



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

**Υβριδικοί μηχανισμοί κινήτρων συνεργασίας σε αδόμητα ασύρματα
δίκτυα 4^{ης} γενιάς και άνω**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
του
ΑΠΟΣΤΟΛΑΚΗ ΝΙΚΟΛΑΟΥ

Επιβλέπων: Αθανάσιος Δ. Παναγόπουλος
Αναπλ. Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2017

Η σελίδα αυτή είναι σκόπιμα λευκή



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Υβριδικοί μηχανισμοί κινήτρων συνεργασίας σε ασύρματα αδόμητα δίκτυα 4^{ης} γενιάς και άνω

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΑΠΟΣΤΟΛΑΚΗ ΝΙΚΟΛΑΟΥ

Επιβλέπων: Αθανάσιος Δ. Παναγόπουλος
Αναπλ. Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή.

(Υπογραφή)

.....

Αθανάσιος Παναγόπουλος
Αναπλ. Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....

Παναγιώτης Κωττής
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....

Γεώργιος Φικιώρης
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2017

(Υπογραφή)

.....

ΑΠΟΣΤΟΛΑΚΗΣ ΝΙΚΟΛΑΟΣ

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

Copyright © Αποστολάκης Νικόλαος, 2017

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Η σελίδα αυτή είναι σκόπιμα λευκή

ΠΕΡΙΛΗΨΗ

Με την ανάπτυξη των δικτύων 4^{ης} και νέας γενιάς, η ιδέα των ασύρματων αδόμητων δικτύων άρχισε να υλοποιείται. Τα δίκτυα αυτού του τύπου είναι αποκεντρωμένα και δεν χαρακτηρίζονται από μία κεντρική οντότητα που το εποπτεύει ούτε βασίζονται σε μία ήδη σταθερή και προϋπάρχουσα δομή. Αντιθέτως, αποτελούνται από απλούς και ισότιμους κόμβους που ανάλογα με τη κατάσταση του δικτύου παίζουν το ρόλο είτε των τερματικών σταθμών είτε των δρομολογητών. Η είσοδος των κόμβων σε ένα τέτοιο δίκτυο δεν φιλτράρεται από έναν κόμβο-επόπτη και επομένως το πλέγμα των κόμβων που σχηματίζεται μπορεί να λάβει γρήγορα μεγάλες διαστάσεις.

Σε ένα τέτοιο δίκτυο, οι προθέσεις των διαφόρων κόμβων δεν συμβάλλουν πάντα στην βελτίωση της συνολικής απόδοσης του δικτύου, καθώς ενδέχεται να παρουσιάζουν εγωιστικές ή κακόβουλες συμπεριφορές και να το υποβαθμίζουν. Για τη συμμόρφωση τους, κρίνεται απαραίτητη η ανάπτυξη μηχανισμών κινήτρων που να ενθαρρύνουν τη συνεργατικότητα και να τιμωρούν τον ατομικισμό.

Οι δύο μεγάλες κατηγορίες μηχανισμών κινήτρων είναι οι μηχανισμοί φήμης (reputation-based mechanisms) και οι μηχανισμοί πίστωσης (credit-based mechanisms).

Στο πλαίσιο της παρούσας διπλωματικής, παρατίθενται τα βασικά στοιχεία που συνθέτουν τους μηχανισμούς φήμης και πίστωσης ενώ περιγράφονται ορισμένοι που έχουν ήδη αναπτυχθεί από τη διεθνή βιβλιογραφία. Εν συνεχεία, αναπτύσσονται τα πλεονεκτήματα που προσφέρουν οι υβριδικοί μηχανισμοί κινήτρων (hybrid mechanisms), οι οποίοι αποτελούν συνδυασμό των άνωθεν μηχανισμών. Στο τέταρτο κεφάλαιο παρουσιάζεται αναλυτικά ο υβριδικός μηχανισμός ICARUS 2, που αποτελεί μία προέκταση του ήδη υπάρχοντος πρωτοκόλλου ICARUS που είχε αναπτυχθεί από την ερευνητική ομάδα του ΕΜΠ. Τέλος, παρατίθενται μία σειρά από προσομοιώσεις που εκτελέστηκαν με τη χρήση του ICARUS 2 καθώς και τα αντίστοιχα αποτελέσματα με σκοπό τη μελέτη της αποτελεσματικότητάς του. Εξάγονται ενδιαφέροντα συμπεράσματα.

