



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Πολυτεχνική Σχολή Βόλου

**Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών
Τηλεπικοινωνιών και Δικτύων**

Πτυχιακή Εργασία

Θέμα

**Κατασκευή δικτύου «ραχοκοκκαλιάς» σε ad hoc δίκτυα
με κινητούς κόμβους.**

Φοιτήτρια

Αλεξίου Βασιλική

Καθηγητές

Επιβλέπων : Κατσαρός Δημήτριος

Συνεπιβλέπων : Τασιούλας Λέανδρος

~Βόλος, 2012~

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Εισαγωγικά.....	4
Τί είναι τα VANETs	4
Γενικά.....	4
Προκλήσεις.....	6
Εφαρμογές.....	7
Τεχνολογίες	8
Πρωτόκολλα Δρομολόγησης	10
Virtual Backbone σε VANETs	14
Τί είναι το VBB	14
Γενικά.....	14
Σύγκριση γνωστών αλγορίθμων	15
Σχεδιασμός ενός αλγορίθμου μετάδοσης, βασισμένος στον Lowest-ID αλγόριθμο	17
Ένας καταναμημένος αλγόριθμος δυναμικής ομαδοποίησης, με σκοπό την δημιουργίας backbone	20
Ο Δικός μας Αλγόριθμος.....	23
Περιγραφή.....	23
Παράδειγμα	28
Πειραματική Αποτίμηση	35
Συμπεράσματα	38
Μελλοντική Εργασία	38
Αναφορές.....	39

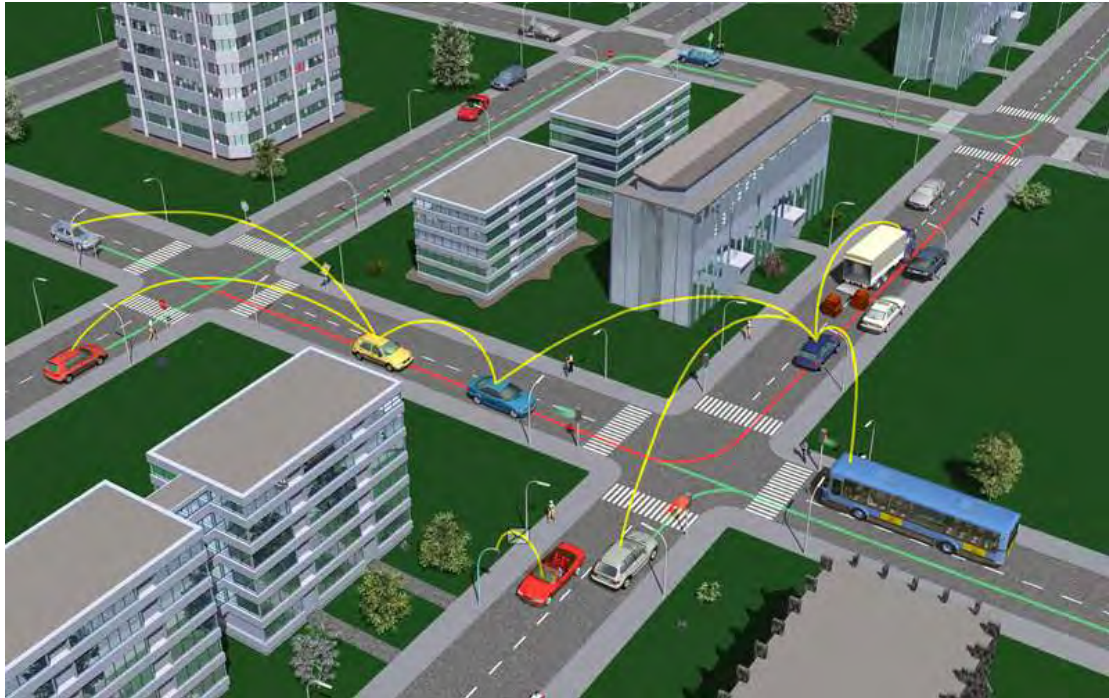
ΤΙ ΕΙΝΑΙ ΤΑ VANETS

ΓΕΝΙΚΑ

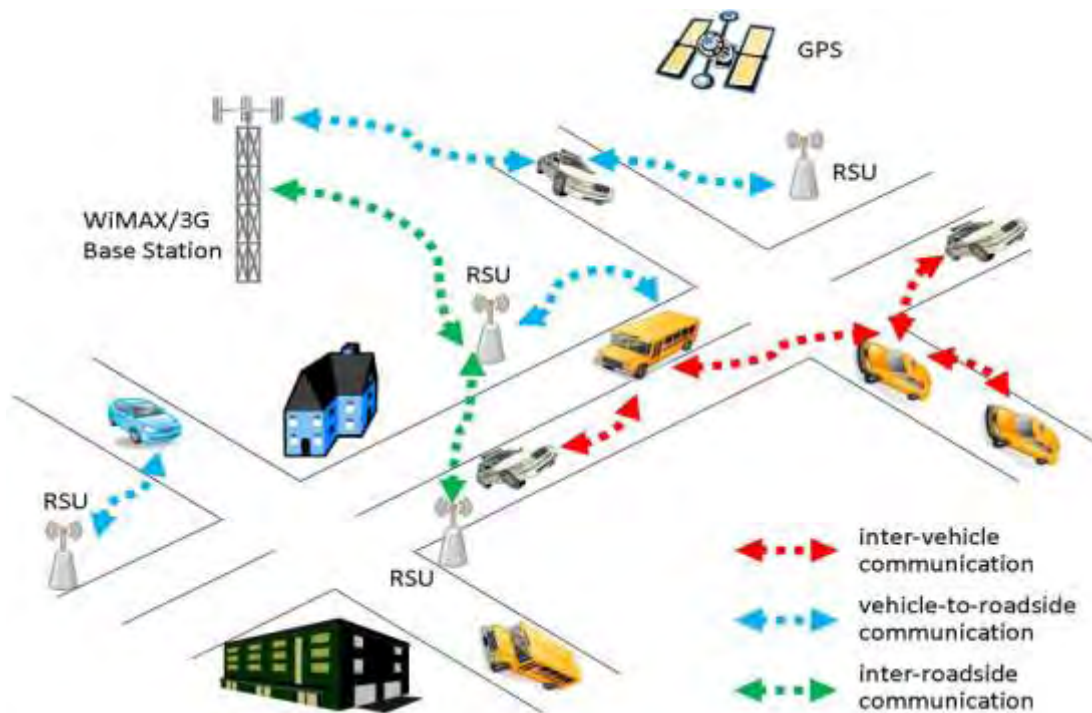
Ένα VANET(Vehicular Ad-Hoc Network), είναι ένα σύνολο από κινούμενα οχήματα(αυτοκίνητα). Κάθε ένα όχημα αντιπροσωπεύεται από έναν κόμβο και έτσι δημιουργείται ένα δίκτυο κινητών κόμβων. Το VANET μετατρέπει κάθε όχημα σε ένα ασύρματο δρομολογητή(router) ή κόμβο και επιτρέπει σε οχήματα που βρίσκονται μέσα στα όρια μιας προκαθορισμένης απόστασης το ένα από το άλλο(100 με 300 μέτρα περίπου) να είναι συνδεδεμένα. Τα οχήματα αυτά δημιουργούν τελικά ένα δίκτυο με ευρεία εμβέλεια. Με την ενσωμάτωση αισθητήρων στα οχήματα, εξασφαλίζεται η δυνατότητα επικοινωνίας μεταξύ τους. Στην περίπτωση που ένα όχημα βρεθεί εκτός της εμβέλειας του δικτύου, έχει στην πραγματικότητα πάψει να αποτελεί μέλος του κινητού δικτύου και δεν μπορεί να ανταλλάξει πλέον πληροφορία με τα οχήματα που εξακολουθούν να ανήκουν στο συγκεκριμένο δίκτυο. Άλλα οχήματα μπορεί να εισέλθουν και πληρώντας τις προκαθορισμένες αποστάσεις από τα υπόλοιπα οχήματα που βρίσκονται ήδη στο δίκτυο, μπορούν τελικά να γίνουν και τα ίδια μέλη του κινητού δικτύου. Πρόκειται όπως μπορούμε εύκολα να συμπεράνουμε για ένα δίκτυο κινούμενων κόμβων που δημιουργείται και αναπροσαρμόζεται δυναμικά.

Τα VANETS είναι αρκετά παρόμοια με τα MANETS(Mobile Ad-Hoc Networks). Τα MANETS αποτελούνται από κινητούς κόμβους που δεν έχουν καμία προκαθορισμένη υποδομή. Οι κόμβοι συνδέονται μεταξύ τους με έναν αποκεντρωμένο και αυτοοργανώμενο τρόπο και καθορίζουν multi-hop συνδέσεις μεταξύ τους. Εάν οι κινούμενοι κόμβοι είναι οχήματα, τότε μιλάμε για έναν τύπο δικτύου που καλείται πλέον VANET. Σημαντικό χαρακτηριστικό που ξεχωρίζει ένα VANET από ένα MANET είναι ότι σε ένα VANET τα οχήματα κινούνται με μεγαλύτερη μέση ταχύτητα και επίσης ο αριθμός των κόμβων(δηλαδή των κινούμενων οχημάτων) είναι πολύ μεγαλύτερος. Τα VANETS επιτρέπουν την επικοινωνία μεταξύ των οχημάτων. Τα δίκτυα οχημάτων αποτελούνται από οχήματα και Road Side Units(RSUs), τα οποία είναι τοποθετημένα σε διάφορα σημεία του οδικού δικτύου και μπορούμε να πούμε πως κατά κάποιο τρόπο διαχειρίζονται την πληροφορία που ανταλλάσσεται

μεταξύ των οχημάτων. Η εικόνα ενός τυπικού VANET καθώς και τα Road Side Units φαίνονται στις επόμενες εικόνες.



ΕΠΙΚΟΙΝΩΝΙΑ ΜΕΤΑΞΥ ΤΩΝ ΟΧΗΜΑΤΩΝ ΕΝΟΣ VANET



ΕΠΙΚΟΙΝΩΝΙΑ ΟΧΗΜΑΤΩΝ ΚΑΙ ΜΕ ROAD SIDE UNITS

ΠΡΟΚΛΗΣΕΙΣ

Τα VANETs έχουν κάποιες μοναδικές προκλήσεις να αντιμετωπίσουν οι οποίες επηρεάζουν σε σημαντικό βαθμό το σχεδιασμό του συστήματος επικοινωνίας και το σχεδιασμό του πρωτοκόλλου ασφάλειας. Μερικές από τις προκλήσεις που έχουμε να αντιμετωπίσουμε κατά το σχεδιασμό ενός VANET είναι οι ακόλουθες :

1. Πιθανός πολύ μεγάλος αριθμός κόμβων, δηλαδή οχημάτων.

Ένα VANET πρέπει να δύναται να ανταπεξέλθει σε μεγάλο αριθμό οχημάτων που έχουν εφοδιαστεί με εξοπλισμό που τους παρέχει τη δυνατότητα ανταλλαγής πληροφορίας με τα υπόλοιπα οχήματα αλλά και με τα Road Side Units.

2. Μεγάλη κινητικότητα και συχνή αλλαγή τοπολογίας.

Τα οχήματα πιθανόν να κινούνται με μεγάλη ταχύτητα. Έτσι, όταν ένα όχημα προσπερνά ένα άλλο και αναπτύσσει μεγάλη ταχύτητα, ο χρόνος που υπάρχει για την ανταλλαγή πληροφορίας μεταξύ των οχημάτων είναι πάρα πολύ μικρός. Επιπρόσθετα, οι ενδιάμεσοι κόμβοι σε ένα ασύρματο δίκτυο, που στην ουσία λειτουργούν ως προωθητές των πακέτων πληροφορίας μεταξύ των οχημάτων, αλλάζουν πάρα πολύ συχνά.

3. Υψηλές απαιτήσεις στη μεταφορά των δεδομένων σε πραγματικό χρόνο.

Η μεταφορά των δεδομένων πρέπει να γίνεται άμεσα και σε πραγματικό χρόνο. Κάτι τέτοιο γίνεται ιδιαίτερα εμφανές το πόσο σημαντικό είναι αν αναλογιστούμε περιπτώσεις που αφορούν την αποφυγή κάποιου ατυχήματος. Οι εφαρμογές που έχουν να κάνουν με την ανθρώπινη ζωή πρέπει να είναι real-time και αξιόπιστες όσον αφορά την μετάδοση της πληροφορίας προκειμένου να κληθεί άμεσα βοήθεια.

4. Δεν υπάρχει εμπιστευτικότητα στην πληροφορία/στα δεδομένα που ανταλλάσσουν τα οχήματα.

Η πληροφορία που υπάρχει σε ένα μήνυμα αφορά όλους τους χρήστες του δρόμου, οπότε δεν υπάρχει λόγος εμπιστευτικότητας.

5. Πρέπει να εξασφαλίζεται η ιδιωτικότητα των χρηστών των οχημάτων.

Η δυνατότητα επικοινωνίας μεταξύ των οχημάτων είναι πολύ πιθανό να αποκαλύψει πληροφορίες για τον οδηγό του οχήματος, όπως για παράδειγμα την ταυτότητά του, την ταχύτητα με την οποία κινείται, τη θέση του ανά πάσα στιγμή κλπ. Πρέπει με κάποιο τρόπο να εξασφαλιστεί η ιδιωτικότητα των χρηστών/οδηγών και ιδιαίτερα ο μη εντοπισμός της θέσης του οχήματος ανά πάσα στιγμή από τον οποιοδήποτε και σαφώς η ανωνυμία του οδηγού.

ΕΦΑΡΜΟΓΕΣ

Τα τελευταία χρόνια, η δικτύωση οχημάτων έχει κεντρίσει ιδιαίτερα το ενδιαφέρον της βιομηχανίας αλλά και της ακαδημαϊκής έρευνας. Απώτερος στόχος είναι η σημαντική βελτίωση της αποδοτικότητας αλλά και της ασφάλειας των μετακινήσεων. Όντας πλέον η τεχνολογία ασύρματων δικτύων διάχυτη και φθηνή, διάφορες καινοτόμες εφαρμογές που αφορούν τα οχήματα συζητώνται.

