ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Διπλωματική Εργασία

"Υλοποίηση Μηχανισμού Προώθησης Συνεργασίας σε Δίκτυα Ad-Hoc"

Κωνσταντίνα Γ. Τερζάκη Παπαδοπούλου

Επιβλέπων: Μαλαματή Λούτα
Επίκουρος Καθηγήτρια

Κοζάνη, Οκτώβριος 2012

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΚΑΡΑΜΑΝΛΗ ΚΑΙ ΛΥΓΕΡΗΣ
|50100 |ΚΟΖΑΝΗ
www.icte.uowm.gr

Διπλωματική Εργασία

"Υλοποίηση Μηχανισμού Προώθησης Συνεργασίας σε Δίκτυα Ad-Hoc"

Κωνσταντίνα Γ. Τερζάκη Παπαδοπούλου

ΑΜ:98

Επιβλέπων: Μαλαματή Λούτα
Επίκουρος Καθηγήτρια

Εγκρίθηκε από την διμελή εξεταστική επιτροπή την 3η Οκτωβρίου 2012

....

Μαλαματή Λούτα Σαρηγιαννίδης Παναγιώτης

Κοζάνη, Οκτώβριος 2012

Περίληψη

Μεγάλη έμφαση στις μέρες μας έχει δοθεί στην μελέτη Μηχανισμών Κινήτρων ή διαφορετικά Μηχανισμών Προώθησης Συνεργασίας σε Ad-hoc Δίκτυα. Τα κινητά Ad-hoc δίκτυα γνωστά και ως MANET είναι ένα είδος δικτύων που δεν χρειάζεται για τη λειτουργία του δρομολογητές καθώς όλη η διαδικασία δρομολόγησης διενεργείται αποκλειστικά από τους κόμβους του δικτύου. Τα πρωτόκολλα δρομολόγησης που χρησιμοποιούν τα δίκτυα αυτά στηρίζονται στην απόλυτη συνεργασία των κόμβων μεταξύ τους, στην περίπτωση όμως εμφάνισης μη συνεργάσιμων κόμβων στο δίκτυο, τα πρωτόκολλα αυτά αδυνατούν να λειτουργήσουν σωστά. Για την αντιμετώπιση τέτοιων προβλημάτων έχουν αναπτυχθεί μηχανισμοί που ωθούν τους κόμβους να είναι συνεργάσιμοι. Οι μηχανισμοί αυτοί ονομάζονται Μηχανισμοί Κινήτρων ή αλλιώς Μηχανισμοί Προώθησης Συνεργασίας.

Στη συγκεκριμένη διπλωματική επιλέγεται και υλοποιείται ένας από τους πολλούς μηχανισμός προώθησης συνεργασίας που έχουν εμφανιστεί έως σήμερα. Οι μηχανισμοί αυτοί χωρίζονται σε τρείς κύριες κατηγορίες, τους βασισμένους στην αμοιβή, τους βασισμένους στη φήμη και τους υβριδικούς. Μετά από μία σύντομη αναφορά στους βασικότερους από αυτούς, επιλέγεται προς υλοποίηση ο μηχανισμός CONFIDANT που κατατάσσεται στους μηχανισμούς κινήτρων βασισμένους στη φήμη.

Η υλοποίηση του μηχανισμού CONFIDANT ορίζει την λειτουργία του ως επιπρόσθετη στο υπάρχον πρωτόκολλο δρομολόγησης DSR. Με τη βοήθεια του προσομοιωτή δικτύων ns-2 υλοποιείται ο μηχανισμός CONFIDANT και αξιολογείται η λειτουργία του όταν μη συνεργάσιμοι κόμβοι είναι παρών στο δίκτυο.

Λέξεις κλειδιά: δίκτυα Ad-hoc, MANET, μηχανισμοί Βασισμένοι σε Φήμη, μηχανισμοί Βασισμένοι σε Αμοιβή, CONFIDANT, DSR

Abstract

Nowadays great emphasis is given in studying Incentive mechanisms for cooperation in Ad-hoc Networks. A Mobile Ad-hoc Network, also known as MANET, is a network that does not need any routing infrastructure; since the nodes dynamically function together in order to accomplish forward process. Current Ad-hoc routing protocols assume that every node is willing to cooperate during the forward process, although if non-cooperative nodes enter the network, these protocols fail to function properly. To address such problems, mechanisms have been developed in order to make misbehaving nodes cooperate. These mechanisms are called Incentive mechanisms.

In this thesis, one of the many Incentive mechanisms has been selected and implemented. These mechanisms are divided into three main categories, Credit-based, Reputation-based and hybrid. After a brief reference to the most important of them, CONFIDANT mechanism is chosen for implementation which is classified under reputation Based Incentive mechanisms.

The implementation of CONFIDANT defines its function as additional to the existing Ad-hoc routing protocol DSR. The CONFIDANT mechanism is implemented using network simulator ns-2 as simulation environment, and evaluated when uncooperative nodes are present in the network.

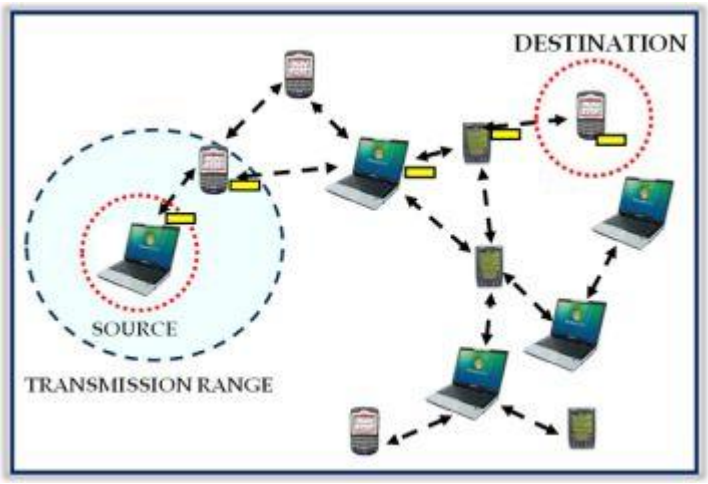
Key words: ad-hoc networks, MANET, Reputation Based Mechanisms, Credit Based Mechanisms, CONFIDANT, DSR

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή στα Ασύρματα Ad-hoc Δίκτυα (Mobile Ad-Hoc Networks, MANET)

Ένα ασύρματο Ad-hoc Δίκτυο ή διαφορετικά ένα αυτοοργανούμενο δίκτυο ή δίκτυο κατ' απαίτηση, είναι ένα αυτόνομο δίκτυο χωρίς καμία σταθερή υποδομή. Το όνομά του το πήρε από την λατινική έκφραση "Ad hoc" που σημαίνει "γι' αυτό το σκοπό". Ένα δίκτυο αυτής της μορφής, δεν χρειάζεται για την λειτουργία του υποδομές όπως ασύρματους δρομολογητές ή συγκεκριμένη τοπολογία. Η διαδικασία δρομολόγησης γίνεται αποκλειστικά από τους κόμβους του δικτύου και καθορίζεται δυναμικά ανάλογα με την συνδεσιμότητα αυτού, οποιαδήποτε χρονική στιγμή. Οι κόμβοι που αποτελούν το δίκτυο οργανώνονται μεταξύ τους για να παρέχουν μονοπάτια διαδρομών για δρομολόγηση πακέτων από οποιοδήποτε σημείο του δικτύου προς οποιοδήποτε προορισμό. Ένα παράδειγμα Ad-hoc δικτύου φαίνεται στο παρακάτω σχήμα:



Όπως φαίνεται, κόμβος ενός ασύρματου Ad-hoc δικτύου μπορεί να είναι οποιαδήποτε συσκευή μπορεί να λάβει ή να στείλει ασύρματα μηνύματα. Για παράδειγμα, ένας κόμβος του δικτύου μπορεί να είναι ένα laptop, ένα κινητό τηλέφωνο ή ένα Pad. Κάθε συσκευή σε ένα MANET είναι ελεύθερη να κινηθεί προς οποιαδήποτε κατεύθυνση και να αλλάξει τις ζεύξεις της με τις υπόλοιπες συσκευές του δικτύου.

1.1 *Χαρακτηριστικά*

Μερικά από τα χαρακτηριστικά ενός ασύρματου Ad-hoc δικτύου είναι:

- Η ***αυτοδυναμία***, δηλαδή η δυνατότητα τους να σχηματίζουν δίκτυο ακόμη και χωρίς τη χρήση άλλων τοπικών δικτύων ή το Internet, (χωρίς βέβαια αυτό να τους εμποδίζει από το να μπορούν να συνδεθούν και σε αυτά).
- Η ***έλλειψη κεντρικού συντονιστή***, δηλαδή δεν χρειάζεται κάποιος κεντρικός κόμβος που να συντονίζει την διαδικασία δρομολόγησης.
- Η ***ισότητα των κόμβων***, δηλαδή όλοι οι κόμβοι έχουν ίσες δυνατότητες και υποχρεώσεις προς το δίκτυο.
- Η ***μεταβαλλόμενη τοπολογία και η αυξημένη κινητικότητα των κόμβων***, δηλαδή η δυνατότητα του δικτύου να λειτουργεί ακόμα και όταν αυτό αλλάζει διαρκώς.
- Η ***μικρή διάρκεια ζωής των κόμβων***, δηλαδή η δυνατότητα ενός κόμβου να εισέρχεται και να εξέρχεται σε ένα Ad-hoc δίκτυο οποιαδήποτε χρονική στιγμή εκείνος το επιθυμεί.
- Το ***μικρό εύρος ζώνης των συχνοτήτων των κόμβων***, δηλαδή η μικρή εμβέλεια των συσκευών υλικού που μπορεί να χρησιμοποιούν οι κόμβοι.
- Και τέλος, η ***περιορισμένη ισχύς***, δηλαδή οι περιορισμοί σε ενέργεια που έχουν οι κινητές συσκευές που απαρτίζουν το δίκτυο, αφού συνήθως λειτουργούν με μπαταρίες και δεν τους επιτρέπεται πάντα να προμηθεύονται ενέργεια όταν τους χρειάζεται.

1.2 *Εφαρμογές*

Τα παραπάνω χαρακτηριστικά κατατάσσουν τα αυτοοργανούμενα δίκτυα πλέον κατάλληλα για πολλές εφαρμογές. Η εφαρμογή τους στον στρατό, υπήρξε η αφορμή για την ανάπτυξη και μελέτη τους (όπως συνήθως συμβαίνει με πολλές τεχνολογίες). Σε ένα εχθρικό περιβάλλον, η ευελιξία και η δυναμική φύση που χαρακτηρίζουν τα MANET τα καθιστούν πλέον κατάλληλα για τις στρατιωτικές τηλεπικοινωνίες, καθώς η δημιουργία σταθμών βάσης, εκτός από χρονοβόρα διαδικασία, είναι και ευάλωτη σε επιθέσεις. Παραδείγματα τέτοιου δικτύου είναι η επικοινωνία μεταξύ στρατιωτικών οχημάτων κατά τη διάρκεια της μάχης ή η επικοινωνία ενός στόλου μέσα στη θάλασσα.

Μία ακόμη σημαντική εφαρμογή ad-hoc δικτύων είναι τα δίκτυα ασύρματων αισθητήρων (Wireless Sensor Networks). Σε αυτά, οι αυτόνομοι κόμβοι του δικτύου περιέχουν αισθητήρες που καταγράφουν μετρήσεις για διάφορα φυσικά ή περιβαλλοντικά μεγέθη όπως η θερμοκρασία, ο ήχος, η δόνηση, η

πίεση ή τα σωματίδια μόλυνσης. Αυτό επιτρέπει για παράδειγμα, ένα δασαρχείο να ενημερωθεί απευθείας για το ακριβές σημείο και την ακριβής χρονική στιγμή που ξέσπασε μία πυρκαγιά στο δάσος ή τον κατασκευαστή μίας γέφυρας να γνωρίζει την κατάσταση της υποδομής της όποτε αυτός το επιθυμεί. Οι όλο και αυξανόμενες δυνατότητες εφαρμογής τους σε διάφορες πτυχές της ανθρώπινης δραστηριότητας -από το σπίτι μέχρι τη βιομηχανία, τις υπηρεσίες υγείας και την καταγραφή φυσικών φαινομένων- καθιστά δεδομένη την ταχύτατη ανάπτυξη και εισχώρησή τους στην καθημερινότητά του αύριο.

1.3 *Δρομολόγηση*

Η διαδικασία Δρομολόγησης σε ένα ασύρματο Ad-hoc Δίκτυο συνίσταται στην εύρεση ενός μονοπατιού από έναν κόμβο πηγής (αποστολέα) σε έναν κόμβο προορισμού (παραλήπτη) , κατά μήκος του οποίου θα μεταδοθεί η πληροφορία. Όμως η μεταβαλλόμενη τοπολογία και η κινητικότητα των κόμβων καθιστά την διαδικασία δρομολόγησης ιδιαίτερα δύσκολη κάτι που αποτελεί αντικείμενο μελέτης για αρκετά χρόνια τώρα.

Η πρώτη απλή προσέγγιση δρομολόγησης είναι η *πλημμύρα*, σε αυτήν την περίπτωση ο κόμβος που θέλει να στείλει ένα μήνυμα το προωθεί σε όποιον κόμβο βρίσκεται εντός της εμβέλειάς του, κατ' επέκταση και κάθε κόμβος που λαμβάνει ένα πακέτο προς προώθηση, εκτελεί την ίδια διαδικασία (δηλαδή το στέλνει προς όλους στην εμβέλειά του) με την διαφορά ότι δεν το στέλνει πίσω σε εκείνον που του το έστειλε. Έτσι, κάποια στιγμή το μήνυμα βρίσκει τον προορισμό του. Ο αλγόριθμος αυτός δεν είναι καθόλου αποδοτικός, καθώς παράγει πολλά πακέτα και αυξάνει τις πιθανότητες συγκρούσεων στο κανάλι.

Υπάρχουν πολλά πρωτόκολλα που έχουν προταθεί για την δρομολόγηση στα MANET δίκτυα. Τα πιο διαδεδομένα από αυτά είναι το DSDV (Destination-Sequenced Distance Vector), το DSR (Dynamic Source Routing) ή αλλιώς Πρωτόκολλο ανεξάρτητης δυναμικής δρομολόγησης, το TORA (Temporally-Ordered Routing Algorithm) και το AODV (Ad-hoc On Demand Distance Vector).

1.3.1 DSDV (Destination-Sequence Distance Vector)

Το DSDV πρωτόκολλο δρομολογεί βασιζόμενο σε πίνακες δρομολόγησης. Ο κάθε κόμβος του δικτύου διαθέτει ένα πίνακα με κόμβους προορισμούς μαζί με τα μονοπάτια που πρέπει να

χρησιμοποιηθούν για να φτάσουν σε αυτούς. Επίσης, η κάθε εγγραφή του πίνακα δρομολόγησης έχει έναν και μοναδικό αριθμό σειράς.

Προκειμένου η δρομολόγηση να είναι εφικτή, ο κάθε κόμβος πρέπει να διατηρεί ενημερωμένους τους πίνακες δρομολόγησής του. Αυτό, έχει ως συνέπεια την περιοδική ενημέρωση των πινάκων, μια διαδικασία με σημαντική επιβάρυνση για το δίκτυο. Για την ενημέρωση των πινάκων, το DSDV χρησιμοποιεί δύο τρόπους, την πλήρη ενημέρωση ή full dump και την αυξητική ενημέρωση ή incremental. Στις περισσότερες περιπτώσεις επιλέγεται ο δεύτερος τρόπος, αφού λειτουργεί ως ενημερωτικός του πρώτου.
Ο τρόπος επιλογής κάποιας διαδρομής, περιορίζεται σε δύο βασικά κριτήρια. Τον χρόνο ανακάλυψης της διαδρομής και το μήκος της σε προωθητικούς κόμβους. Τέλος, το πρωτόκολλο αυτό, λαμβάνει υπόψη του τον χρόνο που απαιτείται προκειμένου να αποθηκευτεί μία διαδρομή. Οι κόμβοι καθυστερούν για ένα προκαθορισμένο χρονικό διάστημα την μετάδοση πληροφοριών δρομολόγησης, με αποτέλεσμα να μειωθεί η επιβάρυνση του δικτύου και να αποφευχθεί η αναμετάδοση ίδιων μονοπατιών.

1.3.2 DSR (Dynamic Source Routing)

Το DSR επιτρέπει στο δίκτυο να είναι εντελώς αυτόνομο και αυτοοργανούμενο , χωρίς την ανάγκη χρήσης κάποιας δικτυακής υποδομής ή διαχείρισης. Αποτελείται από δύο μηχανισμούς την Εύρεση Διαδρομών (Route Discovery) και την Διατήρηση Διαδρομών (Route Maintenance). Οι δύο αυτοί μηχανισμοί, συνεργάζονται έτσι ώστε να επιτρέπουν στους κόμβους να ανακαλύψουν και να διατηρήσουν διαδρομές προς διάφορους προορισμούς σε όλο το Ad-hoc δίκτυο. Η χρήση της ανεξάρτητης δρομολόγησης, δεν επιτρέπει στην μετάδοση των πακέτων να είναι επιπόλαιη και να χρειάζεται συνεχής ενημέρωση διαδρομών ενώ, αντίθετα επιτρέπει στους κόμβους που συμμετέχουν ή ακούν τυχαία μία μετάδοση, να κρατούν πληροφορίες διαδρομών οι οποίες μπορεί να τους χρειαστούν μελλοντικά. Όλες οι πτυχές του πρωτοκόλλου ενεργούν κατ΄ απαίτηση, επιτρέποντας στο overhead του προωθούμενου πακέτου να κλιμακώνεται αυτόματα με σκοπό να περιέχει μόνο τις απαραίτητες πληροφορίες για τις διαδρομές που χρησιμοποιεί εκείνη τη στιγμή το δίκτυο.

1.3.3 TORA (Temporally-Ordered Routing Algorithm)

Το TORA είναι ένα πρωτόκολλο διανεμημένης δρομολόγησης που βασίζεται στην αντιστροφή των συνδέσεων. Έχει σχεδιαστεί για να ανακαλύπτει διαδρομές κατ' απαίτηση, να παρέχει πολλαπλές διαδρομές για έναν προορισμό, να εγκαθιστά διαδρομές γρήγορα και να ελαχιστοποιεί το overhead της επικοινωνίας εντοπίζοντας αλγοριθμικά τυχόν διαφοροποιήσεις στην τοπολογία του δικτύου, όταν αυτό είναι εφικτό. Η εύρεση βέλτιστης διαδρομής θεωρείται δευτερεύουσας σημασίας και μεγαλύτερες διαδρομές συχνά χρησιμοποιούνται κατά την δρομολόγηση, έτσι αποφεύγεται η επιβάρυνση του δικτύου από συχνές απόπειρες εύρεσης καινούργιων διαδρομών.

Το TORA μπορεί να περιγραφεί μεταφορικά σαν νερό που ρέει προς τα κάτω με προορισμό έναν κόμβο μέσα από ένα δίκτυο σωληνώσεων. Οι σωλήνες συμβολίζουν τις συνδέσεις μεταξύ των κόμβων του δικτύου, οι ενώσεις των σωληνώσεων τους κόμβους του δικτύου, και το νερό τα πακέτα που κατευθύνονται προς έναν προορισμό. Ο κάθε κόμβος βρίσκεται σε ένα ύψος ανάλογα με τον προορισμό που θα έχουν τα πακέτα που θα στείλει. Εάν ένας κόμβος μεταξύ π.χ. του Α και του Β φράξει έτσι ώστε το νερό να μην μπορεί να περάσει μέσα από αυτόν, το ύψος του Α αλλάζει σε ένα ύψος μεγαλύτερο από αυτό των γειτόνων του, έτσι ώστε το νερό να μπορέσει να κυλήσει μακριά από τον Α και προς τους υπόλοιπους κόμβους που είχαν προωθήσει πακέτα για προώθηση στον Α.

1.3.4 AODV (Ad-hoc On Demand Distance Vector)

Ο AODV αποτελεί μία βελτίωση του DSDV. Η ιδέα είναι να αποφευχθεί η δαπανηρή διαδικασία της συντήρησης της Λίστας Διαδρομών κάθε κόμβου και η εύρεση διαδρομών να εκτελείται κατ' απαίτηση. Για το σκοπό αυτό χρησιμοποιείται η τεχνική της πλημμύρας που περιγράψαμε παραπάνω.

Αρχικά, ο κάθε κόμβος θα πρέπει να γνωρίζει τους γειτονικούς του. Γι' αυτό το λόγο περιοδικά αποστέλλονται μηνύματα χαιρετισμού, έτσι ώστε να διατηρούνται ενημερωμένοι οι πίνακες γειτνίασης των κόμβων. Για να δημιουργηθεί μια διαδρομή προς τον προορισμό, η αφετηρία δημιουργεί πακέτα αίτησης διαδρομής (ROUTE REQUEST – RREQ). Ο κάθε κόμβος που παραλαμβάνει ένα ROUTE REQUEST πακέτο διατηρεί έναν δείκτη προς τον κόμβο που το έστειλε και το

προωθεί στους γείτονές του εφόσον δεν γνωρίζει τον προορισμό (τεχνική της πλημμύρας). Όταν τελικά το πακέτο φθάσει στον προορισμό του, αυτός με τη σειρά του θα δημιουργήσει ένα πακέτο απάντησης της αίτησης διαδρομής (ROUTE REPLY RREP). Στη συνέχεια, ακολουθώντας τους δείκτες που κράτησαν οι κόμβοι που προώθησαν το πακέτο, το RREP θα βρει τον δρόμο πίσω για τον κόμβο αποστολέα της Αίτησης Δρομολόγησης. Προκειμένου να αποφευχθεί η επανάληψη μίας προώθησης του ίδιου πακέτου από έναν ενδιάμεσο κόμβο, τα πακέτα προσδιορίζονται μονοσήμαντα μέσω της IP διεύθυνσης της αφετηρίας και ενός αναγνωριστικού id για την συγκεκριμένη διαδικασία Εύρεσης Διαδρομής.
Αξίζει να σημειώσουμε πως το πρωτόκολλο AODV χρησιμοποιεί στοιχεία, πέρα από του DSDV , και του DSR (βλέπε ROUTE-REQUEST και ROUTE-REPLY).

Αυτά τα πρωτόκολλα μπορούν να κατηγοριοποιηθούν ανάλογα με το είδος των αλγορίθμων που χρησιμοποιούν σε *Προληπτικά* (Proactive) και *Αντιδραστικά* (Reactive).

- Τα *Προληπτικά Πρωτόκολλα* είναι εκείνα που διατηρούν πληροφορίες δρομολόγησης πολύ πριν τις χρειαστούνε. Σε αυτή την κατηγορία βρίσκεται και το πρωτόκολλο DSDV. Σημαντικό μειονέκτημά τους είναι η χρήση μνήμης για διαδρομές που μπορεί να μην χρειαστούν ποτέ στην διαδικασία δρομολόγησης.
- Τα *Αντιδραστικά Πρωτόκολλα* αντιθέτως, αποθηκεύουν στην μνήμη τους διαδρομές που έχουν χρησιμοποιήσει, ή υπάρχει μεγάλη πιθανότητα να χρησιμοποιήσουν. Διενεργούνται διαδικασίες εύρεσης διαδρομών, μόνο όταν προκύψει η ανάγκη μεταφοράς ενός πακέτου από κόμβο που δεν έχει μέχρι εκείνη τη στιγμή αποθηκευμένο μονοπάτι για τον προορισμό στην μνήμη του. Τέτοια πρωτόκολλα είναι το DSR, το AODV και το TORA. Μειονέκτημα τέτοιων πρωτοκόλλων, είναι οι καθυστερήσεις και ο φόρτος στο δίκτυο, όταν δεν υπάρχει έτοιμο μονοπάτι για μεταφορά.

Τέλος, όσο αφορά τα πρωτόκολλα δρομολόγησης πακέτων σε MANET, αξίζει να σημειωθεί πως σύμφωνα με συγκρίσεις που έχουν γίνει [2] μεταξύ των παραπάνω τεσσάρων σημαντικών πρωτοκόλλων, το πρωτόκολλο DSR φαίνεται να συμπεριφέρεται καλύτερα έναντι των άλλων. Αυτός είναι και ο

λόγος που επιλέγεται να μελετηθεί περαιτέρω παρακάτω και να αποτελέσει το βασικό δομικό στοιχείο της υλοποίησης.

Κεφάλαιο 2

Μηχανισμοί Κινήτρων Συνεργασίας στα Ad-hoc Δίκτυα

2.1. Εισαγωγή

Τα Πρωτόκολλα Δρομολόγησης που έχουν προταθεί, αναφέρονται σε δίκτυα όπου όλοι οι κόμβοι συμπεριφέρονται πάντα σύμφωνα με τους κανόνες του πρωτοκόλλου που ισχύει. Αυτό φαντάζει ιδανικό, ειδικά όταν υποτεθεί πως τους κόμβους τους χειρίζονται χρήστες ικανοί να πραγματοποιήσουν αλλαγές στην δρομολόγηση, ανάλογα με τα συμφέροντά τους. Σε ένα τέτοιο δίκτυο, η συνεργασία των κόμβων δεν μπορεί να θεωρείται πάντα δεδομένη. Μία τέτοια "ανάρμοστη" συμπεριφορά μπορεί να προκαλέσει σοβαρά προβλήματα στην λειτουργία του δικτύου.

Οι χρήστες που χειρίζονται τους κόμβους τους ανάλογα με τα συμφέροντά τους, μπορούν να διακριθούν σε δύο κατηγορίες: τους εγωιστικούς (selfish) και τους κακόβουλους (malicious). Για λόγους συντομίας, στη συνέχεια, θα αναφέρονται οι κόμβοι που χειρίζονται από εγωιστικούς χρήστες ως εγωιστικοί κόμβοι (selfish nodes) και οι κόμβοι που χειρίζονται από κακόβουλους χρήστες ως κακόβουλοι κόμβοι(malicious nodes).

2.2. Η έννοια των Εγωιστικών και των Κακόβουλων κόμβων

Οι *εγωιστικοί κόμβοι* είναι εκείνοι που έχουν ως στόχο να αυξήσουν το κέρδος τους σε ενέργεια κυρίως αδιαφορώντας για την επίδοση του δικτύου. Αυτό που θέλουν είναι να μειώσουν, όσο αυτό είναι εφικτό, το κόστος τους σε ενέργεια και γι' αυτό αρνούνται να προωθήσουν πακέτα προκαλώντας διακοπές στο σημείο του δικτύου που βρίσκονται.

Αντίθετα, οι *κακόβουλοι κόμβοι* προσπαθούν να εξαπατήσουν και να επιτεθούν στο δίκτυο προκαλώντας έτσι σοβαρά προβλήματα. Εκτός από την άρνηση προώθησης πακέτων που κάνουν και οι εγωιστικοί κόμβοι, μπορούν να αλλοιώσουν στοιχεία, να αναγκάσουν την αλλαγή των μονοπατιών αναφερόμενοι σε ψευδή στοιχεία, να απομονώσουν ανυποψίαστους κόμβους ή ακόμα και να υποδυθούν άλλους "καλούς" κόμβους. Πιο συγκεκριμένα ένας κακόβουλος κόμβος μπορεί να προβεί στις παρακάτω επιθετικές ενέργειες:

- Να αρνηθεί να προωθήσει πακέτα και να τα απορρίψει

- Να μην αναφέρει τυχόν λάθη για την δρομολόγηση
- Να αλλάξει το μονοπάτι που ακολουθεί ένα πακέτο
- Να αλλοιώσει στοιχεία που τον αφορούν
- Να αναγκάσει σε αλλαγή δρομολόγησης λόγω μη εφικτής σύνδεσης, ενώ η σύνδεση λειτουργεί κανονικά
- Να ακυρώσει ολόκληρες διαδικασίες
- Να συνεργαστεί με άλλους κακόβουλους κόμβους με σκοπό τον έλεγχο της κίνησης
- Να υποδυθεί κάποιον άλλο κόμβο
- Να απομονώσει τυχόν δυσπρόσιτους κόμβους

Οι παραπάνω ενέργειες προσφέρουν στον κόμβο καλύτερη απόδοση, χρηματικό κέρδος, λιγότερη σπατάλη ενέργειας και απόκτηση εμπιστευτικών πληροφοριών.

2.3. Τι είναι οι Μηχανισμοί Κινήτρων ή Μηχανισμοί Προώθησης Συνεργασίας

Σύμφωνα με την παραπάνω θεωρία, ένα δίκτυο που χρησιμοποιεί απλά πρωτόκολλα δρομολόγησης, είναι εκτεθειμένο σε σοβαρούς κινδύνους. Η θέσπιση και μελέτη τέτοιου είδους μηχανισμών, ούτως ώστε να διασφαλισθεί η σωστή λειτουργία του δικτύου, κρίνεται απαραίτητη. Τέτοιοι μηχανισμοί ονομάζονται ***Μηχανισμοί Κινήτρων*** ή ***Μηχανισμοί Προώθησης Συνεργασίας***.

Η βασική ιδέα των μηχανισμών αυτών, είναι να δώσουν κίνητρα στους κόμβους ενός δικτύου να συμπεριφέρονται σύμφωνα με τους κανόνες των εκάστοτε πρωτοκόλλων. Τα κίνητρα αυτά, μπορούν να θεωρηθούν ως επιβραβεύσεις που παίρνουν την μορφή χρημάτων ή πόντων ή καλύτερης ποιότητας υπηρεσιών. Ακόμα περισσότερο, οι μηχανισμοί αυτοί είναι σχεδιασμένοι έτσι, ώστε οι εγωιστικοί ή κακόβουλοι κόμβοι που δεν θέλουν να συμμορφωθούν, να τιμωρούνται με τέτοιον τρόπο που τα οφέλη τους από την ανάρμοστη συμπεριφορά να μην έχουν αξία.

Οι Μηχανισμοί Προώθησης Συνεργασίας χωρίζονται σε δύο κύριες κατηγορίες: Τους ***Μηχανισμούς βασισμένους σε Αμοιβή (Credit Based)*** και τους ***Μηχανισμούς βασισμένους σε Φήμη (Reputation Based).*** Μια τρίτη

κατηγορία είναι οι ***Υβριδικοί Μηχανισμοί (Hybrid)*** που αποτελούν συνδυασμό των παραπάνω δύο μηχανισμών.

Οι Υβριδικοί Μηχανισμοί χρησιμοποιούν την Φήμη σε συνεργασία με την Αμοιβή και αυτό τους κάνει ξεχωριστούς. Συνήθως αναφέρονται σε υπάρχοντες μηχανισμούς Αμοιβής και Φήμης προσπαθώντας να διατηρήσουν και να αναπτύξουν τα θετικά στοιχεία του καθενός από αυτούς. Έχουν προταθεί αρκετοί μηχανισμοί αυτής της κατηγορίας στην διεθνή βιβλιογραφία[19],[20]. Για λόγους συντομίας όμως, δεν θα αναφερθούμε περαιτέρω σε αυτούς.

2.3. 1. Μηχανισμοί Βασισμένοι σε Αμοιβή (Credit Based)

Οι Μηχανισμοί Προώθησης Συνεργασίας που είναι βασισμένη στην Αμοιβή, χρησιμοποιούν χρηματικές ή άλλες αμοιβές για την κάθε ενέργεια δρομολόγησης στο δίκτυο. Ο κάθε κόμβος έχει στη διάθεσή του ένα αρχικό χρηματικό ποσό το οποίο θα βοηθήσει την επικοινωνία του με τους υπόλοιπους κόμβους του δικτύου. Η κάθε ενέργεια δρομολόγησης που τον καθιστά αποστολέα ή παραλήπτη (ανάλογα τον μηχανισμό) ενός μηνύματος, του αποσπά κάποια χρήματα. Από την άλλη, ο κάθε ενδιάμεσος κόμβος σε μία επικοινωνία, δέχεται ένα ποσό χρημάτων ως επιβράβευση για την προώθηση ενός πακέτου που καλείτε να μεταδώσει από κάποιον άλλο κόμβο. Συνήθως, ο μόνος τρόπος για έναν κόμβο να αυξήσει το ποσό των χρημάτων που έχει στη διάθεσή του, είναι η προώθηση των εισερχόμενων πακέτων όταν αυτό του ζητείται.

Γίνεται εύκολα κατανοητό, πως ένας κόμβος με εγωιστική ή κακόβουλη συμπεριφορά απειλείται να βρεθεί χωρίς χρήματα. Σε μια τέτοια περίπτωση, ο κόμβος απομονώνεται, αφού δεν μπορεί να στείλει ή να δεχτεί μηνύματα. Η επανάκαμψη ενός τέτοιου κόμβου στο δίκτυο είναι εύκολη, αρκεί ο ίδιος να αποφασίσει να σταματήσει την ανάρμοστη συμπεριφορά.

Έχουν προταθεί στη διεθνή βιβλιογραφία αρκετοί Μηχανισμοί Βασισμένοι σε Αμοιβή [11],[12],[13]. Ενδεικτικά παρουσιάονται δύο από αυτούς: ο μηχανισμός Sprite [3] και ο μηχανισμός Nuglets [4].

2.3.1.1. Sprite

A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks

Το Sprite είναι ένα απλό σύστημα που χρησιμοποιεί μηχανισμούς βασισμένους στην αμοιβή για την σωστή λειτουργία προώθησης συνεργασίας σε ένα ad-hoc δίκτυο που απειλείται από εγωιστικούς κόμβους. Περιγραφικά το σύστημα λειτουργεί ως εξής:

1. Όταν ένας κόμβος λαμβάνει ένα πακέτο, αυτός κρατάει μία απόδειξη για το πακέτο που παρέλαβε.
2. Στο δίκτυο υπάρχει μία υπηρεσία που ονομάζεται Credit Clearance Service (CCS) η οποία λειτουργεί ως τράπεζα, προωθούν σε αυτήν την υπηρεσία οι κόμβοι τις αποδείξεις που έλαβαν κατά την λήψη ή προώθηση πακέτων και στη συνέχεια αυτή πιστώνει τους «λογαριασμούς» τους με τους ανάλογους πόντους.
3. Μόλις ένας κόμβος έχει γρήγορη σύνδεση με κάποια CCS, τότε στέλνει τις αποδείξεις που έχει κρατήσει για να λάβει τους πόντους που του αναλογούν.

Τέλος, για λόγους ασφάλειας, χρησιμοποιείται κρυπτογραφία δημοσίου κλειδιού.

Περιγραφή του συστήματος

Κάθε κόμβος του συστήματος είναι προμηθευμένος με τα κατάλληλα network interfaces έτσι ώστε να του επιτρέπουν να στέλνει και να λαμβάνει μηνύματα μέσω ενός ασύρματου δικτύου επικάλυψης (wireless overlay network). Επίσης, το σύστημα Sprite χρησιμοποιεί το πρωτόκολλο DSR (Dynamic Source Routing) για την δρομολόγηση των πακέτων.

Όταν ένας κόμβος θέλει να στείλει ένα πακέτο, ο κόμβος αποστολέας θα χρεωθεί με κάποιους πόντους από το δίκτυο. Αυτό συμβαίνει γιατί οι κόμβοι που θα προωθήσουν το πακέτο στη συνέχεια, θέλουν κάποιους πόντους ως αμοιβή. Οι πόντοι που λαμβάνονται από τους κόμβους που προωθούν κάποιο πακέτο, χρησιμοποιούνται αργότερα για να στείλουν αυτοί κάποιο μήνυμα.

Υπάρχουν δύο τρόποι για έναν κόμβο να κερδίσει πόντους:

1. Ο κόμβος μπορεί να πληρώσει τα χρέη του ή να αγοράσει πόντους με αληθινά λεφτά. Η αναλογία αληθινών χρημάτων και πόντων εξαρτάται από την απόδοση του δικτύου εκείνη τη στιγμή.
2. Ο κόμβος να προωθεί τα πακέτα άλλων, το οποίο είναι και το επιθυμητό, χρησιμοποιώντας το σύστημα με τις αποδείξεις που περιγράψαμε παραπάνω. Οι αποδείξεις χρησιμοποιούνται ως τρόπος εξοικονόμησης χώρου και εύρους ζώνης στους κόμβους αλλά και στο δίκτυο, αφού είναι πολύ μικρότερες από το πραγματικό μήνυμα και περιέχουν όλες τις απαραίτητες πληροφορίες για την CCS υπηρεσία.

Το πλάνο πληρωμής

Υπάρχουν δύο τρόποι χρέωσης της μεταφοράς του μηνύματος, είτε να την χρεώνεται ο παραλήπτης, είτε να την χρεώνεται ο αποστολέας. Στο Sprite επιλέγεται η χρέωση του αποστολέα, αυτό συμβαίνει για λόγους ασφάλειας, καθώς κακόβουλοι κόμβοι μπορεί να στέλνουν σκόπιμα μηνύματα στον παραλήπτη, έτσι ώστε να πληρώνει αυτός χωρίς κάποιο όφελος. Επίσης, με την χρέωση του αποστολέα, κρατάς σε ύφεση το δίκτυο αφού αποτρέπεται η αποστολή άσκοπων μηνυμάτων από κακόβουλους κόμβους (όπως για παράδειγμα, αποτρέπεται η denial-of-service επίθεση).

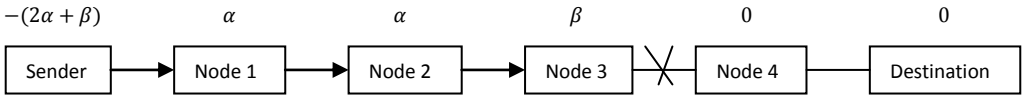
Στη συνέχεια, οι κόμβοι που πληρώνονται από το σύστημα είναι εκείνοι που προωθούν τα πακέτα. Στο σχήμα του Sprite, η CCS υπηρεσία θεωρεί ότι ένας κόμβος έχει προωθήσει ένα πακέτο και πρέπει να πληρωθεί, αν και μόνο αν ο επόμενος κόμβος το έχει λάβει.

Το επόμενο σημαντικότερο σημείο είναι το ποσό της χρέωσης και της πληρωμής. Στο συγκεκριμένο σύστημα σκοπός είναι να αποτρέπονται οι ενέργειες εξαπάτησης και να δίνεται το κίνητρο στους κόμβους να συνεργάζονται. Σύμφωνα με αυτή την ιδέα, το Sprite δεν χρησιμοποιεί σωστό ισοζύγιο πληρωμής-χρέωσης, δηλαδή η χρέωση του αποστολέα δεν είναι ίση με την πληρωμή των κόμβων που λαμβάνουν μέρος στην μετάδοση του μηνύματος. Πιο συγκεκριμένα, ο αποστολέας χρεώνεται περισσότερο από την συνολική πληρωμή των κόμβων που προωθούν. Το σύστημα ωστόσο έχει προβλέψει την αναπόφευκτη εκροή πόντων, έτσι περιοδικά η CCS υπηρεσία δίνει στους κόμβους συγκεκριμένη ποσότητα πόντων. Η ακριβής χρέωση και πληρωμή των κόμβων γίνεται βάση τριών κριτηρίων, την ώθηση των κόμβων

να προωθούν μηνύματα, την ώθηση των κόμβων να αναφέρουν τις αποδείξεις τους στην CCS υπηρεσία και την πρόληψη της αποστολής ψεύτικων αποδείξεων.

