ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΑΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ, ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

Διπλωματική Εργασία

Τίτλος : «Αδιατάρακτη ομαδοποίηση οχημάτων σε Ad Hoc δίκτυα οχημάτων»

“Stable clustering in VANETS”

Όνομα : Νάστος Απόστολος

Επιβλέπων:

Λέκτορας Κατσαρός Δημήτριος,

Καθηγητής Τασιούλας Λέανδρος

Περιεχόμενα

Ευρετήριο εικόνων

Ευρετήριο Πινάκων

Εισαγωγή

Vehicular Ad hoc Networks (VANETs)

Mobile Ad hoc Networks (MANETs)

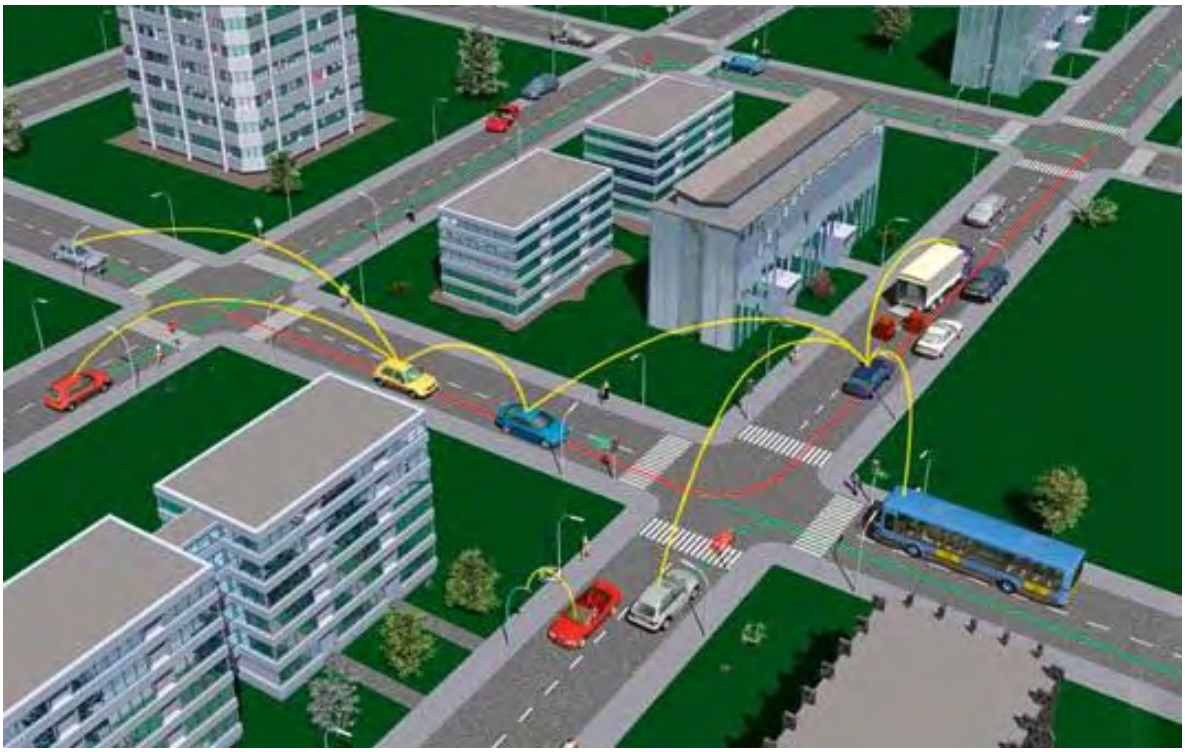
Τα MANETs είναι δίκτυα κινητών συσκευών συνδεδεμένων ασύρματα. Χαρακτηριστικό τους είναι ότι είναι αυτορυθμιζόμενα και χωρίς σταθερή υποδομή. Κάθε κόμβος σε ένα MANET είναι ελεύθερη να κινηθεί ανεξάρτητα προς οποιαδήποτε κατεύθυνση. Για αυτό αλλάζει τις συνδέσεις της (links) με τις άλλες συσκευές συχνά. Κάθε συσκευή επιβαρύνεται με την προώθηση των ροών δεδομένων που δεν την αφορούν, δηλαδή να λειτουργεί ως routers. Βασική δυσκολία στα MANETs είναι ο σχεδιασμός του δικτύου, ώστε κάθε συσκευή να διαθέτει συνεχώς τις απαραίτητες πληροφορίες ώστε να μπορεί προωθήσει κάποια δεδομένα σε κάποια άλλη συσκευή. Τέτοια δίκτυα μπορούν να λειτουργούν αυτόνομα ή ως κομμάτι ενός μεγαλύτερου δικτύου όπως του internet.

Γενικά για τα VANETs

Ένα VANET είναι μια τεχνολογία, η οποία χρησιμοποιεί κινούμενα οχήματα ως κόμβους, ώστε να δημιουργήσει ένα δίκτυο κινητών κόμβων. Το VANET μετατρέπει κάθε συμμετέχον όχημα στο δίκτυο σε ένα ασύρματο router ή κόμβο, επιτρέποντας σε οχήματα πού έχουνε απόσταση περίπου 100 με 300 μέτρα ανάμεσά τους να συνδέονται μεταξύ τους, ουσιαστικά δημιουργώντας ένα δίκτυο μεγάλης εμβέλειας. Καθώς οχήματα απομακρύνονται από την εμβέλεια των γειτόνων τους και βγαίνουνε εκτός δικτύου, άλλα οχήματα εισέρχονται εντός εμβέλειας. Συνδέοντας τα οχήματα μεταξύ τους, δημιουργείτε ένα ασύρματο δίκτυο. Στόχος η επικοινωνία των οχημάτων μελλών του δικτύου έτσι ώστε να προσφερθούν υπηρεσίες σχετικές με την οδική κυκλοφορία, τον έλεγχο την ροή κίνησης των οχημάτων και η σύνδεση στο internet.

Τα VANETs είναι υποκατηγορία των MANETs. Η διαφορά είναι ότι οι κόμβοι των VANETs είναι οχήματα. Εξαιτίας της μεγάλης κινητικότητας και

ταχύτητας με την οποία κινούνται τα οχήματα τα VANETs χρίζουν ειδικής αντιμετώπισης σε σχέση με τα άλλα ad hoc δίκτυα κινητών κόμβων και απαιτούν λύσεις συγκεκριμένα για το περιβάλλον αυτό.



Εικόνα 1-Επικοινωνία κόμβων σε ένα VANET

Χαρακτηριστικά των VANETs

Τα VANETs περιλαμβάνουν οχήματα με δυνατότητα ασύρματης επικοινωνίας. Τα οχήματα ενεργούν ως κινητοί κόμβοι ή δρομολογητές για άλλους κόμβους. Πέρα από τις ομοιότητες με τα υπόλοιπα ad hoc δίκτυα, όπως η μικρή εμβέλεια επικοινωνίας, η αυτό-οργάνωση, η αυτοδιαχείριση και το μικρό εύρος ζώνης, τα VANETs μπορούν να διαχωριστούν από τα υπόλοιπα ad hoc δίκτυα λόγω των παρακάτω χαρακτηριστικών:

Ιδιαίτερα δυναμική τοπολογία: Τα οχήματα κινούνται με υψηλές ταχύτητες. Η τοπολογία του δικτύου μεταβάλλεται συνεχώς και γρήγορα. Για παράδειγμα, ας υποθέσουμε ότι η ασύρματη εμβέλεια εκπομπής του κάθε οχήματος είναι 250 m, έτσι υπάρχει μία σύνδεση μεταξύ δύο αυτοκίνητα αν η απόσταση μεταξύ τους είναι μικρότερη από 250m. Στη χειρότερη περίπτωση,

εάν δύο αυτοκίνητα με την ταχύτητα των 25 m / sec κινούνται σε αντίθετες κατευθύνσεις, η σύνδεση μεταξύ τους θα διαρκέσει το πολύ 10 sec.

Συχνή διάσπαση του δικτύου σε μη συνδεδεμένα υποδίκτυα: εξαιτίας της ταχύτητας και της συνεχής κίνησης των κόμβων η συνδεσιμότητα των κόμβων αλλάζει συχνά, ιδιαίτερα όταν τα οχήματα είναι αραιά τοποθετημένα στο δίκτυο. Έτσι μπορεί να εμφανιστούν κόμβοι ή ομάδες κόμβων στο δίκτυο με τους οποίους δε μπορεί να επιτευχθεί επικοινωνία.

Επαρκή ενέργεια και αποθηκευτικός χώρος: Οι κόμβοι των VANETs έχουν απεριόριστη ενέργεια (δυνατότητα επικοινωνίας) και υπολογιστική ισχύ (δηλαδή αποθηκευτικό χώρο και επεξεργασία), καθώς είναι οχήματα και όχι μικρές συσκευές χειρός.

Επικοινωνία βασισμένη στη γεωγραφική θέση: σε αντίθεση με άλλα δίκτυα, τα οποία χρησιμοποιούν unicast ή multicast όπου τα άκρα της επικοινωνίας ορίζονται μέσω κάποιου αναγνωριστικού, στα VANETs εμφανίζεται ένας νέος τύπος επικοινωνίας ορίζοντας γεωγραφικές περιοχές στις οποίες πρέπει να σταλούν κάποια μηνύματα.

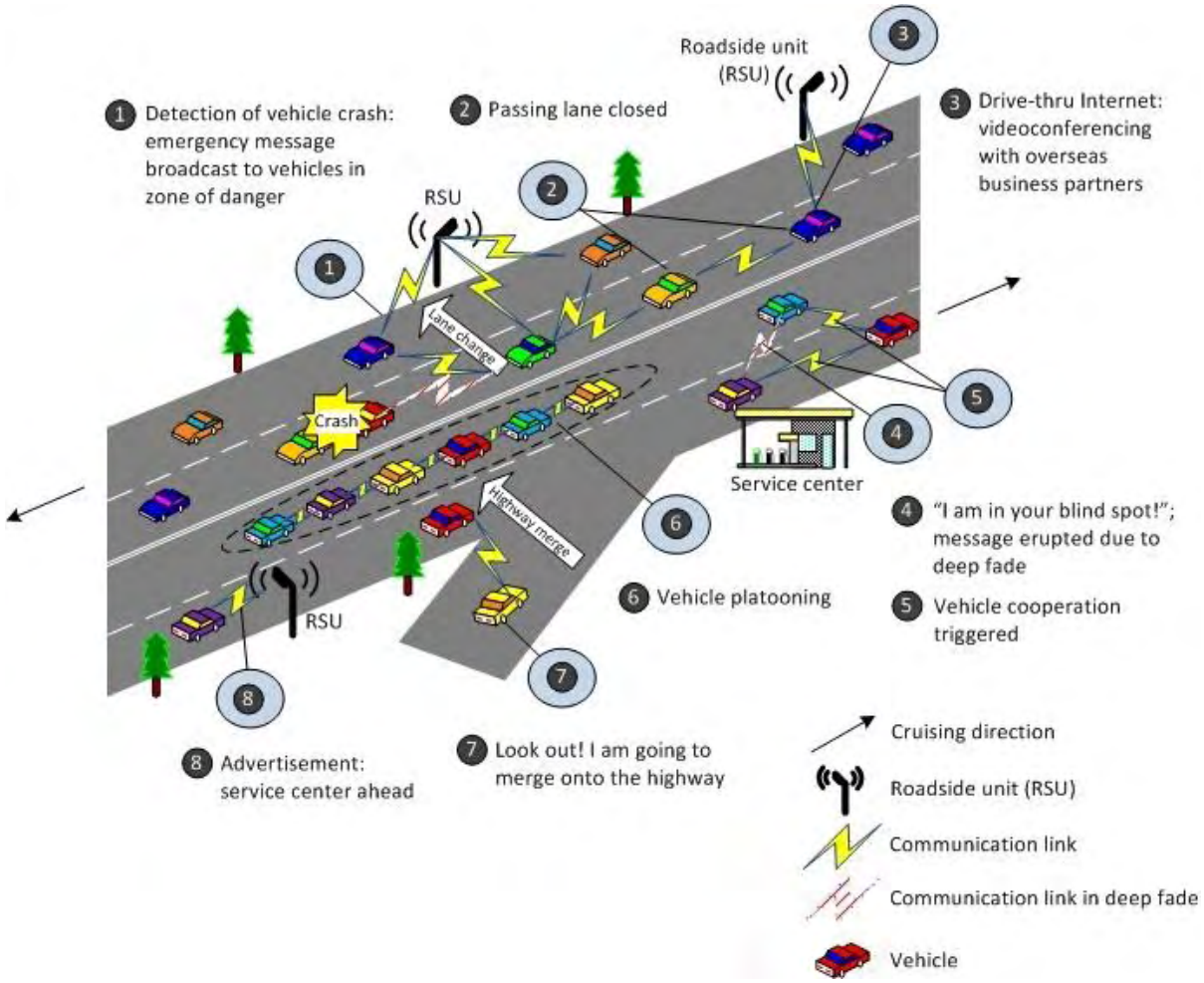
Αυστηρές απαιτήσεις στη μετάδοση δεδομένων σε πραγματικό χρόνο: σε ορισμένες εφαρμογές των VANETs, το δίκτυο δεν απαιτεί υψηλές ταχύτητες μεταφοράς δεδομένων άλλα έχει αυστηρές απαιτήσεις σε σχέση με την καθυστέρηση μιας επικοινωνίας. Ως παράδειγμα ας υποθέσουμε ένα δίκτυο κυκλοφορίας αυτόματης κίνησης. Σε περίπτωση που ένα όχημα καθυστερήσει να στείλει μήνυμα στο μπροστά του όχημα ότι πλησιάζουν μεταξύ τους τότε δε θα μπορεί να αποφευχθεί η σύγκρουση. Γίνεται κατανοητό λοιπόν ότι ορισμένες εφαρμογές δεν δίνουνε βάση στη μέση καθυστέρηση επικοινωνίας, άλλα στη μέγιστη καθυστέρηση που θα έχει αυτή η επικοινωνία.

Εφαρμογές των VANETs

Η έρευνα που γίνεται στον τομέα των VANETs οδηγεί στην εμφάνιση όλο και περισσότερων εφαρμογών για αυτόν τον τομέα. Οι περισσότερες εφαρμογές στα VANETs μπορούν να κατηγοριοποιηθούν σε δύο κατηγορίες:

Εφαρμογές ευφυής μετακίνησης: η κύρια κατηγορία εφαρμογών που συναντώνται στα VANETs. Στην κατηγορία αυτή περιλαμβάνονται εφαρμογές όπως η οδική πλοήγηση, συνεργατική παρακολούθηση κυκλοφορίας, έλεγχος της ροής της κυκλοφορίας, ανάλυση κυκλοφοριακής συμφόρησης και άμεσος εντοπισμός παρακαμπτήρια διαδρομής, λαμβάνοντας υπόψη τις κυκλοφοριακές συνθήκες και τον προορισμό. Για παράδειγμα αισθητήρες στον δρόμο καταγράφουν την πυκνότητα της κυκλοφορίας και την ταχύτητα των οχημάτων και τα στέλνουν σε έναν κεντρικό σταθμό όπου υπολογίζεται η ροή της κυκλοφορίας και το βέλτιστο πρόγραμμα για τους φωτεινούς σηματοδότες. Αυτή η διαδρομή επικοινωνίας μπορεί να μειωθεί σε μεγάλο βαθμό από δίκτυα VANETs, όπου τα οχήματα ανταλλάσουν τις οδικές συνθήκες μεταξύ τους. Σε περίπτωση τροχαίων ατυχημάτων, οι κινητοί κόμβοι μπορούν να ενημερώνουν οδικούς αισθητήρες, ώστε να ενημερώνονται διερχόμενοι οδηγοί για τον κίνδυνο ή να ενημερώνουν για το επείγων περιστατικό την αρμόδια υπηρεσία. VANETs μπορούν να χρησιμοποιηθούν σε περιπτώσεις διασταυρώσεων χωρίς ορατότητα ή στην είσοδο οχημάτων στον αυτοκινητόδρομο, ώστε να αποφεύγονται ατυχήματα.

Εφαρμογές άνεσης: αναφέρονται σε εφαρμογές που επιτρέπουν στους επιβάτες την επικοινωνία με άλλα οχήματα, είτε την σύνδεση στο internet. Για παράδειγμα τα VANETs παρέχουν σύνδεση στο internet στους κινητούς κόμβους ώστε να μπορούν να κατεβάζουνε μουσική ή οι συνοδηγοί να παίζουνε παιχνίδια. Συνήθως προστίθενται στο δίκτυο πύλες επικοινωνίας του δικτύου με το internet, ώστε να μεταδίδονται τα μηνύματα από το δίκτυο προς το internet και αντίστροφα.



