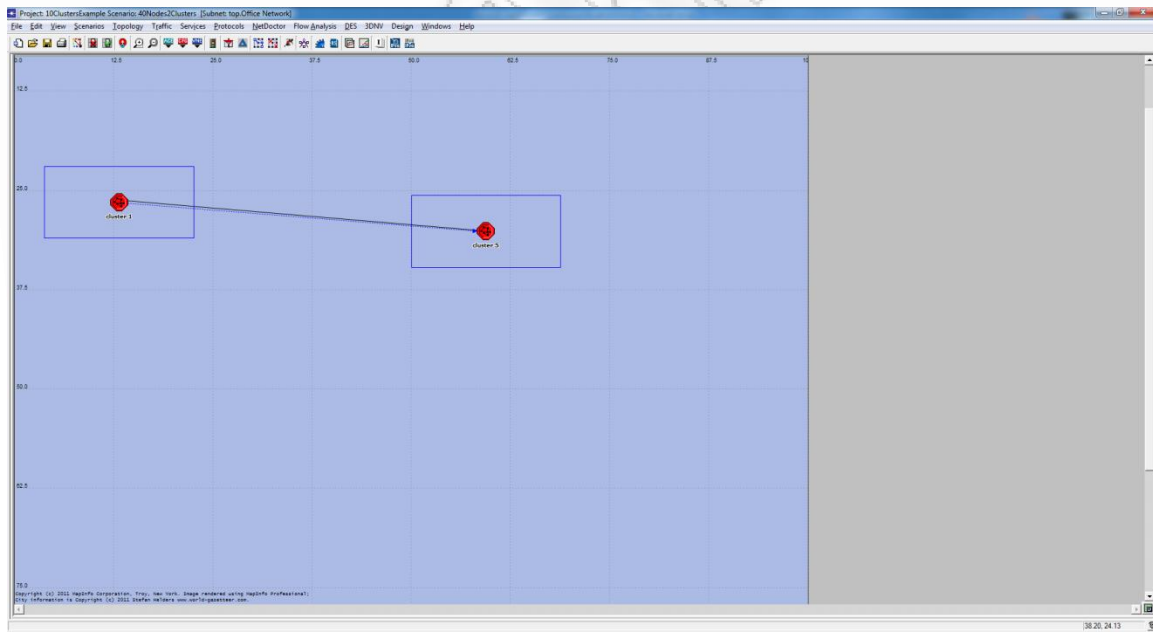
Εικόνα 10: Αλγόριθμος OLSR μέσα από την κονσόλα διαχείρισης.

Πρέπει να σημειωθεί τα αποτελέσματα που βγαίνουν στον προσομοιωτή αποτελούν ένα μέσο όρο των αποτελεσμάτων που έβγαλε η προσομοίωση επομένως στην περίπτωση όπου δεν υπάρχει η δημιουργία των cluster είναι λογικό το throughput και η καθυστέρηση να μην έχουν ιδιαίτερα σημαντικές αυξομειώσεις αλλά να ακολουθούν ένα σταθερό ρυθμό.

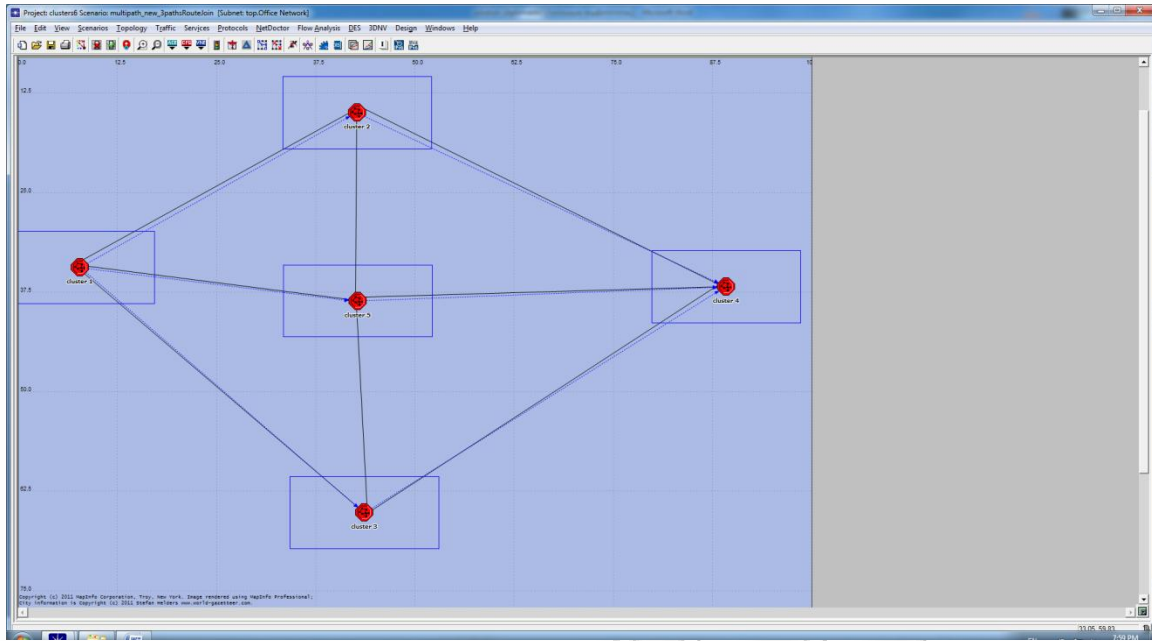
Το επόμενο βήμα της προσομοίωσης είναι να χωρίσουμε τους κόμβους σε clusters. Επομένως για $C=2$ θα έχουμε 2 clusters από 25 κόμβους σε κάθε cluster, για $c=5$ θα έχουμε 10 κόμβους και τέλος για $C=10$ θα έχουμε 5κόμβους. Κατά την δημιουργία των cluster θεωρήσαμε ότι υπάρχει σε κάθε ομάδα ένα cluster gateway ο οποίος θα έπαιζε και τον ρόλο και του cluster head όπου και θα μοιράζει την πληροφορία προς του υπόλοιπους κόμβους μέλη. Στην εικόνα 5 φαίνεται η τοπολογία στο εσωτερικό ενός cluster και στις εικόνες 6, 7 και 8 φαίνεται η συνολική εικόνα της τοπολογίας.