Έχουμε τέσσερις βασικές κατηγορίες εφαρμογών στο πλαίσιο των VANETs.

1. Εφαρμογές που αφορούν την ασφάλεια.

Έχουν ως σκοπό την ενδυνάμωση της ασφάλειας των οχημάτων και κατά συνέπεια των χρηστών αυτών. Αυτό δύναται να επιτευχθεί παρέχοντας πληροφορία που διαμοιράζεται μεταξύ των οχημάτων και έχει να κάνει με την αποφυγή μιας ενδεχόμενης σύγκρουσης, την ενημέρωση για την κατάσταση του οδοστρώματος, την κλήση άμεσης βοήθειας εάν αυτό κριθεί απαραίτητο κλπ.. Στις εφαρμογές αυτού του τύπου, έμφαση δίνεται στην έγκαιρη διάδοση των προειδοποιήσεων στα κοντινά οχήματα, ώστε τελικά να εξασφαλιστεί η ασφάλεια αυτών. Στην εικόνα παρακάτω φαίνεται πως λειτουργεί το σύστημα στην περίπτωση αποφυγής νέας σύγκρουσης.

2. Εφαρμογές που αφορούν τη διαχείριση της κίνησης.

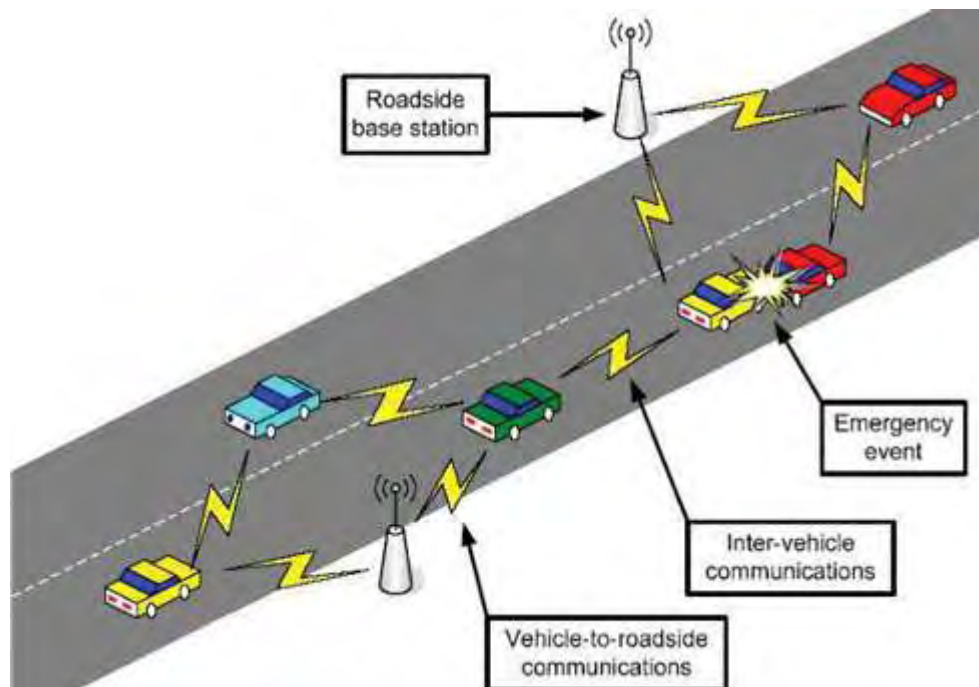
Έχει να κάνει με την ανταλλαγή πληροφορίας σε πραγματικό χρόνο μεταξύ των οχημάτων που αφορά την κίνηση που υπάρχει τη δεδομένη στιγμή στον δρόμο.

3. Εμπορικές εφαρμογές.

Πρόκειται για εφαρμογές που χρησιμοποιούνται για διαφημίσεις διαφόρων προϊόντων ή και υπηρεσιών.

4. Γενικές εφαρμογές.

Πρόκειται για εφαρμογές που επιτρέπουν τη συνδεσιμότητα στο Internet. Η δυνατότητα πρόσβασης στα emails, η περιήγηση στο web, το streaming (ροή) ήχου και video είναι μερικές από τις εφαρμογές που αφορούν αυτήν την κατηγορία και μπορούν να μετατρέψουν πιθανόν χαμένο χρόνο, πχ λόγω κίνησης, σε παραγωγικό και πιο ευχάριστο για τον χρήστη χρόνο. Απαιτείται σταθερή συνδεσιμότητα με υψηλό εύρος ζώνης.



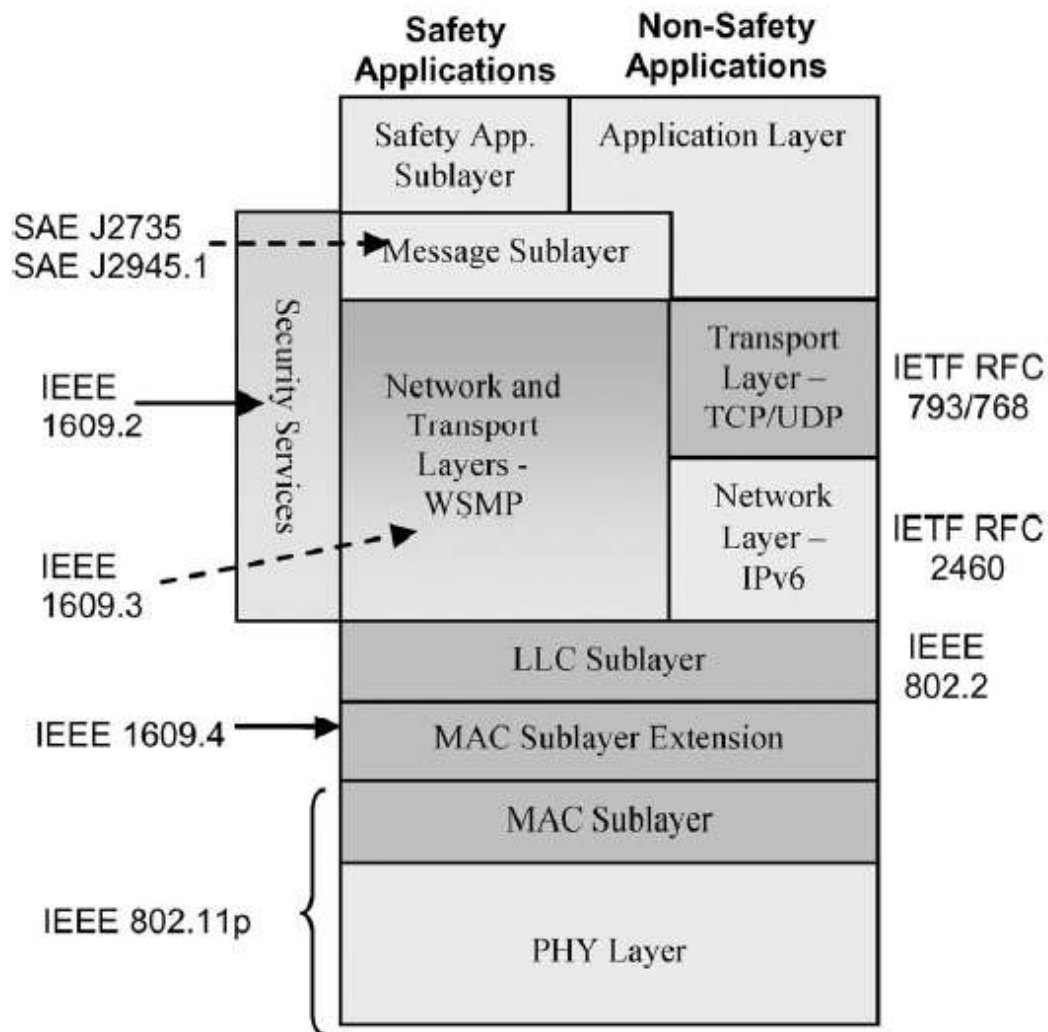
ΠΕΡΙΠΤΩΣΗ ΑΠΟΦΥΓΗΣ ΝΕΑΣ ΣΥΓΚΡΟΥΣΗΣ

ΤΕΧΝΟΛΟΓΙΕΣ

Τα VANETs ορίζουν έναν έξυπνο τρόπο με τον οποίο μπορούμε να επωφεληθούμε της δικτύωσης των οχημάτων. Μπορούμε να χρησιμοποιήσουμε για το σκοπό αυτό διάφορες ad-hoc τεχνολογίες δικτύωσης, όπως για παράδειγμα WiFi IEEE 802.11p, WAVE IEEE 1609, WiMAX

IEEE 802.16, Bluetooth, IRA, ZigBee. Οι τεχνολογίες αυτές μπορούν να εξασφαλίσουν εύκολη, συνεπή, αποδοτική και απλή επικοινωνία μεταξύ των κινούμενων οχημάτων, δυναμικά. Η πιο συχνά χρησιμοποιούμενη στην πράξη είναι η WiFi IEEE 802.11p. Τον τελευταίο καιρό γίνονται σοβαρές προσπάθειες για την ολοκληρωμένη ανάπτυξη της τεχνολογίας DSRC(Dedicated Short-Range Communication). Η αποδοτικότητα της τεχνολογίας αυτής εξαρτάται σημαντικά από τα πρότυπα συνεργασίας για διαλειτουργικότητα.

Στην DSRC τεχνολογία έχουν δεσμευτεί 75MHz του αδειοδοτημένου φάσματος στην μπάντα των 5.9 GHz για DSRC επικοινωνία. Το φάσμα αυτό διαιρείται σε διάφορα κανάλια. Τα V2V(vehicle to vehicle) μηνύματα ασφαλείας αναμένεται να ανταλλάσσονται στο κανάλι 172. Ένα συγκεκριμένο κανάλι έχει οριστεί για ασφάλεια. Επίσης, η επικοινωνία λαμβάνει χώρα σε μία εμβέλεια της τάξεως των εκατοντάδων μέτρων, μία απόσταση σαφώς μικρότερη από αυτή που καλύπτουν οι κινητές ή οι WiMax υπηρεσίες. Στη συνέχεια παρουσιάζουμε εν συντομία την στοίβα των πρωτοκόλλων της DSRC επικοινωνίας.



ΣΤΟΙΒΑ ΤΩΝ ΠΡΩΤΟΚΟΛΛΩΝ ΤΗΣ DSRC ΕΠΙΚΟΙΝΩΝΙΑΣ

Στα κατώτερα επίπεδα PHY και MAC, παρατηρούμε πως η DSRC χρησιμοποιεί IEEE 802.11p WAVE(Wireless Access for Vehicular Environments), μια τροποποίηση του κλασσικού 802.11 (WiFi). Το επίπεδο PHY διαιρείται σε δύο υπο-επίπεδα. Το Physical Medium Dependent, PMD και το Physical Layer Convergence Procedure, PLCP. Χρησιμοποιεί ορθογώνια πολυπλεξία(OFDM), όπως και το 802.11 πρωτόκολλο.

Στα μέσα της στοίβας, χρησιμοποιεί μία σειρά πρωτοκόλλων που ορίζονται από την IEEE 1609 Working Group. Συγκεκριμένα, χρησιμοποιεί το 1609.4 για την εναλλαγή των καναλιών, το 1609.3 για υπηρεσίες δικτύου(συμπεριλαμβάνοντας και το WAVE Short Message Protocol, WSMP) και το 1906.2 για υπηρεσίες ασφάλειας. Το DSRC υποστηρίζει επίσης τη χρήση γνωστών Internet πρωτοκόλλων, όπως το IPv6, το UDP και το TCP. Αυτά τα πρωτόκολλα ορίστηκαν από την Internet Engineering Task Form, IETF και είναι σταθερά. Το αν θα επιλεγεί WSMP ή IPv6 & UDP/TCP εξαρτάται από τις απαιτήσεις της εκάστοτε εφαρμογής.

Τέλος, στην κορυφή της στοίβας η SAE J2735 Message Set Dictionary, ορίζει ένα σύνολο πρότυπων μηνυμάτων που υποστηρίζουν κάποιες εφαρμογές για οχήματα.

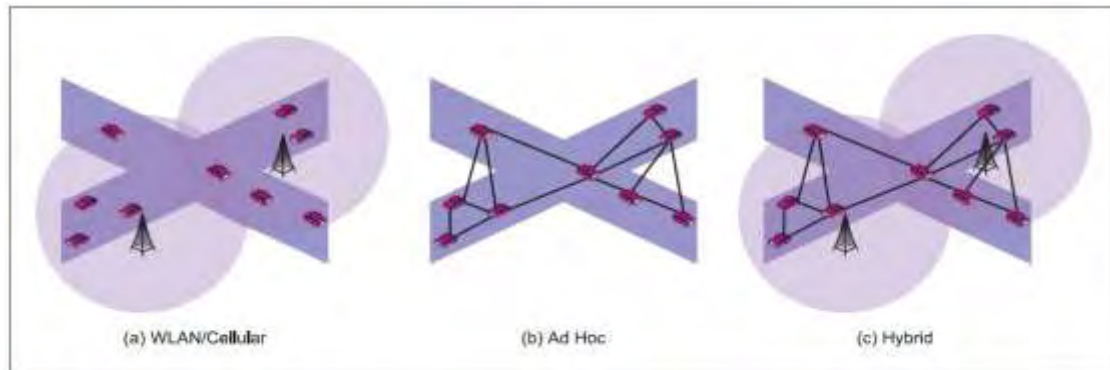
Στο άρθρο [1] παρουσιάζονται αναλυτικά οι πτυχές της DSRC τεχνολογίας και το κάθε επίπεδο λειτουργίας αναλύεται λεπτομερώς.

ΠΡΩΤΟΚΟΛΛΑ ΔΡΟΜΟΛΟΓΗΣΗΣ

Τα VANETs είναι στην ουσία ασύρματα δίκτυα που χρησιμοποιούν multi-hop συνδέσεις μεταξύ των κόμβων/οχημάτων, σε αντίθεση με τα δίκτυα που έχουν στατική δομή, με απώτερο σκοπό τη συνδεσιμότητα του δικτύου. Έχουν αναπτυχθεί εφαρμογές που αφορούν το στρατό, αλλά και άλλες που αφορούν την καθημερινότητα των απλών πολιτών. Η τοπολογία όμως των VANETs αλλάζει με το χρόνο. Για το λόγο αυτό είναι έντονη η ανάγκη για την ανάπτυξη νέων πρωτοκόλλων δρομολόγησης, καθώς τα ήδη υπάρχοντα πιθανόν να μην είναι πλέον κατάλληλα. Έτσι, γίνεται μία προσπάθεια σχεδιασμού νέων πρωτοκόλλων ή επέκτασης και αναπροσαρμογής των ήδη υπαρχόντων.