Εικόνα 2-Εφαρμογές VANETs

Εικόνα 3-Ενημέρωση γειτονικών οχημάτων για πρόβλημα στο οδόστρωμα

Πρωτόκολλα δρομολόγησης επικοινωνίας στα VANETs

Λόγω της δυναμικής φύσης των κινούμενων κόμβων στο δίκτυο η εύρεση και διατήρηση καναλιών επικοινωνίας στα VANETs αποδεικνύεται μια πρόκληση. Τα πρωτόκολλα επικοινωνίας που έχουνε παρουσιαστεί μπορούν να χωριστούν στις εξής κατηγορίες:

- Ad hoc δρομολόγηση
- Position based δρομολόγηση
- Cluster based δρομολόγηση
- Broadcast δρομολόγηση
- Geocast δρομολόγηση

Δρομολόγηση ad hoc

Τα VANETs μοιράζονται την ίδια βασική αρχή με τα MANETs, δηλαδή η επικοινωνία των κόμβων δεν στηρίζεται σε σταθερή υποδομή. Διακρίνονται για την αυτό-οργάνωση αυτοδιαχείριση, το μικρό εύρος ζώνης και τη μικρή ακτίνα μετάδοσης ενός σήματος. Λόγω αυτών των ομοιοτήτων πολλά από τα ήδη υπάρχοντα πρωτόκολλα δρομολόγησης είναι εφαρμόσιμα, όπως το AODV(Ad hoc On demand Distance Vector) και το DSR(Dynamic Source Routing). Αυτά τα δύο πρωτόκολλα είναι σχεδιασμένα για γενικού σκοπού κινητά δίκτυα αλλά μπορούν να εφαρμοστούν και στην περίπτωση των VANETs.

Στο AODV ο κόμβος αποστολέας (source) στέλνει στο δίκτυο πακέτα RREQ για να βρει μια διαδρομή μέχρι τον κόμβο παραλήπτη (destination). Οι ενδιάμεσοι κόμβοι αποθηκεύουν την πληροφορία των RREQ και τα προωθούν περαιτέρω. Όταν τα πακέτα αυτά φτάσουν στον κόμβο παραλήπτη, αυτός στέλνει στον αποστολέα ένα πακέτο RREP μέσω της συντομότερης διαδρομής. Οι ενδιάμεσοι κόμβοι της επιλεγμένης διαδρομής, αποθηκεύουν το ID του κόμβου από τον οποίο δέχθηκαν το RREP και έτσι, όταν αυτό φτάσει στον αποστολέα, ακολουθείται -από κόμβο σε κόμβο- η αντίστροφη διαδρομή για την προώθηση του πακέτου.

Στο DSR προτείνεται μια παρόμοια διαδικασία, με τη διαφορά ότι η διαδρομή δεν αποθηκεύεται στους ενδιάμεσους κόμβους αλλά στα RREQ και RREP πακέτα. Έτσι το RREP πακέτο που θα φτάσει στον αποστολέα θα περιέχει την αλληλουχία των κόμβων που συνθέτουν την διαδρομή.

Εξαιτίας της πολύ μεγάλης κινητικότητας των κόμβων στα VANETs σε σύγκριση με τα MANETs αυτά τα πρωτόκολλα εμφανίζουν προβλήματα χαμηλής απόδοσης στην επικοινωνία.

Δρομολόγηση position based

Η κίνηση των κόμβων στα VANETs είναι συνήθως περιορισμένη μέσα σε δρόμους. Οπότε εμφανίστηκαν στρατηγικές δρομολόγησης που λαμβάνουνε υπόψη γεωγραφικές πληροφορίες σχετικά με την τοποθεσία των κόμβων. Πληροφορίες σχετικά με την τοποθεσία των κόμβων μπορούν να εξαχθούν μέσω οδικών χαρτών, μοντέλα κυκλοφορίας ή ενσωματωμένων συστημάτων πλοήγησης στα οχήματα.

Ένας αλγόριθμος δρομολόγησης position based είναι ο GPSR (Greedy Perimeter Stateless Routing) χρησιμοποιεί πληροφορίες σχετικές με την τοποθεσία των κόμβων και αποστέλλει ένα μήνυμα πάντα στον κόμβο που είναι γεωγραφικά πιο κοντά στον προορισμό του μηνύματος. Πολλά προβλήματα εμφανίζονται όμως και σε αυτό τον τρόπο δρομολόγησης καθώς η δημιουργία ενός καναλιού επικοινωνίας από έναν κόμβο σε ένα άλλο μπορεί να μην είναι εφικτή. Σε ένα VANET μέσα στην πόλη για παράδειγμα η επικοινωνία μεταξύ οχημάτων εμποδίζεται από τα κτήρια ή άλλα εμπόδια που βρίσκονται ανάμεσά στους κόμβους. Σε ένα μεγάλο αυτοκινητόδρομο το πρόβλημα αυτό δεν θα ήταν τόσο εμφανές.

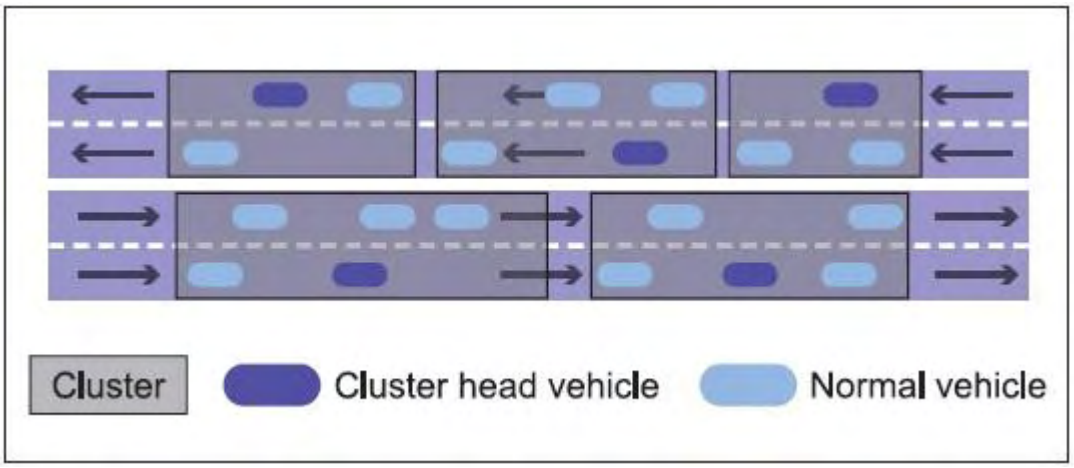
Δρομολόγηση cluster based

Στη δρομολόγηση cluster based σχηματίζεται μία εικονική νοητή δομή του δικτύου μέσω της ομαδοποίησης των κόμβων ώστε να επιτευχτεί κλιμάκωση στην δρομολόγηση. Οι ομάδες αυτές ονομάζονται clusters. Κάθε cluster κόμβων έχει ένα cluster head κόμβο, ο οποίος είναι υπεύθυνος για τον συντονισμό της επικοινωνία κόμβων εντός και εκτός του cluster του. Η επικοινωνία με άλλα clusters γίνεται μέσω των cluster heads των αντίστοιχων

clusters. Η όσο το δυνατόν πιο σταθερή ομαδοποίηση των κόμβων είναι ο βασικός στόχος των αλγορίθμων που αναλαμβάνουν τον σχηματισμό των clusters.

Πολλοί αλγόριθμοι clustering έχουνε εισαχθεί από τα MANETs. Λόγω των έντονων διαφορών μεταξύ των δύο τεχνολογιών όμως οι τεχνικές clustering είναι πολύ ασταθείς στα δίκτυα οχημάτων. Τα clusters που σχηματίζονται έχουνε μικρή διάρκεια ζωής και για να επιτευχθεί κλιμάκωση απαιτητέ μεγάλο κόστος επικοινωνίας. Η δρομολόγηση με clusters μπορεί να πετύχει καλή κλιμάκωση σε μεγάλα δίκτυα, όμως η γρήγορη αλλαγή της μορφής ενός VANET, αναγκάζει την συχνή αναδιάρθρωση των clusters.

Στο επόμενο κεφάλαιο θα αναφερθούν αλγόριθμοι clustering αναλυτικά.



Εικόνα 4-Δομή δικτύου χωρισμένο σε clusters

Δρομολόγηση Broadcast

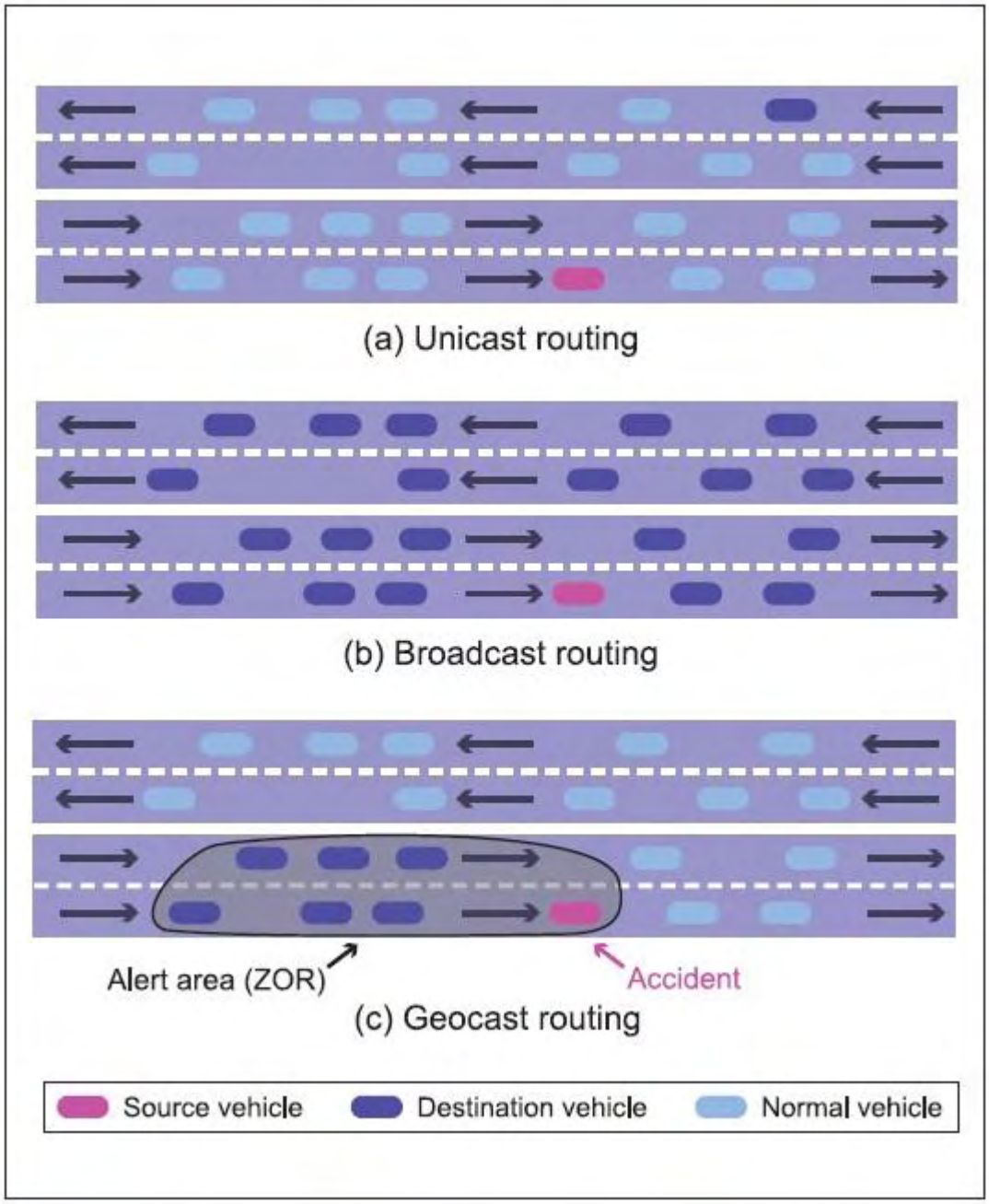
Η τεχνική της δρομολόγησης με broadcast είναι μια συχνά χρησιμοποιούμενη τεχνική στα VANETs. Χρησιμοποιείται για την ενημέρωση των οχημάτων σχετικά με τον καιρό, κυκλοφορία, συνθήκες έκτακτης ανάγκης, κατάσταση δρόμου, και μετάδοση διαφημίσεων και ανακοινώσεων. Τεχνικές broadcast επικοινωνίας χρησιμοποιούν και πρωτόκολλα unicast δρομολόγησης κατά την διαδικασία ανίχνευσης των γειτονικών κόμβων στην περιοχή γύρω τους.

Ο πιο απλός τρόπος υλοποίησης μιας υπηρεσίας broadcast επικοινωνίας είναι μέσω της τεχνικής του flooding, όπου κάθε κόμβος που λαμβάνει ένα μήνυμα το αναμεταδίδει σε όλους τους γείτονές του, εκτός αυτού από τον οποίο έλαβε το μήνυμα. Σε ένα μικρό δίκτυο η τεχνική του flooding έχει καλή απόδοση. Όσο το δίκτυο αυξάνει σε μέγεθος η απόδοση της τεχνικής αυτής πέφτει δραστικά, καθώς το εύρος ζώνης που απαιτείται για την αποστολή ενός broadcast μηνύματος μπορεί να αυξάνεται εκθετικά. Χρησιμοποιώντας επιλεκτική προώθηση ενός μηνύματος η συμφόρηση στο δίκτυο μπορεί να μειωθεί αρκετά.

Το BROADCOMM, είναι ένα broadcast πρωτόκολλο έκτακτης ανάγκης βασισμένο σε μια ιεραρχική δομή των κόμβων του. Σχεδιάστηκε κυρίως για δίκτυα αυτοκινητοδρόμων. Συγκεκριμένα το δίκτυο του αυτοκινητοδρόμου χωρίζεται σε εικονικά κελιά, τα οποία κινούνται όπως κινούνται και τα οχήματα. Σε κάθε κελί υπάρχει ένας αντιπρόσωπος των οχημάτων που βρίσκονται στο κελί αντίστοιχα όπως τα cluster heads στα clusters. Ο αντιπρόσωπος αναλαμβάνει την διαχείριση των πακέτων έκτακτης ανάγκης που έρχονται από οχήματα του κελιού του ή γειτονικών κελιών. Επιπλέον ο αντιπρόσωπος λειτουργεί ως ενδιάμεσος στην δρομολόγηση μηνυμάτων από γειτονικούς αντιπροσώπους κελιών.

Δρομολόγηση Geocast

Η δρομολόγηση geocast based είναι ουσιαστικά μια τεχνική multicast επικοινωνίας βασιζόμενη στην θέση των κόμβων. Στόχος της είναι η παράδοση ενός μηνύματος σε μια ομάδα κόμβων που βρίσκονται σε μια συγκεκριμένη γεωγραφική περιοχή. Πολλές εφαρμογές μπορούνε να επωφεληθούν από τη συγκεκριμένη δρομολόγηση καθώς μπορεί να στέλνονται μηνύματα μόνο σε ομάδες οχημάτων που βρίσκονται σε μια ζώνη ενδιαφέροντος και να αποφεύγεται η μετάδοση μηνυμάτων σε οχήματα τα οποία δεν έχουνε κάποιο κέρδος από τα μηνύματα αυτά. Η Geocast δρομολόγηση μπορεί να επιτευχθεί μέσω multicast απλώς ορίζοντας την multicast ομάδα να είναι τα οχήματα σε μια συγκεκριμένη περιοχή.



Εικόνα 5-Διαφορετικά σενάρια επικοινωνίας σε VANETs

Αρχιτεκτονικές VANETs

Οι συνιστώσες των VANET συστημάτων μπορούν να χωριστούν σε τέσσερεις κατηγορίες. Αυτές είναι τα οχήματα, οι προσωπικές συσκευές, ο εξοπλισμός δρόμου(RSU - Road-Side Units) και ο κεντρικός εξοπλισμός. Ένα σύστημα VANET δεν είναι απαραίτητο να αποτελείται από συστατικά όλων

των παραπάνω κατηγοριών. Με βάση το ποιες κατηγορίες απαρτίζουν το σύστημα γίνεται και ο διαχωρισμός των αρχιτεκτονικών που εμφανίζονται.

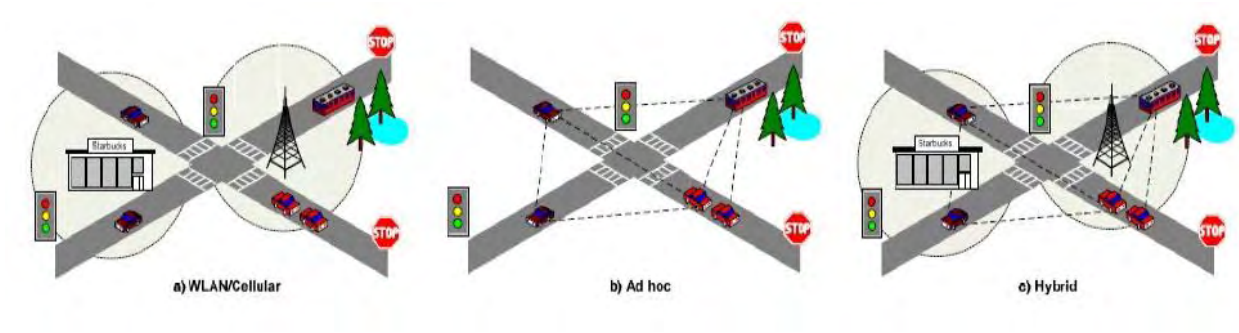
Τα οχήματα είναι αναπόσπαστο κομμάτι των VANETs και αποτελούν τη βασική συνιστώσα όλων των αρχιτεκτονικών. Οι προσωπικές συσκευές τοποθετούνται από το χρήστη πάνω στο όχημα και μπορεί να είναι είτε συσκευές πλοήγησης GPS είτε συσκευές πολυμεσικών εφαρμογών. Επίσης κινητά τηλέφωνα μπορεί να θεωρηθούν ότι ανήκουν σε αυτή την κατηγορία αν χρησιμοποιούν το δίκτυο των οχημάτων, για τη σύνδεση τους στο διαδίκτυο, και όχι κάποιο άλλο δίκτυο.