Λέξεις Κλειδιά: 4G, δίκτυα νέας γενιάς, ασύρματα δίκτυα, αδόμητα δίκτυα, μηχανισμοί κινήτρων συνεργασίας, μηχανισμοί φήμης, μηχανισμοί πίστωσης, υβριδικοί μηχανισμοί κινήτρων, βελτιστοποίηση επίδοσης δικτύου, προγραμματισμός, JAVA

ABSTRACT

Upon the growth of fourth and new generation networks, the concept of wireless ad-hoc networks has started to be implemented. This type of network is decentralized and is not characterized by a central authority that inspects it and neither is based on an existing constant (either wired or wireless) infrastructure. On the contrary, it consists of simple and equivalent nodes, which act either as terminal nodes or as router nodes according to the network state. The entrance of nodes into the ad-hoc networks is not filtered by an inspector and results to a network grid that is shaped and that can grow rapidly.

In ad-hoc networks, the intentions of nodes do not always contribute to the improvement of the overall network performance, as they may show selfish or malicious behavior which downgrade the quality of service that the network serves. For their compliance, it would be necessary to develop incentive mechanisms that encourage nodes to cooperate and punish the individualism strictly.

Incentive mechanisms are divided into two great categories: reputation-based mechanisms and credit-based mechanisms.

In the context of the current diploma thesis, the basic entities of the reputation and credit-based mechanisms are indicated and some examples are cited, that have already been developed and applied. Subsequently, an introduction to hybrid mechanisms is made, which are a combination of the reputation and credit based mechanisms, and their advantages are explained. In chapter four, the hybrid mechanism ICARUS 2, an updated extension of the existing ICARUS protocol that has been developed in the context of the thesis, is analytically cited. Finally, some simulations to measure the algorithm efficiency in a network, that makes use of ICARUS 2 protocol, are executed and their results are presented.

Key words: 4G, next generation networks, wireless ad hoc networks, incentive cooperation mechanisms, reputation-based mechanisms, credit-based mechanisms, hybrid incentive mechanisms, network performance measurement and improvement, programming, JAVA

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα πρωτίστως να ευχαριστήσω θερμά τον Αναπλ. Καθηγητή ΕΜΠ κ. Αθανάσιο Παναγόπουλο, για την εμπιστοσύνη που μου έδειξε στην ανάθεση του συγκεκριμένου θέματος της διπλωματικής εργασίας, το οποίο συνέβαλε στην επέκταση των γνώσεών μου σ' ένα ιδιαίτερα ενδιαφέρον επιστημονικό πεδίο όπως αυτό των ασυρμάτων επικοινωνιών. Επιπρόσθετα, θα ήθελα να τον ευχαριστήσω για την πολύτιμη βοήθεια που μου προσέφερε κάθε φορά που αυτό ήταν απαραίτητο καθώς και για τις άρτια επιστημονικές γνώσεις που μου παρείχε όλα αυτά τα χρόνια μέσα από τα ακαδημαϊκά έδρανα.

Επιπλέον, θα ήθελα να ευχαριστήσω τον απόφοιτο Διδάκτορα Μηχανικό ΕΜΠ κ. Δημήτρη Χαρίλα, για τη συνεχή υποστήριξη κατά την εκπόνηση της διπλωματικής εργασίας μου και τη διαρκή θέληση που επέδειξε να με βοηθήσει κάθε φορά που απευθυνόμουν σε αυτόν.

Τέλος θα ήθελα να ευχαριστήσω από καρδιάς μου όλους τους φίλους μου που ήταν κοντά μου σε όλη την διάρκεια της φοιτητικής μου πορείας με τους οποίους μοιραστήκαμε αυτά τα πέντε υπέροχα χρόνια και κυρίως τους γονείς μου Κωνσταντίνο και Ελισάβετ όπως επίσης και τα αγαπημένα μου αδέρφια Ευγενία και Μηνά για όλη τη στήριξη και την αγάπη που απλόχερα μου έδωσαν, παρέχοντάς μου πίστη και δύναμη για να συνεχίζω σε κάθε νέο μου βήμα.