1. *Ώθηση κόμβων να προωθούν μηνύματα*

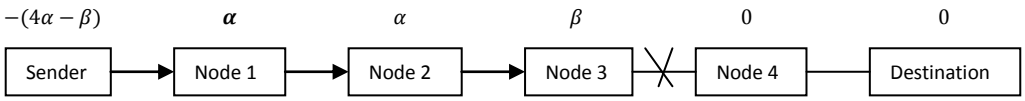
   Για την επιτυχή υλοποίηση του κριτηρίου αυτού δίνεται το παρακάτω σχέδιο. Αρχικά, η CCS υπηρεσία προσδιορίζει τον τελευταίο κόμβο στο μονοπάτι που έλαβε το μήνυμα. Στη συνέχεια, η υπηρεσία ζητάει από τον αποστολέα να πληρώσει τον τελευταίο κόμβο που έλαβε το μήνυμα β πόντους και όλους τους υπόλοιπους κόμβους που προώθησαν α πόντους, όπου $\alpha > \beta$. Προσέχουμε ότι η υπηρεσία δεν ζητάει από τον αποστολέα να πληρώσει τους επόμενους κόμβους από τον τελευταίο. Το σχέδιο αυτό, φαίνεται στο παρακάτω σχήμα:



2. *Ώθηση κόμβων να αναφέρουν τις αποδείξεις στην CCS υπηρεσία*

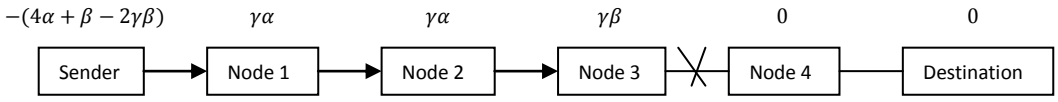
   Εξίσου σημαντικό σημείο για την σωστή λειτουργία του συστήματος είναι όλοι οι κόμβοι να αναφέρουν τις αποδείξεις τους. Γι' αυτό το λόγο, αρχικά το β πρέπει να είναι μεγαλύτερο από το κόστος ενός κόμβου να αναφέρει την απόδειξη στην υπηρεσία.
   Εξετάζοντας το παραπάνω σχήμα, είναι εμφανές ότι εάν ο τελευταίος κόμβος (είτε οι k τελευταίοι κόμβοι) , δεν αναφέρουν την απόδειξη, τότε ο αποστολέας κερδίζει α πόντους (ή $k\alpha$ πόντους αντίστοιχα). Αυτό μπορεί να συμβεί εάν, για παράδειγμα, ο αποστολέας συμφωνήσει με τον τελευταίο κόμβο να μην στείλει την απόδειξή του, δίνοντάς του γι αυτήν του την πράξη ένα ποσό ίσο με $e\beta < \alpha$, όπου $e > 0$. Έτσι ο αποστολέας και ο τελευταίος κόμβος κερδίζουν πόντους περισσότερους από εκείνους που θα κέρδιζαν αν τηρούσαν τους όρους συναλλαγής.
   Για να αποφευχθεί η παραπάνω ενέργεια εξαπάτησης, η CCS υπηρεσία χρεώνει τον αποστολέα ένα επιπλέον ποσό πόντων, εάν ο προορισμός δεν αναφέρει την απόδειξη λήψης του μηνύματος. Το ποσό αυτό δεν δίνεται σε κάποιο κόμβο, αλλά το κρατάει η ίδια η

υπηρεσία. Η όλη χρέωση του αποστολέα, πρέπει να είναι $k\beta$ λιγότερη από την χρέωση εάν το μήνυμα φτάσει στον προορισμό του, όπου k είναι ο αριθμός των αποδείξεων που δεν έχουν αναφερθεί. Σύμφωνα με το παραπάνω σχήμα, η εφαρμογή της χρέωσης έχει ως εξής:



3. *Πρόληψη αποστολής ψεύτικων αποδείξεων*

Μία ακόμη ενέργεια εξαπάτησης είναι η αποστολή ψεύτικων αποδείξεων από τους κόμβους. Στην περίπτωση αυτή, οι ενδιάμεσοι κόμβοι φαίνεται να συνωμοτούν στην αποστολή ψεύτικων αποδείξεων , χωρίς ουσιαστικά να προωθούν το μήνυμα. Για την αποφυγή τέτοιων ενεργειών, το Sprite προβλέπει ο παραλήπτης να στέλνει απόδειξη στην CCS εφόσον λάβει το σωστό μήνυμα. Η μη αποστολή απόδειξης από τον παραλήπτη, έχει ως αποτέλεσμα την αλλαγή της χρέωσης του αποστολέα, όπως περιγράψαμε παραπάνω, αλλά και την μείωση της πληρωμής των ενδιάμεσων κόμβων. Πιο συγκεκριμένα, η πληρωμή των ενδιάμεσων κόμβων μειώνεται στο ποσό των $\gamma\alpha$ μονάδων, όπου $\gamma < 1$. Το τελικό σχήμα του πλάνου πληρωμής είναι το παρακάτω, σε συνάρτηση με το παράδειγμα που αναφέραμε παραπάνω.



Προδιαγραφές του πρωτοκόλλου προώθησης μηνυμάτων

Θεωρούμε ότι κάθε κόμβος n_i έχει ένα δημόσιο και ένα ιδιωτικό κλειδί (PK_i, SK_i). Επίσης ο κάθε κόμβος n_i κρατάει ένα πίνακα seq_i, όπου $seq_i(j,k)$ είναι ο αριθμός σειράς των μηνυμάτων του αποστολέα n_j στον παραλήπτη n_k και ο κόμβος n_i έχει συμμετάσχει σε αυτήν την μεταφορά. Επίσης θεωρούμε πως η ψηφιακή υπογραφή του συστήματος είναι η $(sign_{SK}(\ \ ), verify_{PK}(\ \ ))$.

Όταν ένας κόμβος στέλνει ένα μήνυμα, αυτό περιέχει και τις ακόλουθες πληροφορίες $(m, p, seq_0(0,d), s)$, όπου m το φορτίο του μηνύματος, p η διαδρομή που πρέπει να ακολουθήσει, $seq_0(0,d)$ ο σειριακός αριθμός όπως περιγράψαμε παραπάνω και s η ψηφιακή υπογραφή.

Όταν ένας κόμβος λαμβάνει ένα μήνυμα εξετάζει τις παρακάτω τρείς πληροφορίες:

1. Αν ο κόμβος αυτός βρίσκεται μέσα στο μονοπάτι της διαδρομής
2. Αν ο σειριακός αριθμός του μηνύματος είναι μεγαλύτερος από τον αρχικό, δηλαδή από τον $seq_i(0,d)$.
3. Αν η ψηφιακή υπογραφή είναι έγκυρη.

Εάν δεν ικανοποιούνται όλα τα παραπάνω, τότε το μήνυμα απορρίπτεται. Αντίθετα, εάν ικανοποιούνται όλα τα παραπάνω, τότε ο κόμβος κρατάει ως απόδειξη το $(MD(m), p, seq_0(0,d), s)$, όπου $MD(m)$ είναι η συνάρτηση κατακερματισμού που χρησιμοποιείτε (όπως πχ η MD5). Εάν ο κόμβος αυτός δεν είναι προορισμός, τότε προωθεί το πακέτο στέλνοντας τις αντίστοιχες (m, p, seq, s) πληροφορίες στον επόμενο κόμβο.

Τέλος, η υπηρεσία CCS ελέγχει αν η απόδειξη που λαμβάνει από κάποιο κόμβο είναι σωστή, αν το δημόσιο κλειδί του αποστολέα περιέχεται σωστό σε αυτήν.

Συνεργασία κατά την διαδικασία εύρεσης διαδρομών

Το Sprite θέλοντας να παρακινήσει τους κόμβους να συνεργαστούν ακόμη και κατά τη διαδικασία εύρεσης διαδρομών προτείνει ένα διαφορετικό σχήμα που μπορεί να χρησιμοποιηθεί σαν βελτιστοποίηση του DSR πρωτοκόλλου δρομολόγησης. Ως γνωστό, το DSR πρωτόκολλο για την διαδικασία εύρεσης διαδρομών χρησιμοποιεί μηνύματα ROUTE REQUEST και ROUTE REPLY, στο πλάνο βελτιστοποίησης χρειάζεται μόνο η αποφυγή της επαναληπτικής μετάδοσης ενός ROUTE REQUEST.

Περιγραφικά, όταν στέλνεται ένα ROUTE REQUEST, το μήνυμα αυτό περιέχει την διεύθυνση του αποστολέα και έναν σειριακό αριθμό. Στη συνέχεια ο κόμβος υπογράφει και στέλνει το πακέτο, αυξάνοντας τον σειριακό του αριθμό κατά 1. Όταν ένας κόμβος λαμβάνει ένα μήνυμα τύπου ROUTE REQUEST αρχικά ελέγχει εάν το μήνυμα είναι αυθεντικό ελέγχοντας τον σειριακό του αριθμό. Ο κόμβος στη συνέχεια αποθηκεύει το ROUTE REQUEST

για να πληρωθεί στο μέλλον. Αν ένας κόμβος αποφασίσει να αναμεταδώσει το ROUTE REQUEST, επισυνάπτει σε αυτό την δική του διεύθυνση και το υπογράφει. Εν συνεχεία, όταν έρθει η σειρά της CCS υπηρεσίας να διεκπεραιώσει τις συναλλαγές, το μήνυμα ROUTE REQUEST εγκρίνεται μόνο όταν όλες οι υπογραφές που περιέχει το μήνυμα είναι αυθεντικές. Επίσης, εάν ένα ROUTE REQUEST που υποβάλλεται από έναν κόμβο είναι μέρος ενός άλλου ROUTE REQUEST που είχε υποβληθεί από τον ίδιο κόμβο, τότε το μήνυμα αυτό απορρίπτεται.

Τελικά, η υπηρεσία με τα στοιχεία που έχει, κατασκευάζει ένα δέντρο από τα μηνύματα που έγιναν δεκτά. Ο αποστολέας πληρώνει α σε κάθε κόμβο που δεν βρίσκεται στα κλαδιά του δέντρου, και β σε κάθε κόμβο κλαδί του δέντρου. Για κάθε κόμβο έξω από το δέντρο, ο αποστολέας πληρώνει $\alpha - \beta$ στην υπηρεσία CCS.

Αξιολόγηση του πρωτοκόλλου

Στην προσπάθεια αξιολόγησης του πρωτοκόλλου κοιτάζουμε τα αποτελέσματα των προσομοιώσεων που έχουν γίνει. Σύμφωνα με αυτά, η απόδοση του δικτύου σύμφωνα με τα πακέτα που προωθούνται επιτυχώς, είναι πολύ καλή. Πιο συγκεκριμένα ο λόγος προώθησης πακέτων τείνει στο 1, κάτι που σημαίνει ότι έχει περίπου 100% επιτυχία. Ακόμα και όταν η κίνηση του δικτύου αυξηθεί, δηλαδή στέλνονται περισσότερα πακέτα, ο λόγος των πακέτων που προωθούνται επιτυχώς είναι επίσης κοντά στο 1. Επίσης, όταν η ενέργεια της μπαταρίας των κόμβων είναι χαμηλή, το πρωτόκολλο εξακολουθεί να δίνει ικανοποιητικά αποτελέσματα. Τέλος, φαίνεται ο σκοπός του πρωτοκόλλου να επιτυγχάνεται αφού οι κόμβοι δείχνουν να συνεργάζονται μεταξύ τους ακόμη και κατά τη διάρκεια της διαδικασίας εύρεσης διαδρομών.

Πέρα όμως από τα εντυπωσιακά αποτελέσματα των προσομοιώσεων, παρατηρούνται και κάποια αρνητικά στοιχεία. Το Sprite απαιτεί μεγάλο αριθμό πληροφορίας ελέγχου κάτι που οδηγεί στην κατανάλωση περισσότερης ενέργειας από τους κόμβους, κάτι το οποίο είναι πολύ αρνητικό, αφού βασικό στοιχείο των πρωτοκόλλων των Ad-hoc δικτύων είναι οι χαμηλές απαιτήσεις σε ενέργεια. Επίσης, έχει τοπολογικές απαιτήσεις, αφού κάποιος ακριανός κόμβος θα βρίσκεται σε μειονεκτικότερη θέση να προωθήσει πακέτα λόγω του αυξημένου κόστους.[3]

2.3.1.2. Nuglets

a Virtual Currency to Stimulate Cooperation in Self-Organized Mobile Ad Hoc Networks

Το Nuglets είναι ένα πρωτόκολλο με μηχανισμό βασισμένο στην αμοιβή. Τα nuglets, όπως ονομάζονται, είναι τα εικονικά χρήματα που χρησιμοποιούνται για τις συναλλαγές στο σύστημα, έτσι πήρε και το όνομά του το πρωτόκολλο. Χρησιμοποιεί μηχανισμούς για να χρεώνει/αμείβει υπηρεσίες αποστολής/προώθησης. Οι κόμβοι που χρησιμοποιούν πόρους του δικτύου πρέπει να πληρώνουν για αυτούς σε nuglets στους κόμβους που προσφέρουν τις υπηρεσίες τους. Αυτό κάνει τα εικονικά χρήματα απαραίτητα και γι' αυτό ο κάθε κόμβος ενδιαφέρεται να αυξήσει τα εισοδήματά του. Επίσης, ο μηχανισμός αυτός ενθαρρύνει τους χρήστες των κόμβων να κρατούν ενεργοποιημένες τις συσκευές τους (έτσι ώστε να χρησιμοποιούνται σαν προωθητικοί κόμβοι) ακόμα και όταν δεν περιμένουν να στείλουν ή να δεχτούν μηνύματα.

Περιγραφή του συστήματος

1.Περιγραφή του τρόπου των Οικονομικών Συναλλαγών

Το πρωτόκολλο Nuglets επιλέγει να χρησιμοποιήσει ένα μοντέλο που ονομάζεται Packet Purse Model ή PPM (Μοντέλο Οικονομικά Ανεξάρτητου Πακέτου). Σε αυτό το μοντέλο ο αποστολέας του πακέτου πληρώνει για την μετάδοση του. Οι συναλλαγές μεταξύ των κόμβων γίνονται με τον παρακάτω τρόπο: Όταν ένας κόμβος στέλνει ένα πακέτο, αυτός φορτώνει το πακέτο με έναν αριθμό nuglets (εικονικών χρημάτων) το οποίο εκτιμάει ότι αρκεί για να φτάσει στον προορισμό του. Ο κάθε επόμενος κόμβος που εμπλέκεται στην μετάδοση τραβάει από το πακέτο όσα nuglets χρειάζεται για να προσφέρει τις υπηρεσίες του, σύμφωνα πάντα με το κόστος που έχει γι' αυτόν η προώθηση του πακέτου. Ο ακριβής αριθμός της χρέωσης κάθε υπηρεσίας προώθησης εξαρτάται από πολλά πράγματα, κάποια από αυτά είναι η ενέργεια που θα καταναλώσει ο κόμβος για να προωθήσει το πακέτο, η κατάσταση της μπαταρίας του εκείνη τη χρονική στιγμή και ο αριθμός των nuglets του εκείνη τη στιγμή. Αν ένα πακέτο δεν έχει αρκετά εικονικά χρήματα για να δώσει στον κόμβο έτσι ώστε αυτός να το προωθήσει, τότε το πακέτο απορρίπτεται.

Οι παραπάνω λόγοι όμως κάνουν δύσκολο στον αποστολέα να υπολογίσει με πόσα ακριβώς χρήματα πρέπει να εφοδιάσει το πακέτο έτσι ώστε να είναι σίγουρο ότι αυτό θα φτάσει στον προορισμό του. Ιδανικά, ο κάθε κόμβος που προωθεί ένα πακέτο χρεώνει το λιγότερο ποσό χρημάτων που μπορεί για να ικανοποιήσει τις ανάγκες του. Αυτό το ποσό καθορίζεται ,πρώτον, σύμφωνα με την ενέργεια που ο κόμβος θα καταναλώσει για την προώθηση (για παράδειγμα, ένας κόμβος που πρέπει να προωθήσει ένα πακέτο σε έναν μακρινό για αυτόν κόμβο πρέπει να πληρωθεί περισσότερο για τις υπηρεσίες του) ,και δεύτερον, με την κατάσταση της μπαταρίας του εκείνη τη στιγμή (για παράδειγμα, κόμβοι που έχουν χαμηλότερη μπαταρία μπορεί να ζητούν περισσότερα χρήματα για την μετάδοση). Αλλά τι εμποδίζει έναν κόμβο από το να απαιτήσει όλο το ποσό των χρημάτων με το οποίο είναι εφοδιασμό το πακέτο; Για να λυθεί αυτό το πρόβλημα οι σχεδιαστές επιλέγουν δύο τρόπους που μπορούν να προστεθούν στο αρχικό PPM μοντέλο:

1. *PPM με συγκεκριμένη χρέωση ανά προώθηση*: Ο αποστολέας του πακέτου καθορίζει μία συγκεκριμένη τιμή ανά προώθηση u πριν το στείλει, αυτόν τον αριθμό u τον αναφέρει μέσα στο πακέτο επιπρόσθετα με τον αριθμό των nuglets. Ένας μηχανισμός ασφάλειας (που θα συζητηθεί αργότερα) εξασφαλίζει ότι κάθε κόμβος που θα συμμετάσχει στην μετάδοση θα πάρει ακριβώς το ποσό των u nuglets. Με αυτόν τον τρόπο, η χρέωση είναι ανεξάρτητη της ενέργειας ή της μπαταρίας ή οποιουδήποτε άλλου περιορισμού. Ωστόσο, ένας κόμβος μπορεί να αρνηθεί να προωθήσει το πακέτο, αν θεωρήσει πως το ποσό των u nuglets δεν καλύπτει τις ανάγκες προώθησης.
2. *PPM με χρήση Πλειστηριασμών*: Σε αυτήν την προσέγγιση, ο κάθε προωθητικός κόμβος, αλλά και ο ίδιος ο αποστολέας, εκτελεί μια διαδικασία πλειστηριασμού με κλειστές προσφορές για να αποφασίσει σε ποιόν κόμβο θα στείλει το πακέτο στη συνέχεια. Οι πλειοδότες είναι οι πιθανοί επόμενοι κόμβοι που θα μεταδώσουν το πακέτο. Ο κάθε πλειοδότης b_i δίνει μία τιμή p_i με την οποία είναι διατεθειμένος να προωθήσει το πακέτο και την στέλνει σφραγισμένη στον κόμβο που εκτελεί τον πλειστηριασμό. Όταν ο προωθητικός κόμβος λάβει όλες τις προσφορές, επιλέγει τον νικητή του πλειστηριασμού. Ο νικητής της διαδικασίας αυτής , b_j είναι εκείνος που έδωσε την χαμηλότερη προσφορά, $p_j = min_i p_i$ (αν οι νικητές του πλειστηριασμού είναι παραπάνω από έναν, τότε ο νικητής επιλέγεται τυχαία). Εάν η δεύτερη χαμηλότερη προσφορά έχει την τιμή

$p_k = min_{i,\ i \neq j}\ p_i$, ο προωθητικός κόμβος εφοδιάζει το πακέτο με p_k nuglets και το στέλνει στον b_j . Ένας μηχανισμός ασφάλειας εξασφαλίζει ότι ο b_j θα λάβει το ποσό των $p_k > p_j$ αν προωθήσει το πακέτο ασφαλές στον επόμενο κόμβο.

Πρέπει να σημειωθεί ότι κατά την διάρκεια των πλειστηριασμών οι πλειοδότες δεν μπορούν να γνωρίζουν πόσοι κόμβοι δίνουν προσφορές. Πιο συγκεκριμένα, ένας κόμβος πλειοδότης δεν ξέρει εάν είναι ο μόνος που δίνει προσφορά ή αν υπάρχουν και άλλοι κόμβοι που τον ανταγωνίζονται. Με αυτόν τον τρόπο εξασφαλίζουμε πως ο κάθε κόμβος δίνει την καλύτερη προσφορά που μπορεί, σύμφωνα πάντα με τα συμφέροντά του.

2.Περιγραφή της διαδικασίας προώθησης

Το πρωτόκολλο κάνει χρήση ενός γεωδαιτικού αλγόριθμου προώθησης πακέτων [5]. Ο αλγόριθμος αυτός ενεργεί περιγραφικά ως εξής: Κάθε κόμβος γνωρίζει την δική του γεωγραφική θέση στο δίκτυο καθώς επίσης και τις γεωγραφικές θέσεις των γειτόνων του. Επίσης, ο αποστολέας ενός μηνύματος γνωρίζει την γεωγραφική θέση του προορισμού και την επισυνάπτει στην κεφαλίδα του πακέτου που στέλνει. Στη συνέχεια, επιλέγει ανάμεσα στους γειτονικούς του κόμβους αυτόν που βρίσκεται πιο κοντά στον κόμβο προορισμό και στέλνει το πακέτο. Εφόσον το πακέτο περιέχει τις συντεταγμένες του κόμβου προορισμού, κάθε προωθητικός κόμβος εκτελεί την ίδια διαδικασία με τον αποστολέα για να στείλει το πακέτο. Στην περίπτωση που κάποιος προωθητικός κόμβος δεν βρει κάποιον γειτονικό που να βρίσκεται κοντά στον προορισμό, τότε αυτός απορρίπτει το πακέτο.

3.Περιγραφή της μονάδας ασφάλειας

Η μονάδα ασφάλειας του συστήματος αποθηκεύει τα παρακάτω μακροπρόθεσμα δεδομένα: ένα αναγνωριστικό του όλου συστήματος, τον αριθμό των nuglets του κόμβου, το ιδιωτικό κλειδί της μονάδας, το πιστοποιητικό του δημοσίου κλειδιού της μονάδας ασφάλειας το οποίο εκδίδεται από τον κατασκευαστεί του συστήματος, και το δημόσιο κλειδί του κατασκευαστή της μονάδας.

Σε αυτό το σημείο, πρέπει να αναφέρουμε πως μία μονάδα ασφάλειας υπάρχει ανεξάρτητη μέσα σε κάθε κόμβο. Επίσης, η μονάδα αυτή, κρατάει

έναν πίνακα στον οποίο κάθε δεδομένο αντιστοιχεί σε μία γειτονική μονάδα ασφάλειας και περιέχει τα ακόλουθα πεδία:

- Αναγνωριστικό : Το αντίστοιχο αναγνωριστικό των συστημάτων των γειτονικών του κόμβων.
- Κλειδί συνεδρίασης : Όταν ο κόμβος της μονάδας ασφάλειας A γίνει γείτονας με τον κόμβο της μονάδας ασφάλειας B, τρέχουν μεταξύ τους ένα hello πρωτόκολλο. Χρησιμοποιώντας το, η A και η B μονάδα εγκαθιστούν ένα συμμετρικό κλειδί συνεδρίασης k_{AB}. Αυτό το κλειδί χρησιμοποιείται για να προστατεύσει τα εικονικά χρήματα των πακέτων που στέλνονται μεταξύ τους.
- Αριθμός σειράς : Εκτός από το κλειδί συνεδρίασης, η A και B μονάδα χρησιμοποιεί έναν αριθμό σειράς κάθε φορά που στέλνει ή δέχεται μηνύματα, αυτό συμβαίνει για να εντοπίζονται τυχόν επαναλαμβανόμενα πακέτα. Κάθε φορά που ο κόμβος στέλνει ένα πακέτο αυξάνει τον υπάρχον αριθμό σειράς κατά 1.
- Μετρητή χρέους : Τέλος, η μονάδα ασφάλειας A κρατάει έναν μετρητή $d_{A\to B}$ που συνδέεται με τη μονάδα B, ο οποίος μετράει το χρέος του κόμβου A στον κόμβο B. Αυτός ο μετρητής κρίνεται αναγκαίος για τον εξής λόγο: Όταν ο A κόμβος προωθεί ένα πακέτο στον B, ο A δεν αυξάνει τον μετρητή των nuglet του κατευθείαν, περιμένει την απάντηση του B για να είναι σίγουρος πως το πακέτο έχει επιτυχώς προωθηθεί. Ο κόμβος A και ο κόμβος B πληρώνουν τα χρέη τους ο ένας στον άλλον, κάθε φορά που τρέχουν το hello πρωτόκολλο, κάτι το οποίο συμβαίνει σε τακτά χρονικά διαστήματα.

Στην περίπτωση της χρήσης του μοντέλου PPM με πλειστηριασμό, η μονάδα ασφάλειας επίσης διατηρεί τους πράκτορες των κόμβων που συμμετέχουν στους πλειστηριασμούς. Αυτοί οι πράκτορες συναλλάσσονται και κρατούν ενήμερες τις καταστάσεις τους μέσο του hello πρωτοκόλλου. Επιπροσθέτως, η μονάδα ασφαλείας διατηρεί πληροφορίες δρομολόγησης.

Ανάλυση του συστήματος

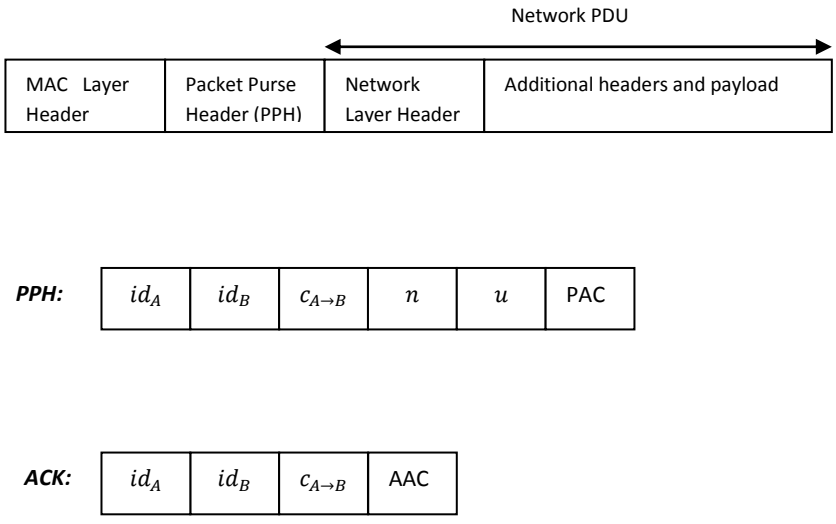
1.Η υλοποίηση του μοντέλου του οικονομικά ανεξάρτητου πακέτου

Σύμφωνα με το μοντέλο του οικονομικά ανεξάρτητου πακέτου(PPM), το κάθε πακέτο πρέπει να είναι εφοδιασμένο με έναν αριθμό nuglets που είναι

απαραίτητα για την μετάδοσή του. Αυτά τα εικονικά χρήματα αποθηκεύονται στην Packet Purse Header (PPH), η οποία είναι μία επιπρόσθετη κεφαλίδα στο πακέτο μεταξύ των κεφαλίδων του επιπέδου MAC και του επιπέδου δικτύου. Η PPH κεφαλίδα δημιουργείται από την μονάδα ασφάλειας του αποστολέα και αναδημιουργείται από τις μονάδες ασφαλείας κάθε επόμενου κόμβου που συμμετάσχει στην μετάδοση. Η κεφαλίδα αυτή προστατεύεται με κρυπτογραφικές μεθόδους με σκοπό να αποτρέπεται τυχόν πλαστογραφία των nuglets ή κάποια άλλη αθέμιτη αλλαγή κατά τη διάρκεια της μετάδοσης.

Η κεφαλίδα PPH περιλαμβάνει το αναγνωριστικό της μονάδας ασφαλείας του A, το αναγνωριστικό της μονάδας ασφαλείας του επόμενου κόμβου B, τον σειριακό αριθμό $c_{A\to B}$, τον αριθμό των nuglets n που περιέχονται στο πακέτο, τον αριθμό των nuglets u που επιτρέπεται ο επόμενος κόμβος B να πάρει, και τέλος ένα κωδικό αυθεντικότητας των nuglets (Purse Authentication Code, PAC), ο οποίος υπολογίζεται από τα πεδία της κεφαλίδας PPH και ενός κρυπτογραφημένου και κατακερματισμένου κώδικα των περιεχομένων του πακέτου με την χρήση μίας συνάρτησης κατακερματισμού g με το κλειδί συνεδρίασης k_{AB} μεταξύ του A και B κόμβου.

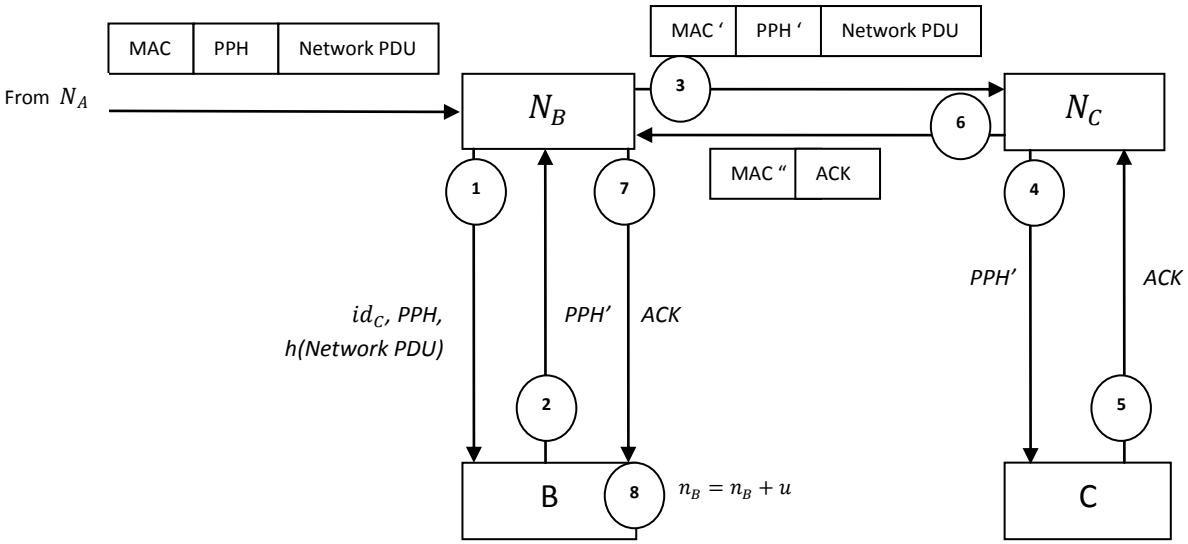
Όταν ένας κόμβος δέχεται ένα πακέτο από κάποιον άλλο κόμβο, τότε αυτός είναι υποχρεωμένος να στείλει ένα πακέτο επιβεβαίωσης λήψης (Acknowledgement, ACK). Το ACK υπολογίζεται από την μονάδα ασφάλειας B του κόμβου και περιέχει το αναγνωριστικό της μονάδας ασφαλείας A του προηγούμενου κόμβου, το αναγνωριστικό του B, τον σειριακό αριθμό της αποστολής $c_{A\to B}$ που είχε λάβει προηγουμένως από την PPH, και έναν κωδικό επιβεβαίωσης λήψης (Acknowledgment Authentication Code, AAC).



Η Packet Purse Header (PPH) και το Acknowledgement (ACK)

1.Η υλοποίηση του πρωτοκόλλου προώθησης

Αρχικά παρουσιάζεται αναλυτικά η περιγραφή του μοντέλου PPM με συγκεκριμένη χρέωση ανά προώθηση. Για την διευκόλυνση της περιγραφής παρατίθεται η παρακάτω απεικόνιση:



Απεικόνιση του Πρωτοκόλλου Προώθησης

Έστω, ο κόμβος N_B έχει λάβει ένα πακέτο από τον κόμβο N_A και ο κόμβος που θα προωθήσει στη συνέχεια το πακέτο αυτό είναι ο N_C . Ο κόμβος N_B είναι διατεθειμένος να προωθήσει το πακέτο στον επόμενο κόμβο N_C με την τιμή που βρίσκεται στην κεφαλίδα PPH του πακέτου. Για να μπορέσει να πάρει τα nuglets που του αναλογούν από το πακέτο, ο N_B πρέπει να περάσει την κεφαλίδα PPH στην μονάδα ασφάλειας B μαζί με το αναγνωριστικό του κόμβου N_C που πρόκειται να προωθηθεί στη συνέχεια το πακέτο καθώς και την κρυπτογραφημένη και κατακερματισμένη τιμή του περιεχομένου του πακέτου (h (Network PDU)).

Η μονάδα B αρχικά επικυρώνει την PPH κεφαλίδα ελέγχοντας εάν ο σειριακός αριθμός αυτής είναι μεγαλύτερος κατά ένα από τον σειριακό αριθμό $c_{B\leftarrow A}$ που σχετιζόταν με τον A. Εάν ισχύει, τότε το PPH δεν είναι επανάληψη κάποιου άλλου και η B προχωράει καθορίζοντας τον σειριακό αριθμό $c_{B\leftarrow A}$

που της στάλθηκε. Η Β επίσης, επικυρώνει την αυθεντικότητα του PPH από την κρυπτογραφημένη και κατακερματισμένη τιμή του πακέτου.
Αφού ο Β τελειώσει επιτυχώς την επικύρωση του εισερχόμενου πακέτου, υπολογίζει μία καινούργια PPH (στο σχήμα μας την PPH ‘) η οποία εν συνεχεία θα σταλεί στον N_C . Αφού λοιπόν κρατήσει ένα αντίγραφο της PPH ο ίδιος, επισυνάπτει την κεφαλίδα στο πακέτο και την προωθεί στον κόμβο N_C . Όταν το πακέτο σταλεί στον N_C , τότε εκείνος στέλνει την κεφαλίδα στην μονάδα ασφάλειάς του C και εκείνη αμέσως κατασκευάζει ένα ACK για την Β που επιβεβαιώνει ότι το πακέτο λήφθηκε σωστά. Εν συνεχεία το στέλνει στον N_B .
Όταν ο N_B λάβει το ACK και το στείλει στην Β, τότε η Β ψάχνει στην μνήμη της να βρει το PPH που ταιριάζει με το ACK ελέγχοντας το αναγνωριστικό της C και τον αριθμό σειράς. Όταν και άμα το PPH βρεθεί, τότε η μονάδα Β μπορεί να πληρωθεί τα nuglets που της αναλογούν και να σβήσει το PPH από τη μνήμη της.

Στην περίπτωση του μοντέλου PPM με χρήση πλειστηριασμών, η μονάδα ασφαλείας εκτελεί τον πλειστηριασμό μεταξύ των πρακτόρων των πλειοδοτών κόμβων και όταν έχει τα αποτελέσματα, κατασκευάζει την PPH επικεφαλίδα σύμφωνα πάντα με τα αποτελέσματα της διαδικασίας πλειστηριασμού. Η υπόλοιπη διαδικασία μένει η ίδια με την παραπάνω περίπτωση του PPM με συγκεκριμένη χρέωση ανά προώθηση.

Αξιολόγηση του πρωτοκόλλου

Σύμφωνα με προσομοιώσεις που έχουν γίνει, συγκρίνοντας αποτελέσματα προσομοιώσεων χωρίς την εφαρμογή του Nuglets, με το Nuglets και την PPM με σταθερή χρέωση ανά προώθηση, και το Nuglets με πλειστηριασμούς, μπορούμε να προχωρήσουμε στα παρακάτω συμπεράσματα για το πρωτόκολλο αυτό. Αρχικά, φαίνεται πως η απόδοση του δικτύου είτε χρησιμοποιώντας το nuglets είτε όχι είναι περίπου σταθερή. Επίσης, στην περίπτωση όπου η μπαταρία των κόμβων του δικτύου είναι χαμηλή , το Nuglets φαίνεται να πέφτει πολύ σε απόδοση σε σχέση με ένα απλό πρωτόκολλο. Ένα ακόμα μειονέκτημα του πρωτοκόλλου, είναι η σχετικά μεγάλες κεφαλίδες που χρησιμοποιεί, οι οποίες σπαταλούν ενέργεια και πόρους του δικτύου. Επίσης, οι διαδικασίες κρυπτογράφησης και κατακερματισμού που χρησιμοποιεί για την ασφάλεια του συστήματος, σπαταλούν επίσης ενέργεια από τους κόμβους. [4]

2.3.2. Μηχανισμοί Βασισμένοι σε Φήμη (Reputation Based)

Οι Μηχανισμοί Προώθησης Συνεργασίας που βασίζονται στην Φήμη, χρησιμοποιούν την κριτική ως τιμωρία προς τους ανάρμοστους κόμβους. Ο κάθε κόμβος έχει αποθηκευμένη μία Λίστα Βαθμολογίας Συμπεριφοράς όλων των υπόλοιπων κόμβων του δικτύου. Κάθε φορά που στέλνει ένα πακέτο, ελέγχει εάν ο επόμενος το προωθεί και αναλόγως ενημερώνει την λίστα του. Έτσι κατατάσσει τους κόμβους σε δύο βασικές κατηγορίες, τους κόμβους που μπορεί να εμπιστεύεται και τους κόμβους που πρέπει να αποφεύγει. Πέρα από τις δικές του κριτικές, ο κάθε κόμβος μπορεί να ενημερώνεται και από τις Λίστες Βαθμολογίας Συμπεριφοράς των υπόλοιπων κόμβων του δικτύου, αφού δίνεται η δυνατότητα στους κόμβους να ανταλλάσουν απόψεις και να συμβουλεύουν ο ένας τον άλλο.

Σε ένα τέτοιο περιβάλλον, ένας κακόβουλος ή εγωιστικός κόμβος θα παίρνει κακές κριτικές από τους κόμβους θύματα της συμπεριφοράς του. Πέρα από το ότι οι υπόλοιποι κόμβοι θα αποφεύγουν να προσθέσουν έναν τέτοιο κόμβο στα μονοπάτια τους -κάτι που δεν είναι απαραίτητα κακό για τον ανάρμοστο κόμβο- θα τον τιμωρήσουν απορρίπτοντας τυχόν πακέτα που πηγάζουν από αυτόν ή προορίζονται για αυτόν. Αυτό θα τον οδηγήσει σταδιακά στην απομόνωση, καθώς η φήμη του θα εξαπλώνεται σε ολόκληρο το δίκτυο.

Έχουν προταθεί αρκετοί Μηχανισμοί Βασισμένοι σε Φήμη [14],[15],[16],[17]. Ενδεικτικά παρουσιάζονται δύο εξ’ αυτών: ο μηχανισμός SORI [7] και ο μηχανισμός CONFIDANT [6].

2.3.2.2. SORI

A Secure and Objective Reputation-based Incentive Scheme for Ad-hoc Networks

Το SORI είναι ένα σχέδιο το οποίο ενθαρρύνει την προώθηση πακέτων και δαμάζει την εγωιστική συμπεριφορά των κόμβων. Βρίσκεται και αυτό στην κατηγορία των μηχανισμών που χρησιμοποιούν την φήμη ως μέτρο σύγκρισης της πειθαρχίας. Η διαφορετικότητα του SORI έγκειται στο ότι η φήμη του κάθε κόμβου επιμετρείται με αντικειμενικά κριτήρια, και η διάδοση της φήμης διασφαλίζεται με μία μονόδρομη συνάρτηση κατακερματισμού (one-way-hash-chain).

Τα χαρακτηριστικά που πλαισιώνουν το SORI είναι τα εξής:

1. Η φήμη του κάθε κόμβου επιμετρείται με αντικειμενικά κριτήρια
2. Η διάδοση της φήμης διασφαλίζεται από μία μονόδρομη συνάρτηση κατακερματισμού
3. Η φήμη ενός κόμβου διαδίδεται μόνο μεταξύ των γειτόνων ,και όχι σε όλο το δίκτυο, κάτι που μειώνει το overhead της επικοινωνίας.
4. Οι εγωιστικοί κόμβοι τιμωρούνται.

Το SORI στηρίζει το σύστημά του στις παρακάτω παραδοχές:

1. Στο δίκτυο οι εγωιστικοί κόμβοι είναι εκείνοι που δεν προωθούν πακέτα εκτός εάν έχουν κάτι να κερδίσουν από αυτό.
2. Δεν υπάρχει κάποια συνωμοσία μεταξύ των κόμβων
3. Τα πακέτα κατά την προώθηση λαμβάνονται από όλους τους γείτονες που βρίσκονται εντός εμβέλειας του κόμβου που μεταδίδει
4. Όλοι οι κόμβοι που συμμετέχουν έχουν την θέληση να επικοινωνούν με κάποιους άλλους.
5. Όλοι οι κόμβοι έχουν μία και μοναδική ταυτότητα για όλη τους τη ζωή
6. Έχουμε μόνο εγωιστικούς κόμβους και όχι κακόβουλους, δηλαδή ένας κόμβος απορρίπτει ένα πακέτο για να μη σπαταλήσει ενέργεια ή υπολογιστική ισχύ, αλλά δεν θα προσπαθήσει να πλήξει το υπόλοιπο δίκτυο.
7. Ο κάθε κόμβος μπορεί να ακούσει κάθε πακέτο που μεταδίδεται από τους γείτονές του, ακόμη και αν δεν προορίζεται γι' αυτόν, και ταυτόχρονα μπορεί να δει ποιός προώθησε το πακέτο αυτό.

Περιγραφή του συστήματος

Η βασική δομή του SORI στηρίζεται σε τρία κομμάτια, την παρακολούθηση γειτόνων, την διάδοση της φήμης, και την τιμωρία.

1. *Παρακολούθηση γειτόνων*

Στο τμήμα αυτό συγκεντρώνονται στοιχεία για την συμπεριφορά που έχουν οι γειτονικοί κόμβοι στην προώθηση πακέτων. Χάρη στις παραδοχές που έχουν γίνει, ο κάθε κόμβος μπορεί να κατασκοπεύσει τους γειτονικούς του (παραδοχή 7) . Με αυτή την δυνατότητα , ένα κόμβος *N* διατηρεί μία λίστα γειτόνων (NNL_N) η οποία περιέχει όλους τους κόμβους εκείνους που ο κόμβος *N* μπορεί να ακούσει. Στην λίστα αυτή, ο κόμβος κρατάει δύο αριθμούς για τον κάθε κόμβο γείτονά του:

- Τον αριθμό $RF_N(X)$ (Request-for-Forwarding), ο οποίος είναι ο συνολικός αριθμός των πακέτων που ο κόμβος *N* έχει προωθήσει στον *X* για μετάδοση.
- Και τον αριθμό $HF_N(X)$ (Has-Forwarded), ο οποίος είναι ο συνολικός αριθμός των πακέτων τα οποία έχουν προωθηθεί από τον *X* και ο *N* τα έχει ακούσει.

