ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Υβριδικοί μηχανισμοί κινήτρων συνεργασίας σε Ασύρματα Αδόμητα Δίκτυα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αρχιμήδης Σ. Χαρισιάδης

Επιβλέπων: Αθανάσιος Δ. Παναγόπουλος

Λέκτορας Ε.Μ.Π.

Αθήνα 2012

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Υβριδικοί μηχανισμοί κινήτρων συνεργασίας σε Ασύρματα Αδόμητα Δίκτυα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αρχιμήδης Σ. Χαρισιάδης

Επιβλέπων: Αθανάσιος Δ. Παναγόπουλος

Λέκτορας Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 27η Αυγούστου 2012.

.....	.....	.....
Αθανάσιος Παναγόπουλος	Φίλιππος Κωνσταντίνου	Ιωάννης Κανελλόπουλος

Αθήνα 2012

.....

Αρχιμήδης Σ. Χαρισιάδης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον κύριο Παναγόπουλο για την ευκαιρία που μου έδωσε να ασχοληθώ με το συγκεκριμένο τομέα των ασυρμάτων επικοινωνιών, καθώς και για τις πολύτιμες συμβουλές και βοήθεια του.

Ιδιαίτερα θερμά θα ήθελα επίσης να ευχαριστήσω τον Δημήτρη Χαρίλα για την συνεχή καθοδήγηση, υποστήριξη και ανεκτίμητη βοήθεια του καθόλη τη διάρκεια περάτωσης της εργασίας. Χωρίς τη συμβολή του η διπλωματική εργασία δεν θα ήταν εφικτή.

Τους ανθρώπους της ζωής μου.

Περίληψη

Ένα ad hoc δίκτυο αποτελεί έναν αποκεντρωμένο τύπο ασύρματου δικτύου. Το βασικό χαρακτηριστικό του είναι ότι δεν βασίζεται σε μια ήδη προϋπάρχουσα και σταθερή υποδομή. Σε ένα τέτοιου είδους δίκτυο όπου μπορεί να μπει και να βγει ο καθένας οφείλουμε να φροντίσουμε για την αντιμετώπιση κακόβουλων και εγωιστών κόμβων, οι οποίοι επιχειρούν να βλάψουν ή να εκμεταλλευτούν τους πόρους του δικτύου. Για την συμμόρφωση αυτών των κόμβων είναι απαραίτητο να αναπτυχθούν κάποιοι μηχανισμοί κινήτρων συνεργασίας, ώστε να τιμωρούνται οι κακόβουλοι κόμβοι και να συνεργάζονται οι εγωιστές κόμβοι με το υπόλοιπο δίκτυο. Τα δύο βασικότερα είδη αυτών των μηχανισμών είναι οι μηχανισμοί φήμης (reputation-based incentive mechanisms) και οι μηχανισμοί πίστωσης (credit-based incentive mechanisms). Βασιζόμενοι σε αυτά τα δύο είδη μηχανισμών δημιουργήθηκαν οι υβριδικοί αλγόριθμοι που συνδυάζουν στοιχεία και από τους δύο.

Σε αυτή την διπλωματική, αναφέρονται εν συντομία τα βασικά χαρακτηριστικά των μηχανισμών φήμης (reputation-based incentive mechanisms) και μηχανισμών πίστωσης (credit-based incentive mechanisms), αλλά το κύριο ενδιαφέρον έγκειται στους υβριδικούς μηχανισμούς. Στο δεύτερο κεφάλαιο γίνεται μια επισκόπηση των ήδη υπαρχόντων υβριδικών μηχανισμών ενώ στην συνέχεια αναπτύσσεται και παρουσιάζεται αναλυτικά ένας νέος προτεινόμενος υβριδικός μηχανισμός. Αυτός ο μηχανισμός είναι ο ICARUS του οποίου η δομή και ακριβή λειτουργία αναλύεται στο τρίτο κεφάλαιο. Εν συνεχεία, γίνεται υλοποίηση του μηχανισμού ICARUS σε γλώσσα Java με στόχο την πραγματοποίηση προσομοίωσης για την μελέτη της αποτελεσματικότητας του αλγορίθμου. Τέλος παρουσιάζονται τα αποτελέσματα του αλγορίθμου μέσω της προσομοίωσης και σχεδιάζονται οι απαραίτητες γραφικές παραστάσεις.

Λέξεις κλειδιά

Ασύρματα αδόμητα δίκτυα, μηχανισμοί κινήτρων συνεργασίας, μηχανισμοί φήμης, μηχανισμοί πίστωσης, υβριδικοί μηχανισμοί, ICARUS, βελτιστοποίηση δικτύου, εγωιστές κόμβοι, κακόβουλοι κόμβοι, προσομοίωση σε Java

Abstract

A wireless ad-hoc network is a decentralized type of wireless network. The network is ad hoc because it does not rely on a pre-existing infrastructure, such as routers in wired networks or access points in managed (infrastructure) wireless networks. In that kind of networks, where anyone can come and go as they please, there ought to be defence mechanisms to tackle selfish and malicious nodes, that could be harmful to the network. Incentive mechanisms provide the essential incentives to nodes for cooperating with each other and dissuading nodes from acting maliciously. The two essential incentive mechanisms are reputation-based and credit-based. Based on those two mechanisms and combining elements from them both, hybrid mechanisms were developed.

The point of interest of this diploma thesis lies in the hybrid mechanisms. In its first part, there is a short presentation of the fundamentals of the reputation and credit based incentive mechanisms. In the second chapter, the two most important hybrid mechanisms are presented in detail. Afterwards, a new hybrid mechanism is developed and its operations are analysed thoroughly. This new hybrid mechanism is called ICARUS and is presented in the third chapter of the diploma thesis. ICARUS is then simulated in Java environment in order to examine the algorithm's efficiency. Lastly, the arithmetic results as well as some graphs of the simulation are presented.

Keywords

Ad-hoc networks, incentive mechanisms, reputation-based mechanisms, credit-based mechanisms, hybrid incentive mechanisms, ICARUS, network's performance improvement, malicious nodes, selfish nodes, Java simulation

Περιεχόμενα

Εισαγωγή....11

1.Μηχανισμοί κινήτρων συνεργασίας....13

1.1Εισαγωγικά στοιχεία....13

1.2 Περιγραφή κυριότερων μηχανισμών....16

1.2.1 CORE (Collaborative Reputation Mechanism)....16

1.2.2 SORI (Secure and Objective Reputation-based Incentive)....17

1.2.3 CONFIDANT (Cooperation of Nodes : Fairness in Dynamic Ad hoc Networks)....17

1.2.4 RISM (semi-distributed Reputation-based Intrusion Detection System)....18

1.2.5 Sprite (Simple Cheat Proof Credit Based System for Mobile Ad hoc Networks)....20

1.2.6 OCEAN (Observation-based Cooperation Enforcement in Ad hoc Networks)....20

1.2.7 DARWIN (Distributed and Adaptive Reputation mechanism for WIreless ad-hoc Networks)....21

1.3 Σύνοψη Πλεονεκτημάτων-Μειονεκτημάτων των μηχανισμών....23

2. Υβριδικοί μηχανισμοί κινήτρων....27

2.1 Εισαγωγικά....27

2.2 Hybrid Reputation Management System for Mobile Ad Hoc Networks....27

2.3 HEAD (A Hybrid Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks)....32

3. Προτεινόμενος υβριδικός μηχανισμός ICARUS....37

3.1 Εισαγωγικά....37

3.2 Μοντελοποίηση δικτύου....38

3.2.1 Στρατηγικές και αποδόσεις....38

3.2.2 Δρομολόγηση....41

3.2.3 Συμπεριφορά εγωιστών κόμβων....42

3.3. Συνεργασία κόμβων....43

3.3.1 Λειτουργία του ICAS....44
3.3.2 Τιμολόγηση....45
3.3.3 Εντοπισμός εγωιστών κόμβων....47
3.3.4 Απομακρυσμένοι κόμβοι....47
4.Υλοποίηση υβριδικού μηχανισμού και προσομοίωση....51
4.1 Στοιχεία Προσομοίωσης....51
4.1.1 Εισαγωγικά....51
4.1.2 Τοπολογία....51
4.2 Υλοποίηση υβριδικού μηχανισμού....52
4.2.1 Σύντομη περιγραφή αλγορίθμου....52
4.2.1.1 Περιγραφή βασικών κλάσεων του αλγορίθμου....52
4.2.1.2 Περιγραφή λειτουργικών κλάσεων του αλγορίθμου....54
4.2.2 Αναλυτική περιγραφή λειτουργίας αλγορίθμου....56
5. Αποτελέσματα προσομοίωσης....65
5.1 Παράμετροι προσομοίωσης....65
5.2 Αριθμητικά αποτελέσματα και γραφικές παραστάσεις....65
5.2.1 Αποτελέσματα και σύγκριση ICARUS - DARWIN....65
5.2.2 Αναλυτική μελέτη του μηχανισμού ICARUS μέσω προσομοίωσης....80
5.3 Συμπεράσματα....88
6. Προτάσεις για μελλοντική έρευνα....89

ΕΙΣΑΓΩΓΗ

Στην σημερινή εποχή των συνεχώς αυξανόμενων τεχνολογικών αναγκών, οι ασύρματες επικοινωνίες έχουν γίνει αναπόσπαστο κομμάτι της καθημερινής ζωής του μέσου ανθρώπου. Παρότι οι κινητές τηλεπικοινωνίες έχουν αναπτυχθεί εδώ και πάρα πολύ καιρό, μόλις την τελευταία δεκαετία έκαναν την είσοδο τους στην καθημερινότητα του κόσμου. Ωστόσο λίγοι θα μπορούσαν να υπολογίσουν τους ρυθμούς εξάπλωσης τους.

Μέσα σε αυτά τα χρόνια, η έναρξη αυτού του τεχνολογικού χειμάρρου ξεκίνησε με την εισαγωγή των συστημάτων 2ης γενιάς που προσέφεραν ικανοποιητική υπηρεσία φωνής καθώς και επαρκή χρησιμοποίηση του φάσματος, με κύριο πρότυπο το GSM (Global System for Mobile Communications). Κάποια χρόνια αργότερα άρχισαν να αναπτύσσονται τα δίκτυα 3ης γενιάς (3G) τα οποία κατέκλυσαν το όλο και πιο απαιτητικό κοινωνικό σύνολο. Παρότι κατάφεραν να καλύψουν τις ανάγκες των καταναλωτών, οι ανάγκες αυτών συνέχισαν να αυξάνονται με δραματικούς ρυθμούς και οδήγησαν στην ανάγκη ανάπτυξης των δικτύων 4ης γενιάς. Αυτά τα δίκτυα βρίσκονται ακόμα σε ερευνητικό επίπεδο αλλά είμαστε σε θέση να γνωρίζουμε κάποια χαρακτηριστικά τους. Αυτό που απασχολεί εμάς είναι η εισαγωγή των ad hoc δικτύων στην δομή των μελλοντικών δικτύων 4ης γενιάς (4G).

Ένα ad hoc δίκτυο αποτελεί έναν αποκεντρωμένο τύπο ασύρματου δικτύου. Το βασικό χαρακτηριστικό ενός τέτοιου δικτύου είναι το γεγονός ότι δεν βασίζεται σε μια ήδη προϋπάρχουσα και σταθερή υποδομή. Δηλαδή δεν υπάρχει ανάγκη για δρομολογητές και κεντρικά access points δεδομένου πως ο κάθε κόμβος έχει την δυνατότητα για άμεση επικοινωνία με οποιονδήποτε άλλο κόμβο του δικτύου, καθώς ο ίδιος ο κόμβος λειτουργεί σαν δρομολογητής.[1][2] Αρχικά αυτού του είδους τα δίκτυα βρήκαν αρκετές στρατιωτικές εφαρμογές. Ωστόσο με το πέρασμα του χρόνου και τις όλο και περισσότερες ανάγκες των σύγχρονων κοινωνιών, άρχισε να γίνεται αντιληπτό το εύρος δυνατοτήτων που παρέχουν τα ad hoc δίκτυα. Το μεγάλο εύρος κινητικότητας των κόμβων, αυτών των δικτύων καθώς και η δυνατότητα των χρηστών των να μπαίνουν και να βγαίνουν από αυτά χωρίς να έχουν ανάγκη από διαφορετικές οντότητες, κάνουν την ιδέα για εφαρμογή τους στην σύγχρονη κοινωνία ιδιαίτερα θελκτική.

Ωστόσο για την αποτελεσματική εφαρμογή αυτού του ιδιαίτερου είδος δικτύου, πρέπει να έχουμε τη δυνατότητα να εγγυηθούμε για την ποιότητα υπηρεσιών που παρέχεται. Σε ένα τέτοιου είδους δίκτυο όπου μπορεί να μπει και να βγει ο καθένας οφείλουμε να φροντίσουμε για την αντιμετώπιση κακόβουλων και εγωιστών κόμβων (η ακριβή έννοια των οποίων θα αναλυθεί στο πρώτο κεφάλαιο), οι οποίοι επιχειρούν να βλάψουν ή να εκμεταλλευτούν τους πόρους του δικτύου. Για την συμμόρφωση αυτών των κόμβων είναι απαραίτητο να αναπτυχθούν κάποιοι μηχανισμοί κινήτρων συνεργασίας, ώστε να τιμωρούνται οι κακόβουλοι κόμβοι και να συνεργάζονται οι εγωιστές κόμβοι με το υπόλοιπο δίκτυο.

Στην πορεία αυτής της διπλωματικής θα γίνει παρουσίαση των μηχανισμών κινήτρων συνεργασίας, της βάσης λειτουργίας τους και στην συνέχεια μια εισαγωγή

στους υβριδικούς μηχανισμούς κινήτρων. Το κύριο μέρος της εργασίας αποτελείται από την ανάπτυξη ενός τέτοιου υβριδικού μηχανισμού/αλγορίθμου κινήτρων συνεργασίας. Στα τελευταία κεφάλαια ακολουθεί η επεξήγηση της λειτουργίας του και τέλος η μελέτη της αποτελεσματικότητας του μέσω προσομοίωσης.

Βιβλιογραφία

[1] http://en.wikipedia.org/wiki/Ad-hoc_network.
[2] http://el.wikipedia.org/wiki/Ασύρματο_δίκτυο

1. ΜΗΧΑΝΙΣΜΟΙ ΚΙΝΗΤΡΩΝ ΣΥΝΕΡΓΑΣΙΑΣ

1.1 Εισαγωγικά στοιχεία

Για την διασφάλιση της ποιότητας υπηρεσίας σε ένα δίκτυο οφείλουμε, όπως αναφέρθηκε παραπάνω, να ωθούμε όλους τους κόμβους του δικτύου σε συνεργατική συμπεριφορά καθώς και να τιμωρούμε αυτούς που βλάπτουν και εκμεταλλεύονται τους πόρους του δικτύου. Πιο συγκεκριμένα αυτή η επιβλαβής συμπεριφορά απέναντι στο δίκτυο μπορεί να διαχωριστεί σε δύο βασικές κατηγορίες.

Πιο συγκεκριμένα σε ένα ad hoc δίκτυο μπορούμε να συναντήσουμε εγωιστές και κακόβουλους κόμβους. Κακόβουλοι χαρακτηρίζονται οι κόμβοι που εσκεμμένα επιτίθενται στο δίκτυο με σκοπό να το βλάψουν ή απλά είναι ελαττωματικοί στην λειτουργία τους. Κάποια παραδείγματα τέτοια συμπεριφοράς είναι η εσκεμμένη απόρριψη πακέτων, κρυφή αλλαγή πινάκων δρομολόγησης, ασυνήθιστα συχνή αλλαγή πινάκων δρομολόγησης, αλλοίωση στοιχείων κίνησης του κόμβου και εξαπάτηση υποδυόμενος άλλο κόμβο. [3][4] Εγωιστές κόμβοι από την άλλη, είναι αυτοί που έχουν σκοπό να μεγιστοποιήσουν το όφελος τους σε βάρος του υπολοίπου δικτύου. Για παράδειγμα, ένας τέτοιος κόμβος προσπαθεί να βρίσκει τρόπους για να στέλνει τα πακέτα του χωρίς ο ίδιος να προωθεί πακέτα άλλων κόμβων.

Για την αντιμετώπιση ακριβώς αυτού του είδους της συμπεριφοράς, υπάρχει ανάγκη για τους μηχανισμούς κινήτρων συνεργασίας. Η βασική διαφοροποίηση των μηχανισμών αυτών γίνεται βάση των κριτηρίων στα οποία βασίζονται. Έτσι υπάρχουν δύο βασικοί τύποι μηχανισμών κινήτρων συνεργασίας.

- μηχανισμοί φήμης (reputation-based incentive mechanisms)
- μηχανισμοί πίστωσης (credit-based incentive mechanisms)

Αρχικά θα γίνει μια σύντομη περιγραφή του τρόπου λειτουργίας των μηχανισμών πίστωσης. Η βασική ιδέα αυτού του μηχανισμού βασίζεται στην χρήση κάποιων θεωρητικών χρηματικών μονάδων (credits). Αυτά τα credits χρησιμοποιούνται από τους κόμβους για την πληρωμή για την αποστολή ενός πακέτου, αλλά δίνονται και σε όποιους κόμβους δρουν συνεργατικά και προωθούν πακέτα τα οποία δρομολογούνται μέσω αυτών. Βάση αυτής της λογικής είναι προφανές ότι κάποιος κόμβος που δεν προωθεί πακέτα δεν θα έχει την "οικονομική" δυνατότητα να αποστείλει πακέτα καθώς δεν θα έχει κερδίσει credits.[1] Αξίζει να σημειωθεί ότι ο μετρητής των credits παρουσιάζει μια ιδιομορφία. Πιο συγκεκριμένα, αυτός ο μετρητής αυξάνεται για κάθε πακέτο που προωθεί ένας κόμβος αλλά σε περίπτωση που γίνεται αποστολή, ο αποστολέας πληρώνει βάσει του αριθμού των hops (αριθμός των βημάτων που πρέπει να κάνει ένα πακέτο περνώντας από ενδιάμεσους κόμβους ώστε να φτάσει στον τελικό του προορισμό) που κάνει το πακέτο συνολικά. Με αυτό τον τρόπο διασφαλίζεται η ομαλή λειτουργία του δικτύου καθότι ο κάθε κόμβος πάντα αναγκάζεται να προωθεί περισσότερα πακέτα από όσα στέλνει.

Όσων αφορά τους μηχανισμούς κινήτρων συνεργασίας που λειτουργούν βάσει φήμης, μπορούμε αρχικά να κάνουμε ένα βασικό διαχωρισμό. Στους κεντροποιημένους μηχανισμούς και τους αποκεντροποιημένους. Κεντροποιημένοι χαρακτηρίζονται αυτοί στους οποίους η φήμη υπολογίζεται από μια κεντρική οντότητα που υπάρχει στο δίκτυο, ενώ αντιστοίχως αποκεντροποιημένοι αυτοί που ευρίσκουν την φήμη κάθε κόμβου σε κάθε κόμβο ξεχωριστά και εν συνεχεία οι πληροφορίες αυτές μοιράζονται στο υπόλοιπο δίκτυο. Η φήμη είναι ένα δυναμικό μέγεθος το οποίο ευρίσκεται συγκεντρώνοντας τις παρατηρήσεις όλων των μελών ενός δικτύου και συνδυάζοντας τα. Βάσει της τιμής της φήμης που έχει ο κάθε κόμβος μπορεί να χαρακτηριστεί ως κακόβουλος/εγωιστής ή όχι.

Σε αυτό το σημείο αξίζει να παρατηρηθεί ένα πρόβλημα που παρουσιάζεται στην μελέτη συμπεριφοράς των κόμβων. Κάποιες φορές υπάρχει ενδεχόμενο να γίνει λάθος παρατήρηση για κάποιο κόμβο εξαιτίας ενός τυχαίου γεγονός όπως η σύγκρουση ενός πακέτου, και έτσι να εκτιμηθεί ως κακόβουλος για παράδειγμα ένας συνεργατικός κόμβος. Θύματα αυτού του προβλήματος μπορούν επίσης να βρεθούν κάποιοι κόμβοι που βρίσκονται σε δυσμενή θέση, όπως απομακρυσμένοι κόμβοι. Σε αυτή την περίπτωση ένας απομακρυσμένος κόμβος μπορεί εύκολα να θεωρηθεί από το σύστημα ως εγωιστής επειδή δεν προωθεί αρκετά πακέτα, ενώ η αλήθεια πίσω από το γεγονός αυτό είναι ότι δεν έχει την δυνατότητα να προωθήσει αρκετά πακέτα εξαιτίας της δυσμενής του θέσης. Αυτό όπως είναι λογικό υποβαθμίζει την ποιότητα υπηρεσίας του δικτύου και για αυτό το λόγο κάποιοι αλγόριθμοι φροντίζουν να λαμβάνουν υπόψη τους τέτοια σενάρια.

Παρακάτω παρατίθεται ένας πίνακας όπου παρουσιάζονται συνοπτικά τα πλεονεκτήματα και μειονεκτήματα των δύο τύπων μηχανισμών κινήτρων συνεργασίας.

	Πλεονεκτήματα	Μειονεκτήματα
Μηχανισμοί Φήμης	• Αντικειμενική αξιολόγηση καθότι γίνεται συλλογή πληροφοριών από μεγάλο δείγμα • Ανθεκτικοί στην διάδοση ψευδών πληροφοριών από μια μικρή ομάδα κακόβουλων κόμβων	• Κόστος σε πόρους του δικτύου • Ευπαθείς στην διάδοση ψευδών πληροφοριών από μια μεγάλη ομάδα κακόβουλων κόμβων • Ανάγκη για ταυτοποίηση κόμβων(διαδικασία που δεν είναι ιδιαίτερα απλοϊκή)
Μηχανισμοί πίστωσης	• Ιδιαίτερα αποτελεσματικοί στην προώθηση συνεργατικής συμπεριφοράς. • Χρήσιμοι σε multi-hop δίκτυα που η δράση και η αμοιβή δεν πραγματοποιούνται ταυτόχρονα	• Κόστος σε πόρους του δικτύου • Αδυναμία εύρεσης κακόβουλων και εγωιστών κόμβων • Πολυπλοκότητα του αλγορίθμου χρέωσης κόμβων • Ατυχής διαμοιρασμός credits για τους ακριανούς κόμβους • Ανάγκη για μηχανισμό ανακατανομής πλούτου

1.2 Περιγραφή κυριότερων μηχανισμών

Σε αυτό το σημείο θα γίνει μια σύντομη περιγραφή των κυριότερων μηχανισμών κινήτρων συνεργασίας, παραθέτοντας την λογική λειτουργίας τους καθώς και τα πλεονεκτήματα αλλά και μειονεκτήματα τους.

1.2.1 CORE (Collaborative Reputation Mechanism)

Ο CORE αποτελεί έναν από τους γνωστότερους μηχανισμούς φήμης. Για τον υπολογισμό του πίνακα φήμης επί του οποίου βασίζεται ο αλγόριθμος, ο κάθε κόμβος χρησιμοποιεί στοιχεία που συνέλλεξε από δική του εμπειρία καθώς και παρατηρήσεις από τους υπόλοιπους κόμβους του δικτύου. Ο συγκεκριμένος αλγόριθμος χρησιμοποιώντας τον μετρητή της φήμης ωθεί τους εγωιστές κόμβους να συνεργαστούν με το υπόλοιπο δίκτυο ώστε να βελτιώσει την ποιότητα υπηρεσιών του δικτύου. Δηλαδή σε ένα τέτοιο σύστημα οι κόμβοι που διατηρούν καλή φήμη έχουν την δυνατότητα να εκμεταλλεύονται τους πόρους το δικτύου ενώ οι κόμβοι που δρουν εγωιστικά απομονώνονται από αυτό. Επίσης φροντίζει και για την προστασία του δικτύου από κακόβουλους κόμβους. Για παράδειγμα, σε περίπτωση που ένας κακόβουλος κόμβος διασπείρει αρνητικές φήμες στους υπόλοιπους κόμβους του δικτύου, υπάρχει η δυνατότητα μέσω ακολουθιακών διαδικασιών να απομονώσει τον κόμβο αυτό.

Η συνολική φήμη που διατηρεί κάθε κόμβος υπολογίζεται από τρεις επιμέρους φήμες οι οποίες είναι οι : Υποκειμενική φήμη (Subjective reputation), Έμμεση φήμη (Indirect reputation) και Λειτουργική Φήμη (Functional reputation).

Υποκειμενική φήμη είναι αυτή που συλλέγει κάθε κόμβος από την δικιά του εμπειρία με τους γειτονικούς του κόμβους. Όταν τα στοιχεία που έχει συλλέξει ένας κόμβος δεν είναι επαρκή για την εξαγωγή συμπεράσματος η τιμή της φήμης παίρνει την τιμή 0. Επίσης αυτού του είδους η φήμη δίνει μεγαλύτερη βαρύτητα σε στοιχεία που είχε μαζέψει στο παρελθόν για ένα κόμβο.

Έμμεση φήμη είναι η αυτή που ο κόμβος λαμβάνει από το υπόλοιπο δίκτυο (δηλαδή άλλους κόμβους) για τον κόμβο που τον ενδιαφέρει να εξάγει ένα συμπέρασμα.

Τέλος η Λειτουργική φήμη αποτελεί συνδυασμός των δύο προηγουμένων για διαφορετικές όμως διαδικασίες.

1.2.2 SORI (Secure and Objective Reputation-based Incentive)

Ο SORI αποτελεί επίσης έναν μηχανισμό φήμης. Σε αντίθεση με τον προηγούμενο μηχανισμό που μελετήσαμε στον SORI η φήμη κάθε κόμβου διαδίδεται μονάχα στους γείτονες του και έτσι εξοικονομούμε πόρους του δικτύου. Η φήμη προκύπτει βάσει αντικειμενικών μετρήσεων και η διάδοση αυτής γίνεται από μια διάταξη αυθεντικότητας, που βασίζεται σε μονομερή hash tables. Με αυτό τον τρόπο διασφαλίζεται η αποτελεσματική διάδοση της φήμης μέσα στο δίκτυο. Ο SORI μπορεί να διασφαλίσει την αποτελεσματική δρομολόγηση εντός του δικτύου.

Όσων αφορά τα συστήματα για την προστασία του δικτύου από κακόβουλους και εγωιστές κόμβους, το SORI στοχεύει κυρίως στην καταπολέμηση των εγωιστών κόμβων. Πιο συγκεκριμένα, γίνεται η παραδοχή ότι το δίκτυο έχει μόνο εγωιστές κόμβους και όχι κακόβουλους, γεγονός το οποίο ενισχύεται από το γεγονός ότι αυτό ο μηχανισμός θεωρεί δεδομένο ότι δεν γίνονται αλλαγές ταυτότητας στο δίκτυο(συμπεριφορά η οποία χαρακτηρίζει τους κακόβουλους κόμβους).

Η λογική λειτουργίας του SORI βασίζεται σε τρία βασικά βήματα, παρακολούθηση συμπεριφοράς, διαμοιρασμός αυτής της πληροφορίας στους γείτονες, εύρεση φήμης και τέλος τιμωρία των εγωιστών κόμβων. Η διάχυση της πληροφορίας γίνεται έχοντας κάθε κόμβο να ακούει για πακέτα που στέλνουν οι γείτονες του και προορίζονται προς αυτόν.

1.2.3 CONFIDANT (Cooperation of Nodes : Fairness in Dynamic Ad hoc Networks)

Ο CONFIDANT είναι επίσης ένας μηχανισμός φήμης ο οποίος στηρίζει την λειτουργία του σε 4 υποσυστήματα. Αυτά είναι το σύστημα παρακολούθησης(monitor), φήμης(reputation system), διαχείρισης διαδρομών(path manager) και διαχείρισης εμπιστοσύνης(trust manager). Πριν επεξηγηθεί το καθένα από αυτά, θα αναφερθεί εν συντομία ο τρόπος με τον οποίο λειτουργεί το σύστημα αυτό. Πιο συγκεκριμένα, ο κάθε κόμβος παρακολουθεί τους γειτονικούς του κόμβους και φροντίζει για την αποφυγή επικίνδυνων δραστηριοτήτων από κάποιο κακόβουλο κόμβο. Αυτές τις πληροφορίες που συγκεντρώνει ο κάθε κόμβος φροντίζει να τις μοιράσει σε κάποιους φιλικούς του κόμβους για την διασφάλιση της ασφάλειας ολόκληρου του δικτύου.

Εν συνεχεία, θα αναλυθεί η λειτουργία του μηχανισμού CONFIDANT όπως αυτή προκύπτει από τα επιμέρους υποσυστήματα του μηχανισμού.

Ο έλεγχος των μεταδόσεων των γειτονικών κόμβων από κάποιο κόμβο αποτελεί βέβαια λειτουργία του συστήματος παρακολούθησης. Επίσης για την εύρεση των επικίνδυνων κόμβων κάθε κόμβος έχει την δυνατότητα να δημιουργήσει ένα αντίγραφο του πακέτου που έστειλε και εν συνεχεία να ελέγχει εάν το ίδιο πακέτο έφτασε στο επόμενο hop της

διαδρομής προς τον τελικό προορισμό. Με αυτό τον τρόπο υπάρχει η δυνατότητα εύρεσης επικίνδυνων κόμβων που παραποιούν πληροφορίες του δικτύου.