Η σελίδα αυτή είναι σκόπιμα λευκή

ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη	1
Abstract	1
Ευχαριστίες	1
Περιεχόμενα	1
1 Εισαγωγή στους μηχανισμούς κινήτρων	1
1.1 Εισαγωγή στα δίκτυα νέας γενιάς	1
1.2 Δίκτυα δημιουργούμενα από το χρήστη	2
1.2.1 Πρόκληση	2
1.2.2 Εφαρμογές	3
1.3 Μηχανισμοί κινήτρων	4
1.4 Μηχανισμοί πίστωσης	5
1.5 Μηχανισμοί φήμης	5
1.5.1 Απαιτήσεις	6
1.5.2 Ταξινόμηση μηχανισμών φήμης	6
1.6 Κοινά προβλήματα μηχανισμών κινήτρων	7
1.7 Σύγκριση των μηχανισμών φήμης και πίστωσης	8
1.8 Περιγραφή μηχανισμών κινήτρων συνεργασίας	9
1.8.1 CONFIDANT	9
1.8.2 CORE	10
1.8.3 SORI	10
1.8.4 OCEAN	11
1.8.5 SPRITE	11
1.8.6 Token-Based Cooperation Enforcement	12
1.8.7 Ad hoc-VCG	13
1.9 Υβριδικοί μηχανισμοί κινήτρων	13
1.9.1 Θεωρία παιγνίων σε συστήματα φήμης	14
1.9.2 Θεωρία παιγνίων σε συστήματα πίστωσης	15
1.9.3 Θεωρία παιγνίων σε υβριδικούς μηχανισμούς	16
1.9.4 HEAD	17
2 Μηχανισμός Darwin	20

2.1	Θεωρητικό υπόβαθρο.....	20
2.2	Στρατηγικές κινήτρων	21
2.2.1	Tit For Tat	21
2.2.2	Generous Tit for Tat.....	22
2.3	Μηχανισμός Darwin.....	22
2.3.1	Εφαρμογή του αλγορίθμου	23
2.3.2	Πλεονεκτήματα – Μειονεκτήματα	24
3	Προτεινόμενος υβριδικός μηχανισμός ICARUS 2	26
3.1	Βασικά Στοιχεία	26
3.2	Μοντελοποίηση δικτύου	27
3.3	Δρομολόγηση.....	28
3.4	Στρατηγική εγωιστικών κόμβων	31
3.5	Στρατηγική Κακόβουλων Κόμβων	31
3.6	Συνεργασία κόμβων στον ICARUS 2	32
3.6.1	Λειτουργίες του ICAS	33
3.6.2	Τιμολόγηση στο ICARUS 2.....	34
3.6.3	Εντοπισμός εγωιστικών κόμβων.....	36
3.6.4	Εντοπισμός κακόβουλων κόμβων	36
3.6.5	Απομακρυσμένοι κόμβοι	37
4	Προσομοίωση μηχανισμών Darwin και ICARUS 2.....	40
4.1	Βασικά Στοιχεία Προσομοίωσης	40
4.2	Προσομοίωση	41
4.2.1	Maven: Σύστημα διαχείρισης	42
4.3	Περιγραφή κλάσεων.....	43
4.3.1	Ανάλυση των υπηρεσιών	43
4.3.2	Προσομοίωση κόμβων.....	50
4.3.3	Προσομοίωση <i>ICAS</i>	56
4.3.4	Εκτύπωση στατιστικών	60
4.3.5	Κοινές παράμετροι και σταθερές	61
4.3.6	Εφαρμογή Android για την εκτύπωση στατιστικών	61
5	Αποτελέσματα Προσομοίωσης.....	66
5.1	Παράμετροι προσομοίωσης	66

5.2	Αριθμητικά αποτελέσματα και γραφικές παραστάσεις.....	67
5.2.1	Σύγκριση του ICARUS 2 με το DARWIN.....	67
5.2.2	Αναλυτική μελέτη του μηχανισμού ICARUS 2	75
5.2.3	Συμπεράσματα.....	82
6	Προτάσεις για μελλοντική έρευνα.....	84
7	Βιβλιογραφία	85