Οι δύο παραπάνω αριθμοί ενημερώνονται με τους εξής κανόνες. Όταν ο κόμβος *N* στέλνει ένα πακέτο στον κόμβο *X* για μετάδοση, ο μετρητής του $RF_N(X)$ αυξάνεται κατά ένα. Στη συνέχει ο *N* ακούει το κανάλι και ελέγχει εάν ο *X* όντως προώθησε το πακέτο. Αν ο *N* ανιχνεύσει ότι ο *X* όντως έχει προωθήσει το πακέτο πριν λήξει ο χρόνος αναμονής, ο μετρητής του $HF_N(X)$ αυξάνεται κατά ένα.

Δεδομένων των αριθμών $RF_N(X)$ και $HF_N(X)$, ο κόμβος N μπορεί να δημιουργήσει ένα αρχείο που ονομάζεται καταγραφή των τοπικών εκτιμήσεων (local evaluation record) και συμβολίζεται με $LER_N(X)$, για τον κάθε γειτονικό κόμβο X. Το $LER_N(X)$, απαρτίζεται από δύο επιμέρους αριθμούς :

1. Τον αριθμό $G_N(X)$, ο οποίος υπολογίζεται ως $G_N(X) = \frac{RF_N(X)}{HF_N(X)}$
2. Και τον αριθμό $C_N(X)$ ο οποίος ονομάζεται *αυτοπεποίθηση* και χρησιμοποιείται για να περιγράψει πόσο σίγουρος είναι ο κόμβος *N*

ως προς την κρίση του, απέναντι στο κόμβο *X*. Σε αυτή την περίπτωση, θέτουμε $C_N(X) = RF_N(X)$.

2. *Διάδοση της Φήμης*

Με το σύστημα Παρακολούθησης γειτόνων, ένας κόμβος μπορεί να δημιουργήσει μία εικόνα για την συμπεριφορά των γειτόνων του, αυτό όμως δεν αρκεί για να τιμωρηθεί ο εγωιστικός κόμβος. Αυτό το πρόβλημα λύνει το σύστημα Διάδοσης της Φήμης. Το σύστημα αυτό χρησιμοποιείται για να κοινοποιούνται πληροφορίες αξιολόγησης μεταξύ των κόμβων, έτσι ώστε ένας εγωιστικός κόμβος να τιμωρείται από όλους τους γείτονές του και όχι αποκλειστικά από μερικούς που έχουν πέσει θύματα της συμπεριφοράς του. Το σύστημα αυτό λειτουργεί ως εξής:

1. Ο κάθε κόμβος *N* περιοδικά ενημερώνει τον πίνακα $LER_N(X)$, που διαθέτει για όλους τους *X* γείτονές του, βασιζόμενος στις αλλαγές των αριθμών $RF_N(X)$ και $HF_N(X)$, και στη συνέχεια διαδίδει την ενημερωμένη εγγραφή στην οποία έχει παρατηρηθεί σημαντική διαφορά στον αριθμό $G_N(X)$.
2. Ο κόμβος N χρησιμοποιεί την δικιά του $LER_N(X)$ και την $LER_i(X)$ που του στάλθηκε (όπου i είναι ο αριθμός της λίστας γειτόνων NNL_N) για να υπολογίσει την συνολική εκτίμηση της εγγραφής του X, σύμφωνα με τον παρακάτω τύπο:

$$OER_N(X) = \frac{\sum_{i \epsilon NNL_N \cup \{N\}, i \neq X} \lambda_N(i) \cdot C_i(X) \cdot G_i(X)}{\sum_{k \epsilon NNL_N \cup \{N\}, k \neq X} \lambda_N(k) \cdot C_k(X)}$$

όπου $\lambda_N(i)$ είναι η αξιοπιστία που έχει κερδίσει ο κόμβος i από την οπτική του κόμβου N. Στο SORI επιλέγεται $\lambda_N(i) = G_N(i)$, και ειδικότερα, $\lambda_N(N) = 1$ και $\lambda_N(i) = 0$ αν $RF_N(i) = 0$, το οποίο σημαίνει ότι ένας κόμβος δεν θα μπορούσε να έχει κάποια εγγραφή για τον N εάν δεν του έχει ζητηθεί ποτέ από τον N να προωθήσει κάποιο πακέτο.

3. *Τιμωρία*

Εφόσον υπάρχουν καταγραφές για εγωιστικούς κόμβους, σειρά έχει η τιμωρία τους, αυτό αναλαμβάνει και το κομμάτι αυτό του συστήματος. Σύμφωνα με την τιμή του $OER_N(X)$ ο κάθε κόμβος N μπορεί να τιμωρήσει τον γείτονά του X απορρίπτοντας τα πακέτα που του στέλνει για προώθηση.

Πιο συγκεκριμένα, αν ο αριθμός $OER_N(X)$ πέσει κάτω από ένα προκαθορισμένο κατώφλι, ο κόμβος N τιμωρεί τον κόμβο X με πιθανολογικό τρόπο. Η πιθανότητα να απορρίψει πακέτο είναι,

$$\mathrm{p} = \begin{cases} \mathrm{q} - \delta & \text{αν } \mathrm{q} > \delta \\ 0 & \text{διαφορετικά} \end{cases}$$

όπου $q = 1 - OER_N(X)$ και $0 < \delta < 1$ είναι τα περιθώρια ανεκτικότητας. Τα περιθώρια αυτά είναι ιδιαίτερα σημαντικά, αφού η απόρριψη ενός πακέτου μπορεί να προκληθεί και από άλλους παράγοντες, όπως οι συγκρούσεις, και όχι μόνο από την εγωιστική συμπεριφορά. Συγκεκριμένα, έχουν σχεδιαστεί, για να βοηθήσουν τους φιλικούς κόμβους να αποφεύγουν καταστάσεις αντεκδίκησης, φερόμενοι λίγο πιο γενναιόδωρα ο ένας στον άλλο.

Ενίσχυση της Ασφάλειας

Αυτή η ενότητα παρουσιάζει έναν τρόπο ενίσχυσης της Ασφάλειας έτσι ώστε να διορθώσει κάποια ευάλωτα σημεία του συστήματος. Χρησιμοποιείται ένας μηχανισμός βασισμένος στον μονόπλευρο κατακερματισμό της πληροφορίας (one-way-hash chain) όπως παρουσιάζεται παρακάτω:

Ο κόμβος *N* παίρνει την ταυτότητά του, που ονομάζεται ID_N , επιλέγοντας έναν τυχαίο αριθμό r_N και αναδρομικά εφαρμόζει μία ψευδό-τυχαία ακολουθία *h* στον αριθμό αυτό επί *k* φορές, αυτό δηλαδή είναι $ID_N = H_k(r_N)$ το οποίο αναδρομικά λαμβάνεται από ,

$$H_i(r_N) = \begin{cases} h\big(H_{i-1}(r_N)\big) & \alpha\nu\ i \in \{1,2,\dots,k\} \\ h(r_N) & \alpha\nu\ i = 0 \end{cases}$$

Όταν ο N εισέρχεται σε ένα ad-hoc δίκτυο, μεταδίδει την ταυτότητά του ID_N σε όλους τους γειτονικούς του κόμβους και αυτοί με τη σειρά τους αποθηκεύουν τον αριθμό αυτό στην λίστα γειτόνων τους NNL_S . Οι γείτονες θα χρησιμοποιήσουν αυτόν τον αριθμό - ταυτότητα για να πιστοποιήσουν μηνύματα που μεταδίδονται από αυτόν.

Στην συνέχει ο κόμβος N διαιρεί τον χρόνο σε ίσα διαστήματα και αναθέτει το i-οστό διάστημα μαζί με ένα κλειδί (που ονομάζεται K_i) το οποίο είναι $K_i = H_{k-i}(r_N)$ στην μονόδρομη συνάρτηση κατακερματισμού. Το μήνυμα που θα αποσταλεί σε αυτό το χρονικό διάστημα θα περιέχει ένα μήνυμα ταυτοποίησης (MAC , Message Authentication Code) το οποίο υπολογίζεται με το αντίστοιχο κλειδί K_i και το εκάστοτε μήνυμα M (δηλώνεται ως

MAC(K,M)). Για παράδειγμα, το περιεχόμενο του πακέτου P_i που στέλνεται στο i-στο διάστημα είναι

$$\{M_i||MAC(K_i', M_i)||K_{i-d}\}$$

όπου M_i είναι το μήνυμα προς αποστολή στο i-στο χρονικό διάστημα, το K_i' υπολογίζεται ως $K_i' = f(K_i)$ όπου f είναι η δεύτερη ψευδό-τυχαία συνάρτηση και $K_i = H_{k-i}(r_N)$, το d είναι η καθυστέρηση παράθεσης του κλειδιού (δηλαδή, στο $(i-d)$ διάστημα, το μήνυμα επικυρώνεται από το κλειδί K_{i-d}, και το κλειδί K_{i-d} θα παρατεθεί στο i-οστό διάστημα). Όταν ένας δέκτης δέχεται ένα πακέτο, ελέγχει αν το κλειδί MAC έχει ήδη παρατεθεί. Αν το κλειδί δεν έχει αποκαλυφθεί, κρύβει το μήνυμα με σκοπό να εξετάσει την αυθεντικότητά του την στιγμή που το K_i αποκαλυφθεί, διαφορετικά, απορρίπτει το πακέτο με την δικαιολογία ότι το κλειδί είχε παρατεθεί πριν την άφιξη του πακέτου και άρα το MAC μπορεί να είχε πλαστογραφηθεί. Επίσης, ένα πακέτο με μη αποδεκτό MAC θα απορριφθεί.

Όπως γίνεται εύκολα κατανοητό, για έναν εγωιστικό κόμβο είναι πολύ δύσκολο να πλαστογραφήσει κάποιο MAC χωρίς να έχει το αντίστοιχο κλειδί. Έτσι κάποια απόπειρα ενός κακόβουλου κόμβου να ενισχύσει τη φήμη του προσποιούμενος έναν «καλό» κόμβο, δεν θα αποδώσει. Η όλη ιδέα του συστήματος ενίσχυσης ασφαλείας, έχει σκοπό να αποφευχθεί η χρήση PKI(Public-Key Infrastructure) ή κάποιου άλλου συστήματος ασφαλείας, τα οποία συνήθως δεν είναι τόσο πρακτικά σε δίκτυα ad-hoc.

Αξιολόγηση του πρωτοκόλλου

Το SORI είναι ένα πρωτόκολλο που ενθαρρύνει την προώθηση πακέτων και τιμωρεί την εγωιστική συμπεριφορά σε ένα μη συνεργαζόμενο ad-hoc δίκτυο. Τα μοναδικά χαρακτηριστικά του είναι πρώτον, η φήμη του κόμβου που μετριέται με αντικειμενικά κριτήρια μέσω της παρακολούθησης γειτόνων, δεύτερων, η προώθηση της φήμης είναι ασφαλής χάρης τον μονομερή κατακερματισμό ο οποίος είναι υπολογιστικά υλοποιήσιμος και τρίτον, η φήμη ενός κόμβου διαδίδεται μόνο στους γείτονές του, το οποίο μειώνει αισθητά το overhead της επικοινωνίας σε σύγκριση με σχήματα που διαδίδουν την φήμη σε όλο το δίκτυο. Τα αποτελέσματα προσομοίωσης του SORI δείχνουν ότι το συγκεκριμένο σχήμα μπορεί να αναγνωρίσει εγωιστικούς κόμβους και να τους τιμωρήσει αναλόγως με μεγάλη επιτυχία.

Το SORI έχει όμως ένα σημαντικό μειονεκτήματα, δεν μπορεί να αναγνωρίσει και να τιμωρήσει κακόβουλους κόμβους όπως κάνει με τους εγωιστικούς, οι κακόβουλοι κόμβοι μπορούν να προκαλέσουν μεγαλύτερο πρόβλημα στο δίκτυο και είναι πιο δύσκολο να εντοπισθούν. [7]

2.3.2.2. CONFIDANT Protocol

(Cooperation Of Nodes : Fairness In Dynamic Ad - hoc NeTworks)

Το CONFIDANT είναι ένα πρωτόκολλο το οποίο ανήκει στην κατηγορία εκείνων που χρησιμοποιούν μηχανισμούς κινήτρων βασισμένων στη φήμη. Είναι ένα πρωτόκολλο που προσπαθεί να χτίσει σχέσεις εμπιστοσύνης ανάμεσα στους κόμβους ενός κινητού Ad -hoc δικτύου. Οι επιλογές δρομολόγησης στηρίζονται στην εμπειρία, στην παρακολούθηση και στην αναφορά της συμπεριφοράς των υπόλοιπων κόμβων κατά τη διάρκεια της δρομολόγησης. Οι δύο βασικές ιδέες στις οποίες στηρίζεται το CONFIDANT είναι οι εξής:

1. Να μαθαίνεις από τους γείτονές σου τις κακές εμπειρίες που τυχόν είχαν με κακόβουλους κόμβους, έτσι ώστε να προτρέψεις μία κακή εμπειρία ο ίδιος.
2. Όταν έχεις μία κακή εμπειρία ο ίδιος τότε μοιράσου την με τους γείτονές σου.

Το CONFIDANT αποτελείται από τέσσερα βασικά υποσυστήματα:

1. Το σύστημα Παρακολούθησης ή Monitor.
2. Το σύστημα Διαχείρισης Αξιοπιστίας ή Trust Manager
3. Το σύστημα Φήμης ή Reputation System.
4. Το σύστημα Διαχείρισης Διαδρομών ή Path Manager.

1. Το σύστημα Παρακολούθησης ή Monitor

Το σύστημα αυτό υπάρχει μέσα σε κάθε κόμβο. Σκοπός του είναι να ανιχνεύει παρεκτροπές από τον επόμενο κόμβο πάνω στην προκαθορισμένη διαδρομή. Αυτό είναι εφικτό είτε με το να παρακολουθεί την μετάδοση στον επόμενο κόμβο (παθητική γνώση) είτε παρακολουθώντας την συμπεριφορά του πρωτοκόλλου διαδρομής. Επίσης, ο κόμβος έχει την δυνατότητα να κρατήσει ένα αντίγραφο του δρομολογούντος πακέτου καθώς ακούει την μετάδοση , κάτι που μπορεί να χρησιμοποιήσει για να διαπιστώσει αν υπήρξε κάποια αλλαγή του περιεχομένου του. Όταν κάποιος κόμβος ανιχνεύσει κάποια κακή συμπεριφορά καλεί το σύστημα Φήμης.

2. Το σύστημα Διαχείρισης Αξιοπιστίας ή Trust Manager

Το σύστημα αυτό για την λειτουργία του χρησιμοποιεί μηνύματα ALARM τα οποία στέλνονται μεταξύ των κόμβων με σκοπό την ειδοποίηση για κακόβουλους κόμβους. Τα εξερχόμενα μηνύματα ALARM προωθούνται είτε από τον κόμβο-θύμα κακόβουλης συμπεριφοράς είτε από τον κόμβο που ανίχνευσε κακόβουλη συμπεριφορά είτε από τον κόμβο που έλαβε μήνυμα ALARM. Οι παραλήπτες των μηνυμάτων αυτών είναι οι επονομαζόμενοι "κόμβοι φίλοι" οι οποίοι είναι καταχωρημένοι στην λίστα φίλων κάθε κόμβου. Ωστόσο, κάθε μήνυμα ALARM που λαμβάνεται πρέπει να ελέγχεται για την αξιοπιστία του βάση την "φήμη" του κόμβου αποστολέα.

Ένας μηχανισμός παρόμοιος με το σύστημα Διαχείρισης Αξιοπιστίας είναι το Pretty Good Privacy (ή PGP). Ο μηχανισμός αυτός κατονομάζει πολλά επίπεδα αξιοπιστίας, όπως για παράδειγμα : άγνωστος (unknown), χωρίς σχόλια (none), οριακά αποδεκτός (marginal) και πλήρως αποδεκτός (complete). Θέλοντας να κατηγοριοποιήσει ένα μήνυμα, υπολογίζει τον βαθμό εγκυρότητάς του ανάλογα με την εγκυρότητα του δημοσίου κλειδιού και των ψηφιακών υπογραφών που περιέχει. Το σύστημα Διαχείρισης Αξιοπιστίας χρησιμοποιεί την ίδια αρχή αλλά για να καθορίσει αν υπάρχουν επαρκής και αξιόπιστες αποδείξεις για κάποια κακόβουλη συμπεριφορά.

Το σύστημα Διαχείρισης Αξιοπιστίας αποτελείται από:

- Έναν πίνακα ειδοποιήσεων (alarm table), ο οποίος περιέχει πληροφορίες για τα εισερχόμενα μηνύματα ALARM.
- Έναν πίνακα αξιοπιστίας (trust table), ο οποίος περιέχει πληροφορίες για την εμπιστευτικότητα των κόμβων.
- Μία λίστα φιλικών κόμβων, στους οποίους θα στείλει μηνύματα ALARM αν χρειαστεί.

3. Το σύστημα Φήμης ή Reputation System

Το σύστημα Φήμης είναι εκείνο που αποφασίζει εάν ένας κόμβος συμπεριφέρεται εγωιστικά ή όχι. Για να το κάνει αυτό, χρησιμοποιεί έναν πίνακα που περιέχει στοιχεία κόμβων και τις βαθμολογίες τους. Η βαθμολογία κάθε κόμβου αλλάζει μόνο όταν υπάρχουν επαρκής αποδείξεις για συνεχή ανάρμοστη συμπεριφορά προς άλλους κόμβους. Πιο συγκεκριμένα, η βαθμολογία τροποποιείται σύμφωνα με μία ποσοστιαία

συνάρτηση η οποία υπολογίζει την βαρύτητα των κακόβουλων περιστατικών η οποία μειώνεται ανάλογα με:

- αν το περιστατικό αποτελεί προσωπική εμπειρία του κόμβου
- αν το περιστατικό καταγράφηκε από παρακολούθηση γειτόνων
- αν το περιστατικό αναφέρθηκε από άλλο κόμβο

Συμπερασματικά, το σύστημα Φήμης στηρίζεται περισσότερο στις προσωπικές εμπειρίες του κάθε κόμβου παρά στις αναφορές από εμπειρίες γειτονικών κόμβων.

Όταν η βαθμολογία ενός κόμβου έχει καθοριστεί, η καταχώριση για τον συγκεκριμένο κόμβο αλλάζει αναλόγως. Εάν η βαθμολογία του κόμβου έχει επιδεινωθεί σημαντικά , δηλαδή περισσότερο από ένα σημείο ανοχής, καλείται το σύστημα Διαχείρισης Διαδρομών ή Path Manager. Τελικά το σύστημα Φήμης είναι βασισμένο σε αρνητικές εμπειρίες, παρά σε καλές εντυπώσεις.

4. Το σύστημα Διαχείρισης Διαδρομών ή Path Manager.

Το σύστημα Διαχείρισης Διαδρομών ακολουθεί τις παρακάτω λειτουργίες:

- Ανάλογα με τα μέτρα ασφαλείας σχεδιάζει τις ασφαλέστερες διαδρομές
- Διαγράφει τα μονοπάτια που περιέχουν κακόβουλους κόμβους
- Δρα κατά των κακόβουλων κόμβων σε περίπτωση που λάβει κάποιο αίτημα από αυτούς
- Αντιδρά σε αιτήματα δρομολόγησης που στο μονοπάτι τους περιλαμβάνουν κάποιον κακόβουλο κόμβο.

Περιγραφή του πρωτοκόλλου

Με τη περιγραφή των παραπάνω συστημάτων, γίνεται εύκολα κατανοητό ότι ο κάθε κόμβος ελέγχει την συμπεριφορά των γειτόνων του. Εάν κάποιο ύποπτο γεγονός συμβεί αναφέρεται αμέσως στο σύστημα Φήμης του εκάστοτε κόμβου που αυτό με τη σειρά του ελέγχει εάν έχει επαναληφθεί ή όχι στο παρελθόν. Με αυτόν τον τρόπο αποφεύγεται η λανθασμένη εκτίμηση για κάποιο κόμβο που ,για παράδειγμα, μπορεί να έχασε κάποια πακέτα λόγω συγκρούσεων. Όταν η βαθμολογία κάποιου κόμβου στον πίνακα φήμης

πέσει κάτω από ένα ανεκτό όριο, τότε στέλνεται πληροφορία στο σύστημα Διαχείρισης Διαδρομών. Τότε ο Path Manager του κόμβου διαγράφει όλες εκείνες τις διαδρομές που περιέχουν τον αναφερθέν κόμβο από τη μνήμη του. Μετά από την διαδικασία αυτή, ο κόμβος συνεχίζει να παρακολουθεί την γειτονιά του και παράλληλα στέλνει μηνύματα κινδύνου (ALARM messages) στους κόμβους-φίλους του.

Τα μηνύματα ALARM στέλνονται μέσω του συστήματος Αξιοπιστίας και περιέχουν προειδοποιητικές αναφορές. Αναλυτικότερα ,ένα μήνυμα Κινδύνου, περιέχει:

1. τον τύπο παράβασης
2. τον αριθμό περιστατικών που παρατηρήθηκαν
3. εάν το μήνυμα το στέλνει ο ίδιος ο κόμβος που παρατήρησε την κακή συμπεριφορά
4. την διεύθυνση του κακόβουλου κόμβου που αναφέρεται
5. την διεύθυνση του κόμβου που παρατήρησε την κακόβουλη συμπεριφορά
6. και την διεύθυνση του παραλήπτη του μηνύματος ALARM.

Όταν το σύστημα Παρακολούθησης λάβει ένα μήνυμα Κινδύνου, το παραδίδει στο σύστημα Αξιοπιστίας, όπου στη συνέχεια γίνεται αξιολόγηση του αποστολέα του μηνύματος αυτού. Εάν ο αποστολέας είναι έμπιστος, τότε ο πίνακας που περιέχει τα μηνύματα ALARM ενημερώνεται. Αντιθέτως, εάν υπάρχουν αρκετές αποδείξεις ότι ο αποστολέας δεν είναι έμπιστος, τότε στέλνεται η ανάλογη πληροφορία στο σύστημα Φήμης, το οποίο με τη σειρά του εξετάζει την σημαντικότητα του γεγονότος με βάση τον αριθμό των ανάλογων συμβάντων που έχουν αναφερθεί και εκτιμάται η φήμη του κόμβου όπως περιγράφηκε παραπάνω (παράγραφος 3).

Αξιολόγηση του πρωτοκόλλου

Σύμφωνα με προσομοιώσεις που έχουν γίνει, διαπιστώνουμε αρχικά ότι το CONFIDANT λειτουργεί ευεργετικά στο δίκτυο σε σύγκριση με άλλα πρωτόκολλα που δεν χρησιμοποιούν αμυντικούς μηχανισμούς. Πιο συγκεκριμένα, τα αποτελέσματα δείχνουν ότι πρωτόκολλα χωρίς μηχανισμούς άμυνας χάνουν δύο φορές περισσότερα πακέτα λόγω κακόβουλων κόμβων σε σχέση με το πρωτόκολλο CONFIDANT. Επιπρόσθετα, όταν αυξηθεί ο αριθμός των κόμβων στο δίκτυο, οι μηχανισμοί άμυνας του

CONFIDANT δείχνουν να κρατούν σταθερό το επίπεδο των πακέτων που εσκεμμένα χάνονται, σε αντίθεση με άλλα, ευάλωτα στην κακοήθη συμπεριφορά , πρωτόκολλα που τείνουν να χάνουν πολύ περισσότερα πακέτα. Εμβαθύνοντας, ακόμα και όταν αυξηθεί ο αριθμός των κακόβουλων κόμβων σε σχέση με τους υπόλοιπους στο δίκτυο, το CONFIDANT αν και είναι ευαίσθητο όσο το ποσοστό αυτό αυξάνει, κρατάει τον αριθμό των πακέτων που χάνονται χαμηλό ακόμα και αν το ποσοστό των κακόβουλων κόμβων ξεπεράσει το 50% των συνολικών κόμβων.

Πέρα όμως από τα παραπάνω πλεονεκτήματα του CONFIDANT, παρατηρούνται και κάποια μειονεκτήματα. Όταν αυξάνεται η κινητικότητα μεταξύ των κόμβων, το πρωτόκολλο δεν δείχνει να συμπεριφέρεται καλύτερα έναντι σε άλλα μη αμυντικά πρωτόκολλα. Συγκεκριμένα, η χρήση αμυντικού μηχανισμού σε μία τέτοια περίπτωση δίνει αποτελέσματα περίπου ίδια με εκείνα που θα έδινε ένα ανυπεράσπιστο δίκτυο. Αυτό είναι λογικό, αφού ο μηχανισμός φήμης στηρίζεται στις εμπειρίες μεταξύ των κόμβων κάτι που προϋποθέτει την γνωριμία μεταξύ τους. Σε ένα συνεχώς μεταβαλλόμενο δίκτυο ο κάθε κόμβος ανά πάσα στιγμή είναι πιθανότερο να βρίσκεται μεταξύ άγνωστων κόμβων, κάτι που τον καθιστά εξίσου ευάλωτο με το αν είχε χρησιμοποιηθεί ένα μη αμυντικό πρωτόκολλο. Ένα ακόμα μειονέκτημα του CONFIDANT είναι τα μηνύματα ALARM. Στέλνοντας μηνύματα ALARM το πρωτόκολλο σπαταλά πόρους του συστήματος, ενώ ταυτόχρονα, γίνεται ευάλωτο σε επιθέσεις.

Συμπερασματικά, το CONFIDANT είναι ένα πρωτόκολλο που προσπαθεί να επιβάλλει την συνεργασία και την δικαιοσύνη στο δίκτυο μέσα από έναν μηχανισμό τιμωρίας των κακόβουλων κόμβων. Αξιοσημείωτες επιθέσεις στην δρομολόγηση μπορούν να εκμηδενιστούν με την χρήση του μηχανισμού ανίχνευσης, ειδοποίησης και δράσης που χρησιμοποιεί το CONFIDANT. [6]

Κεφάλαιο 3

Εισαγωγή στην Υλοποίηση

Αφού μιλήσαμε γενικά για τα ασύρματα ad-hoc δίκτυα, τα σημαντικότερα Πρωτόκολλα Δρομολόγησης και την ανάγκη για Μηχανισμούς Προώθησης Συνεργασίας, συνεχίζουμε περνώντας στο κομμάτι της Υλοποίησης.

Κατά τη διάρκεια της Διπλωματικής αυτής, επιλέξαμε να υλοποιήσουμε στον προσομοιωτή Δικτύων NS-2(Network Simulator 2) ένα από τα πρωτόκολλα Προώθησης Συνεργασίας που αναφέρθηκαν παραπάνω. Ο μηχανισμός CONFIDANT ήταν η πρώτη επιλογή και αυτό για δύο σημαντικούς λόγους:

- Ο μηχανισμός CONFIDANT βασίζεται στο πρωτόκολλο δρομολόγησης DSR (Dynamic Source Routing). Όπως αναφέρθηκε και στο Κεφάλαιο 1, το DSR σε απόπειρες σύγκρισής του με άλλα πρωτόκολλα δρομολόγησης (DSDV, AODV, TORA) , φαίνεται να ενεργεί πιο αποτελεσματικά.[2]
- Ο μηχανισμός CONFIDANT φαίνεται να ενεργεί αρκετά αποδοτικά σε σύγκριση με τα υπόλοιπα πρωτόκολλα. Επίσης η υλοποίηση της Sonja Buchegger, που παρουσίασε τον μηχανισμό CONFIDANT, είναι αρκετά κοντά στις δυνατότητες του ns-2 που χρησιμοποιείται στην παρούσα διπλωματική.

3.1. Αναλυτική Περιγραφή του DSR πρωτοκόλλου

Για να προχωρήσουμε στην υλοποίηση, αρχικά πρέπει να περιγράψουμε αναλυτικά το DSR πρωτόκολλο δρομολόγησης καθώς το CONFIDANT λειτουργεί ως πρόσθετο στοιχείο πάνω σε αυτό.

Το πρωτόκολλο DSR ή αλλιώς Πρωτόκολλο ανεξάρτητης δυναμικής δρομολόγησης (Dynamic Source Routing Protocol) είναι ένα εύκολο και εφαρμόσιμο πρωτόκολλο δρομολόγησης που έχει σχεδιαστεί αποκλειστικά για χρήση σε ασύρματα Ad-Hoc δίκτυα με κινητούς κόμβους. Το DSR επιτρέπει στο δίκτυο να είναι εντελώς αυτόνομο και αυτοοργανούμενο , χωρίς την ανάγκη χρήσης κάποιας δικτυακής υποδομής ή διαχείρισης. Αποτελείται από δύο μηχανισμούς την Εύρεση Διαδρομών (Route Discovery) και την Διατήρηση Διαδρομών (Route Maintenance). Οι δύο αυτοί μηχανισμοί, συνεργάζονται έτσι ώστε να επιτρέπουν στους κόμβους να ανακαλύψουν και να διατηρήσουν διαδρομές προς διάφορους προορισμούς

σε όλο το Ad-hoc δίκτυο. Η χρήση της ανεξάρτητης δρομολόγησης, δεν επιτρέπει στην μετάδοση των πακέτων να είναι επιπόλαιη και να χρειάζεται συνεχής ενημέρωση διαδρομών, ενώ, αντίθετα επιτρέπει στους κόμβους που συμμετέχουν ή ακούν τυχαία μία μετάδοση, να κρατούν πληροφορίες διαδρομών οι οποίες μπορεί να τους χρειαστούν μελλοντικά. Όλες οι πτυχές του πρωτοκόλλου ενεργούν κατ΄ απαίτηση, επιτρέποντας στο overhead του προωθούμενου πακέτου να κλιμακώνεται αυτόματα με σκοπό να περιέχει μόνο τις απαραίτητες πληροφορίες για τις διαδρομές που χρησιμοποιεί εκείνη τη στιγμή στο δίκτυο.

Οι κόμβοι του δικτύου συνεργάζονται για την προώθηση των πακέτων έτσι ώστε να επιτρέπεται η επικοινωνία μεταξύ απομακρυσμένων εκτός εμβέλειας κόμβων δρομολογώντας τα πακέτα από τον ένα κόμβο στον άλλο με σκοπό αυτά να φτάσουν στον προορισμό τους. Καθώς το δίκτυο μπορεί να αλλάξει οποιαδήποτε χρονική στιγμή, όπως να απενεργοποιηθούν κόμβοι ή να αλλάξουν θέση μέσα στο δίκτυο, όλη η δρομολόγηση αυτομάτως ενημερώνεται από το DSR πρωτόκολλο.

Κάθε πακέτο δεδομένων που στέλνεται είναι εφοπλισμένο με την ολοκληρωμένη λίστα των κόμβων της διαδρομής που πρέπει να ακολουθήσει μέσα στην κεφαλίδα του. Έτσι, επιτρέπεται στο πακέτο να είναι αυτόνομο χωρίς να χρειάζεται ενημέρωση για την δρομολόγησή του καθώς περνάει από τους κόμβους που το προωθούν. Εφοδιάζοντας το πακέτο με την πλήρη διαδρομή που επρόκειτο να ακολουθήσει, ο κάθε κόμβος που μεσολαβεί ή ο κάθε κόμβος που ακούει κρυφά το πακέτο, μπορεί να κρατήσει το μονοπάτι που χρησιμοποιείται στην μνήμη του, έτσι ώστε να το χρησιμοποιήσει αν το χρειαστεί ο ίδιος στο μέλλον.

3.1.1. Παραδοχές

1. Το DSR πρωτόκολλο υποθέτει πως όλοι οι κόμβοι στο δίκτυο είναι διατεθειμένοι να επικοινωνούν μεταξύ τους και πρόθυμοι να τηρούν τους κανόνες του πρωτοκόλλου του δικτύου. Αυτό σημαίνει, πως όλοι οι κόμβοι συμμετέχουν πρόθυμα στην προώθηση των πακέτων.
2. Το DSR αναφέρεται στον ελάχιστο αριθμό προωθητικών κόμβων που χρειάζονται σε μία διαδρομή, ακόμα και στην περίπτωση επικοινωνίας δύο ακριανών κόμβων.

3. Πακέτα μπορεί να χαθούν ή να αλλοιωθούν κατά την μετάδοση στο ασύρματο δίκτυο. Ωστόσο, ένας κόμβος που λάβει ένα τέτοιο πακέτο μπορεί να το αναγνωρίσει και να το απορρίψει.
4. Οι κόμβοι που βρίσκονται μέσα στο Ad-hoc δίκτυο μπορούν να αλλάξουν θέση σε οποιαδήποτε χρονική στιγμή και χωρίς ειδοποίηση, ή μπορούν ακόμα και να κινούνται συνεχόμενα. Ωστόσο, οι κατασκευαστές του DSR υποθέτουν πως η ταχύτητα που κινούνται οι σταθμοί πρέπει να είναι μέτρια έτσι ώστε να μπορεί να επιτευχθεί η μετάδοση του πακέτου.
5. Οι κόμβοι μπορούν επίσης να ενεργοποιήσουν την promiscuous (άνευ διακρίσεως) λειτουργία στις διεπαφές του ασύρματου υλικού τους, επιτρέποντας έτσι στις συσκευές τους να δέχονται όλα τα πακέτα του δικτύου χωρίς να φιλτράρουν αν προορίζονται για αυτούς ή όχι.
6. Η ασύρματη δρομολόγηση μεταξύ δύο κόμβων μπορεί να μην είναι πάντα αμφίδρομη. Το DSR πρωτόκολλο μπορεί να στείλει επιτυχώς ένα πακέτο μεταξύ δύο κόμβων των οποίων η επικοινωνία είναι μονόπλευρη.
7. Τέλος, το DSR υποθέτει πως ο κάθε κόμβος έχει ένα και μοναδικό αναγνωριστικό, την IP του.

3.1.2. Περιγραφή του Πρωτοκόλλου

Όπως αναφέραμε και παραπάνω, το DSR πρωτόκολλο, για την λειτουργία του, χρησιμοποιεί δύο μηχανισμούς που συνεργάζονται μεταξύ τους και επιτρέπουν την εύρεση και συντήρηση ανεξάρτητων διαδρομών σε ένα Ad-hoc δίκτυο:

- ***Εύρεση Διαδρομών (Route Discovery)*** :
  Είναι ο μηχανισμός εκείνος όπου όταν ένας κόμβος S θέλει να στείλει ένα πακέτο στον κόμβο D, βρίσκει το μονοπάτι εκείνο που είναι κατάλληλο για την μετάδοση. Η Εύρεση Διαδρομών χρησιμοποιείται μόνο όταν ο S προσπαθεί να στείλει ένα πακέτο στον D και δεν έχει αποθηκευμένη μία διαδρομή στην μνήμη του.
- ***Διατήρηση Διαδρομών (Route Maintenance)***:
  Είναι ο μηχανισμός με τον οποίο ένας κόμβος S είναι ικανός να εντοπίσει, ενώ χρησιμοποιεί ένα μονοπάτι για τον κόμβο προορισμό D, εάν η τοπολογία του δικτύου έχει υποστεί αλλαγές τέτοιες που να αποτρέπουν την επιτυχή προώθηση του πακέτου. Τέτοιες αλλαγές

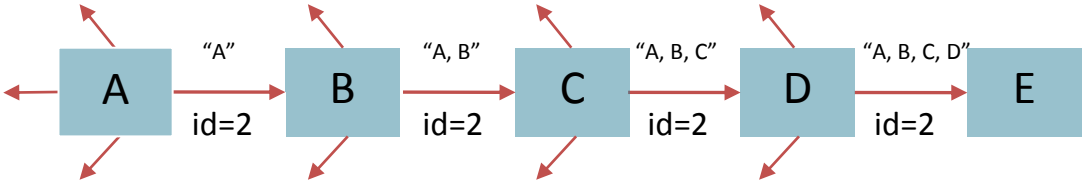
μπορεί να είναι η μετακίνηση κάποιου από τους προωθητικούς κόμβους στο μονοπάτι ή ακόμα και η απενεργοποίησή του. Ο S μπορεί τότε να ψάξει μεταξύ των μονοπατιών που διατηρεί στην μνήμη του εάν έχει άλλο μονοπάτι για τον προορισμού D, ή να καλέσει τον μηχανισμό Εύρεσης Διαδρομών για να ανακαλύψει ένα καινούργιο.

Ο μηχανισμός Εύρεσης Διαδρομών και Συντήρησης Διαδρομών ενεργούν μόνο κατ' απαίτηση. Σε διαφορά με άλλα πρωτόκολλα, το DSR δεν χρειάζεται περιοδικά πακέτα κάθε τύπου ή σε οποιουδήποτε επιπέδου μέσα στο δίκτυο. Για παράδειγμα, το DSR δεν χρησιμοποιεί καμία περιοδική ανακοίνωση δρομολόγησης, αναφορά κατάστασης δικτύου ή πακέτα εύρεσης γειτόνων, ενώ ταυτόχρονα δεν στηρίζεται σε τέτοιου είδος ενέργειες από άλλα πρωτόκολλα που λειτουργούν στο δίκτυο. Αυτή η ολοκληρωτικά αυτόνομη συμπεριφορά επιτρέπει στο overhead των πακέτων που δημιουργούνται από το DSR να φτάνει μέχρι και το μηδέν, όταν βέβαια όλοι οι κόμβοι βρίσκονται σε κατάσταση ηρεμίας με σεβασμό ο ένας στον άλλο και όλες οι υπάρχουσες διαδρομές έχουν ανακαλυφθεί. Όταν οι κόμβοι αρχίζουν να κινούνται περισσότερο και όσο η τοπολογία του δικτύου αλλάζει, το overhead του DSR πακέτου αυτόματα ενημερώνεται και αυξάνεται μόνο όσο χρειάζεται για να εντοπίσει τις διαδρομές που μπορεί να χρησιμοποιήσει εκείνη τη στιγμή.
Ταυτόχρονα, καθώς ο κάθε κόμβος έχει την δυνατότητα να μάθει και να αποθηκεύσει περισσότερες από μία διαδρομές για κάποιον προορισμό, σε περίπτωση ξαφνικής αλλαγής του δικτύου, η ενημέρωση της διαδρομής του πακέτου γίνεται πιο γρήγορα. Αυτή η δυνατότητα, δίνει στο δίκτυο το προνόμιο να κερδίσει overhead, αφού δεν είναι απαραίτητο να χρησιμοποιήσει τον μηχανισμό Εύρεσης Διαδρομών κάθε φορά που ένας κόμβος στο μονοπάτι αλλάζει θέση.
Επίσης οι διαδικασίες Εύρεσης και Διατήρησης Διαδρομών στο DSR έχουν σχεδιαστεί με τέτοιο τρόπο, ώστε να επιτρέπουν και στις μονόπλευρες συνδέσεις να επικοινωνούν μεταξύ τους.
Επίσης, το DSR υποστηρίζει την επικοινωνία μεταξύ διαφορετικού τύπου ασύρματων δικτύων, επιτρέποντας έτσι στον αποστολέα ενός πακέτου να επιλέξει μία διαδρομή που συνδυάζει διάφορους τύπους δικτύων ανάλογα με την διαθεσιμότητά τους. Για παράδειγμα, μερικοί κόμβοι μέσα στο ad-hoc δίκτυο μπορεί να έχουν μόνο πολύ μικρή εμβέλεια, ενώ άλλοι κόμβοι να μπορούν να αυξήσουν ή να μειώσουν την εμβέλεια τους. Ένας τέτοιος συνδυασμός θεωρείται από το DSR σαν ένα ad-hoc δίκτυο.

3.1.2.1. Αναλυτική περιγραφή του Μηχανισμού Εύρεσης Διαδρομών

Όταν ένας κόμβος Α θέλει να στείλει ένα πακέτο με προορισμό έναν κόμβο Ε, επισυνάπτει στην κεφαλίδα του πακέτου ένα μονοπάτι που περιέχει την σειρά των προωθητικών κόμβων που αυτό πρέπει να περάσει για να σταλεί επιτυχώς στον προορισμό του. Συνήθως, ο Α θα ψάξει και θα βρει την καταλληλότερη διαδρομή που έχει αποθηκευμένη για τον Ε μέσα στην Μνήμη Διαδρομών (Route Cache) του, εάν δεν υπάρχουν καθόλου διαδρομές, τότε ο Α θα απευθυνθεί στον μηχανισμό Εύρεσης Διαδρομών για την ανακάλυψη του μονοπατιού που χρειάζεται. Σε αυτήν την περίπτωση, ο κόμβος Α ονομάζεται *εισηγητής* (initiator) και ο κόμβος Ε *στόχος* (target) της διαδικασίας Εύρεσης Διαδρομής.