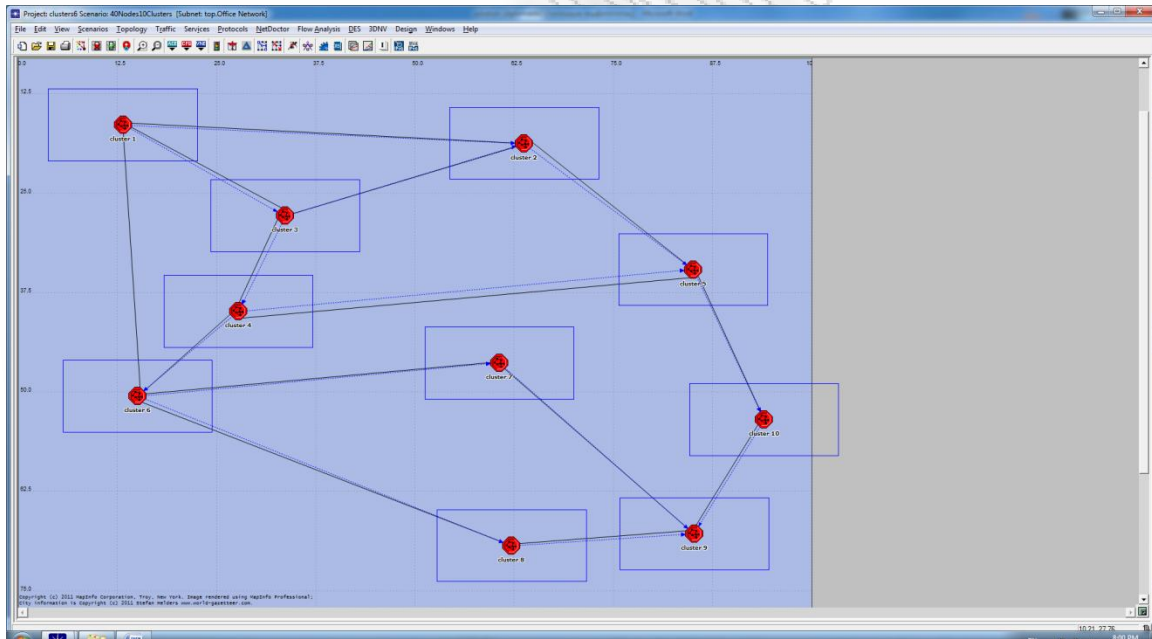
Εικόνα 13: Εσωτερική τοπολογία των cluster



Εικόνα 14: Τοπολογία για C=2

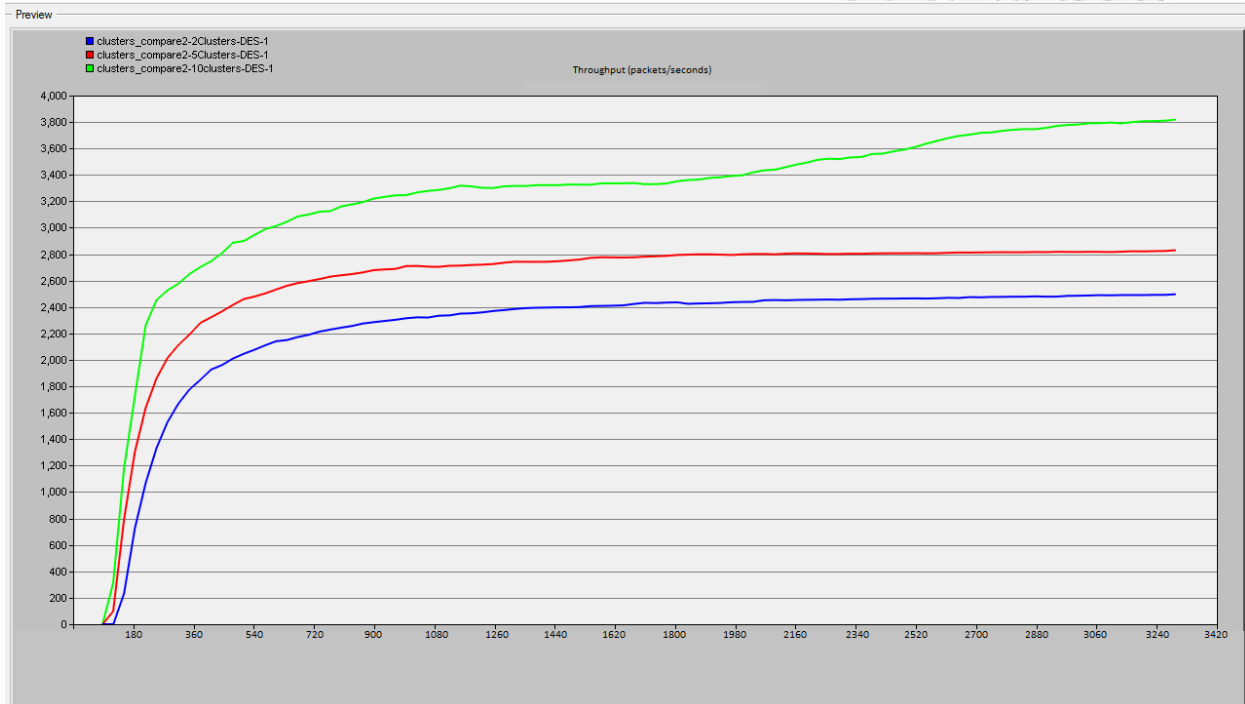


Εικόνα 15: Τοπολογία για C=5



Εικόνα 16: Τοπολογία για C=10

Σκοπός αυτής της προσομοίωσης είναι να δείξουμε ότι χρησιμοποιώντας περισσότερα cluster αυξάνεται η απόδοση του δικτύου και μένει σε ένα σταθερό όριο αν προκύψει κάποιο σφάλμα σε έναν συγκεκριμένο αριθμό κόμβων. Αυτό οφείλεται στο γεγονός ότι κάθε cluster είναι ανεξάρτητο από τα υπόλοιπα επομένως κάποιο σφάλμα σε ένα cluster μπορεί να μην επηρεάσει καθόλου το throughput στα υπόλοιπα clusters. Επομένως επαναλαμβάνοντας την ίδια διαδικασία στην κονσόλα διαχείρισης του ornet εμφανίστηκαν τα παρακάτω αποτελέσματα του throughput για C=2 C=5 και C=10.