1 ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΜΗΧΑΝΙΣΜΟΥΣ ΚΙΝΗΤΡΩΝ

1.1 ΕΙΣΑΓΩΓΗ ΣΤΑ ΔΙΚΤΥΑ ΝΕΑΣ ΓΕΝΙΑΣ

Τα τελευταία χρόνια των συνεχώς αυξανόμενων τεχνολογικών αναγκών, οι ασύρματες επικοινωνίες αποτελούν ένα αναπόσπαστο κομμάτι της καθημερινότητας των ανθρώπων. Το κινητό τηλέφωνο έχει εξελιχθεί από μία μικρού μεγέθους και περιορισμένης επεξεργαστικής ισχύς συσκευή σε ένα υψηλής ευκρίνειας και επαυξημένων δυνατοτήτων μηχάνημα, άμεσα προσβάσιμο στο μέσο άνθρωπο.

Η ναυαρχίδα της εξέλιξης των κινητών επικοινωνιών ήταν τα δίκτυα 2ης γενιάς (Second Generation ή 2G). Το κύριο πρότυπο των δικτύων 2ης γενιάς ήταν το GSM (Global System for Communications) που προσέφερε ικανοποιητική ποιότητα φωνής και αποδοτική χρησιμοποίηση του φάσματος κάνοντας χρήση τεχνικών ψηφιακής διαμόρφωσης. Τα συστήματα 2G εισήχθησαν στην αγορά το 1991 και βρίσκονται σε λειτουργία ακόμα και σήμερα.

Το βασικό μειονέκτημα του GSM είναι ότι περιορίζονται αποκλειστικά σε υπηρεσίες τηλεφωνίας και SMS. Με την παράλληλη άνοδο της πληροφορικής και του παγκοσμίου ιστού, δημιουργήθηκε η ανάγκη για παροχή υπηρεσιών δεδομένων. Για το σκοπό αυτό, αναπτύχθηκαν τα ασύρματα δίκτυα τοπικής περιοχής (Wireless Local Area Networks - WLAN) τα οποία κατάφεραν να παρέχουν υψηλές ταχύτητες μετάδοσης δεδομένων. Ωστόσο, τα δίκτυα αυτά είχαν πολύ μικρή εμβέλεια (μόλις μερικά δεκάδες μέτρα) και διαδόθηκαν λόγω της ευκολίας εγκατάστασης και χρήσης τους για την κάλυψη μικρών χώρων, π.χ. σπίτια, σχολεία, μικρές εταιρείες.

Μέχρι το 2001, ο αριθμός των ανθρώπων που είχαν πρόσβαση στο δίκτυο ξεπερνούσε το 500 εκατομμύρια [1]. Την ίδια χρονιά, εισήχθη το πρώτο δίκτυο 3ης γενιάς (Third Generation ή 3G) γνωστό ως UMTS (Universal Mobile Telecommunication System). Το UMTS προσέφερε υπηρεσίες πλοήγησης ιστού (web browsing), βίντεο κλήσεις και κινητής τηλεόρασης με ταχύτητες μετάδοσης που κυμαίνονται από 128kbps έως και 2mbps αναλόγως τη ταχύτητα της κινητής συσκευής. Πρόσθετο πρότυπο του 3G είναι η τεχνολογία HSPA (High Speed Packet Access) η οποία ενίσχυσε την απόδοση του UMTS, επανακαθορίζοντας τα πρωτόκολλα που χρησιμοποιούν οι κινητές συσκευές και οι σταθμοί βάσης για να επικοινωνήσουν [2].

Παρότι το 3G παρέχει υψηλότερο ρυθμό μετάδοσης από το 2G, η ανάγκη των παρόχων να ικανοποιήσουν την όλο και αυξανόμενη απαίτηση για ευρυζωνικές ασύρματες επικοινωνίες με χαμηλή καθυστέρηση και υψηλές ταχύτητες οδήγησε στην προτυποποίηση της τεχνολογίας LTE (Long Term Evolution). Το LTE ταυτίστηκε με τη μετάβαση από το 3G στα δίκτυα 4ης γενιάς (4G).