Ο σχεδιασμός ενός αποδοτικού πρωτοκόλλου δρομολόγησης για VANETs είναι πολύ κρίσιμος. Ο βασικός στόχος ενός πρωτοκόλλου είναι να επιτυγχάνει τον ελάχιστο δυνατό χρόνο επικοινωνίας και ταυτόχρονα να κάνει χρήση όσο το

δυνατόν λιγότερων πόρων του δικτύου. Πολλά από τα πρωτόκολλα δρομολόγησης που φτιάχτηκαν για εφαρμογή σε MANETs μπορούν να εφαρμοστούν κατευθείαν σε VANETs. Τα περισσότερα όμως θέλουν συνήθως αναπροσαρμογές καθώς επηρεάζονται από τη γρήγορη κίνηση των οχημάτων, από την δυναμική ανταλλαγή πληροφορίας και τη σχετική υψηλή ταχύτητα των οχημάτων σε σχέση με τα MANETs.



ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ ΔΙΚΤΥΟΥ ΣΕ VANETS

Γενικά, υφίστανται τρεις κατηγορίες αρχιτεκτονικών δικτύου σε VANETs:

(a) WLAN

(b) AD HOC

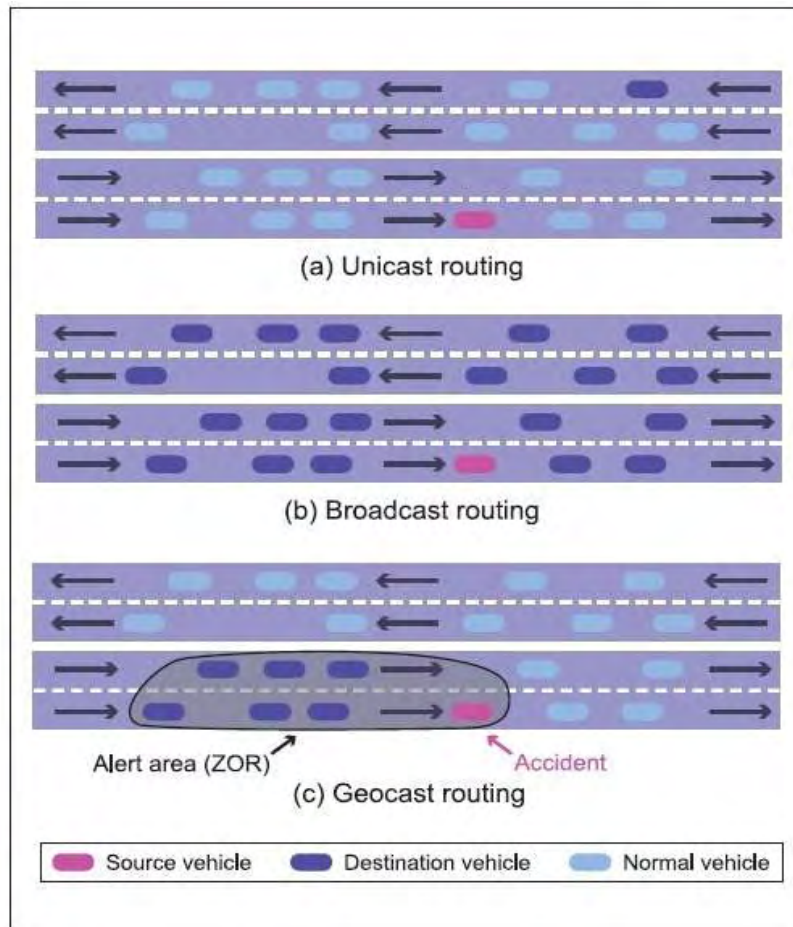
(c) HYBRID (συνδυασμός των δύο πρώτων), οι οποίες φαίνονται και στην παραπάνω εικόνα.

Αναλόγως την εκάστοτε περίπτωση, τα VANETs μπορούν να διακριθούν από τα υπόλοιπα ad hoc δίκτυα, λαμβάνοντας υπόψιν διάφορα χαρακτηριστικά των οχημάτων που τα απαρτίζουν. Έτσι, μπορούμε να έχουμε διάκριση των VANETs από τα υπόλοιπα ad hoc δίκτυα με βάση τα ακόλουθα κριτήρια :

- Δυναμική τοπολογία
- Πιθανή συχνή αποσύνδεση από το δίκτυο
- Επαρκής ενέργεια και αποθήκευση
- Επικοινωνία βάση γεωγραφίας
- Μοντελοποίηση κινητικότητας και κατηγοριοποίηση
- Ποικίλα περιβάλλοντα επικοινωνίας
- Ισχυροί περιορισμοί καθυστέρησης
- Αλληλεπίδραση με On-board αισθητήρες

Εξαιτίας της δυναμικής φύσης των κινητών κόμβων σε ένα δίκτυο, το να βρούμε και να διατηρήσουμε διαδρομές(routes) μεταξύ των κόμβων αποτελεί μεγάλη πρόκληση στα VANETs. Όσο αφορά τη δρομολόγηση σε VANETs με ad hoc δικτύωση, έχουν μελετηθεί και προταθεί πολλά διαφορετικά πρωτόκολλα, τα οποία τα κατατάσσουμε σε πέντε βασικές κατηγορίες : ad hoc, position-based, clusterbased, broadcast και geocast δρομολόγηση.

Στην εικόνα που ακολουθεί παρουσιάζονται ενδεικτικά μερικά σενάρια δρομολόγησης.



ΣΕΝΑΡΙΑ ΔΡΟΜΟΛΟΓΗΣΗΣ

Συμπερασματικά, μπορούμε να σημειώσουμε πως τα position-based και τα geocasting πρωτόκολλα είναι τα πιο ενδεδειγμένα για VANETs καθώς προσφέρουν αρκετούς περιορισμούς μιας και λαμβάνουν υπόψιν την γεωγραφία του δικτύου. Ωστόσο, η απόδοση ενός πρωτοκόλλου δρομολόγησης, εξαρτάται σημαντικά από το μοντέλο κινητικότητας, το περιβάλλον οδήγησης, την πυκνότητα των οχημάτων και από άλλες σημαντικές μετρικές. Στη συνέχεια παρουσιάζεται ένας συγκεντρωτικός πίνακας στον οποίο συνοψίζονται τα χαρακτηριστικά των πρωτοκόλλων δρομολόγησης και πώς αυτά εκτιμώνται, για να μας δώσει, κλείνοντας αυτήν

την ενότητα, μία σφαιρική εικόνα των διαφόρων πρωτοκόλλων δρομολόγησης και των δυνατοτήτων τους.

<i>Routing Protocols</i>	<i>Routing Type</i>	<i>Position Information? (How to Use)</i>	<i>Hierarchical Structure?</i>	<i>Network Simulator</i>	<i>Simulation Scenario</i>
AODV	Unicast	No	No	—	—
DSR	Unicast	No	No	—	—
GPSR	Unicast	Packet Forwarding	No	—	—
PRAODV/PRAODV-M [5]	Unicast	Route-Selection (lifetime prediction)	No	NS2	Simple highway model (20km segment only)
AODV-bis [12]	Unicast	Route-Req-Forwarding	No	—	—
GSR [16]	Unicast	Packet Forwarding	No	NS2	Real city model (from map)
GPCR [19]	Unicast	Packet Forwarding	No	NS2	Real city model (from map)
A-STAR [8]	Unicast	Packet Forwarding (also use traffic info.)	No	NS2	Grid city model
COIN [23]	Unicast	Cluster Formation	Yes	Own	Real highway model
LORA_CBF [10]	Unicast	Packet Forwarding (also location prediction)	Yes	OPNET	Simple circle and square road
Flooding	Broadcast	No	No	—	—
UMB [26]	Broadcast	Packet Forwarding	No	Own	Simple intersection road
V-TRADE/HV-TRADE [27]	Broadcast	Classify Forwarding Group	No	Own	Simple intersection
BROADCOMM [25]	Broadcast	Formation of Cells	Yes	Own	Simple highway model (15 nodes only)
Msg Dis Protcl [14]	Geocast	Packet Forwarding	No	Own	Simple highway model (10 km long)
IVG [30]	Geocast	Packet Forwarding	No	Glomosim	Simple highway model (10 km long, 100/200 nodes)
Cached Geocast [31]	Geocast	Packet Forwarding	No	NS2	Quadratic network (size from 1 km to 4km, 100 nodes)
Abiding Geocast [32]	Geocast	Packet Forwarding	No	—	—

ΣΥΓΚΕΝΤΡΩΤΙΚΟΣ ΠΙΝΑΚΑΣ ΠΡΩΤΟΚΟΛΛΩΝ ΔΡΟΜΟΛΟΓΗΣΗΣ

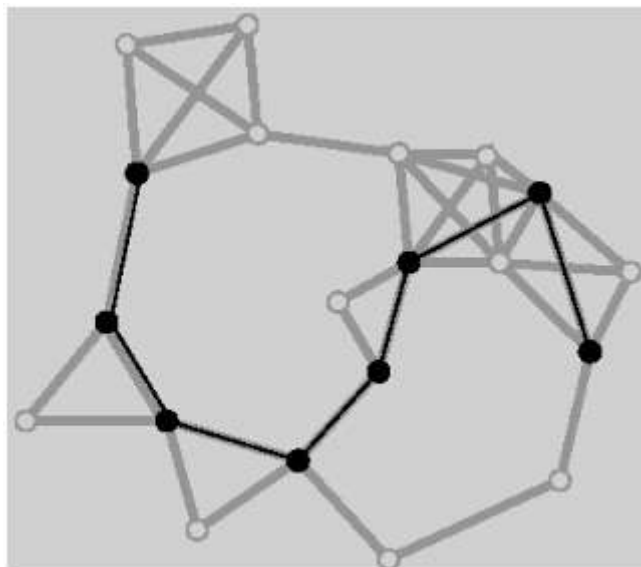
VIRTUAL BACKBONE ΣΕ VANETS

ΤΙ ΕΙΝΑΙ ΤΟ VBB

ΓΕΝΙΚΑ

Στην επικοινωνία, που αφορά την ανταλλαγή δεδομένων μεταξύ των κόμβων, ένα backbone(ραχοκοκαλιά) είναι το μεγαλύτερο μονοπάτι ή κανάλι ενός δικτύου το οποίο εξυπηρετεί την μεγαλύτερη κίνηση δεδομένων, με τη μεγαλύτερη δυνατή ταχύτητα, και το οποίο ενώνει τον κάθε βασικό server και την κάθε βασική συσκευή στο δίκτυο. Το μέγεθος(εύρος) ενός backbone είναι σχετικό. Σε μικρά δίκτυα κόμβων το backbone θα είναι μικρό και μικρότερο από το όλο αρχικό δίκτυο.

Στην ουσία ένα backbone είναι ένα Connected Dominating Set. Όπως γνωρίζουμε, ένα Connected Dominating Set(CDS) ενός δικτύου ή ενός γραφήματος G , είναι ένα dominating set του οποίου το επαγόμενο(induced) γράφημα είναι συνδεδεμένο. Στην εικόνα παρακάτω παρουσιάζεται με τη σκούρα γραμμή ένα παράδειγμα ενός CDS.



ΠΑΡΑΔΕΙΓΜΑ ΕΝΟΣ CDS

Η εύρεση backbone σε ένα δίκτυο κόμβων έχει προταθεί με σκοπό να διευκολύνει την δρομολόγηση, την αναμετάδοση της πληροφορίας και την εγκαθίδρυση μιας δυναμικής υποδομής για βάσεις δεδομένων σε VANETs. Ελαχιστοποιώντας το CDS, έχουμε απλοποίηση της τοπολογίας του VANET και μας παρέχεται η δυνατότητα να χρησιμοποιήσουμε συντομότερες διαδρομές για την αναμετάδοση της πληροφορίας.

Καθώς το ενδιαφέρον για τα VANETs έχει ενταθεί ιδιαίτερα τα τελευταία χρόνια προκαλώντας όλο και μεγαλύτερο ενδιαφέρον στην επιστημονική και ερευνητική κοινότητα, διάφορες προσπάθειες εύρεσης/κατασκευής του backbone ενός δεδομένου γραφήματος/δικτύου έχουν πραγματοποιηθεί. Ενδεικτικά, εντοπίσαμε και αναφέρουμε παρακάτω κάποια επιστημονικά άρθρα που κινήθηκαν προς αυτή την κατεύθυνση, δηλαδή την εύρεση backbone. Επέλεξαν με κάποιο τρόπο ποιοι από τους κόμβους του δικτύου και ποιες ακμές αυτών θα απαρτίζουν το μονοπάτι μέσω του οποίου θα γίνεται η προώθηση των πακέτων πληροφορίας. Σκοπός είναι όλοι οι κόμβοι του αρχικού δικτύου να αποκτούν τελικά την επίμαχη πληροφορία και να μπορούν εν τέλει να τη χρησιμοποιήσουν κατά το δοκούν. Το πώς θα τη χρησιμοποιήσουν ποικίλει ανάλογα με το είδος της εφαρμογής που υποστηρίζουμε και επιθυμούμε να εξυπηρετήσουμε μέσω του VANET.

ΣΥΓΚΡΙΣΗ ΓΝΩΣΤΩΝ ΑΛΓΟΡΙΘΜΩΝ

Όπως έχουμε αναφέρει, τα vehicular ad hoc networks, είναι μία ειδική μορφή των Mobile ad hoc networks. Σαν σκοπό τους έχουν να προσφέρουν επικοινωνία μεταξύ των οχημάτων που κινούνται σε ένα οδικό δίκτυο. Αξίζει να σημειωθεί πως η κίνηση των οχημάτων είναι κατά έναν τρόπο προβλέψιμη, αφού περιορίζεται σημαντικά από την κατεύθυνση κίνησης και την τροχιά του οδικού δικτύου. Στις εφαρμογές για VANETs, το broadcasting(η μετάδοση δηλαδή των πακέτων) είναι μία ουσιαστική διαδικασία κατά την οποία ένας κινητός κόμβος στέλνει πακέτα σε άλλους γειτονικούς του κόμβους.