Για παράδειγμα, έστω ο κόμβος Α του σχήματος θέλει να εκτελέσει μία διαδικασία Εύρεσης Διαδρομής από τον Α στον Ε. Για να εκκινήσει αυτή την διαδικασία, ο Α στέλνει ένα πακέτο ROUTE REQUEST (αίτημα δρομολόγησης) σε όλους τους κόμβους που βρίσκονται εντός εμβέλειας με αυτόν. Το κάθε ROUTE REQUEST προσδιορίζει ποιος είναι ο εισηγητής και ποιος ο στόχος αυτής της ενέργειας. Πέρα από την παραπάνω πληροφορία, το πακέτο περιέχει και ένα μοναδικό id αίτησης, το οποίο προσδιορίζεται από τον εισηγητή αυτής. Το κάθε ROUTE REQUEST επίσης περιέχει μία λίστα από τις διευθύνσεις των κόμβων εκείνων που το έχουν λάβει και το έχουν προωθήσει. Αυτές οι εγγραφές καταγράφονται σε μία άδεια λίστα που έχει δημιουργήσει ο εισηγητής της διαδικασίας Εύρεσης Διαδρομών, στην περίπτωσή μας ο Α.



Παράδειγμα Μηχανισμού Εύρεσης Διαδρομής: Ο εισηγητής είναι ο Α και ο στόχος ο Ε

Όταν ένας κόμβος λάβει ένα ROUTE REQUEST, ελέγχει εάν αυτός είναι ο στόχος του μηνύματος. Εάν αυτό ισχύει, στέλνει στον εισηγητή του ένα μήνυμα ROUTE REPLY (Απάντηση σε αίτηση δρομολόγησης). Όταν με τη

σειρά του ο εισηγητής παραλάβει ένα ROUTE REPLY, αποθηκεύει την διαδρομή που βρίσκεται στην λίστα του μηνύματος στην Μνήμη Διαδρομών του (Route Cache). Διαφορετικά, όταν ένας κόμβος λάβει ένα μήνυμα ROUTE REQUEST και εντοπίσει ότι ο εισηγητής του μηνύματος έχει ξαναστείλει αυτό το ROUTE REQUEST με το ίδιο αναγνωριστικό (id) και ο κόμβος παραλήπτης υπάρχει ήδη στην λίστα του μονοπατιού, τότε αυτός απορρίπτει την Αίτηση. Εάν κανένα από τα παραπάνω δεν ισχύει, τότε ο κόμβος που έλαβε το πακέτο προσθέτει στη λίστα δρομολόγησης το αναγνωριστικό του και το στέλνει προς όλους τους κόμβους που βρίσκονται εντός της εμβέλειάς του.

Κατά την διάρκεια επιστροφής ενός ROUTE REPLY μηνύματος στον εισηγητή της διαδικασίας, όπως στο παράδειγμά μας ο κόμβος Ε στέλνει στον κόμβο Α, ο Ε θα εξετάσει αρχικά εάν στη δική του Μνήμη Διαδρομών υπάρχει κάποιο μονοπάτι που να πηγαίνει πίσω στον Α, εάν βρει, θα το χρησιμοποιήσει για να στείλει το ROUTE REPLY πίσω στον κόμβο εισηγητή. Εάν δεν βρει κάποια διαδρομή που να ταιριάζει με αυτήν που θέλει να πραγματοποιήσει, ο κόμβος Ε θα ξεκινήσει μία καινούργια διαδικασία Εύρεσης Διαδρομών με στόχο τον κόμβο Α. Στην περίπτωση αυτή, για να αντιμετωπισθεί τυχόν επανάληψη διαδικασιών Εύρεσης Διαδρομών, ο κόμβος Ε πρέπει να επισυνάψει μαζί με το δικό του ROUTE REQUEST το ROUTE REPLY για τον κόμβο Α. Εναλλακτικά, ο κόμβος Ε θα μπορούσε να χρησιμοποιήσει την ήδη υπάρχουσα διαδρομή που βρίσκεται στο ROUTE REQUEST, όμως αυτό επιβάλλει την αμφίδρομη σύνδεση όλων των κόμβων μεταξύ τους.

Όταν ένας κόμβος πραγματοποιεί μία διαδικασία Εύρεσης Διαδρομής, αποθηκεύει ένα αντίγραφο του αρχικού πακέτου σε μία τοπική μνήμη αυτή η μνήμη ονομάζεται *Μνήμη Αποστολής* (ή *Send Buffer*). Η Μνήμη Αποστολής περιέχει αντίγραφα κάθε πακέτου που ο κόμβος δεν μπορεί να προωθήσει επειδή δεν έχει βρει ακόμα μία διαδρομή για τους προορισμούς τους. Το κάθε πακέτο μέσα στην Μνήμη, επισφραγίζεται με την ώρα που αυτό τοποθετήθηκε σε αυτήν και απορρίπτεται όταν ο χρόνος αναμονής του λήξει. Είναι χρήσιμο επίσης για την μνήμη, να αποφεύγει την υπερφόρτωση, με την χρήση μίας FIFO (First In First Out) ουράς ή κάποιας άλλης στρατηγικής που θα διαγράφει ένα πακέτο ακόμα και πριν έρθει η λήξη του.

Όσο ένα πακέτο παραμένει στον Send Buffer, ο κόμβος πρέπει περιστασιακά να ξεκινάει διαδικασίες Εύρεσης Διαδρομών με στόχο τον προορισμό αυτού. Ωστόσο, ο κόμβος πρέπει να μετριάσει όσο γίνεται περισσότερο τον ρυθμό

που στέλνει ROUTE REQUEST για την ίδια διεύθυνση, αφού ο κόμβος προορισμός μπορεί να μην είναι προσωρινά διαθέσιμος ή προσπελάσιμος.

Παράδειγμα μη επιτυχούς απόπειρας Εύρεσης Διαδρομών : Ο κόμβος C δεν μπορεί να προωθήσει στον D και ως συνέπεια το πακέτο δεν μπορεί να φτάσει στον στόχο E

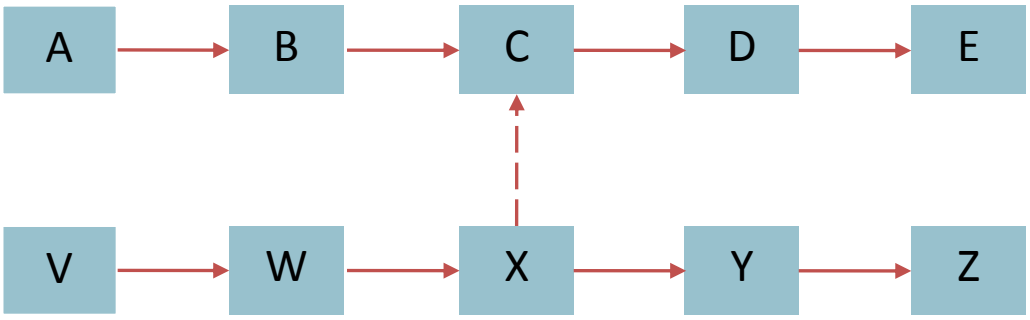
Εάν γίνει μία καινούργια προσπάθεια για Εύρεση Διαδρομών σε μία περίπτωση όπως αυτή του σχήματος, τότε πολλά μη παραγωγικά ROUTE REQUEST πακέτα θα επιβαρύνουν το δίκτυο. Με σκοπό να μειωθεί αυτή η επιβάρυνση, χρησιμοποιείται ένα άνω όριο μέγιστου ρυθμού αποστολής Αιτήσεων για έναν συγκεκριμένο στόχο. Εάν ένα κόμβος προσπαθήσει να στείλει επιπρόσθετα πακέτα δεδομένων σε έναν συγκεκριμένο κόμβο συχνότερα από το όριο αποστολής, τότε τα πακέτα αυτά πρέπει να μπαίνουν στην ουρά του Send Buffer μέχρι ένα ROUTE REPLY να ληφθεί από τον κόμβο, όμως ο κόμβος αυτός δεν πρέπει να ξεκινήσει μία καινούργια διαδικασία Εύρεσης Διαδρομής μέχρι να φτάσει η στιγμή εκείνη που επιτρέπεται από τον περιορισμό, να στείλει μία καινούργια Αίτηση Δρομολόγησης.

3.1.2.2. Επιπρόσθετα στοιχεία του Μηχανισμού Εύρεσης Διαδρομών

Αποθήκευση Πληροφοριών Δρομολόγησης μέσω ακρόασης (Caching Overheard Routing Information)

Ένας κόμβος που προωθεί ή διαφορετικά ακούει (παρατηρεί) κάποιο πακέτο, μπορεί αυτός να προσθέσει τις πληροφορίες δρομολόγησης που αυτό περιέχει, στην δική του Λίστα Διαδρομών (Route Cache). Γενικότερα, το μονοπάτι που είναι αποθηκευμένο σε ένα πακέτο δεδομένων ή σε ένα πακέτο τύπου ROUTE REQUEST ή ROUTE REPLY, μπορεί να διαβαστεί και να αποθηκευθεί από τον οποιοδήποτε κόμβο που τύχει να το λάβει, είτε ως παραλήπτης, είτε ως προωθητικός κόμβος, είτε ως ακροατής.

Η μόνη περίπτωση στην οποία ένας κόμβος δεν μπορεί να αποθηκεύσει ένα μονοπάτι, είναι γιατί αυτό δεν είναι αμφίδρομο. Για παράδειγμα, στο παρακάτω σχήμα, υποθέτουμε πως ο κόμβος C προωθώντας το πακέτο στον κόμβο D θέλει να αποθηκεύσει την διαδρομή. Η διαδρομή που μπορεί να αποθηκεύσει για να είναι σίγουρος ότι ισχύει είναι αυτή που συνεχίζει μετά από αυτόν. Δηλαδή, από τον C στον D και από τον D στον E. Η μόνη περίπτωση που μπορεί να αποθηκεύσει το μονοπάτι από το οποίο ήρθε σε αυτόν το πακέτο, είναι η περίπτωση να γνωρίζει πως οι συνδέσεις αυτής της διαδρομής είναι αμφίδρομες. Τότε δίνεται η δυνατότητα στον C να αποθηκεύσει δύο μονοπάτια στην Λίστα Διαδρομών του, το C->D->E και το C->B->A.



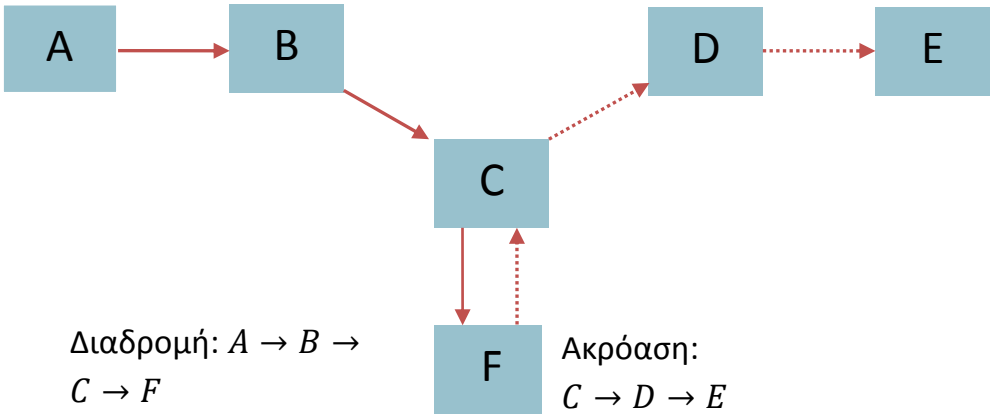
Περιορισμοί Αποθήκευσης Πληροφοριών μέσω ακρόασης: Ο κόμβος C προωθεί πακέτο στον E ενώ ο X αντιλαμβάνεται το πακέτο

Κατά τον ίδιο τρόπο, ο κόμβος V του παραπάνω σχήματος, χρησιμοποιεί μία διαφορετική διαδρομή για να επικοινωνήσει με τον κόμβο Z. Αν ο κόμβος C ακούσει ότι ο κόμβος X προωθεί ένα πακέτο από τον V στον Y, τότε ο κόμβος C πρέπει να αποφασίσει εάν οι συνδέσεις που εμπλέκονται στην μετάδοση είναι αμφίδρομες ή όχι, πριν αποφασίσει να αποθηκεύσει κάποια διαδρομή. Εάν η σύνδεση από τον X στον C είναι γνωστή ως αμφίδρομη, τότε ο C μπορεί να αποθηκεύσει την διαδρομή C->X->Y->Z. Εάν όλες οι συνδέσεις μπορούν να θεωρηθούν αμφίδρομες τότε, εκτός από το παραπάνω μονοπάτι, ο C μπορεί να αποθηκεύσει και την διαδρομή C->X->W->V.

Απάντηση σε Αιτήσεις Δρομολόγησης με Αποθηκευμένες Διαδρομές (Replying to ROUTE REQUESTS using Cached Routes)

Ένας κόμβος που λαμβάνει ένα ROUTE REQUEST για το οποίο δεν είναι ο στόχος, ψάχνει την δική του Λίστα Διαδρομών για μία διαδρομή μέχρι τον στόχο του Αιτήματος. Εάν τη βρει, αντί να προωθήσει το ROUTE REQUEST στον επόμενο κόμβο, ο κόμβος απαντάει με ένα ROUTE REPLY στον εισηγητή επισυνάπτοντας την διαδρομή που βρήκε στην μνήμη του μαζί με αυτήν που το πακέτο έχει καταγράψει μέχρι στιγμής.

Ωστόσο, πριν στείλουν το ROUTE REPLY που κατασκευάζουν χρησιμοποιώντας πληροφορίες από την Λίστα Διαδρομών τους, οι κόμβοι πρέπει να επιβεβαιώσουν πως η διαδρομή που θα προκύψει δεν θα περιέχει παραπάνω από μία φορά έναν προωθητικό κόμβο. Για παράδειγμα, το παρακάτω σχήμα φανερώνει μία περίπτωση κατά την οποία ένα ROUTE REQUEST για τον στόχο E έχει ληφθεί από τον κόμβο F, και ο κόμβος F έχει ήδη μία διαδρομή στην Λίστα του για τον κόμβο E. Όμως αν την προσθέσει στο μονοπάτι της λίστας του πακέτου, τότε θα δημιουργηθεί μία επαναληπτική διαδρομή στον κόμβο C.



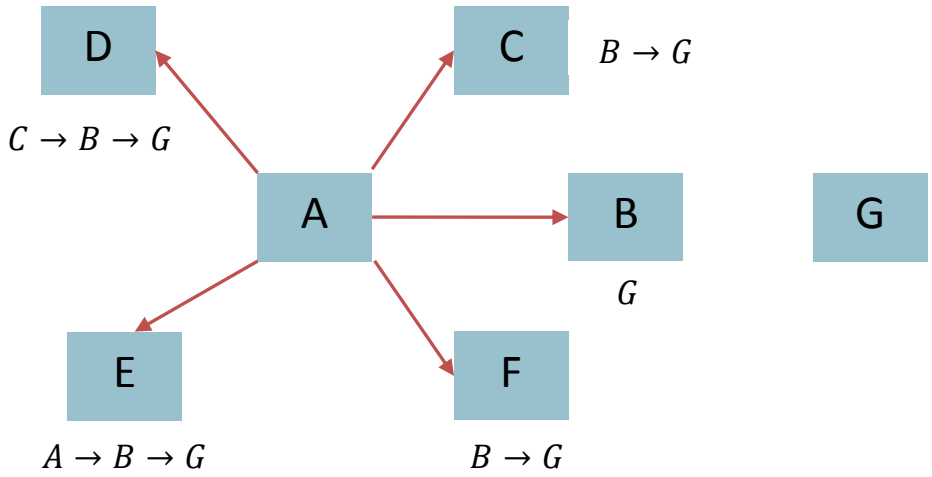
Πιθανή επανάληψη Διαδρομών που αποτρέπει η Εύρεση Διαδρομών μέσω της διαδικασίας απάντησης των Αιτημάτων Δρομολόγησης με χρήση της Λίστας Διαδρομών

Ο κόμβος F σε αυτή την περίπτωση θα μπορούσε να προσπαθήσει να διορθώσει την επανάληψη από την διαδρομή, το οποίο θα είχε ως αποτέλεσμα μία διαδρομή με τη μορφή $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$,όμως με αυτόν τον τρόπο, ο κόμβος F δεν θα μπορεί να βρίσκεται στην λίστα του μονοπατιού που θα στείλει μέσω του ROUTE REPLY. Η διαδικασία Εύρεσης Διαδρομών του DSR απαγορεύει στον κόμβο F, σε αυτή την περίπτωση, να προσθέσει τον εαυτό του στην λίστα διαδρομής της Αίτησης Δρομολόγησης. Αυτό συμβαίνει για δύο λόγους: Πρώτον αυτός ο περιορισμός αυξάνει την πιθανότητα η διαδρομή που προκύπτει να είναι έγκυρη, αφού σε αυτή την περίπτωση ο F θα έπρεπε να είχε λάβει ένα Σφάλμα Δρομολόγησης εάν προηγουμένως η

σύνδεση είχε διακοπεί, και δεύτερον, αυτός ο περιορισμός σημαίνει πως ένα Σφάλμα Δρομολόγησης που προέρχεται από το μονοπάτι, είναι πολύ πιθανόν να περάσει από οποιονδήποτε κόμβο έχει στείλει ROUTE REPLY (το οποίο περιέχει τον F), αυτό βοηθάει στο να εξασφαλίζουν οι κόμβοι πως ξεπερασμένες διαδρομές (όπως με την F) αφαιρούνται από την Λίστα Διαδρομών τακτικά. Διαφορετικά, η επόμενη διαδικασία Εύρεσης Διαδρομής που διενεργείτε από τον A μπορεί επίσης να μολυνθεί από μία απάντηση του F που θα περιέχει πάλι την ίδια ξεπερασμένη διαδρομή. Εάν το ROUTE REQUEST δεν συναντήσει τους παραπάνω περιορισμούς, ο κόμβος F θα προτιμήσει να απορρίψει το πακέτο παρά να το προωθήσει.

Αποτροπή Καταιγιστικής Λήψης Απαντήσεων σε Αιτήματα Δρομολόγησης (Preventing ROUTE REPLY Storms)

Η δυνατότητα που δίνει ο DSR στους κόμβους να απαντάν σε Αιτήσεις Διαδρομών από τις Λίστες Διαδρομών τους μπορεί να προκαλέσει σε μερικές περιπτώσεις, καταιγιστική εισροή από Απαντήσεις Αιτημάτων Δρομολόγησης. Πιο συγκεκριμένα, όταν ένας κόμβος μεταδίδει ένα ROUTE REQUEST στους γειτονικούς του, και αυτοί με τη σειρά τους έχουν αποθηκευμένες στις λίστες Διαδρομών τους από μία διαδρομή για τον στόχο, τότε ο κάθε γείτονας θα επιχειρήσει να στείλει ένα ROUTE REPLY στον εισηγητή. Αυτή η ενέργεια αν εκτελεσθεί ταυτόχρονα από τους περισσότερους (αν όχι όλους) τους γειτονικούς κόμβους, θα προκαλέσει συγκρούσεις πακέτων και αυξημένη κίνηση στο κανάλι του κόμβου που διενεργεί την Αίτηση Διαδρομής.
Για παράδειγμα, στο παρακάτω σχήμα, οι κόμβοι B,C,D,E και F θα λάβουν ένα ROUTE REQUEST από τον κόμβο A με στόχο τον κόμβο G. Έστω ότι ο κάθε ένας από τους γειτονικούς αυτούς κόμβους, έχει στην μνήμη του αποθηκευμένη μία διαδρομή για τον G. Σε φυσιολογική κατάσταση, όλοι θα επιχειρήσουν να απαντήσουν στην Αίτηση του A περίπου την ίδια χρονική στιγμή. Αυτό όμως, θα προκαλούσε συγκρούσεις και σύγχυση στο κανάλι του A.



Απεικόνιση μιας πιθανής καταιγιστικής Απάντησης σε Αίτημα Δρομολόγησης από τον A στον G

Εάν ένας κόμβος μπορεί να ενεργοποιήσει την λειτουργία λήψης χωρίς περιορισμούς (promiscuous receive mode), θα μπορούσε να καθυστερήσει την αποστολή του πακέτου του για ένα μικρό χρονικό διάστημα, ενόσω θα προσπαθούσε να εντοπίσει στο κανάλι μήπως κάποιος άλλος κόμβος στείλει μία διαδρομή μικρότερη από αυτήν που έχει να προτείνει ο ίδιος. Η καθυστέρηση αυτή, μπορεί να προκύψει για μία τυχαία χρονική περίοδο, η οποία μπορεί να καθοριστεί από μία συνάρτηση της μορφής:

$$d = H \times (h - 1 + r)$$

, όπου h είναι η απόσταση σε κόμβους από τον κόμβο εισηγητή, r ένας τυχαίος αριθμός μεταξύ του 0 και του 1, και H είναι η μικρότερη καθυστέρηση που κάνει το πακέτο να φτάσει από έναν κόμβο σε έναν άλλο χωρίς να μεσολαβούν προωθητικοί κόμβοι.

Καθώς ένας κόμβος, περιμένει να έρθει η σειρά του να στείλει την Απάντησή του στην Αίτηση Δρομολόγησης, προσπαθεί να εντοπίσει στο κανάλι πακέτα δεδομένων που έχουν ως αποστολέα τον εισηγητή του ROUTE REQUEST και προορισμό τον στόχο αυτού. Εάν εντοπίσει ένα τέτοιο πακέτο, τότε ελέγχει αν η διαδρομή που ακολουθείτε είναι μικρότερη ή ίση από αυτή που έχει να προτείνει ο ίδιος. Εφόσον η διαδρομή αυτή είναι μεγαλύτερη από τη δική του, ο κόμβος περιμένει την σειρά του να στείλει το ROUTE REPLY, αντιθέτως, εάν είναι μικρότερη ή ίση, ο κόμβος ακυρώνει τον χρόνο καθυστέρησης και δεν στέλνει καμία απάντηση στον εισηγητή του Αιτήματος Δρομολόγησης.

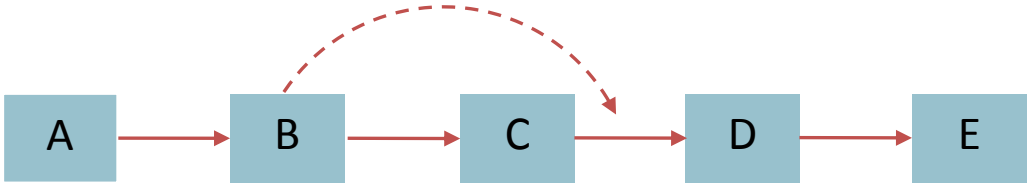
Αίτηση Δρομολόγησης με περιορισμό στις μεταδόσεις (ROUTE REQUEST Hop Limits)

Η κάθε Αίτηση Δρομολόγησης περιέχει και ένα όριο μεταδόσεων το οποίο μπορεί να χρησιμοποιηθεί για να περιορίσει τον αριθμό των ενδιάμεσων κόμβων που επιτρέπεται να προωθήσουν το πακέτο. Καθώς η Αίτηση μεταδίδεται, αυτό το όριο μειώνεται, όταν το όριο φτάσει το μηδέν το πακέτο Αίτησης Δρομολόγησης απορρίπτεται πριν βρει τον προορισμό του. Αυτή η τεχνική μπορεί να φανεί πολύ χρήσιμη, καθώς ένας κόμβος μπορεί να βρει εάν ο στόχος που ψάχνει είναι γείτονάς του ή ένας γείτονάς του έχει ήδη μία διαδρομή για τον στόχο του, πολύ πιο γρήγορη. Το μόνο που έχει να κάνει, είναι να ορίσει το όριο μεταδόσεων στο 0 και να περιμένει μία σύντομη απάντηση. Εάν περάσει μεγαλύτερος χρόνος από όσο περίμενε, τότε στέλνει ένα ROUTE REQUEST με μεγαλύτερα (ή και καθόλου) όρια μεταδόσεων.
Ο μηχανισμός που περιγράψαμε θα μπορούσε να χρησιμοποιηθεί για την υλοποίηση μιας κυκλικής διαδικασίας εξάπλωσης της αναζήτησης (expanding ring search). Πιο συγκεκριμένα, ένας κόμβος που εκτελεί διαδικασία Αίτησης Δρομολόγησης, στέλνει αρχικά μόνο στους γείτονές του με τον τρόπο που περιγράψαμε παραπάνω, όταν ο χρόνος αναμονής απάντησης λήξει, μπορεί να αυξήσει στο 1 το όριο μετάδοσης και να ξαναστείλει το πακέτο Αίτησης. Εάν και τότε ο χρόνος αναμονής λήξει, αυξάνει το όριο μετάδοσης στο 2 και επαναλαμβάνει. Αυτό μπορεί να συμβεί αρκετές φορές, μέχρι τελικά ο αιτώντας κόμβος βρει τον στόχο του.
Αν και αυτή η κυκλική εξάπλωση της αναζήτησης μπορεί να προκαλέσει μεγαλύτερη καθυστέρηση στην εύρεση της διαδρομής, το πακέτο ROUTE REQUEST τις περισσότερες φορές δεν θα χρειάζεται να διατρέχει ολόκληρο το δίκτυο.

3.1.2.3. Αναλυτική περιγραφή του Μηχανισμού Διατήρησης Διαδρομών

Όταν κατασκευάζεται ή προωθείται ένα πακέτο που χρησιμοποιεί μία προσχεδιασμένη διαδρομή, ο κάθε κόμβος που μεταδίδει το πακέτο ευθύνεται να επιβεβαιώσει πως ο επόμενος σε σειρά κόμβος στην διαδρομή έχει παραλάβει το πακέτο. Το πακέτο μπορεί να μεταδοθεί παραπάνω από μία φορά (μέχρι μία τιμή που περιορίζει τις απόπειρες) μέχρι να επιβεβαιωθεί πως το πακέτο έχει μεταδοθεί επιτυχώς. Για παράδειγμα, στην

περίπτωση του παρακάτω σχήματος, ο Α κατασκευάζει ένα πακέτο για τον Ε με χρήση του μονοπατιού $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$. Σε αυτή την περίπτωση, ο κόμβος Α είναι υπεύθυνος για την επιτυχή μετάδοση στον Β, ο Β στον C κ.ο.κ. μέχρι το πακέτο να φτάσει επιτυχώς στον προορισμό του Ε. Η επιβεβαίωση της αποστολής, στις περισσότερες περιπτώσεις μπορεί να παρέχεται ελεύθερα στον DSR, είτε σαν ένα υπάρχον τμήμα του MAC πρωτοκόλλου σε χρήση, είτε μέσο ενός παθητικού αναγνωριστικού (passive acknowledgement) στο οποίο για παράδειγμα, ο Β επιβεβαιώνει την επιτυχή αποστολή στον C μέσω της ακρόασης του καναλιού όταν ο C προωθήσει το πακέτο στον D.



Εάν κανένας από τους παραπάνω τρόπους επιβεβαίωσης δεν είναι διαθέσιμος, ο κόμβος που προωθεί το πακέτο μπορεί να θέσει ένα bit στην κεφαλίδα του πακέτου για να απαιτήσει από τον DSR ένα συγκεκριμένο λογισμικό αναγνώρισης αποστολής να επιστραφεί από τον επόμενο κόμβο· αυτό το λογισμικό αναγνώρισης θα πρέπει φυσιολογικά να μεταδίδεται κατευθείαν από τον κόμβο αποστολέα, αλλά αν η σύνδεση μεταξύ των δύο κόμβων είναι μονής κατεύθυνσης, τότε το αναγνωριστικό μπορεί να ταξιδέψει από διαφορετικό μονοπάτι.

Εάν το πακέτο μεταδοθεί όσες φορές είναι το επιτρεπτό όριο, και ακόμη καμία απάντηση δεν ληφθεί στον κόμβο που το προωθεί, τότε και εκείνος με τη σειρά του στέλνει ένα πακέτο Εσφαλμένης Διαδρομής (ROUTE ERROR) στον αρχικό αποστολέα του μηνύματος, προσδιορίζοντας το σημείο εκείνο της μετάδοσης που παρουσιάζει το σφάλμα. Για παράδειγμα, έστω ότι στο παραπάνω σχήμα, ο κόμβος C δεν μπορεί να επιβεβαιώσει την αποστολή μηνυμάτων στον D όσες φορές και αν έχει προσπαθήσει να το προωθήσει. Τότε ο C θα επιστρέψει ένα πακέτο Εσφαλμένης Διαδρομής στον Α, και θα τον ενημερώσει πως η σύνδεση μεταξύ αυτού και του D δεν είναι εφικτή. Ο κόμβος Α τότε, θα διαγράψει την διαδρομή αυτή από την Λίστα Διαδρομών του και θα ψάξει να βρει εάν έχει αποθηκευμένη μία εναλλακτική διαδρομή

με προορισμό τον Ε. Εάν βρει, τότε κατευθείαν προωθεί το πακέτο με την καινούργια διαδρομή, εάν όχι, εκτελεί την διαδικασία Εύρεσης Διαδρομής.

3.1.2.4. Επιπρόσθετα στοιχεία του Μηχανισμού Διατήρησης Διαδρομών

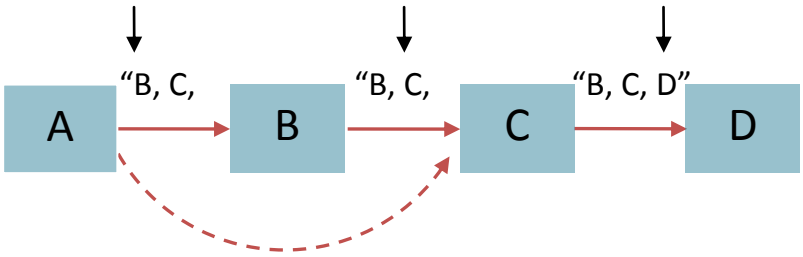
Διάσωση Πακέτων (Packet Salvaging)

Αφού σταλεί ένα μήνυμα Εσφαλμένης Διαδρομής (ROUTE ERROR) ως τμήμα του μηχανισμού Διατήρησης Διαδρομών, ένας κόμβος μπορεί να επιχειρήσει να διασώσει τα δεδομένα του πακέτου που προκάλεσε το ROUTE ERROR παρά να το απορρίψει. Στην απόπειρα διάσωσης του πακέτου, ο κόμβος που στέλνει ένα μήνυμα Εσφαλμένης Διαδρομής, ταυτόχρονα ψάχνει στην δική του Λίστα Διαδρομών (Route Cache) για μία εναλλακτική διαδρομή από τον ίδιο προς τον προορισμό του πακέτου. Αν μία τέτοια διαδρομή βρεθεί, ο κόμβος μπορεί να καταφέρει να διασώσει τελικά το πακέτο αντικαθιστώντας την αρχική του διαδρομή με την εναλλακτική που βρήκε στην μνήμη του. Ο κόμβος στη συνέχεια προωθεί το πακέτο στον επόμενο μεταδότη που βρίσκεται στην καινούργια διαδρομή.
Κατά τη διαδικασία διάσωσης του πακέτου, αυτό μαρκάρεται με τέτοιο τρόπο ώστε να φαίνεται πως έχει διασωθεί. Αυτό συμβαίνει για να αποτρέψει το ίδιο πακέτο να διασωθεί πολλαπλές φορές. Διαφορετικά, θα μπορούσε το πακέτο να εισέλθει σε έναν βρόχο επαναλαμβανόμενων διαδρομών. Ένας εναλλακτικός μηχανισμός διάσωσης θα μπορούσε να είναι η αντικατάσταση μόνο της άχρηστης διαδρομής του αρχικού μονοπατιού και η αντικατάστασή της από την νέα διαδρομή του κόμβου “σωτήρα”. Σε αυτή την περίπτωση οι κανόνες που αποτρέπουν την επανάληψη διαδρομών στην λίστα ενός πακέτου είναι αρκετές για να αποτρέψουν βρόχους επαναλαμβανόμενων διαδρομών.

Αυτόματη Σμίκρυνση Διαδρομών (Automatic Route Shortening)

Οι διαδρομές που χρησιμοποιούνται από τους κόμβους μπορούν να μικρύνουν αυτόματα εάν ένας ή περισσότεροι ενδιάμεσοι κόμβοι του μονοπατιού πάψουν να είναι χρήσιμη για την σωστή προώθηση. Πιο συγκεκριμένα, εάν ένας κόμβος είναι σε θέση να ακούσει ένα πακέτο που περιέχει μία διαδρομή στην οποία αυτός είναι μέλος της αλλά ακόμα το πακέτο δεν έχει περάσει από αυτόν, υποθέτει πως ο προηγούμενος/οι

κόμβος/οι που βρίσκεται στην διαδρομή πριν από αυτόν μπορεί να παρακαμφθεί. Τότε ο κόμβος αυτός στέλνει ένα μήνυμα της μορφής ROUTE REPLY στον αποστολέα του μηνύματος και τον ενημερώνει για αυτήν την καινούργια διαδρομή που μπορεί το πακέτο του να ακολουθήσει.



Ο κόμβος C αντιλαμβάνεται πως η διαδρομή στον D μπορεί να μικρύνει παραλείποντας τον κόμβο B

Για παράδειγμα, στο παραπάνω σχήμα ο κόμβος C ακούει το πακέτο που στέλνει ο A στον B και έχει προορισμό τον D περνώντας όμως πρώτα από τον ίδιο. Τότε ο C καταλαβαίνει πως η διαδρομή μπορεί να μικρύνει παραλείποντας τον κόμβο B από το μονοπάτι (αφού ο ίδιος ήταν σε θέση να ακούσει την μετάδοση, τότε βρίσκεται εντός της εμβέλειας του κόμβου A). Τότε, ο κόμβος C στέλνει ένα μήνυμα της μορφής ROUTE REPLY το οποίο περιέχει την καινούργια διαδρομή A, C, D που μπορεί μελλοντικά να χρησιμοποιήσει ο A για να προωθεί τα πακέτα του στον D.

Αυξημένη Διάδοση των μηνυμάτων Εσφαλμένων Διαδρομών (Increased Spreading of Route Error Messages)

Όταν ένας κόμβος αποστολέας λάβει ένα μήνυμα Εσφαλμένων Διαδρομών από ένα πακέτο δεδομένων που έστειλε, τότε ο κόμβος αυτός αναμεταδίδει το μήνυμα σφάλματος στους γειτονικούς του κόμβους αφού το επισυνάψει στο επόμενο μήνυμα Εύρεσης Διαδρομών που θα στείλει. Με αυτόν τον τρόπο, μη ενημερωμένες Λίστες Διαδρομών των κόμβων που αποτελούν αναμεταδότες του πακέτου, δεν θα στέλνουν Απαντήσεις σε Αιτήματα Διαδρομής, εάν εμπεριέχουν την σύνδεση η οποία προκάλεσε το Σφάλμα.

Εντοπισμός Αρνητικών Πληροφοριών (Caching Negative Information)

Σε μερικές περιπτώσεις, το DSR μπορεί να επωφεληθεί από κόμβους που έχουν αποθηκευμένη “αρνητική” πληροφορία μέσα στις Λίστες Διαδρομών τους. Για παράδειγμα, στο παρακάτω σχήμα, εάν ο κόμβος A εντοπίσει πως η σύνδεση μεταξύ του C και του D δεν είναι εφικτή (αντί απλώς να διαγράψει αυτήν τη διαδρομή από την Route Cache), θα μπορούσε να εγγυηθεί ότι κανένα μήνυμα ROUTE REPLY που λαμβάνει δεν θα τον κάνει να επαναλάβει την εγγραφή της διαδρομής C->D στην λίστα του. Ωστόσο, θα πρέπει να υπάρξει μία σύντομη ώρα λήξης της παραμονής της αρνητικής πληροφορίας στη μνήμη, γιατί διαφορετικά ο A δεν θα ξαναχρησιμοποιήσει ποτέ την σύνδεση C->D ακόμα και αν αυτή επανέλθει μετά από κάποιο χρονικό διάστημα.

Παράδειγμα μη επιτυχούς απόπειρας Εύρεσης Διαδρομών : Ο κόμβος C δεν μπορεί να προωθήσει στον D και ως συνέπεια το πακέτο δεν μπορεί να φτάσει στον στόχο E

Μία ακόμα περίπτωση στην οποία η χρήση αρνητικής πληροφορίας μπορεί να φανεί χρήσιμη είναι η περίπτωση στην οποία μία σύνδεση που παρέχει αρκετά πολύτιμη υπηρεσία δεν είναι σταθερή στην απόδοσή της (δηλαδή μερικές φορές λειτουργεί και άλλες όχι). Αυτή η κατάσταση μπορεί να προκληθεί, για παράδειγμα, όταν η σύνδεση βρίσκεται κοντά στα όρια της εμβέλειας αποστολής του ασύρματου κόμβου και υπάρχουν σημαντικές πηγές παρεμβολών. Σε αυτήν την περίπτωση, με την αποθήκευση της αρνητικής πληροφορίας της μη αξιόπιστης σύνδεσης , ένας κόμβος θα αποφύγει να προσθέσει την προβληματική αυτή σύνδεση στην Λίστα Διαδρομών του. [1]

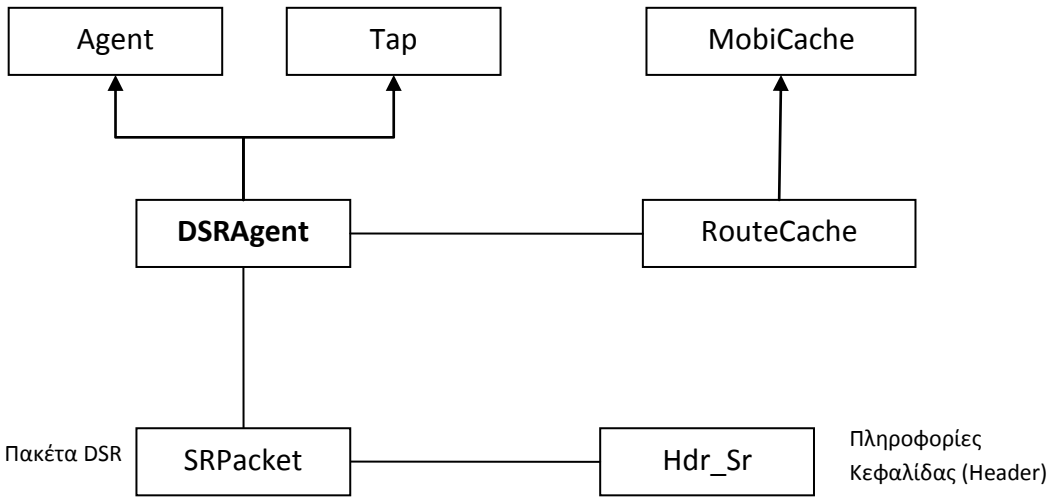
3.2. Ο DSR στον προσομοιωτή NS-2

Το DSR πρωτόκολλο δρομολόγησης υπάρχει ήδη υλοποιημένο στον προσομοιωτή δικτύων NS-2. Για να μπορέσουμε να προχωρήσουμε στην υλοποίηση, πρέπει να μελετήσουμε τον κώδικα που το εκτελεί.

Επειδή η υλοποίηση του DSR έχει αρκετά μεγάλο μέγεθος, για λόγους συντομίας θα αναλυθούν μόνο οι βασικές συναρτήσεις και τάξεις.

3.2.1. Βασικές Τάξεις και Συναρτήσεις στο DSR

Το παρακάτω σχήμα, απεικονίζει τις βασικές τάξεις του DSR κώδικα και το πώς συνδέονται μεταξύ τους:



Διάγραμμα Βασικών Κλάσεων της υλοποίησης του DSR στον NS-2

Όπως φαίνεται από το διάγραμμα, η βασική τάξη που χρησιμοποιεί ο DSR είναι η DSRAgent η οποία είναι παράγωγη της τάξης Agent και Tap. Η τάξη Agent είναι υπεύθυνη για την αναγνώριση της DSRAgent ως τάξη που μπορεί να προσομοιωθεί από τον NS-2. Η Tap είναι εκείνη μέσω της οποίας το DSR μπορεί να ακούσει πακέτα από το κανάλι (όταν η λειτουργία λήψης χωρίς περιορισμούς είναι ενεργοποιημένη, promiscuous mode on).