Εικόνα 17: Throughput για C=2, C=5 και C=10

Παρατηρούμε ότι για C=10 (πράσινη γραμμή) υπάρχει το μεγαλύτερο throughput στο δίκτυο και ανάλογα όσο μικραίνει το C (C=2 μπλε γραμμή) μικραίνει και η απόδοση. Εφόσον δείξαμε ότι με την τεχνική των clusters αυξάνεται η απόδοση του δικτύου, χρησιμοποιούμε την συγκεκριμένη τεχνική για την τεχνική multi-route που προτείνουμε παραπάνω. Επομένως για να δείξουμε ότι με την τεχνική multi-route joint avoidance υπάρχει αύξηση της επίδοσης θα συγκρίνουμε τα αποτελέσματα για προσομοιώσεις που αφορούν:

1) Cluster δίκτυα που χρησιμοποιούν την τεχνική single-route, δηλαδή η δρομολόγηση θα γίνεται μόνο από μία διαδρομή.

2) Cluster δίκτυα που χρησιμοποιούν την τεχνική multi-route χωρίς όμως την τεχνική route-joint avoidance, δηλαδή πολλαπλές διαδρομές χωρίς όμως να υπολογίζεται ότι μερικά links και μερικοί κόμβοι θα χρησιμοποιούνται για παραπάνω από μία διαδρομές.

3) Cluster δίκτυα που χρησιμοποιούν την τεχνική multi-route που χρησιμοποιούν την τεχνική route-joint avoidance, δηλαδή την τεχνική που αναλύθηκε παραπάνω.

Για να επιτευχθεί η τεχνική του multi-route χρησιμοποιήθηκε η επιλογή για multiroute threshold που υπάρχει στην κονσόλα διαχείρισης του ornet. Αυτή η παράμετρος μπορεί να τροποποιηθεί ανάλογα με τις διαθέσιμες εναλλακτικές διαδρομές που μπορεί να έχει ένας κόμβος. Επίσης επειδή δημιουργήθηκε ένα ιεραρχικό μοντέλο χρησιμοποιήθηκαν δύο ειδών αλγόριθμοι:

- OLSR: Ανήκει στην κατηγορία των proactive αλγορίθμων και χρησιμοποιήθηκε για Intra-Cluster routing.
- AODV: Ανήκει στην κατηγορία των reactive αλγορίθμων και χρησιμοποιήθηκε για Inter-Cluster routing

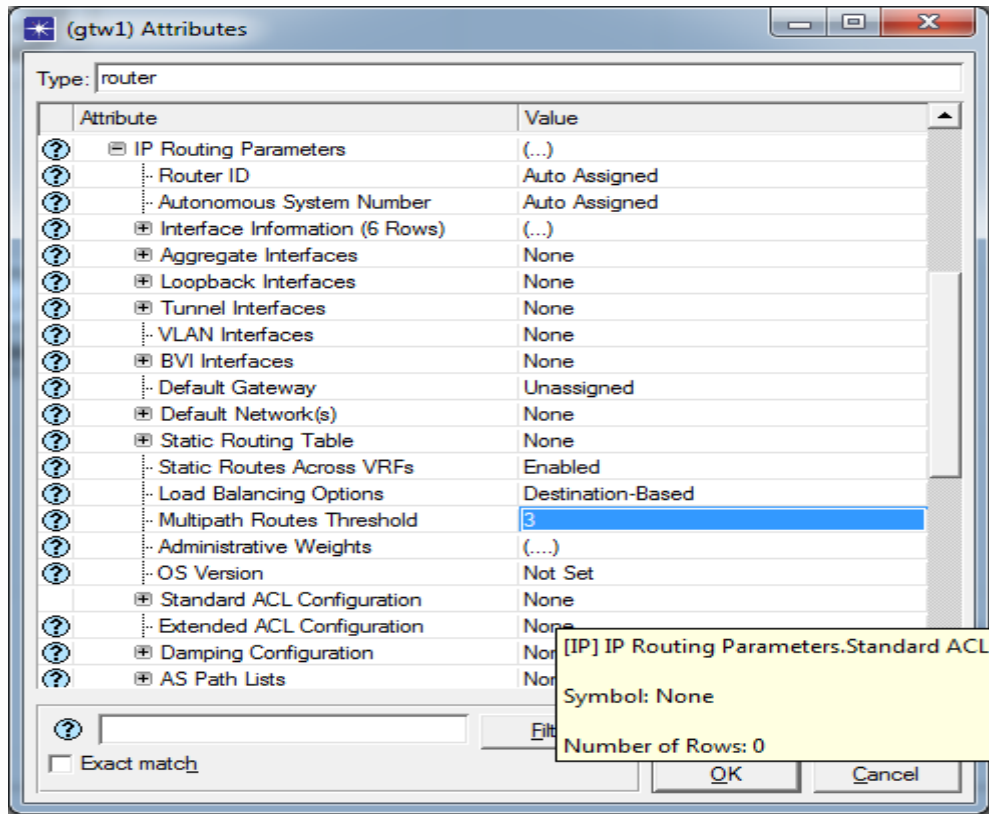
Η επιλογή χρήσης δύο διαφορετικών αλγορίθμων έγινε για τον εξής σκοπό: Μέσα στο cluster πρέπει η τοπολογία να είναι γνωστή σε όλους τους κόμβους. Επομένως με την χρήση ενός proactive αλγορίθμου οι κόμβοι χτίζουν το routing table πριν ξεκινήσει κάποια αίτηση για νέα διαδρομή. Επίσης επειδή υπάρχει και ένας cluster head κόμβος ο οποίος μιλάει με τα άλλα clusters πρέπει οι κόμβοι μέσα σε κάθε cluster να γνωρίζουν που πρέπει να στείλουν το πακέτο σε περίπτωση που ο τελικός κόμβος ανήκει σε διαφορετικό cluster.