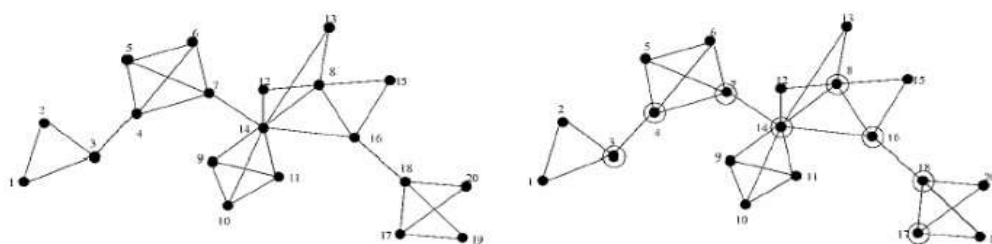
Ο πρώτος τρόπος που μας έρχεται στο μυαλό και μπορεί να επιτευχθεί το broadcasting είναι η γνωστή πλημμύρα(ή αλλιώς flooding). Αποτελεί έναν απλό και ευνόητο τρόπο, ο οποίος ωστόσο δεν είναι και τόσο αποδοτικός. Ουσιαστικά αποτελεί τροχοπέδη για την επεκτασιμότητα του δικτύου μας. Πιο συγκεκριμένα, καθώς το δίκτυο μας μεγαλώνει και γίνεται πιο πυκνό, ακόμη και μία εκπομπή από ένα κόμβο μπορεί να προκαλέσει πολλές συγκρούσεις, πολλά διπλά πακέτα και ανταγωνισμό, με αποτέλεσμα τελικά να καταρρεύσει το δίκτυο. Όλοι θα προσπαθούν να στείλουν σε όλους τους γείτονές τους, με αποτέλεσμα ένας κόμβος να δεχτεί και να στείλει με τη σειρά του το ίδιο πακέτο

πληροφορίας πολλές φορές. Πρόκειται δηλαδή για έναν εύκολο αλλά όχι και τόσο καλό από πλευράς απόδοσης τρόπο.

Στην προσπάθεια επίτευξης της επικοινωνίας μεταξύ των οχημάτων με πιο αποδοτικό τρόπο, έχει προταθεί η εύρεση ενός virtual backbone των οχημάτων του δικτύου. Αυτό θα συνεισφέρει ιδιαίτερα στο να έχουμε εν τέλει μια καλύτερη δρομολόγηση και μια πιο αποδοτική μετάδοση πακέτων πληροφορίας μεταξύ των οχημάτων. Όπως αναφέραμε και παραπάνω, ένα backbone είναι στην ουσία ένα connected dominating set (CDS) των κόμβων που απαρτίζουν το δίκτυο. Ελαχιστοποιώντας τον αριθμό των κόμβων που απαρτίζουν το CDS, απλοποιείται η τοπολογία του VANET και επιπλέον προκύπτουν πιο σύντομες διαδρομές τις οποίες ακολουθεί ένα πακέτο πληροφορίας για να ταξιδέψει από έναν κόμβο/όχημα σε έναν άλλο.

Το να υπολογίσεις το minimum connected dominating set (MCDS) είναι ένα πρόβλημα NP-complete. Επομένως, διάφοροι προσεγγιστικοί και ευριστικοί αλγόριθμοι μπορούν να εφαρμοστούν σε ένα δίκτυο κόμβων και να μας δώσουν σαν αποτέλεσμα ένα MCDS αυτών, δηλαδή ένα backbone του δικτύου. Μερικοί αντιπροσωπευτικοί localized αλγόριθμοι που υπολογίζουν το MCDS ενός δικτύου κόμβων είναι ο αλγόριθμος των Wu και Li, του Stojmenovic, ο MPR (multipoint relays) αλγόριθμος και ο AWF αλγόριθμος.

Υλικό για τη θεωρητική ανάλυση της απόδοσης αυτών των αλγορίθμων υπάρχει αρκετό στη βιβλιογραφία. Ωστόσο, μία ανάλυση βασισμένη σε πραγματικό περιβάλλον ή τουλάχιστον σε περιβάλλον προσομοίωσης, θα ήταν πολύ πιο αντιπροσωπευτική και ουσιαστική. Το άρθρο [7], περιέχει συγκρίσεις μεταξύ των παραπάνω αλγορίθμων και παρουσιάζει σε γραφήματα την απόδοση του καθενός σε σχέση με τους υπόλοιπους χρησιμοποιώντας διαφορετικά κριτήρια κάθε φορά.



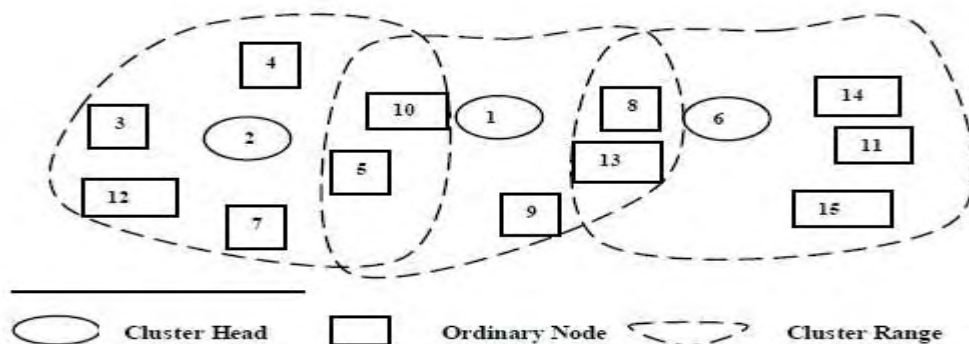
ΕΥΡΕΣΗ MCDS ΜΕ ΤΟΝ ΑΛΓΟΡΥΘΜΟ ΤΩΝ WU & LI. ΟΙ ΚΥΚΛΩΜΕΝΟΙ ΚΟΜΒΟΙ ΑΠΟΤΕΛΟΥΝ ΤΟ BACKBONE ΤΟΥ ΑΡΧΙΚΟΥ ΔΙΚΤΥΟΥ

Καθώς όμως το ενδιαφέρον για τα VANETs έχει γίνει όλο και πιο έντονο τα τελευταία χρόνια, διάφοροι νέοι αλγόριθμοι για το broadcasting των πακέτων μεταξύ των κόμβων έχουν δημιουργηθεί. Στη συνέχεια περιγράφονται ενδεικτικά δύο τέτοιοι αλγόριθμοι, οι οποίοι βασίζονται στην εύρεση του backbone του δικτύου.

ΣΧΕΔΙΑΣΜΟΣ ΕΝΟΣ ΑΛΓΟΡΙΘΜΟΥ ΜΕΤΑΔΟΣΗΣ, ΒΑΣΙΣΜΕΝΟΣ ΣΤΟΝ LOWEST-ID ΑΛΓΟΡΙΘΜΟ

Αρχικά, θα παρουσιάσουμε έναν αλγόριθμο κατασκευής backbone(ραχοκοκαλιάς) βασισμένο στην ομαδοποίηση(clustering) των κόμβων του δικτύου. Εφαρμόζοντας αυτόν τον αλγόριθμο, κάθε κόμβος του δικτύου θα περιέλθει τελικά σε μία από τις τρεις καταστάσεις : clusterhead CH, member node και gateway. Έπειτα, θα αναζητήσουμε το μικρότερο δυνατό σύνολο κόμβων από τους CHs και τους gateways, οι οποίοι τελικά θα συμμετέχουν στην αναμετάδοση των πακέτων πληροφορίας.

Ο αλγόριθμος Lowest-ID χρησιμοποιείται συχνά στη δημιουργία ομάδων κόμβων ενός δικτύου. Κάθε κόμβος έχει εξ αρχής ένα διακριτό αναγνωριστικό, ID. Κάθε κόμβος στέλνει περιοδικά ένα μήνυμα "HELLO" σε όλους τους γειτονικούς του κόμβους επισυνάπτοντας το ID του. Έτσι, ένας κόμβος που «ακούει» αυτά τα μηνύματα μπορεί να φτιάξει μία λίστα με τους γείτονές του. Εάν ένας κόμβος είναι CH και λάβει μήνυμα από άλλο CH με μικρότερο ID, αλλάζει την κατάστασή του και ανακηρύσσεται ως νέο του CH, αυτόν από τον οποίο «άκουσε» το μήνυμα "HELLO". Ένα απλό παράδειγμα ομαδοποίησης(clustering) με τον αλγόριθμο Lowest-ID φαίνεται στο επόμενο σχήμα.



ΟΜΑΔΟΠΟΙΗΣΗ(CLUSTERING) ΜΕ ΤΟΝ ΑΛΓΟΡΙΘΜΟ LOWEST-ID

Στα VANETs, τα οχήματα με διαφορετικές κατευθύνσεις είναι δύσκολο να μείνουν μαζί(σε επαφή) πάνω από ένα χρονικό διάστημα, το οποίο επηρεάζει την συνολική σταθερότητα του cluster και συμβάλει στη μη διατήρηση μόνιμων clusters. Ο αλγόριθμος για τη δημιουργία backbone που θα αναφέρουμε στη συνέχεια, χρησιμοποιεί τον Lowest-ID που περιγράψαμε παραπάνω και συμπεριλαμβάνει τα δεδομένα κατεύθυνσης, καθώς και το διάστημα για το οποίο είναι CH ένας κόμβος.

Αρχικά, κάθε κόμβος ανακηρύσσει τον εαυτό του clusterhead, CH. Στέλνει ένα πακέτο με το ID του και τη διεύθυνσή του(beacon) στους γείτονές του. Όταν ένας κόμβος λάβει beacons από τους γείτονές του είναι σε θέση να γνωρίζει την τωρινή γειτονιά του και μπορεί να αποφασίσει εάν θα αλλάξει ή όχι την κατάστασή του. Μπορεί δηλαδή να γίνει από CH, member ή το αντίθετο. Gateway είναι ένα member του cluster που επικοινωνεί με τουλάχιστον ένα member από άλλο γειτονικό cluster του δικτύου. Το εάν ένας κόμβος θα αλλάξει την κατάστασή του έχει να κάνει με τρεις παραμέτρους. Το ID του κόμβου από τον οποίο λάβαμε το beacon, την τωρινή κατεύθυνση αυτού του κόμβου και το χρονικό διάστημα για το οποίο αυτός ο κόμβος είναι CH, δηλαδή ηγείται. Επομένως, κάθε κόμβος αρχικά υποστηρίζει πως είναι ο ίδιος CH. Όταν λάβει beacon από γειτονικό του κόμβο, ελέγχει εάν έχουν την ίδια κατεύθυνση κίνησης. Εάν όντως έχουν, ελέγχει την κατάσταση του κόμβου. Εάν είναι CH ο κόμβος από τον οποίο έλαβε beacon, κοιτάζει το χρονικό διάστημα για το οποίο αυτός ο κόμβος υπήρξε CH. Εάν είναι μεγαλύτερο από το διάστημα που ο ίδιος υπήρξε CH, ελέγχει το ID. Εάν ο κόμβος από τον οποίο έλαβε το beacon έχει μικρότερο ID από το δικό του, ανακηρύσσει αυτόν τον κόμβο ως νέο του CH και ο ίδιος αλλάζει την κατάστασή του σε member του δικτύου. Σε κάθε άλλη περίπτωση παραμένει ο ίδιος CH.

Με την ολοκλήρωση αυτής της διαδικασίας, δημιουργούνται κατά περιόδους τα clusters του δικτύου και κάθε κόμβος έχει μία κατάσταση. Για την αναμετάδοση των πακέτων πληροφορίας, θα χρησιμοποιήσουμε τους CHs και ένα υποσύνολο των Gateways κόμβων του δικτύου. Για να βρούμε ποιοί gateways θα χρησιμοποιηθούν στην αναμετάδοση και ποιοί όχι, εφαρμόζουμε τη λογική που υπάρχει στον αλγόριθμο που παρουσιάζεται παρακάτω.

Broadcast Decision Algorithm

//invoked when receiving a broadcast packet

IF(the node is CH)

Re-broadcast the packet immediately

ELSE if (the node is gateway){

Initialize Counter = 1

when a packet is hear for the first time.

IF(the Counter has not exceeded threshold)

// Counter-Based Scheme

{

Set a timer, // RAD

IF(unseen package from “outside”)

:Re-broadcast the packet immediately

ELSE (increase Counter by one)

:when timer expires, if Counter<threshold,

:Rebroadcast the packet

}

ELSE Cancel the re-broadcast

ΑΛΓΟΡΙΘΜΟΣ ΕΥΡΕΣΗΣ ΤΩΝ Gateways ΠΟΥ ΘΑ ΧΡΗΣΙΜΟΠΟΙΗΘΟΥΝ ΣΤΗΝ ΑΝΑΜΕΤΑΔΟΣΗ ΤΩΝ ΠΑΚΕΤΩΝ ΠΛΗΡΟΦΟΡΙΑΣ

Έχουμε τελικά κάνει μία πολύ καλή επιλογή κόμβων του δικτύου που θα αναλάβουν την αναμετάδοση των πακέτων πληροφορίας μεταξύ των κόμβων. Κατασκευάσαμε δηλαδή ένα backbone χρησιμοποιώντας clusters, την κατάσταση των κόμβων καθώς και την κατεύθυνση κίνησης και το χρονικό διάστημα που είναι κάποιος κόμβος CH. Όλα αυτά συνέβαλαν στη δημιουργία σταθερών clusters και στη μείωση των συγκρούσεων και του ανταγωνισμού στην ανταλλαγή πακέτων πληροφορίας. Τέλος, το γεγονός ότι οι αναμεταδόσεις πακέτων γίνονται από τους CHs και κάποιους από τους gateways, έχει σαν αποτέλεσμα να προκύπτει τελικά ένας μικρός αριθμός αναμεταδόσεων [8].