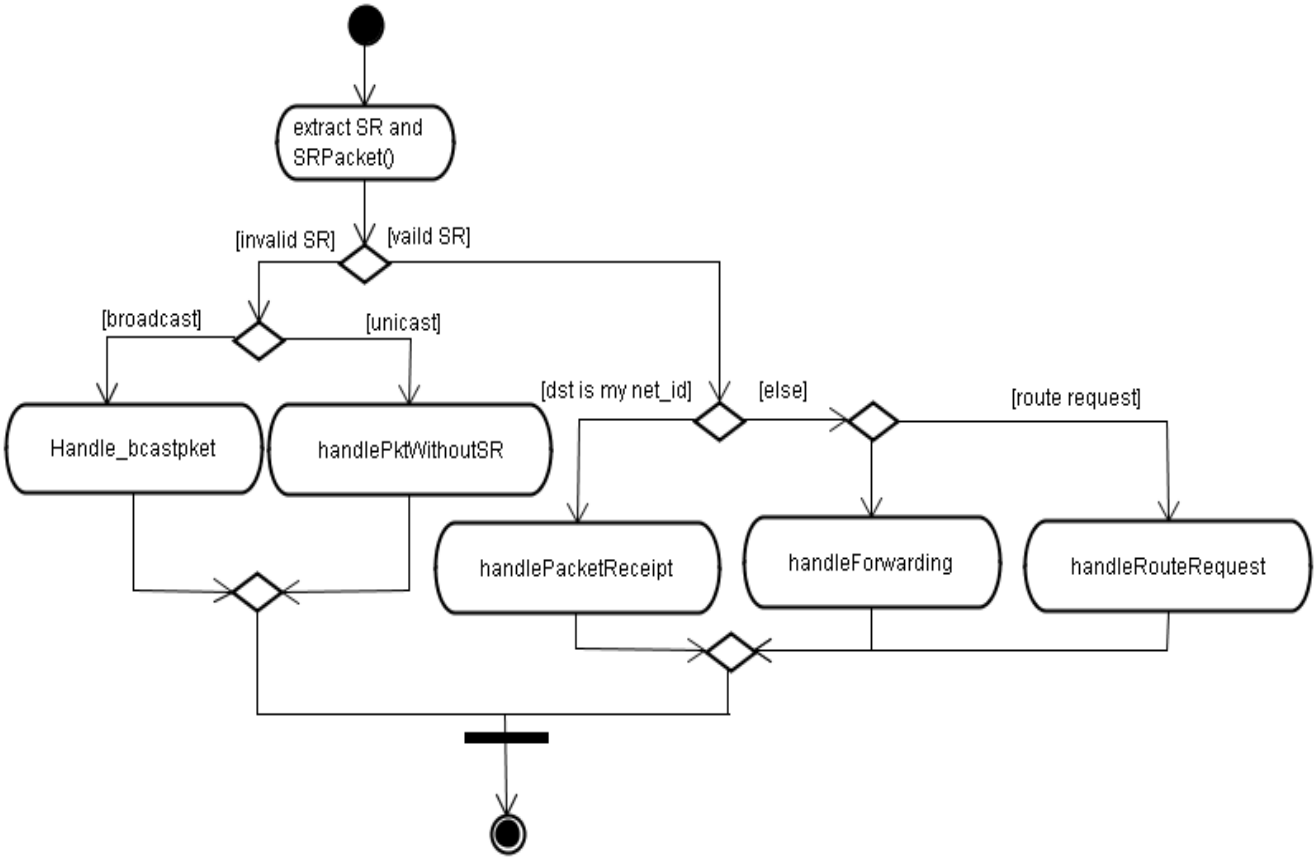
Η τάξη DSRAgent συνδέεται άμεσα και με την τάξη RouteCache. Η τάξη αυτή είναι η βάση για άλλες τάξεις, όπως η MobiCache που είναι προεπιλεγμένη στον NS-2, οι οποίες περιέχουν στρατηγικές για το πώς θα γίνεται η

αποθήκευση στην Λίστα Διαδρομών και το πώς θα επιλέγονται οι κατάλληλες διαδρομές για πακέτα που εξέρχονται από τον εκάστοτε κόμβο.

Μία ακόμη τάξη που συνδέεται άμεσα με την DSRAgent είναι η SRPacket. Η τάξη αυτή είναι υπεύθυνη για την δομή των πακέτων που δρομολογούνται και προσφέρει μεθόδους λήψης και τοποθέτησης πηγαίων διαδρομών στα πακέτα.

Η hdr_sr είναι μία βοηθητική τάξη η οποία περιέχει ορισμούς που βοηθούν στην ευκολότερη υλοποίηση του DSR.

Το παρακάτω διάγραμμα δίνει την κύρια δομή των βασικών συναρτήσεων που χρησιμοποιούνται στην τάξη DSRAgent. Η κάθε μία από αυτές καλείται από την βασική συνάρτηση recv() (receive) στην οποία καταφθάνουν όλα τα εισερχόμενα πακέτα στον κώδικα, εκείνη με τη σειρά της επιλέγει με τον παρακάτω τρόπο πως θα τα χειριστεί:



Διάγραμμα Ροής Εισερχόμενου Πακέτου στο DSR

Το πρώτο που ελέγχεται από το εισερχόμενο πακέτο, είναι αν η πηγαία του διαδρομή (Source Route ή SR) είναι έγκυρη, δηλαδή εάν υπάρχει:

- Εάν δεν υπάρχει πηγαία διαδρομή στο πακέτο, τότε αυτό ή θα είναι πακέτο προς broadcast μετάδοση και στέλνεται στην Handle_bcastpket, ή θα είναι εξερχόμενο πακέτο που χρειάζεται Πηγαία Διαδρομή και αναλαμβάνεται από την συνάρτηση handlePKTWithoutSR.
- Εάν το πακέτο περιέχει πηγαία διαδρομή, τότε ελέγχεται εάν η διεύθυνση του παραλήπτη ταιριάζει με το net_id του κόμβου. Εάν αυτό ισχύει, τότε η handlePacketReceipt αναλαμβάνει. Εάν πάλι η διεύθυνση παραλήπτη δεν ταιριάζει με τον net_id του κόμβου, τότε το πακέτο αυτό θα είναι ή ROUTE REQUEST προς άλλο κόμβο, ή πακέτο προς προώθηση ή άκυρο πακέτο. Εάν είναι ROUTE REQUEST καλείται η συνάρτηση handleRouteRequest, ενώ αν είναι πακέτο δεδομένων προς προώθηση καλείται η handleForwarding.

Ιδιαίτερη σημασία δίνεται στην συνάρτηση handlePacketReceipt. Σε αυτή την περίπτωση, δύο είναι τα πιθανά πακέτα που έχουν προορισμό τον κόμβο, πακέτο ROUTE REQUEST με στόχο τον εκάστοτε κόμβο, ή πακέτο ROUTE REPLY που είναι απάντηση από ROUTE REQUEST που είχε ο κόμβος εκδώσει νωρίτερα. Στην πρώτη περίπτωση, ο κόμβος καλείται να εκδώσει ένα πακέτο προς απάντηση στην αίτηση, αυτή την διαδικασία την αναλαμβάνει η συνάρτηση returnSrcRouteToRequestor(p), όπου p το πακέτο ROUTE REPLY. Στην δεύτερη περίπτωση, το πακέτο περιέχει την διαδρομή που ο κόμβος είχε ρωτήσει νωρίτερα, γι' αυτό πρέπει να αποθηκευτεί στην Λίστα Διαδρομών του, την διαδικασία αυτήν την αναλαμβάνει η συνάρτηση acceptRouteReply(p).

Μία επίσης σημαντική συνάρτηση του DSR κώδικα είναι η sendOutPacketWithRoute(). Η συνάρτηση αυτή, όπως λέει και το όνομά της, αναλαμβάνει την αποστολή οποιουδήποτε εξερχόμενου μηνύματος από τον κόμβο. Η αποστολή μηνυμάτων από τον κόμβο, γίνεται μόνο όταν υπάρχει μία Πηγαία Διαδρομή (SR) σε αυτά.

Τέλος, μία συνάρτηση που είναι αρκετά χρήσιμη και για την υλοποίησή μας είναι η tap(const Packet *packet). Αυτή είναι μία συνάρτηση ανάλογη της recv(). Σε αυτήν καταφθάνουν τα πακέτα που ακούγονται από το κανάλι όταν η λειτουργία λήψης χωρίς περιορισμούς είναι ενεργοποιημένη. [20],[21]

3.3. Ενεργοποίηση και Απενεργοποίηση Δυνατοτήτων του DSR

Στην Ανάλυση του DSR πρωτοκόλλου παραπάνω, συναντήσαμε κάποια επιπρόσθετα χαρακτηριστικά που χρησιμοποιεί στους μηχανισμούς του. Ο κώδικας του DSR στον NS μας δίνει την δυνατότητα να ενεργοποιήσουμε ή να απενεργοποιήσουμε όσα από αυτά θέλουμε. Με κριτήριο την καλύτερη λειτουργία του CONFIDANT πρωτοκόλλου θα αναλύσουμε γιατί πρέπει να κρατήσουμε ή να απενεργοποιήσουμε κάποιο από τα χαρακτηριστικά αυτά.

3.3.1. Αποθήκευση Πληροφοριών Δρομολόγησης μέσω Ακρόασης (Caching Overheard Routing Information)

Η δυνατότητα αυτή, όπως εξηγήθηκε και παραπάνω, δίνει την δυνατότητα εύρεσης διαδρομών με τον εναλλακτικό τρόπο της κατασκοπίας του καναλιού. Αυτή η δυνατότητα αυξάνει τον αριθμό των εναλλακτικών διαδρομών που μπορεί να ακολουθήσει ένα πακέτο για να φτάσει στον προορισμό του. Σε ένα δίκτυο με εγωιστικούς κόμβους όσο περισσότερες εναλλακτικές διαδρομές υπάρχουν για επιλογή, τόσο περισσότερες οι πιθανότητες να βρεθούν ασφαλής διαδρομές. Γι αυτό τον λόγο θα κρατήσουμε την δυνατότητα αυτή ενεργοποιημένη στην υλοποίησή μας.

3.3.2. Απάντηση σε Αιτήσεις Δρομολόγησης με Αποθηκευμένες Διαδρομές (Replying to ROUTE REQUESTS using Cached Routes)

Αυτή η δυνατότητα θα πρέπει να απενεργοποιηθεί για τον εξής λόγο: Όταν ένας κόμβος απαντάει σε ένα ROUTE REQUEST χρησιμοποιώντας τις πληροφορίες από τη δική του Λίστα Διαδρομών, αποτρέπει στην Αίτηση Δρομολόγησης να ανακαλύψει παραπάνω διαδρομές. Μία αίτηση Δρομολόγησης μπορεί να ανακαλύψει παραπάνω από μία διαδρομές από έναν αποστολέα προς έναν παραλήπτη, καθώς συνήθως υπάρχουν πολλές επιλογές σε μονοπάτια που μπορεί να ακολουθήσει. Όταν ένας κόμβος διακόψει αυτή τη διαδικασία, η Αίτηση Δρομολόγησης θα γυρίσει πίσω με πολύ λιγότερες διαδρομές. Όπως προαναφέρθηκε, για την υλοποίηση μας ενδιαφέρει οι εναλλακτικές διαδρομές στην Λίστα Διαδρομών να είναι όσο το δυνατόν περισσότερες.

3.3.3. Αίτηση Δρομολόγησης με περιορισμό στις μεταδόσεις (ROUTE REQUEST hop limit)

Η συγκεκριμένη δυνατότητα είναι αρκετά χρήσιμη για το DSR, αφού όπως περιγράφηκε, μπορεί να ελαττώσει αρκετά τον χρόνο με τον οποίο ένας κόμβος βρίσκει μονοπάτια για τον προορισμό που ψάχνει. Επίσης, αποτρέπει την άσκοπη περιπλάνηση πακέτων ROUTE REQUEST στο δίκτυο. Ο μηχανισμός αυτός θα παραμείνει ενεργοποιημένος στην υλοποίηση για να λειτουργεί ως ανασταλτικός παράγοντας στην φόρτωση του δικτύου από πακέτα δρομολόγησης.

3.3.4. Διάσωση Πακέτων
(Packet Salvaging)

Όταν εκτελείτε η διαδικασία διάσωσης πακέτων, περισσότερα πακέτα έχουν την ευκαιρία να φτάσουν επιτυχώς στον προορισμό τους. Αν και η διάσωση των πακέτων πρέπει να γίνει με μεγάλη προσοχή, αυτό το στοιχείο θεωρείται ευεργετικό για την υλοποίησή μας και γι' αυτό πρέπει να μείνει ενεργοποιημένο.

3.3.5. Αυτόματη Σμίκρυνση Διαδρομών
(Automatic Route Shortening)

Η σμίκρυνση διαδρομών είναι μία ενέργεια που δίνει την δυνατότητα στο πακέτο να φτάσει πιο γρήγορα στον προορισμό του. Επειδή όσο λιγότεροι είναι οι κόμβοι από όπου πρέπει να προωθηθεί το πακέτο τόσο μικρότερη είναι η πιθανότητα κάποιο από αυτά να απορριφθεί από κάποιον εγωιστικό κόμβο, η δυνατότητα αυτή θα παραμείνει ενεργοποιημένη και στην υλοποίηση.

Συνολικά, οι δυνατότητες του DSR πρωτοκόλλου που θα ενεργοποιηθούν ή θα απενεργοποιηθούν, δίνονται στον παρακάτω πίνακα:

Caching Overheard Routing Information	Ενεργοποίηση
Replying to ROUTE REQUESTS using Cached Routes	Απενεργοποίηση
ROUTE REQUEST Hop Limit	Ενεργοποίηση

Packet Salvaging	Ενεργοποίηση
Automatic Route Shortening	Ενεργοποίηση

Επιλογή Δυνατοτήτων του DSR

3.4. Υποθέσεις για τους μη συνεργάσιμους κόμβους

Για την υλοποίηση και επαλήθευση του CONFIDANT μηχανισμού, είναι σκόπιμο πρώτα να καθορίσουμε ποιοι είναι και τι ακριβώς κάνουν οι μη συνεργάσιμοι κόμβοι στις προσομοιώσεις.

Όπως περιγράψαμε και στο δεύτερο κεφάλαιο, υπάρχουν δύο ειδών μη συνεργάσιμοι κόμβοι: οι εγωιστικοί κόμβοι και οι κακόβουλοι κόμβοι. Οι εγωιστικοί είναι οι κόμβοι που δεν προωθούν πακέτα για να εξοικονομήσουν ενέργεια, ενώ οι κακόβουλοι κόμβοι πραγματοποιούν διάφορα είδη επιθέσεων με σκοπό να βλάψουν το δίκτυο. Για την συγκεκριμένη υλοποίηση υποθέτουμε πως έχουμε μόνο εγωιστικούς κόμβους στο δίκτυο. Αυτό επιλέχθηκε για την απλότητα της υλοποίησης αλλά και για την εύκολη ανάλυσή της. Πως όμως συμπεριφέρονται οι εγωιστικοί κόμβοι στο δίκτυό μας;

Οι εγωιστικοί κόμβοι που θα προσομοιωθούν παρακάτω εκτελούν τις εξής κακόβουλες ενέργειες:

- Δεν προωθούν πακέτα δεδομένων που δεν προορίζονται για αυτούς. Αυτή είναι και η βασική λειτουργία ενός εγωιστικού κόμβου γι’ αυτό δεν θα συζητηθεί περαιτέρω.
- Δεν προωθούν πακέτα τύπου ROUTE ERROR. Εφόσον δεν τους ενδιαφέρει η απόδοση του δικτύου, το να μεταφέρουν ένα τέτοιο μήνυμα δεν κρίνεται σκόπιμο, γι’ αυτό οι εγωιστικοί κόμβοι το απορρίπτουν.
- Δεν προωθούν πακέτα τύπου ROUTE REPLY. Και αυτό το είδος πακέτων θεωρείτε σημαντικό για την καλή απόδοση του δικτύου, ένας εγωιστικός κόμβος όμως, θέλει μόνο να ελαττώσει την απώλεια της ενέργειάς του, άρα θα απορρίψει και αυτό το είδος πακέτου.

Πως λειτουργούν οι εγωιστικοί κόμβοι πέρα από τις παραπάνω κακόβουλες ενέργειες:

- Οι εγωιστικοί κόμβοι δέχονται όλα τα μηνύματα που προορίζονται για αυτούς.
- Οι εγωιστικοί κόμβοι απαντούν σε Αιτήματα Δρομολόγησης που έχουν στόχο αυτούς. Αυτό συμβαίνει γιατί ένας εγωιστικός κόμβος δεν θέλει να αποκοπεί από το δίκτυο. Εάν ένας κόμβος εκτελεί διαδικασία Εύρεσης Διαδρομής προς έναν εγωιστικό κόμβο, σημαίνει ότι έχει να του στείλει μηνύματα, εάν ο εγωιστικός κόμβος δεν απαντήσει με ROUTE REPLY τότε δεν θα μπορέσει ποτέ να τα λάβει.
- Οι εγωιστικοί κόμβοι προωθούν πακέτα τύπου ROUTE REQUEST. Σε φυσιολογικές συνθήκες, ένας εγωιστικός κόμβος θα έπρεπε να απορρίψει ένα τέτοιο πακέτο γιατί δεν του προσφέρει κάτι. Αν όμως όλοι οι εγωιστικοί κόμβοι απέρριπταν τέτοια πακέτα, τότε κανένας εγωιστικός κόμβος δεν θα υπήρχε στις Λίστες Διαδρομών των υπόλοιπων κόμβων και η υλοποίηση του CONFIDANT θα ήταν κατά το ήμισυ άσκοπη. Για λόγους λοιπόν υγιές συμπερασμάτων, ένας εγωιστικός κόμβος θα προωθεί πακέτα τύπου ROUTE REQUEST.

3.5. Τιμωρία των μη συνεργάσιμων κόμβων

Ο μηχανισμός CONFIDANT πέρα από την αποφυγή των ανάρμοστων κόμβων από τα μονοπάτια που χρησιμοποιεί, έχει σχεδιαστεί έτσι ώστε να τιμωρεί τους μη συνεργάσιμους κόμβους. Οι μηχανισμοί Προώθησης Συνεργασίας εκτός από το να ανιχνεύουν και να αποφεύγουν την ανάρμοστη συμπεριφορά, πρέπει να δίνουν κίνητρα και στους μη συμμορφούμενους κόμβους να αλλάξουν τη συμπεριφορά τους. Για τον παραπάνω λόγο, κρίνεται σκόπιμη η υιοθέτηση τρόπων τιμωρίας των μη συνεργάσιμων κόμβων.

Οι τιμωρίες που υιοθετούνται από την υλοποίηση είναι οι ακόλουθες:

- Δεν επιλέγονται μονοπάτια που περιέχουν μη συνεργάσιμους κόμβους.
- Οι κόμβοι δεν προωθούν πακέτα δεδομένων που προέρχονται από μη συνεργάσιμους κόμβους.
- Οι κόμβοι δεν προωθούν και δεν απαντούν σε πακέτα τύπου ROUTE REQUEST των οποίων ο εισηγητής είναι ένας μη συνεργάσιμος κόμβος.
- Οι κόμβοι δεν προωθούν ή δέχονται μηνύματα τύπου ROUTE ERROR ή ROUTE REPLY των οποίων εισηγητής είναι ένας ανάρμοστος κόμβος.

Κεφάλαιο 4

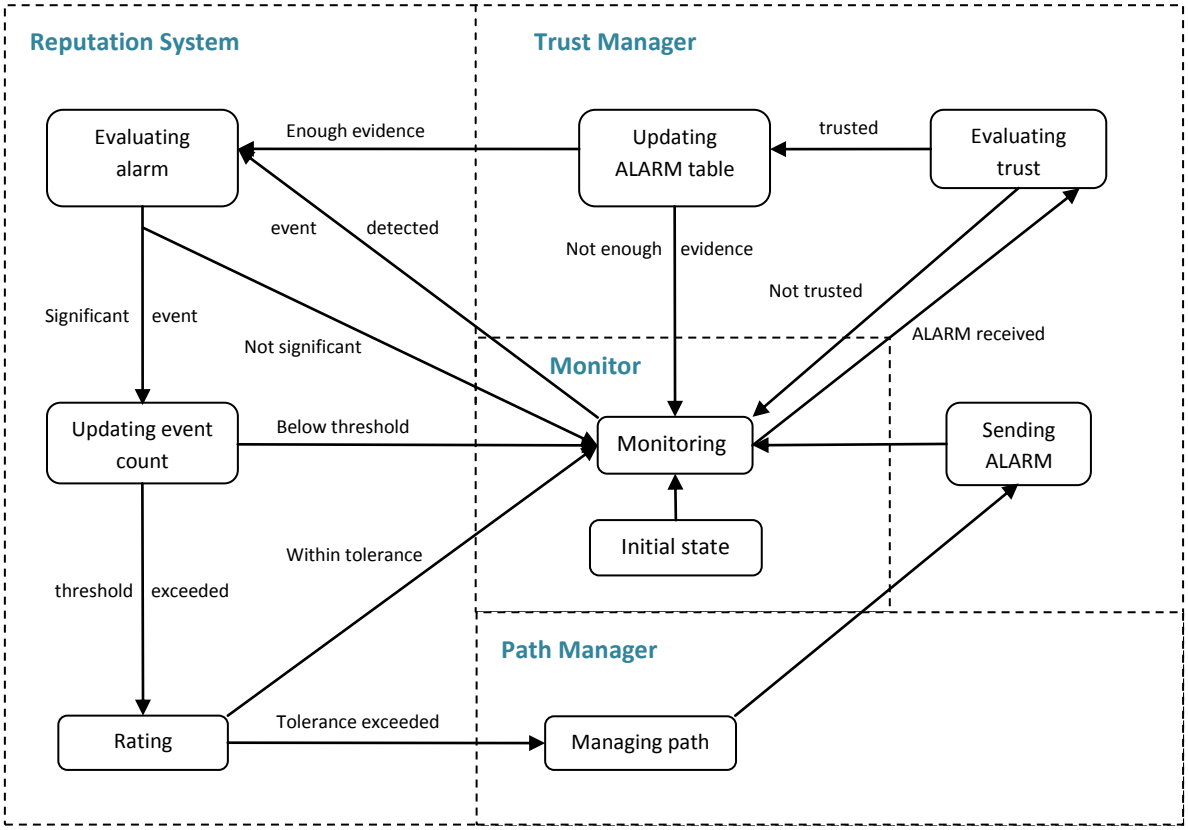
Ανάλυση της Υλοποίησης

Σε αυτό το κεφάλαιο θα παρουσιαστούν τα βήματα υλοποίησης του μηχανισμού CONFIDANT, οι μέθοδοι που χρησιμοποιήθηκαν, καθώς και αναλυτικά διαγράμματα που παρουσιάζουν τον τρόπο λειτουργίας του. Βασικές τάξεις και Συναρτήσεις θα αναλυθούν έτσι ώστε να γίνει κατανοητή η υλοποίηση και στη συνέχεια η εξήγηση των αποτελεσμάτων που θα παρουσιασθούν στο επόμενο κεφάλαιο.

Η βασική δομή και ανάπτυξη του κώδικα στηρίχθηκε στην διδακτορική διατριβή της Shanshan Song .[9]

4.1. Θεωρητική Ανάλυση του Μηχανισμού CONFIDANT

Όπως αναφέραμε και στο κεφάλαιο 2 (παράγραφος 2.3.2) το CONFIDANT πρωτόκολλο λειτουργεί κάνοντας χρήση τεσσάρων βασικών μονάδων που λειτουργούν επιπρόσθετα στις μονάδες του DSR πρωτοκόλλου. Οι μονάδες αυτές είναι: Το Σύστημα Παρακολούθησης ή Monitor, Το Σύστημα Φήμης ή Reputation System, Το Σύστημα Διαχείρισης Διαδρομών ή Path Manager και το Σύστημα Διαχείρισης Αξιοπιστίας ή Trust Manager. Το επόμενο σχήμα μας δείχνει περιληπτικά πως συνεργάζονται αυτές οι μονάδες μεταξύ τους:



Μονάδες του CONFIDANT

4.1.1. Η Διαχείριση των Πακέτων

Το Σύστημα Παρακολούθησης (Monitor) είναι η βασική μονάδα της υλοποίησης, αυτή είναι η μονάδα που επικοινωνεί άμεσα με το DSR και κάνει χρήση των υπόλοιπων μονάδων για την σωστή λειτουργία του Μηχανισμού. Τα πακέτα που χειρίζεται το Σύστημα Παρακολούθησης είναι τα εξής:

- Εξερχόμενα Πακέτα Δεδομένων. Κάθε φορά που ο DSR στέλνει ένα πακέτο σε επόμενο κόμβο, η μονάδα Monitor καταγράφει τα στοιχεία του πακέτου που χρειάζονται για να μπορέσει να καταλάβει αργότερα αν αυτά προωθήθηκαν από τον επόμενο κόμβο.
- Εξερχόμενα πακέτα τύπου ROUTE REQUEST, ROUTE REPLY και ROUTE ERROR. Τα πακέτα αυτά, επίσης καταγράφονται από την μονάδα, για τον ίδιο λόγο με τα Πακέτα Δεδομένων.
- Εισερχόμενα πακέτα που ανιχνεύθηκαν από το κανάλι, όταν η λειτουργία αποδοχής πακέτων χωρίς περιορισμούς (promiscuous mode) είναι ενεργή. Όταν εισέρχεται ένα τέτοιο πακέτο στον κόμβο, η μονάδα Monitor είναι υπεύθυνη να ανιχνεύσει αν πρόκειται για πακέτο που επιβεβαιώνει πως ο επόμενος κόμβος προώθησε το πακέτο που του στάλθηκε.
- Πακέτα τύπου ALARM. Αυτό το πακέτο παρέχει έμμεσες πληροφορίες στο CONFIDANT από παρατηρήσεις άλλων κόμβων.

Ανάλογα με τα παραπάνω πακέτα, το Σύστημα Παρακολούθησης συνεργάζεται με τις υπόλοιπες μονάδες αποστέλλοντας και λαμβάνοντας τις απαραίτητες πληροφορίες για τον χειρισμό των πακέτων αυτών. Πιο αναλυτικά ο χειρισμός των πληροφοριών του κάθε τύπου πακέτου γίνεται ως εξής:

4.1.1.1. Εξερχόμενο Πακέτο Δεδομένων.

Υπάρχουν δύο ειδών εξερχόμενα πακέτα, τα πακέτα που περιέχουν πηγαίες διαδρομές και χρειάζονται μόνο προώθηση και τα πακέτα που δεν περιέχουν πηγαίες διαδρομές και χρειάζονται ενημέρωση.

- Τα πακέτα που περιέχουν πηγαίες διαδρομές αρχικά εξετάζονται αν έχουν σταλεί από κάποιον εγωιστικό κόμβο. Για να γίνει αυτό, ο Manager καλεί το Σύστημα Φήμης (Reputation System) να ελέγξει αν το πακέτο που λήφθηκε έχει αποστολέα με καλή ή κακή φήμη στέλνοντας το αναγνωριστικό του. Αν το σύστημα φήμης επιστρέψει πως ο αποστολέας είναι ένας εγωιστικός κόμβος, τότε

το πακέτο θα απορριφθεί για τιμωρία, αντίθετα αν επιστρέψει πως ο κόμβος αποστολέας έχει καλή φήμη, τότε τα αναγνωριστικά του πακέτου θα αποθηκευθούν από τον MONITOR καθώς και η χρονική στιγμή προώθησης του πακέτου.

- Τα πακέτα που δεν περιέχουν πηγαίες διαδρομές είναι τα μόνα που δεν χειρίζονται από το Monitor. Όταν ένα πακέτο ζητάει πηγαία διαδρομή, τότε απευθείας καλείται από το DSR το σύστημα Διαχείρισης Διαδρομών (Path Manager). Εκεί ο Path Manager ψάχνει μία Διαδρομή στην Λίστα Διαδρομών (Route Cache) με κριτήριο την αποφυγή εγωιστικών κόμβων στο μονοπάτι.

4.1.1.2. Εξερχόμενα Πακέτα ROUTE REQUEST, ROUTE REPLY και ROUTE ERROR.

Τα πακέτα αυτά αντιμετωπίζονται με τον ίδιο τρόπο που αντιμετωπίζονται και τα πακέτα δεδομένων παραπάνω. Όταν έχουν ως αποστολέα κάποιον κακόφημο κόμβο απορρίπτονται, ενώ όταν αποστέλλονται από καλούς κόμβους προωθούνται αφού πρώτα διατηρηθούν οι απαραίτητες πληροφορίες για την εξακρίβωση της προώθησής τους.

Ενδιαφέρον παρουσιάζει η αντιμετώπιση των πακέτων τύπου ROUTE REPLY που αποστέλλονται από τον εκάστοτε κόμβο. Το DSR δίνει δύο πιθανούς τρόπους που ένα πακέτο ROUTE REPLY μπορεί να σταλεί. Ο πρώτος τρόπος είναι να σταλεί από το μονοπάτι από το οποίο προσήλθε η Αίτηση Δρομολόγησης, που προϋποθέτει όλα τα κανάλια επικοινωνίας να είναι αμφίδρομα, και ο δεύτερος είναι να σταλεί με τον τρόπο που στέλνονται και τα πακέτα δεδομένων, δηλαδή να βρεθεί ένα μονοπάτι από την Λίστα Διαδρομών.

Στην υλοποίησή μας, αν και θεωρούμε όλα τα κανάλια επικοινωνίας αμφίδρομα, θα χρησιμοποιήσουμε τον δεύτερο τρόπο αποστολής ROUTE REPLY. Αυτό συμβαίνει για τον εξής λόγο, παραπάνω υποθέσαμε πως οι εγωιστικοί κόμβοι προωθούν πακέτα τύπου ROUTE REQUEST αλλά απορρίπτουν πακέτα τύπου ROUTE REPLY, εάν η αποστολή ενός ROUTE REPLY γίνει μέσω της διαδρομής του ROUTE REQUEST τότε όταν αυτό φτάσει στον εγωιστικό κόμβο θα απορριφθεί. Επειδή θέλουμε η αποστολή των Απαντήσεων στις Αιτήσεις Δρομολόγησης να είναι ασφαλής, η επιλογή του μονοπατιού που θα ακολουθήσουν επιλέγεται από το Σύστημα Διαχείρισης Διαδρομών (Path Manager).

4.1.1.3. Εισερχόμενο πακέτο που ανιχνεύθηκε από το κανάλι.

Τα πακέτα αυτά έχουν πολύ σημαντικό ρόλο για την υλοποίησή μας, γι΄ αυτό είναι σκόπιμο να αναφέρουμε πως η λειτουργία λήψης χωρίς περιορισμούς στο σύστημά μας είναι πάντα ενεργοποιημένη. Κάθε φορά που ένα πακέτο ανιχνεύεται από το κανάλι, το DSR το στέλνει στην μονάδα Monitor του μηχανισμού CONFIDANT. Στη συνέχεια, η μονάδα αυτή ελέγχει εάν το πακέτο είναι ίδιο με κάποιο από τα πακέτα των εξερχόμενων πακέτων στη μνήμη της, αυτό γίνεται με τη βοήθεια των αναγνωριστικών του πακέτου που αναφέρθηκαν παραπάνω. Εάν το πακέτο που ανιχνεύτηκε ταιριάζει με κάποιο στη μνήμη της μονάδας, τότε καλείται το Σύστημα Φήμης (Reputation System) και ενημερώνεται ο πίνακας που διατηρεί τις βαθμολογίες κατάλληλα. Στη συνέχεια, εάν η βαθμολογία του κόμβου του οποίου η βαθμολογία ενημερώθηκε περάσει ένα όριο ανοχής, τότε ειδοποιείται από το Σύστημα Φήμης το Σύστημα Διαχείρισης Διαδρομών έτσι ώστε να διαγράψει όλα τα μονοπάτια που περιέχουν αυτόν τον κόμβο από την Λίστα Διαδρομών του εκάστοτε κόμβου.

4.1.1.4. Πακέτο ALARM

Τα πακέτα ALARM στέλνονται μέσω του μηχανισμού CONFIDANT κάθε φορά που βαθμολογείται ένας κόμβος ως κακόφημος. Όταν ένας κόμβος λαμβάνει ένα πακέτο ALARM μέσο του Monitor, το στέλνει στο σύστημα Διαχείρισης Αξιοπιστίας (Trust Manager). Το Σύστημα Διαχείρισης Αξιοπιστίας αναλαμβάνει να αποφασίσει αν το μήνυμα ALARM περιέχει αξιόπιστες πληροφορίες ή όχι. Αρχικά, ελέγχει αν ο αποστολέας του μηνύματος θεωρείται αξιόπιστος ελέγχοντας τον Πίνακα Αξιοπιστίας που διατηρεί. Αν ο αποστολέας είναι αξιόπιστος, τότε στέλνει τις πληροφορίες του πακέτου στο Σύστημα Φήμης, έτσι ώστε να ενημερωθεί ο πίνακας Φήμης του κόμβου. Αν πάλι ο αποστολέας δεν θεωρείται αξιόπιστος, τότε ενημερώνει τον πίνακα αξιοπιστίας του ανάλογα.

Σε αυτό το σημείο, επιβάλλεται να αναφέρουμε το πώς το Σύστημα Αξιοπιστίας ενημερώνει τον πίνακα Αξιοπιστίας που διατηρεί. Κάθε φορά που φτάνει ένα μήνυμα ALARM, το σύστημα ελέγχει εάν ο κόμβος αποστολέας

βρίσκεται ήδη στον πίνακα αξιοπιστίας του και έχει επωνυμία, εάν ναι, τότε ακολουθείτε η παραπάνω διαδικασία που περιγράψαμε, εάν πάλι δεν υπάρχει, ο Trust Manager καλείται να εκτελέσει παραπάνω ελέγχους για να μπορέσει να πάρει την απόφασή του.
Οι έλεγχοι που πρέπει να εκτελέσει αφορούν τις πληροφορίες που περιέχει το μήνυμα ALARM. Αρχικά καλεί το Σύστημα Φήμης να διασταυρώσει εάν οι πληροφορίες που περιέχει μέσα το μήνυμα συμπίπτουν με τον Πίνακα Φήμης. Εάν δεν υπάρχουν σημαντικές διαφορές στις βαθμολογίες των κόμβων, τότε το Σύστημα Διαχείρισης Αξιοπιστίας προσθέτει στον πίνακα Αξιοπιστίας του τον κόμβο αποστολέα με θετική βαθμολογία. Εάν πάλι παρατηρηθούν σημαντικές διαφορές (για παράδειγμα ένας κόμβος που βαθμολογείται σύμφωνα με τις προσωπικές παρατηρήσεις του κόμβου "καλός", στο μήνυμα βαθμολογείται ως "κακός"), το Σύστημα Διαχείρισης Αξιοπιστίας θα προσθέσει τον αποστολέα του μηνύματος στον Πίνακα Αξιοπιστίας του με αρνητική βαθμολογία. Θέλοντας να δώσουμε μία δεύτερη ευκαιρία στους κόμβους όσο αφορά την αξιοπιστία τους, ένας κόμβος παίρνει κάθε φορά ή αρνητική ή θετική βαθμολογία μέχρι να επονομαστεί ως "αξιόπιστος" ή "αναξιόπιστος". Η αξιοπιστία ενός κόμβου, ορίζεται κάθε φορά που αυτός ξεπεράσει κάποιες προκαθορισμένες τιμές "κατώφλια" που έχουν ορισθεί.

4.1.2. Βαθμολογίες των κόμβων και χαρακτηρισμοί

Σε αυτή την υποενότητα θα αναφερθούμε στους πίνακες που διατηρούνται οι βαθμολογίες των κόμβων, στους χαρακτηρισμούς που χρησιμοποιεί το CONFIDANT για την λειτουργία του, και στο πώς ενημερώνεται η βαθμολογία.

4.1.2.1. Οι πίνακες Αξιοπιστίας, Φήμης και Εμπειριών

Αναφερθήκαμε παραπάνω για πίνακες Αξιοπιστίας και πίνακες Φήμης. Ο μηχανισμός CONFIDANT για την ακρίβεια, χρησιμοποιεί τριών ειδών πίνακες βαθμολογιών για τον χαρακτηρισμό των υπόλοιπων κόμβων του δικτύου.

- Ο πίνακας Αξιοπιστίας ή T_{ij}, είναι εκείνος που χαρακτηρίζει την αξιοπιστία που έχει ένας κόμβος i για έναν άλλο κόμβο j του δικτύου. Για την ενημέρωση και χρήση του πίνακα αυτού ευθύνεται το σύστημα Διαχείρισης Αξιοπιστίας (Trust Manager)
- Ο πίνακας Φήμης ή R_{ij}, είναι εκείνος που χαρακτηρίζει την άποψη ενός κόμβου i για έναν άλλο κόμβο j του δικτύου. Για την ενημέρωση και

χρήση του πίνακα αυτού ευθύνεται το σύστημα Φήμης (Reputation System)

- Ο πίνακας Εμπειριών ή F_{ij}, είναι εκείνος που περιέχει τις προσωπικές εμπειρίες που είχε ένας κόμβος i στην επικοινωνία του με έναν κόμβο j του δικτύου (First hand information). Για την ενημέρωση και χρήση του πίνακα αυτού ευθύνεται επίσης το σύστημα Φήμης.

Αν και οι παραπάνω πίνακες φαινομενικά μπορούν να χαρακτηρισθούν ως παρόμοιοι, για την ακρίβεια είναι πολύ διαφορετικοί. Πιο συγκεκριμένα, ο πίνακας αξιοπιστίας περιέχει πληροφορίες για το πόσο ένας κόμβος μπορεί να εμπιστεύεται τις πληροφορίες που παίρνει από κάποιον άλλο κόμβο. Αντίθετα, ο πίνακας φήμης περιέχει πληροφορίες για την συμπεριφορά των υπόλοιπων κόμβων στο δίκτυο. Ο πίνακας Εμπειριών, από την άλλη, διαφέρει από τα παραπάνω είδη πινάκων στο ότι συντηρεί πληροφορίες μόνο για τις προσωπικές εμπειρίες που είχαν οι κόμβοι στην επικοινωνία τους με τους υπόλοιπους κόμβους του δικτύου και όχι τους τελικούς τους χαρακτηρισμούς.

4.1.2.2. Οι χαρακτηρισμοί Αξιοπιστίας, Φήμης και Εμπειριών

Το σύστημα CONFIDANT ορίζει κάποιους χαρακτηρισμούς όσο αφορά τους τρείς τύπους αναγνώρισης της συμπεριφοράς.

- Ένας κόμβος μπορεί να χαρακτηριστεί σύμφωνα με την Αξιοπιστία του ως: “αξιόπιστος” ή “αναξιόπιστος” .
- Ένας κόμβος μπορεί να χαρακτηριστεί σύμφωνα με την Φήμη του ως: “καλός” ή “κακός”
- Ένας κόμβος μπορεί να χαρακτηρίσει την Εμπειρία του με κάποιον άλλο κόμβο ως: “καλή” ή “κακή” .

Όπως είπαμε και παραπάνω για τους πίνακες, οι χαρακτηρισμοί αυτοί, ενώ φαίνονται παρόμοιοι, είναι τελείως διαφορετικοί. Για παράδειγμα, ένας κόμβος μπορεί να χαρακτηρίζεται ως “αναξιόπιστος” ενώ ταυτόχρονα είναι “καλός” και οι εμπειρίες μαζί του είναι “καλές”. Ένα τέτοιο παράδειγμα μπορεί να είναι απόρροια των εξής γεγονότων:

Έστω ένας κόμβος i λαμβάνει από έναν κόμβο j ένα μήνυμα που χαρακτηρίζει έναν κόμβο k ως “καλό”. Ο κόμβος j για τον κόμβο i θεωρείτε “αξιόπιστος” και ο κόμβος i από προηγούμενες εμπειρίες με τον κόμβο k έχει χαρακτηρίσει την Εμπειρία μαζί του ως “καλή”, άρα ενημερώνει τον πίνακα

Φήμης του για την πληροφορία που μόλις έλαβε χαρακτηρίζοντας τον κόμβο k ως "καλό". Ωστόσο, ο κόμβος i μπορεί να λάβει από τον κόμβο k ένα μήνυμα που χαρακτηρίζει έναν κόμβο z ως "καλό", ενώ ο κόμβος i από εμπειρίες του έχει χαρακτηρίσει τον κόμβο z ως "κακό", τότε ο κόμβος k για τον κόμβο i θα χαρακτηρισθεί στον πίνακα Αξιοπιστίας του ως "αναξιόπιστος". Έτσι τελικά οι χαρακτηρισμοί στους αντίστοιχους πίνακες του κόμβου i για τον k θα είναι:

- $T_{i,k}$ = "αναξιόπιστος"
- $R_{i,k}$ = "καλός"
- $F_{i,k}$ = "καλή"

4.1.2.3. Η ενημέρωση της Βαθμολογίας

Η βαθμολογία στο σύστημά μας παίζει πολύ σημαντικό ρόλο καθώς από το πώς αυτή ενημερώνεται κρίνεται η συμπεριφορά και η αποτελεσματικότητα του Μηχανισμού. Η βαθμολογία θα πρέπει να είναι δίκαιη, δηλαδή ένας κόμβος δεν μπορεί να καθορίζεται οριστικά ως "κακός" διότι μία προώθηση που έκανε δεν ήταν αποτελεσματική. Μη ξεχνάμε ότι υπάρχουν και άλλοι λόγοι που ένα πακέτο μπορεί να απορριφθεί ή να χαθεί, όπως για παράδειγμα η ανεπιτυχής μετάδοση λόγω κίνησης στο κανάλι. Γι' αυτό επιλέγουμε να χαρακτηρίζουμε έναν κόμβο σύμφωνα με έναν αριθμό γεγονότων και όχι από μόνο μία παρατήρηση. Για να γίνει αυτό εφικτό χρησιμοποιούμε τον παρακάτω τρόπο.