Στην περίπτωση του Inter-Cluster routing πρέπει το overhead αλλά και το utilization να είναι μειωμένο ώστε η απόδοση του δικτύου να είναι σε υψηλά επίπεδα. Για τον λόγο αυτό χρησιμοποιήθηκε ένας reactive αλγόριθμος κατά τον οποίο κάθε κόμβος δημιουργεί το routing table μόνο εάν γίνει αίτηση για κάποια διαδρομή. Επομένως υπάρχει περίπτωση αν δε γίνει αίτηση για δρομολόγηση σε κάποιο άλλο cluster μπορεί ο cluster head να μην έχει εγγραφές στον πίνακα δρομολόγησης κανένα εξωτερικό cluster.

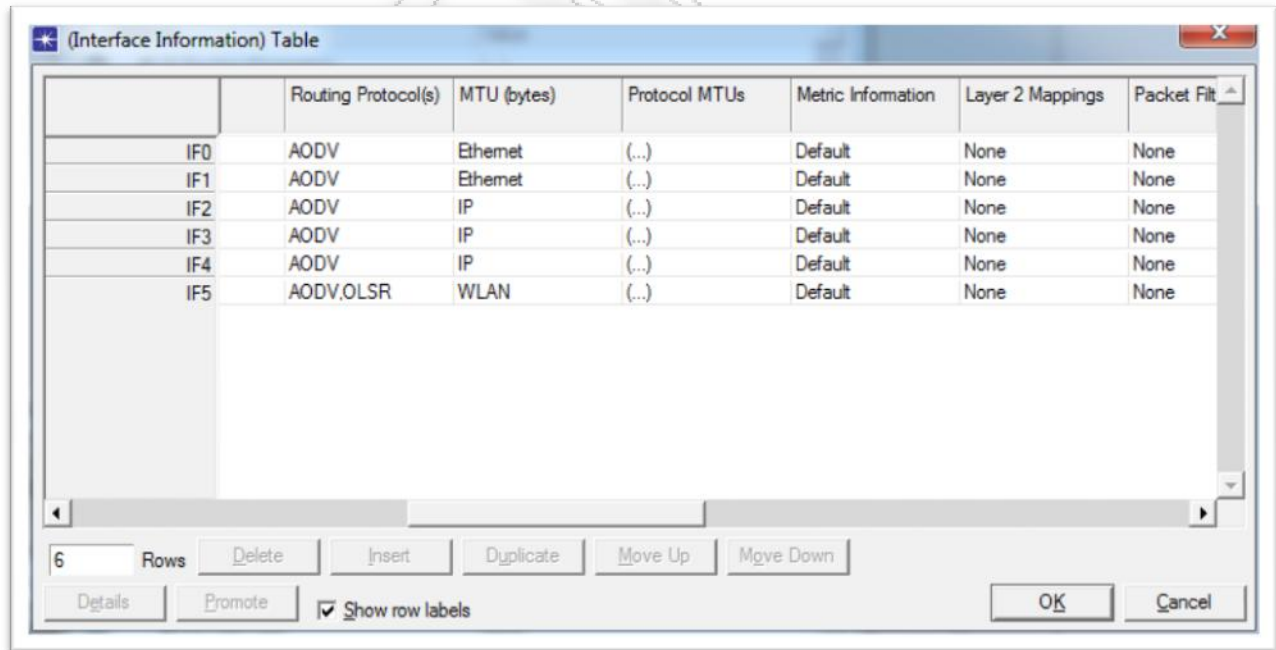
Για να επιτευχθεί η τεχνική του inter και intra cluster routing έγινε η εξής τροποποίησης στην κονσόλα διαχείρισης του ornet:

- Σε κάθε cluster gateway δίνεται η δυνατότητα να χρησιμοποιούν δύο αλγόριθμους (AODV, OLSR)
- Στις παραμέτρους της προσομοίωσης ορίστηκε ως πρωτόκολλο custom το οποίο παίρνει τις προεπιλογές στο interface που υπάρχει στο κάθε cluster gateway.