ΈΝΑΣ ΚΑΤΑΝΕΜΗΜΕΝΟΣ ΑΛΓΟΡΙΘΜΟΣ ΔΥΝΑΜΙΚΗΣ ΟΜΑΔΟΠΟΙΗΣΗΣ, ΜΕ ΣΚΟΠΟ ΤΗ ΔΗΜΙΟΥΡΓΙΑ BACKBONE

Μπορούμε να φανταστούμε το backbone σαν μία αλυσίδα από κόμβους. Σε ένα δίκτυο οχημάτων, είναι μια εικονική αλυσίδα που απαρτίζεται από κάποια από τα οχήματα του δικτύου. Κάθε κόμβος που αποτελεί μέλος του backbone πρέπει να ενώνεται με τον προηγούμενο και τον επόμενο κόμβο-μέλος του backbone ώστε να έχουμε μια συνεχή αλυσίδα. Στη συνέχεια θα παρουσιάσουμε ένα cross-layered σχεδιασμό, για γρήγορη διάδοση των μηνυμάτων που μεταδίδονται σε ένα δίκτυο οχημάτων, VANET. Κατά το σχηματισμό του backbone λαμβάνουμε υπόψη δύο παραμέτρους. Την τωρινή απόσταση μεταξύ των οχημάτων που είναι υποψήφια για μέλη του backbone και την εκτιμώμενη διάρκεια ύπαρξης της ασύρματης σύνδεσης μεταξύ γειτονικών backbone μελών.

Κάθε κόμβος έχει ένα μοναδικό ID. Η κατάσταση στην οποία μπορεί να περιέλθει είναι Normal Vehicle, NV και Backbone Member, BM. Εάν ένας κόμβος δε λάβει μήνυμα beacon από κάποιο backbone κόμβο μέσα σε ένα χρονικό διάστημα, τότε ανακηρύσσει τον εαυτό του ως backbone member και στέλνει μήνυμα beacon σε κάθε γειτονικό κόμβο. Το BEACON μήνυμα περιέχει την εξής πληροφορία,

$\langle ID, (x,y), R, speed, dir, horizon \rangle$, όπου

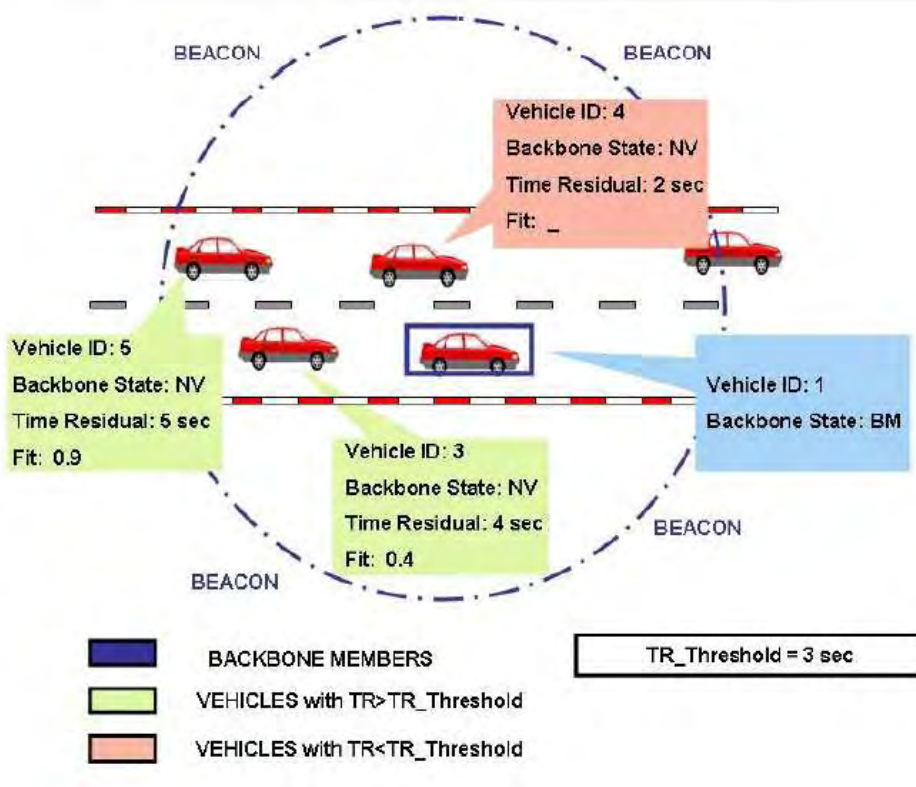
ID : το μοναδικό αναγνωριστικό του κόμβου

(x,y) : οι GPS συντεταγμένες

Speed : η μέση ταχύτητα

Dir : η κατεύθυνση κίνησης

Horizon : το όριο σε χώρο της ζώνης επικινδυνότητας.



ΑΝΑΠΑΡΑΣΤΑΣΗ VANET ΚΑΙ ΤΩΝ ΚΑΤΑΣΤΑΣΕΩΝ ΤΩΝ ΚΟΜΒΩΝ ΤΟΥ ΔΙΚΤΥΟΥ. (!)ΤΑ ΚΑΤΩΦΛΙΑ ΠΟΥ ΦΑΙΝΟΝΤΑΙ ΣΤΟ ΣΧΗΜΑ ΠΑΡΟΥΣΙΑΖΟΝΤΑΙ ΣΤΗ ΣΥΝΕΧΕΙΑ

Υποθέτουμε πως ο κόμβος με ID=1 έστειλε μήνυμα BEACON στους υπόλοιπους κόμβους. Τότε οι κόμβοι που λάβανε αυτό το μήνυμα και κινούνται προς την ίδια κατεύθυνση με τον κόμβο 1 είναι υποψήφιοι για τον επόμενο (next hop) κόμβο του backbone. Για παράδειγμα, θεωρούμε πως ο κόμβος A ακολουθεί τον κόμβο B. Ως RT ορίζουμε το χρονικό διάστημα που ο A θα εξακολουθεί να βρίσκεται μέσα στο πεδίο διάδοσης του B ώστε να μπορεί να επικοινωνεί μαζί του. Όταν ο A λάβει ένα BEACON μήνυμα από τον κόμβο B, το όχημα A υπολογίζει το $RT(A,B)$ της μεταξύ τους σύνδεσης.

Εάν το $RT(A,B)$ είναι μικρότερο από το BB_REFR , $RT(A,B) < BB_REFR$, (το χρονικό διάστημα ανανέωσης του backbone), τότε ο A δεν είναι καλός υποψήφιος για next hop μέλος του backbone καθώς αναμένεται να εξέλθει από το πεδίο διάδοσης του B. Στο παράδειγμα που φαίνεται στο σχήμα, τα οχήματα 3 και 5 είναι πιθανοί next hop υποψήφιοι του backbone, αλλά ο 4 δεν είναι αφού πρόκειται να προσπεράσει το όχημα 1 και ισχύει $RT(A,B) < BB_REFR$.

Τα οχήματα με $RT(A,B) > BB_REFR$, βρίσκονται σε μια ανταγωνιστική μεταξύ τους σχέση για το ποιο τελικά θα ανακηρυχθεί το next hop (του οχήματος 1) μέλος του backbone.

Το όχημα A αρχικά υπολογίζει τον παράγοντά του FF. Πρόκειται για μία σταθερά που χαρακτηρίζει τον κάθε κόμβο/όχημα και έχει να κάνει με το πόσο καλός υποψήφιος είναι ο κόμβος ώστε να γίνει ο next hop στο backbone. Στέλνει μήνυμα CANDIDATURE στον B και περιμένει να λάβει μήνυμα ACK_WINNER από τον B. Αν τελικά όντως λάβει τέτοιο μήνυμα, αλλάζει την κατάσταση του σε BM(backbone member) και στέλνει με τη σειρά του BEACON μήνυμα.

Το όχημα B με τη σειρά του, περιμένει να λάβει μήνυμα CANDIDATURE από κάποιο όχημα A. Όταν το λάβει, ελέγχει εάν η μεταβλητή next_hop που διατηρεί ο B είναι null. Εάν είναι, στέλνει στον A μήνυμα ACK_WINNER. Αλλιώς δε στέλνει τίποτα. Το πώς υπολογίζονται οι διάφοροι παράγοντες που αφορούν τα οχήματα καθώς και οι σημασία των διάφορων μεταβλητών που κρατά κάθε κόμβος, αναλύονται λεπτομερώς στο σχετικό άρθρο [9].

Ακολουθώντας τη διαδικασία που αναφέραμε παραπάνω, έχουμε σαν αποτέλεσμα το σχηματισμό μίας αλυσίδας οχημάτων, του backbone δηλαδή του VANET. Έχοντας καταλήξει, μπορούμε να χρησιμοποιήσουμε τα backbone μέλη για να μεταδώσουμε πακέτα πληροφορίας ή/και να μειώσουμε τις αναμεταδόσεις που λαμβάνουν χώρα στο δίκτυο.

Ο ΔΙΚΟΣ ΜΑΣ ΑΛΓΟΡΙΘΜΟΣ

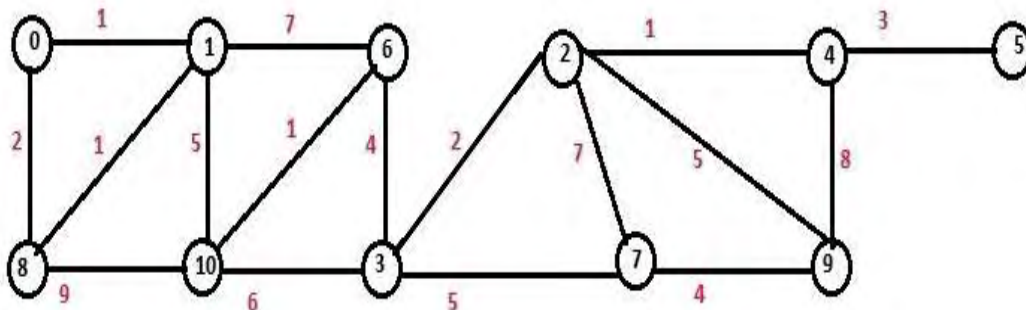
ΠΕΡΙΓΡΑΦΗ

Στο κεφάλαιο αυτό θα παρουσιάσουμε έναν δικό μας αλγόριθμο εύρεσης εικονικής ραχοκοκκαλιάς(virtual backbone), σε ένα δίκτυο οχημάτων. Ο αλγόριθμος αυτός είναι ευκαιριακός/καιροσκοπικός(opportunistic) και σκοπός του είναι η εύρεση ενός spanning tree του αρχικού γραφήματος και έπειτα το backbone που προκύπτει εύκολα από αυτό.

Στην ανάπτυξη αυτού του αλγορίθμου, κάνουμε τις ακόλουθες παραδοχές :

- Θα αναπαραστήσουμε το κάθε όχημα με έναν κόμβο.
- Η σύνδεση μεταξύ δύο οχημάτων θα καθορίζεται από την ύπαρξη ή μη ακμής μεταξύ αυτών των δύο κόμβων στο δίκτυο.
- Το γράφημα(δίκτυο) θα είναι βεβαρημένο, μη κατευθυνόμενο.
- Τέλος, το βάρος της κάθε ακμής μεταξύ δύο κορυφών(κόμβων) του γραφήματος θα παίρνει τιμές στο διάστημα $(0,9]$. (Θα μπορούσαμε κάλλιστα να ορίσουμε κ κάποιο άλλο βάρος).

Στην εικόνα παρακάτω, φαίνεται ένα αντιπροσωπευτικό παράδειγμα γραφήματος πάνω στο οποίο θα αναπτύξουμε τον αλγόριθμό μας. Πρόκειται για ένα γράφημα που αποτελείται από 11 κορυφές. Δίπλα σε κάθε ακμή φαίνεται το βάρος της.



ΑΝΑΠΑΡΑΣΤΑΣΗ ΔΙΚΤΥΟΥ ΠΟΥ ΑΠΟΤΕΛΕΙΤΑΙ ΑΠΟ 11 ΚΟΜΒΟΥΣ

Δεδομένου ενός γραφήματος, ξεκινάμε τη διαδικασία εύρεσης του backbone. Αρχικά, για κάθε κορυφή του γράφου μας, βρίσκουμε την 2-hop γειτονιά του. Ως 2-hop γειτονιά χαρακτηρίζονται όλοι οι κόμβοι του γραφήματος με τους οποίους μπορεί να επικοινωνήσει ένας κόμβος εκτελώντας μέχρι 2 hops, δηλαδή να φτάσει σε αυτόν ταξιδεύοντας μέσω το πολύ 2 ακμών του γράφου. Στο γράφημα της εικόνας παραπάνω για παράδειγμα, η 2-hop γειτονιά του κόμβου 2 αποτελείται από τους κόμβους 3, 4, 7, 9, 5, 6, 10. Όπως είναι αναμενόμενο, οι 2-hop γειτονιές για διαφορετικούς κόμβους δύναται να καλύπτουν η μία την άλλη, δηλαδή να υπάρχουν κοινοί κόμβοι στο σύνολο των 2 hop γειτόνων. Κάτι τέτοιο είναι λογικό και δεν αποτελεί πρόβλημα.

Έχοντας σχηματίσει και καταγράψει τις 2-hop γειτονιές για κάθε κόμβο του γραφήματος, επόμενος στόχος είναι να ανακηρύξουμε ένα τοπικό κέντρο βάρους για κάθε μία από αυτές τις γειτονιές. Να ανακηρύξουμε δηλαδή έναν local network centroid (LNC) κάθε γειτονιάς.

Για να καταλήξουμε ποιός κόμβος μπορεί να είναι LNC, αρχικά υπολογίζουμε το άκρη προς άκρη βάρος (end to end weight) του κόμβου x με κάθε άλλο κόμβο y που αποτελεί μέλος της γειτονιάς του. Υπολογίζεται αθροίζοντας τα βάρη των ακμών μέσω των οποίων ο κόμβος x μπορεί να προσεγγίσει τον κόμβο y εκτελώντας 2 hops. Για να γίνει πιο κατανοητό, για την 2-hop γειτονιά του κόμβου 2 που αναφέραμε παραπάνω, υπολογίζουμε το end to end weight του κόμβου 2 με τον κόμβο 3, του κόμβου 2 με τον κόμβο 4, του κόμβου 2 με τον κόμβο 7 κ.ο.κ. .