Το σύστημα φήμης είναι υπεύθυνο για τον υπολογισμό της φήμης κάθε κόμβου. Κάθε κόμβος που ανήκει στο δίκτυο διατηρεί λίστες φήμης για τους υπολοίπους καθώς και ¨μαύρες¨ λίστες στην οποίες περιέχονται οι αναξιόπιστοι κόμβοι του δικτύου. Όλες οι λίστες ανταλλάσσονται με τους γειτονικούς κόμβους του καθενός εκτός και εάν κάποιος γείτονας κόμβος ανήκει στην "μαύρη" λίστα. Από την άλλη για να υπάρχει η δυνατότητα επανένταξης στο δίκτυο ενός πρώην αναξιόπιστου κόμβου που πλέον λειτουργεί συνεργατικά, οι λίστες που περιλαμβάνουν τους κακόβουλους κόμβους ανανεώνονται συνέχεια.

Το σύστημα διαχείρισης διαδρομών φροντίζει για την ανανέωση των διαδρομών προς τους κόμβους του δικτύου. Πιο συγκεκριμένα, κάθε κόμβος παρακολουθεί τους γείτονες του για οποιαδήποτε επικίνδυνη δραστηριότητα. Εάν κάτι τέτοιο εντοπιστεί ο κόμβος φροντίζει να μειώσει την φήμη του κόμβου που προκάλεσε το γεγονός και μειώνοντας την τιμή της φήμης, εάν κάποια στιγμή πέσει κάτω από ένα κατώφλι ο κόμβος καταγγέλλεται στον διαχειριστή διαδρομών ο οποίος αφαιρεί τον επικίνδυνο κόμβο από όλες τις διαδρομές του δικτύου και ο κόμβος στέλνει μήνυμα ALARM στους φιλικούς του κόμβους.

Τέλος θα αναφέρουμε τις λειτουργίες του συστήματος διαχείρισης εμπιστοσύνης. Ένα ιδιαίτερο χαρακτηριστικό του CONFIDANT είναι η αποστολή μηνυμάτων ALARM για ενημέρωση ύπαρξης ενός κακόβουλου κόμβου. Έτσι κάθε κόμβος στέλνει μήνυμα ALARM σε μία λίστα φίλων που διατηρεί σε περίπτωση που από προσωπικές παρατηρήσεις εντοπίσει κάποιο κακόβουλο κόμβο ή λάβει μήνυμα ALARM από κάποιο άλλο έμπιστο κόμβο. Έμπιστος χαρακτηρίζεται ένας κόμβος από έναν μηχανισμό διαχείρισης εμπιστοσύνης που ονομάζεται Pretty Good Privacy (PGP) και διαθέτει τέσσερα επίπεδα εμπιστοσύνης. Άγνωστος κόμβος(unknown), χωρίς σχόλια(none), οριακά αποδεκτός(marginal) και πλήρως αποδεκτός(complete).

1.2.4 RISM (semi-distributed Reputation-based Intrusion Detection System)

Το RISM αποτελεί ένα σύστημα φήμης με ημικατανεμημένο χαρακτήρα, δηλαδή αυτό σημαίνει ότι οι παρατηρήσεις που συγκεντρώνονται σε ένα κόμβο δεν παραμένουν σε αυτόν αλλά ούτε διαμοιράζονται στο υπόλοιπο σύστημα άμεσα. Το σύστημα λειτουργεί κρατώντας στην μνήμη του την κίνηση και άλλα στοιχεία του συστήματος. Όπως ο CONFIDANT έτσι και αυτό ο μηχανισμός αποτελείται από τέσσερα υποσυστήματα, την παρακολούθηση, το σύστημα φήμης, την διαχείριση διαδρομών και την λύτρωση και υποχώρηση.

Το σύστημα παρακολούθησης είναι υπεύθυνο για τον εντοπισμό των κόμβων που παρουσιάζουν ύποπτη δραστηριότητα χρησιμοποιώντας τα PACKs (Passive Acknowledgements). Σε αυτό τον μηχανισμό ο κάθε κόμβος παρακολουθεί όλα τα

μηνύματα που στέλνουν οι γειτονικοί του κόμβοι και τα καταγράφει. Μετά από ένα συγκεκριμένο χρονικό διάστημα ελέγχει αν έλαβε PACKs για όλα αυτά τα πακέτα, αν κάτι τέτοιο δεν συνέβη ενημερώνει το σύστημα φήμης για τον ύποπτο κόμβο.

Το σύστημα φήμης υπολογίζει και καταχωρεί την τιμή της φήμης κάθε κόμβου του δικτύου και έχει τιμή από 0 μέχρι MaliciousThreshold. Η τιμή της φήμης για έναν κόμβο μπορεί να αλλάξει από παρατηρήσεις κάποιου κόμβου, μήνυμα συναγερμού για το πρόσωπο του ή τέλος από την λίστα αποφυγής που έχει προστεθεί στην επικεφαλίδα του rreq/rrep. Η τιμή της φήμης όπως ειπώθηκε και προηγούμενα ανανεώνεται ανά περιόδους και η πληροφορία φήμης αποστέλλεται στο δίκτυο με την βοήθεια των λιστών φήμης. Σύμφωνα με την φήμη που έχει κάθε κόμβος χαρακτηρίζεται φυσιολογικός, ύποπτος και κακόβουλος. Η αρχική φήμη κάθε κόμβου έχει την τιμή 0. Όταν ευρεθεί ένα κακόβουλος κόμβος αρχικά ενημερώνονται όλοι οι γείτονες του και στην συνέχεια ενημερώνεται το υπόλοιπο δίκτυο με την αποστολή της λίστας αποφυγής που τοποθετείται στην επικεφαλίδα των πακέτων που οδεύουν προς τους κόμβους. Οι κόμβοι που λαμβάνουν αυτή την λίστα ανανεώνουν τις φήμες των κόμβων που υπάρχουν σε αυτή. Η ιδιομορφία του συστήματος είναι πως ένας κόμβος δεν μπορεί να χαρακτηριστεί ως κακόβουλος παρά μόνο με προσωπική παρατήρηση από κάποιον κόμβο. Αυτό σημαίνει ότι μέσω των λιστών φήμης καθώς και των μηνυμάτων συναγερμού ένας κόμβος μπορεί στην χειρότερη περίπτωση να χαρακτηριστεί ως ύποπτος, για να μεταπηδήσει όμως στην κακόβουλη φήμη πρέπει να ελέγχει με προσωπική εμπειρία αν πράγματι είναι κακόβουλος. Κάτι το οποίο πραγματοποιείται εύκολα στέλνοντας στον εν λόγω κόμβο ένα πακέτο και παρατηρώντας εάν θα το απορρίψει ή όχι. Η παραπάνω διαδικασία ονομάζεται τεστ κρούσης και σε περίπτωση που ο κόμβος απορρίψει εν τέλει το πακέτο χαρακτηρίζεται ως κακόβουλος ενώ εάν το προωθήσει κανονικά παίρνει θετική αποτίμηση. Όπως ο προηγούμενος μηχανισμός έτσι και εδώ λαμβάνονται κάποια μέτρα για την επανένταξη κόμβων που αρχίζουν να παρουσιάζουν συνεργατική συμπεριφορά ενώ είναι κακόβουλοι, αυτό σημαίνει ότι αν για ένα χρονικό διάστημα ο κόμβος δεν έχει παρουσιάσει ύποπτη δραστηριότητα η φήμη του βελτιώνεται ελαφρώς και πηγαίνει στην μέση της κατηγορίας του υπόπτου.

Στην διαχείριση διαδρομών όπως και στον CONFIDANT, μια διαδρομή μπορεί να αλλάξει εάν ευρεθεί ένας κακόβουλος κόμβος ο οποίος και θα απομονωθεί ή εάν επιστρέψει ένας κακόβουλος κόμβος στο δίκτυο.

1.2.5 Sprite (Simple Cheat Proof Credit Based System for Mobile Ad hoc Networks)

Το Sprite αποτελεί ένα σύστημα πίστωσης το οποίο αποτελείται από τους κινητούς κόμβους όπως σε κάθε ad hoc δίκτυο και μια κεντρική οντότητα η οποία δρα ως υπηρεσία εξουσιοδότησης πιστώσεων(Credit Clearance Service - CCS). Η λογική με την οποία στήνεται το οικονομικό σύστημα πίστωσης έχει ως εξής. Κάθε κόμβος που προωθεί ένα πακέτο κερδίζει πόντους για τις υπηρεσίες του προς το δίκτυο ενώ όταν κάποιος αποφάσισε να στείλει ένα πακέτο αναγκάζεται να πληρώσει πόντους, με αποτέλεσμα να μην μπορεί να στείλει μήνυμα ένας κακόβουλος κόμβος ο οποίος δεν έχει προωθήσει κανένα πακέτο. Επίσης υπάρχει η δυνατότητα να κερδίσει ένας κόμβος πόντους πληρώνοντας σε πραγματικά χρήματα, αλλά η κύρια πηγή πόντων αποτελεί η προώθηση πακέτων. Με αυτό τον τρόπο, το σύστημα αυτό ωθεί τους χρήστες του σε συνεργατική συμπεριφορά αλλά και αποτρέπει τα άχρηστα και κακόβουλα μηνύματα.

1.2.6 OCEAN (Observation-based Cooperation Enforcement in Ad hoc Networks)

Το OCEAN ξεχωρίζει από ότι είδαμε προηγούμενα καθότι βασίζεται αποκλειστικά στην παρακολούθηση των γειτονικών κόμβων, χωρίς να γίνεται κάποια μετάδοση πληροφορίας ανάμεσα στους κόμβους.

Πιο συγκεκριμένα ο τρόπος με τον οποίο λειτουργεί αυτός ο μηχανισμός χωρίζεται σε κάποιες βασικές διαδικασίες και έχει ως εξής[11]:

Αρχικά, έχουμε την παρακολούθηση των γειτόνων ενός κόμβου. Αναλυτικότερα, ο κάθε κόμβος στέλνει ένα πακέτο στον γειτονικό του κόμβου του οποίου αποθηκεύει το checksum. Εν συνεχεία αναμένει για κάποιο χρονικό διάστημα και εάν ο γειτονικός κόμβος δεν στείλει το πακέτο, καταγράφεται το αρνητικό συμβάν και διαγράφει το checksum από τον buffer. Ομοίως εάν αποστείλει ο γειτονικός κόμβος το πακέτο γίνεται έλεγχος για το ταίριασμα του checksum, καταγράφεται το θετικό συμβάν και τέλος το checksum διαγράφεται από τον buffer.

Στην συνέχεια έχουμε την βαθμολόγηση των διαδρομών. Έχοντας συγκεντρώσει θετικά και αρνητικά συμβάντα από την παρακολούθηση των γειτονικών κόμβων ο κόμβος αυξάνει ή μειώνει τον μετρητή κάθε γείτονα. Εάν ο μετρητής αυτός πέσει κάτω από κάποια τιμή για έναν γειτονικό κόμβο αυτός ο κόμβος τοποθετείται σε μια ελαττωματική λίστα. Τέλος οι διαδρομές που ευρίσκει ο κόμβος χαρακτηρίζονται ως καλές ή κακές ανάλογα με το εάν υπάρχει ένας ελαττωματικός κόμβος ή όχι στην εκάστοτε διαδρομή. Η βαθμολογία λειτουργεί ως εξής: Για το θετικό συμβάν δίνεται ένας βαθμός ενώ για το αρνητικό συμβάν αφαιρούνται δύο. Η αρχική βαθμολογία κάθε κόμβου είναι 0 ενώ το ελαττωματικό κατώφλι είναι το -40.

Η δρομολόγηση διαδρομών βασισμένη στη βαθμολογία, εφαρμόζει τις πληροφορίες που συλλέχθηκαν κατά την παρακολούθηση των γειτόνων για τον σχεδιασμό της διαδρομής. Συγκεκριμένα κάθε κόμβος που λαμβάνει ένα πακέτο προς προώθηση, ελέγχει αν ο επόμενος κόμβος στην διαδρομή ανήκει στην ελαττωματική λίστα, εάν αυτό ισχύει τότε το πακέτο απορρίπτεται. Ο κόμβος τότε επαναμεταδίδει το πακέτο(από άλλη διαδρομή προφανώς) και απαντάει στον αποστολέα με ένα DSR route-reply. Με αυτό τον τρόπο ο κάθε κόμβος από τον οποίο περνάει ένα πακέτο καταλήγει τελικά να στέλνει το πακέτο σε φιλικούς προς αυτών κόμβους και να αποφεύγει τους ελαττωματικούς.

Τέλος υπάρχει και ο μηχανισμός δεύτερης ευκαιρίας ο οποίος δίνει την δυνατότητα σε πρώην ελαττωματικούς που πλέον συνεργάζονται συνεργατικά να ξαναενταχθούν στο δίκτυο. Όπως και στους άλλους μηχανισμούς που εφάρμοζαν το συγκεκριμένο σύστημα, ύστερα από ένα χρονικό διάστημα ο κόμβος αφαιρείται από την ελαττωματική λίστα χωρίς όμως να μηδενίζεται κιόλας ο μετρητής βαθμολογίας του.

1.2.7 DARWIN (Distributed and Adaptive Reputation mechanism for WIreless ad-hoc Networks)

Το DARWIN βασίζει την λειτουργία του στην μέθοδο Contrite Tit For Tat (CTFT). Η λογική λειτουργίας αυτής της μεθόδου είναι πως ένας κόμβος που έκανε λάθος θα πρέπει να το διορθώσει παρά να απομονωθεί από το δίκτυο. Πιο συγκεκριμένα κάθε κόμβος βρίσκεται σε καλή θέση όσο συνεργάζεται με το δίκτυο και ακολουθεί τις υποδείξεις της στρατηγικής CTFT. Εάν κάποιος βρίσκεται σε κακή θέση ή ο αντίπαλος του βρίσκεται σε καλή θέση θα πρέπει να συνεργαστεί.

Βάση των παραπάνω ορίζεται η πιθανότητα απόρριψης στο DARWIN.

$$\tilde{p}_{iDARWIN}^{(k)} = \left[\gamma * (q_{-i}^{(k-1)} - q_i^{(k-1)})\right]_0^1 \quad for\ k \geq 0 \qquad (1.1)$$

$$q_i^{(k)} = \begin{cases} [\hat{p}_i^{(k)} - \tilde{p}_{iDARWIN}^{(k)}]_0^1 & for\ k \geq 0 \\ 0 & for\ k = -1 \end{cases} \qquad (1.2)$$

$$[x]_0^1 = \begin{cases} 1 & if\ x \geq 1 \\ x & if\ 0 < x < 1 \\ 0 & if\ x \leq 0 \end{cases} \qquad (1.3)$$

Όπου,

$\tilde{p}_{iDARWIN}^{(k)}$: πιθανότητα απόρριψης του DARWIN

$q_i^{(k)}$: δείχνει την κατάσταση του κάθε κόμβου

$\hat{p}_i^{(k)}$: αντιλαμβανόμενη πιθανότητα απόρριψης

Οι κόμβοι που είναι σε καλύτερη κατάσταση από τους κόμβους που συνεργάζονται τιμωρούν τον αντίπαλο κόμβο με τη διαφορά της μεταξύ τους κατάστασης. Έχοντας κάνει την παραδοχή ότι οι κόμβοι δεν κλέβουν, οι αντιλαμβανόμενες πιθανότητες απόρριψης ανταλλάσσονται μεταξύ των κόμβων. Οι αντιλαμβανόμενες πιθανότητες αυτές υπολογίζονται ως το μέσο όρο των λαμβανόμενων πιθανοτήτων απόρριψης.

Εν συνεχεία θα περιγραφεί η λειτουργία του αλγορίθμου. Κάθε κόμβος του δικτύου διατηρεί δύο μετρητές για τους γειτονικούς του κόμβους. Αυτοί είναι ο αριθμός πακέτων που έστειλε προς κάθε γειτονικό κόμβο για προώθηση καθώς και ο αριθμός των πακέτων που τελικά προωθήθηκαν.

Δηλαδή

$N_i^{(k)}$: γειτονικοί κόμβοι του κόμβου i σε χρονικό διάστημα κ

$S_{ij}^{(k)}$: αριθμός πακέτων που εστάλησαν από τον κόμβο i στον j σε χρονικό διάστημα κ

$F_{ij}^{(k)}$: αριθμός πακέτων που προωθήθηκαν από τον κόμβο j σε χρονικό διάστημα κ

Έχοντας υπολογίσει τα παραπάνω ο κόμβος ευρίσκει τον λόγο συνδεσιμότητας που ορίζεται ως : $c_{ij}^{(k)} = \frac{F_{ij}^{(k)}}{S_{ij}^{(k)}}$, τον οποίο και στέλνει στους γείτονες του. Έχοντας βέβαια παραλάβει τον αντίστοιχο λόγο και ο ίδιος από τους γειτονικούς του κόμβους προχωράει στην εύρεση του μέσου λόγου συνδεσιμότητας που ορίζεται ως

$$\hat{c}_j^{(k)} = \frac{\sum_{m \in N_i^{(k)} \cup \{i\}, m \neq j} c_{im}^{(k)} * c_{mj}^{(k)}}{\sum_{m \in N_i^{(k)} \cup \{i\}, m \neq j} c_{im}^{(k)}} \quad (1.4)$$

Βάσει αυτού του λόγου και τις σχέσεις που ορίζουν την πιθανότητα απόρριψης του DARWIN, υπολογίζεται η συνολική πιθανότητα απόρριψης που στην συνέχεια χρησιμοποιείται για την επιλογή γειτονικού κόμβου στον οποίο θα προωθήσει τα πακέτα στον επόμενο γύρο.

1.3 Σύνοψη Πλεονεκτημάτων-Μειονεκτημάτων των μηχανισμών

	Πλεονεκτήματα	Μειονεκτήματα
CORE	• Πλήρης απομόνωση των αρνητικά βαθμολογημένων κόμβων • Δεν επιτρέπεται η διασπορά αρνητικής φήμης στο δίκτυο • Άμεσος εντοπισμός εγωιστών κόμβων με χαμηλό κόστος πόρων για το δίκτυο	• Οι συστάσεις ενός του δικτύου θεωρούνται έμπιστες, γεγονός που το κάνει αδύναμο απέναντι σε κακόβουλους κόμβους [5] • Όχι ιδιαίτερα αποτελεσματικό στην εύρεση κακόβουλων κόμβων • Υποβάθμιση απόδοσης του δικτύου, καθότι προκαλεί επιβάρυνση σε όλους τους κόμβους • Ακατάλληλο για ετερογενή δίκτυα καθώς το βάρος των συναρτήσεων για τον υπολογισμό φήμης είναι προκαθορισμένο
SORI	• Μείωση κατανάλωσης πόρων δικτύου καθώς η φήμη διαμοιράζεται μονάχα στους γείτονες • Ακριβής υπολογισμός φήμης επειδή βασίζεται σε αντικειμενικές μετρήσεις • Συνεργάσιμοι κόμβοι χαρακτηριστικά μεγαλύτερο ρυθμό απόδοσης συγκρινόμενοι με τους εγωιστές, ανεξαρτήτως από στοιχεία δικτύου(ποσοστό εγωιστών κόμβων, αριθμό συνδέσεων, ...)	• Αδυναμία αντιμετώπισης κακόβουλων κόμβων • Αύξηση πληροφοριών ελέγχου σε περίπτωση χρήσης beacons [6]
CONFIDANT	• Ανθεκτικό σε επιθέσεις, καθότι κάθε κόμβος διαχειρίζεται μόνος του την φήμη των άλλων κόμβων. • Ιδιαίτερα χαμηλές απώλειες πακέτων λόγω κακόβουλων κόμβων • Ανθεκτικό σε μεγάλο αριθμό κόμβων • Ανθεκτικό σε μεγάλη αύξηση του αριθμού των κακόβουλων κόμβων • Δεν επηρεάζεται η ποιότητα υπηρεσιών του δικτύου ακόμα και σε συνθήκες αυξημένης κίνησης • Παρέχει την δυνατότητα για επανένταξη κόμβων που πλέον δρουν συνεργατικά	• Σπατάλη πόρων του δικτύου εξαιτίας του μεγάλου αριθμού μηνυμάτων που κυκλοφορούν στο δίκτυο(ALARM, συστάσεις, λίστες) καθώς και ευάλωτο σε επιθέσεις σε αυτά τα μηνύματα.[3][4] • Μέτρια απόδοση όταν το δίκτυο παρουσιάζει μεγάλη κινητικότητα. • Βαρύτητα των συστάσεων δεν διαφοροποιείται σύμφωνα με το πόσο αξιόπιστος είναι ο αποστολέας [5]

RISM	• Ιδιαίτερα αποδοτικός στην αντιμετώπιση κακόβουλων κόμβων • Ανεκτικότητα σε μεγάλους αριθμούς κακόβουλων κόμβων στο δίκτυο • Ικανοποιητική απόδοση στην κινητικότητα • Φροντίζει για την επανένταξη πρώην κακόβουλων κόμβων • Βαρύτητα στις παρατηρήσεις του ίδιου του κόμβου, δηλαδή κάποιος κόμβος δεν χαρακτηρίζεται κακόβουλος αν δεν αποτύχει το τεστ κρούσης	• Μεγάλη επιβάρυνση στους πόρους του δικτύου εξαιτίας των μηνυμάτων πληροφορίας που κυκλοφορούν
Sprite	• Ιδιαίτερα αποτελεσματικός στην προώθηση πακέτων(κοντά στο 100% επιτυχία) • Εξίσου καλά αποτελέσματα και σε αυξημένο αριθμό πακέτων • Αποτελεσματική εύρεση πιο γρήγορων διαδρομών [7] • Προστασία ενάντια σε συμπαιγνία κόμβων	• Ιδιαίτερα απαιτητικό σε πόρους δικτύου εξαιτίας του μεγάλου αριθμού πληροφοριών ελέγχου που απαιτούνται. [8] • Ευάλωτο απέναντι σε ενεργές επιθέσεις στο δίκτυο (Denial Of Service attacks - DOS) • Προβλήματα για τους ακριανούς κόμβους που δεν έχουν τη δυνατότητα προώθησης πακέτων [9]
OCEAN	• Μικρός αριθμός πακέτων πληροφορίας που κυκλοφορούν στο δίκτυο • Απλή και σχετικά εύκολα υλοποιήσιμη δομή • Ανθεκτικότητα σε επιθέσεις κακόβουλων κόμβων • Παρέχει την δυνατότητα για επανένταξη κόμβων που πλέον δρουν συνεργατικά	• Επιεικές στην τιμωρία των κακόβουλων κόμβων, που υπό συνθήκες μπορεί να έχουν καλύτερη απόδοση από τους συνεργατικούς • Έλλειψη προστασίας ενάντια σε συνεργασία μεταξύ των κακόβουλων κόμβων • Προβλήματα υπολογισμού φήμης σε δίκτυα μεγάλης κινητικότητας
DARWIN	• Αυστηρή τιμωρία των εγωιστών κόμβων. Όσο πιο εγωιστές είναι τόσο πιο έντονη η τιμωρία • Ακόμα και σε πολύ υψηλά ποσοστά εγωιστών κόμβων(90%) σε ένα δίκτυο η απόδοση των συνεργατικών κόμβων είναι καλύτερη από τους εγωιστές • Σταθερός λόγος προώθησης ακόμα και σε υψηλούς ρυθμούς μετάδοσης • Αποτελεσματική αντιμετώπιση συμπαιγνίας κακόβουλων κόμβων[10]	

Βιβλιογραφία

[1] Dimitris E. Charilas, Stavroula G. Vassaki, Athanasios D. Panagopoulos, Philip Constantinou, “Cooperation Incentives in 4G Networks”, published in “Game Theory for Wireless Communications and Networking”, Pages 295-314, CRC Press, June 2011.
[2] Animesh Kr Trivedi, Rajan Arora, Rishi Kapoor, Sudip Sanyal, Sugata Sanyal, “A Semi-distributed Reputation-based Intrusion Detection System for Mobile Adhoc Networks”, Journal of Information Assurance and Security 1 , 2006.
[3] S. Buchegger , J.-Y. L. Boudec, “Performance analysis of the CONFIDANT protocol: Cooperation of nodes - fairness in dynamic ad-hoc networks”, IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC), 2002.
[4] Σαΐνης Θεόδωρος , “ΣΥΣΤΗΜΑΤΑ ΥΠΟΛΗΨΗΣ”, Διπλωματική Εργασία, 2005.
[5] Jinshan Liu, Valerie Issarny , “Enhanced Reputation Mechanism for Mobile Ad Hoc Networks”, Second International Conference, 2004.
[6] Qi He, Dapeng Wu, Pradeep Khosla , “A Secure Incentive Architecture for Adhoc Networks”, Wireless Communications and Mobile Computing, vol. 6, no. 3, pp. 333–346, 2006.
[7] S. Zhong, J. Chen, Y. R. Yang, “Sprite: A Simple, Cheat-Proof, Credit-Based Systemfor Mobile Ad-Hoc Networks”, INFOCOM ,2003.
136
[8] M. Jakobsson, J. P. Hubaux, L. Buttyan, “A micropayment scheme encouraging collaboration in multi-hop cellular networks,”, Proceedings of Financial Crypto 2003, 2003.
[9] Elgan Huang, Jon Crowcroft, Ian Wassell, “Rethinking Incentives for Mobile Ad Hoc Networks”, Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems, 2004.
[10] 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.
[11]Sorav Bansal, Mary Baker, “Observation-based Cooperation Enforcement in Ad hoc Networks”, Stanford Technical Report, 2003.

2. ΥΒΡΙΔΙΚΟΙ ΜΗΧΑΝΙΣΜΟΙ ΚΙΝΗΤΡΩΝ

2.1 Εισαγωγικά

Για την αποτελεσματική συνεργασία των κόμβων ενός ασύρματου Ad Hoc δικτύου έχουν αναπτυχθεί δύο είδη μηχανισμών κινήτρων, όπως έχει αναφερθεί παραπάνω.

Πρωτευόντως υπάρχουν οι μηχανισμοί οι οποίοι βασίζονται στην φήμη (reputation-based incentive mechanisms). Αυτοί οι μηχανισμοί βασίζονται στην μέτρηση της φήμης κάθε κόμβου βασιζόμενοι σε κάποια κριτήρια και παρουσιάζουν ιδιαίτερη αποτελεσματικότητα όσων αφορά στον εντοπισμό των "μη-συνεργάσιμων" κόμβων. Ωστόσο δεν παρέχουν αρκετά κίνητρα για την ώθηση των κόμβων σε μια συνεργάσιμη συμπεριφορά.

Το δεύτερο είδος μηχανισμών βασίζεται στην πίστωση (credit-based incentive mechanisms). Σε αυτή την περίπτωση κάθε συνεργάσιμη συμπεριφορά από τον κόμβο αμείβεται, με αποτέλεσμα να παρέχεται κίνητρο σε κάθε κόμβο για την προώθηση αυτού του είδους τη συμπεριφορά. Σε αντίθεση με τον προηγούμενο μηχανισμό, δίνονται ισχυρά κίνητρα για την συνεργασία των κόμβων, αλλά αυτή η προσέγγιση δεν είναι τόσο αποτελεσματική στον εντοπισμό κακόβουλων κόμβων.

Όλα τα παραπάνω δημιούργησαν την ανάγκη για μία νέα προσέγγιση η οποία θα μπορεί να συνδυάσει τα πλεονεκτήματα κάθε μηχανισμού. Εν συνεχεία θα γίνει παρουσίαση δύο υβριδικών αλγορίθμων και των χαρακτηριστικών τους.

2.2 Hybrid Reputation Management System for Mobile Ad Hoc Networks

Ο παρακάτω υβριδικός μηχανισμός βασίζει τη λειτουργία του στην παράλληλη εφαρμογή των δύο διαφορετικών προσεγγίσεων που αναφέρθηκαν παραπάνω. Δηλαδή λειτουργεί εκμεταλλευόμενος τη λογική της φήμης αλλά και της πίστωσης του κάθε κόμβου ταυτόχρονα για την λήψη μιας απόφασης.

Η λογική λειτουργίας του παραπάνω μηχανισμού βασίζεται στο γεγονός ότι η πίστωση που κάθε κόμβος πληρώνει για να στείλει ένα πακέτο εξαρτάται από την φήμη που έχει αποκτήσει. Έτσι όσο υψηλότερη η φήμη ενός κόμβου τόσο μικρότερη η πίστωση που απαιτείται να πληρώσει για την αποστολή ενός πακέτου. Δηλαδή ένας κόμβος με υψηλή φήμη μπορεί να αποστείλει ένα πακέτο για μικρότερη πίστωση από κάποιον που έχει χαμηλή φήμη. Ωστόσο για την βέβαιη τιμωρία των εγωιστών κόμβων υπάρχει ένα κάτω κατώφλι για την φήμη κάθε κόμβου. Με αυτό τον τρόπο εάν η φήμη ενός κόμβου πέσει χαμηλότερα από αυτό το κατώφλι, οι υπόλοιποι αρνούνται να προωθήσουν τα πακέτα που στέλνει ανεξαρτήτως των χρημάτων που διαθέτει ο συγκεκριμένος κόμβος.

Παρακάτω αναλύεται συνοπτικά η λογική λειτουργίας αυτού του μηχανισμού.

Αρχικά παρουσιάζεται ο τρόπος υπολογισμού του οφέλους που έχει κάθε κόμβος με την συνεργασία του ή μη με κάποιο άλλο κόμβο. Στον παρακάτω πίνακα φαίνονται οι τιμές από τα οφέλη που υπάρχουν σε κάθε πιθανό ενδεχόμενο συνεργασίας.