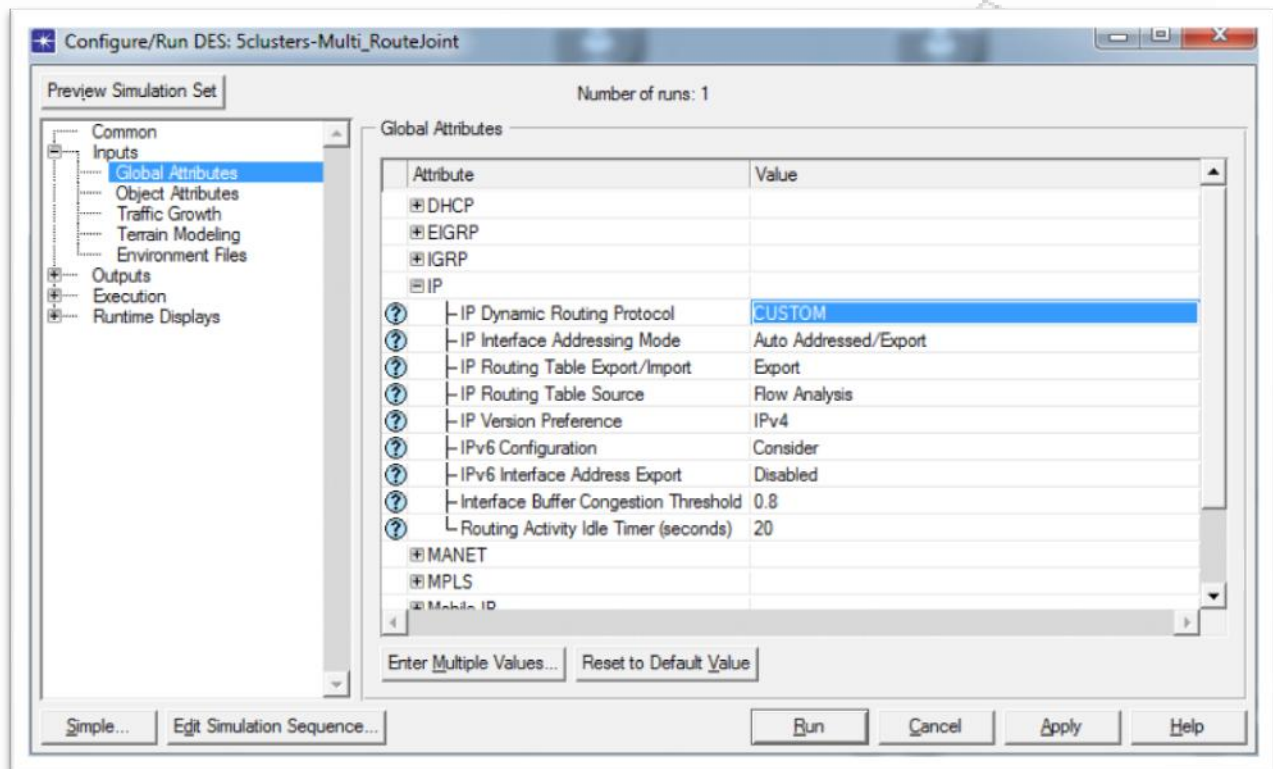
Στις παρακάτω εικόνες φαίνονται και τροποποιήσεις που έγιναν