Ο εξοπλισμός δρόμου παρουσιάζει μεγάλη ποικιλία ως προς το είδος των συσκευών που μπορεί να αποτελείται. Παραδοσιακές συσκευές όπως φωτεινοί σηματοδότες και πινακίδες σήμανσης δύναται να ενσωματώνουν τμήματα με υπολογιστική ισχύ και ικανότητα ασύρματης επικοινωνίας(πομπούς), έτσι ώστε να διανέμουν την πληροφορία τους στο δίκτυο πέρα από την οπτική επαφή. Επίσης σταθεροί αναμεταδότες που λειτουργούν ως ακίνητοι κόμβοι του δικτύου, και βοηθούν στην πιο γρήγορη και αποτελεσματική μετάδοση των μηνυμάτων. Τα RSU μπορεί να είναι είτε αποκομμένα είτε να συνδέονται μεταξύ τους με ένα ιδιωτικό δίκτυο. Επίσης μπορούν να συνδέονται με τον κεντρικό εξοπλισμό ή και κατευθείαν στο Διαδίκτυο.

Ο κεντρικός εξοπλισμός λειτουργεί ως το κέντρο ελέγχου του δικτύου. Κύριες λειτουργίες του είναι να συλλέγει δεδομένα από τα RSU, να προβλέπει μελλοντικές συνθήκες του δικτύου(π.χ. κυκλοφοριακή συμφόρηση), να δίνει εντολές στα RSU.

Οι τρείς αρχιτεκτονικές που εμφανίζονται στα VANETs και παρουσιάζονται στην Εικόνα είναι οι εξής:

- Αμιγώς κυψελωτή / WLAN αρχιτεκτονική
- Αμιγώς αδόμητη (Ad-hoc) αρχιτεκτονική
- Υβριδική αρχιτεκτονική



Εικόνα 6-Αρχιτεκτονικές VANETs

Αμιγώς κυψελωτή / WLAN αρχιτεκτονική : Το δίκτυο χρησιμοποιεί πύλες δικτύου(gateways) ή access points, που βρίσκονται συνήθως στις διασταυρώσεις, για να συνδέσει τους κόμβους-οχήματα με το Διαδίκτυο, για να συλλέξει δεδομένα από αυτούς, για να τους μεταδώσει σημαντικές πληροφορίες ή για να κάνει τη δρομολόγηση.

Ο σταθερός εξοπλισμός παρέχει στους κινούμενους κόμβους συνδεσιμότητα και καλύτερη ποιότητα επικοινωνίας. Όμως το μειονέκτημα αυτής της αρχιτεκτονικής είναι το υψηλό κόστος εγκατάστασης και λειτουργίας του σταθερού εξοπλισμού, που την κάνει γενικά να φαίνεται ασύμφορη σε σχέση με τις υπόλοιπες.

Αμιγώς αδόμητη (Ad-hoc) αρχιτεκτονική : Σε αυτή την αρχιτεκτονική υφίστανται μόνο κινητοί κόμβοι(οχήματα) και συσκευές RSU. Η επικοινωνία γίνεται μέσω γειτονικών κόμβων ή για πιο μακρινές αποστάσεις μέσω πολλαπλών συνδέσεων-αλμάτων(hops). Το δίκτυο οργανώνεται μόνο του και δεν απαιτείται κάποια σταθερή υποδομή. Το μειονέκτημα είναι ότι παρουσιάζει μικρότερη αξιοπιστία λόγω της κίνησης των κόμβων και τη μη συνεχή παρουσία τους σε όλα τα σημεία.

Υβριδική αρχιτεκτονική : Είναι ένας συνδυασμός των προηγούμενων αρχιτεκτονικών. Το δίκτυο υλοποιείται μέσω συνδέσεων οχημάτων-σταθρού εξοπλισμού(V2I – Vehicle to Infrastructure) και μέσω συνδέσεων οχημάτων-οχημάτων(V2V-Vehicle to Vehicle). Ο κάθε τύπος σύνδεσης χρησιμοποιείται ανάλογα με τις συνθήκες που επικρατούν στο δίκτυο και τις ανάγκες της επιθυμητής επικοινωνίας.

Αλγόριθμοι Clustering

Γενικά

Clustering είναι η διαδικασία όπου οι κόμβοι του δικτύου μοιράζονται σε ομάδες, έτσι ώστε να δημιουργηθεί μια ιεραρχία ανάμεσά τους. Σε κινητά δίκτυα, που αποτελούνται από πολλούς κόμβους οι συνεχώς μεταβαλλόμενες τοπολογίες οδηγούν σε συχνές κατατμήσεις του δικτύου. Έτσι εμφανίζεται το πρόβλημα της κλιμάκωσης και του ελέγχου του δικτύου, ειδικότερα στην διαδικασία σχεδιασμού πρωτοκόλλων υψηλότερου επιπέδου, όπως της δρομολόγησης ή στην υλοποίηση υπηρεσιών Quality of Service. Η δημιουργία μιας ιεραρχικής δρομολόγησης είναι μια τεχνική να αντιμετωπιστούν σε ένα βαθμό τα προβλήματα αυτά. Αυτό πετυχαίνεται με την χρήση αλγορίθμων clustering από τους κόμβους του δικτύου.

Στη συνέχεια θα γίνει μια αναφορά σε ορισμένους αλγόριθμους clustering. Μεγαλύτερη αναφορά θα γίνει στην διαδικασία σχηματισμού clusters για τους αλγόριθμους CBLR (Cluster Based Location Routing), CBMAC (Cluster Based Medium Access Control) και DMAC (Distributed Mobility Adaptive Clustering).

Clustering μικρότερου αναγνωριστικού (Lowest ID)

Γενικά

Ο Lowest ID είναι ο πιο απλός αλγόριθμος clustering. Κάθε κόμβος του δικτύου διαθέτει ένα μοναδικό αναγνωριστικό (ID). Κάθε κόμβος περιοδικά αποστέλλει μηνύματα “Hello” τα οποία περιλαμβάνουν το αναγνωριστικό του. Κάθε κόμβος συγκρίνει το αναγνωριστικό του με αυτά των γειτόνων του, και ο κόμβος με το μικρότερο αναγνωριστικό αποφασίζει να λειτουργήσει ως cluster head. Ένας κόμβος που επικοινωνεί με δύο cluster heads αναλαμβάνει το ρόλο του gateway.

Σε αυτόν τον αλγόριθμου δεν υπάρχει κάποιος περιορισμός σχετικά με το πόσους κόμβους μέλη μπορεί να περιλαμβάνει ένα cluster. Δεν λαμβάνονται υπόψη παράμετροι και χαρακτηριστικά του δικτύου για την

επιλογή των κόμβων cluster head και για αυτό οι επιδόσεις του σε ένα κινητό δίκτυο είναι απρόβλεπτες και τυχαίες.

Ψευδοκώδικας

Ορίζονται δύο βασικές καταστάσεις για την λειτουργία των κόμβων, απλός κόμβος (MN) και cluster-head (CH). Οι κόμβοι συμβολίζονται ως v, u, z, x. Και ως Γ(v) ορίζεται το σύνολο των γειτονικών κόμβων του v. Περιοδικά μηνύματα στέλνονται από τους κόμβους από την συνάρτηση Send_Periodic_Msg(period), με στόχο την ανακάλυψη νέων κόμβων καθώς και αφαίρεση αυτών. Οι κόμβοι CH στέλνουν μηνύματα CH(v) και οι υπόλοιποι Join(v,u). Σε περίπτωση που οι κόμβοι δεν ανήκουν σε κάποιο Cluster στέλνουν Join(v,NULL). Οι κόμβοι ανακαλύπτουν την απόλυα μιας σύνδεσης με κάποιον άλλο κόμβο σε περίπτωση που δεν λάβουν μήνυμα από αυτόν για κάποιο χρονικό διάστημα. Κάθε κόμβος στα μηνύματα που στέλνει περιλαμβάνει το ID του, ώστε να είναι εύκολη η σύγκριση από τον λαμβανόμενο κόμβο. Η λειτουργία του lowest-id περιγράφεται από τον παρακάτω ψευδοκώδικα.

Lowest-id Procedure

MN state

Init:

```
Send_Periodic_Msg(period);
Set_CH_Timer(CH_timeout);
```

On Receive CH(u):

```
CH(u) = true;
If ( for every z in Γ(v), so that ID_z < ID_u AND Join(z, x) == true )
{
        Clusterhead = u;
        Send Join(v, Clusterhead);
}
```

On receive Join(u, t):

```
Join(u,t) = true;
If( CH(v) )
{
        If( CH(v) == t )UpdateClusterTable();
}
```

On timeout timer CH:

```
If ( for every u in Γ(v) with ID_v > ID_u AND Join(u,x) == true)
{
        Change_State( CH )
        UpdateClusterTable();
        CH(v) = true;
        Clusterhead = v;
        Send CH(v);
} else { Set_CH_Timer(CH_timeout); }
```

CH state

Init :

```
        Send_Periodic_Msg(period);
```

On Receive CH(u):

```
If (  ID_u < ID_v )
{
        ChangeState( MN );
        Clusterhead = u;
        Send Join(v, Clusterhead);
}
```

On receive Join(u, t):

```
Join(u,t) = true;
If( CH(v) )
{
        If(Clusterhead == t ) { UpdateClusterTable(); }
}
```

Clustering Highest Degree

Γενικά

Ο αλγόριθμος αυτός προσπαθεί να υπολογίσει ένα βαθμό για κάθε κόμβο, για παράδειγμα πόσους γείτονες σε απόσταση μιας μετάδοσης μηνύματος έχει ένας κόμβος. Κάθε κόμβος περιοδικά μεταδίδει στους γείτονές του την τιμή που έχει υπολογίσει. Ο κόμβος με τον μεγαλύτερο βαθμό επιλέγεται ως cluster head και οι γείτονές του γίνονται τα μέλη του cluster. Η διαδικασία επαναλαμβάνεται από τους εναπομείναντες κόμβους έως ότου κάθε κόμβος να ανατεθεί σε ένα cluster. Οι ισοπαλίες μπορούν να λυθούν με το κριτήριο του Lowest ID. Αυτή η μέθοδος δεν θέτει κάποιο πάνω όρια στο μέγεθος στον αριθμό των κόμβων σε ένα cluster, συνεπώς τα cluster heads υπερφορτώνονται και υπάρχει μείωση της απόδοσης. Επιπλέον λόγω της συχνής αλλαγής της τοπολογίας του δικτύου, αν ένα cluster head χάσει έστω και μια σύνδεση με τους γείτονές του είναι πιθανό να αποτύχει να εκλεγεί ξανά ως cluster head. Αυτό έχει ως αποτέλεσμα συνεχείς αναδομήσεις στα clusters.

Ψευδοκώδικας

Λαμβάνονται οι ίδιες υποθέσεις όπως αναφέρονται παραπάνω στον lowest-id. Επιπλέον οι κόμβοι κάθε φορά που λαμβάνουν ένα μήνυμα είναι υποχρεωμένοι να υπολογίσουνε το degree τους, καθώς είναι πιθανό το μήνυμα να στάλθηκε από κόμβο, ο οποίος δεν βρισκόταν πριν σε εμβέλεια. Κάθε κόμβος όταν στέλνει ένα μήνυμα πρέπει να περιλάβει το βάρος του(degree) σε αυτό. Η λειτουργία του αλγορίθμου περιγράφεται από τον παρακάτω ψευδοκώδικα

Highest Degree Procedure

MN state

Init:

```
Send_Periodic_Msg(period);
Set_CH_Timer(CH_timeout);
```

On Receive CH(u):

UpdateDegree(u);
CH(u) = true;
If(for every z in Γ(v), so that $Degree_z > Degree_u$ AND Join(z, x) == true)
{
    Clusterhead = u;
    Send Join(v, Clusterhead);
}

On receive Join(u, t):

UpdateDegree(u);
Join(u,t) = true;
If(CH(v))
{
    If(CH(v) == t) { UpdateClusterTable(); }
}

On timeout timer CH:

If (for every u in Γ(v) with $Degree_v < Degree_u$ AND Join(u,x) == true)
{
    Change_State(CH)
    UpdateClusterTable();
    CH(v) = true;
    Clusterhead = v;
    Send CH(v);
} else { Set_CH_Timer(CH_timeout); }

CH state

Init :

Send_Periodic_Msg(period);

On Receive CH(u):

UpdateDegree(u);
CH(u) = true;
If ($Degree_u > Degree_v$)
{
    ChangeState(MN);
    Clusterhead = u;

```
        Send Join(v, Clusterhead);
}

On receive Join(u, t):
UpdateDegree(u);
Join(u,t) = true;
If( CH(v) )
{
        If(Clusterhead == t ) { UpdateClusterTable(); }
}
```

MOBIC

Γενικά

Ο αλγόριθμος είναι παρόμοιος στην λειτουργία του με τον Lowest ID εκτός του ότι χρησιμοποιείτε ένα μέτρο κινητικότητας των κόμβων ως βάση για τον σχηματισμό των clusters και όχι το ID. Ο αλγόριθμος χρησιμοποιεί ως μέτρο κινητικότητας την ALM (Aggregate Local Mobility) για να εκλέξει το cluster head. Το ALM υπολογίζεται ως κλάσμα της λαμβανόμενης ισχύς διαδοχικών μεταδόσεων περιοδικών μηνυμάτων “Hello” (Εξίσωση 1). Αυτό δίνει μια αίσθηση της σχετικής κίνησης των οχημάτων ως προς τους γείτονές τους.

$$M_Y^{rel}(X) = 10\log_{10}\frac{RxPr_{X\to Y}^{new}}{RxPr_{X\to Y}^{old}}$$

Εξίσωση 1

Εάν η ισχύς του νέου μηνύματος είναι μικρότερη από την ισχύ του προηγούμενου τότε τα οχήματα απομακρύνονται το ένα από το άλλο και η σχετική κινητικότητα τους είναι αρνητική. Αντίστοιχα στην αντίθετη περίπτωση όπου η ισχύς ενός μηνύματος είναι μεγαλύτερη από την ισχύ του

προηγούμενου τα οχήματα πλησιάζουν και η σχετική κινητικότητα τους είναι θετική.

Για τον υπολογισμό του ALM κάθε κόμβος διαθέτει ένα σύνολο από τιμές σχετικής κινητικότητας ως προς όλους τους γείτονές του. Έτσι υπολογίζοντας την διακύμανση, ως προς το μηδέν, στις τιμές αυτές υπολογίζεται το ALM (Εξίσωση 2). Ο κόμβος με το μικρότερο ALM θα γίνει τελικά cluster head.

$$\begin{aligned} M_Y &= var_0(M_Y^{rel}(X_1), M_Y^{rel}(X_2), \dots M_Y^{rel}(X_m)) \\ &= E[(M_Y^{rel})^2] \end{aligned}$$

Εξίσωση 2

Κύριο μειονέκτημα του αλγορίθμου αυτού είναι η χρήση της ισχύος του λαμβανόμενου μηνύματος ως μέτρο για την κινητικότητα του κόμβου. Όμως, εξαιτίας του θορύβου, διαφορών στην ισχύ των μπαταριών και εμποδίων το βάρος υπολογισμένο με βάση την διακύμανση στην ισχύ του λαμβανόμενου σήματος μπορεί να μην είναι ακριβής και η σταθερότητα των κόμβων ως cluster heads να μην είναι απόλυτα εξακριβωμένη. Μια βελτίωση στον υπολογισμό του ALM θα παρουσιαστεί σε επόμενη ενότητα.

Ψευδοκώδικας

Λαμβάνονται οι ίδιες υποθέσεις όπως αναφέρονται παραπάνω στον lowest-id. Επιπλέον κάθε φορά που ένα μήνυμα λαμβάνετε από ένα κόμβο v ο κόμβος θα πρέπει να υπολογίζει το νέο βάρος του(Aggregate Local Mobility). Κάθε κόμβος περιλαμβάνει στα μηνύματα που στέλνει και το βάρος του, για να μπορεί να γίνει η σύγκριση από τον κόμβο που θα λάβει το μήνυμα. Η λειτουργία του αλγορίθμου περιγράφεται από τον παρακάτω ψευδοκώδικα.