		Κόμβος j	
		Συνεργασία	Μη Συνεργασία
Κόμβος i	Συνεργασία	P(Ci,Cj)	P(Ci,Ij)
	Μη Συνεργασία	P(Ii,Cj)	(0,0)

με

$$P(C_i,C_j)=\left(p-c+\left(m_f-\frac{m_p}{R_i}\right)\right),\left(p-c+\left(m_f-\frac{m_p}{R_j}\right)\right)$$

$$P(C_i,I_j)=\begin{cases}\left(-c+m_f,p-\frac{m_p}{R_i}\right) & V_j>0 \text{ και } R_{I(j)}>R_t\\ (0,0) & \text{Αλλιώς}\end{cases}$$

$$P(I_i,C_j)=\begin{cases}\left(p-\frac{m_p}{R_j},-c+m_f\right) & V_i>0 \text{ και } R_{I(i)}>R_t\\ (0,0) & \text{Αλλιώς}\end{cases}$$

Όπου,

p:όφελος του κόμβου από την προώθηση ενός πακέτου

C:κόστος του κόμβου για την προώθηση ενός πακέτου

m_p :τιμή που πληρώνει κάποιος κόμβος για την αποστολή του πακέτου

m_f :ανταμοιβή του κόμβου που προωθεί το προς αποστολή πακέτο

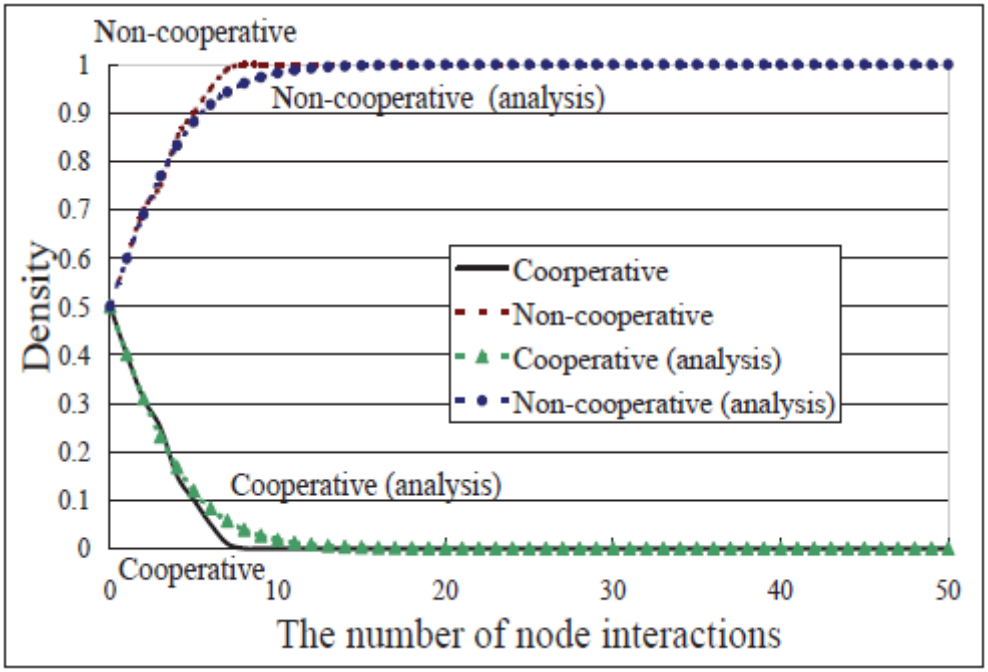
R_i : τιμή της φήμης

R_t :τιμή του κατωφλίου της φήμης

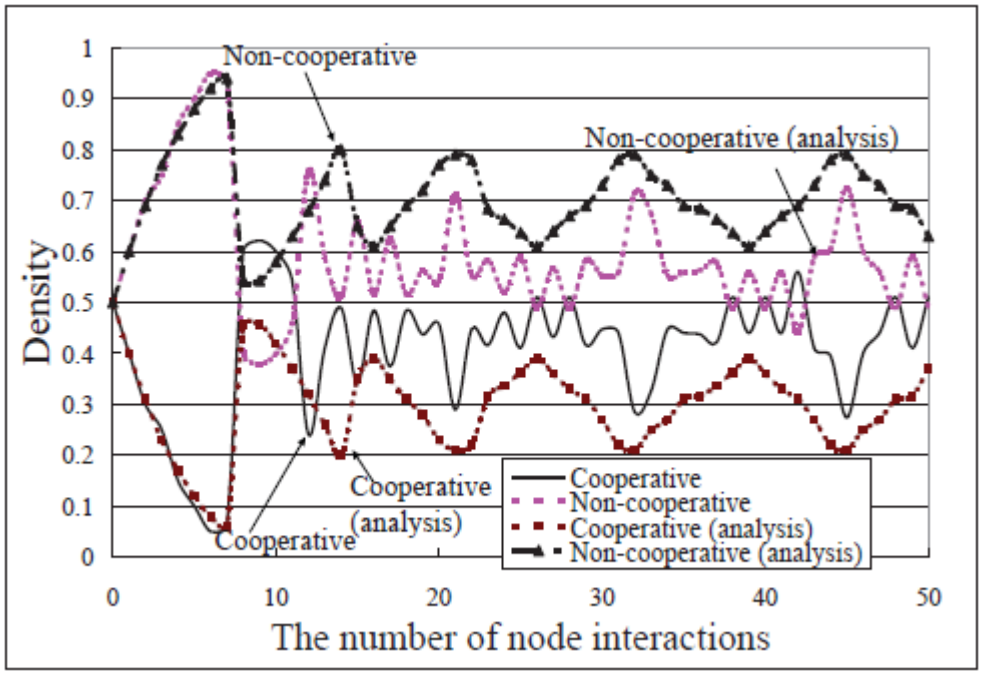
V_i :ποσό χρημάτων που διαθέτει κάποιος κόμβος

Η αμοιβαία συνεργασία των κόμβων στην οποία και βασίζεται ο παραπάνω μηχανισμός είναι σημείο ισορροπίας κατά Nash και Pareto efficient. Το αποτέλεσμα αυτού είναι η διασφάλιση της αμοιβαίας συνεργασίας μεταξύ των κόμβων καθώς παρέχεται ισχυρό κίνητρο. Επίσης σε ενδεχόμενο που κάποιοι κόμβοι παρεκκλίνουν από την αμοιβαία συνεργασία είναι διασφαλισμένο ότι το κέρδος των υπολοίπων δεν θα μεταβληθεί. Σχετικός λόγος επιτυχίας μιας στρατηγικής είναι ο λόγος της συνολικής ανταμοιβής μιας στρατηγικής προς την αντίστοιχη ανταμοιβή όλων των στρατηγικών. Με την αύξηση αυτού του λόγου έχει δειχθεί ότι μεγαλώνει ο αριθμός των κόμβων που συνεργάζονται. Με αυτό τον τρόπο, στην περίπτωση που κάποιοι κόμβοι με κακή φήμη περάσουν από την κατάσταση μη συνεργασίας στην κατάσταση συνεργασίας θα παρατηρήσουν πτώση στην απόδοση των υπηρεσιών του δικτύου.[1]

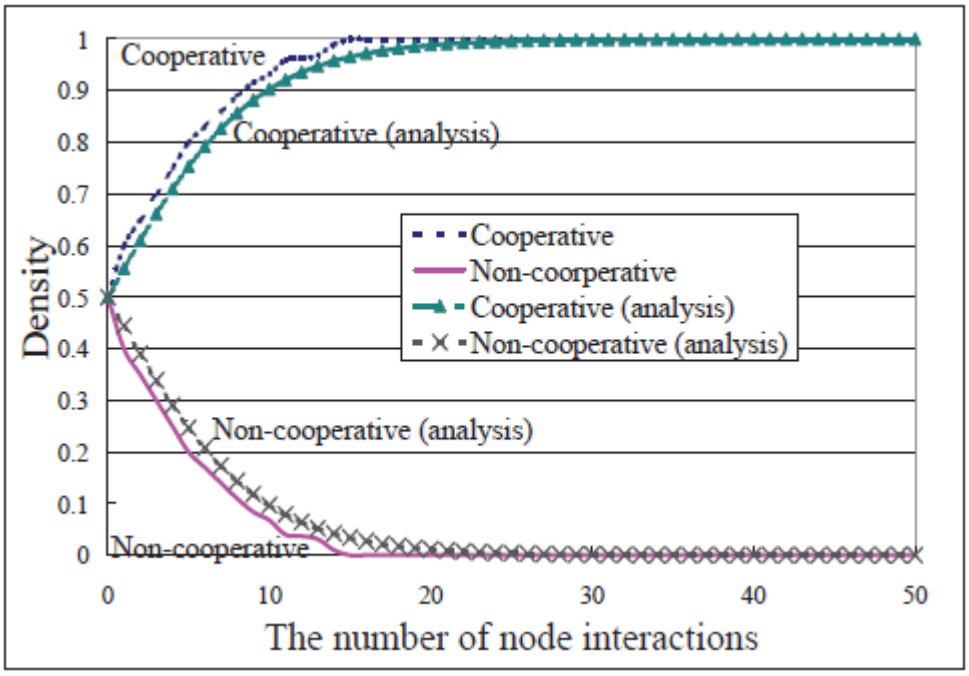
Αναλυτικότερα, ύστερα από προσομοιώσεις που πραγματοποιήθηκαν για MANET δίκτυα (κινητό ad hoc δίκτυο) [1] έγινε σύγκριση του υβριδικού αυτού μηχανισμού(Διάγραμμα 2.4) με ένα σύστημα που δεν χρησιμοποιεί κανένα μηχανισμό κινήτρων (Διάγραμμα 2.1), με ένα σύστημα που εφαρμόζει μηχανισμό φήμης (Διάγραμμα 2.2) και τέλος με κάποιο σύστημα που εφαρμόζει μηχανισμό αμοιβής(Διάγραμμα 2.3).[1] Στον κάθετο άξονα των διαγραμμάτων παρουσιάζεται η πυκνότητα των κόμβων εντός του δικτύου και στον οριζόντιο άξονα ο αριθμός των συναλλαγών που έχουν πραγματοποιηθεί μεταξύ των κόμβων.



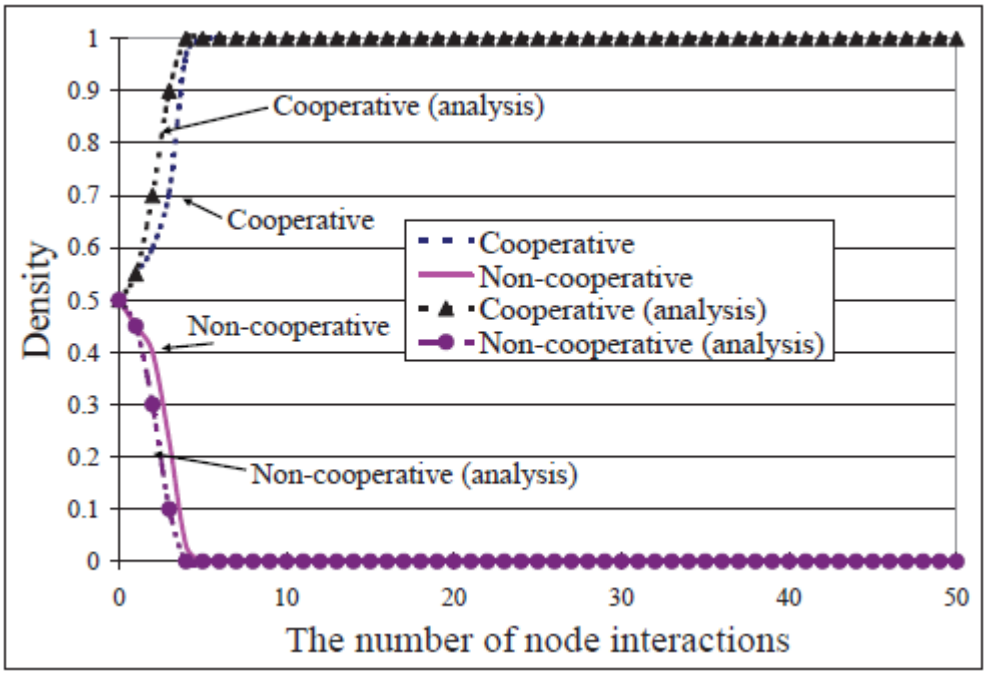
Διάγραμμα 2.1(Πλήρης απουσία μηχανισμού κινήτρων)



Διάγραμμα 2.2(Μηχανισμός φήμης)



Διάγραμμα 2.3(Μηχανισμός αμοιβής)



Διάγραμμα 2.4(Υβριδικός μηχανισμός)

Όπως φαίνεται στο διάγραμμα 2.1, στην περίπτωση απουσίας μηχανισμού κινήτρων η πυκνότητα των μη συνεργατικών κόμβων αυξάνεται δραματικά και εν τέλει κατακλύουν το δίκτυο. Αυτό οδηγεί βέβαια στην κατάρρευση του δικτύου. Στην περίπτωση του μηχανισμού φήμης(Διάγραμμα 2.2), παρουσιάζεται μια κυριαρχία του δικτύου από μη συνεργατικούς κόμβους για τις πρώτες συναλλαγές ενώ στην συνέχεια όταν η τιμή της φήμης των μη συνεργατικών κόμβων πέσει κάτω από ένα κατώφλι, ο αριθμός αυτών μειώνεται και τείνει να εξισορροπηθεί με αυτόν των συνεργατικών κόμβων. Εν συνεχεία, καθώς η τιμή φήμης των μη συνεργατικών αυξηθεί ξανά, επιστρέφουν στην κακή τους συμπεριφορά και ο αριθμός τους αυξάνεται. Έτσι έχουμε αυτή την "τραμπάλα" των τιμών συνεργατικών και μη συνεργατικών κόμβων. Από την άλλη ο μηχανισμός πίστωσης (Διάγραμμα 2.3) δείχνει να αντιμετωπίζει ιδιαιτέρως αποτελεσματικά τους μη συνεργατικούς κόμβους καθώς ήδη ύστερα από περίπου 10 συναλλαγές των κόμβων οι συνεργατικοί κόμβοι έχουν κατακλύσει το δίκτυο. Τέλος, μελετώντας το διάγραμμα 2.4 του υβριδικού μηχανισμού παρατηρείται ότι αντιμετωπίζει ακόμα καλύτερα τους μη συνεργατικούς κόμβους και οδηγεί τους κόμβους του δικτύου σε συνεργατική συμπεριφορά. Όπως φαίνεται και από το διάγραμμα οι συνεργατικοί κόμβοι ύστερα από 3 περίπου γύρους (πολύ γρηγορότερα σε σχέση με τον μηχανισμό πίστωσης) έχουν την απόλυτη κυριαρχία του δικτύου ενώ οι μη συνεργατικοί κόμβοι έχουν εξαλειφθεί.

Πλεονεκτήματα - Μειονεκτήματα

Ο παραπάνω μηχανισμός κινήτρων συνεργασίας καταφέρνει να συνδυάσει δύο διαφορετικές προσεγγίσεις, από την μια αυτή της φήμης και αυτή της πίστωσης. Με αυτό

τον τρόπο καταφέρνει να διασφαλίσει υψηλό επίπεδο υπηρεσιών καθώς ωθεί τους κόμβους να συνεργάζονται σε διαρκή βάση. Αυτό επιτυγχάνεται από το γεγονός ότι είναι στο όφελος του κάθε κόμβου να διατηρεί μια υψηλή φήμη για να έχει την δυνατότητα αποστολής πακέτων σε χαμηλές τιμές. Από προσομοιώσεις που έγιναν (κοίτα τέλος της ενότητας 2.2), αποδεικνύεται ότι η συγκεκριμένη υβριδική προσέγγιση υπερτερεί των μηχανισμών που βασίζονται μονάχα στην φήμη ή στην πίστωση. Πιο συγκεκριμένα οι κόμβοι οδηγούνται στην συνεργασία αρκετά γρηγορότερα από ότι σε ένα μηχανισμό που βασίζεται στην πίστωση.[1]

2.3 HEAD (A Hybrid Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks)

Το HEAD είναι βασισμένο στον μηχανισμό φήμης OCEAN και εφαρμόζει τους μηχανισμούς φήμης και πίστωσης ξεχωριστά για την αντιμετώπιση διαφορετικών θεμάτων σε ένα δίκτυο και όχι παράλληλα όπως γινόταν προηγουμένως. Πιο συγκεκριμένα χρησιμοποιεί τον μηχανισμό φήμης για την εύρεση και τιμωρία των κακόβουλων κόμβων σε ένα δίκτυο, ενώ κάνει χρήση του μηχανισμού πίστωσης για να ωθήσει τους εγωιστές κόμβους σε συνεργατική συμπεριφορά.

Σε αυτό τον μηχανισμό η φήμη κάθε κόμβου είναι προϊόν παρατηρήσεων που κάνει ο ίδιος ο κόμβος στις επαφές του με το περιβάλλον του. Ο κάθε κόμβος διατηρεί δύο λίστες τις οποίες συμβουλεύεται στις συναλλαγές του με τους υπόλοιπους κόμβους. Πιο συγκεκριμένα έχει μια ελαττωματική λίστα για τους κακόβουλους κόμβους και μια εγωιστική λίστα για τους εγωιστές κόμβους. Σε κάθε επαφή του με κάποιο κόμβο γίνεται έλεγχος αν αυτός ο κόμβος ανήκει σε μια από αυτές τις λίστες και σε περίπτωση που ανήκει, το πακέτο που του αποστάλθηκε απορρίπτεται.

Παρακάτω παρατίθεται η μορφή του πίνακα φήμης που διατηρεί ο κάθε κόμβος.

	ID γείτονικού κόμβου	Αξιολόγηση	Chipcount	Λίστα αποφυγής	Λίστα nonchip
Κ					
ό					
μ					
β					
ο					
ι					

Όπου

Αξιολόγηση: τιμή της φήμης κάθε κόμβου, που αυξάνεται και μειώνεται σύμφωνα με την συνεργασία ή μη συνεργασία του κόμβου. Αν αυτή η τιμή πέσει κάτω από ένα κατώφλι ο κόμβος αυτός μπαίνει στην ελαττωματική λίστα.

Chipcount: το ίδιο όπως στο OCEAN

Λίστα αποφυγής: λίστα με κακόβουλους κόμβους

Λίστα nonchip: λίστα με εγωιστές κόμβους

Για την διασφάλιση της ομαλής λειτουργίας του μηχανισμού πρέπει να γίνεται βέβαιο πως οι πληροφορίες που συλλέγει ο κάθε κόμβος για τους γείτονες του, γίνονται γνωστές και στους υπόλοιπους κόμβους του δικτύου. Αυτό συμβαίνει με την περιοδική αποστολή των δύο αυτών λιστών που διατηρεί ο κάθε κόμβος σε κάθε γειτονικό του κόμβο. Επίσης για την δυνατότητα εξισορρόπησης των υπηρεσιών του δικτύου μετά από κάποιο χρονικό διάστημα κάθε κόμβος που βρίσκεται στην ελαττωματική λίστα αφαιρείται από αυτή για να ξαναγίνει χρήσιμος στο δίκτυο. Ωστόσο διατηρεί την τιμή φήμης που είχε πριν μπει στην λίστα, ώστε να ξαναμπεί γρήγορα στην λίστα σε περίπτωση που συνεχίσει την προηγούμενη του συμπεριφορά.

Παρακάτω θα παρουσιαστούν συντόμως δύο μηχανισμοί σύμφωνα με τους οποίους ο HEAD πραγματοποιεί την δρομολόγηση των πακέτων στο δίκτυο.[2]

1ος Μηχανισμός : Επαναποστολή πακέτου

- ο κόμβος δέχεται και ελέγχει τα πεδία του πακέτου
- Εάν ο αποστολέας ανήκει στην λίστα με τους κακόβουλους κόμβους
    - απόρριψη πακέτου
- Εάν ο αποστολέας ανήκει στην λίστα με τους εγωιστές κόμβους
    - απόρριψη πακέτου
- Αλλιώς επαναποστολή πακέτου

2ος Μηχανισμός : Προώθηση πακέτου

- ο κόμβος δέχεται και ελέγχει τα πεδία του πακέτου
- Εάν ο κόμβος στο προηγούμενο ή το επόμενο βήμα (hop) του πακέτου ανήκει στην λίστα με τους κακόβουλους κόμβους
    - απόρριψη πακέτου

- Εάν ο κόμβος στο προηγούμενο ή το επόμενο βήμα (hop) του πακέτου ανήκει στην λίστα με τους εγωιστές κόμβους
    - απόρριψη πακέτου
- Αλλιώς προώθησε το πακέτο στο επόμενο βήμα (hop)

Τέλος παρουσιάζεται ο αλγόριθμος που ακολουθείται από τον μηχανισμό HEAD για την προώθηση των πακέτων.[2]

- ο κόμβος δέχεται και ελέγχει τα πεδία του πακέτου
- Εάν ο κόμβος στο προηγούμενο ή το επόμενο βήμα (hop) του πακέτου ανήκει στην λίστα με τους κακόβουλους κόμβους
    - απόρριψη πακέτου
- Εάν ο προηγούμενος κόμβος ανήκει στη λίστα με τους εγωιστές κόμβους
    - απόρριψη πακέτου
    - μείωση του chipcount του κόμβου του προηγούμενου βήματος(hop) κατά 1
    - αύξηση του chipcount του κόμβου του επόμενου βήματος (hop) κατά 1
    - διατήρηση του πακέτου στην μνήμη και αρχικοποίηση ενός μετρητή timeout
    - προώθηση πακέτου
- Εάν ο κόμβος που απέχει δύο βήματα (hops) ανήκει στην λίστα με τους κακόβουλους ή εγωιστές κόμβους
    - κατάργηση του μετρητή timeout και επιστροφή
- Εάν ευρεθεί ο κόμβος στο επόμενο βήμα της διαδρομής προτού μηδενιστεί ο μετρητής timeout
    - πυροδότηση ενός θετικού συμβάντος
- Αλλιώς
    - πυροδότηση ενός αρνητικού συμβάντος

Πλεονεκτήματα - Μειονεκτήματα

Ο παραπάνω μηχανισμός κινήτρων συνεργασίας καταφέρνει να είναι ιδιαίτερα αποτελεσματικός στην αντιμετώπιση των κακόβουλων κόμβων, ως επίσης και στην ώθηση των εγωιστών κόμβων σε συνεργατική συμπεριφορά. [2] Ωστόσο καθότι βασίζεται στις παρατηρήσεις του κάθε κόμβου για την ορθή διαμόρφωση των λιστών φήμης, δημιουργείται πρόβλημα σε ενδεχόμενο μεγάλης κινητικότητας των κόμβων.

Βιβλιογραφία

[1] Ze Li , Haiying Shen, "Analysis of A Hybrid Reputation Management System for Mobile Ad hoc Networks", IEEE, 2009

[2] 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.

3. ΠΡΟΤΕΙΝΟΜΕΝΟΣ ΥΒΡΙΔΙΚΟΣ ΜΗΧΑΝΙΣΜΟΣ ICARUS

3.1 Εισαγωγικά

Ένα ad hoc δίκτυο αποτελεί έναν αποκεντρωμένο τύπο ασύρματου δικτύου. Το βασικό χαρακτηριστικό ενός τέτοιου δικτύου είναι το γεγονός ότι δεν βασίζεται σε μια ήδη προϋπάρχουσα και σταθερή υποδομή. Δηλαδή δεν υπάρχει ανάγκη για δρομολογητές και κεντρικά access points δεδομένου πως ο κάθε κόμβος έχει την δυνατότητα για άμεση επικοινωνία με οποιονδήποτε άλλο κόμβο του δικτύου, καθώς ο ίδιος ο κόμβος λειτουργεί σαν δρομολογητής. Ωστόσο για την ομαλή και αποτελεσματική λειτουργία ενός τέτοιο δικτύου πρέπει να διασφαλιστεί η συνεργασία των κόμβων που απαρτίζουν το δίκτυο. Έτσι για την αντιμετώπιση των κακόβουλων (κόμβοι που κάνουν ενέργειες με στόχο να βλάψουν το δίκτυο) και εγωιστών(κόμβοι που επιχειρούν να εκμεταλλευτούν τους πόρους του δικτύου χωρίς οι ίδιοι να παρέχουν τις υπηρεσίες τους στο δίκτυο - προώθηση πακέτων) κόμβων έχουν αναπτυχθεί μια σειρά μηχανισμών κινήτρων συνεργασίας για την τιμωρία των μη συνεργάσιμων κόμβων και τον εξαναγκασμό των για συνεργασία. Αυτοί οι μηχανισμοί έχουν διαχωριστεί με βάση τα κριτήρια στα οποία βασίζουν την λειτουργία τους.

Οι δύο κατηγορίες στις οποίες έχουν διαχωριστεί οι μηχανισμοί κινήτρων συνεργασίας, είναι οι μηχανισμοί φήμης και πίστωσης (κοίτα Ενότητα 1 της παρούσας διπλωματικής). Οι μηχανισμοί φήμης διατηρούν έναν μετρητή φήμης για κάθε κόμβο του οποίου η τιμή αυξάνεται σε περίπτωση που κάποιος κόμβος λειτουργήσει συνεργατικά και αντιστοίχως μειώνεται σε αντίθετη περίπτωση. Βάση αυτού του μετρητή το δίκτυο ξεχωρίζει τους κακόβουλους και εγωιστές κόμβους και τους τιμωρεί. Από την άλλη πλευρά υπάρχουν και οι μηχανισμοί πίστωσης που κάνουν χρήση μιας θεωρητικής οικονομικής μονάδας που δίνεται στους κόμβους με συνεργατική συμπεριφορά ώστε να έχουν την δυνατότητα να αποστείλουν τα πακέτα τους.

Τέλος συνδυάζοντας τους δύο παραπάνω μηχανισμούς κινήτρων συνεργασίας, υπάρχουν οι υβριδικοί μηχανισμοί κινήτρων. Ο μηχανισμός με τον οποίο κάθε αλγόριθμος συνδυάζει τους μηχανισμούς φήμης και πίστωσης έχει αναλυθεί ήδη στην ενότητα 2 της παρούσας διπλωματικής.

Σε αυτή την διπλωματική έχει αναπτυχθεί ένας υβριδικός μηχανισμός, που καλείται ICARUS (hybrId inCentive mechAnism for coopeRation stimUlation in ad hoc networks) και βασίζεται στο μηχανισμό DARWIN, ο οποίος περιγράφηκε στην Ενότητα 1.2.7. Ο ICARUS υιοθετεί τη χρήση μιας κεντρικής οντότητας παρόμοια με το Sprite, έναν μηχανισμό που βασίζεται σε credits (Ενότητα 1.2.5). Ο στόχος είναι να βελτιωθεί η απόδοση του DARWIN χρησιμοποιώντας στοιχεία από τους μηχανισμούς πίστωσης .

Παρακάτω παρουσιάζεται ένας πίνακας με τους συμβολισμούς που χρησιμοποιούνται στον ICARUS.

N	A set containing all nodes
$\mathbf{N_c}$	A set containing only cooperating nodes
$edp_i^{(k)}$	Estimated dropping probability for node *i* in time slot *k*
$p_i^{(k)}$	Expected dropping probability of node *i* at time slot *k*
$N_i^{(k)}$	A set containing the neighbors of node *i* at time slot *k*
$S_{ij}^{(k)}$	Number of packets sent from node *i* to node *j* for forwarding until time slot *k*
$F_{ij}^{(k)}$	Number of packets forwarded from node *j* for node *i* until time slot *k*
$c_{ij}^{(k)}$	Forwarding ratio of node *j* at time slot *k*, as perceived by node *i*
$cp_j^{(k)}$	Average connectivity ratio of node *j* at time slot *k*
$p_{i,r}$	Packet sent from node *i* with index *r*
SBI_i	Selfish Behavior Index for node *i*
$cp_{ICAS,i}^{(k)}$	Forwarding ratio of node *i* at time slot *k*, as perceived by ICAS
$m_i^{(k)}$	The number of credits that node *i* possesses at time slot *k*
$r_{i,z}$	The reward (in credits) of node *i* for forwarding packet *z* at time slot *k*
$\cos t_{i,z}$	The total cost for a single packet *z* sent by node *i*
$rl_{i,z}$	A list of all the nodes that acted as relays for packet *z*
$S_i^{(k)}$	Total number of packets to node *i* for forwarding until time slot *k*

3.2 Μοντελοποίηση δικτύου

Σε αυτή την ενότητα θα επεξηγηθεί ο τρόπος σχεδιασμού του δικτύου επί του οποίου θα εφαρμοστεί και θα αξιολογηθεί ο υβριδικός μηχανισμός.

3.2.1 Στρατηγικές και αποδόσεις

Αρχικά θεωρούμε ότι ένα μέρος του δικτύου αποτελείται από εγωιστές κόμβους οι οποίοι αρνούνται να προωθήσουν τα πακέτα που τους στέλνονται. Οι κόμβοι

εξετάζονται σε ζευγάρια, ως παίγνιο δύο παικτών όπου ο ένας συμβολίζεται με i και ο άλλος με -i. Το ζευγάρι αυτών των κόμβων επαναλαμβάνουν ένα παίγνιο όπου ο ένας στέλνει πακέτα στον άλλο και αποφασίζουν αν θα προωθήσουν ή θα απορρίψουν πακέτα. Όπως έχει αναλυθεί σε προηγούμενες ενότητες ο κάθε κόμβος αμείβεται για την προώθηση ενός πακέτου και αυτή του η αμοιβή ορίζεται ως α. Ο πίνακας αποδόσεων του παιγνίου αποφάσεων φαίνεται παρακάτω.

		Node 2	
		Forward	Drop
Node 1	Forward	$\alpha -1$ $\alpha -1$	$-\alpha -1$ α
	Drop	α $-\alpha -1$	$-\alpha$ $-\alpha$

Πίνακας 3.1

Αντιστοίχως η κανονικοποιημένη μορφή του πίνακα έχει ως εξής.