Εικόνα 19: Τροποποίηση πολλαπλών διαδρομών

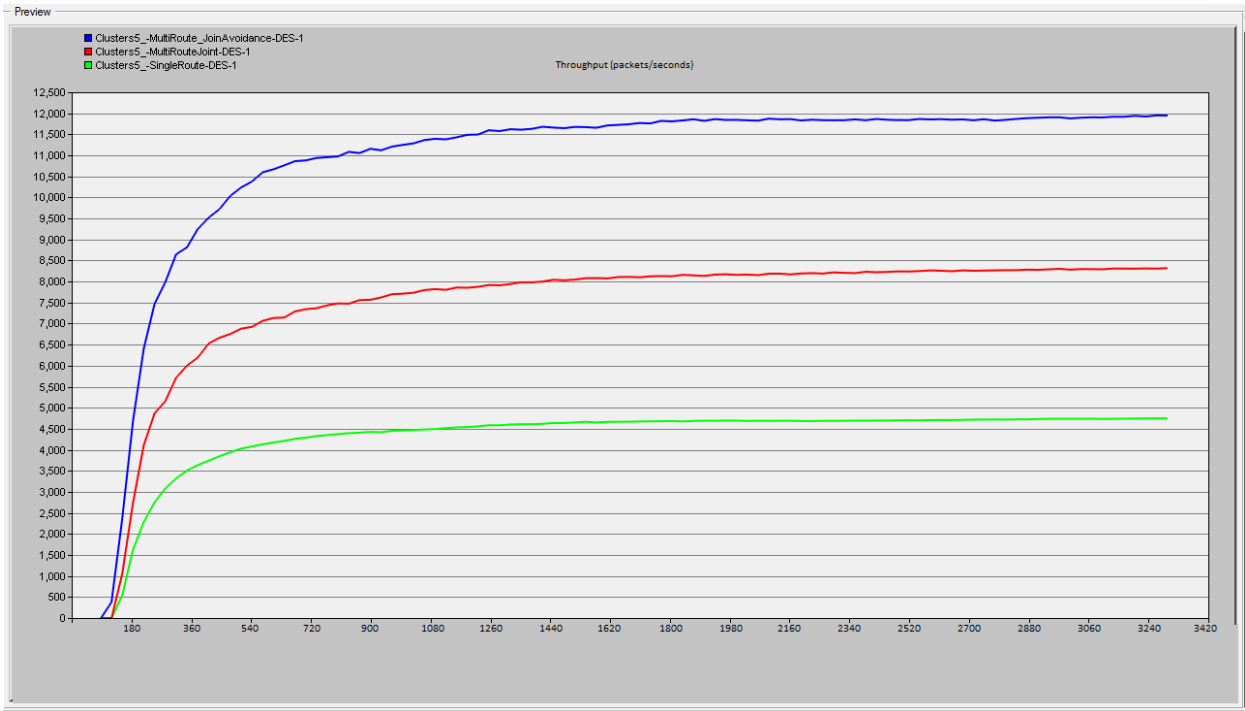


Εικόνα 20: Το ασύρματο interface IF5 δέχεται δύο αλγορίθμους

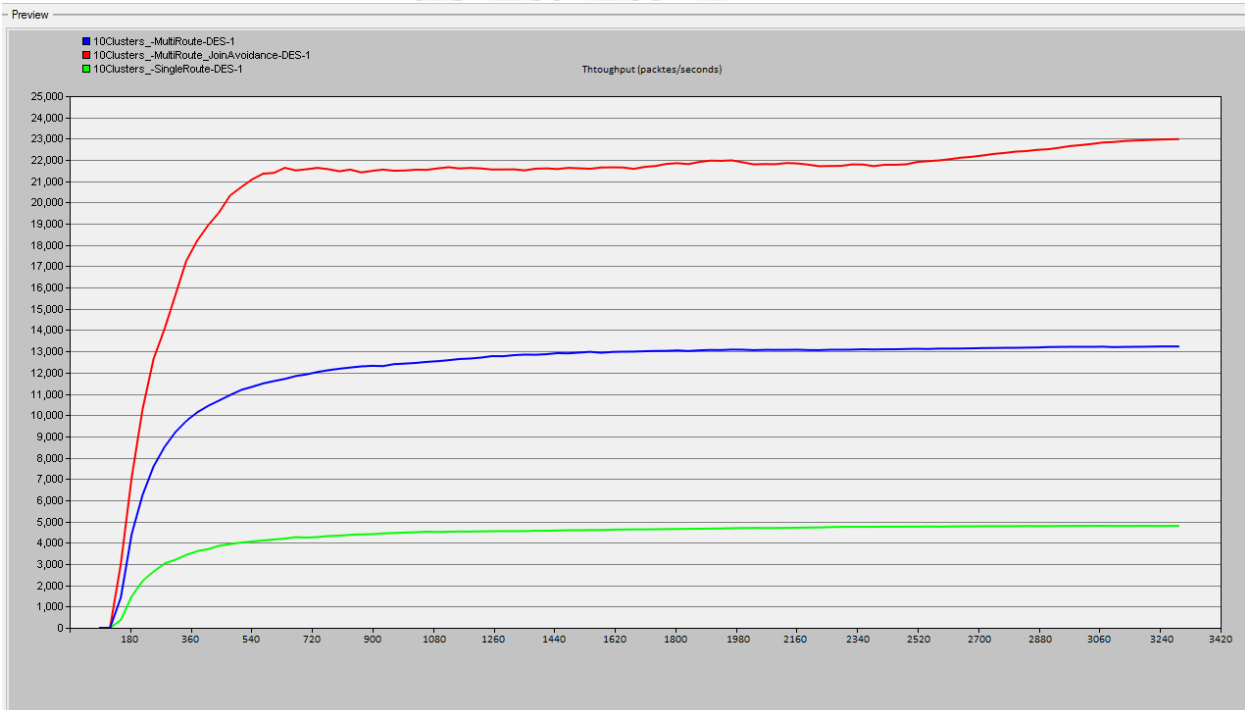


Εικόνα 21: Παράμετροι προσομοίωσης

Ως αποτελέσματα τις προσομοίωσης βάλουμε το throughput και το delay. Αυτό που μας ενδιαφέρει να δείξουμε είναι ότι χρησιμοποιώντας την τεχνική multi-route joint avoidance υπάρχει αύξηση στο throughput άρα και στην απόδοση του δικτύου. Επίσης εξετάζονται και τα αποτελέσματα της καθυστέρησης στο δίκτυο ώστε να αποδείξουμε ότι η χρήση αυτής της τεχνικής αποτελεί και αξιόπιστη μεταφορά πακέτων. Οι παρακάτω εικόνες δείχνουν την διαφορά που προκύπτει στο throughput και στο delay χρησιμοποιώντας και τις τρεις τεχνικές που αναφέρθηκαν παραπάνω αρχικά για C=5 και τέλος για C=10.



Εικόνα 22: Σύγκριση αποτελεσμάτων για single-route, multi-route, multi-route route-join avoidance για C=5

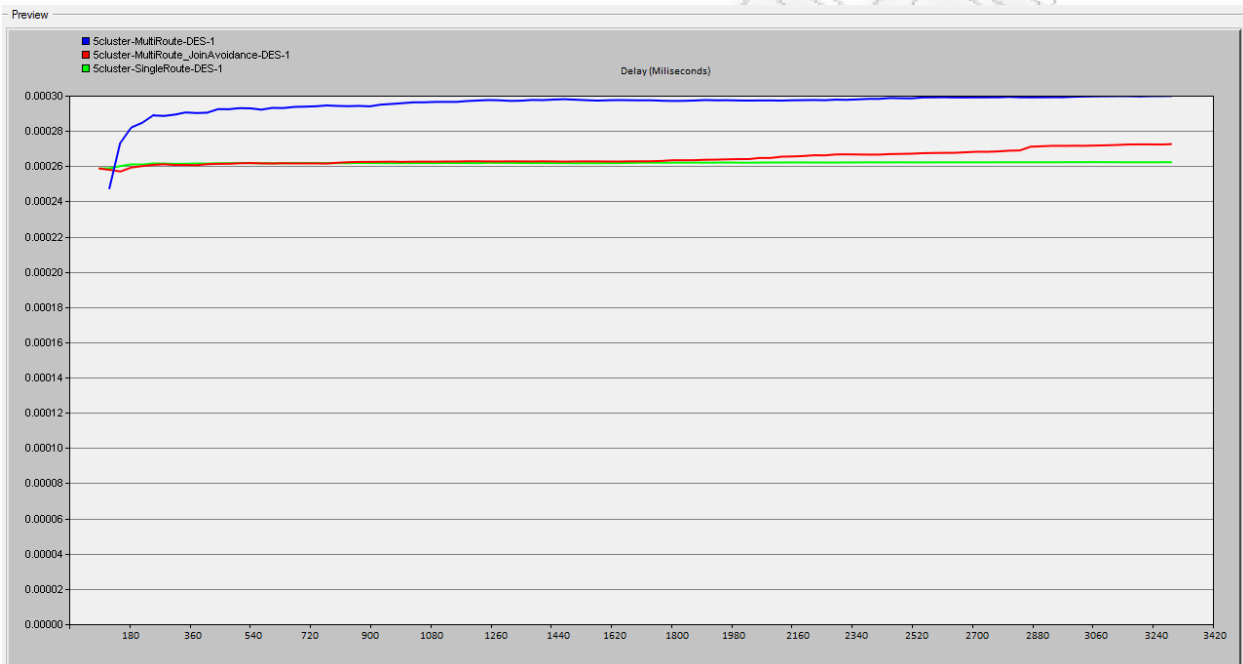


Εικόνα 23: Σύγκριση αποτελεσμάτων για single-route, multi-route, multi-route route-join avoidance για C=10