MOBIC procedure

MN state

Init:

```
Send_Periodic_Msg(period);
Set_CH_Timer(CH_timeout);
```

On Receive CH(u):

```
UpdateWeight();
CH(u) = true;
If(for every z in Γ(v), so that Weight_z < Weight_u AND Join(z, x) == true)
{
        Clusterhead = u;
        Send Join(v, Clusterhead);
}
```

On receive Join(u, t):

```
UpdateWeight();
Join(u,t) = true;
If( CH(v) )
{
        If( CH(v) == t ) { UpdateClusterTable(); }
}
```

On timeout timer CH:

```
If ( for every u in Γ(v) with Weight_v > Weight_u AND Join(u,x) == true)
{
        Change_State( CH )
        UpdateClusterTable();
        CH(v) = true;
        Clusterhead = v;
        Send CH(v);
} else { Set_CH_Timer(CH_timeout); }
```

CH state

Init :

```
Send_Periodic_Msg(period);
```

```
On Receive CH(u):
UpdateWeight ();
CH (u) = true;
If (  Weight_u < Weight_v )
{
        ChangeState( MN );
        Clusterhead = u;
        Send Join(v, Clusterhead);
}

On receive Join(u, t):
UpdateWeight();
Join(u,t) = true;
If( CH(v) )
{
        If(Clusterhead == t ) { UpdateClusterTable(); }
}
```

Ο Cluster Based Routing Αλγόριθμος (CBLR)

Κύρια χαρακτηριστικά του CBLR είναι ότι κάθε cluster ορίζεται από ένα cluster head και τουλάχιστον από έναν κόμβο μέλους. Κάθε cluster head διατηρεί ένα “Cluster Table”, όπου κρατά τις διευθύνσεις και την γεωγραφική θέση των κόμβων μελών. Κάθε κόμβος μπορεί να ανακαλύψει την θέση του μέσω μιας συσκευής GPS. Ο κόμβος cluster head διατηρεί επίσης έναν πίνακα με τα γειτονικά του clusters (διευθύνσεις και γεωγραφική θέση). Σε αυτόν τον πίνακα κρατούνται οι διευθύνσεις των κόμβων μέσω των οποίων μπορεί να πετύχει επικοινωνία με τα γειτονικά clusters.

Ο αλγόριθμος αποτελείται από τέσσερα στάδια:

1. Σχηματισμός των clusters.

2. Εντοπισμός τοποθεσίας (μηνύματα LREQ και LREP).
3. Δρομολόγηση πακέτων δεδομένων.
4. Διατήρηση των πληροφοριών τοποθεσίας.

Σχηματισμός των clusters

Αρχικά ο αλγόριθμος σχηματίζει τα clusters των κόμβων του δικτύου. Κάθε cluster περιλαμβάνει ένα cluster head και μηδέν ή περισσότερους κόμβους μέλη. Κάθε κόμβος στην εκκίνηση του ξεκινά στην κατάσταση Undecided. Αρχικοποιεί ένα χρονόμετρο και αρχίζει να κάνει broadcast ένα “Hello” μήνυμα. Εάν ο κόμβος στην κατάσταση Undecided λάβει μήνυμα από ένα cluster head πριν λήξει το χρονόμετρο του, γίνεται κόμβος μέλος στο συγκεκριμένο cluster head. Αν το χρονόμετρο λήξει και δεν έχει λάβει κάποιο μήνυμα “Hello” από κάποιο cluster head γίνεται ο ίδιος cluster head.

Το cluster head είναι ο κόμβος, που είναι υπεύθυνος για τα μέλη του cluster και πρέπει να στέλνει περιοδικά μηνύματα “Hello”, όταν ένας κόμβος μέλος λάβει ένα τέτοιο μήνυμα δηλώνει το cluster head και του απαντά με ένα αντίστοιχο μήνυμα. Το cluster head έτσι ανανεώνει το “cluster table” του με την θέση και διεύθυνση κάθε μέλους του cluster.

Όταν ένας κόμβος μέλους λάβει μήνυμα από ένα cluster head και είναι διαφορετικό από το cluster head στο οποίο έχει εγγραφεί, στέλνει τις πληροφορίες στο cluster head του, ώστε να ενημερώσει τον πίνακα των γειτονικών clusters και να επικοινωνεί μέσω του συγκεκριμένου μέλους με αυτό.

Όταν ένας κόμβος λειτουργεί ως cluster head και λάβει ένα “Hello” μήνυμα από ένα άλλο cluster head, τότε πρέπει να συγκρίνει μια τιμή βάρους που κρατάει για να αποφασίσει αν θα συνεχίσει ως cluster head ή θα γίνει μέλος. Το βάρος που χρησιμοποιείται εδώ είναι το σύνολο των κόμβων γειτόνων, πού έχει το κάθε cluster. Προτεραιότητα δύνεται στο μεγαλύτερο cluster και το μικρότερο σπάει.

Εντοπισμός θέσης

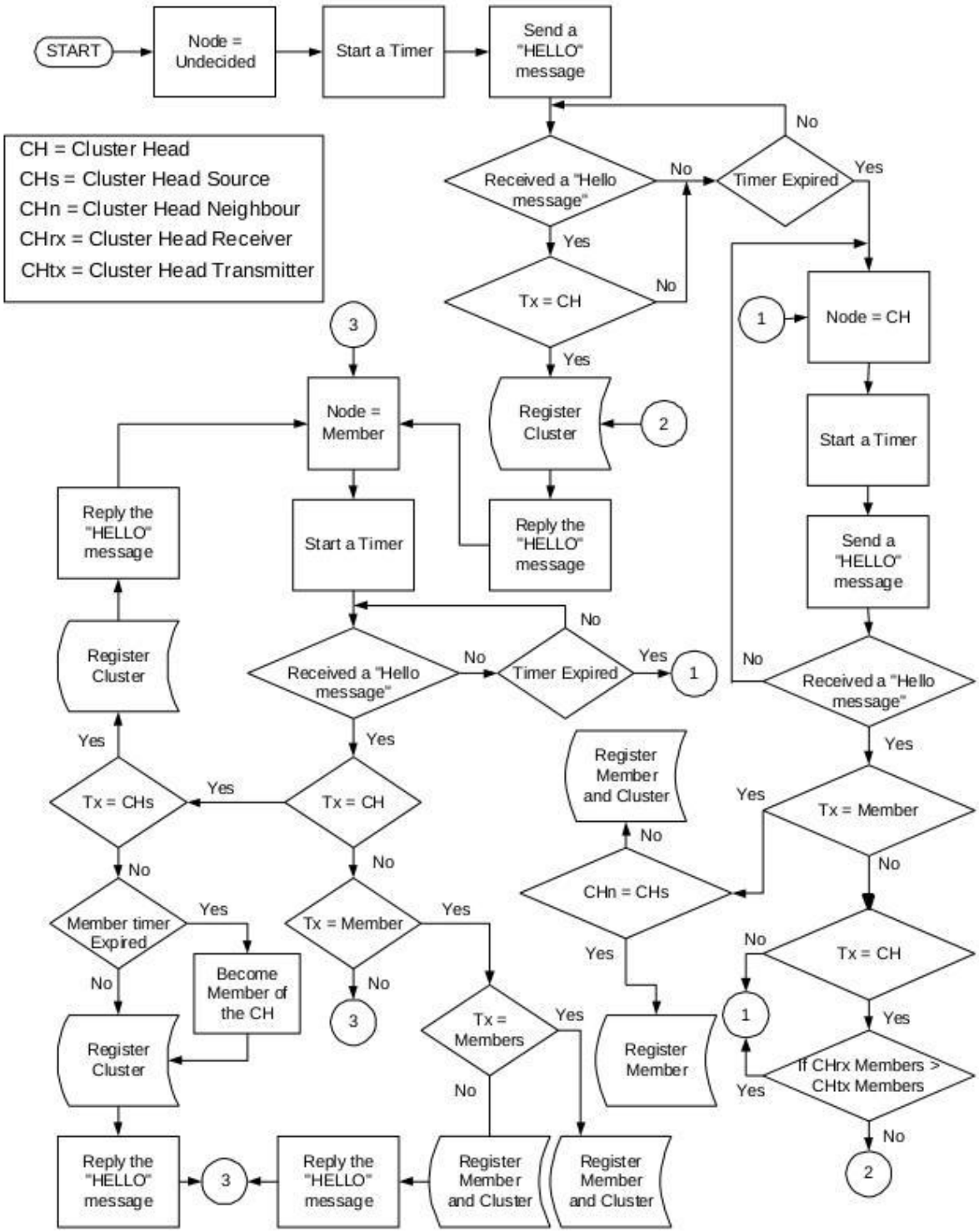
Όταν ένας αποστολέας θέλει να στείλει δεδομένα σε έναν κόμβο, πρώτα ελέγχει τον πίνακα των γειτονικών clusters που διαθέτει, για να ελέγξει εάν γνωρίζει την τοποθεσία του προορισμού. Εάν την γνωρίζει στέλνει τα δεδομένα στο κοντινότερο cluster head. Εάν δεν την γνωρίζει αποθηκεύει τα δεδομένα, αρχικοποιεί ένα χρονόμετρο και κάνει broadcast πακέτα αίτησης θέσης (Location Request LREQ). Όταν ένα cluster head λάβει ένα τέτοιο πακέτο ελέγχει αν ο κόμβος που αναζητείται είναι μέλος του cluster του. Σε περίπτωση επιτυχίας στέλνεται απάντηση θέσης(Location Reply LREP) στον αποστολέα. Η αποτυχία οδηγεί σε αναμεταδόσεις της αίτησης από το cluster head στα γειτονικά του clusters.

Μόλις ο αποστολέας λάβει την θέση του παραλήπτη στέλνει το πακέτο δεδομένων στο κοντινότερο cluster head, το οποίο αναλαμβάνει την εργασία της αποστολής του στον κατάλληλο παραλήπτη μέσω των πληροφοριών που πλέον γνωρίζει.

Δρομολόγηση πακέτων δεδομένων

Η δρομολόγηση των πακέτων δεδομένων βασίζεται στην τοποθεσία του αποστολέα, του παραλήπτη, των cluster heads τους και των κόμβων μελών.

Η δρομολόγηση των πακέτων δεν επιβαρύνει τον αποστολέα, και τα πακέτα μεταδίδονται βάση της τοποθεσίας. Διατηρώντας τις τοποθεσίες των κόμβων στα cluster heads η μετάδοση γίνεται προς μια περιοχή, όπου βρίσκονται οι κόμβοι, έτσι το μονοπάτι που θα βρεθεί θα είναι συντομότερο από αυτό που θα έβρισκαν άλλες μέθοδοι δρομολόγησης. Σε μια τυπική δρομολόγηση πακέτων, το μονόπατη μετάδοσης ενός πακέτου δεν θα είναι τόσο καλό όσο μια δρομολόγηση με βάση τα clusters, τα οποία λαμβάνουνε υπόψη την τοποθεσία των κόμβων, καθώς τα μέλη ενός cluster βρίσκονται στην ίδια γεωγραφική γειτονιά.



Εικόνα 7-Λειτουργία CBLR

Διατήρηση των πληροφοριών τοποθεσίας

Ο αλγόριθμος CBLR είναι κατάλληλος για δίκτυα με γρήγορους κινούμενους κόμβους, διότι διατηρεί τις πληροφορίες τοποθεσίας αποστολέα και παραλήπτη κατά την διάρκεια μιας μετάδοσης δεδομένων. Ο αποστολέας ενημερώνει τις πληροφορίες για την θέση του πριν στείλει ένα πακέτο. Όταν ένας κόμβος παραλάβει ένα πακέτο ενημερώνει επίσης τις πληροφορίες θέσης του και απαντά με ένα μήνυμα acknowledgment στον αποστολέα.

Ψευδοκώδικας

Λαμβάνονται οι ίδιες υποθέσεις όπως αναφέρονται παραπάνω στον lowest-id. Επιπλέον κάθε κόμβος συνδέεται με ένα βάρος. Το βάρος αυτό είναι το πλήθος των γειτονικών κόμβων, όπως και στον highest-degree. Κάθε φορά που ένας κόμβος στέλνει ένα μήνυμα πρέπει να περιλαμβάνει και το βάρος του σε αυτό, για να μπορεί να γίνει η σύγκριση από τον κόμβο που θα λάβει το μήνυμα. Η λειτουργία του αλγορίθμου περιγράφεται από τον παρακάτω ψευδοκώδικα.

CBLR procedure

Undecided state

Init:
Clusterhead = NULL;
send_Periodic_Hello_Msg (period);
set_CH_Timer (timeout_CH);

On receive HELLO_CH(u):
Clusterhead = u;
change_State (MN);

On timeout timer_CH:
change_State (CH);

MN state

Init:

send_Periodic_Hello_Msg (period);
set_CH_Timer (timeout_CH);

On timeout timer CH:

change_State (UN);

On timeout timer Neighbor:

change_State (UN);

CH state

Init:

Clusterhead = v;
send_Periodic_Hello_Msg (period);
set_Neighbor_Timer (timeout_Neighbor);

On timeout timer Neighbor:

change_State (UN);

On receive HELLO_CH(u):

If($Weight_u$ > $Weight_v$)
{
    Clusterhead = u;
    Change_State(MN);
}

Ο Cluster Based Medium Access Control Αλγόριθμος (CBMAC)

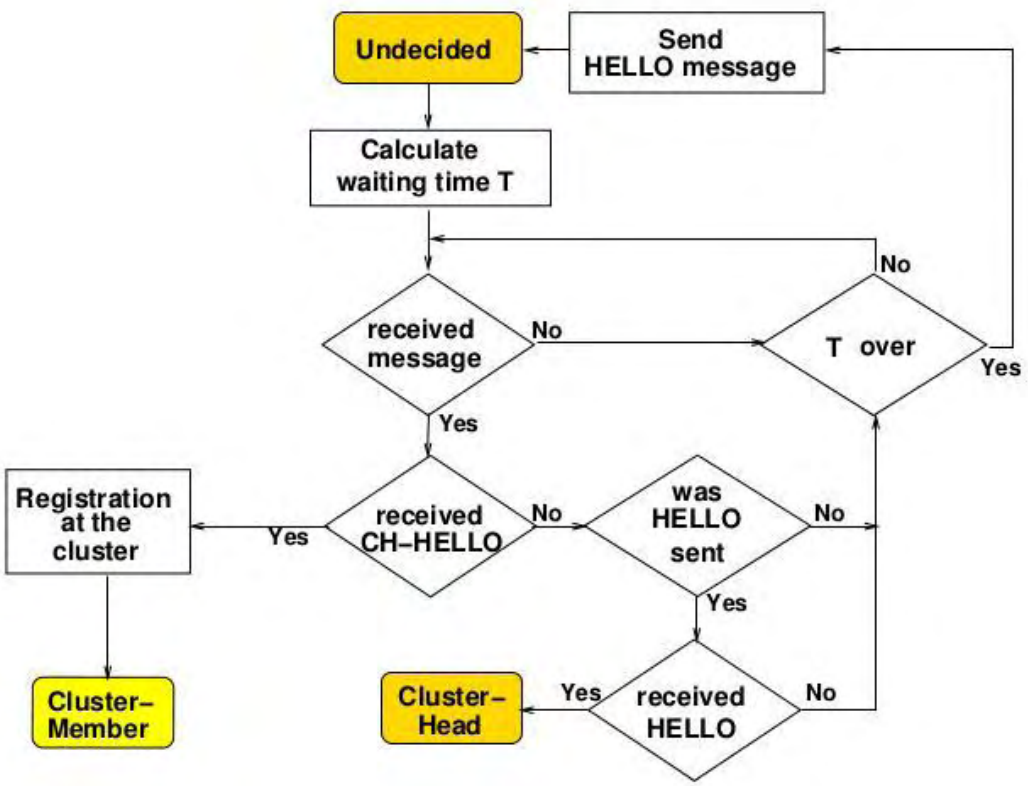
Ο αλγόριθμος CBMAC ακολουθεί τις βασικές αρχές του αλγορίθμου CBLR. Τα κοινά τους στοιχεία είναι η περιοδική μετάδοση μηνυμάτων “Hello” για τον κατανεμημένο προσδιορισμό της λειτουργίας των κόμβων. Στόχος είναι η αποφυγή των αλλαγών στα clusters και για αυτό ένα cluster head αλλάζει κατάσταση μόνο αν λάβει μήνυμα “Hello” από ένα άλλο cluster head.

Στον αλγόριθμο CBMAC ένας κόμβος, κατά τη διάρκεια της ζωής του, μπορεί να λειτουργεί σε μία από τις παρακάτω τέσσερεις καταστάσεις:

- Undecided
- Member
- Gateway
- Cluster head

Αρχική κατάσταση κάθε κόμβου ορίζεται η Undecided. Κάθε νέος κόμβος ξεκινάει σε αυτή την κατάσταση. Επιπλέον οι κόμβοι που δεν έχουνε κανένα γείτονα καταλήγουν σε αυτήν την κατάσταση. Ένας κόμβος στην Undecided κατάσταση στέλνει περιοδικά μηνύματα “Hello” για να ανακοινώσει την παρουσία του σε πιθανούς γειτονικούς κόμβους. Με την αποστολή των πακέτων ο κόμβος θα πρέπει επίσης να περιμένει για κάποιο χρονικό διάστημα αρκετά μεγάλο ώστε να λάβει μηνύματα από γειτονικά cluster heads. Για να είναι εφικτό αυτό θα πρέπει οι κόμβοι cluster heads να στέλνουν μηνύματα με μεγαλύτερη συχνότητα από τους αναποφάσιστους κόμβους.