		Node 2	
		Forward	Drop
Node 1	Forward	$\alpha -1$ $\alpha -1$	$-\alpha -1$ α
	Drop	α $-\alpha -1$	$-\alpha$ $-\alpha$

Πίνακας 3.2

Η κανονικοποιημένη μορφή του πίνακα ευρίσκεται ως εξής: αν *v* είναι μια τιμή του Πίνακα 3.1 και v_n η αντίστοιχη κανονικοποιημένη τιμή του Πίνακα 3.2, τότε $v_n = (v+a)/(2\cdot a-1)$.

Η αναμενόμενη πιθανότητα απόρριψης του κόμβου i σε μια χρονοθυρίδα κ ορίζεται ως $p_i^{(k)}$. Ένα πακέτο θεωρείται απορριφθέν είτε αν ο *–i* το απορρίψει είτε εάν δεν χαθεί αλλά ο *i* δεν ενημερωθεί ποτέ για τη μετάδοση. Με βάση τον Πίνακα 3.2, η μέση απόδοση $u_i^{(k)}$ του *i* τη χρονοθυρίδα *k* είναι

$$u_i^{(k)} = (1-p_i^{(k)})(1-p_{-i}^{(k)}) + \frac{2\cdot a}{2\cdot a-1}\cdot p_i^{(k)}\cdot(1-p_{-i}^{(k)}) - \frac{2\cdot a}{2\cdot a-1}\cdot(1-p_i^{(k)})\cdot p_{-i}^{(k)} =$$
$$= 1 + \frac{1}{2\cdot a-1}\cdot p_i^{(k)} - \frac{2\cdot a}{2\cdot a-1}\cdot p_{-i}^{(k)} \qquad (3.1)$$

[1][2]

Η αποστολή πακέτων στο δίκτυο γίνεται από όλους τους κόμβους προς παραλήπτες οι οποίοι αλλάζουν. Όταν ο παραλήπτης λάβει το πακέτο που προορίζεται για αυτόν αποστέλλει ένα πακέτο επιβεβαίωσης (ack) το οποίο ακολουθώντας την ίδια διαδρομή καταλήγει στον αποστολέα. Έτσι ένας κόμβος ανανεώνει τον μετρητή του μονάχα μόλις λάβει το ack, το οποίο με την σειρά του εάν χαθεί ο γειτονικός κόμβος θα θεωρηθεί ότι το απέρριψε.

Για λόγους απλότητας, γίνεται η παραδοχή ότι όλοι οι κόμβοι θα προωθούν πάντα τα acks, αφού το μικρό τους μέγεθος δεν απαιτεί πολλούς πόρους για την προώθηση. Η εκτιμώμενη πιθανότητα απόρριψης $u_i^{(k)}$ του κόμβου i τη χρονοθυρίδα k υπολογίζεται όπως στις αναφορές [1][2].

Κάθε κόμβος του δικτύου διατηρεί δύο μετρητές για τους γειτονικούς του κόμβους. Αυτοί είναι ο αριθμός πακέτων που έστειλε προς κάθε γειτονικό κόμβο για προώθηση καθώς και τον αριθμό των πακέτων που τελικά προωθήθηκαν.

Δηλαδή

$N_i^{(k)}$: γειτονικοί κόμβοι του κόμβου i σε χρονικό διάστημα κ

$S_{ij}^{(k)}$: αριθμός πακέτων που εστάλησαν από τον κόμβο i στον j σε χρονικό διάστημα κ

$F_{ij}^{(k)}$: αριθμός πακέτων που προωθήθηκαν από τον κόμβο j σε χρονικό διάστημα κ

Έχοντας υπολογίσει τα παραπάνω ο κόμβος ευρίσκει τον λόγο συνδεσιμότητας που ορίζεται ως : $$c_{ij}^{(k)} = \frac{F_{ij}^{(k)}}{S_{ij}^{(k)}} \qquad (3.2)$$

, τον οποίο και στέλνει στους γείτονες του. Έχοντας βέβαια παραλάβει τον αντίστοιχο λόγο και ο ίδιος από τους γειτονικούς του κόμβους προχωράει στην εύρεση του μέσου λόγου συνδεσιμότητας που ορίζεται ως

$$cp_j^{(k)} = \frac{\sum_{m \in N_i^{(k)} \cup \{i\}, m \neq j} c_{im}^{(k)} \cdot c_{mj}^{(k)}}{\sum_{m \in N_i^{(k)} \cup \{i\}, m \neq j} c_{im}^{(k)}} \quad \text{όπου } c_{ii}^{(k)} = 1 \qquad (3.3)$$

Τέλος αξίζει να σημειωθεί πως ο μέσος σταθμίζεται με την εκτιμώμενο ποσοστό συνδεσιμότητας που μετράει ο i για τον m. Με αυτό τον τρόπο προστατεύεται το σύστημα μας από επιθέσεις και διάδοση ψευδούς πληροφορίας. Με αυτό τον τρόπο ελαχιστοποιούμε τη σημασία της πληροφορίας που προέρχεται από αναξιόπιστους κόμβους.

3.2.2 Δρομολόγηση

Σε αυτή την ενότητα θα γίνει περιγραφή της δρομολόγησης όπως αυτή πραγματοποιείται από τον υβριδικό μηχανισμό που παρουσιάζεται. Βέλτιστη διαδρομή στο δίκτυο μας ορίζεται η διαδρομή με τα λιγότερα βήματα (hops) για να φτάσει στον τελικό προορισμό. Στην περίπτωση όπου ευρεθούν δύο ή περισσότερες διαδρομές με τον ίδιο αριθμό βημάτων επιλέγεται αυτή που παρουσιάζει την μικρότερη συνολική απόσταση.

Όσων αφορά στην εύρεση της βέλτιστης διαδρομής ο κάθε κόμβος δεν γνωρίζει ολόκληρη την διαδρομή στο σύνολο της αλλά μόνο το επόμενη βήμα της διαδρομής από αυτόν. Πιο συγκεκριμένα κάθε κόμβος διατηρεί δύο είδη πινάκων. Ο πρώτος περιλαμβάνει μόνο τους γείτονες του κόμβου και περιέχει πληροφορίες όπως φήμη, στατιστικά, κτλ. Ο δεύτερος πίνακας περιλαμβάνει μόνο τους μη γειτονικούς κόμβους και περιέχει μόνο ελάχιστες πληροφορίες: ένδειξη για το ποιος γειτονικός κόμβος πρέπει να χρησιμοποιηθεί ως ενδιάμεσος, συνολικός αριθμός απαιτούμενων βημάτων και συνολική απόσταση σε μέτρα.[3]

Αναλόγως με άλλους μηχανισμούς που είδαμε στην ενότητα 1, όταν ένας κόμβος έχει χαρακτηριστεί ως εγωιστής, δεν επιλέγεται ποτέ σαν βέλτιστος ενδιάμεσος κόμβος σε μια διαδρομή. Ο κόμβος i θεωρεί τον κόμβο j εγωιστή αν το ποσοστό προώθησης $c_{ij}^{(k)}$ είναι μικρότερο από ένα προκαθορισμένο κατώφλι. Το $c_{ij}^{(k)}$ υπολογίζεται περιοδικά έτσι ώστε να εντοπίζονται αποδοτικά οι εγωιστές κόμβοι. Επίσης κανένας κόμβος δεν προωθεί πακέτα σε κάποιον ενδιάμεσο εγωιστή κόμβο αλλά αν το επόμενη βήμα της διαδρομής είναι εγωιστής κόμβος γίνεται εύρεση νέας διαδρομής. Για την διασφάλιση της ευρωστίας του δικτύου και την αποφυγή του φαινομένου του ping pong, ένα πακέτο δεν μπορεί να προωθηθεί σε έναν κόμβο από τον οποίο έχει ήδη περάσει. Επιπροσθέτως ορίζεται ένα h_{max} το οποίο αποτελεί

τον μέγιστο αριθμό βημάτων (hops) που μπορεί να κάνει ένα πακέτο προτού απορριφθεί από το δίκτυο.

Παρακάτω παρατίθεται ο αλγόριθμος σε μορφή ψευδοκώδικα για την λογική λειτουργίας της δρομολόγησης όπως αυτή αναπτύχθηκε παραπάνω. [3]

```
1.  next_node is specified by old route to B;
2.  if (next_node is not selfish and next_node is not in the existing path) then
3.      send packet to next_node;
4.  end if
5.  else do
6.      for each neighbor j do
7.          if (j is not selfish and j is not in the existing path) then
8.              if (proposed route to B is optimal) then
9.                  new route to B has been found;
10.             end if
11.         end if
12.     end for
13.     if (a new route to B has been found) then
14.         new_next_node is specified by new route;
15.         send packet to new_next_node;
16.         if (next_node is selfish) then
17.             A stores new route to B;
18.         end if
19.     end if
20.     else do
21.         if (next_node is not in the existing path) then
22.             send packet to next_node;
23.         end if
24.         else do
25.             search all neighbors that are not in the existing path to find new route to B;
26.             if (a new route to B has been found) then
27.                 new_next_node is specified by new route;
28.                 send packet to new_next_node;
29.             end if
30.             else do
31.                 drop the packet;
32.             end else
33.         end else
34.     end else
35. end else
```

3.2.3 Συμπεριφορά εγωιστών κόμβων

Στο δίκτυο μας θεωρούμε ότι ένα μέρος των συνολικών κόμβων είναι εγωιστές. Αυτό σημαίνει ότι αυτοί οι κόμβοι απορρίπτουν πακέτα που λαμβάνουν προς προώθηση και εκμεταλλεύονται με αυτό τον τρόπο τους πόρους του δικτύου χωρίς να προσφέρουν. Αυτοί

οι κόμβοι βέβαια κάποια στιγμή απομονώνονται από το δίκτυο και μόλις το αντιληφθούν αυτό αρχίζουν να προωθούν πακέτα για να πάψουν να θεωρούνται εγωιστές.

Ένας κόμβος ορίζεται ως απομονωμένος μόνο όταν αποτυγχάνει να μην έχει κάποιο προωθημένο πακέτο για ένα συγκεκριμένο χρονικό διάστημα. Ο συνολικός αριθμός των πακέτων που πρέπει να απορριφθούν εώς ότου ένας εγωιστής κόμβος θεωρήσει τον εαυτό του απομονωμένο συμβολίζεται ως IFN (Isolation Flag Number).

Υποθέτοντας ότι όλα τα πακέτα που αποστέλλονται από τον κόμβο *i* συμβολίζονται ως $p_{i,index}$ και ότι *f* είναι ο αριθμός των πακέτων που προωθούνται με σειριακό αριθμό index ανάμεσα στο διάστημα $r - IFN \leq index \leq r$, τότε ο δείκτης *SBI (Selfish Behaviour Index)* για τον κόμβο *i* αφότου έχει σταλεί το πακέτο με σειριακό αριθμό *r*, ορίζεται ως

$$SBI_i^{(r)} = f \qquad (3.4)$$

Βάση της προηγούμενης σχέσης, οι εγωιστές κόμβοι θα ξεκινήσουν να συνεργάζονται μόνο όταν $SBI_i = 0$. Ωστόσο σε περίπτωση όπου τους προωθηθεί ακόμα και ένα πακέτο (δηλαδή $SBI_i > 0$) μέσα σε αυτό το διάστημα θα συνεχίζουν την εγωιστική τους συμπεριφορά, κλέβοντας. Η στρατηγική επομένως των εγωιστών κόμβων συνοψίζεται ως

- Απόρριψη πακέτων, αν $SBI_i > 0$
- Εφαρμογή μηχανισμού κινήτρων (δηλαδή συμπεριφορά συνεργατικού κόμβου) αν $SBI_i = 0$

3.3 Συνεργασία κόμβων

Σε αυτό το σημείο θα περιγραφεί ο μηχανισμός συνεργασίας ICARUS ο οποίος βασιζόμενος στον DARWIN εκμεταλλεύεται υβριδικά τα συστήματα φήμης και πίστωσης ταυτόχρονα. Πιο συγκεκριμένα στο ICARUS εκμεταλλευόμαστε τη φήμη για να ελέγχουμε την ανταλλαγή credits. Όλες οι συναλλαγές πραγματοποιούνται από μια κεντρική οντότητα διαχείρισης των credits που ονομάζεται ICAS (ICARUS Credit Accounts Service).

Οι κύριοι στόχοι του ICARUS συνοψίζονται ως: [3]

- Ταχύτερη και αυστηρότερη τιμωρία εγωιστών κόμβων
- Βελτίωση QoS για μη εγωιστές κόμβους
- Κίνητρα για συνεργατικούς κόμβους ώστε να συνεχίσουν να προωθούν πακέτα
- Δικαιοσύνη για μακρινούς κόμβους
- Αποκλεισμός της πιθανότητας οι εγωιστές κόμβοι να εκμεταλλευτούν μια μακρινή θέση για να πλουτίσουν

- Προστασία του συστήματος από κακόβουλους κόμβους που διαδίδουν ψευδείς πληροφορίες.

3.3.1 Λειτουργία του ICAS

Ο ρόλος του ICAS μέσα στο δίκτυο είναι αυτή του παρατηρητή και του ελεγκτή της κίνησης των credits(πληρωμές, αμοιβές, κοστολόγηση, κτλ). Ο κύριος στόχος αυτού είναι η ύπαρξη ενός κεντρικού παρατηρητή με στόχο να διασφαλίσουμε το δίκτυο από κλοπές credits καθώς και διάχυση ψευδών πληροφοριών από κακόβουλους κόμβους.

Οι κύριες λειτουργίες του ICAS παρατίθενται παρακάτω [3]

- Το ICAS διατηρεί τα credits κάθε κόμβου.
- Το ICAS αναθέτει ένα μικρό αριθμό credits m_0 σε όλους τους καινούργιους κόμβους
- Όταν λαμβάνεται επιβεβαίωση λήψης πακέτου, το ICAS υπολογίζει τις αμοιβές για όλους τους ενδιάμεσους κόμβους και το κόστος για τον αποστολέα κόμβο. Τα ποσά αυτά ανταλλάσσονται στη συνέχεια μεταξύ των συμβαλλόμενων κόμβων.
- Το ICAS επιτρέπει ή απαγορεύει στους κόμβους να στείλουν πακέτα ανάλογα με τον αριθμό των credits που διαθέτουν.
- Το ICAS βοηθάει τους απομακρυσμένους κόμβους αν χρειάζεται

Στις επόμενες ενότητες επεξηγούνται οι λειτουργίες του ICAS. Κάθε κόμβος υπολογίζει το ποσοστό προώθησης $c_{ij}^{(k)}$ για όλους τους γείτονές του, με βάση την εξίσωση .

$$cp_j^{(k)} = \frac{\sum_{m \in N_i^{(k)} \cup \{i\}, m \neq j} c_{im}^{(k)} \cdot c_{mj}^{(k)}}{\sum_{m \in N_i^{(k)} \cup \{i\}, m \neq j} c_{im}^{(k)}} \quad \text{όπου } c_{ii}^{(k)} = 1 \qquad (3.5)$$

Το ποσοστό αυτό το προωθούν στο ICAS, το οποίο στη συνέχεια υπολογίζει το μέσο ποσοστό συνδεσιμότητας $cp_{ICAS,i}^{(k)}$ κάθε κόμβου, με βάση τη συγκεντρωτική πληροφορία. Υποθέτοντας ότι *N* είναι το σύνολο των κόμβων και $\mathrm{N_c} \; \Xi \; \mathrm{N}$ είναι ένα υποσύνολο που περιέχει μόνο τους συνεργατικούς κόμβους,

$$cp_{ICAS,i}^{(k)} = \frac{\sum_{j \in N_c}^{\text{cardinality}\{N\}} c_{ji}^{(k)}}{\text{cardinality}\{N_c\}} \qquad (3.6)$$

Αν το ποσοστό αυτό είναι μεγαλύτερο από ένα συγκεκριμένο κατώφλι $cp_{ICAS,th}$, τότε το ICAS θεωρεί τον κόμβο *i* εγωιστή

$$i = \begin{Bmatrix} \text{selfish, if } cp_{ICAS,i}^{(k)} < cp_{ICAS,th} \\ \text{non selfish, if } cp_{ICAS,i}^{(k)} \geq cp_{ICAS,th} \end{Bmatrix} \qquad (3.7)$$

Για την αποφυγή της διάδοσης ψευδής πληροφορίας στο δίκτυο από μια μικρή ομάδα εγωιστών κόμβων, το ICAS λαμβάνει το μέτρο να αγνοεί τις παρατηρήσεις που παρουσιάζουν οι κόμβοι που έχουν μαρκαριστεί ως εγωιστές. Αφού λοιπόν συγκεντρώνεται πληροφορία μόνο από αξιόπιστους κόμβους, είναι αδύνατο για τους κακόβουλους κόμβους να διαδώσουν ψευδή πληροφορία. Ακόμα κι αν λίγοι τέτοιο κόμβοι διαδώσουν ψευδή πληροφορία, το σύστημα του ICAS δε θα επηρεαστεί σημαντικά. Προβλήματα φυσικά θα παρουσιαστούν στην περίπτωση που ο αριθμός των κόμβων που ψεύδονται είναι υπερβολικά μεγάλος (π.χ. > 50%).

3.3.2 Τιμολόγηση

Κάθε κόμβος στον μηχανισμό του ICARUS διαφοροποιείται σε τρεις διακριτές καταστάσεις ανάλογα με τον αριθμό των credits που διαθέτει. Αν ο αριθμός των credits που έχει ο κόμβος i τη χρονοθυρίδα k είναι $m_i^{(k)}$ και το κατώφλι των credits είναι St, τότε οι καταστάσεις ορίζονται ως

- Κατάσταση 0 – Ο κόμβος μπορεί να στείλει πακέτα οπουδήποτε: αν $m_i^{(k)} \geq S_t$.
- Κατάσταση 1 – Ο κόμβος μπορεί να στείλει πακέτα μόνο στους γείτονές του: αν $m_i^{(k)} < S_t$.
- Κατάσταση 2 – Ο κόμβος δεν μπορεί να στείλει πακέτα πουθενά : αν $m_i^{(k)} < 0$.

Ακολουθώντας τις παραπάνω καταστάσεις, ένας κόμβος που δεν διαθέτει credits δεν έχει δικαίωμα να αποστείλει πακέτα. Η κατάσταση του κάθε κόμβου ορίζεται από το ICAS που καταγράφει τα credits κάθε κόμβου, δίνοντας άδεια σε όποιους κόμβους έχουν

δικαίωμα να αποστείλουν πακέτα(δεχόμαστε ότι για να αποστείλει ένα πακέτο ο κάθε κόμβος πρέπει αναγκαστικά να πάρει την άδεια του ICAS πρώτα).

Η τιμολόγηση στο ICARUS βασίζεται στη φήμη κάθε κόμβου. Οι αμοιβές βασίζονται στο εκτιμώμενο μέσο ποσοστό συνδεσιμότητας των κόμβων. Πιο συγκεκριμένα, η αμοιβή $r_{i,z}$ (σε credits) του κόμβου i για την προώθηση του πακέτου z τη χρονοθυρίδα είναι

$$r_{i,z} = a + b \cdot \left[cp_i^{(k)} \right]^2 \qquad (3.8)$$

όπου *a*, *b* είναι σταθερές.

Για το συγκεκριμένο σύστημα που εξετάζεται επιλέξαμε *a* = 0.5 και *b* = 2.3. Αφού ο αποστολέας υποχρεούται να ανταμείψει όλους τους κόμβους που λειτούργησαν ως ενδιάμεσοι, το συνολικό κόστος $\cos t_{i,z}$ για το πακέτο *z* που στάλθηκε από τον κόμβο *i* υπολογίζεται ως

$$\cos t_{i,z} = \sum_{j \in N,\, j \in rl_{i,z}} r_{i,z} \qquad (3.9)$$

όπου $rl_{i,z}$ είναι μια λίστα όλων των κόμβων που προώθησαν το πακέτο *z.*

Στο ICARUS Οι ανταλλαγές των credits γίνονται αφού ένα πακέτο φτάσει στον προορισμό του και ο παραλήπτης ενημερώσει το ICAS για την επιτυχημένη αποστολή. Κατά συνέπεια, σε περίπτωση που ένα πακέτο δεν φτάσει στον προορισμό του ποτέ, το ICAS δεν ενημερώνεται ποτέ με συνέπεια να μην ανταμειφθούν οι κόμβοι που προώθησαν το πακέτο. Παρόλα αυτά, η ενημέρωση του ICAS από τον παραλήπτη θεωρείται ως η πιο ασφαλής προσέγγιση, αφού έτσι οι κακόβουλοι κόμβοι δεν μπορούν να ισχυριστούν για τον εαυτό τους ψευδώς ότι προώθησαν κάποιο πακέτο, ούτε απαιτούνται υπερβολικά πολλοί πόροι για την εξακρίβωση του κάθε ισχυρισμού.

Σε ορισμένες περιπτώσεις ωστόσο το ICARUS παρουσιάζει το πρόβλημα της υπερβολικής συσσώρευσης credits σε ιδιαίτερα συνεργάσιμους κόμβους. Για την αντιμετώπιση αυτού του ζητήματος, οι «πλούσιοι» κόμβοι χρεώνονται επιπλέον από το ICAS. Όμοια με τις αμοιβές, η επιπλέον χρέωση εξαρτάται από το τετράγωνο του *cp* και η χρέωση προσαρμόζεται ως

$$r_{i,z}^{adj} = \begin{Bmatrix} 1.3 \cdot r_{i,z}, & m_i^{(k)} > 2.5 \cdot m_o \\ 4.6 \cdot r_{i,z}, & m_i^{(k)} > 5.7 \cdot m_o \end{Bmatrix} \quad (3.10)$$

3.3.3 Εντοπισμός εγωιστών κόμβων

Όσων αφορά στον εντοπισμό των εγωιστών κόμβων ο ICARUS ακολουθεί παρόμοιο μηχανισμό με αυτό που χρησιμοποιεί και ο DARWIN. Πιο συγκεκριμένα, μεταξύ δύο κόμβων i και j, στην περίπτωση όπου ο j δεν θεωρείται εγωιστής ο i θα του προωθεί πάντα τα πακέτα του. Ωστόσο στην περίπτωση που ο j έχει θεωρηθεί εγωιστής ο i θα απορρίπτει τα πακέτα που του στέλνει με βάση την εκτιμώμενη πιθανότητα απόρριψης $edp_i^{(k)}$ [1][2].

Ο εντοπισμός των εγωιστών κόμβων επιτυγχάνεται με τους παρακάτω τρόπους:

- Το ICAS μεταδίδει περιοδικά μια καθολική λίστα με όλους τους αναγνωρισμένους κακόβουλους κόμβους. Αν ένας κόμβος εντοπίσει κάποιον γείτονά του σε αυτή τη λίστα, τότε τον προσθέτει στην δική του τοπική λίστα.
- Για κάθε κόμβο, αν η εκτιμώμενη πιθανότητα απόρριψης $edp_i^{(k)}$ του γείτονά του ξεπερνάει ένα προκαθορισμένο κατώφλι $edp_i^{(k)}$, τότε τον προσθέτει στην τοπική του λίστα.

Με βάση τα παραπάνω, ο μόνος τρόπος για να αφαιρεθεί ένας κόμβος από μια τοπική λίστα, είναι να ικανοποιεί και τις δύο παραπάνω απαιτήσεις.

3.3.4 Απομακρυσμένοι κόμβοι

Ένα σύνηθες πρόβλημα που παρουσιάζουν πολλοί από τους μηχανισμούς που μελετήσαμε στην ενότητα 1, και ιδιαιτέρως οι μηχανισμοί αμοιβής, έχει να κάνει με την αποτελεσματική αντιμετώπιση των απομακρυσμένων κόμβων ώστε να μην αντιμετωπίζουν προβλήματα αδικίας από το σύστημα. Αυτοί οι κόμβοι, εξαιτίας της θέσης τους δεν τους δίνεται η δυνατότητα να προωθήσουν πακέτα, με αποτέλεσμα εάν δεν αντιμετωπιστούν με ιδιαιτερότητα να παρουσιάζονται ως εγωιστές κόμβοι στο δίκτυο.

Ένας δίκαιος μηχανισμός οφείλει να φροντίσει για αυτό και να ενισχύει αυτούς τους κόμβους. Ο ICARUS ωστόσο δεν χρησιμοποιεί ακριβώς αυτή την προσέγγιση για την αποφυγή της δημιουργίας υπερβολικά "πλούσιων" κόμβων (σε credits) χωρίς να προωθούν κανένα πακέτο. Για την αντιμετώπιση αυτού του ζητήματος προσφέρει την δυνατότητα στους απομακρυσμένους κόμβους να αποστέλλουν δωρεάν τα πακέτα τους αλλά σε περίπτωση "ολικής χρεοκοπίας". Με αυτό τον τρόπο οι απομακρυσμένοι μπορούν να ικανοποιούν τις ανάγκες τους αλλά χωρίς να πλουτίσουν σε βάρος του δικτύου.

Στο ICARUS η βοήθεια που αναφέραμε παρέχεται από το ICAS. Οι απομακρυσμένοι κόμβοι εντοπίζονται από τον αριθμό των πακέτων που τους έχει ζητηθεί να προωθήσουν στο παρελθόν. Αν ο αριθμός αυτός είναι πολύ μικρός και η φήμη του κόμβου είναι καλή, τότε το ICAS αποφασίζει να τους ενισχύσει. Η ενίσχυση αυτή πραγματοποιείται αρχικά με ένα μικρό αριθμό από επιπλέον credits, ο οποίος εξαρτάται από μια σταθερή τιμή και την τιμή cp του κόμβου. Τα επιπλέον credits προσφέρονται περιοδικά μέχρι τα συνολικά credits του κόμβου να φτάσουν το όριο $m_{e,th}$. Θεωρούμε ότι

$$m_{e,th} = m_0 \cdot \left[cp_i^{(k)} \right]^2 \qquad (3.11)$$

Παρακάτω παρουσιάζεται ο ψευδοκώδικας βάσει του οποίου πραγματοποιείται η ενίσχυση των απομακρυσμένων κόμβων

1. **if** (i not selfish && $m_i^{(k)} \le m_{e,th}$ && $S_i^{(k)} \le k \cdot pf_2 \cdot cp_i^{(k)}$ && i not allowed to send packets for free) **then**
2. **if** ($S_i^{(k)} \le k \cdot pf_1 \cdot cp_i^{(k)}$) **then**
3. $m_i^{(k)} += 1.5 + m_{e,0} \cdot \left[cp_i^{(k)} \right]^2$;
4. **end if**
5. **else do**
6. $m_i^{(k)} += 2 + m_{e,0} \cdot \left[cp_i^{(k)} \right]^2$;
7. **end else**
8. **end if**
9. **If** (i not selfish && i in State 0 && $S_i^{(k)} \le k \cdot pf_2 \cdot cp_i^{(k)}$ && $m_i^{(k)} < 0$) **then**
10. i is allowed to send packets for free;
11. **end if**

Με $S_i^{(k)}$ συμβολίζουμε το συνολικό αριθμό πακέτων που έχουν σταλεί στον *i* για προώθηση μέχρι τη χρονοθυρίδα *k*. Προφανώς $S_i^{(k)} = \sum_{\forall j \in N_i} S_{ji}^{(k)}$. Ο υποστηρικτικός μηχανισμός του ICARUS ξεχωρίζει δύο κατηγορίες κόμβων: αυτούς που χρειάζονται

ενίσχυση για να αποκτήσουν τον απαιτούμενο αριθμό credits και αυτούς που δε διαθέτουν καθόλου credits, οπότε χρειάζονται δωρεάν αποστολή πακέτων. Η διαφοροποίηση αυτή επιτυγχάνεται μέσω δύο κατωφλίων pf_1 και pf_2.

Η πρώτη κατηγορία κόμβων ικανοποιεί τη συνθήκη $S_i^{(k)} \le k \cdot pf_1 \cdot cp_i^{(k)}$ και λαμβάνει επιπλέον credits μέχρι να εισέλθει στην κατάσταση 1 ή 2. Στην περίπτωση αυτή το ICAS επιτρέπει δωρεάν αποστολή πακέτων. Η δεύτερη κατηγορία απομακρυσμένων κόμβων, για την οποία ισχύει $S_i^{(k)} > k \cdot pf_1 \cdot cp_i^{(k)}$, λαμβάνει περισσότερα credits αλλά δεν της επιτρέπεται να στείλει πακέτα δωρεάν.

Βιβλιογραφία

[1]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.

[2]F. Milan, J. J. Jaramillo, R. Srikant, “Achieving cooperation in multihop wireless networks of selfish nodes”, Workshop on Game Theory for Networks (GameNets 2006), 2006.