Θεωρούμε λοιπόν, πως η βαθμολογία κάθε κόμβου στηρίζεται σε δύο βασικές μεταβλητές (α, β). Όταν δεν υπάρχει καμία παρατήρηση οι μεταβλητές αυτές έχουν τις τιμές $\alpha = 1, \beta = 1$. Όταν ένας κόμβος παρατηρήσει κάτι για κάποιον άλλο κόμβο, ενημερώνει τις μεταβλητές σύμφωνα με την παρακάτω εξίσωση:

$$\begin{cases} \alpha = f\alpha + n \\ \beta = f\beta + (1 - n) \end{cases}$$

Όπου n, είναι η παρατήρηση (note) του κόμβου, όταν $n = 1$ η παρατήρηση είναι αρνητική, όταν $n = 0$ η παρατήρηση είναι θετική. Όπου f, είναι μία σταθερά που λειτουργεί ως ανασταλτικός παράγοντας (fading factor) και θα εξηγηθεί παρακάτω.

Η παραπάνω εξίσωση χρησιμοποιείται και από τους τρείς πίνακες Βαθμολογιών που χρησιμοποιεί το CONFIDANT. Το μόνο που αλλάζει στον υπολογισμό των (α, β) στον καθένα από αυτούς, είναι ο ανασταλτικός παράγοντας f ο οποίος ανάλογα με τον πίνακα παίρνει και διαφορετική τιμή. Έχουμε λοιπόν τους εξής ανασταλτικούς παράγοντες για τον υπολογισμό των (α, β):

- Για την Αξιολόγηση $f = f_t$
- Για την Φήμη $f = f_r$
- Για τις Εμπειρίες $f = f_f$

Για τον χαρακτηρισμό των κόμβων χρησιμοποιούμε την παρακάτω εξίσωση:

$$E(\alpha, \beta) = \frac{\alpha}{\alpha + \beta}$$

Όταν το αποτέλεσμα της παραπάνω εξίσωσης υπερβεί μία προκαθορισμένη τιμή (που ονομάζουμε “κατώφλι”) τότε ο χαρακτηρισμός του αλλάζει. Πιο συγκεκριμένα για κάθε τύπο χαρακτηρισμού ισχύουν τα παρακάτω:

- Για την Αξιολόγηση:

$$T_{i,j} = \begin{cases} \text{"αξιόπιστος"} & ,\text{όταν } E(\alpha, \beta) < t \\ \text{"αναξιόπιστος"} & ,\text{όταν } E(\alpha, \beta) \geq t \end{cases}$$

- Για την Φήμη:

$$R_{i,j} = \begin{cases} \text{"καλός"} & ,\text{όταν } E(\alpha, \beta) < r \\ \text{"κακός"} & ,\text{όταν } E(\alpha, \beta) \geq r \end{cases}$$

- Για τις Εμπειρίες:

$$F_{i,j} = \begin{cases} \text{"καλή"} & ,\text{όταν } E(\alpha, \beta) < s \\ \text{"κακή"} & ,\text{όταν } E(\alpha, \beta) \geq s \end{cases}$$

Όπου t, r και s οι προκαθορισμένες τιμές ¨κατώφλια¨ για την Αξιολόγηση, την Φήμη και τις Εμπειρίες αντίστοιχα.

Τέλος, πρέπει να λάβουμε υπόψη μας, πως ένας κόμβος μπορεί να θέλει να σταματήσει να συμπεριφέρεται άσχημα, ή ένας κόμβος λόγω τεχνικών προβλημάτων, να απέκτησε κακή φήμη. Για αυτές τις περιπτώσεις πρέπει να χρησιμοποιούμε έναν ανασταλτικό παράγοντα που θα μετριάζει τις

βαθμολογίες των κόμβων ανά κάποια χρονικά διαστήματα. Έτσι, οι τιμές των (α, β) θα μετριάζονται κάθε τόσο σύμφωνα με την παρακάτω εξίσωση:

$$\begin{cases} \alpha = f\alpha \\ \beta = f\beta + 1 \end{cases}$$

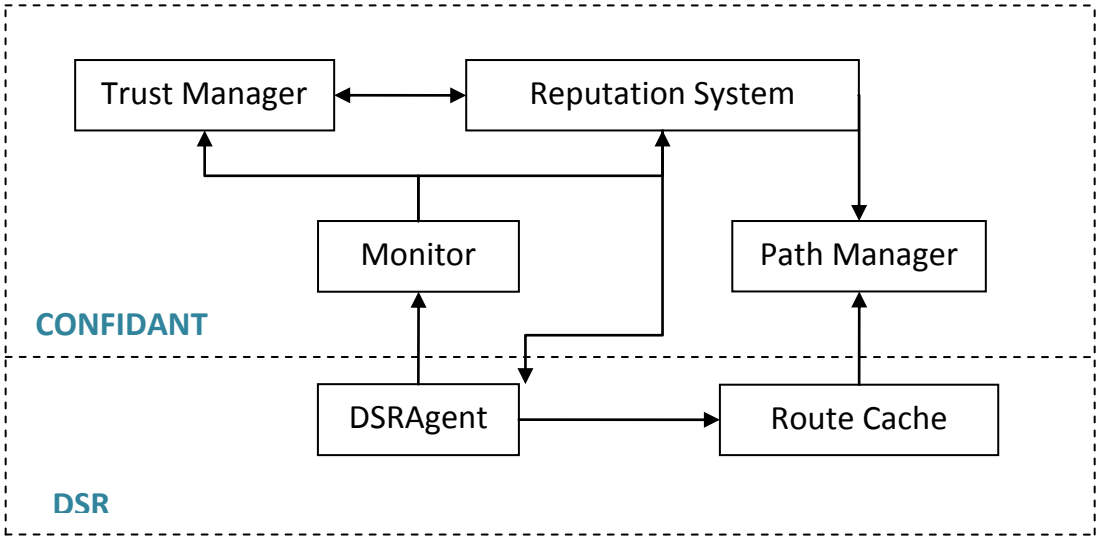
Όπου f, φυσικά, ο ανασταλτικός παράγοντας κάθε περίπτωσης.

4.2. Βασικές Τάξεις και Συναρτήσεις

Σε αυτήν την ενότητα θα παρουσιασθεί η προσπάθεια μεταφοράς όλων των παραπάνω παραδοχών σε κώδικα. Οι βασικές τάξεις και συναρτήσεις που χρησιμοποιήθηκαν θα αναλυθούν έτσι ώστε ο αναγνώστης να είναι σε θέση να κατανοήσει την λογική που χρησιμοποιήθηκε για την υλοποίηση του μηχανισμού.

4.2.1. Οι Βασικές Τάξεις

Όπως αναφέραμε και παραπάνω, ο μηχανισμός CONFIDANT εξ' ορισμού χρησιμοποιεί πέντε βασικές μονάδες για την λειτουργία του. Αυτές είναι το Σύστημα Παρακολούθησης (Monitor), το Σύστημα Φήμης (Reputation System), το Σύστημα Διαχείρισης Διαδρομών (Path Manager), το Σύστημα Διαχείρισης Αξιοπιστίας (Trust Manager) και το Πρωτόκολλο Δρομολόγησης DSR. Η αρχιτεκτονική του κώδικά μας φαίνεται στο παρακάτω σχήμα:



Αρχιτεκτονική του CONFIDANT DSR

Η κάθε μία μονάδα από τις παραπάνω αποτελεί μία σειρά συναρτήσεων στην C++, ορισμένη στην τάξη του DSRAgent, με εξαίρεση την μονάδα Path Manager η οποία υλοποιείται στην δική της τάξη. Έτσι όταν αναφερόμαστε από εδώ και στο εξής σε κάποια από τις μονάδες, εννοούμε πως είναι η αντίστοιχη σειρά συναρτήσεων στο πρόγραμμά μας.

Σύμφωνα λοιπόν με την παραπάνω αρχιτεκτονική γίνεται και η σχεδίαση του κώδικα. Αναλυτικά η κάθε μονάδα του μηχανισμού CONFIDANT αναφέρεται παρακάτω.

4.2.1.1. Η μονάδα Monitor

Η μονάδα Monitor είναι υπεύθυνη για την αναγνώριση της "κακής" συμπεριφοράς των κόμβων και την αναφορά της στο σύστημα Φήμης. Επίσης είναι υπεύθυνη να στέλνει τα μηνύματα ALARM που δέχεται στο σύστημα Αξιοπιστίας για αξιολόγηση. Οι βασικές συναρτήσεις της φαίνονται στον παρακάτω πίνακα:

Monitor
registerSentPacket() handleTap() packTimeoutHandler()

Κάθε φορά που γίνεται αποστολή ενός οποιουδήποτε μηνύματος από ένα κόμβο, καλείται η συνάρτηση *registerSentPacket()* η οποία είναι υπεύθυνη να αποθηκεύσει τα απαραίτητα αναγνωριστικά του πακέτου αυτού σε έναν πίνακα που ονομάζεται Pack Table. Ο πίνακας αυτός περιέχει το *uid* του πακέτου (κάθε πακέτο στο NS-2 χαρακτηρίζεται από ένα συγκεκριμένο αναγνωριστικό που ονομάζεται *uid*) και την χρονική στιγμή που αυτό στάλθηκε.

Κάθε φορά που ο κόμβος ανιχνεύει ένα πακέτο από το κανάλι επικοινωνίας, η συνάρτηση *handleTap()* καλείται (τα πακέτα αυτά στο εξής θα τα ονομάζουμε Tap για συντομία). Η συνάρτηση αυτή είναι υπεύθυνη να ελέγξει εάν το Tap πακέτο που λήφθηκε ταιριάζει με κάποιο από τα πακέτα που βρίσκονται αποθηκευμένα στον πίνακα Pack Table. Συγκρίνοντας το *uid* του Tap πακέτου και τα *uid* που βρίσκονται στον πίνακα, αναγνωρίζει εάν όντως ο επόμενος κόμβος προώθησε το πακέτο. Εάν ανιχνεύσει ότι ο επόμενος κόμβος έστειλε το πακέτο, στέλνει την πληροφορία στο σύστημα Φήμης για

να ενημερώσει εκείνο με τη σειρά του τον πίνακα Φήμης. Εάν το *uid* του Tap πακέτου δεν ταιριάξει με κανένα από τα *uid* του πίνακα Pack Table, τότε αγνοείται.

Τέλος, η συνάρτηση *packTimeoutHandler()* είναι υπεύθυνη να ελέγχει ανά τακτά χρονικά διαστήματα, εάν κάποιο από τα πακέτα που είναι αποθηκευμένα στον Pack Table έχει λήξει. Ο τρόπος που το σύστημά μας αναγνωρίζει ότι ο επόμενος κόμβος δεν έχει προωθήσει το πακέτο είναι να κρατάει τον χρόνο από τότε που στάλθηκε το πακέτο μέχρι έναν προκαθορισμένο χρόνο λήξης. Εάν πριν τον χρόνο λήξης η μονάδα Monitor δεν λάβει ένα Tap πακέτο του οποίου το *uid* να ταιριάζει με εκείνο στον Pack Table, τότε η Monitor υποθέτει πως ο επόμενος κόμβος δεν προώθησε το πακέτο και έτσι ενημερώνει το σύστημα Φήμης.

4.2.1.2. Η μονάδα Reputation System

Η μονάδα Reputation System είναι υπεύθυνη για την ενημέρωση των πινάκων Φήμης και Εμπειριών. Είναι εκείνη που καθορίζει έναν κόμβο ως “καλό” ή “κακό” και τις εμπειρίες μαζί του “καλές” ή “κακές”. Επίσης είναι υπεύθυνη να ενημερώνει την μονάδα Path Manager κατάλληλα έτσι ώστε εκείνη με τη σειρά της να διαγράφει τα μονοπάτια με τους “κακούς” κόμβους στη Λίστα Δρομολογίων της. Οι βασικές συναρτήσεις της μονάδας Reputation System φαίνονται στον παρακάτω πίνακα:

Reputation System
handleFirsthandInfo()
handleSecondhandInfo()
isMisbehaviorNode()
deviationTest()
updateFirsthandRating()
updateReputationRating()
publishTimeoutHandler()
inactivityTimeoutHandler()

Η συνάρτηση *handleFirsthandInfo()* είναι υπεύθυνη για τον χειρισμό των δεδομένων που βασίζονται στις προσωπικές εμπειρίες του κόμβου. Όταν η Monitor εντοπίσει πως ένας κόμβος δεν προώθησε το επόμενο πακέτο, οι πληροφορίες για το πακέτο αυτό περνάν στη συνάρτηση αυτή. Τότε εκείνη με τη σειρά της αναγνωρίζει τον κόμβο που δεν προώθησε, υπολογίζει τα *(α, β)* σύμφωνα με τους κανόνες βαθμολογίας που αναλύσαμε στην παράγραφο 4.1.2.3 και καλεί την *updateFirsthandRating()* για να ενημερώσει τον πίνακα

Εμπειριών. Στη συνέχεια, επαναϋπολογίζει τα *(α, β)* σύμφωνα πάλι με τους κανόνες βαθμολόγησης που αναλύσαμε παραπάνω και καλεί την συνάρτηση *updateReputationRating()* η οποία με τη σειρά της είναι υπεύθυνη να ενημερώσει τον πίνακα Φήμης με τις αντίστοιχες βαθμολογίες. Αφού τελειώσει και αυτή η διαδικασία, καλεί την συνάρτηση *isMisbehaviorNode()* για να ελέγξει αν η βαθμολογία του κόμβου που ενημερώθηκε έχει ξεπεράσει την τιμή "κατώφλι", εάν η απάντηση είναι θετική, τότε ενημερώνεται η μονάδα Path Manager.

Η συνάρτηση *handleSecondhandInfo()* είναι υπεύθυνη για τον χειρισμό των πληροφοριών που λαμβάνονται μέσο των μηνυμάτων ALARM. Η συνάρτηση καλείται από τη μονάδα Trust Manager. Αρχικά ελέγχει μέσω της συνάρτησης *deviationTest()* αν οι πληροφορίες βαθμολόγησης που περιέχονται στο μήνυμα έχουν μεγάλες αποκλίσεις από τις υπάρχουσες βαθμολογίες του πίνακα Εμπειριών. Εάν οι αποκλίσεις είναι μεγάλες, τότε ενημερώνει την Trust Manager ότι οι πληροφορίες δεν είναι αξιόπιστες. Εάν οι πληροφορίες θεωρηθούν αξιόπιστες, τότε ο πίνακας Φήμης ενημερώνεται καλώντας την *handleReputationRating().* Στη συνέχεια, όπως και στην συνάρτηση *handleFirsthandInfo()*, καλείται η συνάρτηση *isMisbehaviorNode()* για να υπολογίσει αν υπάρχουν καινούργιοι "κακοί" κόμβοι. Εάν ναι, τότε και αυτή ενημερώνει την μονάδα Path Manager.

Οι συναρτήσεις *publishTimeoutHandler()* και *inactivityTimeoutHandler()* λειτουργούν ως χρονόμετρα για το πρόγραμμα. Η *publishTimeoutHandler()* ελέγχει το πότε πρέπει ο κόμβος να στείλει μηνύματα ALARM στους υπόλοιπους κόμβους. Το κάθε πότε καθορίζεται από έναν προκαθορισμένο χρόνο λήξης. Κάθε φορά που ο χρόνος λήγει, η συνάρτηση στέλνει το πακέτο ALARM στον DSR για αποστολή. Η *inactivityTimeoutHandler()* ελέγχει το κάθε πότε οι βαθμολογίες των πινάκων πρέπει να μετριάζονται (παράγραφος 4.1.2.3.) και πράττει ανάλογα.

4.2.1.3. Η μονάδα Trust Manager

Η μονάδα Trust Manager είναι υπεύθυνη να ελέγχει αν ένας κόμβος είναι αξιόπιστος ή όχι. Είναι εκείνη που έχει υπ' ευθύνη της τον πίνακα Αξιοπιστίας. Οι βασικές συναρτήσεις που την απαρτίζουν δίνονται στο παρακάτω πίνακα:

Trust Manager
updateTrustRaiting()
isNodeTrustworthy()

Η συνάρτηση *updateTrustRaiting()* είναι υπεύθυνη να υπολογίζει τις βαθμολογίες Αξιοπιστίας και να ενημερώνει τον πίνακα Αξιοπιστίας ανάλογα. Στη συνέχεια καλή την συνάρτηση *isNodeTrustworthy()* η οποία ελέγχει αν κάποιος από τους κόμβους έχει υπερβεί την τιμή "κατώφλι" της αξιοπιστίας. Στη συνέχεια στέλνει το αποτέλεσμα της τελευταίας συνάρτησης στην μονάδα Reputation System για να κρατήσει ή να παρακάμψει τις πληροφορίες που πήρε μέσο ALARM μηνυμάτων.

4.2.1.4. Η μονάδα Path Manager

Η μονάδα Path Manager είναι υπεύθυνη για την ενημέρωση και χρήση της Route Cache του DSR σύμφωνα με τις μεθόδους του CONFIDANT. Πιο συγκεκριμένα είναι υπεύθυνη να επιλέγει τα κατάλληλα μονοπάτια για τα εξερχόμενα πακέτα του κόμβου. Οι βασικές συναρτήσεις που χρησιμοποιεί είναι οι ακόλουθες:

Path Manager
addMisbehavedNode() removeMisbehavedNode() isNodeSafe() isPathSafe()

Στη μονάδα αυτή ανατρέχει ο DSR κάθε φορά που χρειάζεται μονοπάτι για έναν κόμβο. Είναι σημαντικό να αναφέρουμε ότι στην υλοποίησή μας δεν θα διαγράφουμε μονοπάτια που έχουν κακούς κόμβους, αυτό γίνεται γιατί ο χαρακτηρισμός για έναν κόμβο μπορεί να αλλάξει και θέλουμε να δώσουμε στους κόμβους μία δεύτερη ευκαιρία. Γι' αυτό η συνάρτηση *addMisbehavedNode()* είναι υπεύθυνη να μαρκάρει και να εισάγει τα μονοπάτια που περιέχουν κακούς κόμβους στην λίστα Route Cache. Εάν ένας κόμβος σταματήσει να θεωρείται εγωιστικός τότε η συνάρτηση *removeMisbehavedNode()* αναλαμβάνει, επαναφέροντας μονοπάτια που έχουν μαρκαριστεί ως μονοπάτια προς αποφυγή. Η συνάρτηση *isPathSafe()* είναι υπεύθυνη να χαρακτηρίσει εάν ένα μονοπάτι που έχει επιλεγεί είναι ασφαλές ή όχι. Η συνάρτηση *isNodeSafe()* είναι υπεύθυνη να χαρακτηρίσει εάν ένα εισερχόμενο μήνυμα έχει σταλεί από έναν "κακό" κόμβο.

4.2.2. Η σύνδεση με τον DSR

Είδαμε αναλυτικά τις τάξεις και τις συναρτήσεις του DSR πρωτοκόλλου στο κεφάλαιο 3 (παράγραφος 3.2), σε αυτή την παράγραφο θα αναφέρουμε μόνο ποιές συναρτήσεις και τάξεις του DSR αλληλεπιδρούν με τις συναρτήσεις του CONFIDANT και πώς.

- Η καταγραφή των εξερχόμενων πακέτων
  Στην καταγραφή των εξερχόμενων πακέτων αλληλεπιδρούν δύο τάξεις, η τάξη DSRAgent του DSR και η Monitor του Confidant. Κάθε φορά που γίνεται η αποστολή ενός πακέτου μέσω της συνάρτησης *sendOutPacketWithRoute()* της DSRAgent καλείται η συνάρτηση *registerSentPacket()* της Monitor.
  Monitor-> registerSentPacket(packet)
- Η λήψη Tap πακέτων
  Η αποστολή των πακέτων τύπου Tap (πακέτα που ακούστηκαν από το κανάλι) γίνεται μέσω της μονάδας DSRAgent στην μονάδα Monitor του Confidant. Από την συνάρτηση tap() της DSRAgent καλούμε την συνάρτηση handleTap() της Monitor.
  Monitor-> handleTap(packet)
- Η λήψη των πακέτων ALARM
  Για την επεξεργασία των πακέτων ALARM συνεργάζονται η μονάδα DSRAgent και η μονάδα Reputation System. Όταν ο DSRAgent λάβει ένα τέτοιο πακέτο μέσω της συνάρτησης *recv()* στέλνει το πακέτο αυτό στην μονάδα Reputation System μέσω της συνάρτησης *handleSecondhandInfo().*
- Η αποστολή πακέτων ALARM
  Όταν η μονάδα Reputation System θέλει να στείλει ένα πακέτο τύπου ALARM στέλνει το πακέτο αυτό στην DSRAgent. Εκείνη αναλαμβάνει να τελειοποιήσει το πακέτο και να το στείλει.
- Η εύρεση μονοπατιού
  Όταν ο DSRAgent ζητάει μονοπάτι για ένα πακέτο που θέλει να στείλει, τότε αυτός καλεί την μονάδα Path Manager μέσω της συνάρτησης *findRoute()* η οποία αλληλεπιδρά με την μονάδα Route Cache του DSR για να βρει το κατάλληλο μονοπάτι και να το επιστρέψει. Εάν δεν βρει κάποιο μονοπάτι που πληροί τις προϋποθέσεις, επιστρατεύει την διαδικασία Εύρεσης Διαδρομών του DSR.

4.2.3. Η ανάλυση της Συμπεριφοράς

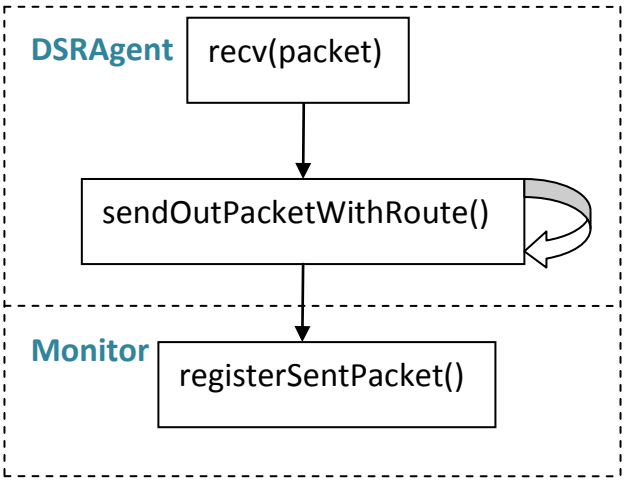
Στην παράγραφο αυτή, θα αναλυθεί με τη βοήθεια των τάξεων και των συναρτήσεών τους η συμπεριφορά του προγράμματος σε κάθε μία από τις παρακάτω περιπτώσεις:

1. Αποστολή Πακέτων
2. Λήψη πακέτων τύπου Tap
3. Αναγνώριση ανάρμοστης συμπεριφοράς
4. Αποστολή πακέτων τύπου ALARM
5. Λήψη πακέτων τύπου ALARM
6. Τιμωρία

Η ανάλυσή τους θα γίνει μέσω διαγραμμάτων για να βοηθήσει στην κατανόησή τους.

4.2.3.1. Αποστολή Πακέτων

Το παρακάτω διάγραμμα απεικονίζει την συμπεριφορά των βασικών Τάξεων και Συναρτήσεων στην περίπτωση αποστολής πακέτων δεδομένων, ROUTE REQUEST, ROUTE REPLY και ROUTE ERROR από το σύστημα. Και στις τέσσερις αυτές περιπτώσεις, ακολουθείτε η ίδια διαδικασία όσο αναφορά την συμπεριφορά του CONFIDANT.

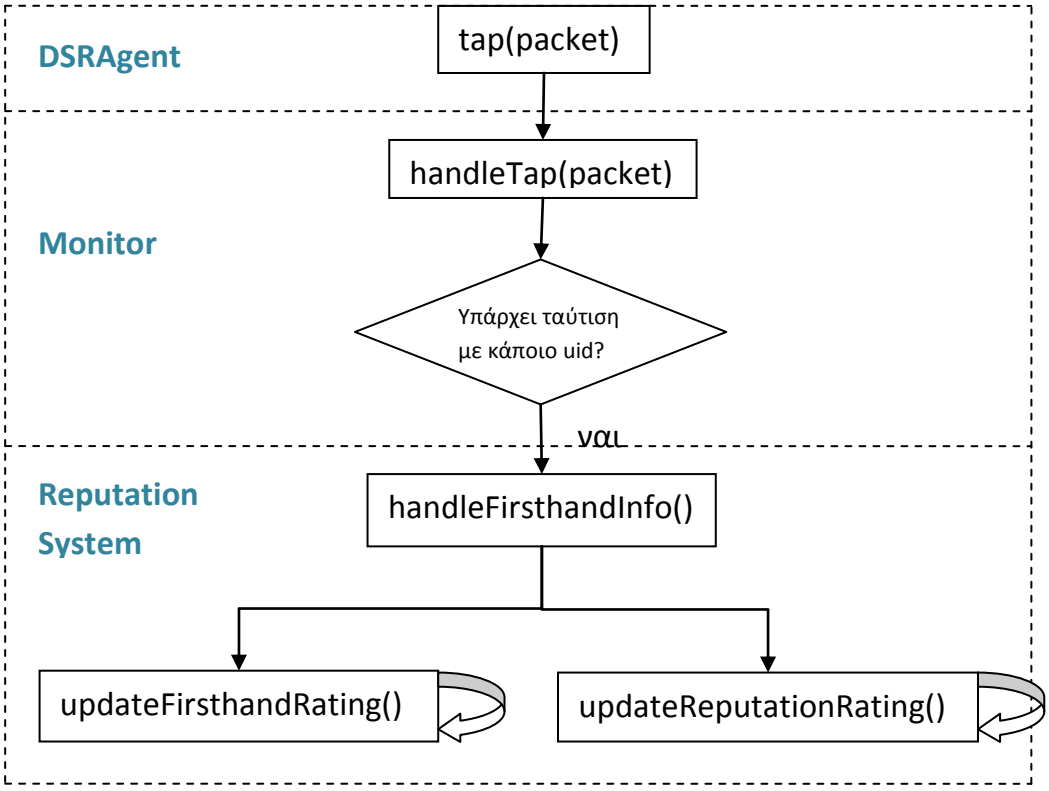


Διάγραμμα Αποστολής Πακέτων

Όταν ένας κόμβος στέλνει ένα οποιοδήποτε πακέτο, το στέλνει στην μονάδα Monitor για καταγραφή στο Pack Table που διατηρεί έτσι ώστε στη συνέχεια να διαπιστώσει ότι ο επόμενος κόμβος προώθησε το πακέτο.

4.2.3.2. *Λήψη πακέτων τύπου Tap*

Το παρακάτω διάγραμμα απεικονίζει την συμπεριφορά του προγράμματος όταν ένα πακέτο ανιχνεύεται από το κανάλι επικοινωνίας (πακέτο τύπου Tap)

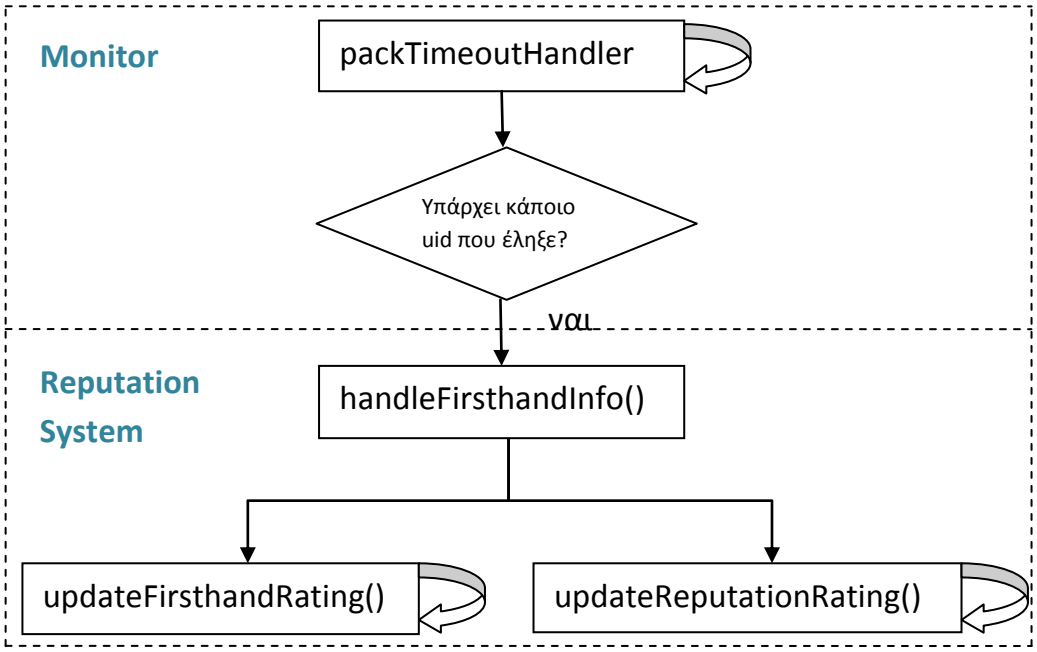


Διάγραμμα Λήψης πακέτων τύπου Tap

Όταν γίνεται ανίχνευση ενός Tap πακέτου στο κανάλι, ο DSRAgent το στέλνει στον Monitor για να εντοπίσει εάν ταιριάζει με κάποιο από τα πακέτα που περιμένουν στον Pack Table για επιβεβαίωση. Εάν ναι, τότε ενημερώνει το Reputation System για την ανίχνευση της καλής συμπεριφοράς. Το Reputation System στη συνέχεια ενημερώνει τις βαθμολογίες των πινάκων Φήμης και Εμπειριών.

4.2.3.3. Ανίχνευση Ανάρμοστης Συμπεριφοράς

Το παρακάτω διάγραμμα απεικονίζει τις ενέργειες εκείνες από τις οποίες το πρόγραμμά μας αναγνωρίζει πως κάποιος κόμβος δεν προώθησε τελικά το πακέτο που του στάλθηκε.

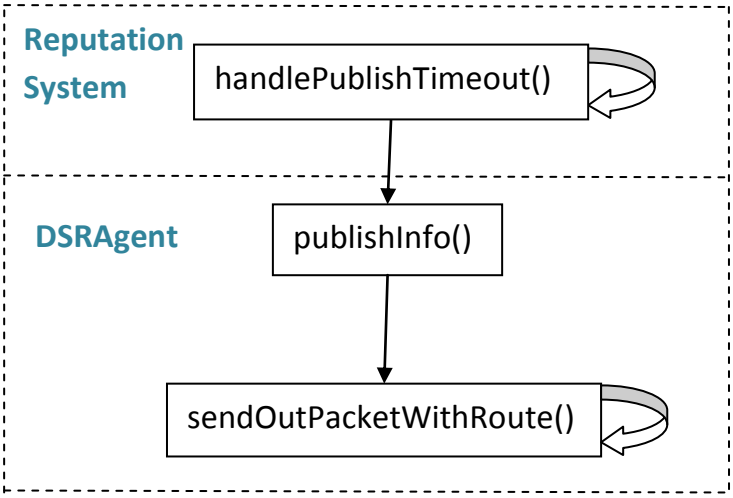


Διάγραμμα Ανίχνευσης Ανάρμοστης Συμπεριφοράς

Όταν ο χρόνος που βρίσκεται ένα πακέτο στον Pack Table λήγει χωρίς να έχει βρεθεί ένα πακέτο Tap που επιβεβαιώνει την προώθησή του, ο Monitor υποθέτει πως ο επόμενος κόμβος δεν το έχει προωθήσει. Γι αυτό ενημερώνει το Reputation System που εκείνο με τη σειρά του ενημερώνει τους πίνακες Φήμης και Εμπειριών.

4.2.3.4. Αποστολή πακέτων τύπου ALARM

Το παρακάτω διάγραμμα απεικονίζει την διαδικασία η οποία χρειάζεται για να γίνει αποστολή ενός μηνύματος τύπου ALARM.

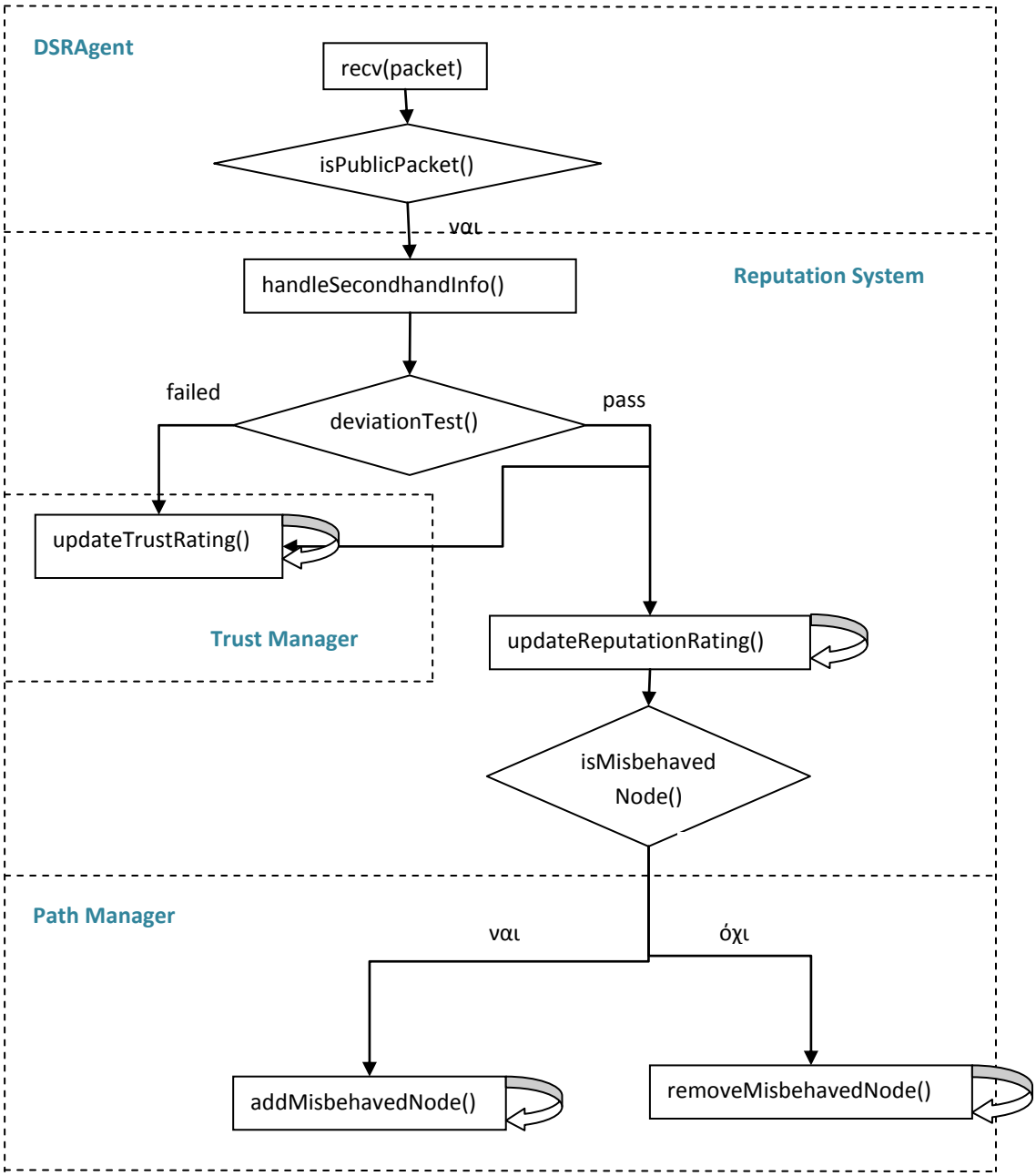


Διάγραμμα Αποστολής Πακέτων ALARM

Κάθε φορά που λήγει ο προκαθορισμένος χρόνος αναμονής για την αποστολή των πληροφοριών προσωπικών παρατηρήσεων, το Reputation System στέλνει στον DSRAgent το πακέτο με τις πληροφορίες που θέλει να στείλει.

4.2.3.5. Λήψη πακέτων τύπου ALARM

Το παρακάτω διάγραμμα απεικονίζει την διαδικασία που ακολουθεί το πρόγραμμά μας όταν λαμβάνει ένα πακέτο τύπου ALARM.

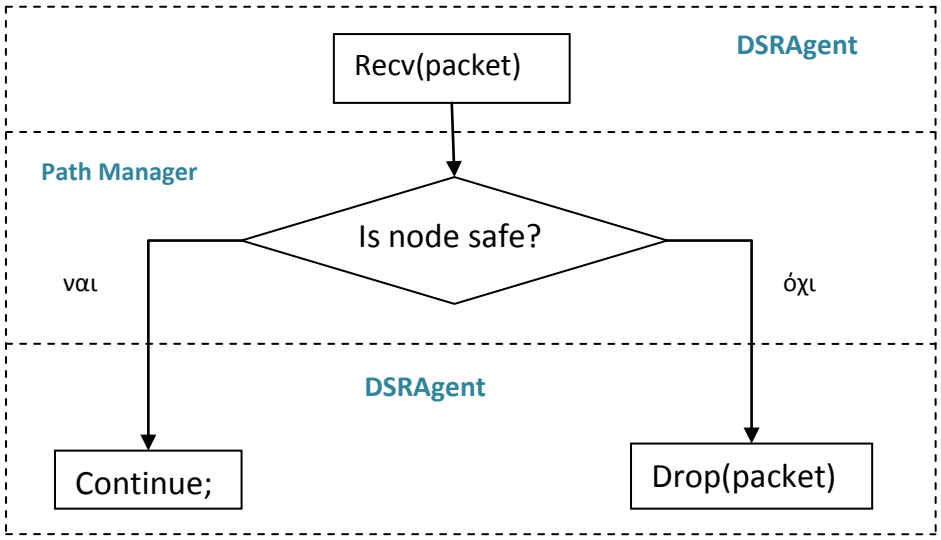


Διάγραμμα Λήψης πακέτου ALARM

Όταν ένα τέτοιο πακέτο φτάνει στον DSRAgent εκείνος το αναγνωρίζει και το στέλνει κατευθείαν στο Reputation System. Εκείνο με τη σειρά του διενεργεί τα απαραίτητα test έτσι ώστε να καταλάβει αν η πληροφορία του μηνύματος μπορεί να είναι αξιόπιστη ή όχι. Ανάλογα με το αποτέλεσμα του test ενημερώνει το Trust Manager. Αν το πακέτο περιέχει αξιόπιστη πληροφορία τότε ενημερώνει τον πίνακα Φήμης ανάλογα. Στη συνέχεια εξετάζεται αν κάποια από τις βαθμολογίες που ενημερώθηκαν χαρακτηρίζει κάποιον κόμβο ως "κακό". Εάν ναι τότε ενημερώνεται ο Path Manager.

4.2.3.6. Τιμωρία

Το παρακάτω διάγραμμα απεικονίζει το πώς αντιμετωπίζεται ένα οποιοδήποτε εισερχόμενο πακέτο που προέρχεται από έναν κακόφημο κόμβο.



Διάγραμμα της Τιμωρίας

Κάθε εισερχόμενο πακέτο στον DSRAgent εξετάζεται εάν προέρχεται από κακόφημο κόμβο μέσω του Path Manager. Εάν ναι, τότε το πακέτο αυτό απορρίπτεται.

Κεφάλαιο 5

Προσομοιώσεις και Αποτελέσματα

Το κεφάλαιο αυτό χωρίζεται σε δύο βασικά υποκεφάλαια, τις προσομοιώσεις και τα αποτελέσματα. Στο πρώτο υποκεφάλαιο θα αναλυθεί η κατασκευή του αρχείου προσομοίωσης, θα αναφερθούν οι παράμετροι των προσομοιώσεων και τέλος, θα αναφερθεί πως έγινε η προσομοίωση των εγωιστικών κόμβων. Στο δεύτερο υποκεφάλαιο, θα παρουσιασθούν τα αποτελέσματα των προσομοιώσεων, θα συγκριθούν με αποτελέσματα του απλού DSR και θα βγουν συμπεράσματα για το αν τελικά το CONFIDANT ενεργεί θετικά στο δίκτυο.

5.1. Προσομοιώσεις

Για τις προσομοιώσεις, όπως είπαμε και παραπάνω, χρησιμοποιήθηκε ο προσομοιωτής δικτύων NS-2.34. Το πρόγραμμα εγκαταστάθηκε σε περιβάλλον Ubuntu 10.04 με τη βοήθεια του προγράμματος VMware. Ο προσομοιωτής δικτύων NS-2 χρησιμοποιεί δύο βασικές γλώσσες για την εκτέλεσή του. Η μία και βασική γλώσσα προγραμματισμού είναι η C++ στην οποία και κατασκευάστηκαν οι μονάδες του CONFIDANT, η άλλη είναι η TCL η οποία χρησιμοποιείται για την κατασκευή σεναρίων κατάλληλων για την προσομοίωση δικτύων. Εφόσον περιγράψαμε παραπάνω πως κατασκευάστηκε το CONFIDANT ως μηχανισμός βασισμένος στον DSR, σε αυτό το υποκεφάλαιο θα περιγράψουμε πως έγιναν οι προσομοιώσεις με τη βοήθεια της TCL.