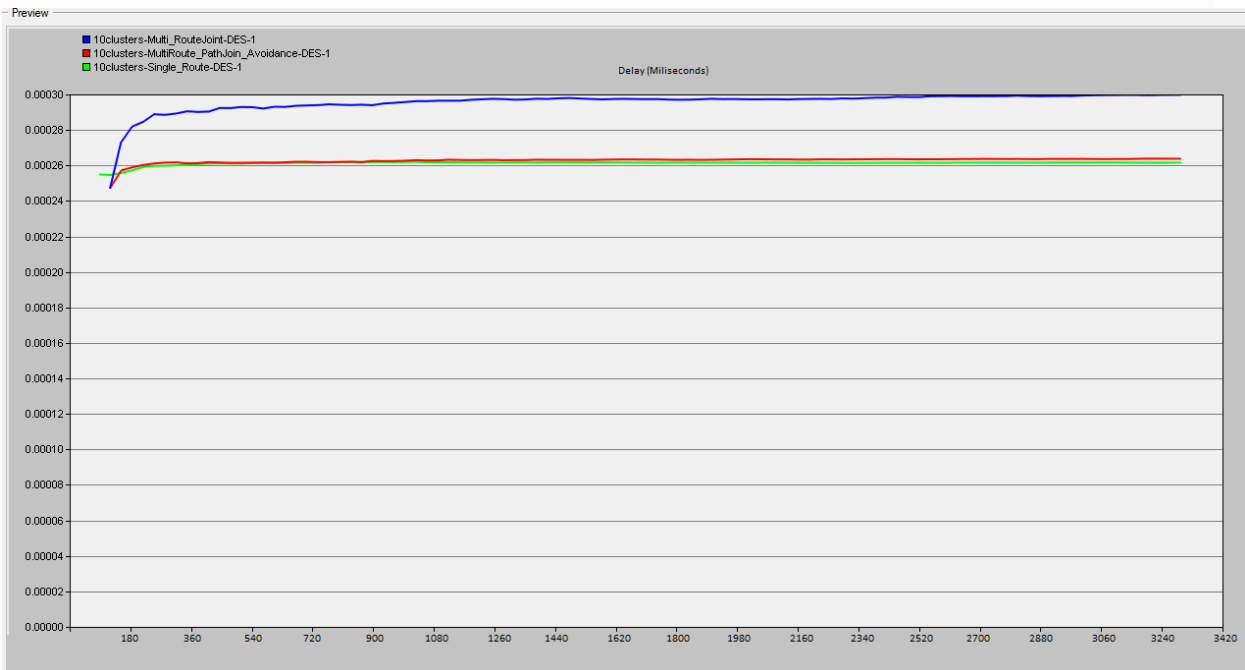
Όπως βλέπουμε το throughput αυξάνεται όταν χρησιμοποιούμε την τεχνική route-join avoidance (μπλε γραμμή για C=5 και κόκκινη γραμμή για C=10) γιατί τα πακέτα ακολουθούν μοναδικές διαδρομές μέχρι τον προορισμό κι έτσι το utilization κάθε γραμμής δεν υπερφορτώνεται από άλλες διασταυρωμένες διαδρομές.

Επίσης βλέπουμε ότι η τεχνική multi-route έχει μεγαλύτερο throughput από την single-route (πράσινη γραμμή).

Στις εικόνες 23 και 24 φαίνεται η καθυστέρηση και στις τρεις παραπάνω τεχνικές.



Εικόνα 24: Σύγκριση delay για single-route, multi-route, multi-route route-join avoidance για C=5



Εικόνα25: Σύγκριση delay για single-route, multi-route, multi-route route-join avoidance για C=10

Σε ότι αφορά το delay παρατηρούμε ότι η τεχνική που προτείνουμε βρίσκεται περίπου στα ίδια επίπεδα καθυστέρησης με την single-route τεχνική επομένως με αυτό τον τρόπο τα πακέτα ακολουθούν τις καλύτερες δυνατές διαδρομές χωρίς να επηρεάζεται η καθυστέρηση του δικτύου.

6. ΠΑΡΑΤΗΡΗΣΕΙΣ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ

Κοιτάζοντας τα αποτελέσματα της εξομοίωσης, στους πιο πάνω πίνακες και γραφικές παραστάσεις, μπορούμε να κάνουμε ορισμένες παρατηρήσεις και να συγκρίνουμε τις διάφορες τεχνολογίες ως προς την επίδοσή τους. Έχοντας ως αρχικό στόχο την δημιουργία ενός ιεραρχικού μοντέλου δρομολόγησης δείξαμε ότι με την χρήση cluster και cluster heads επιτυγχάνεται μια ανεξαρτησία στο δίκτυο η οποία προσφέρει καλύτερη επίδοση ειδικά στην περίπτωση όπου παρουσιάζονται πολλά σφάλματα σε συνδέσεις και κόμβους. Επομένως επειδή κάθε cluster λειτουργεί σαν ένα "διαφορετικό δίκτυο" από τα υπόλοιπα cluster κάθε παράγοντας που επηρεάζει την απόδοση ή την καθυστέρηση μένει εσωτερικά και διανύει όλη την τοπολογία.

Έχοντας επιτύχει τον πρώτο στόχο, δείξαμε έπειτα τον βαθμό που βελτιώνεται το throughput χρησιμοποιώντας τεχνικές multipath και συγκρίνοντας τις με τους αλγορίθμους μονής διαδρομής

(single route). Μέσα στις γραφικές παραστάσεις προστέθηκε και η λύση που προτείνουμε η οποία όπως αναμενόταν παρουσιάζεται ως βέλτιστη για τα σενάρια που εξετάσαμε (5 και 10 clusters αντίστοιχα). Πιο συγκεκριμένα τα αποτελέσματα έδειξαν οι single route τεχνικές δεν αξιοποιούν το εύρος ζώνης κάθε γραμμής επομένως είναι λογικό να έχουν το χαμηλότερο throughput. Επιπλέον οι απλές τεχνικές multi route είναι πιθανό να χρησιμοποιούν ήδη χρησιμοποιημένες διαδρομές για την αποστολή και λήψη πακέτων με αποτελέσματα να υπάρχει ένα αυξημένο overhead στις περιοχές όπου υπάρχει αυτό το φαινόμενο. Αντίθετα η τεχνική που προτείνουμε αποτρέπει αυτή την χρήση και χρησιμοποιεί ανεξάρτητες διαδρομές πράγμα που αυξάνει το throughput του δικτύου.