[3]Δημήτρης Ε. Χαρίλας, "Τεχνικές Βέλτιστης Διαχείρισης Πόρων σε Ασύρματα Δίκτυα Επόμενης Γενιάς για Διασφάλιση Ποιότητας Υπηρεσίας", Διδακτορική Διατριβή, Μάιος 2012

4. ΥΛΟΠΟΙΗΣΗ ΥΒΡΙΔΙΚΟΥ ΜΗΧΑΝΙΣΜΟΥ ΚΑΙ ΠΡΟΣΟΜΟΙΩΣΗ

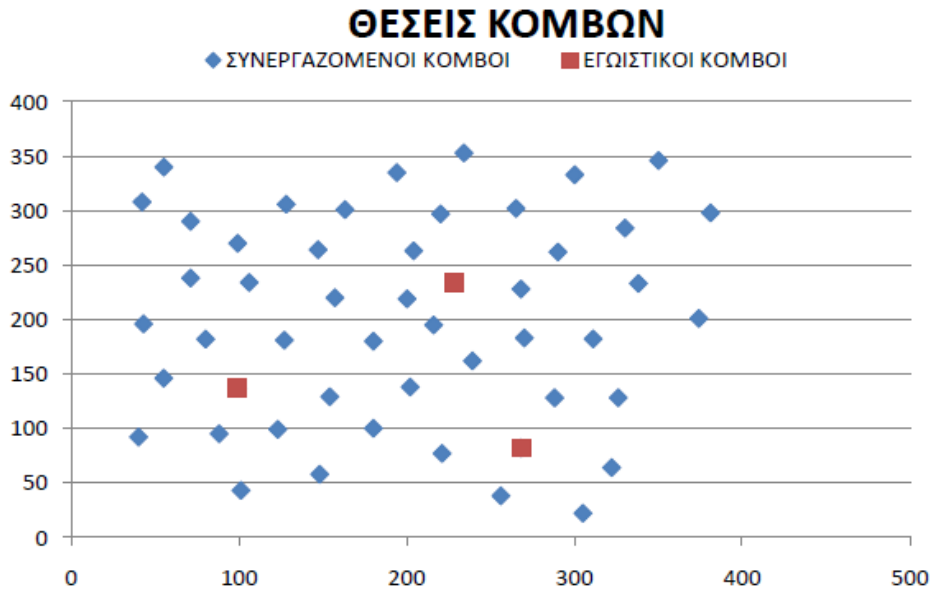
4.1 Στοιχεία Προσομοίωσης

4.1.1 Εισαγωγή

Σε αυτή την ενότητα της διπλωματικής θα παρουσιαστεί αναλυτικά ο τρόπος υλοποίησης του υβριδικού μηχανισμού του οποίου η λειτουργία αναλύθηκε στην ενότητα 3. Όπως ειπώθηκε και προηγούμενα είναι ένας μηχανισμός που συνδυάζει το σύστημα φήμης και πίστωσης παράλληλα για μεγαλύτερη αποτελεσματικότητα και αυξημένη ποιότητα υπηρεσίας στο δίκτυο. Εν συνεχεία, η αποτελεσματικότητα του μηχανισμού δοκιμάζεται μέσω προσομοίωσης της οποίας οι συνθήκες περιγράφονται.

4.1.2 Τοπολογία Προσομοίωσης

Για τη μελέτη της αποτελεσματικότητας του υβριδικού αλγορίθμου πραγματοποιήθηκε μια προσομοίωση σε μια τοπολογία 50 κόμβων οι οποίοι είναι διεσπαρμένοι σε ένα χώρο με εμβαδόν 400μx400μ, φροντίζοντας να έχουμε πιο πυκνή κατανομή κόμβων στο κέντρο παρά στη περιφέρεια του χώρου. Γειτονικός κόμβος θεωρείται όποιος κόμβος απέχει από λιγότερο από 60 μέτρα από κάποιον άλλο.



4.2 Υλοποίηση υβριδικού μηχανισμού

4.2.1 Σύντομη περιγραφή αλγορίθμου

Για να γίνει ξεκάθαρο το πως λειτουργεί ο αλγόριθμος, αρχικά θα γίνει μια σύντομη περιγραφή της λογικής λειτουργίας της προσομοίωσης.

Στο πρώτο στάδιο του αλγορίθμου ο κάθε κόμβος εκτελεί μια διαδικασία αρχικοποίησης κατά την οποία ευρίσκει τους γειτονικούς αλλά και τους μακρινούς του κόμβους. Σε αυτό το σημείο αρχίζει η διαδικασία της αποστολής πακέτων από κάθε κόμβο σε τυχαίους παραλήπτες. Για την ρεαλιστική προσέγγιση της διαδικασίας αυτής ο κάθε κόμβος ανεξάρτητα από τους υπολοίπους αποστέλλει το κάθε πακέτο ανά μια συγκεκριμένη περίοδο που ορίστηκε στα 2s. Εν συνεχεία, ο κάθε κόμβος που λαμβάνει το πακέτο που προοριζόταν για αυτόν, αποστέλλει ένα μήνυμα επιβεβαίωσης ack το οποίο ακολουθώντας την ίδια διαδρομή επιστρέφει στον αρχικό αποστολέα. Με αυτό τον τρόπο μπορούμε να επιβεβαιώσουμε και να καταμετρήσουμε τα πακέτα που εστάλησαν επιτυχώς από κάθε κόμβο.

Για την καλύτερα ρεαλιστική προσέγγιση ενός πραγματικού συστήματος, η προσομοίωση έγινε σε Java και πιο συγκεκριμένα με την βοήθεια των threads(δηλαδή διεργασιών που έχουν την δυνατότητα να τρέχουν παράλληλα πάνω στο σύστημα μας, πραγματοποιώντας διαφορετικές ενέργειες μεταξύ τους). Επίσης, για την αποτελεσματική χρονική δρομολόγηση των διεργασιών των threads χρησιμοποιήθηκαν και timers.

Στην συνέχεια της ενότητας θα γίνει μια περιγραφή των βασικών κλάσεων που έχουν σχεδιαστεί στην Java και οι οποίες περιγράφουν τα βασικότερα κομμάτια του δικτύου (κόμβοι, ζεύξεις, πακέτα, κτλ). Κατόπιν θα προχωρήσουμε στην ανάλυση των λειτουργικών κλάσεων του αλγορίθμου που πραγματοποιούν τις λειτουργίες του δικτύου (αποστολή πακέτων, προώθηση πακέτων, αποστολή ack, κτλ)

4.2.1.1 Περιγραφή βασικών κλάσεων του αλγορίθμου

Αρχικά θα γίνει μια περιγραφή των βασικών κλάσεων που έχουν σχεδιαστεί στην Java και οι οποίες περιγράφουν τα βασικότερα κομμάτια του δικτύου (κόμβοι, ζεύξεις, πακέτα, κτλ).

Packet: αυτή η κλάση υλοποιεί ένα πακέτο και περιλαμβάνει τα βασικά στοιχεία που χαρακτηρίζουν ένα πακέτο. Δηλαδή το id, το μέγεθος, τον αποστολέα, τον παραλήπτη, τον επόμενο κόμβο στον οποίο προωθείται, τον αριθμό των συνολικών βημάτων που έχει κάνει, την πληροφορία για το εάν βρίσκεται ακόμα στο δίκτυο ή έχει απορριφθεί.

Ack: αυτή η κλάση υλοποιεί ένα πακέτο επιβεβαίωσης και περιλαμβάνει τα βασικά στοιχεία που το χαρακτηρίζουν. Αυτά είναι ακριβώς τα ίδια με αυτά ενός κανονικού πακέτου όπως αναφέρονται παραπάνω και για αυτό τον λόγο δεν επαναλαμβάνονται.

Link: αυτή η κλάση υλοποιεί μια ζεύξη και περιλαμβάνει τα βασικά στοιχεία που την χαρακτηρίζουν. Δηλαδή το id, το μήκος της ζεύξης σε μέτρα, τα δύο άκρα της ζεύξης, την πληροφορία αν αυτή η ζεύξη υπάρχει (δηλαδή δύο κόμβοι γειτονεύουν μεταξύ τους) και τέλος το downlink που αποθηκεύουμε τα πακέτα που ανά πάσα στιγμή βρίσκονται επί της ζεύξης και το uplink στο οποίο αποθηκεύουμε τα ack που βρίσκονται στην ζεύξη.

Node: αυτή η κλάση υλοποιεί ένα κόμβο και περιλαμβάνει τα βασικά στοιχεία που τον χαρακτηρίζουν καθώς και κάποιες διαδικασίες των οποίων θα επεξηγηθεί η λειτουργία τους. Τα βασικά μεγέθη που χαρακτηρίζουν τον κόμβο είναι αρκετά και απλά θα αναφερθούν τα βασικότερα αυτών. Το id, την θέση του κόμβου(συντεταγμένη x και y) μετρητής που κρατά αν ένας κόμβος είναι εγωιστής ή όχι, την απόσταση από τους γείτονες, όσα στοιχεία σχετίζονται με τους μηχανισμούς φήμης, κτλ. Οι διαδικασίες που περιέχει η κλάση Node μέσα της είναι οι *addpackettolink, addacktolink, removepacket, removeack*. Η λειτουργία της *addpackettolink* είναι να παίρνει σαν ορίσματα τον κόμβο και το id του, την ζεύξη που συνδέει δύο κόμβους, ένα πακέτο και τελικά να τοποθετεί το εν λόγω πακέτο μέσα στην ζεύξη που συνδέει τους δύο κόμβους. Δηλαδή τοποθετώντας το στην downlink που αναφέρθηκε παραπάνω του link. Ακριβώς την ίδια διαδικασία επιτελεί η *addacktolink* με την μόνη διαφορά ότι αντί για πακέτο τοποθετείται πλέον ack στην ζεύξη. Τέλος η *removepacket και η removeack* κάνει ακριβώς την αντίστροφη διαδικασία από αυτή του *addpackettolink* δηλαδή αφαιρεί το πακέτο από την τυχούσα ζεύξη κάθε φορά.

Data: αυτή η κλάση περιέχει κάποιες προκαθορισμένες πληροφορίες που ορίζουν την δομή του δικτύου. Πιο συγκεκριμένα ορίζονται οι κόμβοι που είναι εγωιστές, κακόβουλοι, η θέση των κόμβων(συντεταγμένες x και y) και οι προορισμοί των πακέτων που επιλέγει κάθε κόμβος.

Statistics: αυτή η κλάση απλά τυπώνει τα αποτελέσματα της προσομοίωσης σε ένα αρχείο .txt

ICAS: αυτή η κλάση υλοποιεί την ελεγκτική κεντρική μονάδα ICAS και τις διαδικασίες που επιτελεί όπως αυτές αναφέρθηκαν στην ενότητα 3. Αναλυτικότερα περιέχει την *ypol_credit* η οποία υπολογίζει τα credits που δίνονται σε κάθε κόμβο που προώθησε ένα πακέτο και αυτά που αφαιρούνται από τον αποστολέα. Επίσης η *ypol_cp* που υπολογίζει την cp_j όπως αυτή ορίστηκε στην ενότητα 3.Η is_egoistikos ελέγχει μια τιμή κατωφλίου και βάσει αυτής συμπεραίνει εάν ένας κόμβος είναι εγωιστής. Τέλος υπάρχουν *οι extra* και *extra_plus* που δίνουν τα δωρεάν credits στους ακριανούς κόμβους σε μικρά ή μεγάλα χρονικά διαστήματα καθώς και η *mesi_timi_credits* που βρίσκει τα credits των εγωιστών και συνεργατικών κόμβων.

Routing: αυτή η κλάση ορίζει τις συναρτήσεις για προώθηση των πακέτων και ack. Θα γίνει ανάλυση τους στην συνέχεια της ενότητας.

Decisions: αυτή η κλάση περιέχει συναρτήσεις που σχετίζονται με τις αποφάσεις που λαμβάνουν οι κόμβοι. Δηλαδή οι συναρτήσεις που την απαρτίζουν είναι οι *decide_to_cheat, decide_to_forward, find_alternate_relay.* Όπως γίνεται προφανές και από τα ονόματα που τους έχουν δοθεί σχετίζονται με τις αποφάσεις εγωιστικής συμπεριφοράς, προώθησης και εύρεσης εναλλακτικής διαδρομής αντίστοιχα.

Darwin: αυτή η κλάση σχετίζεται με τα κομμάτια του υβριδικού αλγορίθμου που αφορούν διεργασίες του αλγορίθμου Darwin. Πιο συγκεκριμένα υπάρχουν οι εξής διαδικασίες: *ypol_clogou, ypol_ownp, ypol_ownq, ypol_pdarwin.* Οι οποίες υπολογίζουν αντιστοίχως το c_{ij}, την πιθανότητα q_{-i}, την αντιλαμβανόμενη πιθανότητα του κόμβου, την πιθανότητα q_i και την πιθανότητα Darwin. Όλες οι παραπάνω πιθανότητες έχουν οριστεί στην ενότητα 1.2.7.

4.2.1.2 Περιγραφή λειτουργικών κλάσεων του αλγορίθμου

Σε αυτό το σημείο θα γίνει μια περιγραφή των λειτουργικών κλάσεων που και πάλι έχουν σχεδιαστεί σε Java και επιτελούν τις λειτουργίες των κόμβων(αποστολή πακέτων, προώθηση πακέτων, προσθήκη πακέτων στις ζεύξεις, κτλ).

MainIndex: αυτή η κλάση αποτελεί την βάση εκκίνησης του αλγορίθμου καθώς πραγματοποιεί την αρχικοποίηση/δημιουργία κάθε κόμβου του συστήματος και των μεγεθών που χρησιμοποιούνται στον υπόλοιπο κώδικα. Αφότου πραγματοποιηθεί η αρχικοποίηση κάθε κόμβου, δημιουργείται ένα thread για καθέναν από αυτούς το οποίο και επιτελεί τις περιοδικές διαδικασίες για τις οποίες είναι υπεύθυνος ο κάθε κόμβος.

Init: αυτή η κλάση καλείται από την ***MainIndex*** και ολοκληρώνει την αρχικοποίηση κάθε κόμβου, ευρίσκοντας τους κοντινούς και μακρινούς γείτονες αυτού. Σε αυτή την κλάση ορίζονται επίσης τα Timers για την περιοδική πραγματοποίηση των λειτουργιών κάθε κόμβου(**SendPack**, **CalcICAS**, **CalcNewKatastasi**, **CalcFree**, **CalcDarwin**, **pStatRnd**).

SendPack: αυτή η κλάση ευθύνεται για την δημιουργία ενός πακέτου το οποίο αποστέλλει ο κάθε κόμβος και εν συνεχεία τοποθετείται στην αντίστοιχη ζεύξη καλώντας την *addpackettolink* από την κλάση **Node**(κοίτα ενότητα 4.2.2). Η ***SendPack*** καλείται περιοδικά από κάθε κόμβο.

dropPack: αυτή η κλάση καλείται για την αφαίρεση ενός πακέτου από μια ζεύξη και χρησιμοποιείται αφότου περάσει κάποιο χρονικό διάστημα από την είσοδο ενός πακέτου σε μια ζεύξη. Πιο συγκεκριμένα, η ***dropPack*** καλεί αρχικά την *removepacket* από την κλάση **Node**(κοίτα ενότητα 4.2.2). Εν συνεχεία, ελέγχει εάν το πακέτο έφτασε στον προορισμό του στην οποία περίπτωση στέλνει το πακέτο επιβεβαίωσης ack στον αποστολέα. Σε περίπτωση που ο τρέχων κόμβος δεν αποτελεί τον προορισμό του πακέτου, φροντίζει να το προωθήσει καλώντας την *addpackettolink* από την κλάση **Node**(κοίτα ενότητα 4.2.2).

dropAck: αυτή η κλάση καλείται για την αφαίρεση ενός πακέτου επιβεβαίωσης ack από μια ζεύξη και χρησιμοποιείται αφότου περάσει κάποιο χρονικό διάστημα από την είσοδο ενός πακέτου σε μια ζεύξη. Πιο συγκεκριμένα, η ***dropAck*** καλεί αρχικά την *removeack* από την κλάση **Node**(κοίτα ενότητα 4.2.2). Εν συνεχεία, ελέγχει εάν το πακέτο επιβεβαίωσης ack έφτασε στον προορισμό του. Σε περίπτωση που ο τρέχων κόμβος δεν αποτελεί τον προορισμό του ack, φροντίζει να το προωθήσει καλώντας την *addacktolink* από την κλάση **Node**(κοίτα ενότητα 4.2.2).

CalcICAS: αυτή η κλάση καλείται περιοδικά από κάθε κόμβο υπολογίζοντας και ενημερώνοντας τους μετρητές που κρατάει το ICAS για κάθε κόμβο. Οι μετρητές αυτοί αναφέρθηκαν παραπάνω στην ενότητα 3.

CalcDarwin: αυτή η κλάση καλείται περιοδικά από κάθε κόμβο υπολογίζοντας και ενημερώνοντας τους μετρητές που υπολογίζονται και στον μηχανισμό DARWIN(κοίτα ενότητα 1.2.7) και των οποίων γίνεται χρήση από τον ICARUS.

CalcFree: αυτή η κλάση καλείται περιοδικά από κάθε κόμβο υπολογίζοντας τις δωρεάν χρηματικές μονάδες(credits) που δίνονται σε κάθε κόμβο που τις δικαιούται.

CalcNewKatastasi: αυτή η κλάση καλείται περιοδικά από κάθε κόμβο υπολογίζοντας την κατάσταση "δυνατότητας αποστολής" στην οποία βρίσκεται ο κάθε κόμβος. Πιο συγκεκριμένα, οι καταστάσεις είναι τρεις και είχαν αναφερθεί παραπάνω στην ενότητα 3.3.2. Συνοπτικά ξαναπαρουσιάζονται παρακάτω.

- Κατάσταση 0 – Ο κόμβος μπορεί να στείλει πακέτα οπουδήποτε: αν $m_i^{(k)} \geq S_t$.
- Κατάσταση 1 – Ο κόμβος μπορεί να στείλει πακέτα μόνο στους γείτονές του: αν $m_i^{(k)} < S_t$.
- Κατάσταση 2 – Ο κόμβος δεν μπορεί να στείλει πακέτα πουθενά : αν $m_i^{(k)} < 0$.

pStatRnd: αυτή η κλάση καλείται περιοδικά από κάθε κόμβο και ευθύνεται για την εκτύπωση πληροφοριών κίνησης του δικτύου για την αποτελεσματική παρακολούθηση της λειτουργίας του δικτύου.

KillThread: αυτή η κλάση καλείται μια φορά για κάθε κόμβο μετά από ένα προεπιλεγμένο χρονικό διάστημα (έγινε υλοποίηση με Timers) για τον τερματισμό του κάθε TimerTask που καλείται από κάθε κόμβο. Κατά συνέπεια, η **KillThread** ορίζει την ολοκλήρωση της προσομοίωσης και ο χρόνος μετά από τον οποίο καλείται αποτελεί τον χρόνο προσομοίωσης.

4.2.2 Αναλυτική περιγραφή λειτουργίας αλγορίθμου

Σε αυτή την ενότητα θα γίνει η ανάλυση του τρόπου λειτουργίας του αλγορίθμου, όπως υλοποιήθηκε σε περιβάλλον Java, στα πλαίσια υλοποίησης του για τις ανάγκες της προσομοίωσης.

Το σημείο εκκίνησης του αλγορίθμου, όπως αναφέρθηκε και στην ενότητα 4.2.1, αποτελεί η κλάση MainIndex.java. Η συγκεκριμένη κλάση αποτελεί και σημείο αναφοράς όλου του αλγορίθμου καθώς σε αυτή γίνεται ο ορισμός κάποιων καθολικών σταθερών που ρυθμίζουν τα στοιχεία προσομοίωσης του δικτύου. Παρακάτω παρατίθενται αυτές οι σταθερές όπως ορίστηκαν στον αλγόριθμο.

```
final static int node_num = 50;   // o synolikos ari8mos twn kombwn
final static int link_capac=20;    // i xwritikotita tis ka8e zeu3is se
paketa
final static int simul_time=320; // o synolikos xronos prosomoiwsis se
deuterolepta
final static int snd_pkt_period=2; // i periodos apostolis paketwn se
deuterolepta
final static int CalcICAS_period=10; // i periodos ypologismou stoixeiwn
ICAS  se deuterolepta
final static int CalcDarwin_period=10; // i periodos ypologismou Darwin se
deuterolepta
final static int CalcFree_period=10; // i periodos ypologismou free credits
se deuterolepta
final static int CalcStatRnd=20;  //i periodos ypologismou twn statistikwn
tou diktyou se deuterolepta
final static int CalcNewKatastasi=10;    //i periodos ypologismou tis
katastasis ka8e kombou se deuterolepta
```

Η βασική λειτουργία της MainIndex ωστόσο είναι η δημιουργία και αρχικοποίηση κάθε κόμβου του δικτύου δίνοντας κάποιες αρχικές τιμές στα πεδία της κλάσης Node. Εν συνεχεία, πραγματοποιείται η ουσιαστική έναρξη της προσομοίωσης με την δημιουργία των threads που το καθένα από αυτά αντιπροσωπεύει ένα κόμβο και λειτουργεί παράλληλα και ανεξάρτητα από τα υπόλοιπα. Ο κώδικας υλοποίησης των threads παρατίθεται παρακάτω.

```
Init ni[] = new Init[node_num];
for(int i=0; i<node_num; i++){
      doulos ="Node"+i;
      ni[i] = new Init(doulos,i,n,links,hybrid, stats);
      Thread t1 = new Thread(ni[i]);
      t1.start();
}
```

Όπως φαίνεται στο παραπάνω σύντομο κομμάτι κώδικα, δημιουργούνται τόσα thread όσοι είναι και οι κόμβοι. Κατά την δημιουργία, αυτών των threads φροντίζεται να μεταφερθούν οι πληροφορίες αρχικοποίησης που πραγματοποιήθηκαν προηγούμενα στην MainIndex.java. Αυτά τα threads είναι της κλάσης Init.java της οποίας η λειτουργία θα μελετηθεί στην συνέχεια. Δημιουργείται επίσης και ένα ακόμα Thread του οποίου η μοναδική λειτουργία είναι η περιοδική (με την βοήθεια της scheduleAtFixedRate) εκτύπωση κάποιων στατιστικών στοιχείων που συλλέγονται κατά την διάρκεια εκτέλεσης του αλγορίθμου με την βοήθεια της pStatRnd.

```
Timer mytim = new Timer();
mytim.scheduleAtFixedRate(new pStatRnd(n,n[0].round,hybrid,stats), 10000,
(MainIndex.CalcStatRnd*1000));
triggerototo.schedule(newkillCalcThrd(mytim),((MainIndex.simul_time+5)*1000
));
```

Η Init.java ολοκληρώνει την διαδικασία αρχικοποίησης του κάθε κόμβου, ευρίσκοντας τους κοντινούς και μακρινούς γείτονες αυτών. Σε περίπτωση που ευρεθεί ένας γειτονικός κόμβος φροντίζεται να δημιουργηθεί και η αντίστοιχη ζεύξη μεταξύ των δύο κόμβων.

```
if(links[j][nmb].active == false){
	links[nmb][j].id=""+nmb+"-"+j;
	links[nmb][j].transmitterid=nmb;
	links[nmb][j].receiverid=j;
	links[nmb][j].distnc=distance;
	links[nmb][j].active = true;
	System.out.println("Link"+nmb+" ID: '"+ links[nmb][j].id+ "'  from "
+ links[nmb][j].transmitterid + " to "+ links[nmb][j].receiverid + "
created");
}
```

Η συνθήκη `if(links[j][nmb].active == false)` φροντίζει να αποφεύγεται η δημιουργία της ίδιας ζεύξης δύο φορές. Για παράδειγμα μεταξύ των κόμβων 1 και 2, χωρίς την ύπαρξη αυτής της συνθήκης θα δημιουργούνταν μια ζεύξη 1-2 και μια ζεύξη 2-1, γεγονός και το οποίο αποφεύγεται με την παραπάνω συνθήκη. Στην συνέχεια, αρχικοποιούνται τα πεδία της κλάσης Link.

Η βασική βέβαια λειτουργία της Init.java είναι ο χρονικός προγραμματισμός των λειτουργιών κάθε κόμβου με την βοήθεια των timers.

```
Timer teme = new Timer();
Timer teme2 = new Timer();
Timer teme3 = new Timer();
Timer teme4 = new Timer();
Timer teme5 = new Timer();
teme.scheduleAtFixedRate(new SendPack(name,nmb,n, links, hybrid, stats),
4000, (MainIndex.snd_pkt_period*1000));
teme2.scheduleAtFixedRate(new CalcICAS(n,nmb,hybrid, stats), 7000,
(MainIndex.CalcICAS_period*1000));
teme3.scheduleAtFixedRate(new CalcDarwin(n,nmb,hybrid, stats), 7000,
(MainIndex.CalcDarwin_period*1000));
```

```
teme4.scheduleAtFixedRate(new CalcFree(n,nmb,hybrid, stats), 7000,
(MainIndex.CalcFree_period*1000));
teme5.scheduleAtFixedRate(new CalcNewKatastasi(n,nmb,hybrid), 7000, 500)

Timer triggo = new Timer();
Timer triggitty = new Timer();
Timer trigger = new Timer();
Timer triggero = new Timer();
Timer triggerototo = new Timer();
triggo.schedule(new
killThread(teme,n,stats,nmb),(MainIndex.simul_time*1000));
triggitty.schedule(new killCalcThrd(teme2), (MainIndex.simul_time*1000));
trigger.schedule(new killCalcThrd(teme3), (MainIndex.simul_time*1000));
triggero.schedule(new killCalcThrd(teme4), (MainIndex.simul_time*1000));
triggerototo.schedule(new killCalcThrd(teme5),(MainIndex.simul_time*1000));
```

Στο πρώτο κομμάτι του παραπάνω κώδικα γίνεται ο χρονικός προγραμματισμός των διαδικασιών που επιτελεί ο κάθε κόμβος ενώ στην συνέχεια υπάρχουν κάποια timers που φροντίζουν ώστε να καταργηθούν οι περιοδικές κλήσεις των διαδικασιών στο τέλος του χρόνου προσομοίωσης(*simul_time* όπως αυτή είχε αρχικοποιηθεί στην MainIndex.java). Πιο συγκεκριμένα χρησιμοποιώντας την scheduleAtFixedRate υπάρχει η δυνατότητα να ορίσουμε μια κλάση TimerTask η οποία θα καλείται ανά μια προκαθορισμένη περίοδο, εκτελώντας με αυτό τον τρόπο κάποιες ενέργειες ανά σταθερή περίοδο, κάνοντας την ιδανική για την προσομοίωση του δικτύου. Παρακάτω παρατίθεται η δομή των ορισμάτων της scheduleAtFixedRate :

scheduleAtFixedRate(διαδικασία η οποία εκτελείται από την scheduleAtFixedRate, αρχική καθυστέρηση πριν την εκκίνηση της περιοδικής κλήσης της διαδικασίας, βήμα περιόδου επανάληψης της κληθείσας διαδικασίας).

Αυτά τα TimerTasks ουσιαστικά αποτελούν κάποιες διαδικασίες που απλά έχουν μια συγκεκριμένη δομή χωρίς κάποια άλλη ιδιαιτερότητα όσων αφορά στην λειτουργία τους. Οι διαδικασίες που καλούνται είναι οι λειτουργικές κλάσεις που ορίστηκαν στην 4.2.1.2(SendPack, CalcFree, CalcICAS, CalcDarwin, killThread) και θα αναλυθούν περαιτέρω, στην πορεία.

Η SendPack αποτελεί τον κορμό της προσομοίωσης του δικτύου καθώς επιτελεί την βασικότερη λειτουργία στο δίκτυο, δηλαδή την αποστολή πακέτων από έναν κόμβο σε κάποιον άλλο. Η κλήση της γίνεται ανά τακτά χρονικά διαστήματα και από όλους τους κόμβους όπως θα συνέβαινε σε ένα πραγματικό δίκτυο. Κατά την έναρξη της, γίνεται ένας βασικός έλεγχος για την δυνατότητα που έχει κάθε κόμβος να στείλει ένα πακέτο ή όχι.

```
btemp = (Boolean) hybrid.dorean.get(nmb);

if ((n[nmb].katastasi == 0)|| ((n[nmb].katastasi == 1) &&
(n[nmb].Spackets.containsKey(n[nmb].target_node_id)))|| (btemp == true))
```

Όπως φαίνεται και από το παραπάνω κομμάτι κώδικα, δυνατότητα αποστολής πακέτου έχουν τρεις κατηγορίες κόμβων:

i. Αυτοί που σύμφωνα με τις καταστάσεις τιμολόγησης που αναφέρθηκαν στην ενότητα 3.3.2, ανήκουν στην πρώτη κατηγορία και έχουν την δυνατότητα να στείλουν πακέτα σε οποιοδήποτε κόμβο του δικτύου.
ii. Αυτοί που σύμφωνα με τις καταστάσεις τιμολόγησης που αναφέρθηκαν στην ενότητα 3.3.2, ανήκουν στη δεύτερη κατηγορία και έχουν την δυνατότητα να στείλουν πακέτα μονάχα στους γείτονες τους.
iii. Αυτοί που έχουν την δυνατότητα να αποστέλλουν πακέτα δωρεάν, αφού τους έχει δοθεί ανάλογη άδεια από τον μηχανισμό ICAS.

Στην συνέχεια, και αφού αρχικοποιηθεί το πακέτο(δίνοντας αρχικές τιμές σε όλα τα πεδία της κλάσης Packet.java), πραγματοποιείται η δρομολόγηση του πακέτου για τον προορισμό που έχει επιλεχθεί με την βοήθεια μια συνάρτησης της βασικής κλάσης Routing.java.

```
Packet packetTest = new Packet();
packetTest.id = nmb +"_ in Round " +n[nmb].round;
packetTest.src_node = nmb;
packetTest.dst_node = n[nmb].target_node_id[rnd / allagi_proorismou];
packetTest.previous_node = nmb;
packetTest.total_hops = 0;
packetTest.dropped = false;
packetTest.received = false;
packetTest.forwarded_nodes = new ArrayList();
packetTest.current_node =
route.FindNextNode(n,packetTest,MainIndex.node_num,nmb, nmb);
```

Εν τέλει, με την ολοκλήρωση της SendPack τοποθετείται το νεοδημιουργηθέν πακέτο στην τυχούσα ζεύξη που του αναλογεί.

```
if(links[nmb][destNode].id != null || links[destNode][nmb].id != null){
     Link linkUsed = (links[nmb][destNode].id != null) ?
links[nmb][destNode] : links[destNode][nmb];
      n[nmb].addpackettolink(n, nmb, links, packetTest ,
packetTest.current_node,hybrid,stats, linkUsed);
      System.out.println(" Packet  " + packetTest.id + " is put into link
"+linkUsed.id);
}
```

Η τοποθέτηση του κάθε πακέτου στην αντίστοιχη ζεύξη πραγματοποιείται με την βοήθεια της addpackettolink που λαμβάνει σαν ορίσματα τον κόμβο και το id του, την ζεύξη που συνδέει δύο κόμβους, το νεοδημιουργηθέν πακέτο και τελικά το τοποθετεί μέσα στην ζεύξη που συνδέει τους δύο κόμβους. Σαν όρισμα επίσης λαμβάνει τις πληροφορίες του δικτύου που έχουν οριστεί μέχρι αυτό το χρονικό σημείο, ώστε να γίνει η μεταφορά τους στην επόμενη κλάση που καλείται από τον κώδικα, για να αποφευχθεί η απώλεια δεδομένων.