Κατά τη διάρκεια αναμονής ο Undecided κόμβος θα πρέπει να ελέγχει για μηνύματα “Hello” από κάποιο cluster head. Σε περίπτωση που λάβει ένα τέτοιο μήνυμα ο κόμβος θα περάσει στην κατάσταση Member του συγκεκριμένου cluster head. Εάν λαμβάνει μηνύματα “Hello” από άλλους κόμβους σε άλλες καταστάσεις, ελέγχει αν έστειλε το δικό του “Hello” μήνυμα. Αυτό γίνεται για να εξακριβώσει ότι ο χρόνος αναμονής ήταν αρκετός για να εντοπίσει cluster heads, που βρίσκονται εντός εμβέλειας. Εάν ο κόμβος είχε στείλει μήνυμα “Hello” τότε γίνεται ο ίδιος cluster head, διαφορετικά παραμένει στην Undecided κατάσταση μέχρι να σιγουρευτεί ότι δεν υπάρχει cluster head στην γειτονιά του.



Εικόνα 8-Λειτουργία CBMAC από την κατάσταση Undecided

Στην κατάσταση του Gateway λειτουργούν οι κόμβοι εάν είναι μέλη σε παραπάνω από ένα cluster ή εάν έχουνε την δυνατότητα να λαμβάνουνε μηνύματα από μέλη διπλανών clusters. Η πρώτη περίπτωση συμβαίνει όταν ένας κόμβος λαμβάνει “Hello” μηνύματα άμεσα από δύο ή περισσότερα cluster heads. Στην δεύτερη περίπτωση είναι απαραίτητο να εξακριβώσει ότι ο κόμβος είναι μέλος ενός άλλου cluster. Για αυτό το σκοπό οι κόμβοι συμπεριλαμβάνουν στα “Hello” πακέτα τους τα αναγνωριστικά των cluster heads στα οποία είναι δηλωμένοι. Έτσι όλοι οι παραλήπτες του πακέτου γνωρίζουνε για την ύπαρξη ενός γειτονικού cluster head. Πληροφορίες σχετικά με τα γειτονικά clusters αποθηκεύονται σε έναν πίνακα ώστε οι κόμβοι να γνωρίζουν την τρέχουσα τοπολογία του δικτύου.

Όλοι οι κόμβοι από τους οποίους λαμβάνονται μηνύματα κρατούνται σε έναν πίνακα γειτόνων, που ανανεώνεται σύμφωνα με τα περιοδικά “Hello”

μηνύματα. Όποτε ένας κόμβος λαμβάνει ένα τέτοιο μήνυμα ο αποστολέας αποθηκεύεται στον πίνακα των γειτόνων. Επιπλέον ο cluster head χρησιμοποιεί τα μηνύματα δήλωσης ενός μέλους για να ενημερώσει τον πίνακα γειτόνων του. Για να υπάρξει σιγουριά ότι ο πίνακας μένει ενημερωμένος οι κόμβοι πρέπει να μεταδίδουν μηνύματα “Hello” σε μικρές περιόδους. Εάν δεν λαμβάνονται μηνύματα “Hello” για μία συγκεκριμένη χρονική περίοδο από έναν κόμβο, τότε θεωρείτε ότι ο κόμβος είναι εκτός εμβέλειας λήψης και έτσι αφαιρείται από τον πίνακα γειτόνων. Για ένα cluster head αυτό συνεπάγεται ότι ο κόμβος δεν είναι πλέον μέλος του cluster του.

Αλλαγές στους πίνακες του cluster και γειτόνων μπορούν να προκαλέσουνε αλλαγές και στην κατάσταση των κόμβων. Ένας κόμβος στην κατάσταση Member, που πλέον δεν λαμβάνει “Hello” από το cluster head του, θα πρέπει να αλλάξει κατάσταση σε Undecided. Ένας κόμβος Gateway, που χάνει επικοινωνία με ένα cluster head θα πρέπει να ελέγξει με πόσα cluster heads είναι ακόμα συνδεδεμένο. Αν ανήκει μόνο σε ένα θα αλλάξει κατάσταση σε Member, αν ανήκει σε περισσότερα παραμένει Gateway.

Υπάρχουνε ακόμα οι περιπτώσεις , όπου cluster heads πρέπει να αλλάξουνε κατάσταση. Η πρώτη αιτία είναι ότι το cluster head δεν έχει κανένα κόμβο μέλος στο cluster του. Σε αυτή την περίπτωση δεν είναι πλέον συνδεδεμένο στο δίκτυο και αλλάζει κατάσταση σε Undecided. Δεύτερη αιτία είναι η εμφάνιση δύο cluster heads σε εμβέλεια επικοινωνίας μεταξύ τους. Σε αυτήν την περίπτωση είναι απαραίτητο ένα από τα cluster heads να αλλάξει κατάσταση και να γίνει Member του άλλου. Η απόφαση για το ποιός κόμβος θα αλλάξει κατάσταση βασίζεται σε έναν παράγοντα βάρους. Για τον υπολογισμό του βάρους λαμβάνεται υπόψη η συνδεσιμότητα, η κινητικότητα και η μέση απόσταση των γειτόνων. Η συνδεσιμότητα ορίζεται η διαφορά ως προς τον βέλτιστο αριθμό γειτόνων. Κινητικότητα είναι η μέση σχετική ταχύτητα των γειτονικών κόμβων. Ένα cluster head, που έχει περίπου την ίδια ταχύτητα όπως οι γείτονές του μπορεί να μείνει στην κατάσταση cluster head περισσότερο από έναν κόμβο που είναι πολύ πιο αργός ή γρήγορος από τους γύρω κόμβους του.

Για τον υπολογισμό της κινητικότητας ενός κόμβου M_v:

$$M_v = \frac{1}{k_1 * N} * \sum_{i=1}^{N} dv_i$$

Εξίσωση 3

Όπου k_1: συντελεστής κλίμακας ώστε η κινητικότητα να είναι αδιάστατο μέγεθος.

N: αριθμός των γειτονικών κόμβων.

dv_i: η σχετική ταχύτητα του κόμβου i.

Για τον υπολογισμό της μέσης απόστασης από όλους τους γείτονες D_v :

$$D_v = \frac{1}{k_2 * N} * \sum_{i=1}^{N} d_i$$

Εξίσωση 4

Όπου k_2 : συντελεστής κλίμακας ώστε η μέση απόσταση να είναι αδιάστατο μέγεθος.

N: αριθμός γειτονικών κόμβων.

d_i: η απόσταση του κόμβου i.

Παρόμοια με την κινητικότητα μικρές τιμές της μέσης απόστασης είναι επιθυμητές, καθώς όπως είναι αναμενόμενο οι κόμβοι που είναι τοποθετημένοι κοντά μεταξύ τους θα παραμένουν για περισσότερο χρόνο εντός εμβέλειας επικοινωνίας.

Συνδυάζοντας αυτά τα μέτρα, υπολογίζεται ένας συντελεστής βάρους, ο οποίος δείχνει την καταλληλότητα ενός κόμβου να γίνει cluster head. Όσο μικρότερο το βάρος W_v τόσο καταλληλότερος είναι ο κόμβος να γίνει cluster head.

$$W_v = w_1\Delta_v + w_2 D_v + w_3 M_v$$

$w_1+w_2+w_3=1,0 <= w_1,w_2,w_3 <= 1$

Εξίσωση 5

Ο συντελεστής βάρους πρέπει να περιλαμβάνεται σε όλα τα “Hello” μηνύματα. Έτσι ώστε όταν δύο cluster heads βρεθούν στην ίδια περιοχή να μπορέσουν να αποφασίσουνε ποιός είναι καταλληλότερος για τον ρόλο αυτό.

Ψευδοκώδικας

Λαμβάνονται οι ίδιες υποθέσεις όπως αναφέρονται παραπάνω στον lowest-id. Επιπλέον κάθε κόμβος συνδέεται με ένα βάρος. Το βάρος αυτό υπολογίζεται όπως περιγράφεται παραπάνω και στον ψευδοκώδικα θεωρείται η συνάρτηση UpdateWeight(), ότι αναλαμβάνει αυτή τη διαδικασία. Κάθε φορά που ένας κόμβος στέλνει ένα μήνυμα πρέπει να περιλαμβάνει και το βάρος του σε αυτό, για να μπορεί να γίνει η σύγκριση από τον κόμβο που θα λάβει το μήνυμα. Η λειτουργία του αλγορίθμου περιγράφεται από τον παρακάτω ψευδοκώδικα.

CBMAC procedure

Undecided state

Init:
Clusterhead = NULL;
send_Periodic_Hello_Msg (period);
set_CH_Timer (timeout_CH);

On receive HELLO_CH(u):
Clusterhead = u;
change_State (MN);

On timeout timer_CH:
change_State (CH);

MN state

Init:

send_Periodic_Hello_Msg (period);
set_CH_Timer (timeout_CH);
set_Neighbor_Timer (timeout_Neighbor);

On timeout timer CH:

change_State (UN);

On timeout timer Neighbor:

change_State (UN);

CH state

Init:

Clusterhead = v;
send_Periodic_Hello_Msg (period);
set_Neighbor_Timer (timeout_Neighbor);

On timeout timer Neighbor:

change_State (UN);

On receive HELLO_CH:

If($Weight_u$ < $Weight_v$)
{
    Change_State(MN);
}

Distributed and Mobility-Adaptive Clustering (DMAC)

Γενικά

Ο DMAC είναι ένας κατανεμημένος αλγόριθμος με στόχο την δημιουργία clusters σε ένα δίκτυο και την διατήρησης αυτών. Σε κάθε κόμβο αντιστοιχίζεται ένα βάρος το οποίο μπορεί να είναι είτε σταθερό είτε μεταβλητό. Κύρια απαίτηση του αλγορίθμου είναι όταν ένας κόμβος ορίζεται ως Cluster-head δεν πρέπει να έχει κανένα κόμβο μέλος με μεγαλύτερο βάρος. Ένας κόμβος v στέλνει μηνύματα CH(v), αν λειτουργεί ως Cluster-head ή Join(v,z) με z να είναι το Cluster-head, στο οποίο ανήκει. Οι αναγνώριση νέων γειτονικών κόμβων και η απομάκρυνση παλιών γίνεται με μετάδοση “Hello” μηνυμάτων.

Οι καταστάσεις στις οποίες λειτουργούν οι κόμβοι είναι δύο, απλός κόμβος (MN) ή κόμβος Cluster-head (CH). Κάθε κόμβος ξεκινά στην κατάσταση MN, χωρίς να έχει επιλέξει κάποιον κόμβο Cluster-head. Ελέγχει την κατάσταση και το βάρος των κόμβων γειτόνων του για να αποφασίσει σε τι κατάσταση θα λειτουργεί ο ίδιος. Αν ανάμεσα στους κόμβους γείτονες υπάρχει τουλάχιστον ένας κόμβος Cluster-head με μεγαλύτερο βάρος από τον ίδιο, τότε ο κόμβος γίνετε μέλος του Cluster-head με το μεγαλύτερο βάρος. Σε αντίθετη περίπτωση ο κόμβος θα μπει σε κατάσταση CH.

Όταν ένας κόμβος v αντιλαμβάνεται την απομάκρυνση ενός κόμβου u από τη γειτονιά του, ο κόμβος v πρέπει να ελέγξει αν ο u ανήκε στο ίδιο Cluster. Σε αυτή την περίπτωση ο v πρέπει να κάνει ανανέωση στον πίνακα του Cluster τον οποίο διατηρεί. Στην περίπτωση που ο v είναι κόμβος MN και ο u ήταν ο κόμβος Cluster-head του, ο v θα πρέπει να αποφασίσει έναν νέο ρόλο. Συγκεκριμένα ο v ελέγχει αν υπάρχει στη γειτονιά του ένας κόμβος Cluster-head με μεγαλύτερο βάρος από τον ίδιο. Αν υπάρχει τότε ο v γίνεται μέλος του Cluster-head με το μεγαλύτερο βάρος, αλλιώς γίνεται ο ίδιος Cluster-head.

Όταν ο κόμβος v αναγνωρίσει έναν νέο γειτονικό κόμβο u, πρέπει να ελέγξει αν ο u είναι Cluster-head. Αν αυτό ισχύει και το βάρος του u είναι μεγαλύτερο από το βάρος του Cluster-head του v τότε ο κόμβος v γίνεται μέλος του Cluster του u, άσχετα από την προηγούμενη κατάστασή του.

Όταν ένας κόμβος v λάβει μήνυμα CH από έναν κόμβο u, ο κόμβος v πρέπει να ελέγξει αν πρέπει να γίνει μέλος του κόμβου u. Δηλαδή ελέγχει αν ο u έχει μεγαλύτερο βάρος από το Cluster-head, που ανήκει.

Όταν ένας κόμβος λάβει μήνυμα Join(u,z) από έναν γειτονικό κόμβο u, η λειτουργία του κόμβου v εξαρτάται από την κατάσταση στην οποία λειτουργεί. Αν ο κόμβος v είναι Cluster-head ελέγχει αν ο κόμβος z είναι ο ίδιος και αν ναι ανανεώνει τον πίνακα του Cluster του, αλλιώς ελέγχει αν ο κόμβος u ανήκε πριν στο Cluster του για να τον αφαιρέσει από αυτό. Αν ο κόμβος v είναι απλός κόμβος θα πρέπει να ελέγξει αν ο u ήταν το Cluster-head του και αν αυτό ισχύει θα πρέπει να αποφασίσει τον νέο του ρόλο. Θα ελέγξει τους γειτονικούς κόμβους για κάποιο Cluster-head με μεγαλύτερο βάρος από τον ίδιο. Αν υπάρχει τουλάχιστον ένας τέτοιος κόμβος ο v γίνεται μέλος του Cluster-head με το μεγαλύτερο βάρος, σε αντίθετη περίπτωση γίνεται ο ίδιος ο v Cluster-head.

Ψευδοκώδικας

Λαμβάνονται οι ίδιες υποθέσεις όπως αναφέρονται παραπάνω στον lowest-id. Επιπλέον κάθε κόμβος συνδέεται με ένα βάρος. Το βάρος αυτό μπορεί να είναι μεταβαλλόμενο ή σταθερό. Στον ψευδοκώδικα που ακολουθεί θεωρείται ότι το βάρος ενός κόμβου είναι σταθερό και ορισμένο εκ των προτέρων. Κάθε φορά που ένας κόμβος στέλνει ένα μήνυμα πρέπει να περιλαμβάνει και το βάρος του σε αυτό, για να μπορεί να γίνει η σύγκριση από τον κόμβο που θα λάβει το μήνυμα. Η λειτουργία του αλγορίθμου περιγράφεται από τον παρακάτω ψευδοκώδικα.

DMAC procedure

MN state

Init:

```
Send_Periodic_Msg(period);
Set_CH_Timer(CH_timeout);
```

On Receive CH(u):

```
CH(u) = true;
If(u == Clusterhead) { return; }
If(for every z in Γ(v), so that Weight_z > Weight_u, Join(z, x) == true)
{
        Clusterhead = u;
        Send Join(v, Clusterhead);
}
```

On receive Join(u, t):

```
Join(u,t) = true;
If( CH(v) == t ) { UpdateClusterTable(); }
If( CH(v) == u)
{
        Find A(z) in Γ(v), so that (CH(z)==true AND Weight_z > Weight_v )
        If((x = maxWeight(A(z))) != NULL)
        {
            Clusterhead = x;
            Send Join(v, Clusterhead);
        }
}
```

On timeout timer CH:

```
If ( for every u in Γ(v) with Weight_v < Weight_u, Join(u,x) == true)
{
        Change_State( CH )
        UpdateClusterTable();
        CH(v) = true;
        Clusterhead = v;
} else { Set_CH_Timer(CH_timeout); }
```

CH state

Init :

```
Send_Periodic_Msg(period);
```

On Receive CH(u):

CH (u) = true;

If ($Weight_u > Weight_v$)

{

ChangeState(MN);

Clusterhead = u;

Send Join(v, Clusterhead);

}

On receive Join(u, t):

Join(u,t) = true;

If(CH(v))

{

If(Clusterhead == t) { UpdateClusterTable(); }

}

Ο αλγόριθμος ALM

Γενικά

Ο αλγόριθμος ALM είναι μια παραλλαγή των αλγορίθμων που παρουσιάστηκαν στο προηγούμενο κεφάλαιο. Κύρια προβλήματα του CBLR είναι η προτεραιότητα που δίνει στα μεγαλύτερα clusters και ότι έχει πολλούς κόμβους στην κατάσταση cluster head. Χρησιμοποιώντας ένα μέτρο κινητικότητας για την σταθεροποίηση της επιλογής του cluster head τα προβλήματα αντιμετωπίζονται. Όμως πάλι τα cluster heads μπορεί να μεταβάλλουν την κατάσταση τους γρήγορα και αναίτια εάν βρεθούν εντός εμβέλειας επικοινωνίας μεταξύ τους.