5.1.1. Κατασκευή του Σεναρίου Προσομοίωσης

Το σενάριο προσομοίωσης κατασκευάστηκε στην γλώσσα προγραμματισμού TCL που χρησιμοποιεί ο NS-2. Το αρχείο αυτό, περιέχει τις βασικές παραμέτρους τις προσομοίωσης. Αυτές μπορεί να είναι ο αριθμός των κόμβων που θα προσομοιωθούν, το πρωτόκολλο δρομολόγησης, η κίνηση και η θέση των κόμβων ή οι συνδέσεις μεταξύ τους.

Στην προσομοίωσή μας θα χρησιμοποιήσουμε το πρωτόκολλο δρομολόγησης DSR με και χωρίς την προσθήκη του CONFIDANT. Ο αριθμός των κόμβων επιλέγεται να είναι μεταξύ 20 και 50. Όλες οι προσομοιώσεις θα χρησιμοποιούν ασύρματες συνδέσεις και ως πρωτόκολλο του επιπέδου εφαρμογής ορίζεται το CBR (Constant Bit Rate). Αναλυτικά οι βασικές παράμετροι του σεναρίου Προσομοίωσης, φαίνονται στον παρακάτω πίνακα:

Παράμετρος	Τιμή
Πρωτόκολλο Επιπέδου Εφαρμογής	CBR
Περιοχή	500m ×500m/ 1000m x 1000m
Αριθμός Κόμβων	20 - 50
Εμβέλεια κεραιών	250m
Μέγεθος πακέτου	64 bytes
Πρωτόκολλο Επιπέδου MAC	802.11
Κίνηση κόμβων	Τυχαία
Κίνηση δικτύου	Τυχαία
Μέγιστη ταχύτητα κόμβων	10m/s
Ρυθμός μετάδοσης	2 πακέτα/s
Χρόνος προσομοίωσης	600s/900s

Πίνακας 5.1. Παράμετροι προσομοίωσης

Οι προσομοιώσεις χωρίζονται σε δύο επίπεδα δυσκολίας. Αρχικά, θα προσομοιωθούν 20 κόμβοι σε έναν υποτιθέμενο χώρο διαστάσεων 500 $\times 500m^2$ για χρόνο ίσο με 600s. Στη συνέχεια, εκτελούνται μεγαλύτερες προσομοιώσεις που περιλαμβάνουν 20 έως 50 κόμβους, σε έναν υποτιθέμενο χώρο διαστάσεων 1000 $\times 1000m^2$ για χρόνο ίσο με 900s.

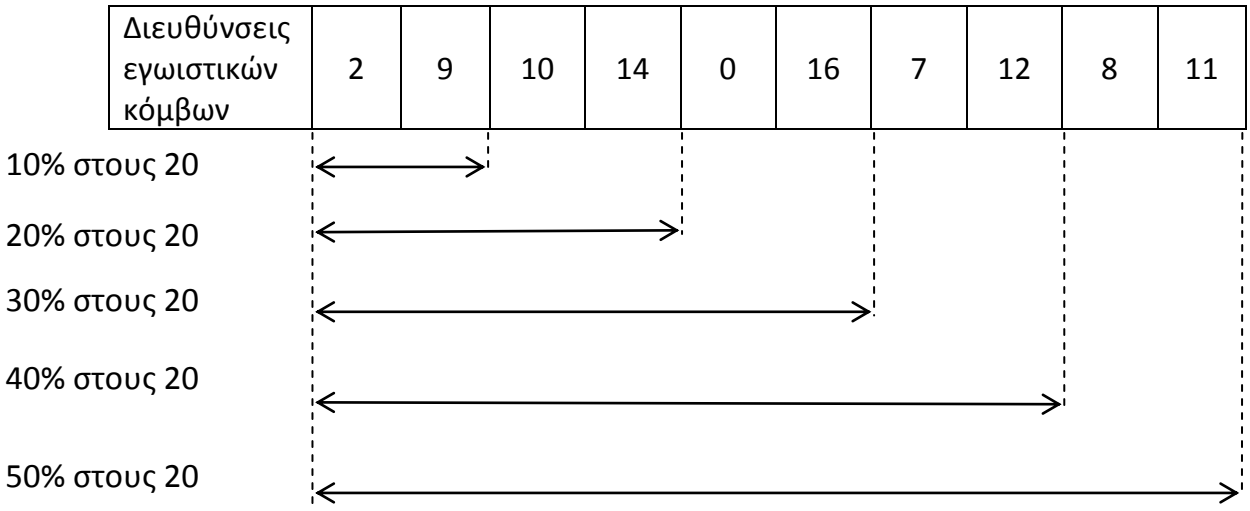
Η εμβέλεια των κεραιών των κινητών κόμβων είναι 250m. Το μέγεθος των πακέτων ορίζεται στα 64byte και ο ρυθμός μετάδοσης είναι 2 πακέτα/s, αυτό μας λέει ότι το εύρος (bandwidth) της πληροφορίας είναι αρκετό στα 2Mbps υπολογίζοντας και τον φόρτο του δικτύου λόγω των πακέτων του πρωτοκόλλου δρομολόγησης.

Σε κάθε προσομοίωση η κίνηση των κόμβων και η κίνηση του δικτύου καθορίζεται τυχαία. Αυτό γίνεται με τη βοήθεια αρχείων παραγωγής κίνησης δικτύου και κόμβων που περιέχονται στον NS-2.

Η εξαγωγή των αποτελεσμάτων γίνεται σε αρχεία τύπου trace(.tr) και σε αρχεία τύπου nam (.nam) . Στα πρώτα καταγράφονται όλα τα πακέτα που στάλθηκαν, λήφθηκαν, προωθήθηκαν και απορρίφθηκαν από τον κάθε κόμβο κατά τη διάρκεια της προσομοίωσης με κάθε λεπτομέρεια, όπως, ποια χρονική στιγμή στάλθηκαν, σε ποιο επίπεδο βρίσκονται, αν είναι πακέτα δρομολόγησης κ.ο.κ. Τα δεύτερα αρχεία (τύπου nam) χρησιμοποιούνται για την απεικόνιση της προσομοίωσης.

5.1.2. Προσομοίωση των Εγωιστικών Κόμβων

Για την προσομοίωση των Εγωιστικών Κόμβων χρησιμοποιείται ένας πίνακας ο οποίος περιέχει από 20 έως 50 διαφορετικές διευθύνσεις κόμβων που μπορούν να ενεργήσουν ως εγωιστικοί. Οι κόμβοι αυτοί δεν ενεργούν πάντα εγωιστικά ανάλογα με τον αριθμό των κόμβων που προσομοιώνουμε σε κάθε σενάριο και το ποσοστό τους έναντι των απλών κόμβων, καθορίζονται ποιοι κόμβοι τελικά θα ενεργήσουν εγωιστικά στο δίκτυο. Για την καλύτερη κατανόηση παρατίθεται ο παρακάτω πίνακας, που χρησιμοποιείται για προσομοίωση συνολικά 20 κόμβων.



Άρα στο σενάριο των 10% εγωιστικών κόμβων, οι εγωιστικοί κόμβοι θα είναι ο 2 και ο 9. Αντίστοιχα στο σενάριο των 20% εγωιστικών κόμβων θα είναι ο 2, ο 9, ο 10 και ο 14 κ.ο.κ.

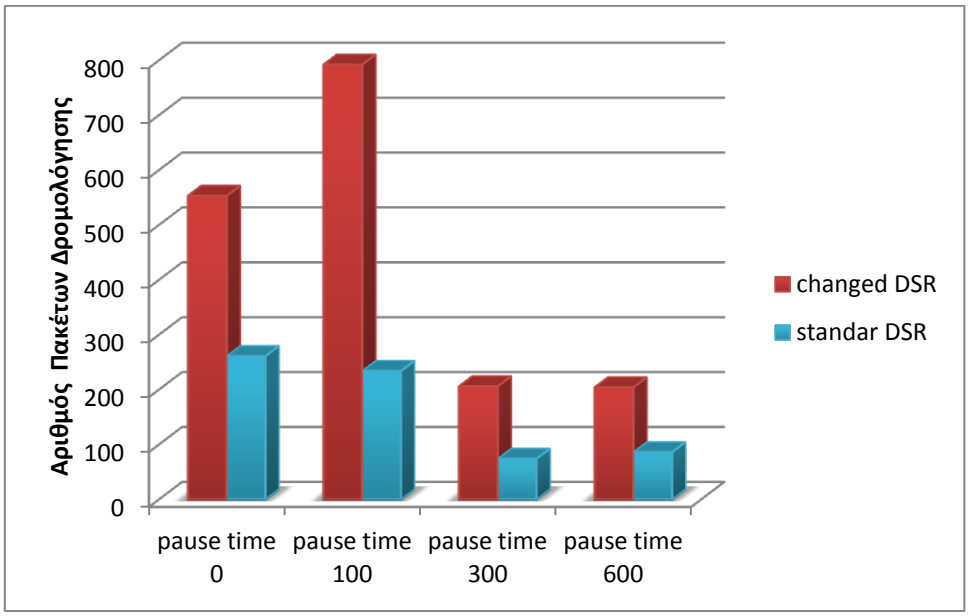
5.2 Αποτελέσματα

Σε αυτή την παράγραφο παρουσιάζονται τα αποτελέσματα των προσομοιώσεων και αναλύεται η εγκυρότητά τους.

5.2.1. Δυνατότητες DSR

Αρχικά, θέλοντας να κατανοήσουμε τα αποτελέσματα καλύτερα, πρέπει να ελέγξουμε τις επιπτώσεις που έχουν οι αλλαγές των δυνατοτήτων που έγιναν στον DSR για την σωστή λειτουργία του CONFIDANT. Για να το καταλάβουμε, προσομοιώνουμε πρώτα το DSR με τις δυνατότητες που δίνονται ως

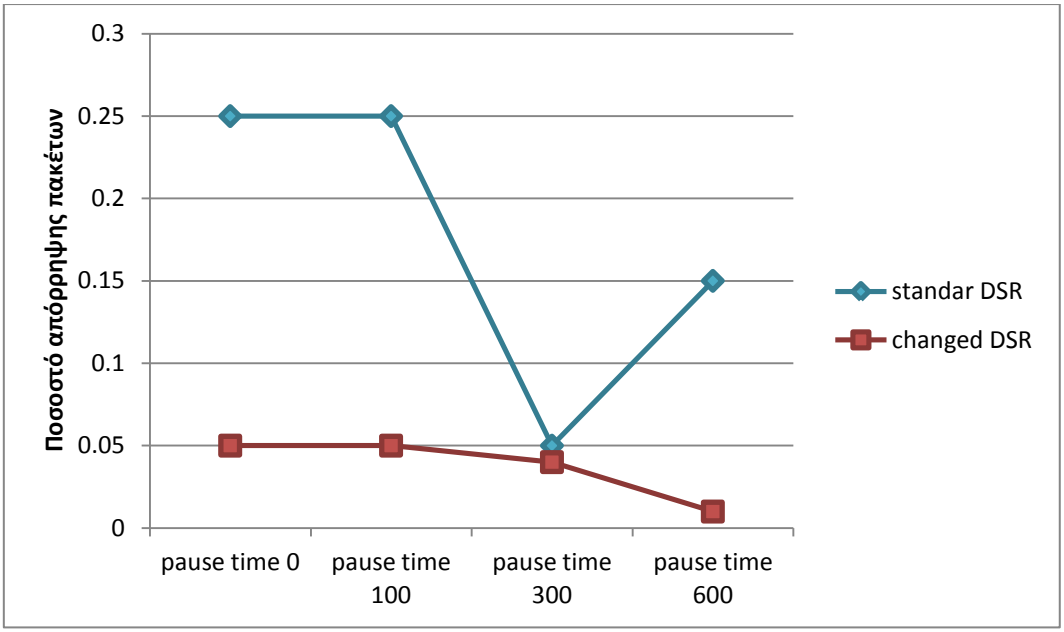
“προεπιλεγμένες” από τον NS-2 και στη συνέχεια προσομοιώνουμε με τις αλλαγές που κάναμε για το CONFIDANT. Τα αποτελέσματα φαίνονται παρακάτω:



Γράφημα 5.1. Αριθμός πακέτων δρομολόγησης ως προς χρόνους αδράνειας

Είναι εμφανές, πως οι αλλαγές που κάναμε, επηρεάζουν αισθητά των αριθμό πακέτων δρομολόγησης. Αυτό είναι και το επιθυμητό, αφού περισσότερα πακέτα δρομολόγησης σημαίνουν περισσότερες απόπειρες εύρεσης διαδρομών και κατ’ επέκταση περισσότερα μονοπάτια αποθηκευμένα στην λίστα Διαδρομών του DSR. Αντίθετα, υπάρχει αισθητή αύξηση της κίνησης στο δίκτυο με αποτέλεσμα να το επιβαρύνει.

Στη συνέχεια, για να γίνει έλεγχος πως όντως η λίστα Διαδρομών περιέχει περισσότερα μονοπάτια, συγκρίνεται το ποσοστό των πακέτων που απορρίφθηκαν στην πρώτη και στην δεύτερη περίπτωση. Τα αποτελέσματα φαίνονται παρακάτω:



Γράφημα 5.2. Ποσοστό απόρριψης πακέτων σε σχέση με τους χρόνους αδράνειας

Όπως βλέπουμε, και στις δύο περιπτώσεις, το ποσοστό των πακέτων που απορρίφθηκαν είναι πολύ μικρό. Όμως υπάρχουν μικρές διαφορές μεταξύ του DSR με τις "προεπιλεγμένες" δυνατότητες, και του DSR με την αλλαγή των δυνατοτήτων. Όπως φαίνεται, στην δεύτερη περίπτωση, υπάρχει μείωση του ποσοστού των πακέτων που απορρίφθηκαν κάτι που επιβεβαιώνει πως όντως η λίστα διαδρομών περιέχει περισσότερες διαδρομές.

5.2.2. Ποσοστό επιτυχούς προώθησης πακέτων

Αφού εξετάσαμε πως συμπεριφέρεται ο DSR με τις αλλαγές που κάναμε, είμαστε έτοιμοι να εξηγήσουμε τα πρώτα αποτελέσματα που δίνει το CONFIDANT.

Αρχικά, παρατίθεται το ποσοστό επιτυχούς δρομολόγησης πακέτων ως προς τους χρόνους αδράνειας. Για τα αποτελέσματα, προσομοιώθηκαν στον CONFIDANT DSR και στον απλό DSR (defenseless DSR) 20 κόμβοι, εκ των οποίων το 30% είναι εγωιστικοί. Αναλυτικά, τα χαρακτηριστικά της προσομοίωσης φαίνονται στον παρακάτω πίνακα:

Παράμετρος	**Τιμή**
Πρωτόκολλο Επιπέδου Εφαρμογής	CBR
Περιοχή	500m ×500m
Αριθμός Κόμβων	20
Εμβέλεια κεραιών	250m
Μέγεθος πακέτου	64 bytes
Πρωτόκολλο Επιπέδου MAC	802.11

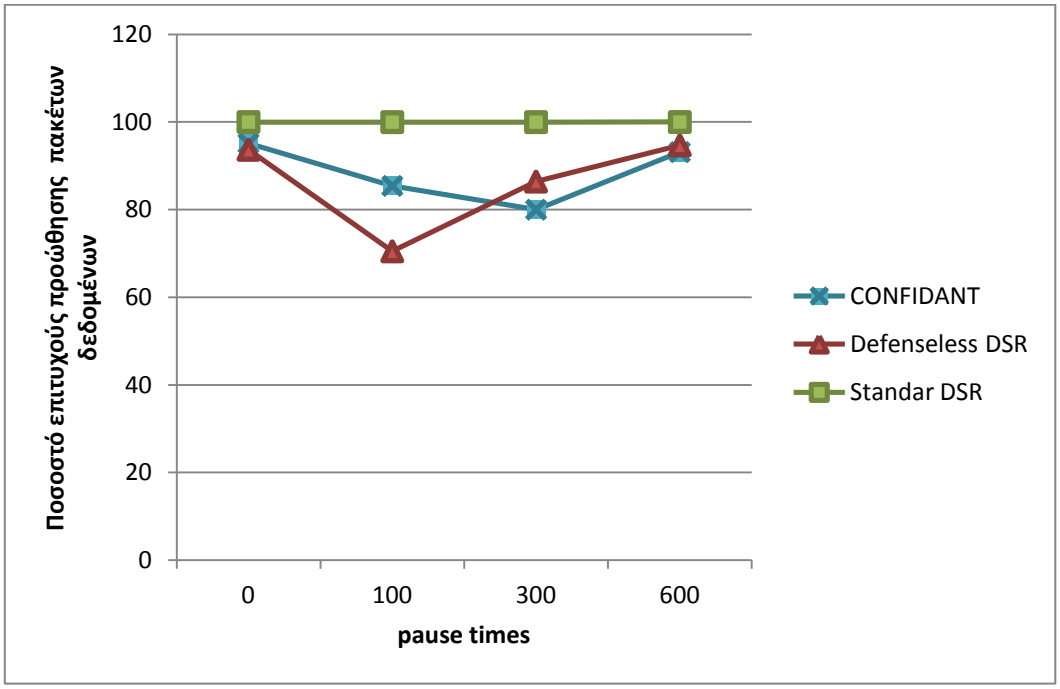
Κίνηση κόμβων	Τυχαία
Κίνηση δικτύου	Τυχαία
Μέγιστη ταχύτητα κόμβων	10m/s
Ρυθμός μετάδοσης	2 πακέτα/s
Χρόνος προσομοίωσης	600s
Ποσοστό εγωιστικών κόμβων	30%
Χρόνος Αδράνειας	0s – 600s

Πίνακας 5.2. Παράμετροι προσομοίωσης Ποσοστού επιτυχούς προώθησης πακέτων ως προς χρόνους αδράνειας

Τα αποτελέσματα φαίνονται στο γράφημα 5.2.

Το ποσοστό επιτυχούς δρομολόγησης πακέτων ορίζεται ως τα πακέτα δεδομένων που λήφθηκαν ως προς τα συνολικά πακέτα δεδομένων που στάλθηκαν (Packet Delivery Ratio, PDF).

$$PDF = \frac{\sum \pi\alpha\kappa\acute{\epsilon}\tau\omega\nu\ \delta\epsilon\delta o\mu\acute{\epsilon}\nu\omega\nu\ \pi o\upsilon\ \lambda\acute{\eta}\varphi\theta\eta\kappa\alpha\nu\ \epsilon\pi\iota\tau\upsilon\chi\acute{\omega}\varsigma}{\sum \pi\alpha\kappa\acute{\epsilon}\tau\omega\nu\ \delta\epsilon\delta o\mu\acute{\epsilon}\nu\omega\nu\ \pi o\upsilon\ \sigma\tau\acute{\alpha}\lambda\theta\eta\kappa\alpha\nu}$$



Γράφημα 5.3. Ποσοστό επιτυχούς προώθησης πακέτων ως προς χρόνους αδράνειας

Σύμφωνα με τα πρώτα αποτελέσματα, παρατηρείτε πως το CONFIDANT είναι σε θέση να αυξήσει το ποσοστό επιτυχούς δρομολόγησης πακέτων

δεδομένων στις περισσότερες περιπτώσεις. Όταν ο χρόνος αδράνειας είναι ίσος με το μηδέν (pause time = 0s) το CONFIDANT πλησιάζει την απόδοση του DSR χωρίς εγωιστικούς κόμβους. Ωστόσο, η διαφορά του CONFIDANT με το επιτιθέμενο DSR (defenseless DSR) δεν είναι μεγάλη. Όταν ο χρόνος αδράνειας είναι ίσος με εκατό (pause time = 100s) το CONFIDANT πετυχαίνει την καλύτερη επίδοσή του σε σχέση με το επιτιθέμενο DSR που δείχνει να έχει την χειρότερη απόδοση. Στην περίπτωση του χρόνου αδράνειας ίσου με τριακόσια δευτερόλεπτα (pause time = 300s) το CONFIDANT έχει την χειρότερη απόδοση με ποσοστό επιτυχούς προώθησης στο 80%, ενώ αντίθετα το επιτιθέμενο DSR δείχνει να τα πηγαίνει καλύτερα με ποσοστό προώθησης στο 86%. Τέλος, όταν το δίκτυο είναι στάσιμο (pause time = 600s) το CONFIDANT και το defenseless DSR πλησιάζουν την απόδοση της πρώτης περίπτωσης που το δίκτυο είναι συνεχώς σε κίνηση (pause time =0s).

Τα παραπάνω αποτελέσματα φανερώνουν πως το CONFIDANT αποδίδει καλύτερα σε σχέση με το απλό DSR στην περίπτωση του χρόνου αδράνειας 100s. Αντίθετα παίρνει τα χειρότερα αποτελέσματα στην περίπτωση των 300s αδράνειας. Οι περιπτώσεις των 0s και 600s αδράνειας, αποδίδουν το ίδιο περίπου με τον απλό DSR και κοντεύουν την απόδοση του DSR χωρίς καθόλου εγωιστικούς κόμβους.

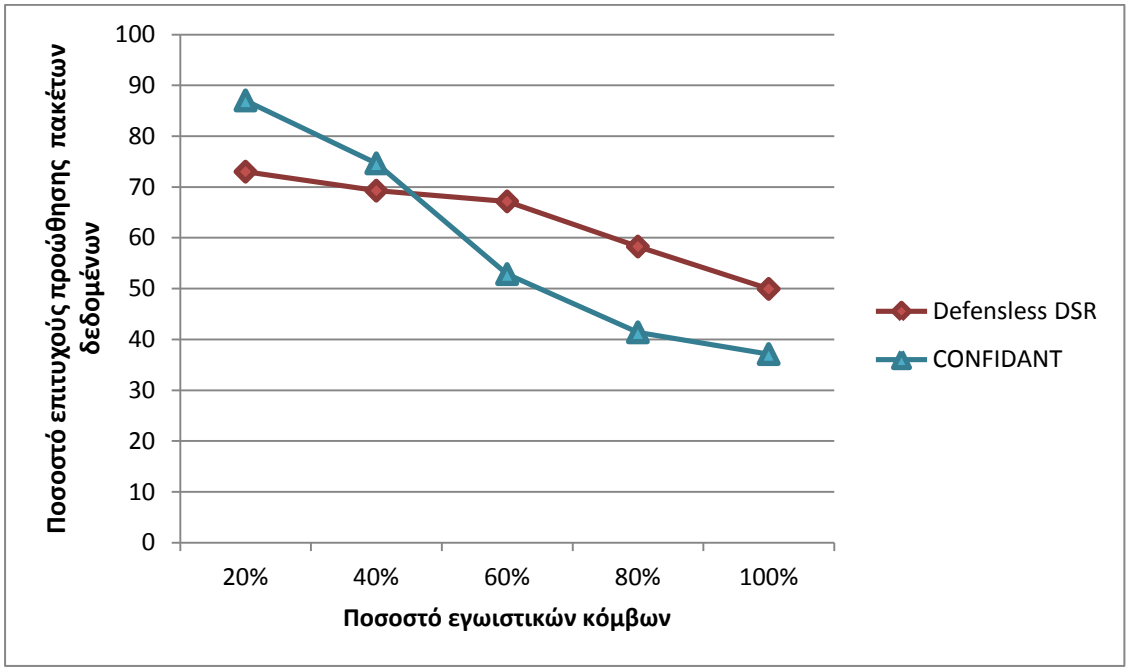
Στη συνέχεια εξετάζεται περαιτέρω η απόδοση του δικτύου για τις δύο περιπτώσεις χρόνων αδράνειας που παρατηρείται μεγαλύτερη διαφορά μεταξύ του CONFIDANT και του επιτιθέμενου DSR (defenseless DSR).

Παράμετρος	**Τιμή**
Πρωτόκολλο Επιπέδου Εφαρμογής	CBR
Περιοχή	500m ×500m
Αριθμός Κόμβων	20
Εμβέλεια κεραιών	250m
Μέγεθος πακέτου	64 bytes
Πρωτόκολλο Επιπέδου MAC	802.11
Κίνηση κόμβων	Τυχαία
Κίνηση δικτύου	Τυχαία
Μέγιστη ταχύτητα κόμβων	10m/s
Ρυθμός μετάδοσης	2 πακέτα/s
Χρόνος προσομοίωσης	600s
Ποσοστό εγωιστικών κόμβων	20% - 100%
Χρόνος Αδράνειας	100s

Πίνακας 5.3. Ποσοστό επιτυχούς προώθησης ως προς ποσοστό εγωιστικών κόμβων

Αρχικά εξετάζεται η περίπτωση του χρόνου αδράνειας ίσου με 100s. Όπως είδαμε παραπάνω, αυτή η περίπτωση είναι ενδιαφέρουσα γιατί παρατηρείτε η μεγαλύτερη διαφορά απόδοσης σε σχέση με το επιτιθέμενο DSR. Για την εξέταση της περίπτωσης αυτής, συγκρίνουμε το CONFIDANT με το απλό επιτιθέμενο DSR, σε σχέση με την απόδοσή τους όταν το ποσοστό των εγωιστικών κόμβων του δικτύου αυξάνει. Οι παράμετροι της προσομοίωσης φαίνονται στο πίνακα 5.3.

Σύμφωνα με τα αποτελέσματα του γραφήματος 5.4. παρατηρείται πως καθώς το ποσοστό εγωιστικών κόμβων στο δίκτυο αυξάνει, η επιτυχής προώθηση πακέτων δεδομένων μειώνετε. Πιο συγκεκριμένα, μέχρι την περίπτωση των 40% εγωιστικών κόμβων, το CONFIDANT κατέχει μεγαλύτερο ποσοστό επιτυχούς προώθησης πακέτων από το απλό DSR.Αντίθετα, από το 40% και πάνω, το CONFIDANT μειώνει την απόδοσή του φτάνοντας στο 37% όταν όλοι οι κόμβοι του δικτύου ενεργούν εγωιστικά.



Γράφημα 5.4. Ποσοστό προώθησης πακέτων ως προς ποσοστό εγωιστικών κόμβων (pause time = 100s)

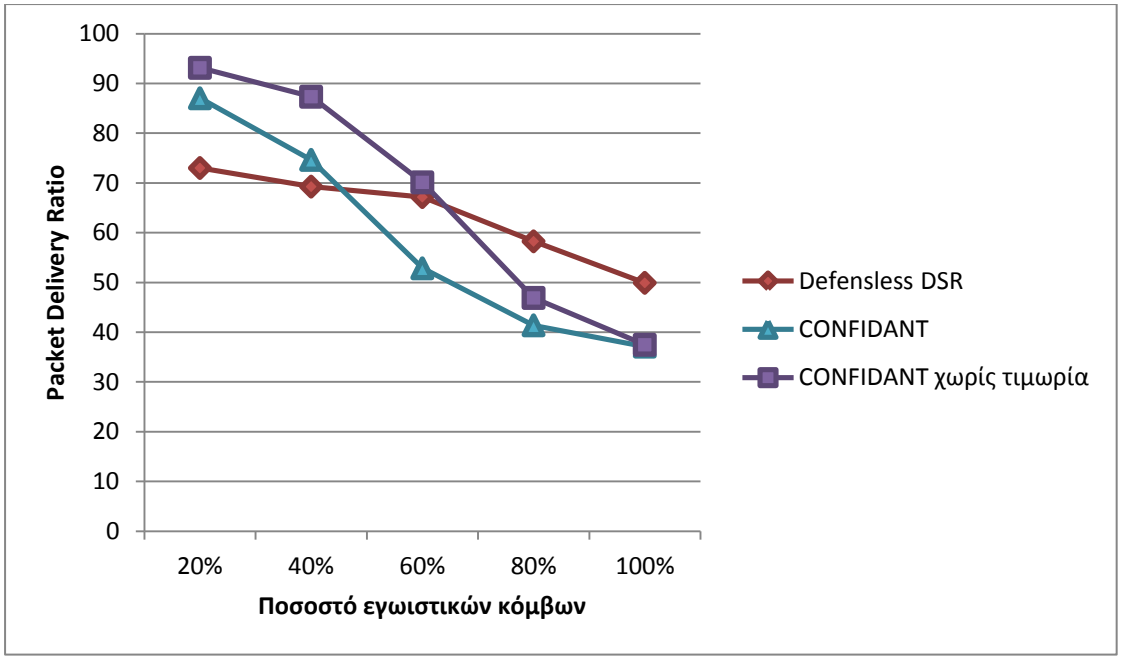
Τα αποτελέσματα του γραφήματος 5.4. είναι λογικά αν σκεφτεί κανείς την λειτουργία του μηχανισμού CONFIDANT. Όταν αναγνωριστεί ένας κόμβος ως

μη συνεργάσιμος, το CONFIDANT απομονώνει όλα τα μονοπάτια που περιέχουν τον κόμβο αυτό. Επίσης, λόγω της τιμωρίας, κάθε φορά που ένας κόμβος λαμβάνει κάποιο πακέτο από έναν μη συνεργάσιμο κόμβο, το απορρίπτει. Όσο το ποσοστό των εγωιστικών κόμβων διατηρείται σε χαμηλά επίπεδα, η λίστα Διαδρομών του πρωτοκόλλου περιέχει περισσότερες εναλλακτικές διαδρομές για την προώθηση πακέτων και έτσι τα πακέτα που προωθούνται επιτυχώς είναι περισσότερα. Όσο το ποσοστό των εγωιστικών κόμβων αυξάνει, οι εναλλακτικές διαδρομές μειώνονται με αποτέλεσμα περισσότερα πακέτα να απορρίπτονται.

Θέλοντας να παρατηρήσουμε την απόδοση του δικτύου χωρίς την λειτουργία της τιμωρίας, δηλαδή χωρίς να απορρίπτονται πακέτα που προέρχονται από μη συνεργάσιμους κόμβους, κατασκευάζουμε το γράφημα 5.5. σύμφωνα πάλι με τις παραμέτρους του πίνακα 5.3.

Γίνεται αντιληπτό από το γράφημα 5.5. πως η τιμωρία λειτουργεί αρνητικά στην απόδοση του δικτύου. Το CONFIDANT χωρίς την απόρριψη πακέτων που προέρχονται από εγωιστικούς κόμβους, πετυχαίνει μεγαλύτερο ποσοστό προώθησης μέχρι και την περίπτωση των 60% μη συνεργάσιμων κόμβων. Από το 60% και έπειτα, η απόδοση πέφτει και φτάνει ίση με το CONFIDANT όταν όλοι οι κόμβοι του δικτύου ενεργούν εγωιστικά.

Τα συγκεκριμένα αποτελέσματα είναι εύκολα κατανοητά, αφού τα πακέτα που προωθούνται από εγωιστικούς κόμβους φτάνουν επιτυχώς στον προορισμό τους και έτσι το συνολικό ποσοστό επιτυχούς προώθησης πακέτων αυξάνεται.



Γράφημα 5.5. Ποσοστό επιτυχούς προώθησης πακέτων ως προς ποσοστό εγωιστικών κόμβων (pause time = 100s, σύγκριση τιμωρίας)

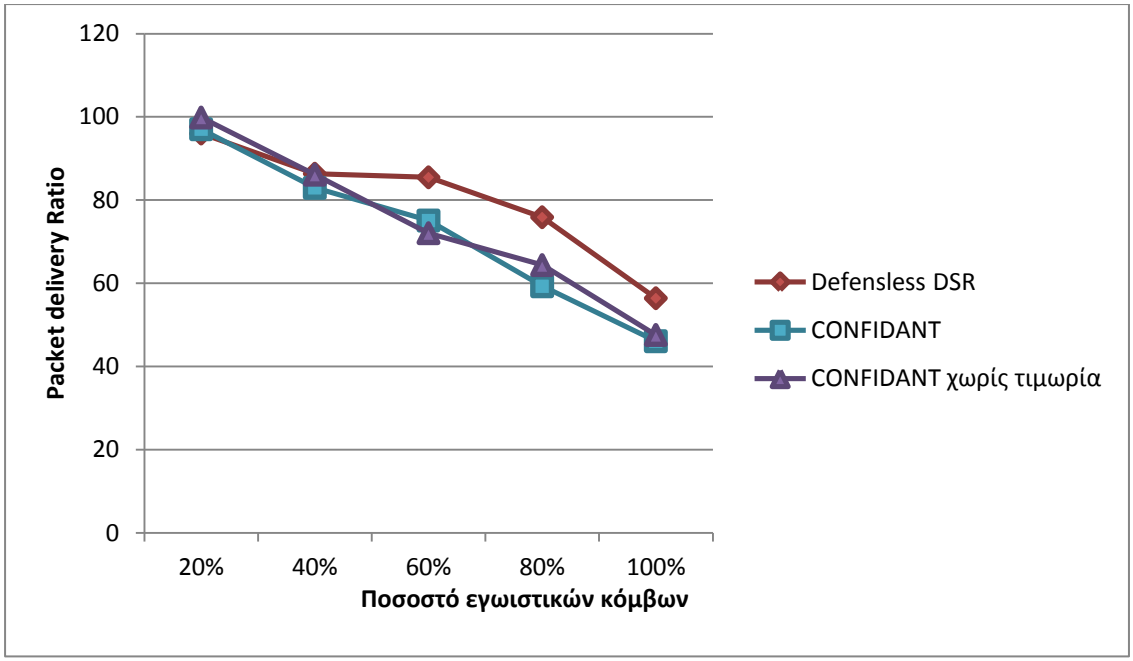
Η συνολική καλή απόδοση της περίπτωσης των 100 δευτερολέπτων αδράνειας σε σύγκριση με το απλό DSR εξηγείται αν παραπέμψουμε στα αποτελέσματα του γραφήματος 5.1. Σε αυτή την περίπτωση βλέπουμε πως τα πακέτα δρομολόγησης είναι πολύ περισσότερα από οποιαδήποτε άλλη περίπτωση χρόνου αδράνειας. Αυτό σημαίνει πως το πρωτόκολλο DSR διενεργεί πολύ περισσότερες ενέργειες εύρεσης διαδρομών και ως επακόλουθο έχει περισσότερα μονοπάτια διαθέσιμα στην λίστα διαδρομών του. Η ενέργεια αυτή του DSR είναι ιδιαίτερα ευεργετική για τον μηχανισμό CONFIDANT που επωφελείται από τις περισσότερες διαδρομές στην λίστα διαδρομών, αφού μπορεί να χρησιμοποιήσει περισσότερα εναλλακτικά μονοπάτια. Ωστόσο, το πρωτόκολλο DSR δείχνει να λειτουργεί χειρότερα σε αυτή την περίπτωση (γράφημα 5.2.) κάτι που επαληθεύει την κακή επίδοσή του όταν κακόβουλοι κόμβοι είναι παρών στο δίκτυο.

Στη συνέχεια παρουσιάζεται η περίπτωση του χρόνου αδράνειας ίσου με 300s. Σε αυτή την περίπτωση το CONFIDANT είχε τη χαμηλότερη επίδοση σε σύγκριση με το απλό επιτιθέμενο DSR. Για την εξέταση της περίπτωσης αυτής, συγκρίνουμε το CONFIDANT, το CONFIDANT χωρίς τιμωρία και το απλό επιτιθέμενο DSR. Η σύγκριση γίνεται ξανά σε σχέση με την απόδοσή τους όταν το ποσοστό των εγωιστικών κόμβων του δικτύου αυξάνει. Οι παράμετροι της προσομοίωσης φαίνονται στο πίνακα 5.4.

Παράμετρος	Τιμή
Πρωτόκολλο Επιπέδου Εφαρμογής	CBR
Περιοχή	500m ×500m
Αριθμός Κόμβων	20
Εμβέλεια κεραιών	250m
Μέγεθος πακέτου	64 bytes
Πρωτόκολλο Επιπέδου MAC	802.11
Κίνηση κόμβων	Τυχαία
Κίνηση δικτύου	Τυχαία
Μέγιστη ταχύτητα κόμβων	10m/s
Ρυθμός μετάδοσης	2 πακέτα/s
Χρόνος προσομοίωσης	600s
Ποσοστό εγωιστικών κόμβων	20% - 100%
Χρόνος Αδράνειας	300s

Πίνακας 5.4. Ποσοστό επιτυχούς προώθησης ως προς ποσοστό εγωιστικών κόμβων

Στο γράφημα 5.6. παρατηρείτε μειωμένη απόδοση του CONFIDANT σε σχέση με το απλό DSR. Όπως παρατηρήθηκε και στα πρώτα αποτελέσματα (γράφημα 5.3.), το CONFIDANT είτε με την τιμωρία είτε χωρίς αυτήν, αποδίδει χειρότερα από το απλό DSR στην περίπτωση του χρόνου αδράνειας ίσου με 300s. Από το ποσοστό των 40% έως το ποσοστό των 100% εγωιστικών κόμβων στο δίκτυο, το CONFIDANT προωθεί επιτυχώς λιγότερα πακέτα. Μόνο στην περίπτωση των 20% εγωιστικών κόμβων το CONFIDANT επιτυγχάνει να προωθήσει περισσότερα πακέτα σε σχέση με τον απλό DSR.



Γράφημα 5.6. Ποσοστό επιτυχούς προώθησης πακέτων ως προς ποσοστό εγωιστικών κόμβων (pause time = 300s, σύγκριση τιμωρίας)

Η συμπεριφορά αυτή του μηχανισμού CONFIDANT είναι λογική αν παραπέμψουμε στο γράφημα 5.1. Εκεί παρατηρούμε πως στην περίπτωση των 300 δευτερολέπτων αδράνειας, το πρωτόκολλο DSR με αλλαγμένες δυνατότητες παράγει πολύ λίγα πακέτα δρομολόγησης σε σύγκριση με αυτά των 100 και 0 δευτερολέπτων αδράνειας. Αυτό σημαίνει πολύ λιγότερες διαδικασίες εύρεσης διαδρομών και ακολούθως λιγότερα διαθέσιμα μονοπάτια στην λίστα Διαδρομών του. Κάτι τέτοιο λειτουργεί αρνητικά για τον μηχανισμό CONFIDANT που στηρίζει ένα μεγάλο ποσοστό της απόδοσής του στις εναλλακτικές διαδρομές. Αντίθετα, το πρωτόκολλο DSR δείχνει να λειτουργεί καλύτερα σε αυτή την περίπτωση (γράφημα 5.2.) κάτι που επαληθεύει την καλύτερη απόδοση του απλού DSR με επίθεση εγωιστικών κόμβων.

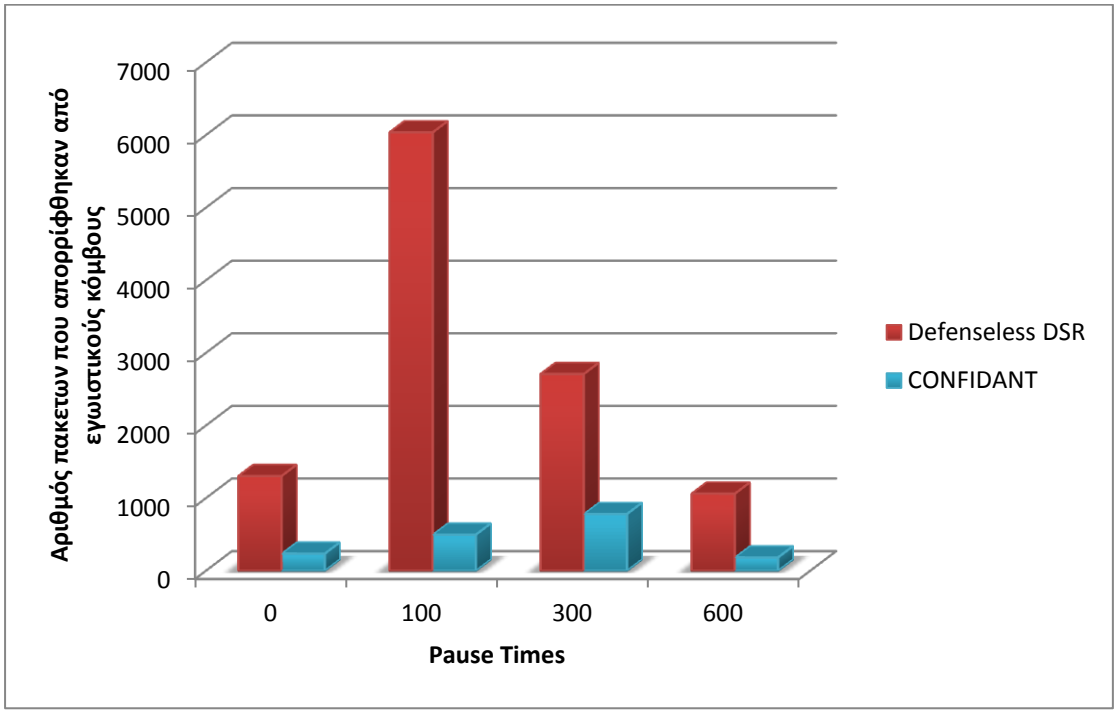
5.2.3. Απόρριψη πακέτων από εγωιστικούς κόμβους

Σκοπός του CONFIDANT είναι η επιτυχής αναγνώριση εγωιστικών κόμβων και ως αποτέλεσμα η αποφυγή απόρριψης πακέτων από εγωιστικούς κόμβους. Γι’ αυτό κρίνεται απαραίτητο να ελεγχθεί αν το CONFIDANT εκπληρώνει τον σκοπό του.