Η addpackettolink αποτελεί συνάρτηση της Node.java όπως αναφέρθηκε και στην ενότητα 4.2.2.1. Λαμβάνοντας σαν όρισμα το πακέτο που απαιτείται να τοποθετηθεί στην ζεύξη, προσθέτει, με προϋπόθεση να μην είναι γεμάτη η ζεύξη, το εν λόγω πακέτο στο πεδίο downlink της ζεύξης με πιθανότητα 99.9% λαμβάνοντας έτσι υπόψη της πολύ μικρή πιθανότητα να χαθεί το πακέτο.

```
if(linkUsed.downlink.size()<MainIndex.link_capac) {
if (prob()<9990){          //e3etazw thn pithanothta na mhn stalei to paketo
       linkUsed.downlink.add(packetTest);
}
}
```

Αφότου το πακέτο προστεθεί στη ζεύξη, με την βοήθεια ενός Timer ρυθμίζεται η κλήση της dropPack ώστε να αφαιρεθεί το πακέτο από την ζεύξη μετά από ένα συγκεκριμένο χρονικό διάστημα που ορίζεται ως trans_time. Αυτός ο χρόνος αντιστοιχεί στον χρόνο που στην πραγματικότητα απαιτείται ώστε να μεταφερθεί ένα πακέτο μέσα από την ζεύξη, ο οποίος και εύκολα υπολογίζεται γνωρίζοντας την απόσταση μεταξύ δύο κόμβων αλλά και την ταχύτητα διάδοσης.

```
Timer teme = new Timer();
teme.schedule(new
dropPack(u,nod1,l,packetTest,dst,slo,pro,linkUsed),trans_time);
```

Η πρωταρχική λειτουργία της dropPack είναι, όπως αναφέρθηκε και στην ενότητα 4.2.2.1, όπου και περιγράφεται η Node.java, να καλεί την removepacket που απλά αφαιρεί το τυχών πακέτο από το downlink της ζεύξης. Στην συνέχεια, πραγματοποιούνται δύο διαφορετικές διαδικασίες, που σχετίζονται με το εάν ο κόμβος που έλαβε το πακέτο είναι ο παραλήπτης του παρόντος πακέτου ή απλά ένας από τους ενδιάμεσους κόμβους της διαδρομής του πακέτου.

Στην περίπτωση όπου ο παρών κόμβος αποτελεί απλά έναν κόμβο του δικτύου και ενδιάμεσο στη διαδρομή από τον αποστολέα στον παραλήπτη του πακέτου, φροντίζει να βρει τον επόμενο κόμβο στην διαδρομή του πακέτου, με την βοήθεια και πάλι της κλάσης Routing.java, και να τοποθετήσει το πακέτο στην νέα ζεύξη πλέον προς τον καινούργιο προορισμό του πακέτου.

```
Link linkUsedEFED = (links[dest][dest2].id != null) ? links[dest][dest2] :
links[dest2][dest];
n[nmb].addpackettolink(n,nmb,links,packetTest,packetTest.current_node,hybri
d,stats, linkUsedEFED);
```

Η παραπάνω διαδικασία όπως είναι φανερό επαναλαμβάνεται έως ότου το πακέτο φτάσει και πάλι στην dropPack, αυτή την φορά όμως σε κόμβο που αποτελεί τον τελικό προορισμό του πακέτου. Σε αυτή την περίπτωση έχουμε μια επιτυχημένη άφιξη ενός πακέτου στον τελικό του προορισμό. Ο παραλήπτης φροντίζει τότε να δημιουργήσει ένα πακέτο επιβεβαίωσης ack(αρχικοποιώντας τα πεδία της κλάσης Ack.java).

```
Ack ackTest = new Ack();
ackTest.id = packetTest.id;
ackTest.packet_dst_node = packetTest.dst_node;
ackTest.packet_src_node = packetTest.src_node;
ackTest.packet_total_hops = packetTest.total_hops;
ackTest.packet_forwarded_nodes = packetTest.forwarded_nodes;
```

```
ackTest.total_hops = 0;
ackTest.current_node = route.FindNextNode(n,
ackTest,MainIndex.node_num,dest);
ackTest.dropped = false;
ackTest.received = false;
ackTest.ack_forwarded_nodes = new ArrayList();
```

Αυτό το πακέτο επιβεβαίωσης αποστέλλεται στον αποστολέα του πακέτου καθώς και ένα δεύτερο ack το οποίο απλά ενημερώνει το ICAS για την επιτυχημένη αποστολή του πακέτου ώστε να γίνουν οι απαραίτητοι υπολογισμοί των credit σε κάθε κόμβο.

```
int dest3 = ackTest.current_node;
Link linkUsedAc = (links[dest][dest3].id != null) ? links[dest][dest3] :
links[dest3][dest];
n[dest].addACKtolink(n, ackTest.packet_src_node, links, ackTest,
ackTest.current_node,hybrid, stats, packetTest, linkUsedAc);
```

Η τοποθέτηση του κάθε πακέτου επιβεβαίωσης στην αντίστοιχη ζεύξη πραγματοποιείται με την βοήθεια της addacktolink που λαμβάνει σαν ορίσματα τον κόμβο και το id του, την ζεύξη που συνδέει δύο κόμβους, το νεοδημιουργηθέν πακέτο επιβεβαίωσης και τελικά το τοποθετεί μέσα στην ζεύξη που συνδέει τους δύο κόμβους. Σαν όρισμα επίσης λαμβάνει τις πληροφορίες του δικτύου που έχουν οριστεί μέχρι αυτό το χρονικό σημείο, ώστε να γίνει η μεταφορά τους στην επόμενη κλάση που καλείται από τον κώδικα, για να αποφευχθεί η απώλεια δεδομένων.

Η addacktolink αποτελεί συνάρτηση της Node.java όπως αναφέρθηκε και στην ενότητα 4.2.2.1. Λαμβάνοντας σαν όρισμα το πακέτο επιβεβαίωσης (ackTest) που απαιτείται να τοποθετηθεί στην ζεύξη, προσθέτει το εν λόγω πακέτο στο πεδίο uplink της ζεύξης με πιθανότητα 99.9% λαμβάνοντας έτσι υπόψη της πολύ μικρή πιθανότητα να χαθεί το πακέτο.

```
if (prob()<9990){          //e3etazw thn pithanothta na mhn stalei to paketo
       linkUsed.uplink.add(ackTest);
}
```

Στην περίπτωση του addacktolink δεν γίνεται έλεγχος, όπως είχαμε στην addpackettolink, για το ενδεχόμενο να είναι γεμάτη πακέτα επιβεβαίωσης η ζεύξη(πιο συγκεκριμένα το πεδίο uplink της ζεύξης) καθότι γίνεται η θεώρηση ότι τα πακέτα επιβεβαίωσης ack έχουν αμελητέο μέγεθος.

Αφότου το πακέτο προστεθεί στη ζεύξη, με την βοήθεια ενός Timer ρυθμίζεται η κλήση της dropAck ώστε να αφαιρεθεί το πακέτο επιβεβαίωσης από την ζεύξη μετά από ένα συγκεκριμένο χρονικό διάστημα που ορίζεται ως trans_time. Αυτός ο χρόνος αντιστοιχεί στον χρόνο που στην πραγματικότητα απαιτείται ώστε να μεταφερθεί ένα πακέτο μέσα από την ζεύξη, ο οποίος και εύκολα υπολογίζεται γνωρίζοντας την απόσταση μεταξύ δύο κόμβων αλλά και την ταχύτητα διάδοσης.

```
Timer temei = new Timer();
temei.schedule(new
dropAck(u,nod1,l,ackTest,dst,hybrok,stok,packetTest,linkUsed), trans_time);
```

Η πρωταρχική λειτουργία της dropAck είναι, όπως αναφέρθηκε και στην ενότητα 4.2.2.1, όπου και περιγράφεται η Node.java, να καλεί την removeack που απλά αφαιρεί το τυχών πακέτο επιβεβαίωσης από το uplink της ζεύξης. Στην συνέχεια, πραγματοποιούνται δύο διαφορετικές διαδικασίες, που σχετίζονται με το εάν ο κόμβος που έλαβε το πακέτο επιβεβαίωσης είναι ο παραλήπτης του παρόντος πακέτου ή απλά ένας από τους ενδιάμεσους κόμβους της διαδρομής του πακέτου επιβεβαίωσης.

Στην περίπτωση όπου ο παρών κόμβος αποτελεί απλά έναν κόμβο του δικτύου και ενδιάμεσο στη διαδρομή από τον αποστολέα στον παραλήπτη του πακέτου επιβεβαίωσης, φροντίζει να βρει τον επόμενο κόμβο στην διαδρομή του πακέτου, με την βοήθεια και πάλι της κλάσης Routing.java, και να τοποθετήσει το πακέτο επιβεβαίωσης στην νέα ζεύξη πλέον προς τον καινούργιο προορισμό του πακέτου.

```
int dest3 = ackTest.current_node;
Link linkUsedAc = (links[dest][dest3].id != null) ? links[dest][dest3] :
links[dest3][dest];
n[dest].addACKtolink(n, ackTest.packet_src_node, links, ackTest,
ackTest.current_node,hybrid, stats, packetTest, linkUsedAc);
```

Η παραπάνω διαδικασία όπως είναι φανερό επαναλαμβάνεται έως ότου το πακέτο επιβεβαίωσης φτάσει και πάλι στην dropAck, αυτή την φορά όμως σε κόμβο που αποτελεί τον τελικό προορισμό του πακέτου. Σε αυτή την περίπτωση έχουμε μια επιτυχημένη άφιξη ενός πακέτου επιβεβαίωσης στον τελικό του προορισμό. Έτσι γίνονται οι απαραίτητες μεταβολές στην κλάση Statistics.java που διατηρεί τα στατιστικά του δικτύου και ολοκληρώνεται η κλήση της SendPack.

Παράλληλα με την SendPack όπως είχε αναφερθεί παραπάνω εκτελούνται και κάποιες άλλες διεργασίες με την βοήθεια των TimerTasks, όπως είχαν προγραμματιστεί στην κλάση Init.java. Αυτές είναι οι CalcDarwin, CalcICAS, CalcFree, CalcNewKatastasi οι οποίες έχουν διαφορετική περίοδο κλήσης από την SendPack.

Αναλυτικότερα, η CalcICAS είναι υπεύθυνη για τον υπολογισμό και την ενημέρωση των μετρητών που κρατάει το ICAS για κάθε κόμβο. Οι μετρητές αυτοί αναφέρθηκαν στην ενότητα 3. Αρχικά γίνεται ο υπολογισμός του μέσου λόγου συνδεσιμότητας που ορίζεται από την σχέση (3.3) ως

$$cp_j^{(k)} = \frac{\sum_{m \in N_i^{(k)} \cup \{i\}, m \neq j} c_{im}^{(k)} \cdot c_{mj}^{(k)}}{\sum_{m \in N_i^{(k)} \cup \{i\}, m \neq j} c_{im}^{(k)}} \quad \text{όπου } c_{ii}^{(k)} = 1$$

έχοντας βέβαια πρώτα υπολογίσει τον λόγο συνδεσιμότητας που ορίζεται από την σχέση (3.2). Στην συνέχεια, υπολογίζεται εάν ο κόμβος είναι εγωιστής σύμφωνα με το μέσο λόγο

συνδεσιμότητας που ευρέθηκε προηγουμένως. Πιο συγκεκριμένα, εάν η τιμή του cp είναι μικρότερη από το 0.5 ο κόμβος θεωρείται και αποθηκεύεται ως εγωιστής. Τέλος, γίνονται οι απαραίτητοι υπολογισμοί για την εύρεση των κόμβων που δικαιούνται δωρεάν αποστολή πακέτων καθώς και ο υπολογισμός των credits που λαμβάνει ή χάνει ο κόμβος.

Η CalcDarwin με την σειρά της είναι υπεύθυνη για τον υπολογισμό και την ενημέρωση των μετρητών που υπολογίζονται στον μηχανισμό Darwin (του οποίου έγινε συνοπτική παρουσίαση στην ενότητα 1) και των οποίων γίνεται χρήση στον υβριδικό αλγόριθμο ICARUS. Αναλυτικότερα, γίνεται υπολογισμός των :

$$\tilde{p}_{iDARWIN}^{(k)} = \left[\gamma * (q_{-i}^{(k-1)} - q_i^{(k-1)})\right]_0^1 \quad for\ k \geq 0$$

$$q_i^{(k)} = \begin{cases} [\hat{p}_i^{(k)} - \tilde{p}_{iDARWIN}^{(k)}]_0^1 \ for\ k \geq 0 \\ 0 \ for\ k = -1 \end{cases}$$

$$[x]_0^1 = \begin{cases} 1 & if\ x \geq 1 \\ x & if\ 0 < x < 1 \\ 0 & if\ x \leq 0 \end{cases}$$

Όπως ορίστηκαν στις σχέσεις (1.1), (1.2), (1.3) της ενότητας 1.

Όπου,

$\tilde{p}_{iDARWIN}^{(k)}$: πιθανότητα απόρριψης του DARWIN

$q_i^{(k)}$: δείχνει την κατάσταση του κάθε κόμβου

Βάσει των παραπάνω πιθανοτήτων, υπολογίζεται και πάλι εάν κάποιος κόμβος δρα εγωιστικά.

Η CalcFree είναι υπεύθυνη για τον υπολογισμό των δωρεάν χρηματικών μονάδων(credits) που δίνονται σε κάθε κόμβο που τις δικαιούται καθώς και την τροποποίηση των credits κάποιου κόμβου που έχει χαρακτηριστεί ως "υπερ"πλούσιος.

Τέλος έχουμε την περιοδική κλήση της CalcNewKatastasi η οποία φροντίζει για τον τακτικό υπολογισμό και μετάβασης της κατάστασης στην οποία βρίσκεται ένας κόμβος. Πιο συγκεκριμένα, οι καταστάσεις είναι τρεις και είχαν αναφερθεί παραπάνω στην ενότητα 3.3.2. Συνοπτικά ξαναπαρουσιάζονται παρακάτω.

- Κατάσταση 0 – Ο κόμβος μπορεί να στείλει πακέτα οπουδήποτε: αν $m_i^{(k)} \geq S_t$.
- Κατάσταση 1 – Ο κόμβος μπορεί να στείλει πακέτα μόνο στους γείτονές του: αν $m_i^{(k)} < S_t$.

- Κατάσταση 2 – Ο κόμβος δεν μπορεί να στείλει πακέτα πουθενά : αν $m_i^{(k)} < 0$.

Μετά το πέρας του χρόνου προσομοίωσης που έχει επιλεγεί, γίνεται η καθυστερημένη κλήση της KillThread. Η μόνο λειτουργία της συγκεκριμένης κλάσης είναι η αναστολή λειτουργίας των ανάλογων TimerTasks που με την σειρά τους σημαίνουν τον τερματισμό της προσομοίωσης. Σε αυτό το σημείο επίσης γίνεται κλήση της διαδικασίας printStats που αποτελεί μέρος της Statistics.java και φροντίζει για την εκτύπωση των στατιστικών στοιχείων του δικτύου που συλλέχθηκαν κατά την διάρκεια της προσομοίωσης.

5. ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΣΟΜΟΙΩΣΗΣ

5.1 Παράμετροι Προσομοίωσης

Παρακάτω παρατίθενται οι παράμετροι προσομοίωσης οι οποίες σποραδικά αναφέρονται και στην ενότητα 4.2.2 όπου γίνεται η αναλυτική περιγραφή του αλγορίθμου. Η τοπολογία της προσομοίωσης παρουσιάστηκε προηγούμενα στην ενότητα 4.1.2.

Μεταβλητές	**Τιμή**	**Περιγραφή**
m_0	220	Number of credits assigned to each node initially
pf_1	1.1	1st threshold used to assist distant nodes
pf_2	4	2nd threshold used to assist distant nodes
$cp_{ICAS,th}$	0.5	Threshold used by ICAS to determine if a node is selfish
S_t	4	Credit threshold to determine node state
IFN	5	Number of packets that must be dropped before a selfish begins cooperating
edp_{th}	0.85	Threshold used by nodes to determine if another node is selfish
$m_{e,0}$	28	Initial amount of credits that are given to distant nodes
h_{max}	15	Maximum number of hops allowed per packet

Πίνακας 5.0

5.2 Αριθμητικά αποτελέσματα και γραφικές παραστάσεις

5.2.1 Αποτελέσματα και σύγκριση ICARUS - DARWIN

Παρακάτω γίνεται παρουσίαση των αποτελεσμάτων της προσομοίωσης σε μορφή Πινάκων με τα αριθμητικά αποτελέσματα και στην συνέχεια παρουσιάζονται οι γραφικές παραστάσεις αυτών. Η τοπολογία των κόμβων παραμένει ως παρουσιάστηκε στην ενότητα 4.1.2. Σε κάθε περίπτωση ωστόσο θα γίνεται αναλυτική αναφορά των συνθηκών προσομοίωσης, όσων αφορά στα στοιχεία της προσομοίωσης όπως

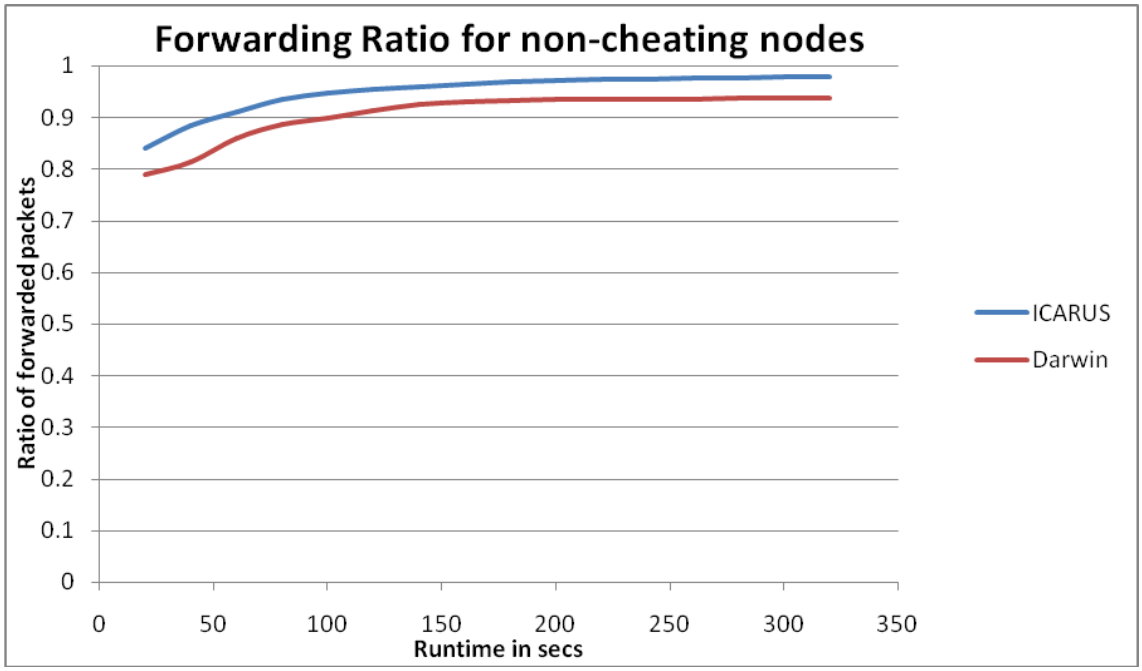
- Αριθμός των κόμβων του δικτύου
- Αριθμός των κόμβων που δρουν εγωιστικά στο δίκτυο
- Περίοδος κλήσης CalcICAS σε δευτερόλεπτα
- Περίοδος κλήσης CalcDarwin σε δευτερόλεπτα
- Περίοδος κλήσης CalcNewKatastasi σε δευτερόλεπτα
- Περίοδος κλήσης CalcIFree σε δευτερόλεπτα

Αρχικά παρουσιάζονται τα αριθμητικά αποτελέσματα του ποσοστού προώθησης πακέτων (Packet Forwarding Ratio) των μη εγωιστών κόμβων για τον μηχανισμό ICARUS, ως επίσης και τα αποτελέσματα του μηχανισμού DARWIN, με σκοπό να γίνει σύγκριση των δύο μηχανισμών. Τα αποτελέσματα αυτά προέκυψαν για την βασική μας τοπολογία με περιόδους υπολογισμού των CalcICAS, CalcDarwin, CalcNewKatastasi, CalcFree σταθερά στα 10 δευτερόλεπτα, με συνολικό χρόνο προσομοίωσης τα 320 δευτερόλεπτα.

Runtime in secs	ICARUS	DARWIN
20	0.841129857	0.789127764
40	0.883125	0.813469825
60	0.911111111	0.860507246
80	0.933937824	0.886010363
100	0.946428571	0.899185336
120	0.954180064	0.912751678
140	0.959459459	0.924731183
160	0.964452214	0.930942895
180	0.96772541	0.932960894
200	0.971219735	0.935049737
220	0.972749794	0.935251799
240	0.973991708	0.93487395
260	0.975051975	0.935935936
280	0.976580045	0.937076476
300	0.978215458	0.936210131
320	0.979358438	0.937215651

Πίνακας 5.1 - Ποσοστό προώθησης πακέτων των μη εγωιστών κόμβων στους μηχανισμούς ICARUS και DARWIN	
Κόμβοι	50
Εγωιστές	3
Περίοδος Calc ICAS (secs)	10
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	10
Περίοδος Calc Free(secs)	10

Παρακάτω παρατίθεται η γραφική παράσταση που αντιστοιχεί στα παραπάνω αριθμητικά αποτελέσματα, με σκοπό να γίνουν πιο εμφανείς οι διαφορές των δύο αλγορίθμων.



Διάγραμμα 5.1 - Ποσοστό προώθησης πακέτων των μη εγωιστών κόμβων στους μηχανισμούς ICARUS και DARWIN	
Κόμβοι	50
Εγωιστές	3
Περίοδος Calc ICAS (secs)	10
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	10
Περίοδος Calc Free(secs)	10

Κρίνοντας από τα παραπάνω αποτελέσματα διαφαίνεται ότι ο προτεινόμενος μηχανισμός ICARUS παρουσιάζει ιδιαίτερη αποτελεσματικότητα στην προώθηση των πακέτων των μη εγωιστών κόμβων μέσα σε ένα δίκτυο που χαρακτηρίζεται από 6% ποσοστό εγωιστών κόμβων(3 στους 50 συνολικούς κόμβους). Συγκρινόμενος με τον μηχανισμό DARWIN, ο ICARUS αποδεικνύεται καλύτερος, παρουσιάζοντας σταθερά καλύτερα αποτελέσματα προώθησης πακέτων των μη εγωιστών κόμβων καθ' όλη την διάρκεια της προσομοίωσης (320 δευτερόλεπτα).

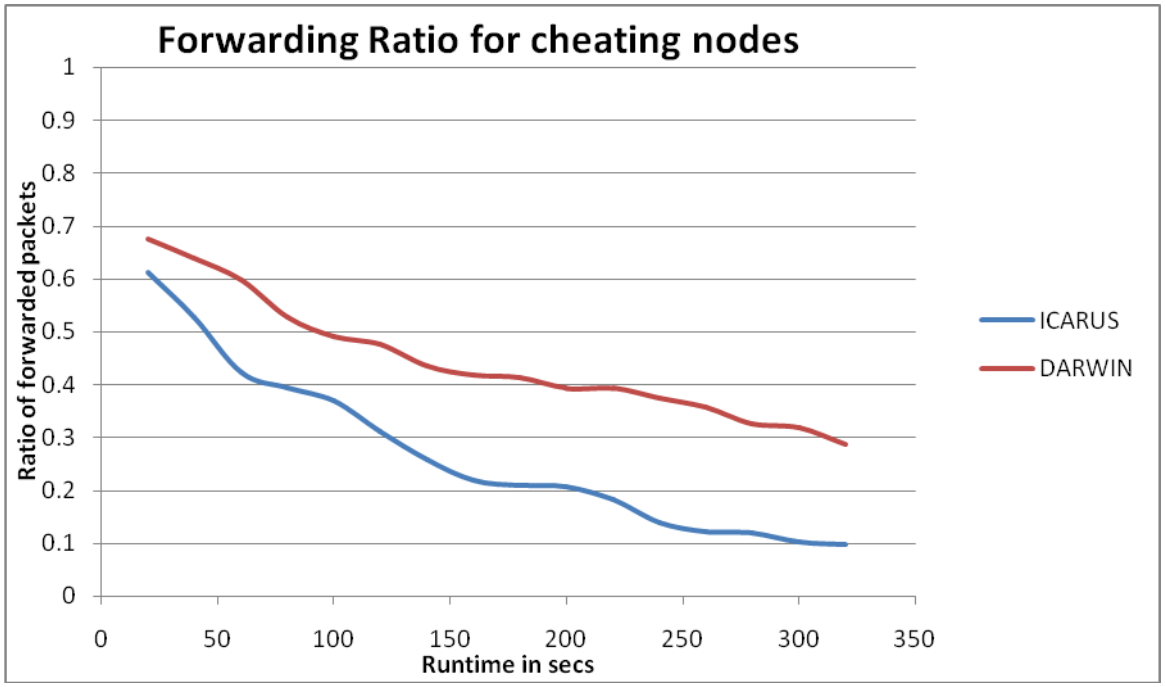
Εν συνεχεία, παρουσιάζονται τα αριθμητικά αποτελέσματα του ποσοστού προώθησης πακέτων (Packet Forwarding Ratio) των εγωιστών κόμβων για τον μηχανισμό ICARUS, ως επίσης και τα αποτελέσματα του μηχανισμού DARWIN, με σκοπό να γίνει σύγκριση των δύο μηχανισμών. Τα αποτελέσματα αυτά προέκυψαν για την βασική μας τοπολογία με περιόδους υπολογισμού των CalcICAS, CalcDarwin, CalcNewKatastasi, CalcFree σταθερά στα 10 δευτερόλεπτα, με συνολικό χρόνο προσομοίωσης τα 320 δευτερόλεπτα.

Runtime in secs	ICARUS	DARWIN
20	0.61163292	0.67475862
40	0.52660782	0.63875491
60	0.42199252	0.59795925
80	0.39454336	0.52733658
100	0.36869341	0.491814
120	0.31170857	0.47532345
140	0.25790351	0.43475228
160	0.21900357	0.41829273
180	0.21005249	0.41382319
200	0.2072366	0.39309862
220	0.18164257	0.39309862
240	0.1398522	0.37425908
260	0.12230685	0.35683583
280	0.11480644	0.32711297
300	0.10279045	0.31900625
320	0.098566227	0.28614779

Πίνακας 5.2 - Ποσοστό προώθησης πακέτων των εγωιστών κόμβων στους μηχανισμούς ICARUS και DARWIN

Κόμβοι	50
Εγωιστές	3
Περίοδος Calc ICAS (secs)	10
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	10
Περίοδος Calc Free(secs)	10

Παρακάτω παρατίθεται η γραφική παράσταση που αντιστοιχεί στα παραπάνω αριθμητικά αποτελέσματα, με σκοπό να γίνουν πιο εμφανείς οι διαφορές των δύο αλγορίθμων.