Κύριος στόχος του αλγορίθμου ALM είναι η δημιουργία όσο το δυνατόν πιο σταθερών clusters. Για να επιτευχθεί αυτό απαιτείται οι κόμβοι κεφαλές των clusters να μην αλλάζουν συχνά κατάσταση λειτουργίας. Η γενική ιδέα ώστε να επιτευχθεί αυτό είναι η μελέτη της τοπικής κινητικότητας των κόμβων γύρο από ένα cluster head, για να επιτευχθεί μια σταθερότητα των δομών του δικτύου.

Στον αλγόριθμο MOBIC, που παρουσιάστηκε στο προηγούμενο κεφάλαιο, προτείνεται μια μετρική σχετικής κινητικότητας για τους κόμβους των VANETs. Η ισχύς του λαμβανόμενου σήματος χρησιμοποιείτε ως μέτρο για την απόσταση μεταξύ του αποστολέα και παραλήπτη. Μετρώντας το κλάσμα της ισχύος ανάμεσα σε δύο συνεχόμενες λήψεις μηνυμάτων μπορεί να υπολογιστεί ένα μέτρο σχετικής κινητικότητας των κόμβων. Ως μια επιπλέον τεχνική σταθεροποίησης των clusters χρησιμοποιείτε η καθυστέρηση αναδιάρθρωσης των clusters για ένα χρονικό διάστημα όταν δύο cluster heads βρεθούν σε εμβέλεια επικοινωνίας μεταξύ τους. Έτσι μια τυχαία επικοινωνία ανάμεσα σε δύο cluster heads δεν θα προκαλέσει άσκοπη αναπροσαρμογή του δικτύου.

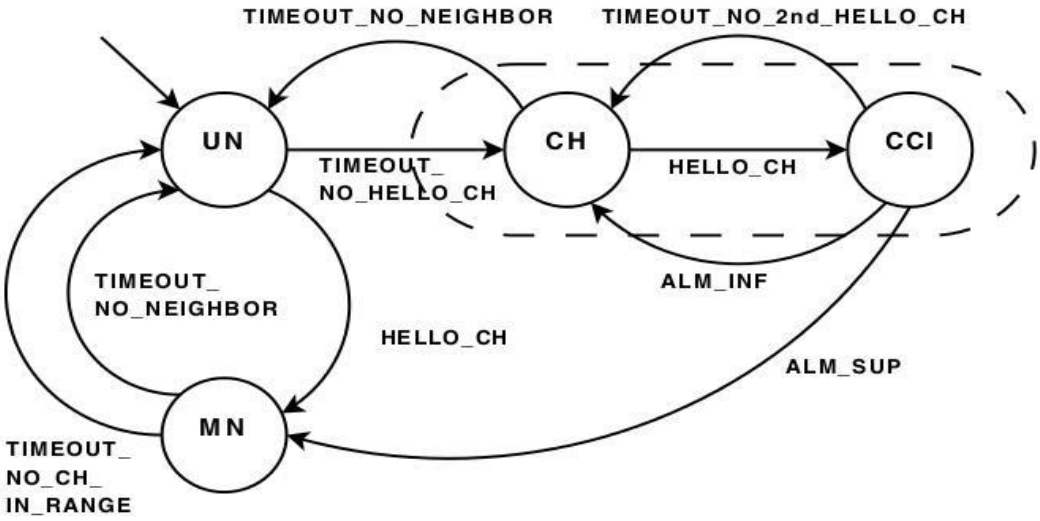
Βασισμένος στις παραπάνω αρχές είναι και αλγόριθμος ALM, αλλά θέτοντας πιο συγκεκριμένους κανόνες.

- Απαιτείται τα cluster heads να ανταλλάσουν παραπάνω από ένα μηνύματα, έτσι ώστε να αρχίσουν την θεώρηση μιας αναδιάρθρωσης του δικτύου.

- Η απόφαση των κόμβων σχετικά με την αλλαγή της κατάστασής τους βασίζεται στην εικόνα που έχει κάθε κόμβος σχετικά με την συνολική κίνηση των κόμβων στο περιβάλλον του. Τα cluster heads, των οποίων οι κόμβοι γύρο τους παρουσιάζουν μικρότερη κινητικότητα διατηρούν την κατάστασή τους, ενώ τα άλλα αλλάζουνε την κατάσταση τους.

- Για να αποφευχθεί η δημιουργία πολλών κόμβων cluster heads, ορίζεται ότι ένας κόμβος μέλος απαγορεύεται να αλλάξει άμεσα την κατάστασή του σε cluster head όταν σταματήσει να λαμβάνει περιοδικά μηνύματα από το cluster head του, και δεν υπάρχει άλλο cluster head στην περιοχή του. Ένας κόμβος στην κατάσταση Member θα πρέπει πρώτα να περάσει στην κατάσταση Undecided. Αυτή η αλλαγή αναβάλλει την δημιουργία ενός νέου cluster, το οποίο θα μπορούσε να πυροδοτήσει μια περιττή αναδιάρθρωση των clusters. Επίσης δίνεται χρόνος στον κόμβο να εντοπίσει ένα άλλο cluster head εντός εμβέλειας και να συνδεθεί σε αυτό.

Τα παραπάνω χαρακτηριστικά ενσωματώνονται στον αλγόριθμο και δίνουνε την μηχανή πεπερασμένων καταστάσεων που φαίνεται στην Εικόνα 8.



Εικόνα 9-Μηχανή πεπερασμένων καταστάσεων για τον αλγόριθμο ALM

Όπως φαίνεται στην Εικόνα 8 στον αλγόριθμο υπάρχουν τέσσερεις καταστάσεις λειτουργίας των κόμβων:

- **Κατάσταση Cluster Head (CH)** : κεφαλή του cluster. Είναι ο κόμβος που αναλαμβάνει να λειτουργεί ως αντιπρόσωπος του cluster.
- **Κατάσταση Member (MN)** : κόμβος μέλος ενός cluster. Ανήκει σε ένα cluster και έχει εντός εμβέλειας επικοινωνίας τουλάχιστον ένα cluster head, το οποίο είναι ο αντιπρόσωπός του και ο συντονιστής του.
- **Κατάσταση Undecided (UN)** : ο κόμβος δεν έχει λάβει ακόμα απόφαση σε ποιά εκ των παραπάνω καταστάσεων θα λειτουργήσει.
- **Κατάσταση Contention (CCI)** : κατάσταση στην οποία ο κόμβος πρέπει να αποφασίσει αν θα συνεχίσει ως cluster head ή θα αλλάξει κατάσταση σε member ενός γειτονικού cluster head.

Undecided Node κατάσταση (UN)

Η αρχική κατάσταση κάθε νέου κόμβου είναι η UN. Ο κόμβος λαμβάνει μηνύματα από το ασύρματο κανάλι και περιοδικά στέλνει “Hello” μηνύματα στο κανάλι για να ενημερώσει τυχών γείτονες κόμβους για την παρουσία του. Ορίζει επίσης ένα χρονικό όριο αναμονής για τον εντοπισμό κάποιου cluster head στην γειτονιά του. Μόλις ο κόμβος λάβει ένα μήνυμα “Hello” από κάποιο cluster head αλλάζει την κατάστασή του σε MN.

Init:
send_Periodic_Hello_Msg (period);
set_CH_Timer (timeout_CH);

On receive HELLO_CH:
change_State (MN);

On timeout timer_CH:
change_State (CH);

Member Node κατάσταση (MN)

Στην κατάσταση MN ένας κόμβος λειτουργεί ως απλό μέλος ενός cluster. Στην εμβέλεια λήψης του κόμβου υπάρχει τουλάχιστον ένας κόμβος cluster head. Σε αυτή την κατάσταση ο κόμβος περιμένει για μηνύματα “Hello” από τον cluster head του καθώς και από τους γειτονικούς κόμβους. Αν περάσει ένα συγκεκριμένο χρονικό όριο όπου ο κόμβος δεν λάβει μήνυμα “Hello” από το cluster head του, θα πρέπει να αλλάξει την κατάστασή του σε UN. Αν περάσει ένα χρονικό όριο όπου δεν έχει λάβει μήνυμα “Hello” από κανένα γείτονα κόμβο (cluster head ή απλό κόμβο) τότε αλλάζει την κατάστασή του σε UN.

Init:
send_Periodic_Hello_Msg (period);
set_CH_Timer (timeout_CH);
set_Neighbor_Timer (timeout_Neighbor);

On timeout timer CH:
change_State (UN);

On timeout timer Neighbor:
change_State (UN);

Cluster Head κατάσταση (CH)

Ο κόμβος είναι η κεφαλή του cluster. Η λειτουργία του στην κατάσταση CH είναι πιο περίπλοκη από την λειτουργία στις άλλες. Ο κόμβος στέλνει περιοδικά “Hello” μηνύματα στους γείτονες κόμβους για να ενημερώνει ότι είναι ακόμα ενεργός. Αν ο CH δεν λάβει “Hello” μηνύματα από κανένα άλλο κόμβο, που λειτουργεί στην κατάσταση CH συνεχίζει να λειτουργεί ως CH. Αν δεν λάβει “Hello” μηνύματα από κανένα κόμβο γείτονα, τότε πρέπει να αλλάξει την κατάστασή του σε UN.

Το ενδιαφέρον παρουσιάζετε όταν δύο κόμβοι που λειτουργούν στην CH κατάσταση ανταλλάξουν μηνύματα. Όταν ένα CH λάβει ένα μήνυμα

“Hello” από ένα άλλο CH, θα πρέπει να αλλάξει στην κατάσταση Contention CCI.

Init:
send_Periodic_Hello_Msg (period);
set_Neighbor_Timer (timeout_Neighbor);

On timeout timer Neighbor:
change_State (UN);

On receive HELLO_CH:
change_State (CCI);

Κατάσταση Contention (CCI)

Αυτή είναι μια κατάσταση ανταγωνισμού ανάμεσα στους δύο κόμβους που λειτουργούν στην κατάσταση CH. Ο κόμβος συνεχίζει να λειτουργεί ως CH στέλνοντας περιοδικά μηνύματα “Hello” στους γείτονές του. Ο πρώτος κόμβος περιμένει για ένα δεύτερο μήνυμα HELLO από τον κόμβο, από όπου έλαβε το πρώτο “Hello”. Αν ο κόμβος δεν λάβει δεύτερο μήνυμα “Hello”, τότε το πρώτο μήνυμα απλά αγνοείτε και ο κόμβος συνεχίζει στην κατάσταση CH. Σε περίπτωση που ο κόμβος στην κατάσταση CCI λάβει ένα δεύτερο μήνυμα “Hello” από τον ίδιο γείτονα CH θα πρέπει να αποφασίσει αν θα αλλάξει την κατάστασή του ή όχι. Την απόφαση αυτή την παίρνει βασισμένο στην σχετική κινητικότητα που έχουν οι δύο κόμβοι. Τα βάρη ALM (Aggregate Local Mobility) στέλνονται μέσα στα “Hello” μηνύματα των κόμβων. Έτσι ο κόμβος με το μικρότερο βάρος ALM παραμένει στην κατάσταση CH, ενώ ο άλλος θα πάρει την απόφαση να πάει στην κατάσταση MN στο cluster του πρώτου.

Init:
send_Periodic_Hello_Msg (period);

On timeout timer CH:

```
change_State ( CH );

On receive HELLO_CH:
m_ALM = calculate_ALM ();
if ( msg.ALM < ALM) {
        change_State (MN);
} else {
        change_state (CH);
}
```

Υπολογισμός του Aggregate Local Mobility (ALM)

Για τον υπολογισμό της συσσωρευμένης τοπικής κινητικότητας των οχημάτων μπορεί να χρησιμοποιηθεί η ισχύς της λήψης ενός σήματος (received signal strength), όπως στον αλγόριθμο MOBIC, το οποίο είναι πολύ αναξιόπιστο. Στην συγκεκριμένη θεώρηση χρησιμοποιείται η τοποθεσία των κόμβων. Η υπόθεση είναι ότι κάθε κόμβος γνωρίζει την θέση του με την χρήση για παράδειγμα ενός GPS. Οπότε σε κάθε μηνύματος, προς αποστολή κάθε κόμβος προσθέτει την τοποθεσία του. Το κλάσμα των αποστάσεων των κόμβων ανάμεσα σε δύο διαδοχικές λήψεις ενός “Hello” μηνύματος δείχνει την σχετική κινητικότητα ανάμεσα στους δύο κόμβους, όπως παρουσιάζεται στην Εξίσωση 6.

$$M_Y^{rel}(X) = log \frac{Dist_{current}}{Dist_{previous}}$$

Εξίσωση 6

Έχοντας την σχετική κινητικότητα κάθε κόμβου με τους γείτονές του η ALM ενός κόμβου υπολογίζεται παίρνοντας την διακύμανση των σχετικών κινητικοτήτων γύρω από το μηδέν (Εξίσωση 7).

$$M_Y = var_0(M_Y^{rel}(X_j))$$

Εξίσωση 7

Δηλαδή η συνολική τοπική κινητικότητα ενός κόμβο αντιστοιχεί στο μέσο όρο του τετράγωνο των σχετικών κινητικοτήτων των κόμβων.

Εάν $Dist_{currentX->Y} < Dist_{previousX->Y}$ => $M_y^{rel}(X) < 0$. Δηλαδή όταν δύο κόμβοι κινούνται κοντά ο ένας στον άλλο τότε εμφανίζεται αρνητική σχετική κινητικότητα μεταξύ τους.

Εάν $Dist_{currentX->Y} > Dist_{previousX->Y}$ => $M_y^{rel}(X) > 0$. Δηλαδή όταν δύο κόμβοι κινούνται κοντά ο ένας στον άλλο τότε εμφανίζεται θετική σχετική κινητικότητα μεταξύ τους.

Διαισθητικά γίνετε κατανοητό ότι όσο μικρότερη η διακύμανση της σχετικής κινητικότητας ενός κόμβου σε σχέση με τους γείτονές του, τόσο λιγότερες είναι οι μετακινήσεις ανάμεσα τους και άρα η επιλογή του κόμβου με το μικρότερο ALM ως CH έχει ως στόχο την πιο σταθερή επιλογή.

Προσομοιώσεις και αποτελέσματα

Γενικά

Για την προσομοίωση της λειτουργίας του αλγορίθμου σε δίκτυο κινούμενων κόμβων τα εργαλεία λογισμικού που χρησιμοποιήθηκαν είναι το ns-3 και το SUMO(Simulator of Urban Mobility). Για τον έλεγχο των επιδόσεων του αλγορίθμου ALM Clustering υλοποιήθηκε και ο αλγόριθμος DMAC με μεταβλητό βάρος κόμβων. Οι αλγόριθμοι έτρεξαν στα ίδια δίκτυα.

Παρακάτω στο κεφάλαιο παρουσιάζονται οι προσομοιωτές ns-3 και SUMO. Στη συνέχεια υπάρχει αναφορά στη μορφή των οδικών δικτύων πάνω στο οποίο έγινε η προσομοίωση και στις μεταβλητές του περιβάλλοντος. Τέλος, παρουσιάζονται τα αποτελέσματα από τις προσομοιώσεις που εκτελέστηκαν και παρατηρήσεις πάνω σε αυτά.

NS-3

Ο ns-3 είναι ένας προσομοιωτής δικτύου διακριτών γεγονότων, ο οποίος έχει κυρίως εκπαιδευτική και ερευνητική εφαρμογή. Ο ns-3 είναι ένα ελεύθερο λογισμικό, βρίσκεται κάτω από την άδεια GNU GPLv2 και είναι διαθέσιμο στον κόσμο για έρευνα, ανάπτυξη και χρήση.

Στόχος του έργου του ns-3 είναι η ανάπτυξη ενός περιβάλλοντος εξομοίωσης για μελέτες δικτύων. Θα πρέπει να ευθυγραμμίζεται με τις ανάγκες της σύγχρονης έρευνας στον τομέα των δικτύων και να ενθαρρύνει την συνεισφορά, επανεξέταση και επικύρωση του λογισμικού από την κοινότητα του ανοιχτού λογισμικού.

Στο ns-3 υλοποιήθηκε ο αλγόριθμος ALM και ο αλγόριθμος DMAC, ως ανταγωνιστής για τι σύγκριση των επιδόσεων των δύο. Η τοπολογία του οδικού δικτύου και η κίνηση των αυτοκινήτων μοντελοποιήθηκε στο SUMO και έγινε εισαγωγή της στο ns-3. Τα βάρη στην αρχικοποίηση των κόμβων

τέθηκαν σε μη έγκυρες τιμές. Οι αναγνώριση της εμφάνισης νέων γειτόνων γίνεται μέσω περιοδικών μηνυμάτων με περίοδο 0.5 δευτερολέπτων. Και η αναγνώριση της εξαφάνισης ενός κόμβου γίνεται με την αποτυχία να παραδοθεί ένα περιοδικό μήνυμα από έναν συγκεκριμένο κόμβο. Τα timeouts για το μήνυμα “HELLO” είναι ορισμένο στα 1.5 δευτερόλεπτα. Τα μηνύματα “HELLO” περιέχουν πληροφορίες για τον υπολογισμό του βάρους ALM, δηλαδή το αναγνωριστικό του κόμβου, τη θέση του, την κατάστασή του και το βάρος του.