Ωστόσο, ο κόμβος x υπάρχει πιθανότητα να μπορεί να επικοινωνήσει με τον κόμβο y με περισσότερες της μίας εναλλακτικές διαδρομές (πάντα ταξιδεύοντας μέσω το πολύ δύο ακμών). Στην περίπτωση αυτή υπολογίζουμε όλες τις δυνατές περιπτώσεις, και τελικά κρατάμε αυτή με τη μικρότερη τιμή. Ας γυρίσουμε πάλι στο παράδειγμά μας και θα δούμε πως ο κόμβος 2 μπορεί να επικοινωνήσει με τον κόμβο 7 μέσω τριών διαφορετικών διαδρομών :

Κόμβος 2 -> Κόμβος 7 με συνολικό βάρος 7

Κόμβος 2 -> Κόμβος 3 -> Κόμβος 7 με συνολικό βάρος 7

Κόμβος 2 -> Κόμβος 9 -> Κόμβος 7 με συνολικό βάρος 9

Από αυτές τις δύο θα επιλέξουμε και θα κρατήσουμε αυτή που έχει το μικρότερο συνολικό βάρος. Σε περίπτωση ισοπαλίας θα κρατήσει μία από τις περιπτώσεις με το μικρότερο συνολικό βάρος. Τέλος, έχοντας υπολογίσει το end to end weight του κόμβου x με κάθε άλλο κόμβο y που αποτελεί μέλος της γειτονιάς του, κρατάμε τη μικρότερη τιμή όλων αυτών. Ο κόμβος u στον οποίο αντιστοιχεί αυτή η μικρότερη όλων τιμή ανακηρύσσεται και local network centroid της 2 hop γειτονιάς του κόμβου x . Για τον κόμβο 2, ο local network

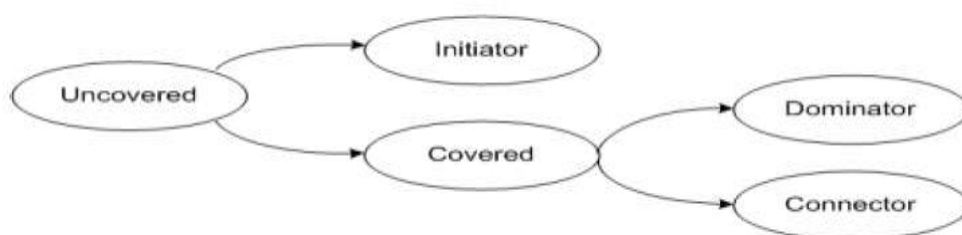
centroid της 2-hop γειτονιάς του είναι ο κόμβος $u=4$ με συνολικό end to end weight 1.

Θα προχωρήσουμε τώρα στη δημιουργία του backbone του γραφήματος. Πρόκειται για ένα backbone που θα προκύψει από ένα spanning tree. Το spanning tree αυτό θα είναι απόρροια της εύρεσης και ακολούθως της ένωσης πολλών άλλων μικρών υποδέντρων του αρχικού γραφήματος που θα υπολογίσουμε σε πρώτη φάση.

Να σημειώσουμε στο σημείο αυτό ότι κάθε κορυφή του γραφήματός μας χαρακτηρίζεται μοναδικά από ένα αναγνωριστικό ID και την κατάστασή της. Το ID είναι ένας αριθμός που την ξεχωρίζει από κάθε άλλη κορυφή. Η κατάσταση στην οποία βρίσκεται η συγκεκριμένη κορυφή περιγράφεται με ένα αλφαριθμητικό(string). Έτσι, κάθε ένας από τους κόμβους έχει ένα δικό του ρόλο, ο οποίος είναι πιθανόν να αλλάζει κατά τη διάρκεια της διαδικασίας εύρεσης του τελικού spanning tree. Η αρχική κατάσταση όλων των κορυφών όλου του γραφήματος είναι uncovered(μη καλυμμένος). Οι καταστάσεις στις οποίες μπορεί να περιέλθει ένας κόμβος είναι συνολικά 5,

- *Initiator*
- *Dominator*
- *Covered*
- *Uncovered*
- *Connector*

Από τους παραπάνω, οι δύο πρώτοι αρχικά θα αποτελούν μέλη του backbone και οι δύο επόμενοι μη μέλη του backbone. Οι Connectors θα προκύψουν αργότερα. Τώρα αναλυτικά, *Initiator*, είναι ένας αρχικός κόμβος ο οποίος είναι ρίζα κάποιου υποδέντρου. *Dominator*, είναι ένας επικρατών κόμβος, ή αλλιώς εσωτερικός κόμβος ενός υποδέντρου. *Covered*, είναι ένας καλυμμένος κόμβος, ή αλλιώς φύλλο ενός υποδέντρου. *Uncovered*, ένας μη καλυμμένος κόμβος(η αρχική κατάσταση όλων των κόμβων). Τέλος, ως *Connector* θα χαρακτηριστεί κάποιος από τους παραπάνω κόμβους, ο οποίος θα ενώσει δύο ή περισσότερα υποδέντρα. Τελικά, θα έχουμε ένα virtual backbone που θα αποτελείται από έναν *initiator*, *dominators* και *connectors*.



ΟΙ ΚΑΤΑΣΤΑΣΕΙΣ ΣΤΙΣ ΟΠΟΙΕΣ ΜΠΟΡΕΙ ΝΑ ΠΕΡΙΕΛΘΕΙ ΕΝΑΣ ΚΟΜΒΟΣ ΤΟΥ ΔΙΚΤΥΟΥ

Η όλη διαδικασία εύρεσης του virtual backbone ξεκινά με την ανακήρυξη των initiators(αρχικών κόμβων). Κάθε κόμβος είναι uncovered και έχει υπολογίσει πιο πάνω τον local network centroid της 2-hop γειτονιάς του. Initiator μπορεί να ανακηρυχθεί μόνο ένας local network centroid. Στην εύρεση των initiators συμμετέχουν όλοι οι uncovered κόμβοι του γραφήματος, οι οποίοι μπαίνουν σε μία ανταγωνιστική μεταξύ τους φάση. Έστω λοιπόν πως ο uncovered κόμβος u συναντά τον uncovered κόμβο v ,

- Εάν ο local network centroid του κόμβου u είναι ο ίδιος ο u , τότε ο κόμβος u ψηφίζει τον εαυτό του.
- Εάν ο local network centroid του κόμβου u είναι ο κόμβος v , τότε ο u θα ψηφίσει τον v σαν έναν πιθανό initiator.
- Εάν ο u ή ο v συγκεντρώσουν A ψήφους, τότε γίνονται όντως initiators.

A , είναι ένα κατώφλι ψήφων που πρέπει να συγκεντρώσει ένας υποψήφιος κόμβος. Από τη στιγμή που ο κόμβος v γίνει ένας initiator, ο u θα αλλάξει την κατάστασή του σε covered(δηλαδή καλυμμένος κόμβος), και το αντίστροφο.

Μετά την ολοκλήρωση της παραπάνω διαδικασίας, έχουν προκύψει κόμβοι-initiators και κόμβοι-covered. Στη συνέχεια μπορούν να προκύψουν διάφοροι συνδυασμοί κόμβων που συναντούν ο ένας τον άλλο. Με τον όρο «συναντούν», εννοούμε να έχουν δυνατότητα επικοινωνίας μεταξύ τους μέσω 1hop ή το πολύ 2hops. Μερικά τυπικά σενάρια περιπτώσεων που δύναται να προκύψουν παρουσιάζονται στη συνέχεια.

Όταν ένας uncovered κόμβος u συναντά ένα backbone μέλος(δηλαδή initiator ή dominator) v , πρέπει να αποφασίσει ο ίδιος ένα είναι πρόθυμος να καλυφθεί από τον κόμβο v . Να χαρακτηριστεί δηλαδή u ως covered και να αποδεχτεί τον v για initiator. Την απόφαση αυτή θα την πάρει ο u ελέγχοντας εάν το συνολικό βάρος του μονοπατιού που τον ενώνει με τον v , είναι μικρότερο από κάποιο κατώφλι βάρους B . Μόνο εφόσον το συνολικό βάρος του μονοπατιού είναι μικρότερο του κατωφλιού B που έχει οριστεί, ο u θα ανακηρύξει τον κάποιο κόμβο ως initiator δικό του και τον εαυτό του ως covered. Αν v είναι initiator, ο u θα ανακηρύξει τον v ως initiator δικό του και τον εαυτό του ως covered. Αν v είναι dominator ο u θα ανακηρύξει τον initiator του v ως initiator δικό του και τον εαυτό του ως covered. Σε κάθε άλλη περίπτωση ο u παραμένει uncovered. Στη δική μας περίπτωση θεωρήσαμε $B=10$, ενώ το βάρος που μπορεί να δοθεί σε μία ακμή περιέχεται στο διάστημα $(0,10)$, όπως αναφέραμε και πιο πάνω.

Στην περίπτωση που ένας uncovered κόμβος συναντά έναν covered, η διαδικασία είναι όμοια με αυτή της περίπτωσης που αναφέρθηκε μόλις προηγουμένως. Ο uncovered κόμβος u που συναντά τον covered κόμβο v , πρέπει να αποφασίσει ποια θα είναι η κατάσταση στην οποία θα περιέλθει. Μπορεί να δεχτεί να καλυφθεί από τον v και να χαρακτηριστεί ως covered ή

να παραμείνει *uncovered*. Εάν αποφασίσει να γίνει *covered*, αναγνωρίζει ως *initiator* δικό του τον *initiator* του v . Πλέον θα ανήκουν στο ίδιο υποδέντρο. Το κριτήριο με βάση θα πάρει την απόφαση ο u είναι πάλι το εάν το συνολικό βάρος του μονοπατιού που τον ενώνει με τον v , είναι μικρότερο από κάποιο κατώφλι B . Όπως και πάνω, θα πρέπει το βάρος του μονοπατιού να είναι μικρότερο του 10.

Σαν τελευταία αφήσαμε την περίπτωση όπου ένα backbone μέλος (*initiator* ή *dominator*) συναντά ένα άλλο backbone μέλος. Πλέον, ο κόμβος u και ο κόμβος v μπορεί να ανήκουν σε διαφορετικά μεταξύ τους υποδέντρα. Πρόκειται συνεπώς και για τη φάση στην οποία θα ανακηρυχθούν κάποιοι κόμβοι ως *connectors*. Στόχος μας να μπορέσουμε να ελέγξουμε το μέγεθος του backbone που θα προκύψει. Για να το πετύχουμε αυτό, προσπαθήσαμε να ελαττώσουμε τον αριθμό των κόμβων που θα γίνουν από *covered*, *connectors*. Έτσι, μόνο εφόσον δύο *spanning trees* δεν μπορούν να ενωθούν χρησιμοποιώντας τα backbone μέλη (*initiator* ή *dominator*), θα χρησιμοποιούνται οι *covered* κόμβοι.

Όταν λοιπόν, ένας *initiator* ή *dominator* κόμβος u συναντά έναν άλλο *initiator* ή *dominator* κόμβο v , πρέπει να αποφασίσει εάν θα γίνει *connector* και κατά συνέπεια αν θα προσχωρήσει στο υποδέντρο του v . Η απόφαση αυτή πρέπει να παρθεί προσεκτικά. Η πολιτική που ακολουθήσαμε είναι η εξής : μόνο ο κόμβος u που ανήκει σε υποδέντρο με *initiator* που έχει μικρότερο ID, έχει τη δυνατότητα να κάνει αίτηση για σύνδεση σε κάποιο άλλο υποδέντρο με *initiator* που έχει μεγαλύτερο ID. Ο κόμβος u , εφόσον έχει δικαίωμα αίτησης, θα πρέπει επιπροσθέτως να λάβει άδεια από τους κόμβους που είναι μέλη στο ίδιο υποδέντρο με τον u (έχουν δηλαδή τον ίδιο *initiator*). Εφόσον ο u λάβει την άδεια από τους υπόλοιπους κόμβους του υποδέντρου στο οποίο ανήκει, θα γίνει *connector*. *Connector* θα γίνει και ένας κόμβος από το υποδέντρο στο οποίο έκανε αίτηση σύνδεσης ο u . Στο δικό μας αλγόριθμο, για να πάρει ο u άδεια για ένωση, αρκεί να λάβει το OK από τα μισά τουλάχιστον μέλη του υποδέντρου στο οποίο ανήκει.

Εάν μετά το πέρας της παραπάνω διαδικασίας, εξακολουθούν να υπάρχουν περισσότερα του ενός δέντρα, τότε επαναλαμβάνουμε την όλη διαδικασία χρησιμοποιώντας και μη backbone μέλη. Δηλαδή *covered* κόμβους (*uncovered* κόμβοι δεν υπάρχουν διότι τα υποδέντρα έχουν ήδη σχηματιστεί).

Τελικά, θα έχουμε ένα *spanning tree* με έναν μοναδικό *initiator*. Το virtual backbone στο οποίο θέλαμε και να καταλήξουμε μέσα από αυτή τη διαδικασία θα απαρτίζεται από τον *initiator*, τους *dominators* και τους *connectors*.