Διάγραμμα 5.2 - Ποσοστό προώθησης πακέτων των εγωιστών κόμβων στους μηχανισμούς ICARUS και DARWIN	
Κόμβοι	50
Εγωιστές	3
Περίοδος Calc ICAS (secs)	10
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	10
Περίοδος Calc Free(secs)	10

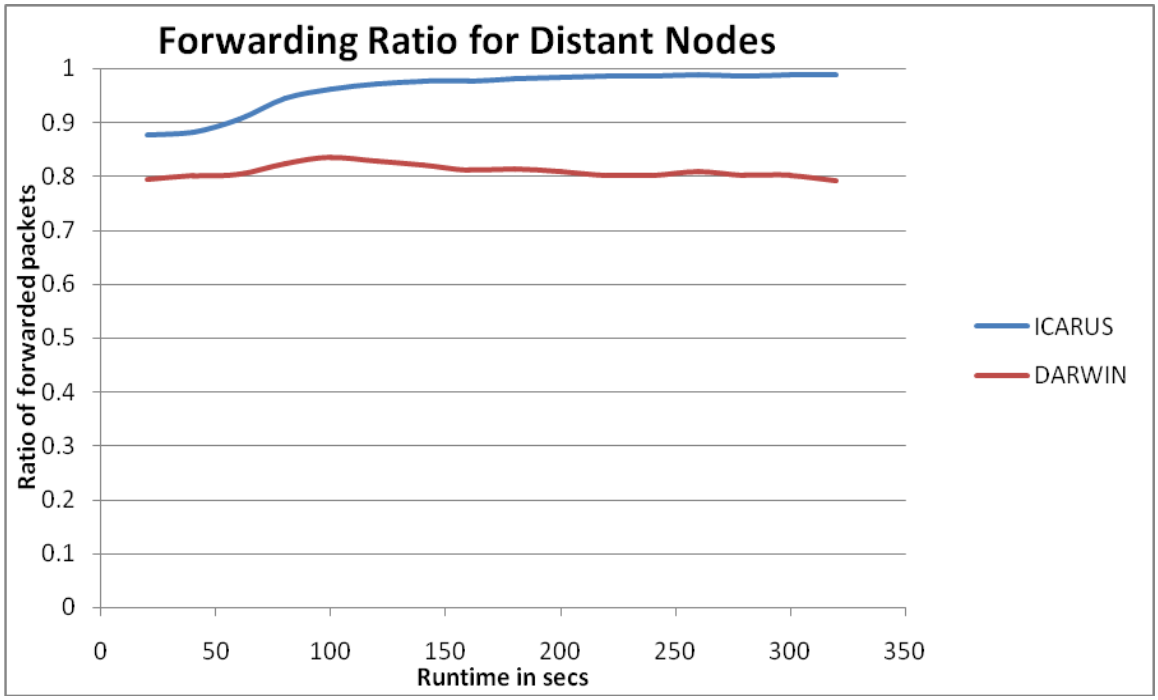
Στην περίπτωση των εγωιστών κόμβων τα αποτελέσματα του προτεινόμενου μηχανισμού ICARUS είναι ιδιαίτερα εντυπωσιακά όσων αφορά στην καταπολέμηση των εγωιστών κόμβων στο δίκτυο. Η μείωση των προωθούμενων πακέτων ενός εγωιστή κόμβου στον ICARUS είναι ταχύτατη και τελικά το ποσοστό προώθησης καταλήγει να είναι οριακά μικρότερο του 10%. Γεγονός το οποίο αποδεικνύει την σκληρότητα αντιμετώπισης των εγωιστών κόμβων, με σκοπό να τους αναγκάσει να επιστρέψουν σε συνεργατική συμπεριφορά. Συγκρινόμενος με τον μηχανισμό DARWIN, ο ICARUS αποδεικνύεται σαφώς καλύτερος καθ' όλη την διάρκεια της προσομοίωσης. Πιο συγκεκριμένα παρουσιάζεται αποτελεσματικότερος σε κάθε χρονική στιγμή της προσομοίωσης, ενώ στο τέλος καταλήγει να έχει πλεονέκτημα 19% μεγαλύτερης αποτελεσματικότητας, αφού ο DARWIN καταλήγει με ποσοστό 28.61% ενώ ο ICARUS 9.85% (28.61-9.85 = 18.76, περίπου 19% δηλαδή).

Παρακάτω, παρουσιάζονται τα αριθμητικά αποτελέσματα του ποσοστού προώθησης πακέτων (Packet Forwarding Ratio) των ακριανών κόμβων του δικτύου για τον μηχανισμό ICARUS, ως επίσης και τα αποτελέσματα του μηχανισμού DARWIN, με σκοπό να γίνει σύγκριση των δύο μηχανισμών. Τα αποτελέσματα αυτά προέκυψαν για την βασική μας τοπολογία με περιόδους υπολογισμού των CalcICAS, CalcDarwin, CalcNewKatastasi, CalcFree σταθερά στα 10 δευτερόλεπτα, με συνολικό χρόνο προσομοίωσης τα 320 δευτερόλεπτα.

Runtime in secs	ICARUS	DARWIN
20	0.87554123	0.79411563
40	0.882352941	0.801178367
60	0.906551724	0.803741189
80	0.944444444	0.823958333
100	0.962025316	0.834317343
120	0.971153846	0.827710367
140	0.976744186	0.821554987
160	0.977272727	0.812272727
180	0.980446927	0.813471224
200	0.982843137	0.809205438
220	0.984716157	0.801557812
240	0.986220472	0.802352212
260	0.987455197	0.80787234
280	0.986820428	0.802054381
300	0.98778626	0.801554987
320	0.988636364	0.792272727

Πίνακας 5.3 - Ποσοστό προώθησης πακέτων των ακριανών κόμβων στους μηχανισμούς ICARUS και DARWIN	
Κόμβοι	50
Εγωιστές	3
Περίοδος Calc ICAS (secs)	10
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	10
Περίοδος Calc Free(secs)	10

Παρακάτω παρατίθεται η γραφική παράσταση που αντιστοιχεί στα παραπάνω αριθμητικά αποτελέσματα, με σκοπό να γίνουν πιο εμφανείς οι διαφορές των δύο αλγορίθμων.



Διάγραμμα 5.3 - Ποσοστό προώθησης πακέτων των ακριανών κόμβων στους μηχανισμούς ICARUS και DARWIN	
Κόμβοι	50
Εγωιστές	3
Περίοδος Calc ICAS (secs)	10
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	10
Περίοδος Calc Free(secs)	10

Όμοια με προηγούμενα ο προτεινόμενος μηχανισμός ICARUS φαίνεται να είναι αποτελεσματικότατος στην αντιμετώπιση των ακριανών κόμβων, που όπως αναφέρθηκε και σε προηγούμενες ενότητες απαιτούν ιδιαίτερη αντιμετώπιση εξαιτίας της φυσικής αδυναμίας που έχουν λόγω της θέσης τους στο δίκτυο. Αρχικά αξίζει να σημειωθεί ότι τα ποσοστά προώθησης είναι απολύτως ικανοποιητικά και για τους δύο μηχανισμούς, αφού ξεπερνούν το 80%. Ωστόσο και σε αυτό το σημείο, ο ICARUS διατηρεί μεγάλο πλεονέκτημα αφού τα ποσοστά προώθησης του για τους ακριανούς κόμβους αγγίζουν το 100% προς το τέλος της προσομοίωσης σε αντίθεση με τον DARWIN που βρίσκονται στο 80%, με την διαφορά τους έτσι να ανέρχεται σε σχεδόν 20 ποσοστιαίες μονάδες(98.86% - 79.22% = 19.64%).

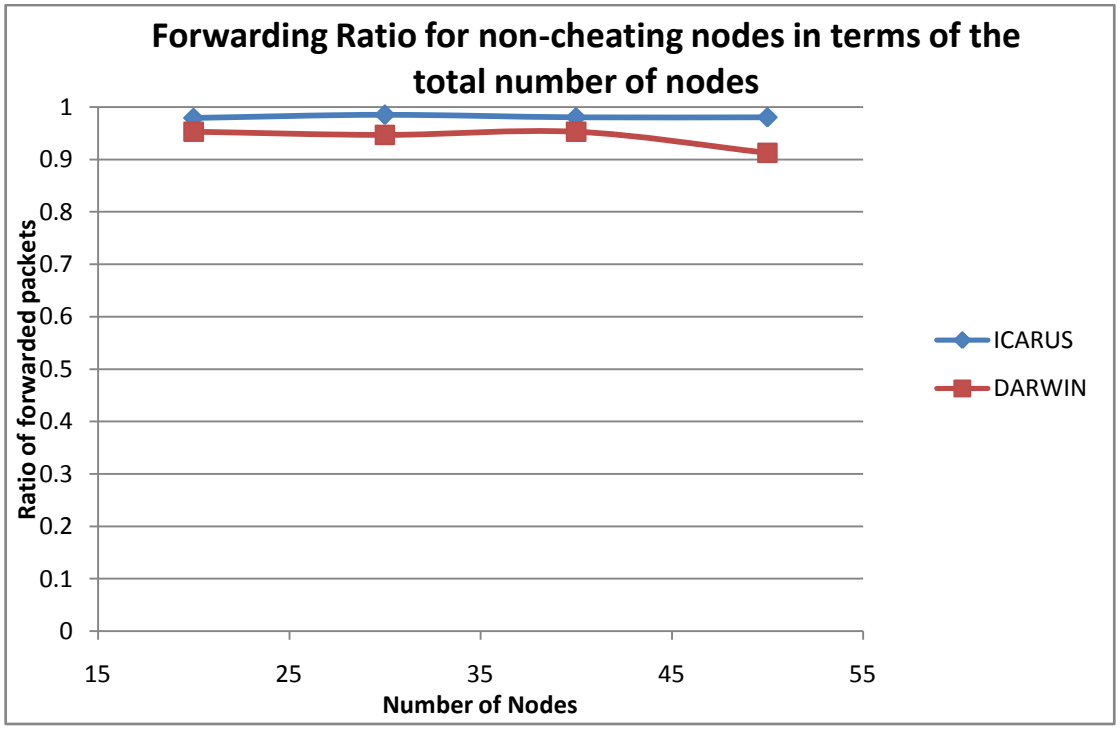
Κατόπιν, παρουσιάζονται τα αριθμητικά αποτελέσματα του ποσοστού προώθησης πακέτων (Packet Forwarding Ratio) των μη εγωιστών κόμβων για τον μηχανισμό ICARUS, ως επίσης και τα αποτελέσματα του μηχανισμού DARWIN, με σκοπό να γίνει σύγκριση των δύο μηχανισμών. Αυτή την φορά ωστόσο στον οριζόντιο άξονα αντί να μελετήσουμε την συμπεριφορά των μηχανισμών συναρτήσει του χρόνου την μελετήσαμε συναρτήσει του συνολικού αριθμού των κόμβων του δικτύου. Ομοίως με προηγούμενα οι περίοδοι υπολογισμού των CalcICAS, CalcDarwin, CalcNewKatastasi, CalcFree διατηρήθηκαν σταθερά στα 10 δευτερόλεπτα.

Number of Nodes	ICARUS	DARWIN
20	0.980645161	0.953125
30	0.980603448	0.947019868
40	0.985090522	0.952971731
50	0.979358438	0.912875867

Πίνακας 5.4 - Ποσοστό προώθησης πακέτων των μη εγωιστών κόμβων στους μηχανισμούς ICARUS και DARWIN συναρτήσει του συνολικού αριθμού κόμβων του δικτύου

Κόμβοι	-
Εγωιστές	-
Περίοδος Calc ICAS (secs)	10
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	10
Περίοδος Calc Free(secs)	10

Παρακάτω παρατίθεται η γραφική παράσταση που αντιστοιχεί στα παραπάνω αριθμητικά αποτελέσματα, με σκοπό να γίνουν πιο εμφανείς οι διαφορές των δύο αλγορίθμων.



Διάγραμμα 5.4 - Ποσοστό προώθησης πακέτων των μη εγωιστών κόμβων στους μηχανισμούς ICARUS και DARWIN συναρτήσει του συνολικού αριθμού κόμβων του δικτύου	
Κόμβοι	-
Εγωιστές	-
Περίοδος Calc ICAS (secs)	10
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	10
Περίοδος Calc Free(secs)	10

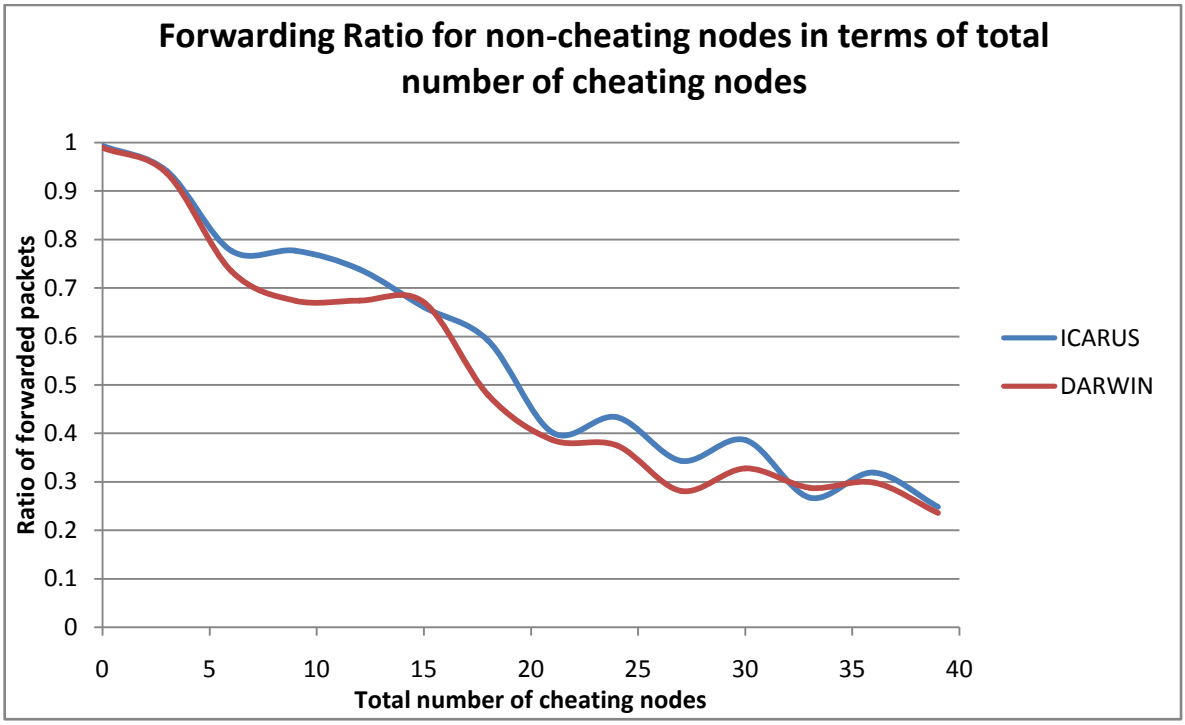
Ακόμα και σε αυτή την περίπτωση ο προτεινόμενος μηχανισμός ICARUS φαίνεται να διατηρεί την αποτελεσματικότητα του αφού παρουσιάζει παρόμοια ποσοστά προώθησης για οποιονδήποτε αριθμό κόμβων βρεθούν στο δίκτυο. Σε αυτό τον τομέα, ο DARWIN παρουσιάζει και αυτός αξιοσημείωτη σταθερότητα στην απόδοση του. Παρόλα' αυτά ο ICARUS υπερέχει καθώς σε όλες τις μετρήσεις εμφανίζεται λίγο καλύτερος στην τελική προώθηση των πακέτων.

Παρακάτω, παρουσιάζονται τα αριθμητικά αποτελέσματα του ποσοστού προώθησης πακέτων (Packet Forwarding Ratio) των μη εγωιστών κόμβων για τον μηχανισμό ICARUS, ως επίσης και τα αποτελέσματα του μηχανισμού DARWIN, με σκοπό να γίνει σύγκριση των δύο μηχανισμών. Αυτή την φορά ωστόσο στον οριζόντιο άξονα αντί να μελετήσουμε την συμπεριφορά των μηχανισμών συναρτήσει του χρόνου την μελετήσαμε συναρτήσει του συνολικού αριθμού των εγωιστών κόμβων του δικτύου. Οι μετρήσεις ξεκινούν από 0 κόμβους και με περίοδο 3 φτάνουμε στο πέρας της προσομοίωσης να έχουμε 39 εγωιστές κόμβους στο σύστημα (78% του συνολικού αριθμού κόμβων είναι εγωιστές). Ομοίως με προηγούμενα οι περίοδοι υπολογισμού των CalcICAS, CalcDarwin, CalcNewKatastasi, CalcFree διατηρήθηκαν σταθερά στα 10 δευτερόλεπτα.

Number of Cheating Nodes	ICARUS	DARWIN
0	0.99463899	0.989472365
3	0.9411752	0.9371699
6	0.777069002	0.735437946
9	0.777004919	0.673925053
12	0.738784921	0.674259356
15	0.660651294	0.670502709
18	0.591546073	0.478724722
21	0.401795007	0.386379119
24	0.433460164	0.374894304
27	0.343134857	0.281065526
30	0.386211445	0.327732818
33	0.267256898	0.287777059
36	0.31883997	0.298467148
39	0.24830016	0.235871559

Πίνακας 5.5 - Ποσοστό προώθησης πακέτων των μη εγωιστών κόμβων στους μηχανισμούς ICARUS και DARWIN συναρτήσει του συνολικού αριθμού εγωιστών κόμβων του δικτύου

Κόμβοι	50
Εγωιστές	-
Περίοδος Calc ICAS (secs)	10
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	10
Περίοδος Calc Free(secs)	10



Διάγραμμα 5.5 - Ποσοστό προώθησης πακέτων των μη εγωιστών κόμβων στους μηχανισμούς ICARUS και DARWIN συναρτήσει του συνολικού αριθμού εγωιστών κόμβων του δικτύου	
Κόμβοι	50
Εγωιστές	-
Περίοδος Calc ICAS (secs)	10
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	10
Περίοδος Calc Free(secs)	10

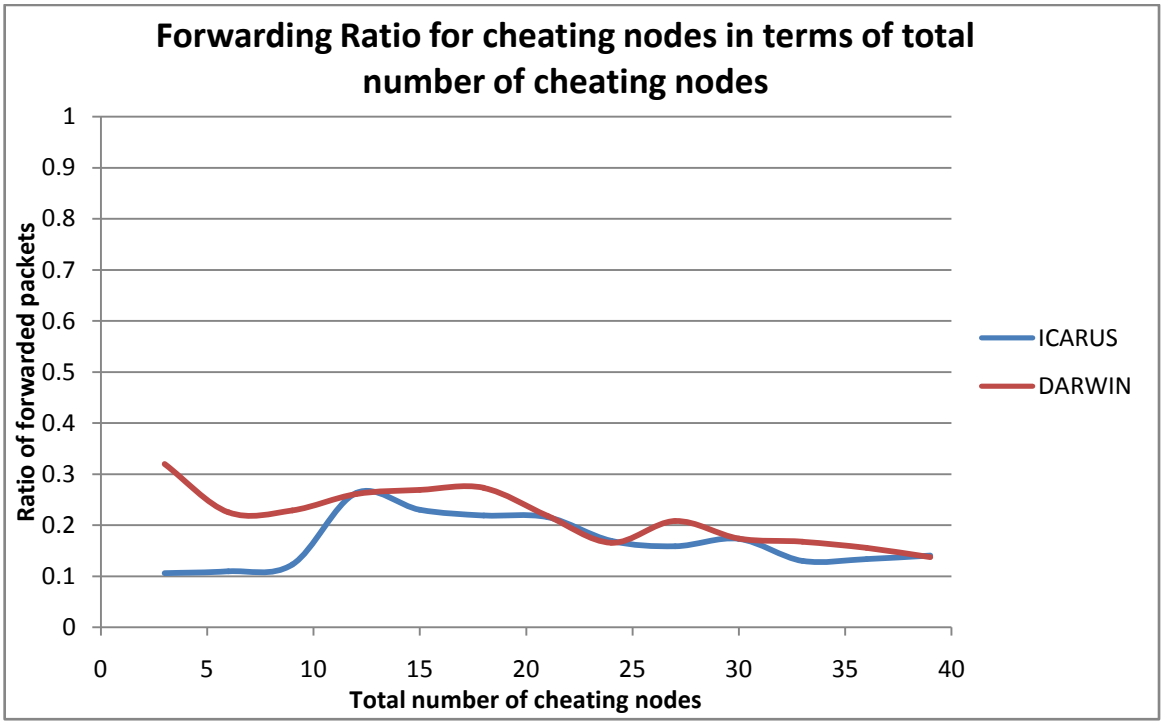
Παρατηρώντας, την παραπάνω γραφική γίνεται εμφανές ότι η αποδοτικότητα και η αποτελεσματικότητα και των δύο μηχανισμών μειώνεται αισθητά με την αύξηση του αριθμού των εγωιστών κόμβων αφού μειώνεται χαρακτηριστικά το ποσοστό προώθησης πακέτων των μη εγωιστών κόμβων. Γεγονός απόλυτα κατανοητό αφού δεν είναι δυνατό κάποιος μηχανισμός να μπορεί να είναι αποτελεσματικός όταν το δίκτυο έχει κατακλυστεί από εγωιστές κόμβους (78% του συνολικού αριθμού κόμβων). Σε αυτή την περίπτωση δεν παρατηρείται κάποια σαφή υπεροχή για κάποιον από τους δύο αλγορίθμους, παρότι ο ICARUS παρουσιάζει ελαφρώς ανώτερα ποσοστά προώθησης (η μέγιστη διαφορά ανέρχεται σε 11 ποσοστιαίες μονάδες σε βάρος του DARWIN για 18 εγωιστές κόμβους, 59,15% - 47,87% = 11.28%).

Στην πορεία, παρουσιάζονται τα αριθμητικά αποτελέσματα του ποσοστού προώθησης πακέτων (Packet Forwarding Ratio) των εγωιστών κόμβων για τον μηχανισμό ICARUS, ως επίσης και τα αποτελέσματα του μηχανισμού DARWIN, με σκοπό να γίνει σύγκριση των δύο μηχανισμών. Αυτή την φορά ωστόσο στον οριζόντιο άξονα αντί να μελετήσουμε την συμπεριφορά των μηχανισμών συναρτήσει του χρόνου την μελετήσαμε συναρτήσει του συνολικού αριθμού των εγωιστών κόμβων του δικτύου. Οι μετρήσεις ξεκινούν από 3 κόμβους και με περίοδο 3 φτάνουμε στο πέρας της προσομοίωσης να έχουμε 39 εγωιστές κόμβους στο σύστημα (78% του συνολικού αριθμού κόμβων είναι εγωιστές). Ομοίως με προηγούμενα οι περίοδοι υπολογισμού των CalcICAS, CalcDarwin, CalcNewKatastasi, CalcFree διατηρήθηκαν σταθερά στα 10 δευτερόλεπτα.

Number of Cheating Nodes	ICARUS	DARWIN
3	0.105758413	0.31998064
6	0.109331367	0.225209585
9	0.11213885	0.22860798
12	0.262808869	0.260986604
15	0.230211382	0.268621001
18	0.218936066	0.272877002
21	0.215543095	0.218242946
24	0.169108571	0.165376933
27	0.158538201	0.189959066
30	0.173101063	0.173385646
33	0.129751692	0.167454927
36	0.133290406	0.155028815
39	0.140016087	0.137447304

Πίνακας 5.6 - Ποσοστό προώθησης πακέτων των εγωιστών κόμβων στους μηχανισμούς ICARUS και DARWIN συναρτήσει του συνολικού αριθμού εγωιστών κόμβων του δικτύου

Κόμβοι	50
Εγωιστές	-
Περίοδος Calc ICAS (secs)	10
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	10
Περίοδος Calc Free(secs)	10



Διάγραμμα 5.6 - Ποσοστό προώθησης πακέτων των εγωιστών κόμβων στους μηχανισμούς ICARUS και DARWIN συναρτήσει του συνολικού αριθμού εγωιστών κόμβων του δικτύου	
Κόμβοι	50
Εγωιστές	-
Περίοδος Calc ICAS (secs)	10
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	10
Περίοδος Calc Free(secs)	10

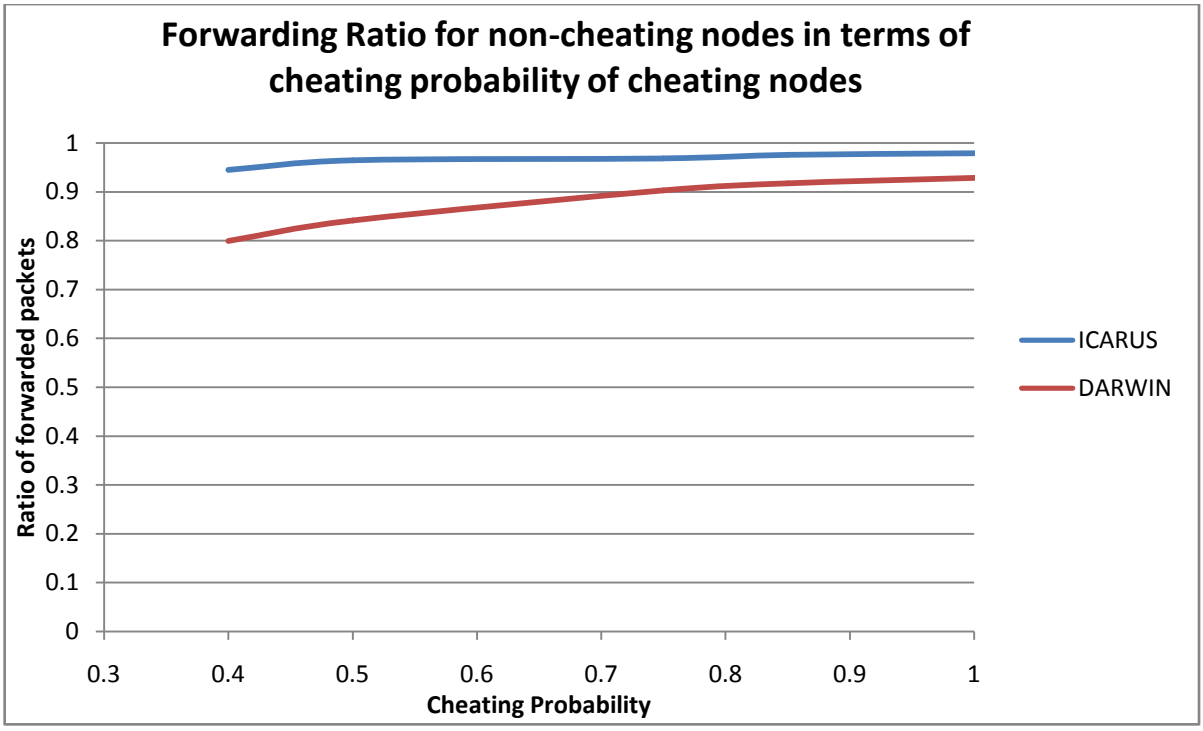
Παρατηρώντας, την παραπάνω γραφική διαφαίνεται ότι η αποδοτικότητα και η αποτελεσματικότητα και των δύο μηχανισμών δεν δέχεται σοβαρό πλήγμα στην αντιμετώπιση των εγωιστών κόμβων. Οι μηχανισμοί συνεχίζουν να διατηρούν το ποσοστό προώθησης των εγωιστών κόμβων σε χαμηλά επίπεδα ακόμα και για πολύ μεγάλα ποσοστά εγωιστών κόμβων στο δίκτυο. Σε αυτή την περίπτωση δεν παρατηρείται κάποια εντυπωσιακή διαφορά παρότι ο ICARUS δείχνει να είναι σταθερά λίγο καλύτερος διατηρώντας σε χαμηλότερα επίπεδα, σε σχέση με τον DARWIN τα ποσοστά προώθησης πακέτων των εγωιστών κόμβων (με μέγιστη διαφορά που ανέρχεται στις 21 ποσοστιαίες μονάδες περίπου για 3 εγωιστές κόμβους, 31.99% - 10.57% = 21.42%, όπου ο ICARUS είναι σαφώς αποτελεσματικότερος του DARWIN).

Συνεχίζοντας, παρουσιάζονται τα αριθμητικά αποτελέσματα του ποσοστού προώθησης πακέτων (Packet Forwarding Ratio) των μη εγωιστών κόμβων για τον μηχανισμό ICARUS, ως επίσης και τα αποτελέσματα του μηχανισμού DARWIN, με σκοπό να γίνει σύγκριση των δύο μηχανισμών. Αυτή την φορά ωστόσο στον οριζόντιο άξονα αντί να μελετήσουμε την συμπεριφορά των μηχανισμών συναρτήσει του χρόνου την μελετήσαμε συναρτήσει της πιθανότητας οι εγωιστές κόμβοι να κλέψουν. Οι μετρήσεις ξεκινούν από πιθανότητα 0.4 και καταλήγει σε πιθανότητα 1.0 δηλαδή όποτε του δίνεται η ευκαιρία κλέβει. Ομοίως με προηγούμενα οι περίοδοι υπολογισμού των CalcICAS, CalcDarwin, CalcNewKatastasi, CalcFree διατηρήθηκαν σταθερά στα 10 δευτερόλεπτα.

Cheating Probability of Cheating Nodes	ICARUS	DARWIN
0.4	0.945065617	0.799429365
0.5	0.964894684	0.841475399
0.75	0.968582375	0.902793654
0.85	0.97556391	0.917568272
1	0.979358438	0.928441754

Πίνακας 5.7 - Ποσοστό προώθησης πακέτων των μη εγωιστών κόμβων στους μηχανισμούς ICARUS και DARWIN συναρτήσει της πιθανότητας των εγωιστών κόμβων να κλέψουν.

Κόμβοι	50
Εγωιστές	3
Περίοδος Calc ICAS (secs)	10
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	10
Περίοδος Calc Free(secs)	10



Διάγραμμα 5.7 - Ποσοστό προώθησης πακέτων των μη εγωιστών κόμβων στους μηχανισμούς ICARUS και DARWIN συναρτήσει της πιθανότητας των εγωιστών κόμβων να κλέψουν.	
Κόμβοι	50
Εγωιστές	3
Περίοδος Calc ICAS (secs)	10
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	10
Περίοδος Calc Free(secs)	10

Το ποσοστό προώθησης των μη εγωιστών κόμβων δείχνει να μειώνεται και στους δύο μηχανισμούς όσο μειώνεται η πιθανότητα των εγωιστών κόμβων να κλέψουν. Όσων αφορά την αποτελεσματικότητα των δύο μηχανισμών ο ICARUS φαίνεται να είναι λίγο καλύτερος του DARWIN δεδομένου ότι η πτώση που παρουσιάζει για cheating probability 0.4 είναι αρκετά καλύτερη (ICARUS 0.945065617 ενώ ο DARWIN 0.799429365 δηλαδή διαφορά της τάξεως του 17%).