SUMO (Simulator of Urban Mobility)

"**S**imulation of **U**rban **MO**bility", ή "SUMO" είναι ένα λογισμικό ανοιχτού κώδικα, το οποίο επιτρέπει την προσομοίωση των απαιτήσεων κυκλοφορίας, που αποτελείται από την κίνηση διακριτών οχημάτων σε ένα συγκεκριμένο οδικό δίκτυο. Η προσομοίωση επιτρέπει την αντιμετώπιση ενός μεγάλου συνόλου θεμάτων διαχείρισης της κυκλοφορίας. Το SUMO είναι καθαρά μικροσκοπικό, καθώς κάθε όχημα μοντελοποιείται ρητά, έχει τη δική του διαδρομή, και κινείται μεμονωμένα μέσα στο δίκτυο.

Στο SUMO κάθε όχημα περιγράφεται από ένα σύνολο παραμέτρων. Πιο συγκεκριμένα επιτάχυνση και επιβράδυνση οχήματος, μέγεθος και μέγιστη ταχύτητα. Το μήκος οχήματος που ορίστηκε στα πειράματα ήταν τρία μέτρα. Οι μέγιστες ταχύτητες των οχημάτων ορίστηκαν στα 30km/h, 50km/h και 70km/h. Οι επιταχύνσεις και επιβραδύνσεις ορίστηκαν σε τέτοιες τιμές ώστε να φτάνουνε άμεσα τη ορισμένη μέγιστη ταχύτητα και αντίστοιχα να σταματάνε άμεσα.

Στο SUMO έγινε ο σχεδιασμός της τοπολογίας του οδικού δικτύου. Έγινε ο ορισμός των διαδρομών που ακολουθούνε τα οχήματα μέσα στο δίκτυο αυτό. Ορίστηκε το πλήθος των οχημάτων-κόμβων, τα οποία κινούνται μέσα σε αυτές τις τοπολογίες. Τα αποτελέσματα της κίνησης των κόμβων έγιναν εισαγωγή στο ns-3 για τον έλεγχο των αλγορίθμων που συγκρίθηκαν.

Ανεξάρτητες μεταβλητές της προσομοίωσης

Στις προσομοιώσεις οι ανεξάρτητες μεταβλητές του περιβάλλοντος, οι οποίες λήφθηκαν υπόψη είναι:

- Αριθμός των οχημάτων.
- Ταχύτητα των οχημάτων.
- Μέγεθος τοπολογίας.

Αριθμός οχημάτων

Ο αριθμός των οχημάτων δηλώνει το πόσο πυκνό ή αραιό είναι το δίκτυο. Σε κάθε προσομοίωση που έγινε ο αριθμός των οχημάτων μένει σταθερός κατά τη διάρκεια αυτής. Δηλαδή τα οχήματα τα οποία ξεκινάνε από μια θέση προς έναν προορισμό, όταν τελειώσουνε το δρομολόγιο τους, δρομολογούνται ξανά προς μία νέα τυχαία κατεύθυνση. Κανένα όχημα δεν χάνεται από το δίκτυο και νέα οχήματα δεν προστίθενται σε αυτό κατά τη διάρκεια της προσομοίωσης.

Στις προσομοιώσεις που έγιναν ο αριθμός των οχημάτων παίρνει τις τιμές 10, 20, 30, 40 και 50 για το οδικό δίκτυο 1 και τις τιμές 40, 80, 120, 160 και 200 για το οδικό δίκτυο 2.

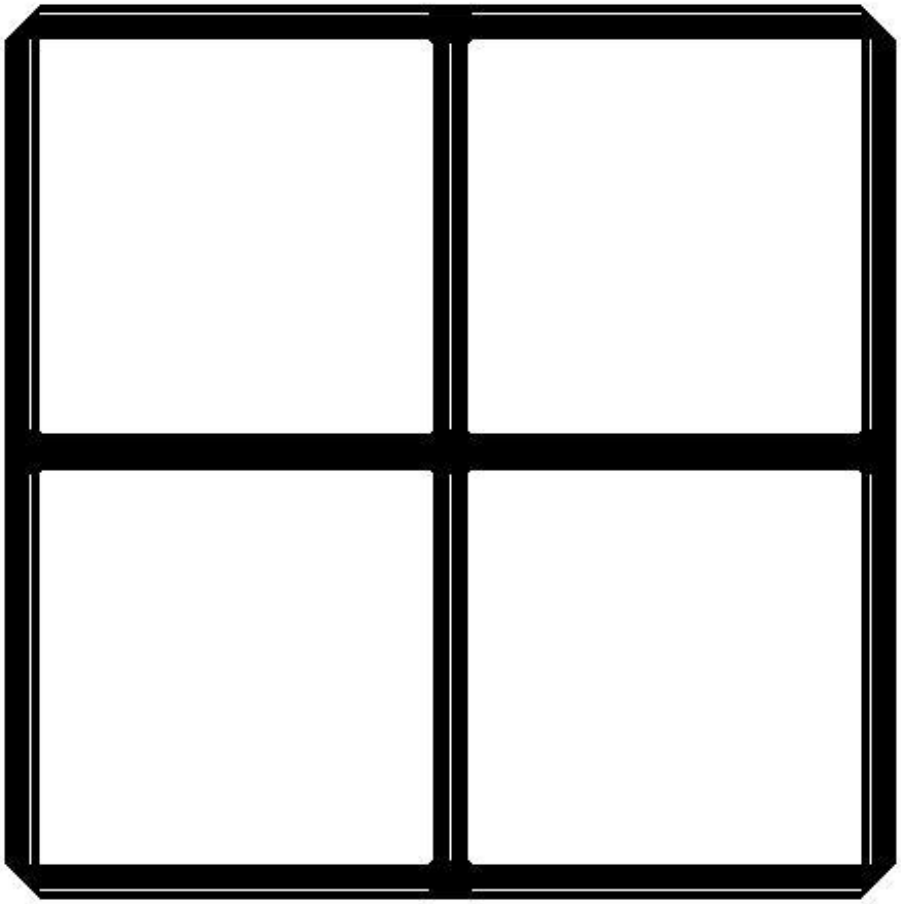
Ταχύτητα οχημάτων

Η ταχύτητα με την οποία κινούνται τα οχήματα μέσα στο οδικό δίκτυο δείχνει το πόσο γρήγορα αλλάζει η εικόνα που υπάρχει για το δίκτυο. Όσο μεγαλύτερη η ταχύτητα των οχημάτων τόσο πιο δυναμικό θα είναι το δίκτυο και θα υπάρχει συνεχή αλλαγή στους γειτονικούς κόμβους. Αυτό επηρεάζει την σταθερότητα των αλγορίθμων clustering, και είναι σημαντικό να ελεγχθεί η συμπεριφορά του αλγορίθμου σε διαφορετικές ταχύτητες οχημάτων.

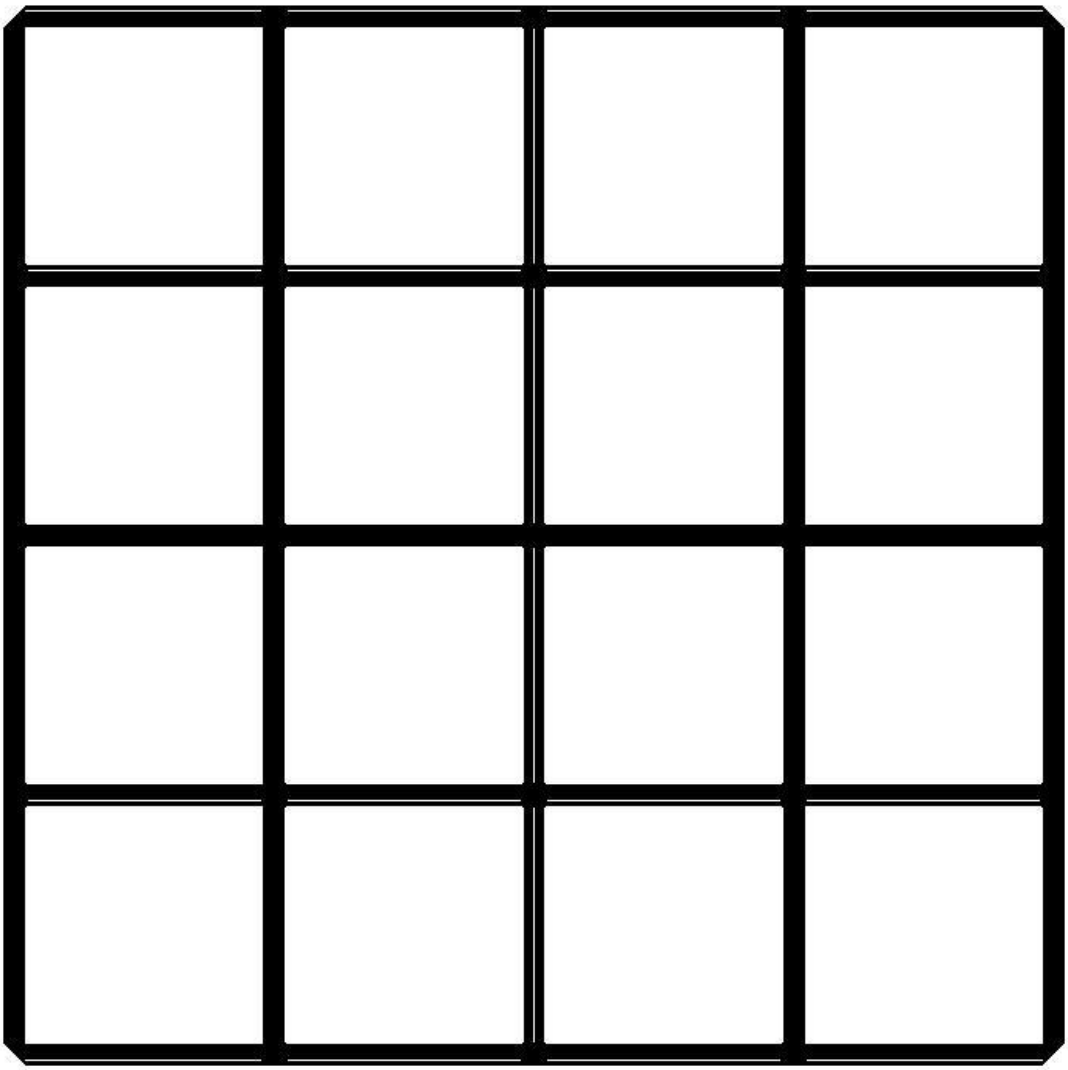
Στις προσομοιώσεις που έγιναν η ταχύτητα των οχημάτων παίρνουν τις τιμές 30km/h 50km/h και 70km/h και για τα δύο οδικά δίκτυα.

Μέγεθος τοπολογίας

Το μέγεθος της τοπολογίας καθώς και η μορφή της καθορίζουνε την συμπεριφορά των οχημάτων σε αυτό. Είναι λογικό ότι διαφορετικά συμπεριφέρεται ένα όχημα σε έναν αυτοκινητόδρομο και διαφορετικά εντός πόλεως. Το οδικό δίκτυο στο οποίο έγιναν οι προσομοιώσεις έχει τη μορφή πλέγματος. Σχεδιάστηκαν δύο τοπολογίες. Η πρώτη είναι ένα πλέγμα 3x3(Εικόνα 10 - οδικό δίκτυο 1) και η δεύτερη πλέγμα 5x5(Εικόνα 11 - οδικό δίκτυο 2). Οι δρόμοι που αποτελούν το πλέγμα έχουνε μήκος 160 μέτρων ο καθένας και κάθε δρόμος διαθέτει δύο λωρίδες κυκλοφορίας προς κάθε κατεύθυνση. Δεν υπάρχουνε φωτεινοί σηματοδότες στις διασταυρώσεις και τα οχήματα ακολουθούν σύστημα προτεραιότητας κίνησης για να αποφύγουνε συγκρούσεις μεταξύ τους.



Εικόνα 10-Οδικό δίκτυο 1



Εικόνα 11-Οδικό δίκτυο 2

Εξαρτημένες μεταβλητές προσομοίωσης

Στις προσομοιώσεις που εκτελέστηκαν τα μεγέθη τα οποία μετρήθηκαν είναι:

- **Μέσος χρόνος ζωής ενός Cluster-heads**: μέση διάρκεια συνεχών περιόδων χρόνου, όπου ο κόμβος λειτουργούσε ως Cluster-head. Ο χρόνος μετράτε σε δευτερόλεπτα.

- **Μέσος αριθμός αλλαγών καταστάσεων που έγιναν σε ένα κόμβο**: ο μέσος όρος των αλλαγών καταστάσεων ανά κόμβο κατά τη διάρκεια της ζωής του.

- **Μέσο μέγεθος των clusters**: το πλήθος των κόμβων που λειτουργούν στην κατάσταση μέλους προς το σύνολο των clusters ανά τακτές χρονικές στιγμές.

- **Μέσος αριθμός clusters**: το πλήθος των κόμβων που λειτουργούν στην κατάσταση Cluster-head ανά τακτές χρονικές στιγμές.

Ο μέσος χρόνος ζωής των Cluster-heads και ο μέσος αριθμός αλλαγών καταστάσεων σε έναν κόμβο χρησιμοποιούνται ως μέτρο για την σταθερότητα του αλγορίθμου. Όσο πιο σταθερός είναι ο αλγόριθμος τόσο λιγότερες είναι οι αλλαγές των καταστάσεων στους κόμβους και τόσο μεγαλύτερος ο χρόνος ζωής των Cluster-heads.

Για τον υπολογισμό του μέσου αριθμού των clusters και του μέσου μεγέθους αυτών, ορίζονται συναρτήσεις που περιοδικά ελέγχουνε όλους τους κόμβους του δικτύου. Οι συναρτήσεις αυτές ελέγχουνε τους κόμβους κάθε δύο δευτερόλεπτα. Μετρούν τον αριθμό των ενεργών Cluster-heads που είναι ο αριθμός των ενεργών clusters και υπολογίζουνε το μέσο όρο με τον παλιό αριθμό των ενεργών clusters. Με παρόμοιο τρόπο μετρούν και το μέσω μέγεθος του κάθε cluster.

Αποτελέσματα και παρατηρήσεις

Αποτελέσματα οδικό δίκτυο 1

Στο οδικό δίκτυο 1 παρατηρήθηκαν τα παρακάτω αποτελέσματα:

Πίνακας 1 Αριθμός αλλαγών κατάστασης - Οδικό δίκτυο 1

#οχημάτων	10		20		30		40		50	
Ταχύτητα οχημ.	**ALM**	**DMAC**	**ALM**	**DMAC**	**ALM**	**DMAC**	**ALM**	**DMAC**	**ALM**	**DMAC**
30 km/h	19.7	10.5	11	11	8.3	11.6	10.2	18.1	7.4	14.1
50 km/h	22.4	15.3	12	18.1	13.7	26.4	8.8	22.9	9.6	25.3
70 km/h	24.3	22.3	14	26	11.8	27	10.5	27.8	9.3	30

Πίνακας 2 Μέσος χρόνος ζωής ενός Cluster-head - Οδικό δίκτυο 1

#οχημάτων	10		20		30		40		50	
Ταχύτητα οχημ.	ALM	DMAC	ALM	DMAC	ALM	DMAC	ALM	DMAC	ALM	DMAC
30 km/h	14.3	21.9	8.4	12	7.2	5.1	4.6	4.5	3.3	2
50 km/h	8.1	11	6.6	7.9	3.5	2.3	3.6	2.7	3.1	2.3
70 km/h	7	10.6	4.2	4.5	4.6	3.3	3.6	2.2	3.5	1.9

Πίνακας 3 Μέσος αριθμός clusters - Οδικό δίκτυο 1

#οχημάτων	10		20		30		40		50	
Ταχύτητα οχημ.	ALM	DMAC	ALM	DMAC	ALM	DMAC	ALM	DMAC	ALM	DMAC
30 km/h	2.9	3	3.1	3.9	3.2	3.9	4	4	4	3.2
50 km/h	3.2	3.2	4	4.8	6	3.8	3.8	3.1	3.2	3.9
70 km/h	3	3	4.7	5	3.1	3.2	3.1	2.2	3	3.2

Πίνακας 4 Μέσο μέγεθος των Clusters - Οδικό δίκτυο 1

#οχημάτων	10		20		30		40		50	
Ταχύτητα οχημ.	ALM	DMAC	ALM	DMAC	ALM	DMAC	ALM	DMAC	ALM	DMAC
30 km/h	3.2	3.3	5.8	5.2	8.9	7.9	10.2	10.1	12.4	15
50 km/h	3	2.9	5.2	4.4	6	7.1	10.7	12.3	14.6	13
70 km/h	3.3	3.3	4.3	4	9.4	8.7	12.7	16	16.3	15.2