ΠΑΡΑΔΕΙΓΜΑ

Εάν εφαρμόσουμε τον αλγόριθμο που περιγράψαμε παραπάνω για το κύκλωμα της εικόνας, η έξοδος που θα πάρουμε είναι η ακόλουθη : (παρουσιάζονται αναλυτικά τα στάδια από τα οποία περνάει ο αλγόριθμος ώστε να γίνει κατανοητή η όλη διαδικασία)

run:

Connection of each node with its 2-hop neighbors with the minimum path weight.

node 0 is connecting with node 1 with path weight : 1.0 via node 1

node 0 is connecting with node 6 with path weight : 8.0 via node 1

node 0 is connecting with node 8 with path weight : 2.0 via node 8

node 0 is connecting with node 10 with path weight : 6.0 via node 1

node 1 is connecting with node 0 with path weight : 1.0 via node 0

node 1 is connecting with node 3 with path weight : 11.0 via node 6

node 1 is connecting with node 6 with path weight : 6.0 via node 10

node 1 is connecting with node 8 with path weight : 1.0 via node 8

node 1 is connecting with node 10 with path weight : 5.0 via node 10

node 2 is connecting with node 3 with path weight : 2.0 via node 3

node 2 is connecting with node 4 with path weight : 1.0 via node 4

node 2 is connecting with node 5 with path weight : 4.0 via node 4

node 2 is connecting with node 6 with path weight : 6.0 via node 3

node 2 is connecting with node 7 with path weight : 7.0 via node 7

node 2 is connecting with node 9 with path weight : 5.0 via node 9

node 2 is connecting with node 10 with path weight : 8.0 via node 3

node 3 is connecting with node 1 with path weight : 11.0 via node 6

node 3 is connecting with node 2 with path weight : 2.0 via node 2

node 3 is connecting with node 4 with path weight : 3.0 via node 2

node 3 is connecting with node 6 with path weight : 4.0 via node 6

node 3 is connecting with node 7 with path weight : 5.0 via node 7

node 3 is connecting with node 8 with path weight : 15.0 via node 10

node 3 is connecting with node 9 with path weight : 7.0 via node 2

node 3 is connecting with node 10 with path weight : 5.0 via node 6

node 4 is connecting with node 2 with path weight : 1.0 via node 2

node 4 is connecting with node 3 with path weight : 3.0 via node 2

node 4 is connecting with node 5 with path weight : 3.0 via node 5

node 4 is connecting with node 7 with path weight : 8.0 via node 2

node 4 is connecting with node 9 with path weight : 6.0 via node 2

node 5 is connecting with node 2 with path weight : 4.0 via node 4

node 5 is connecting with node 4 with path weight : 3.0 via node 4

node 5 is connecting with node 9 with path weight : 11.0 via node 4

node 6 is connecting with node 0 with path weight : 8.0 via node 1

node 6 is connecting with node 1 with path weight : 6.0 via node 10

node 6 is connecting with node 2 with path weight : 6.0 via node 3

node 6 is connecting with node 3 with path weight : 4.0 via node 3

node 6 is connecting with node 7 with path weight : 9.0 via node 3

node 6 is connecting with node 8 with path weight : 8.0 via node 1

node 6 is connecting with node 10 with path weight : 1.0 via node 10

node 7 is connecting with node 2 with path weight : 7.0 via node 2

node 7 is connecting with node 3 with path weight : 5.0 via node 3

node 7 is connecting with node 4 with path weight : 8.0 via node 2

node 7 is connecting with node 6 with path weight : 9.0 via node 3

node 7 is connecting with node 9 with path weight : 4.0 via node 9

node 7 is connecting with node 10 with path weight : 11.0 via node 3

node 8 is connecting with node 0 with path weight : 2.0 via node 0

node 8 is connecting with node 1 with path weight : 1.0 via node 1

node 8 is connecting with node 3 with path weight : 15.0 via node 10

node 8 is connecting with node 6 with path weight : 8.0 via node 1

node 8 is connecting with node 10 with path weight : 6.0 via node 1

node 9 is connecting with node 2 with path weight : 5.0 via node 2

node 9 is connecting with node 3 with path weight : 7.0 via node 2

node 9 is connecting with node 4 with path weight : 6.0 via node 2

node 9 is connecting with node 5 with path weight : 11.0 via node 4

node 9 is connecting with node 7 with path weight : 4.0 via node 7

node 10 is connecting with node 0 with path weight : 6.0 via node 1

node 10 is connecting with node 1 with path weight : 5.0 via node 1

node 10 is connecting with node 2 with path weight : 8.0 via node 3

node 10 is connecting with node 3 with path weight : 5.0 via node 6

node 10 is connecting with node 6 with path weight : 1.0 via node 6

node 10 is connecting with node 7 with path weight : 11.0 via node 3

node 10 is connecting with node 8 with path weight : 6.0 via node 1

The local network centroid(lnc), of each 2-hop neighborhood of each node.

2hop neighborhood of node 0 with LNC : 1

2hop neighborhood of node 1 with LNC : 0

2hop neighborhood of node 2 with LNC : 4

2hop neighborhood of node 3 with LNC : 2

2hop neighborhood of node 4 with LNC : 2

2hop neighborhood of node 5 with LNC : 4

2hop neighborhood of node 6 with LNC : 10

2hop neighborhood of node 7 with LNC : 9

2hop neighborhood of node 8 with LNC : 1

2hop neighborhood of node 9 with LNC : 7

2hop neighborhood of node 10 with LNC : 6

There are 8 Local Network Centroids.

Find the initiators and create the sub-trees.

#uncovered = 0

Υπαρχουν 2 initiators. Einai oi :

Node 1

Node 2

The sub-trees finally are :

state[0] => covered , 1

state[1] => initiator , 1

state[2] => initiator , 2
state[3] => dominator , 2
state[4] => dominator , 2
state[5] => covered , 2
state[6] => dominator , 1
state[7] => dominator , 2
state[8] => covered , 1
state[9] => covered , 2
state[10] => covered , 1
initiator 1 has 5 members
initiator 2 has 6 members

Initially, try to relate the sub-trees using only
the VBB members->initiators and dominators.

Using only VBB members(initiators and dominators) we have :

state[0] => covered , 1
state[1] => initiator , 1
state[2] => initiator , 2
state[3] => dominator , 2
state[4] => dominator , 2
state[5] => covered , 2
state[6] => dominator , 1
state[7] => dominator , 2
state[8] => covered , 1

state[9] => covered , 2

state[10] => covered , 1

We had 2 initiators and now we have 2

There are no connectors

Now, we try to relate the sub-trees using and the
non-VBB members->covered(the leaves of the trees).

Finally, we have only one tree, with only one initiator

state[0] => covered , 2

state[1] => dominator , 2

state[2] => initiator , 2

state[3] => dominator , 2

state[4] => dominator , 2

state[5] => covered , 2

state[6] => dominator , 2

state[7] => dominator , 2

state[8] => covered , 2

state[9] => covered , 2

state[10] => covered , 2

The backbone members are :

Node 1

Node 2

Node 3

Node 4

Node 6

Node 7

The connectors are :

Node 3

Node 6

Connection between node 0 and node 1

Connection between node 1 and node 6

Connection between node 2 and node 3

Connection between node 3 and node 7

Connection between node 4 and node 2

Connection between node 4 and node 5

Connection between node 6 and node 3

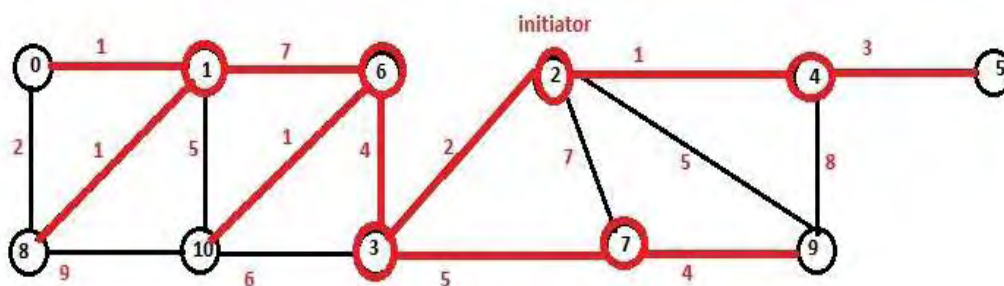
Connection between node 6 and node 10

Connection between node 7 and node 9

Connection between node 8 and node 1

BUILD SUCCESSFUL (total time: 0 seconds)

Η παραπάνω έξοδος αντικατοπτρίζεται στο ακόλουθο γράφημα :



Initiator ανακηρύχθηκε ο κόμβος 2. Ο ελάχιστος αριθμός ψήφων (votes) που έπρεπε να λάβει ορίστηκε στις 2(δηλαδή $Votes > 1$). Με κόκκινο περίγραμμα

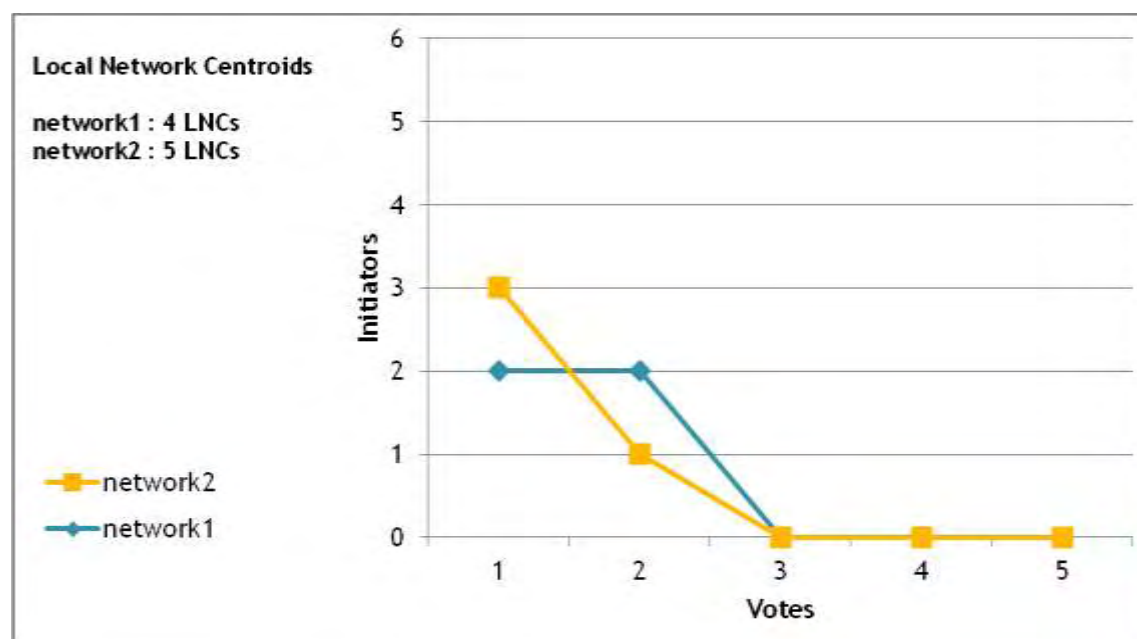
απεικονίζονται τα backbone μέλη, ενώ με κόκκινη γραμμή φαίνεται η διαδρομή μέσω της οποίας η πληροφορία που επιθυμούμε θα φτάσει σε όλα τα μέλη του γραφήματος.

ΠΕΙΡΑΜΑΤΙΚΗ ΑΠΟΤΙΜΗΣΗ

Στον αλγόριθμο εύρεσης virtual backbone(εικονικής ραχοκοκκαλιάς) ενός γραφήματος/κυκλώματος που υλοποιήσαμε, στο αρχικό στάδιο, για ανακηρυχθεί κάποιος κόμβος initiator(από τους local network centroids που έχουν προκύψει σε άλλο στάδιο προηγούμενως), πρέπει να λάβει έναν αριθμό θετικών ψήφων(votes) από τους υπόλοιπους κόμβους του κυκλώματος. Αναλόγως το πώς θα καθορίσουμε αυτό το κατώφλι ψήφων(κατώφλι A που αναφέραμε και παραπάνω), θα προκύψουν και οι αρχικοί initiators του κυκλώματος και στη συνέχεια θα εξελιχθεί και θα ολοκληρωθεί ο αλγόριθμος.

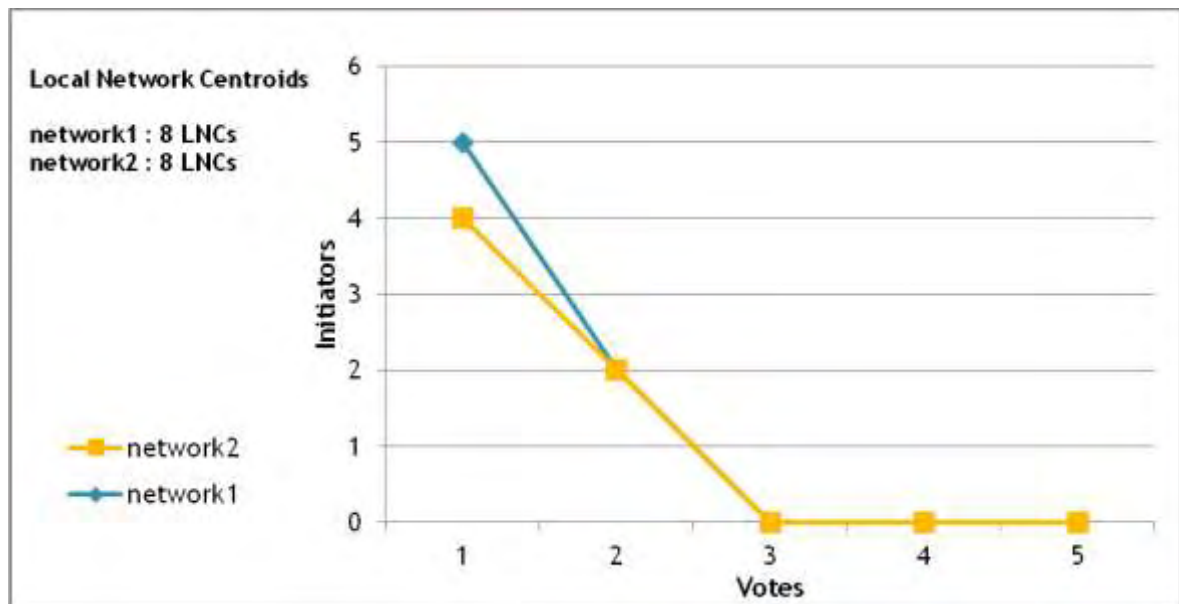
Στη συνέχεια παρουσιάζονται κάποια σχετικά γραφήματα. Πρόκειται για κυκλώματα που απαρτίζονται από 6 κόμβους συνολικά, από 11, από 30, από 40 και από 50 αντίστοιχα. Περιγράφουμε ενδεικτικά το πρώτο από τα γραφήματα, ενώ τα υπόλοιπα απλά παρατίθενται.

Networks with 6 Nodes.