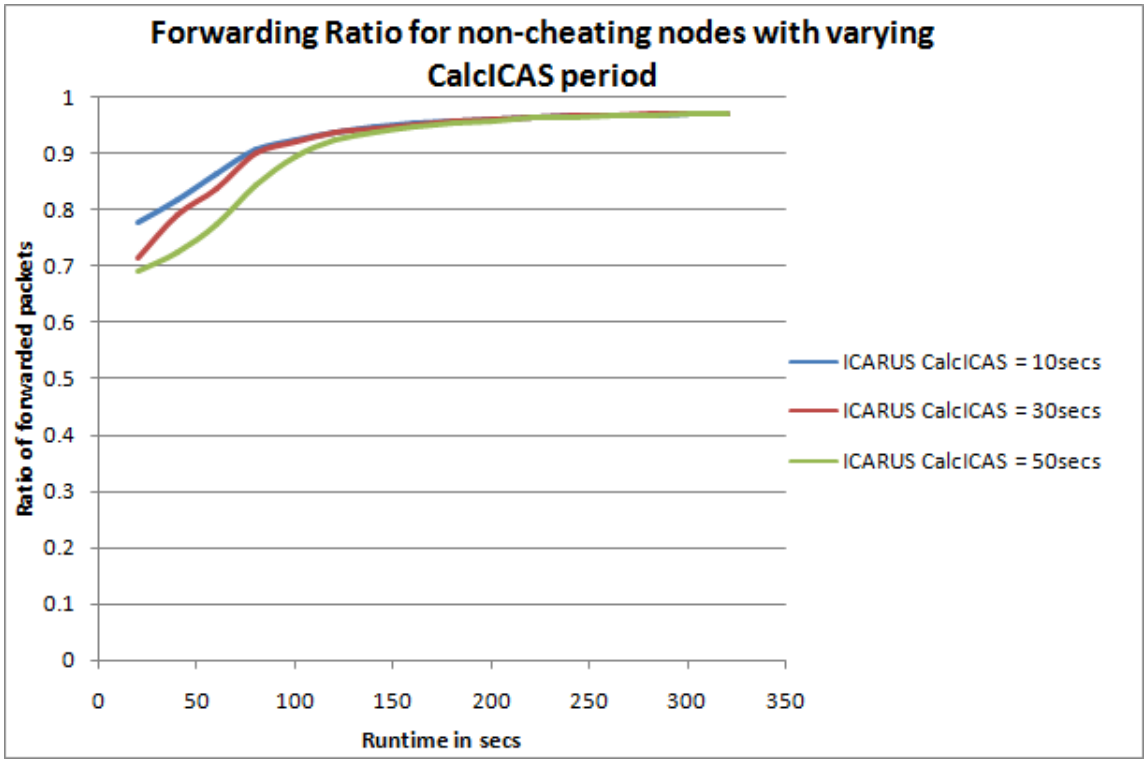
5.2.2 Αναλυτική μελέτη του μηχανισμού ICARUS μέσω προσομοίωσης

Αρχικά, παρουσιάζονται τα αριθμητικά αποτελέσματα του ποσοστού προώθησης πακέτων (Packet Forwarding Ratio) των μη εγωιστών κόμβων για τον μηχανισμό ICARUS. Η σημαντική διαφορά είναι ότι αυτή την φορά θα υπάρχει μεταβολή των περιόδων υπολογισμού των CalcICAS και CalcNewKatastasi ώστε να γίνει ξεκάθαρα κατανοητή η λειτουργία του αλγορίθμου και πως τα αποτελέσματα εξαρτώνται από τις περιόδους. Για αυτό τον λόγο παρατίθενται και οι γραφικές παραστάσεις των αποτελεσμάτων συγκρίνοντας την αποτελεσματικότητα της κάθε μορφής του ICARUS.

Runtime in secs	ICARUS CalcICAS = 10sec	ICARUS CalcICAS = 30sec	ICARUS CalcICAS = 50sec
20	0.778412733	0.714896337	0.691472368
40	0.817679558	0.791178934	0.724419834
60	0.866425993	0.837209302	0.773563218
80	0.906801008	0.900158479	0.844074844
100	0.926499033	0.921320887	0.896189225
120	0.93956044	0.939814815	0.923671498
140	0.947820343	0.945131376	0.938521401
160	0.954960091	0.952475248	0.947882736
180	0.95887663	0.957973518	0.95477671
200	0.961951656	0.961065574	0.959899749
220	0.964011322	0.964581417	0.963431151
240	0.966801918	0.967042136	0.96640721
260	0.968474576	0.968223583	0.967693464
280	0.969896519	0.971378092	0.96816609
300	0.970536756	0.97277796	0.970748955
320	0.972176759	0.973096912	0.97179718

Πίνακας 5.8 - Ποσοστό προώθησης πακέτων των μη εγωιστών κόμβων στον μηχανισμό ICARUS με μεταβλητή τιμή της περιόδου CalcICAS

Κόμβοι	50
Εγωιστές	3
Περίοδος Calc ICAS (secs)	-
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	10
Περίοδος Calc Free(secs)	10



Διάγραμμα 5.8 - Ποσοστό προώθησης πακέτων των μη εγωιστών κόμβων στον μηχανισμό ICARUS με μεταβλητή τιμή της περιόδου CalcICAS	
Κόμβοι	50
Εγωιστές	3
Περίοδος Calc ICAS (secs)	-
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	10
Περίοδος Calc Free(secs)	10

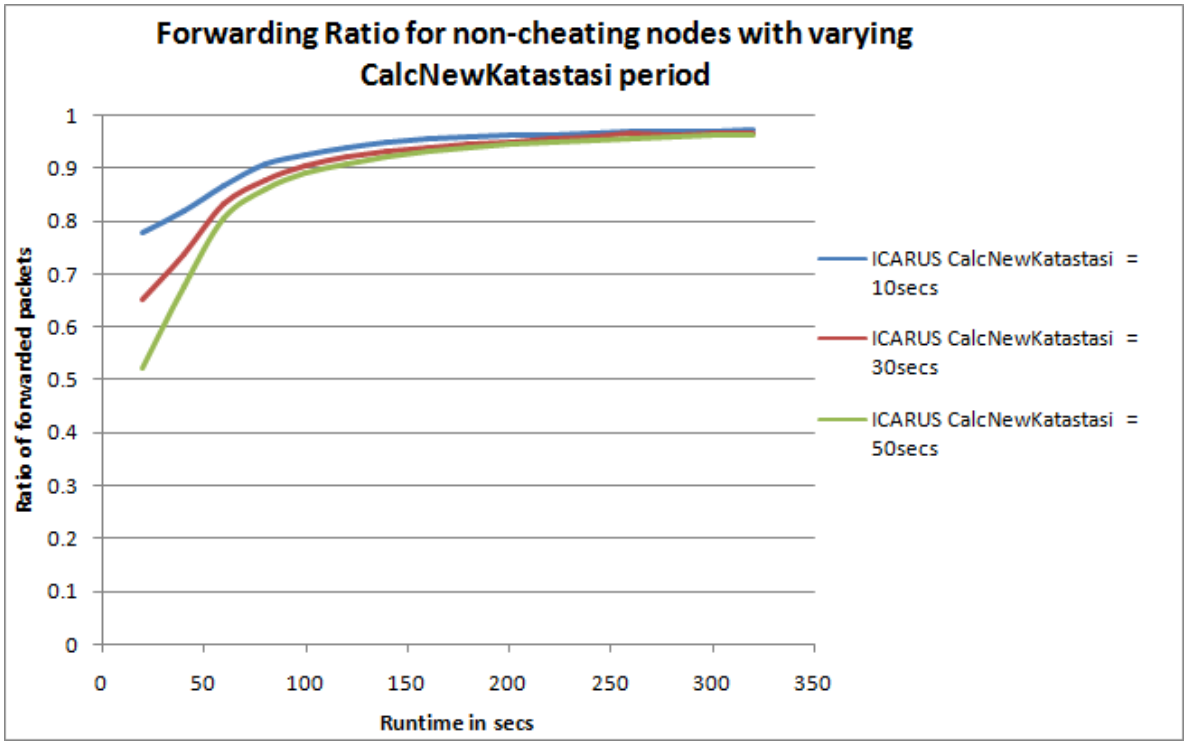
Όπως αναφέρθηκε και παραπάνω η βασική λειτουργία του CalcICAS είναι ο υπολογισμός του μέσου λόγου συνδεσιμότητας που ορίζεται από την σχέση (3.3), βάσει της οποίας στην συνέχεια υπολογίζει εάν ένας κόμβος είναι εγωιστής. Συνεπώς αυξάνοντας της περίοδο του CalcICAS, η συχνότητα ελέγχου και εντοπισμού ενός εγωιστή κόμβου αναμένεται να φθίνει. Αυτό βέβαια έχει ως αποτέλεσμα την μείωση του ποσοστού προώθησης πακέτων των μη εγωιστών κόμβων ή τουλάχιστον την όχι τόσο ταχεία αύξηση του έως ότου φτάσει την μέγιστη τιμή. Το παραπάνω γράφημα επιβεβαιώνει αυτού του είδους τις υποθέσεις καθώς δείχνει μια όλο και πιο αργή αύξηση του ποσοστού προώθησης πακέτων όσο η περίοδος του CalcICAS αυξάνεται. Γίνεται εύκολα αντιληπτό άρα ότι σε ιδανικές συνθήκες θα επιθυμούσαμε όσων το δυνατό μικρότερη περίοδο, αλλά αυτού του είδους η πολυτέλεια έρχεται σε βάρος των πόρων του δικτύου καθότι πιο συχνή διακίνηση πληροφορίας συνεπάγεται μεγαλύτερο συνολικό όγκο δεδομένων που διακινούνται στο δίκτυο.

Τα αντίστοιχα αποτελέσματα για την εναλλαγή της περιόδου CalcNewKatastasi παρατίθενται παρακάτω.

Runtime in secs	ICARUS CalcNewKatastasi = 10sec	ICARUS CalcNewKatastasi = 30sec	ICARUS CalcNewKatastasi = 50sec
20	0.778412733	0.651376147	0.523364486
40	0.817679558	0.73641791	0.676375405
60	0.866425993	0.831404959	0.805293006
80	0.906801008	0.878440367	0.860776439
100	0.926499033	0.904302019	0.890269151
120	0.93956044	0.921052632	0.908326325
140	0.947820343	0.932456665	0.922096317
160	0.954960091	0.940206186	0.932026944
180	0.95887663	0.945627549	0.938544474
200	0.961951656	0.950242718	0.944123314
220	0.964011322	0.954645208	0.949041812
240	0.966801918	0.95831944	0.953081511
260	0.968474576	0.965782984	0.956505848
280	0.969896519	0.962585034	0.959404601
300	0.970536756	0.96492616	0.962204724
320	0.972176759	0.966962525	0.964075383

Πίνακας 5.9 - Ποσοστό προώθησης πακέτων των μη εγωιστών κόμβων στον μηχανισμό ICARUS με μεταβλητή τιμή της περιόδου CalcNewKatastasi

Κόμβοι	50
Εγωιστές	3
Περίοδος Calc ICAS (secs)	10
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	-
Περίοδος Calc Free(secs)	10



Διάγραμμα 5.9 - Ποσοστό προώθησης πακέτων των μη εγωιστών κόμβων στον μηχανισμό ICARUS με μεταβλητή τιμή της περιόδου CalcNewKatastasi	
Κόμβοι	50
Εγωιστές	3
Περίοδος Calc ICAS (secs)	10
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	-
Περίοδος Calc Free(secs)	10

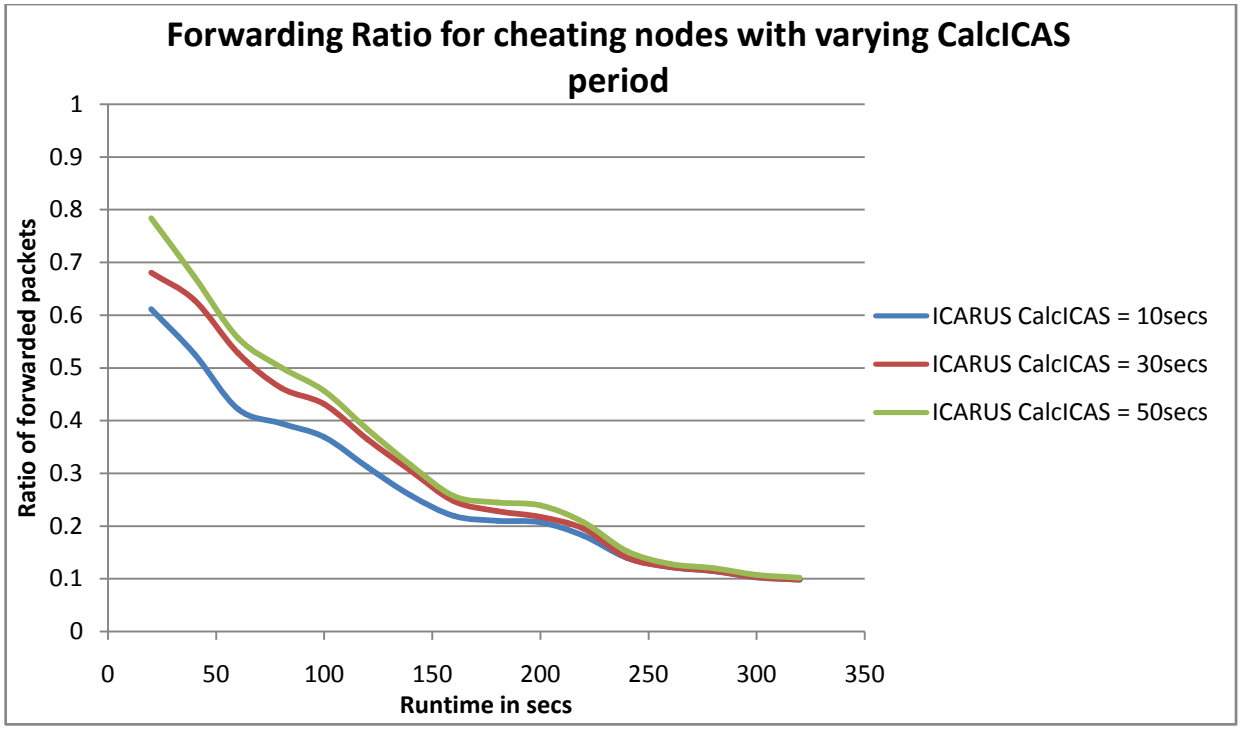
Η CalcNewKatastasi είναι υπεύθυνη για τον τακτικό υπολογισμό και μετάβασης της κατάστασης στην οποία βρίσκεται ένας κόμβος. Καθεμία από τις τρεις πιθανές καταστάσεις στις οποίες μπορεί να βρίσκεται ένας κόμβος (κοίτα ενότητα 3.3.2 για τις συνθήκες ένταξης σε κάθε κατάσταση), συνεπάγονται την δυνατότητα ή μη του κόμβου να στείλει ένα πακέτο. Με αυτό τον τρόπο η κεντρική οντότητα δίνει την άδεια στους μη εγωιστές κόμβους να αποστείλουν πακέτα και αφαιρεί αυτό το δικαίωμα από τους εγωιστές κόμβους. Όπως φαίνεται και σε αυτή την περίπτωση καθώς η περίοδος ανανέωσης της κατάστασης του κάθε κόμβου αυξάνεται μειώνεται και η συνολική αποτελεσματικότητα του δικτύου έχοντας δηλαδή μείωση του ποσοστού προώθησης πακέτων των μη εγωιστών κόμβων.

Τέλος, παρουσιάζονται τα αριθμητικά αποτελέσματα του ποσοστού προώθησης πακέτων (Packet Forwarding Ratio) των εγωιστών κόμβων για τον μηχανισμό ICARUS. Η διαφορά όπως και παραπάνω είναι ότι αυτή την φορά θα υπάρχει μεταβολή των περιόδων υπολογισμού των CalcICAS και CalcNewKatastasi.

Runtime in secs	ICARUS CalcICAS = 10sec	ICARUS CalcICAS = 30sec	ICARUS CalcICAS = 50sec
20	0.61163292	0.680515019	0.783624097
40	0.52660782	0.628569626	0.672957399
60	0.42199252	0.528457013	0.557022109
80	0.39454336	0.462274619	0.500797832
100	0.36869341	0.43091411	0.456235973
120	0.31170857	0.36430004	0.383301794
140	0.25790351	0.304635626	0.315965327
160	0.21900357	0.247303211	0.256599913
180	0.21005249	0.228535009	0.244805674
200	0.2072366	0.217161782	0.239134457
220	0.18164257	0.194736965	0.207205129
240	0.1398522	0.140217354	0.152623503
260	0.12230685	0.122586566	0.12837327
280	0.11480644	0.11970101	0.125330715
300	0.10279045	0.102996031	0.107297811
320	0.098566227	0.098369095	0.101854396

Πίνακας 5.10 - Ποσοστό προώθησης πακέτων των εγωιστών κόμβων στον μηχανισμό ICARUS με μεταβλητή τιμή της περιόδου CalcICAS

Κόμβοι	50
Εγωιστές	3
Περίοδος Calc ICAS (secs)	-
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	10
Περίοδος Calc Free(secs)	10



Διάγραμμα 5.10 - Ποσοστό προώθησης πακέτων των εγωιστών κόμβων στον μηχανισμό ICARUS με μεταβλητή τιμή της περιόδου CalcICAS	
Κόμβοι	50
Εγωιστές	3
Περίοδος Calc ICAS (secs)	-
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	10
Περίοδος Calc Free(secs)	10

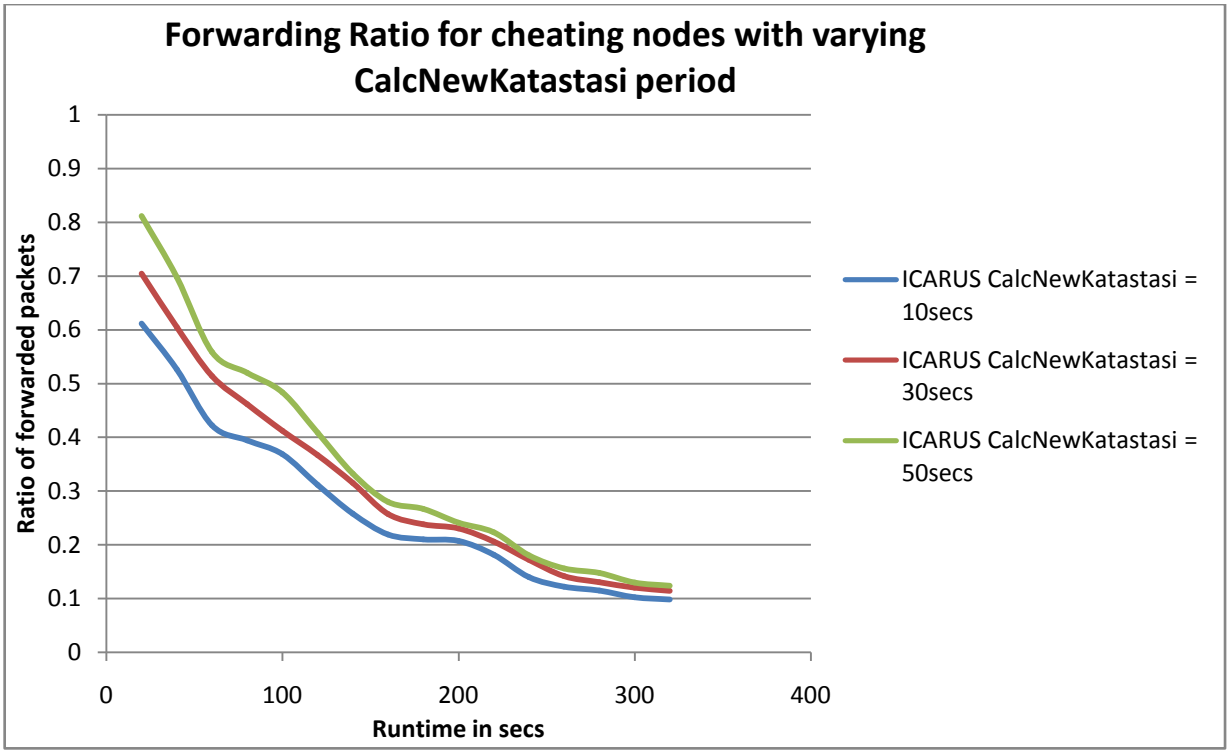
Εν συντομία, η βασική λειτουργία του CalcICAS όπως αναφέρθηκε και στον σχολιασμό του διαγράμματος 5.5, είναι ο εντοπισμός των εγωιστών κόμβων. Συνεπώς αυξάνοντας της περίοδο του CalcICAS, η συχνότητα ελέγχου και εντοπισμού ενός εγωιστή κόμβου αναμένεται να φθίνει. Αυτό με την σειρά του έχει ως αποτέλεσμα το αυξημένο ποσοστό προώθησης των εγωιστών κόμβων, όσο αυξάνεται η τιμή της περιόδου του CalcICAS. Πιο συγκεκριμένα όσο πιο μεγάλη είναι η περίοδος αυτή τόσο πιο αργά θα γίνεται ο εντοπισμός των εγωιστών, με συνέπεια να αργεί ο αλγόριθμος να μειώσει το ποσοστό προώθησης αυτών των κόμβων. Η συγκεκριμένη λογική επιβεβαιώνεται πλήρως από την παραπάνω γραφική όπου φαίνεται ότι όσο πιο μεγάλη είναι η τιμή της περιόδου τόσο πιο αργά πραγματοποιείται η μείωση του ποσοστού προώθησης των εγωιστών κόμβων και άρα τόσο μικρότερη και η αποτελεσματικότητα του μηχανισμού. Συνεπώς υπό ιδανικές συνθήκες θα θέλαμε να έχουμε όσο το δυνατό μικρότερη τιμή ανανέωσης των μετρητών του ICAS αλλά αυτό συνεπάγεται και μεγαλύτερη επιβάρυνση σε πόρους του δικτύου γιατί η συχνή διακίνηση πληροφορίας συνεπάγεται μεγαλύτερο συνολικό όγκο δεδομένων που διακινούνται στο δίκτυο. Ωστόσο με αυτό τον τρόπο ανάλογα με τις ανάγκες του δίκτυο

στην εκάστοτε περίσταση, υπάρχει η δυνατότητα να θυσιαστούν πόροι του δικτύου για την ταχύτερη καταπολέμηση των εγωιστών κόμβων σε αυτό.

Runtime in secs	ICARUS CalcNewKatastasi = 10sec	ICARUS CalcNewKatastasi = 30sec	ICARUS CalcNewKatastasi = 50sec
20	0.61163292	0.704380936	0.811716397
40	0.52660782	0.604867008	0.697734297
60	0.42199252	0.513869575	0.558612598
80	0.39454336	0.461319824	0.519830604
100	0.36869341	0.412284032	0.483784745
120	0.31170857	0.366949563	0.40846291
140	0.25790351	0.315170984	0.332424729
160	0.21900357	0.2576358	0.279939123
180	0.21005249	0.238041984	0.267056535
200	0.2072366	0.2305777	0.241016166
220	0.18164257	0.206247873	0.223255248
240	0.1398522	0.172124494	0.180777149
260	0.12230685	0.141823354	0.156337508
280	0.11480644	0.130803353	0.147527254
300	0.10279045	0.120543286	0.129824338
320	0.098566227	0.114217558	0.124283141

Πίνακας 5.11 - Ποσοστό προώθησης πακέτων των εγωιστών κόμβων στον μηχανισμό ICARUS με μεταβλητή τιμή της περιόδου CalcNewKatastasi

Κόμβοι	50
Εγωιστές	3
Περίοδος Calc ICAS (secs)	10
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	-
Περίοδος Calc Free(secs)	10



Διάγραμμα 5.11 - Ποσοστό προώθησης πακέτων των εγωιστών κόμβων στον μηχανισμό ICARUS με μεταβλητή τιμή της περιόδου CalcNewKatastasi	
Κόμβοι	50
Εγωιστές	3
Περίοδος Calc ICAS (secs)	10
Περίοδος Calc Darwin(secs)	10
Περίοδος Calc NewKatastasi(secs)	-
Περίοδος Calc Free(secs)	10

Η CalcNewKatastasi όπως αναφέρθηκε και στον σχολιασμό του διαγράμματος 5.6 είναι υπεύθυνη για τον τακτικό υπολογισμό και μετάβασης της κατάστασης στην οποία βρίσκεται ένας κόμβος. Καθεμία από τις τρεις πιθανές καταστάσεις στις οποίες μπορεί να βρίσκεται ένας κόμβος, συνεπάγονται την δυνατότητα ή μη του κόμβου να στείλει ένα πακέτο. Με αυτό τον τρόπο η κεντρική οντότητα αφαιρεί το δικαίωμα τον εγωιστών κόμβων να αποστέλλουν πακέτα και οδηγεί στην καταπολέμηση των. Έτσι όσο πιο συχνά γίνεται αυτή η ενημέρωση/έλεγχος τόσο πιο αποτελεσματικά αντιμετωπίζονται οι εγωιστές κόμβοι αφού τους αφαιρείται η δυνατότητα αποστολής πακέτων και μειώνεται έτσι το ποσοστό προώθησης των πακέτων τους. Στην αντίθετη περίπτωση βέβαια (αύξηση της περιόδου) οι εγωιστές συνεχίζουν για μεγαλύτερα χρονικά διαστήματα να είναι λειτουργικοί, γεγονός το οποίο καθιστά τον μηχανισμό κινήτρων λιγότερο αποτελεσματικό. Αυτό διαφαίνεται εντόνως στο παραπάνω διάγραμμα όπου το ποσοστό προώθησης των εγωιστών κόμβων στις περιπτώσεις των περιόδων 30 και 50 δευτερολέπτων, αργεί χαρακτηριστικά να μειωθεί και εν τέλει δεν μειώνεται όσο στην περίοδο των 10 δευτερολέπτων.

5.3 Συμπεράσματα

Συμπερασματικά, όσων αφορά την σύγκριση του με τον μηχανισμό DARWIN ο προτεινόμενος υβριδικός μηχανισμός ICARUS αποδείχθηκε καλύτερος στα περισσότερα σενάρια προσομοίωσης. Πέραν όμως των συγκριτικών πλεονεκτημάτων του σε σχέση με τον DARWIN ο ICARUS αποδείχθηκε ιδιαίτερα αποτελεσματικός στις λειτουργίες του χωρίς να παρουσιάζει ιδιαίτερες αδυναμίες στα γενικότερα πλαίσια των μηχανισμών κινήτρων συνεργασίας.

Τέλος, στην μελέτη που έγινε στην ενότητα 5.2.2 για την αντίδραση του μηχανισμού σε μεταβολές των περιόδων CalcICAS και CalcNewKatastasi, δείχθηκε όπως ήταν άλλωστε και αναμενόμενο ότι όσο μικρότερες είναι οι τιμές αυτών των περιόδων, δηλαδή όσο συχνότερα γίνονται οι υπολογισμοί, τόσο αποτελεσματικότερος ήταν ο αλγόριθμος στην αντιμετώπιση των εγωιστών κόμβων στο δίκτυο. Αυτή η αυξημένη αποτελεσματικότητα έρχεται βέβαια με το κόστος των πόρων του δικτύου, καθότι η πιο συχνή διακίνηση πληροφορίας συνεπάγεται μεγαλύτερο συνολικό όγκο δεδομένων που διακινούνται στο δίκτυο. Συνεπώς υπό ιδανικές συνθήκες θα θέλαμε να έχουμε όσο το δυνατό μικρότερη τιμή ανανέωσης των μετρητών του ICAS αλλά στις προσομοιώσεις επιλέχθηκε η τιμή των 10secs γιατί επιτυγχάνει ικανοποιητικά αποτελέσματα προώθησης πακέτων χωρίς να είναι παράλογα απαιτητική σε πόρους δικτύου.

6. ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ

Οι όλο και αυξανόμενες τεχνολογικές ανάγκες της σύγχρονης κοινωνίας, καθιστούν απαραίτητη την ύπαρξη των ασύρματων επικοινωνιών. Τα ad-hoc δίκτυα έρχονται να καλύψουν τις ανάγκες της σημερινής αλλά και μελλοντικής αγοράς στον τομέα των τηλεπικοινωνιών. Το μεγάλο εύρος κινητικότητας των κόμβων, αυτών των δικτύων καθώς και η δυνατότητα των χρηστών των να μπαίνουν και να βγαίνουν από αυτά χωρίς να έχουν ανάγκη από διαφορετικές οντότητες, κάνουν την ιδέα για εφαρμογή τους στην σύγχρονη κοινωνία ιδιαίτερα θελκτική. Αυτή η ιδέα ενισχύεται από την ένταξη των ad-hoc δικτύων στα μελλοντικά δίκτυα τέταρτης γενιάς(4G). Συνεπώς γίνεται εύκολα αντιληπτό ότι η έρευνα θα πρέπει να επικεντρωθεί στην βελτιστοποίηση των παρεχόμενων υπηρεσιών των ad-hoc δικτύων. Σημαντικό ρόλο στην βελτίωση των υπηρεσιών παίζει βέβαια η αποτελεσματική αντιμετώπιση των εγωιστών αλλά και κακόβουλων κόμβων. Αυτό όπως όλα δείχνουν οι σημερινές μελέτες, επιτυγχάνεται καλύτερα από τους υβριδικούς μηχανισμούς κινήτρων. Άρα η έρευνα θα πρέπει να ενταθεί στον τομέα της ανάπτυξης νέων αποτελεσματικότερων υβριδικών μηχανισμών.

Δεδομένων των αποτελεσμάτων της προσομοίωσης του ICARUS δεν θα ήταν παράλογο να ειπωθεί ότι θα άξιζε τον κόπο η περεταίρω έρευνα στον συγκεκριμένο υβριδικό μηχανισμό. Περισσότερα συμπεράσματα επί της αποτελεσματικότητας του μηχανισμού αυτού θα μπορούσαν να προκύψουν από την μελέτη του για διαφορετικά μεγέθη παραμέτρων και συνθηκών σε κάποιο άλλο σενάριο προσομοίωσης.