Αποτελέσματα οδικό δίκτυο 2

Στο οδικό δίκτυο 5x5 παρατηρήθηκαν τα παρακάτω αποτελέσματα

Πίνακας 5 Αριθμός αλλαγών κατάστασης - Οδικό δίκτυο 2

#οχημάτων	40		80		120		160		200	
Ταχύτητα οχημ.	ALM	DMAC	ALM	DMAC	ALM	DMAC	ALM	DMAC	ALM	DMAC
30 km/h	9	10.7	7.5	13.2	9.8	21.9	10.7	23.8	8.3	28.4
50 km/h	13.4	15.3	11.6	21.4	11.7	28.4	11.5	33.7	9.2	33
70 km/h	17.5	21.6	12.2	27.8	11.6	33.8	9.6	31.9	9.9	32

Πίνακας 6 Μέσος χρόνος ζωής ενός Cluster-head - Οδικό δίκτυο 2

#οχημάτων	40		80		120		160		200	
Ταχύτητα οχημ.	ALM	DMAC	ALM	DMAC	ALM	DMAC	ALM	DMAC	ALM	DMAC
30 km/h	9.1	12.7	6.9	6.1	3.3	3.9	2.5	2.1	3	1.7
50 km/h	6.7	8.3	4.6	4.8	3.8	2.5	2.4	1.4	2.4	1.3
70 km/h	3.8	7.7	4.9	3	2.8	2	2.7	1.7	2.7	1.3

Πίνακας 7 Μέσος αριθμός clusters - Οδικό δίκτυο 2

#οχημάτων	40		80		120		160		200	
Ταχύτητα οχημ.	ALM	DMAC	ALM	DMAC	ALM	DMAC	ALM	DMAC	ALM	DMAC
30 km/h	9.8	9.9	8.7	10.5	12.5	10.9	16	14.1	14.6	15.1
50 km/h	8.1	9.8	11.9	10.9	10.4	11.2	9.4	22.6	11.4	13
70 km/h	11.2	9.2	10.4	9.4	11.2	11.6	14.2	12.9	14.2	14

Πίνακας 8 Μέσο μέγεθος των Clusters - Οδικό δίκτυο 2

#οχημάτων	40		80		120		160		200	
Ταχύτητα οχημ.	ALM	DMAC	ALM	DMAC	ALM	DMAC	ALM	DMAC	ALM	DMAC
30 km/h	3.7	4.2	9.5	8.1	9.2	11.2	8.9	11.3	12.6	14.1
50 km/h	4.9	4.2	6.7	7.4	10.6	10.5	14.6	9	16.4	15.3
70 km/h	3.9	4.3	7.2	8	9.9	10.8	11.3	12.7	11.3	15.2

Παρατηρήσεις

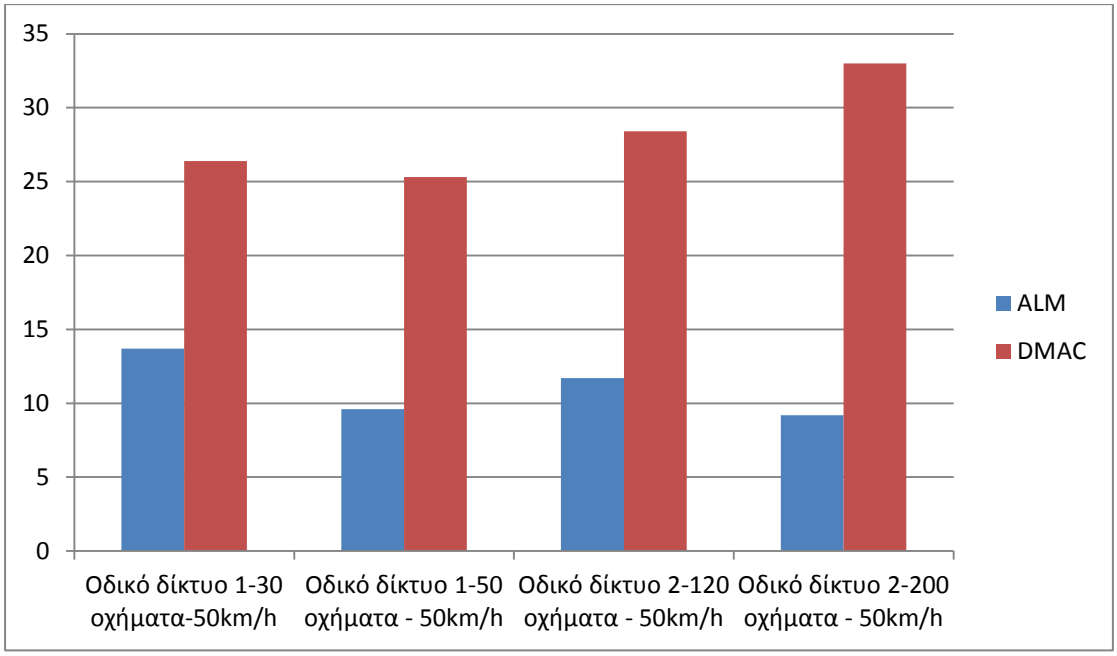
Οι επιδόσεις των αλγορίθμων είναι καλύτερες όσο μικρότερες είναι η ταχύτητες των οχημάτων και όσο μικρότερο το πλήθος των οχημάτων. Όταν το δίκτυο είναι ιδιαίτερα αραιό, δηλαδή στην περίπτωση που υπάρχουνε 10 ή 20 οχήματα στο οδικό δίκτυο 1 και 40 στο οδικό δίκτυο 2 ο DMAC έχει ελαφρός καλύτερες επιδόσεις από τον ALM. Εμφανίζει λιγότερες αλλαγές καταστάσεων και μεγαλύτερο μέσο χρόνο ζωής των Cluster-heads. Αυτό οφείλεται κυρίως στην απαίτηση του ALM να μη διατηρεί κόμβους cluster-heads οι οποίοι δεν διαθέτουνε κανένα γείτονα. Στον ALM όταν ένας κόμβος δε διαθέτει γείτονες ταλαντεύεται στις καταστάσεις CH και Undecided. Η συνεχή αλλαγή της κατάστασή των κόμβων αυτών είναι η αιτία των χαμηλών επιδόσεων σε τόσο αραιές τοπολογίες.

Όταν στο δίκτυο προστίθενται περισσότεροι κόμβοι και υπάρχει μια πιο πυκνή τοπολογία, αν και οι επιδόσεις των αλγορίθμων πέφτουν αισθητά, ο ALM δίνει καλύτερα αποτελέσματα σε σύγκριση με τον DMAC. Οι μέσες αλλαγές καταστάσεων των κόμβων είναι αρκετά μειωμένες σε σχέση με τον DMAC, πράγμα που οδηγεί στο συμπέρασμα ότι οι κόμβοι λειτουργούνε πιο σταθερά με χρήση του ALM, σε σύγκριση με τον DMAC. Επίσης η μέση διάρκεια ζωής των Cluster-heads είναι μεγαλύτερος για τον ALM.

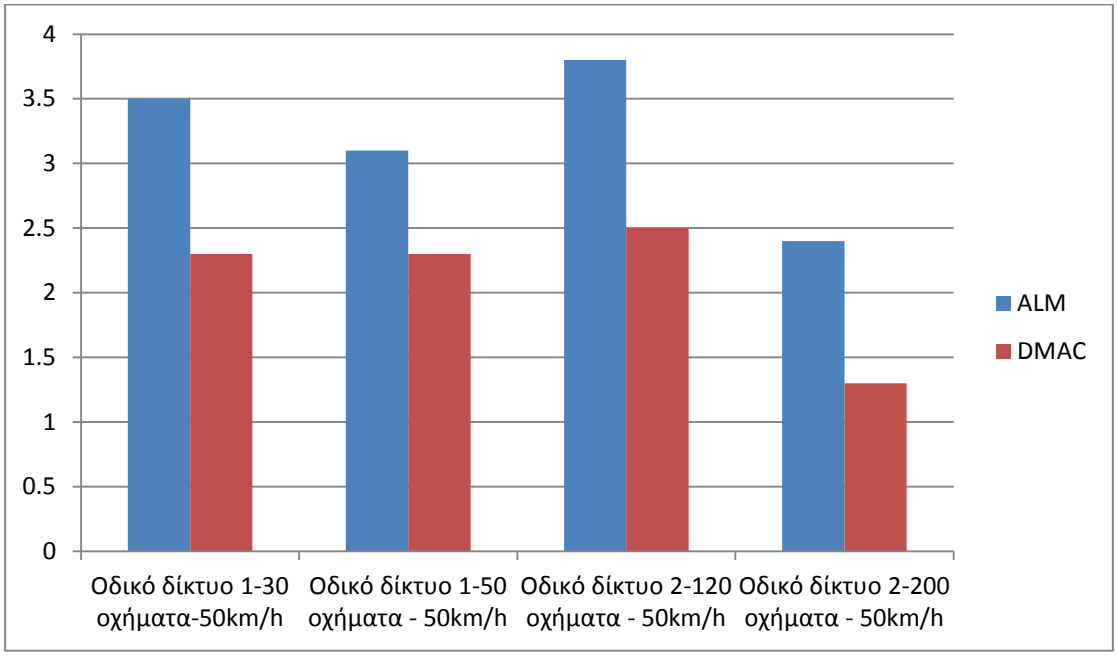
Μπορεί να παρατηρηθεί μια σταθερή απόδοση στον ALM σε αρκετά υψηλές ταχύτητες σε σχέση με τον DMAC. Ο μέσος αριθμός καταστάσεων και ο μέσος χρόνος ζωής στα cluster heads δεν παρουσιάζει μεγάλες διαφορές στα πυκνά δίκτυα στις ταχύτητες των 50km/h και 70km/h. Δηλαδή η λειτουργία του αλγορίθμου στο σχηματισμό clusters σε αυτό το περιβάλλον παρουσιάζει αντοχή στη συνεχή κίνηση των οχημάτων. Αντίθετα ο DMAC αλγόριθμος σε πυκνά δίκτυα για ταχύτητες 50km/h και 70km/h παρουσιάζει αισθητά χειρότερες επιδόσεις. Δείχνοντας την αδυναμία του αλγορίθμου σε πολύ δυναμικά περιβάλλοντα.

Γραφήματα

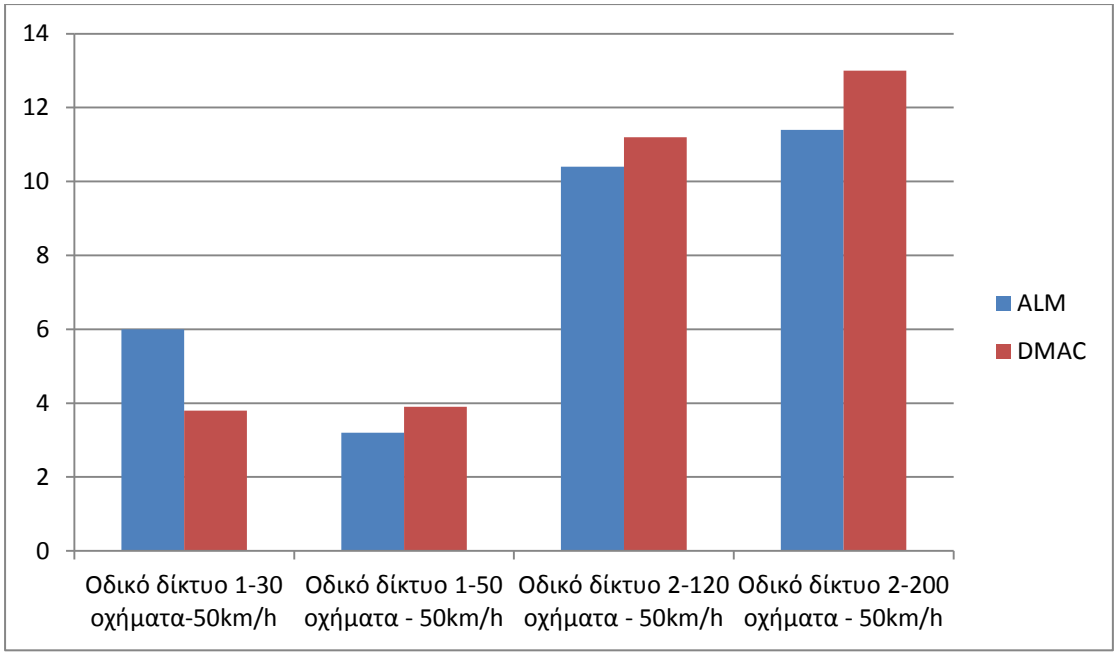
Στην συνέχεια παρουσιάζονται γραφήματα με τα αποτελέσματα για τις περιπτώσεις του οδικού δικτύου 1 με 30 και 50 οχήματα και του οδικού δικτύου 2 με 120 και 200 οχήματα, με ταχύτητες 50 km/h.



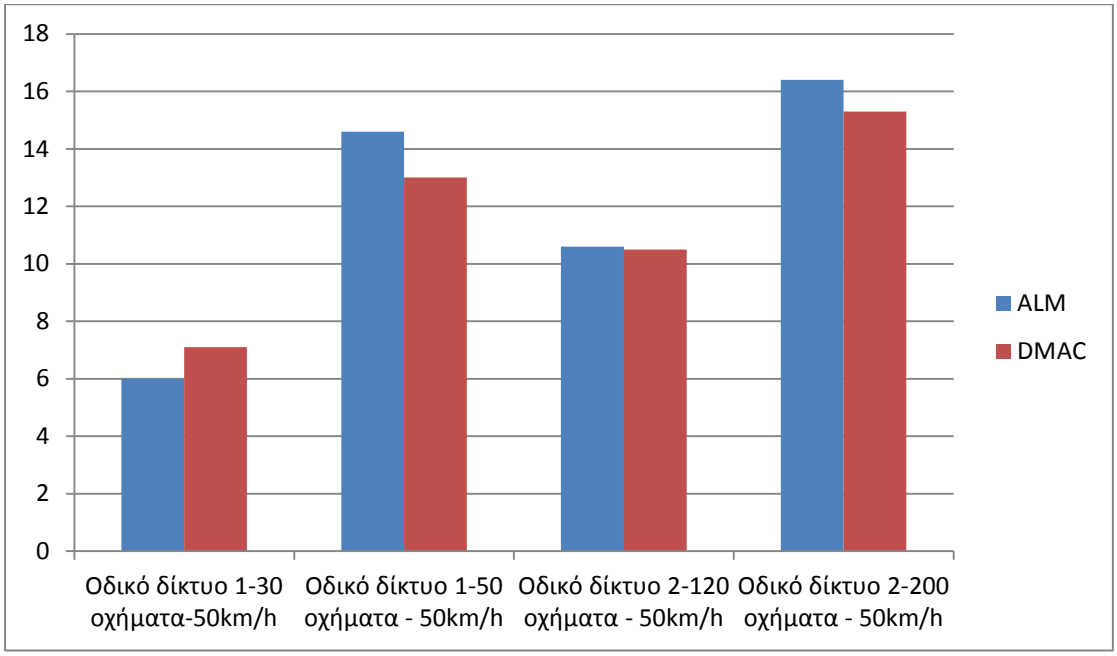
Εικόνα 12-Μέσος αριθμός αλλαγών καταστάσεων



Εικόνα 13-Μέσος χρόνος ζωής Cluster-head



Εικόνα 14-Μέσος αριθμός Clusters



Εικόνα 15-Μέσο μέγεθος Cluster

Αναφορές

1. E. Souza, I. Nikolaidis and P. Gburzynski, “A New Aggregate Local Mobility (ALM) Clustering Algorithm for VANETs”, IEEE Communications Society, IEEE ICC, 2010.
2. F. Li and Y. Wang, “Routing in vehicular ad hoc networks: A survey”, Vehicular Technology Magazine, IEEE, vol. 2, no. 2, pp. 12–22, June 2007.
3. P. Basu, N. Khan, and T. Little, “A mobility based metric for clustering in mobile ad hoc networks”, Distributed Computing Systems Workshop, 2001 International Conference on, pp. 413–418, Apr 2001.
4. S.Basagni, “Distributed Clustering for Ad Hoc Networks”, in Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms and Networks(I-SPAN’99), Fremantle, Australia, pp310-315, IEEE Computer Society, June 23-25, 1999.
5. Y. Gunter, B. Wiegel, and H. P. Grossmann, “Cluster-based medium access scheme for vanets”, Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE, pp. 343–348, 30 2007-Oct. 3 2007.
6. R. A. Santos, R. M.Edwards AMIEE, MIEEE and N. L. Seed, “Inter Vehicular Data Exchange Between Fast Moving Road Traffic Using an Ad-hoc Cluster-Based Location Routing Algorithm and 802.11b Direct Sequence Spread Spectrum Radio”, PGNet, 2003.
7. Michael Behrisch, Laura Bieker, Jakob Erdmann and Daniel Krajzewicz. SUMO - Simulation of Urban MObility: An Overview In: SIMUL 2011, The Third International Conference on Advances in System Simulation, 2011.
8. ns-3, http://www.nsnam.org.
9. P. Fan, J. G. Haran, J. Dillenburg, and P. C. Nelson, “Cluster-based framework in vehicular ad-hoc networks”, Lecture Notes in Computer Science, vol. 3738, pp. 32–42, October 2005.