Τέλος, παρουσιάζεται η παράμετρος της καθυστέρησης η οποία υπολογίζει την συνολική καθυστέρηση από την πηγή στον προορισμό. Τα αποτελέσματα δείχνουν ότι η τεχνική multiroute που προτείνουμε πλησιάζει σε αρκετά ικανοποιητικό βαθμό την καθυστέρηση που υπάρχει στην single route τεχνική. Στα πιο υψηλά επίπεδα βρίσκεται η multi route joint τεχνική η οποία εξαιτίας των κοινών διαδρομών που χρησιμοποιούνται δημιουργεί μεγαλύτερο χρόνο επεξεργασίας και χρόνο αναμονής στους κόμβους ώστε να δρομολογήσουν τα πακέτα. Χρησιμοποιώντας την τεχνική που προτείνουμε ο χρόνος επεξεργασίας και αναμονής δεν είναι μεγάλος γιατί δεν υπάρχουν διασταυρομένες διαδρομές μεταξύ αρχικού και τελικού κόμβου.

6.1 ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

Οι στόχοι της παρούσας διπλωματικής εργασίας, έχουν προσεγγιστεί σε αρκετά μεγάλο βαθμό και τα όσα έχουν φτιαχτεί, μπορούν να αποτελέσουν τη βάση για μελλοντικές εργασίες. Το κύριο κομμάτι της εργασίας ήταν η ανάλυση σε ασύρματα δίκτυα κορμού όπου παρουσιάζεται χαμηλή έως καθόλου παρεμβολές. Κατά την δημιουργία πολλαπλών διαδρομών είναι φυσικό οι παρεμβολές μεταξύ των κόμβων να αυξάνονται. Επομένως σαν μελλοντική έρευνα μπορεί να είναι η ανάλυση της συμπεριφοράς του δικτύου όταν εισαχθεί σε αυτό και η παράμετρος της παρεμβολής. Επίσης θα μπορούσε να γίνει μια πιο λεπτομερής μοντελοποίηση της ασύρματης διάδοσης. Να εισαχθεί δηλαδή η παράμετρος SNR (signal to noise ration) και path loss οι οποίες όταν αλλάζουν θα πρέπει να ανανεώνονται και οι routing αποφάσεις.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Amir, Y., C. Danilov, M. Kaplan, R. Musaloiu-Elefteri, & N. Rivera (2008, June). On redundant multipath operating system support for wireless mesh networks. In Sensor, Mesh and Ad Hoc Communications and Networks Workshops, 2008. SECON Workshops '08.
2. Badis, H. & K. Al Agha (2005). QOLSR, QoS routing for ad hoc wireless networks using OLSR. European Transactions on Telecommunications.
3. Basagni, S. (2008). Distributed clustering for ad hoc networks. Published by the IEEE Computer Society.
4. Chan, H. & A. Perrig (2008). Ace: An emergent algorithm for highly uniform cluster formation. In in Proceedings of the First European Workshop on Sensor Networks
5. Jacquet, P., T. Clausen, A. Laouiti, A. Qayyum, & L. Viennot (2001). Optimized link state routing protocol for ad hoc networks. In Proceedings of the 5th IEEE Multi Topic Conference (INMIC 2001)
6. Lee, S.-J. & M. Gerla (2010). Split multipath routing with maximally disjoint paths in ad hoc networks. In Communications, 2010. ICC 2010.
7. Liaw, Y., A. Dadej, & A. Jayasuriya (2007, jun.). Throughput performance of multiple independent paths in wireless multihop network. In IEEE International Conference on Communications.
8. Marc Mosko, J. G.-L.-A. (2005). Multipath routing in wireless mesh networks. In in Proc. IEEE Workshop on Wireless Mesh Networks (WiMesh)
9. Rangarajan, S. (2007). On demand loop free multipath routing in ad hoc networks using source sequence numbers.
10. Tsai, J. & T. Moors (2006). A review of multipath routing protocols: From wireless ad hoc to mesh networks.
11. Waharte, S. & R. Boutaba (2006, jun.). Totally disjoint multipath routing in multihop wireless networks.
12. Yi, J., A. Adnane, S. David, & B. Parrein (2010). Multipath Optimized Link State Routing for Mobile ad hoc Networks. Ad Hoc Networks.
13. Yashar G. and Abtin K.: Load balancing in Ad hoc networks: single-path routing vs. multipath routing. IEEE Infocom, vol 2,(2004)

14. Xiaoguang Niu, Zihua Tao, Gongyi Wu, Changcheng Huang, Li Cui.: Hybrid Cluster Routing: An Efficient Routing Protocol for Mobile Ad Hoc Networks. Communications, IEEE International Conference, (2006)
15. N. F. Maxemchuk, "Diversity Routing," in Proc. IEEE ICC, San Francisco, CA, 1975.
16. C. Perkins, "Ad hoc On-Demand Distance Vector (AODV) Routing,
17. M. K. Marina and S. R. Das, "Ad hoc On-demand Multipath Distance Vector Routing," Computer Science Department, Stony Brook University, 2003.
18. Marina, M.K., Das, S.R.: On-demand Multipath Distance Vector Routing in Ad Hoc Networks. Proceedings of the International Conference for Network Protocols (2009)
19. Tsung-Chuan Huang, Chun-Kai Liao, Chyi-Ren Dow "Zone-based hierarchical routing in two-tier backbone ad hoc networks", Networks, ICON 2004 Proceedings, 12th IEEE International Conference on 16-19 Nov. 2004,
20. Viswanathan, H.; Mukherjee, S., "Throughput-range tradeoff of wireless mesh backhaul networks" IEEE Selected Areas in Communications Journal, Volume 24, Issue 3, March 2006
21. Zygmunt J. Haas and Marc R. Pearlman; "The Zone Routing Protocol (ZRP) for Ad Hoc Networks" *Internet Draft*,