Για να γίνει αυτό συγκρίνουμε τον αριθμό πακέτων που απορρίφθηκαν από εγωιστικούς κόμβους όταν το DSR λειτουργεί με τον μηχανισμό CONFIDANT και όταν το DSR δεν χρησιμοποιεί κάποιον μηχανισμό προώθησης

συνεργασίας. Τα αποτελέσματα συγκρίνονται πάλι με κριτήριο τους χρόνους αδράνειας.

Για τις προσομοιώσεις χρησιμοποιήθηκαν και πάλι οι παράμετροι του πίνακα 5.2.



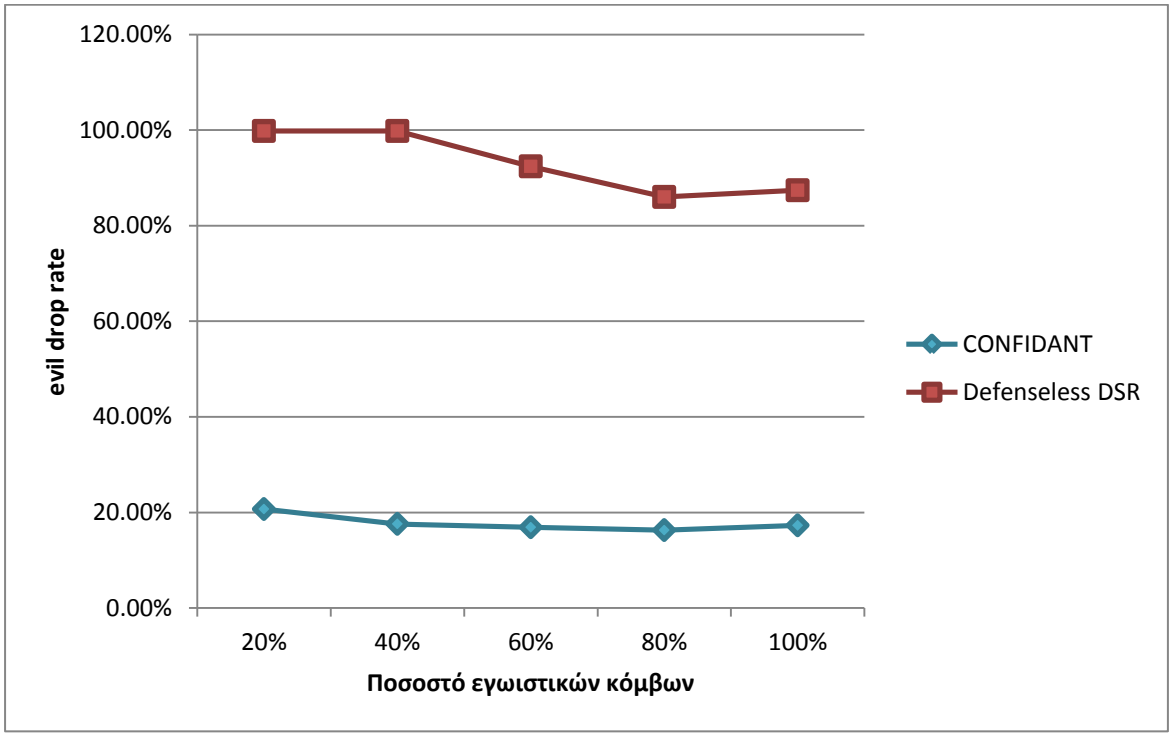
Γράφημα 5.7. Αριθμός πακέτων που απορρίφθηκαν από εγωιστικούς κόμβους σε σύγκριση με τους χρόνους αδράνειας.

Και στις τέσσερις περιπτώσεις αδράνειας, το CONFIDANT μειώνει κατά πολύ την εγωιστική απόρριψη πακέτων σε σχέση με το απλό DSR. Η μεγαλύτερη διαφορά παρατηρείται στην περίπτωση των 100 δευτερολέπτων αδράνειας, ενώ η μικρότερη στην περίπτωση των 300 δευτερολέπτων αδράνειας.

Συνεχίζουμε ελέγχοντας το ποσοστό εγωιστικής απόρριψης πακέτων για τις παραπάνω δύο ακραίες περιπτώσεις των 100 και 300 δευτερολέπτων, για διάφορα ποσοστά εγωιστικών κόμβων στο δίκτυο. Οι προσομοιώσεις χρησιμοποιούν τις παραμέτρους των πινάκων 5.3. και 5.4. αντίστοιχα. Συγκρίνεται σε κάθε γράφημα το ποσοστό αυτό για το CONFIDANT DSR και τον απλό DSR.

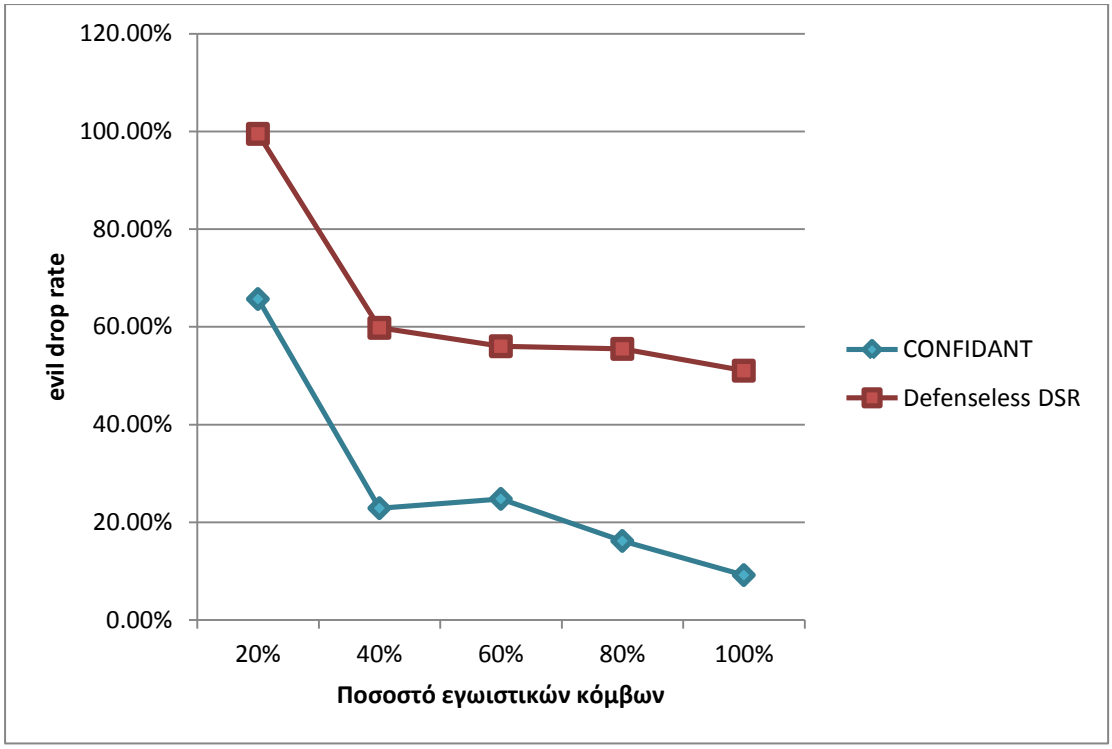
Για την μέτρηση του ποσοστού απόρριψης πακέτων από εγωιστικούς κόμβους, χρησιμοποιείται η εξίσωση:

$$evil\ drop\ rate = \frac{\sum \pi\alpha\kappa\acute{\epsilon}\tau\omega\nu\ \pi o\upsilon\ \alpha\pi o\rho\rho\acute{\iota}\varphi\theta\eta\kappa\alpha\nu\ \alpha\pi\acute{o}\ \epsilon\gamma\omega\iota\sigma\tau\iota\kappa o\acute{\upsilon}\varsigma\ \kappa\acute{o}\mu\beta o\upsilon\varsigma}{\sum \pi\alpha\kappa\acute{\epsilon}\tau\omega\nu\ \pi o\upsilon\ \alpha\pi o\rho\rho\acute{\iota}\varphi\theta\eta\kappa\alpha\nu}$$



Γράφημα 5.8. Ποσοστό απόρριψης πακέτων από εγωιστικούς κόμβους για χρόνο αδράνειας 100s

Στο γράφημα 5.8. διακρίνεται πως σχεδόν το 100% των συνολικά απορριφθέντων πακέτου του δικτύου οφείλεται στους εγωιστικούς κόμβους για την περίπτωση του απλού DSR. Αντίθετα, το ποσοστό εγωιστικά απορριφθέντων πακέτων για το CONFIDANT βρίσκεται κάτω από το 20% στις περισσότερες περιπτώσεις.



Γράφημα 5.9. Ποσοστό απόρριψης πακέτων από εγωιστικούς κόμβους για χρόνο αδράνειας 300s

Στο γράφημα 5.9. παρατηρούμε πως και πάλι το ποσοστό των εγωιστικά απορριφθέντων πακέτων είναι υψηλότερο στην περίπτωση του απλού DSR και αρκετά χαμηλότερο στην περίπτωση του CONFIDANT. Ωστόσο, στην περίπτωση αυτή διακρίνουμε πως το απλό DSR επηρεάζεται και από άλλους παράγοντες απόρριψης πακέτων πέρα από τα εγωιστικά drops. Αυτό επαληθεύει και την καλύτερη απόδοσή του σε σχέση με το CONFIDANT στα γραφήματα 5.3. και 5.6. που αναλύθηκαν παραπάνω. Όταν διαφορετικοί παράγοντες απόρριψης επηρεάζουν το DSR, σημαίνει πως οι εγωιστικοί κόμβοι δεν επηρεάζουν αρκετά τα αποτελέσματά του και άρα δείχνει να έχει καλύτερη απόδοση σε σχέση με το CONFIDANT. Αντίθετα, το CONFIDANT επηρεάζεται από τους υπόλοιπους παράγοντες απόρριψης πακέτων που πλήττουν το DSR και άρα η απόδοσή του μειώνεται.

5.2.4. Λόγοι που ευθύνονται για την απόρριψη πακέτων στο CONFIDANT.

Μετά την ανάλυση των παραπάνω αποτελεσμάτων, δημιουργείται μία εύλογη απορία. Αφού το ποσοστό των εγωιστικά απορριφθέντων πακέτων μειώνεται, γιατί η απόδοσή του CONFIDANT δεν είναι αρκετά μεγαλύτερη σε σύγκριση με αυτή του απλού DSR;

Για να λύσουμε την απορία αυτή, δημιουργούμε ένα γράφημα στο οποίο φαίνονται οι λόγοι που το CONFIDANT απορρίπτει πακέτα. Τα αποτελέσματα που παίρνουμε είναι απόρροια πολλών προσομοιώσεων με διαφορετικές παραμέτρους και εξαγωγή της μέσης τιμής αυτών.

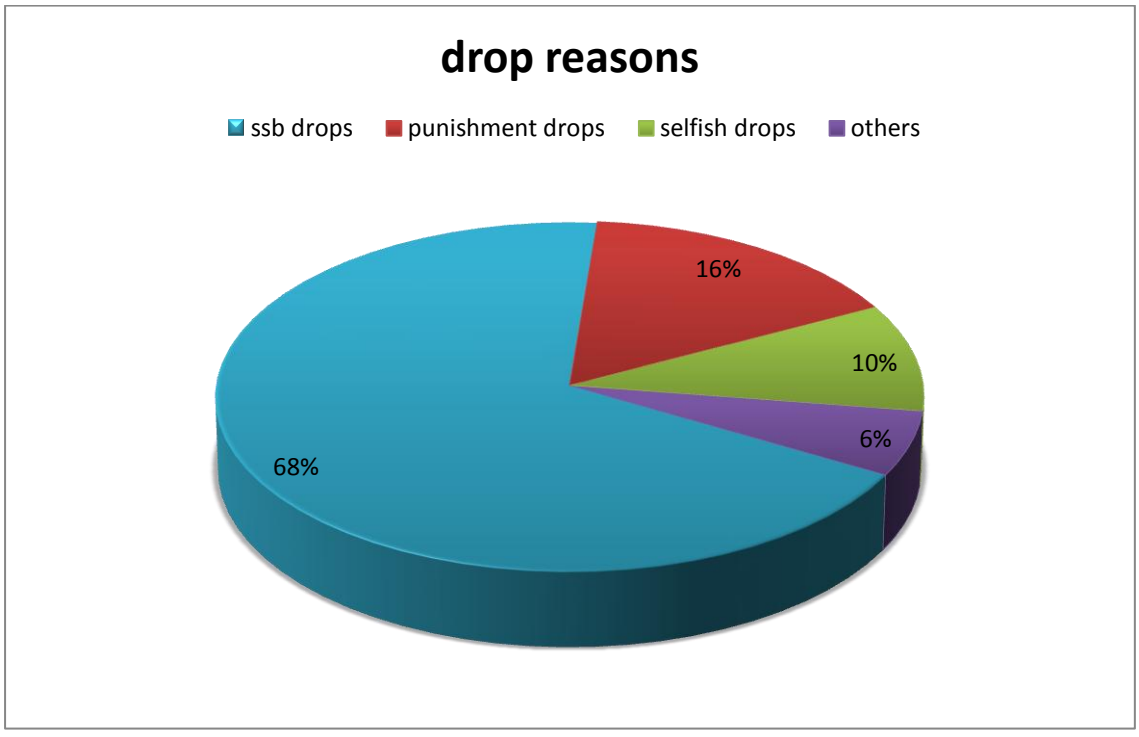
Πριν προχωρήσουμε στα αποτελέσματα ορίζουμε ως:

Selfish drops: τα πακέτα που απορρίπτονται από μία εγωιστική ενέργεια, δηλαδή τα πακέτα που απορρίπτονται από εγωιστικούς κόμβους.

Punishment drops: τα πακέτα που απορρίπτονται λόγω της τιμωρίας, δηλαδή τα πακέτα που απορρίπτονται επειδή ο αποστολέας τους θεωρείτε εγωιστικός.

SSb drops: είναι τα πακέτα που απορρίπτονται όταν λήξουν στον Send Buffer του DSR. Στην υλοποίηση του πρωτοκόλλου DSR κάθε πακέτο που περνάει στο επίπεδο δρομολόγησης, μπορεί να παραμείνει εκεί έως ότου ο χρόνος του λήξει. Συνήθως αυτό το είδος drop οφείλεται στην έλλειψη μονοπατιών.

Others: είναι άλλοι λόγοι απόρριψης πακέτων, όπως για παράδειγμα η αυξημένη κίνηση στο δίκτυο ή η λήξη μίας σύνδεσης.



Γράφημα 5.10. Λόγοι απόρριψης πακέτων στο CONFIDANT

Το μεγαλύτερο ποσοστό απόρριψης πακέτων οφείλεται στα λεγόμενα SSb drops, όπως εύκολα μπορούμε να διακρίνουμε στο γράφημα 5.10. Η απόρριψη πακέτων για τον λόγο αυτό φτάνει το 68% των συνολικών απορρίψεων. Αυτό σημαίνει πως το CONFIDANT ή δεν έχει αρκετές εναλλακτικές διαδρομές ή δεν έχει διαδρομές ή περιορίζει αρκετά τις διαδρομές του επειδή αναγνωρίζει περισσότερους κόμβους ως εγωιστικούς απ' ότι πρέπει. Η τελευταία περίπτωση απορρίπτεται λόγω του χαμηλού ποσοστού πακέτων λόγω της τιμωρίας. Εάν το CONFIDANT αναγνώριζε πολύ περισσότερους κόμβους λανθασμένα ως εγωιστικούς, το ποσοστό τιμωρίας (punishment drop) θα έπρεπε να ήταν πολύ μεγαλύτερο.

Με αυτό τον τρόπο συμπεραίνουμε πως το CONFIDANT οφείλει την χαμηλή του απόδοση στην αδυναμία του DSR πρωτοκόλλου να ανακαλύψει όσο το δυνατόν περισσότερες εναλλακτικές διαδρομές κατά την εφαρμογή του στον προσομοιωτή ns-2. Αυτό επαληθεύεται και από το γράφημα 5.1. σε συνεργασία με τα αποτελέσματα της παραγράφου 5.2.2. Όταν το DSR ανακαλύπτει περισσότερες διαδρομές, τότε το CONFIDANT αποδίδει καλύτερα (περίπτωση 100 δευτερολέπτων χρόνου αδράνειας).

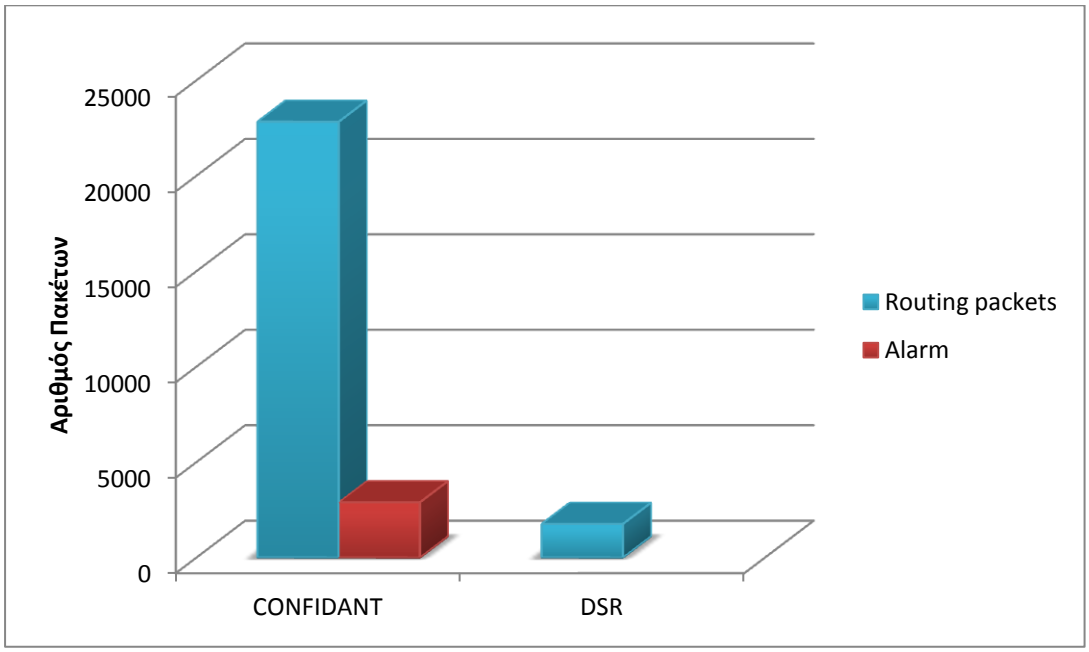
5.2.5. Η επιβάρυνση του δικτύου

Όπως αναφέρθηκε στην ανάλυση της υλοποίησης του CONFIDANT, για την λειτουργία του επιβαρύνει το δίκτυο με πακέτα δρομολόγησης τύπου ALARM. Εκτός από τα πακέτα ALARM, το δίκτυο επίσης επιβαρύνεται με περισσότερα πακέτα δρομολόγησης τύπου ROUTE REQUEST και ROUTE REPLY λόγω της απόρριψης διαδρομών.

Είναι σκόπιμο λοιπόν να συγκρίνουμε τα πακέτα δεδομένων που στέλνονται συνολικά από το DSR και από το CONFIDANT. Για την εξαγωγή των αποτελεσμάτων χρησιμοποιήθηκαν αποτελέσματα διάφορων προσομοιώσεων και υπολογίστηκε η μέση τιμή τους.

Στο γράφημα 5.11. διακρίνουμε η επιβάρυνση είναι αισθητά μεγαλύτερη σε σχέση με τον απλό DSR. Ωστόσο η επιβάρυνση των ALARM πακέτων που στέλνονται από το CONFIDANT είναι μικρή σε σχέση με τα υπόλοιπα πακέτα δρομολόγησης, αυτό σημαίνει πως η αποστολή τους δεν επιβαρύνει τόσο το δίκτυο.

Μεγάλη όμως είναι η επιβάρυνση του δικτύου από τα υπόλοιπα πακέτα δρομολόγησης. Η αύξηση των ROUTE REQUEST και ROUTE REPLY οφείλεται στην αδυναμία του DSR να βρει μονοπάτια αποστολής για τα πακέτα του.



Γράφημα 5.11. Επιβάρυνση του δικτύου λόγω CONFIDANT

Κάθε φορά που το CONFIDANT αναγνωρίζει έναν εγωιστικό κόμβο, απομονώνει όλα τα μονοπάτια που βρίσκονται στην λίστα Διαδρομών του DSR στα οποία υπάρχει αυτός ο κόμβος. Ως αποτέλεσμα, όταν ο κόμβος θελήσει να στείλει ένα πακέτο δεδομένων είτε στον εγωιστικό κόμβο είτε σε έναν κόμβο που βρίσκεται σε σημείο όπου αναγκαστικά πρέπει να προωθήσει ο εγωιστικός κόμβος, δεν θα υπάρχει διαδρομή στην λίστα Διαδρομών του. Γι' αυτό το λόγο θα προσπαθεί συνέχεια να βρει μια διαδρομή εκτελώντας διαδικασία εύρεσης διαδρομής στέλνοντας συνεχώς πακέτα τύπου ROUTE REQUEST.

Αν ο προορισμός είναι ένας εγωιστικός κόμβος, εκείνος θα απαντήσει στέλνοντας ένα ROUTE REPLY, αλλά και πάλι ο Path Manager του CONFIDANT θα απομονώσει την διαδρομή με αποτέλεσμα ο κόμβος αποστολέας να εκκινήσει ξανά μία διαδικασία Εύρεσης Διαδρομών. Αυτό θα συνεχιστεί μέχρι κάτι να αλλάξει στο δίκτυο ή μέχρι η βαθμολογία του εγωιστικού κόμβου στον κόμβο αποστολέα να μειωθεί.

Αν ο προορισμός είναι ένας κόμβος που βρίσκεται εκτός εμβέλειας του κόμβου αποστολέα και αναγκαστικά στην αποστολή ενός πακέτου πρέπει να συμπεριληφθεί ένας εγωιστικός κόμβος, τότε ο εγωιστικός κόμβος θα προωθήσει το ROUTE REQUEST του κόμβου αποστολέα, αλλά στη συνέχεια θα απορρίψει το ROUTE REPLY του κόμβου παραλήπτη. Αυτό θα επαναληφθεί αρκετές φορές έως ότου κάτι αλλάξει στο δίκτυο ή μειωθεί η βαθμολογία του εγωιστικού κόμβου από τον κόμβο αποστολέα.

Και στις δύο παραπάνω περιπτώσεις, το δίκτυο επιβαρύνεται με πολλά πακέτα δρομολόγησης. Επίσης, οι παραπάνω περιπτώσεις παίζουν σημαντικό ρόλο στην απόρριψη πακέτων και στην όχι τόσο καλή απόδοση του CONFIDANT σε σχέση με τον απλό DSR.

5.2.6. Προσομοίωση μεγαλύτερων σεναρίων

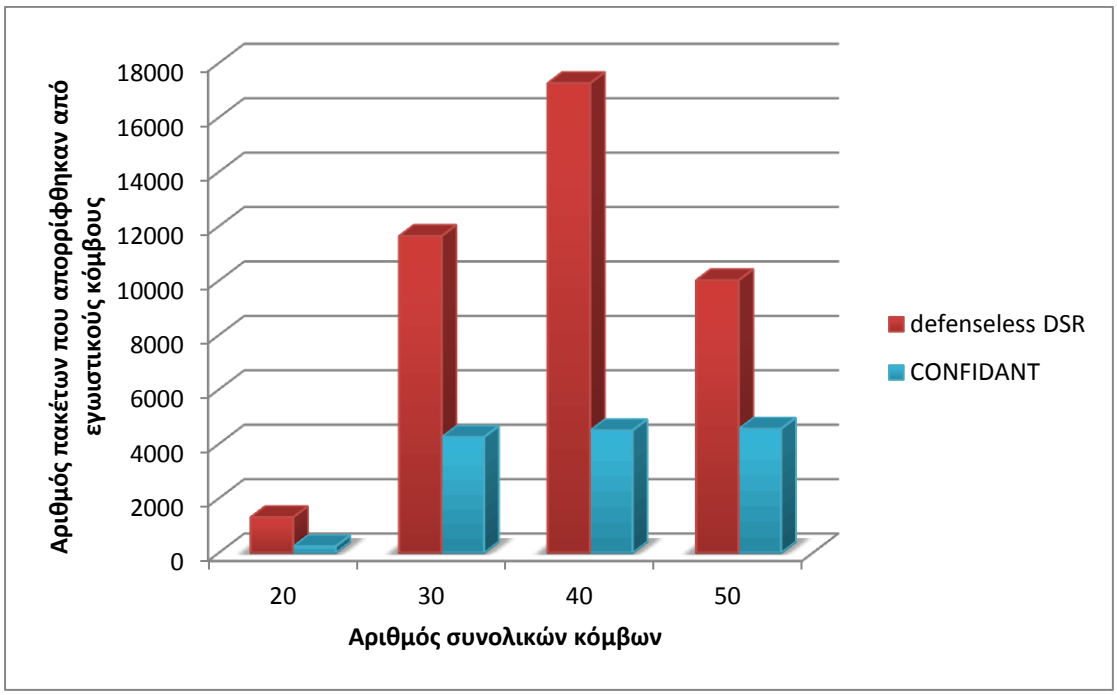
Τέλος, για τον έλεγχο της λειτουργίας του CONFIDANT σε μεγαλύτερα σενάρια, παρατίθενται γραφήματα που συγκρίνουν την απόδοση του CONFIDANT για διαφορετικούς αριθμούς κόμβων. Για την εξαγωγή των αποτελεσμάτων χρησιμοποιήθηκαν οι παράμετροι του πίνακα 5.5.

Παράμετρος	**Τιμή**
Πρωτόκολλο Επιπέδου Εφαρμογής	CBR
Περιοχή	1000m ×1000m
Αριθμός Κόμβων	20 - 50
Εμβέλεια κεραιών	250m
Μέγεθος πακέτου	64 bytes
Πρωτόκολλο Επιπέδου MAC	802.11
Κίνηση κόμβων	Τυχαία
Κίνηση δικτύου	Τυχαία
Μέγιστη ταχύτητα κόμβων	10m/s
Ρυθμός μετάδοσης	2 πακέτα/s
Χρόνος προσομοίωσης	900s
Ποσοστό εγωιστικών κόμβων	30%
Χρόνος Αδράνειας	0s

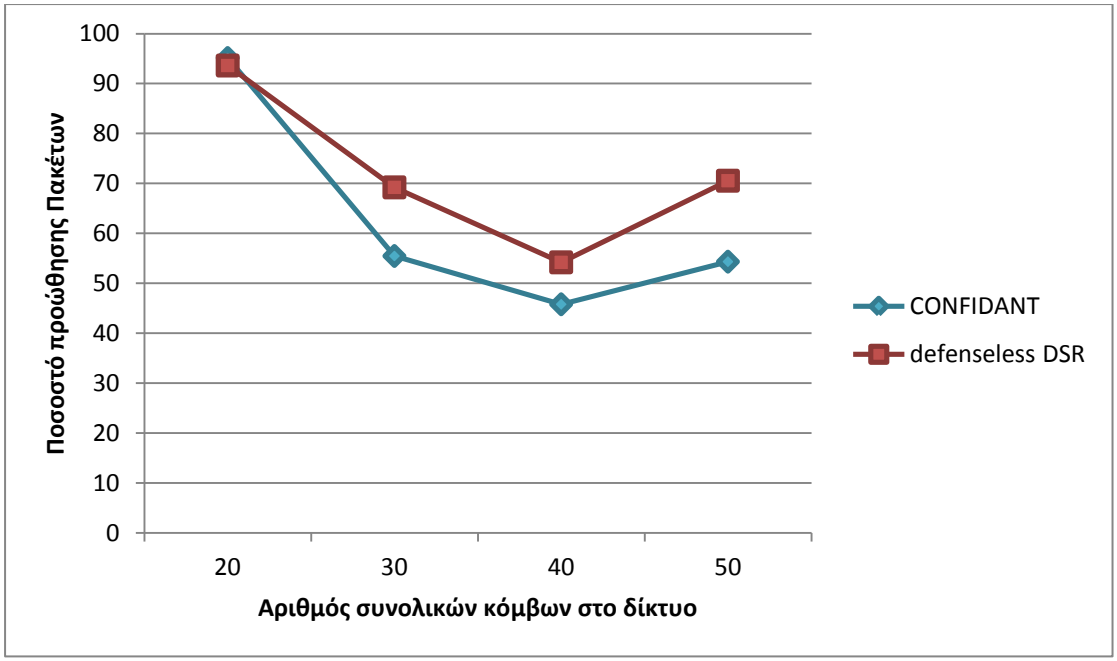
Πίνακας 5.5.

Ακόμα και όταν ο αριθμός των κόμβων του δικτύου αυξηθεί, το CONFIDANT εξακολουθεί να εκπληρώνει το σκοπό του, επιτυγχάνοντας μικρό αριθμό εγωιστικά απορριφθέντων πακέτων σε σχέση με το απλό DSR (γράφημα 5.12.). Ωστόσο, η απόδοσή του συγκριτικά με το απλό DSR είναι μικρή, αφού

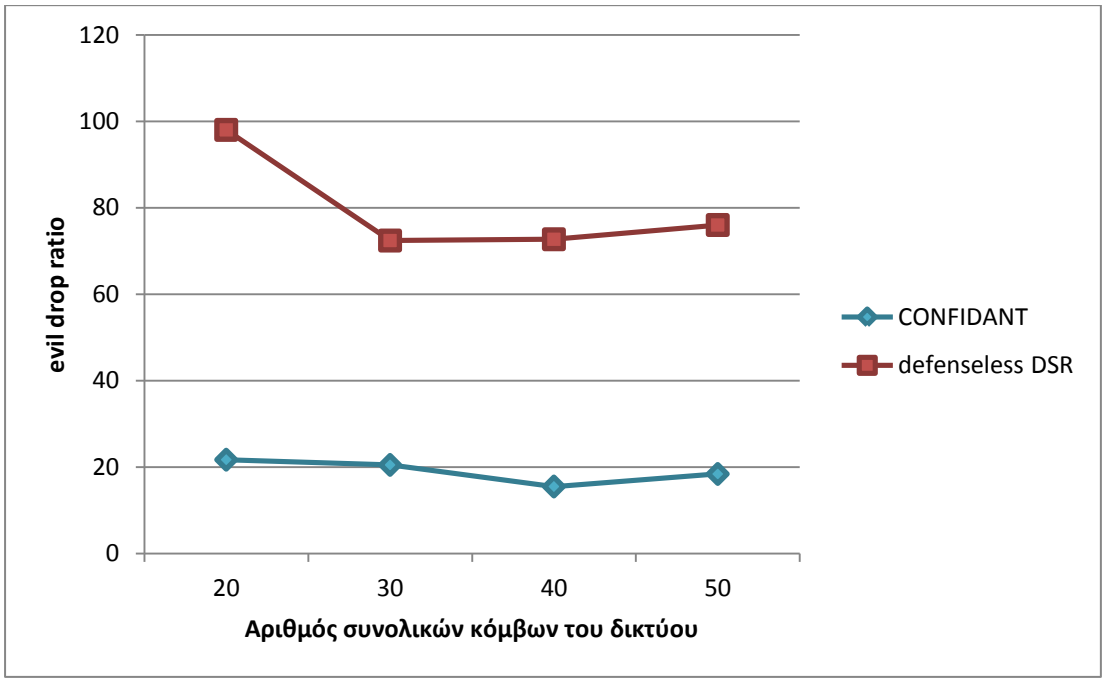
το ποσοστό επιτυχούς προώθησης πακέτων είναι μικρότερο (γράφημα 5.13.) . Καθώς ο αριθμός των κόμβων του δικτύου αυξάνει, το DSR επηρεάζεται λιγότερο από τα εγωιστικά drops , παράλληλα ο μηχανισμός CONFIDANT επίσης μειώνει το ποσοστό του ανάλογα με εκείνο του απλού DSR. Αυτό σημαίνει πως όσο η συνολική κίνηση του δικτύου αυξάνει, η απόδοση του CONFIDANT επηρεάζεται ακόμη περισσότερο από τους παράγοντες του DSR αλλά και την τοπολογία του σεναρίου (γράφημα 5.14.).



Γράφημα 5.12. Αριθμός πακέτων που εγωιστικά απορρίφθηκαν σε σχέση με αριθμούς κόμβων στο δίκτυο



Γράφημα 5.13. Ποσοστό προώθηση πακέτων σε σχέση με αριθμό κόμβων στο δίκτυο



Γράφημα 5.14. Ποσοστό απόρριψης πακέτων από εγωιστικούς κόμβους σε σχέση με αριθμό κόμβων στο δίκτυο

5.3 Συμπεράσματα

Συμπερασματικά, το CONFIDANT είναι ένας μηχανισμός που επιτυχώς αναγνωρίζει τους εγωιστικούς κόμβους και μειώνει αισθητά την απόρριψη

πακέτων λόγω αυτών. Τα πακέτα τύπου ALARM δεν επιβαρύνουν αισθητά το δίκτυο και βοηθούν στην γρηγορότερη εντόπιση της κακής συμπεριφοράς. Η αναγνώριση των εγωιστικών κόμβων στέφεται με επιτυχία και η αποστολή πακέτων προς αυτούς περιορίζεται. Ακόμα και στην περίπτωση που το 100% των κόμβων συμπεριφέρεται εγωιστικά το CONFIDANT είναι πάλι σε θέση να κρατήσει την απόρριψη πακέτων λόγω κακής συμπεριφοράς σε χαμηλά επίπεδα.

Αντίθετα, το CONFIDANT επηρεάζει αρνητικά την λειτουργία του πρωτοκόλλου DSR και το ωθεί να λειτουργεί αρνητικά προς το δίκτυο. Μεγάλος αριθμός πακέτων δρομολόγησης παράγονται από το πρωτόκολλο και πολλά πακέτα απορρίπτονται λόγω έλλειψης διαδρομών στην λίστα Διαδρομών. Όσο το ποσοστό των εγωιστικών κόμβων αυξάνει, τόσο λιγότερα μονοπάτια στη λίστα διαδρομών είναι διαθέσιμα. Ως αποτέλεσμα, το DSR παράγει πολλά πακέτα δρομολόγησης για να βρει καινούργιες διαδρομές ενώ ταυτόχρονα ο χρόνος αποθήκευσης των πακέτων δεδομένων προς προώθηση λήγει.

Γενικά ο μηχανισμός CONFIDANT δεν έχει σταθερή απόδοση. Η συμπεριφορά του επηρεάζεται από πολλούς παράγοντες που συνήθως οφείλονται στην τοπολογία του σεναρίου. Όταν η τοπολογία είναι ωφέλιμη, δηλαδή υπάρχουν περισσότερες εναλλακτικές διαδρομές στο δίκτυο, η παρουσία του CONFIDANT είναι ωφέλιμη, διαφορετικά η απόδοση του δικτύου πέφτει.

Η παρουσία της τιμωρίας των εγωιστικών κόμβων επηρεάζει αρνητικά την απόδοση του δικτύου. Αυτό είναι λογικό εάν θυμηθούμε τις υποθέσεις που κάναμε για την λειτουργία των εγωιστικών κόμβων. Οι εγωιστικοί κόμβοι παραμένουν εγωιστικοί καθ’ όλη τη διάρκεια της προσομοίωσης. Σε φυσιολογικές συνθήκες οι κόμβοι αυτοί θα έπρεπε να σταματούν την κακή συμπεριφορά τους καθώς παρατηρούσαν την αρνητική στάση του δικτύου ως προς αυτούς (μη προώθηση πακέτων προς εγωιστικούς κόμβους, απόρριψη πακέτων που έχουν αποστολέα εγωιστικό κόμβο). Εάν οι εγωιστικοί κόμβοι είχαν την δυνατότητα να σταματήσουν την μη συνεργάσιμη συμπεριφορά τους, τότε η παρουσία του CONFIDANT θα μπορούσε να γίνει ευεργετική προς το δίκτυο αυξάνοντας το ποσοστό επιτυχούς προώθησης πακέτων. Άλλωστε αυτός είναι και ο σκοπός ανάπτυξης μηχανισμών προώθησης συνεργασίας.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1]D. B. Johnson, D. A. Maltz, J. Broch, “DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks” , Addison-Wesley, 2001

[2]Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, Jorjeta Jetcheva.
A Performance Comparison of Multi-Hop Wireless Ad Hoc Nework Routing Protocols.

[3]S. Zhong, J. Chen, Y. R. Yang, “Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks”, INFOCOM, 2003.

[4]L. Buttyan , J.-P. Hubaux , “Nuglets: a Virtual Currency to Stimulate Cooperation in Self-Organized Mobile Ad Hoc Networks” , Switzerland , January 2001

[5]J.-P. Hubaux, J.-Y. Le Boudec, S. Giordano, M. Hamdi, L. Blazevic, L. Buttyan, and M. Vonjovic. “Towards mobile ad-hoc WANs: Terminodes. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Chicago, September 2000

[6]Sonja Buchegger , Jean - Yves Le Boudec, "Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes: Fairness In Dynamic Ad-hoc NeTworks)", MOBIHOC'02 , June 9-11 , 2002

[7]Qi He, Dapeng Wu, Pradeep Khosla “SORI: A Secure and Objective Reputation-based Incentive Scheme for Ad-hoc Networks”, Wireless Communications and Networking Conference (WCNC 2004)

[8] Lennart Conrad M.Sc. Thesis Secure Routing in Mobile Ad Hoc Networks

[9] Shanshan Song M.Sc. Thesis “Dynamic feed-back mechanisms in Trust-Based DSR”, 2005

[10] Kevin Fall, Kannan Varadhan The ns Manual (formerly ns Notes and Documentation)

[11]Shanstry N., Adve R.S. , “Stimulating Cooperative Diversity in Wireless Ad Hoc Networks through Pricing”, IEEE International Conference on Communications, 2006.

[12]M. Jakobsson, J.P. Hubaux, L. Buttyan, “A micropayment scheme encouraging collaboration in multi-hop cellular networks”, Procceedings of Financial Crypto 2003.

[14]Dan Zhang, Ileri O. Mandayam N., “Bandwidth exchange as an incentive for relaying” , 42nd Annual Conference on Information Sciences and Systems, 2008.

[15]P. Michiardi, R. Molva, “Core: A cooperative reputation mechanism to enforce node cooperation in mobile ad hoc networks” , Communications and Multimedia Security Conference (CMS), 2002.

[16]Animesh Kr Trivedi, Rishi Kapoor, Rajan Arora, Sudip Sanyal, “RISM-Reputation Based Intrusion Detection System for Mobile Adhoc Networks” , CODEC’06, 2006.

[17]Sorav Bansal, Mary Baker, “Observation-based Cooperation Enforcement in Adhoc Networks” , arXiv:cs/0307012v2[cs.NI], 2003.

[18]Juan Jos Jaramillo, R. Srikant, “DARWIN: distributed and adaptive reputation mechanism for wireless ad-hoc networks”, 13th ACM international conference on Mobile computing and networking, 2007.

[19]GUO Jianl, LIU Hongwei, DONG Jian, Yang Xiaozong, “HEAD: A Hybrid Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks”, TSINGHUA SCIENCE AND TECHNOLOGY, pp202-207, Volume 12, 2007.

[20]Ze Li, Haiying Shen, “Analysis of A Hybrid Reputation Management System for Mobile Ad hoc Networks”, IEEE, 2009.

[19]MANET : http://el.wikipedia.org/wiki/MANET

[20]Bryan’s NS-2 Dynamic Source Routing FAQ:
http://www.skynet.ie/~bryan/dsr_faq/

[21]DSR in ns-2:
http://www.winlab.rutgers.edu/~zhibinwu/html/DSR_ns2.html#recv

[22]MANET (Mobile Adhoc NETwork):
http://www.saching.com/Article/MANET---Mobile-Adhoc-NETwork--/334

[23] DSRAgent Class Reference: http://www.auto-nomos.de/ns2doku/class_d_s_r_agent.html

[24]Mark Greis' Tutorial for the Network Simulator "ns": http://www.isi.edu/nsnam/ns/tutorial/

[25]C++ Language Tutorial: http://www.cplusplus.com/doc/tutorial/

[26] http://newdata.box.sk/bx/c/