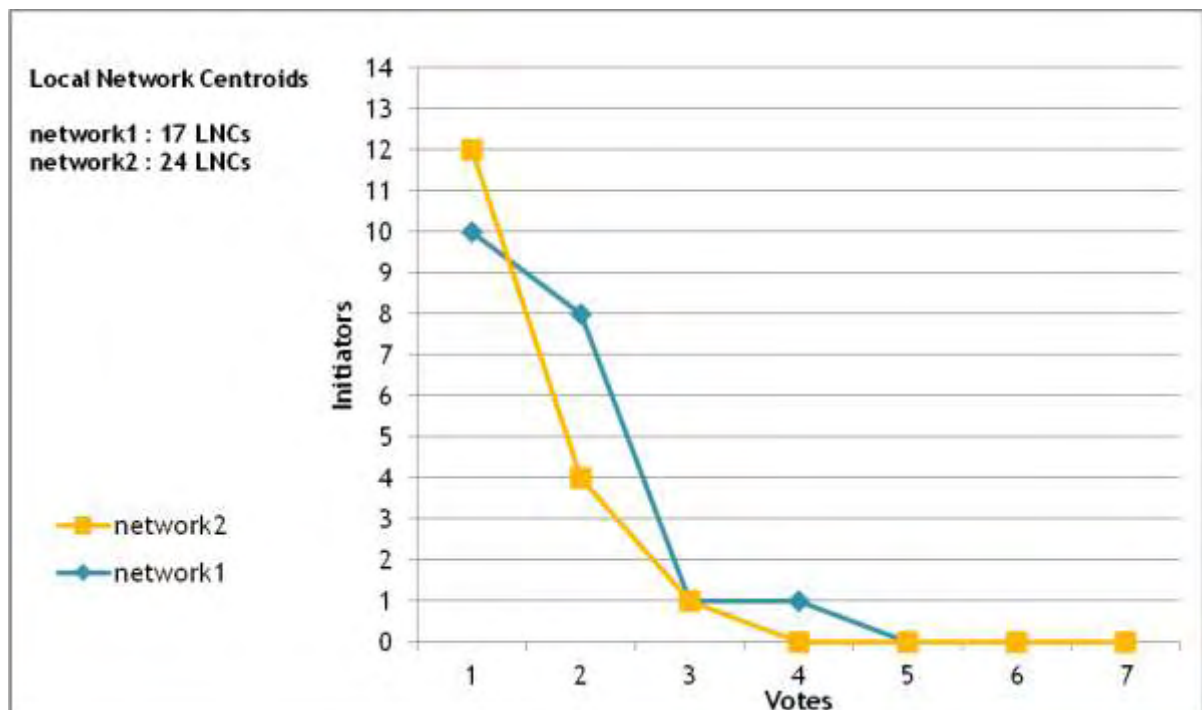


Μελετήσαμε δύο κυκλώματα των 6 κόμβων το καθένα. Στο network1, εάν απαιτείται 1 ψήφος τουλάχιστον, προκύπτουν αρχικά 2 initiators. Αντίστοιχα, στο network2 προκύπτουν 2 initiators. Εάν απαιτούνται 2 ψήφοι, τόσο στο network1 όσο και στο network2 προκύπτουν 2 initiators, ενώ εάν απαιτούνται 3 ή περισσότεροι ψήφοι και στις 2 περιπτώσεις κυκλωμάτων δεν προκύπτει κανένας initiator. Αυτό σημαίνει πως πρέπει να αναπροσαρμόσουμε το κατώφλι Α των ψήφων που έχουμε ορίσει(συγκεκριμένα να το ελαττώσουμε).

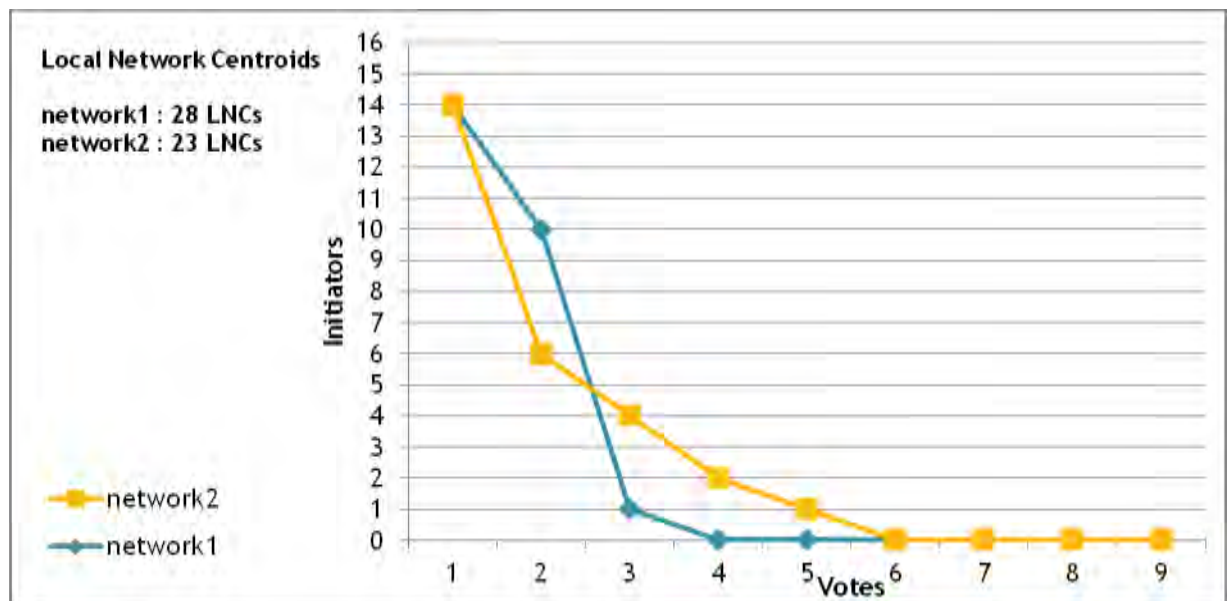
Networks with 11 Nodes.



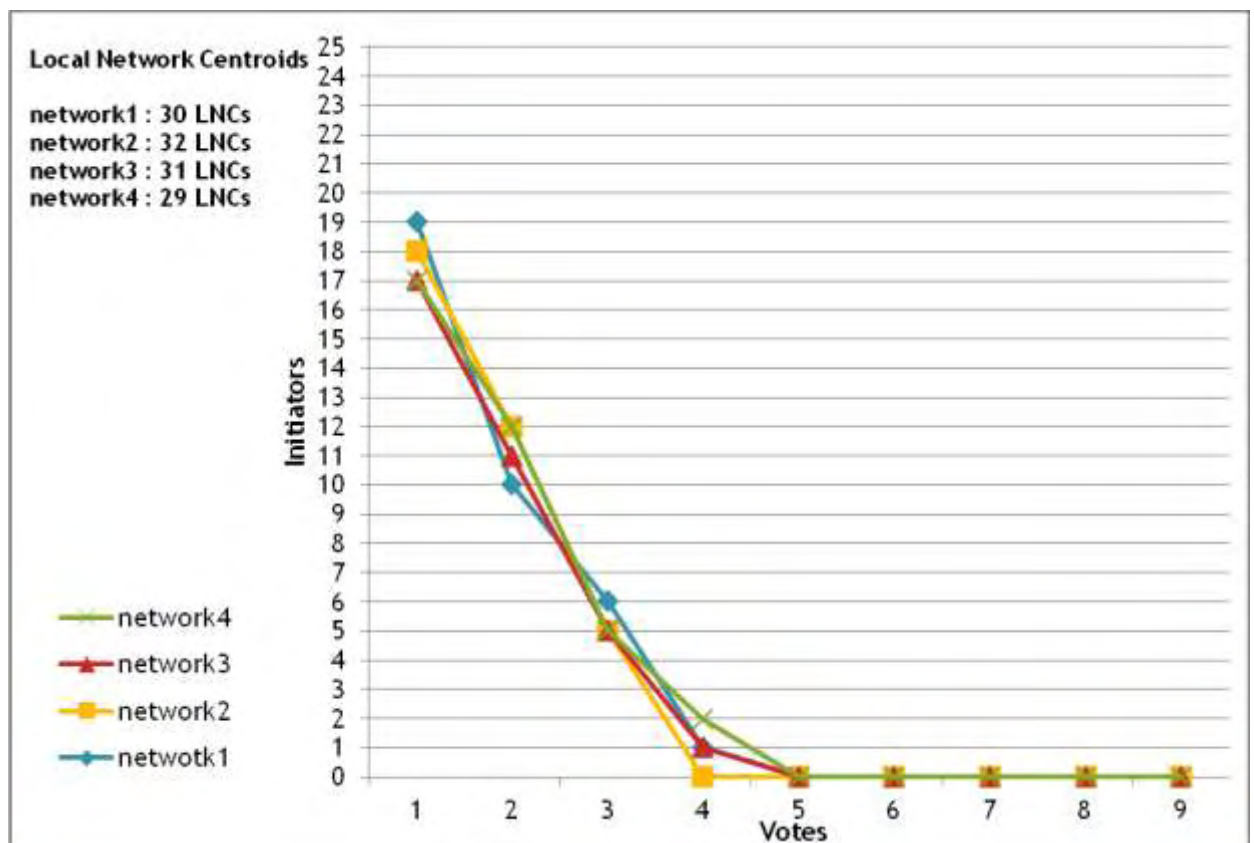
Networks with 30 Nodes.



Networks with 40 Nodes.



Networks with 50 Nodes.



(*)0 σημαίνει πως όλοι οι κόμβοι μείνανε uncovered και επομένως δεν έχει προκύψει αρχικά κανένας Initiator. Στην περίπτωση αυτή δεν έχουμε σχηματισμό backbone. Θα πρέπει να προσαρμόσουμε τον αριθμό των ψήφων ώστε τελικώς να μην προκύπτουν όλοι οι κόμβοι uncovered.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Παρατηρούμε πώς όσο μικρότερο είναι το κατώφλι των ψήφων A που ορίζουμε, τόσοι περισσότεροι είναι οι αρχικοί initiators που προκύπτουν. Καθώς αυξάνουμε αυτό το κατώφλι ψήφων, οι συνολικοί αρχικοί initiators ελαττώνονται. Τελικά καταλήγουμε στην περίπτωση που όλοι οι κόμβοι παραμένουν uncovered και επομένως ο αλγόριθμός μας δεν μπορεί να εκτελεστεί και να παράγει το backbone που επιθυμούμε. Πρέπει επομένως να εντοπίσουμε εκείνο το κατώφλι που είναι ιδανικό σε κάθε περίπτωση κυκλώματος και ανάλογα με τον αριθμό των συνολικών κόμβων να το ορίσουμε. Όσο μεγαλύτερος ο αριθμός των κόμβων του αρχικού κυκλώματος τόσο μεγαλώνει και η τιμή του κατωφλίου A που θα ορίσουμε. Σημαντικό ρόλο όπως είναι λογικό παίζει και η τοπολογία του εκάστοτε κυκλώματος. Τέλος, αξίζει να σημειώσουμε πως όσο μεγαλύτερος είναι ο αριθμός των local network centroids (LNCs) που προκύπτουν αρχικά, τόσο περισσότεροι είναι και οι αρχικοί initiators σε κάθε κύκλωμα, πράγμα που φαίνεται ξεκάθαρα και στα γραφήματα.

ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

Ο αλγόριθμος εύρεσης backbone που υλοποιήσαμε δεν έχει μελετηθεί με τη βοήθεια κάποιου simulator. Στοχεύουμε επομένως ως μελλοντική εργασία να ορίσουμε τη μελέτη του αλγορίθμου σε περιβάλλον προσομοίωσης και να μπορέσουμε να πάρουμε κάποια πειραματικά και πιο αντιπροσωπευτικά αποτελέσματα για την απόδοση του αλγορίθμου μας. Επιπλέον, ο αλγόριθμος εξελίσσεται λαμβάνοντας υπόψιν πρώτα τους κόμβους με τα μικρότερα IDs και στη συνέχεια αυτούς με τα μεγαλύτερα. Έτσι τα υποδέντρα χτίζονται και στη συνέχεια ενώνονται σε ένα τελικό ενιαίο δέντρο με έναν τρόπο σειριακό, από το μικρότερο στο μεγαλύτερο ID. Θα μπορούσαμε να δοκιμάσουμε τα υποδέντρα και όλη η διαδικασία να πραγματοποιούνται ταυτόχρονα, έχοντας δηλαδή παράλληλο προγραμματισμό με νήματα σε επίπεδο υλοποίησης.

- [1] Dedicated Short-Range Communications (DSRC) Standards in the United States, By John B. Kenney, Member IEEE
- [2] Routing in Vehicular Ad Hoc Networks: A Survey, Fan Li and Yu Wang, University of North Carolina at Charlotte
- [3] Yaozhou Ma and Abbas Jamalipour, Opportunistic Virtual Backbone Construction in Intermittently Connected Mobile Ad Hoc Networks, in the IEEE ICC 2011.
- [4] Gayathri Chandrasekaran, VANETs: The Networking Platform for Future Vehicular Applications.
- [5] http://en.wikipedia.org/wiki/Vehicular_ad-hoc_network .
- [6] http://www.ece.mtu.edu/~zli1/index_files/Page452.htm .
- [7] Khalid Almahorg, Otman Basir, Simulation-based Performance Comparison of VANETs Backbone Formation Algorithms.
- [8] Peng Fan, Improving Broadcasting Performance by Clustering with Stability for Inter-Vehicle Communication.
- [9] Luciano Bononi & Marco Di Felice, A Cross Layered MAC and Clustering Scheme for Efficient Broadcast in VANETs.
- [10] <http://www.wifinotes.com/mobile-communication-technologies/what-is-vanet.html> .
- [11] Arzoo Dahiya , Dr.R.K.Chauhan, A Comparative study of MANET and VANET Environment.
- [12] Shastri, R. Dadhich, Ramesh C. Poonia, PERFORMANCE ANALYSIS OF ON-DEMAND ROUTING PROTOCOLS FOR VEHICULAR AD-HOC NETWORKS, International Journal of Wireless & Mobile Networks (IJWMN) Vol. 3, No. 4, August 2011.
- [13] <http://www.businessdictionary.com/definition/backbone.html>
- [14] <http://bit.ly/Um2JE0> , εικόνα.
- [15] <http://bit.ly/Svr2IY> , εικόνα.
- [16] <http://bit.ly/P2vqhl> , εικόνα.