Το LTE προσφέρει αποδοτική χρησιμοποίηση των πόρων του δικτύου και υψηλή ποιότητα υπηρεσιών. Κάνει χρήση της OFDMA τεχνικής ψηφιακής διαμόρφωσης για την κάτω (downlink) ζεύξη και SCFDMA (Single carrier FDMA) για την πάνω (uplink) ζεύξη. Οι προδιαγραφές του αναφέρουν 14.4mbps μετάδοση για την downlink ζεύξη και 384kbps για την uplink ζεύξη. Το LTE εισήχθη το 2011 και μέχρι το τέλος του 2012 μετρούσε 60 εκατομμύρια συνδρομητές, ενώ ο συνολικός αριθμός των συνδεδεμένων συσκευών ανερχόταν στα 7 δισεκατομμύρια [2]. Το νούμερο αυτό αναμένεται να φτάσει τα 9 δισεκατομμύρια το 2021 [3], με την εξέλιξη των Machine to Machine (M2M) επικοινωνιών.

Μέχρι το 2021 αναμένεται να έχει προτυποποιηθεί και η πέμπτη γενιά ασύρματων δικτύων (5G). Το 5G θα ικανοποιεί ακόμα πιο απαιτητικές σε όγκο δεδομένων εφαρμογές προσφέροντας υψηλή ταχύτητα μεταφοράς, αξιοπιστία και μειωμένη καθυστέρηση. Λόγω της ευελιξίας του, θα έχει τη δυνατότητα να συνδέει ένα τεράστιο αριθμό συσκευών, όπως αυτοκίνητα και αισθητήρες, στο Internet σχηματίζοντας το διαδίκτυο των πραγμάτων (Internet of Things ή IoT).

1.2 ΔΙΚΤΥΑ ΔΗΜΙΟΥΡΓΟΥΜΕΝΑ ΑΠΟ ΤΟ ΧΡΗΣΤΗ

Η μαζική συγκέντρωση έξυπνων ασύρματων συσκευών σε μικρούς χώρους δημιουργεί έναν εναλλακτικό τρόπο επικοινωνίας που δίνει τη δυνατότητα στους χρήστες να ανταλλάσσουν δεδομένα χωρίς την παρουσία ενός εξωτερικού παρόχου υπηρεσιών διαδικτύου (Internet Service Provider ή ISP). Τέτοια δίκτυα αποτελούνται από κόμβους που επικοινωνούν μεταξύ τους ασύρματα και δεν βασίζονται σε μία ήδη προϋπάρχουσα και σταθερή δομή (infrastructure). Δηλαδή, δεν υπάρχει η έννοια του δρομολογητή ή ενός κεντρικού σημείου πρόσβασης (access point ή AP). Αντίθετα, ο κάθε κόμβος του δικτύου έχει τη δυνατότητα για άμεση επικοινωνία με κάθε άλλο κόμβο αφού όλοι οι κόμβοι λειτουργούν ταυτόχρονα ως δρομολογητές. Αυτά τα δίκτυα αναφέρονται στη βιβλιογραφία ως δίκτυα δημιουργούμενα από το χρήστη (User Provided Network ή UPN) ενώ συναντάται και η ορολογία Mobile Ad-Hoc Network (MANET). Τα UPN στις μέρες μας είναι ιδιαίτερα σημαντικά. Οι εφαρμογές γίνονται όλο και πιο απαιτητικές από την άποψη κατανάλωσης πόρων. Γι' αυτό η προοπτική διαμοιρασμού του κόστους εργασίας ενός κόμβου σε περισσότερους, προκειμένου να παρέχουν καλύτερη ποιότητα υπηρεσίας, είναι ιδιαίτερα θελκτική.

1.2.1 Πρόκληση

Η πρόκληση που συναντούν οι σχεδιαστές δικτύων UPN είναι η βελτίωση της τελικής εμπειρίας του χρήστη (Quality of Experience ή QoE). Οι διαδικασίες λήψης αποφάσεων, διαμοιρασμού του

φόρτου εργασίας και ανταλλαγής δεδομένων μεταξύ των κόμβων πρέπει να είναι διαφανείς. Παράλληλα, εγείρονται ζητήματα ασφάλειας των δεδομένων και συνολικού χρόνου εξυπηρέτησης των αιτημάτων.

1.2.2 Εφαρμογές

Η χρησιμότητα ύπαρξης των Ad-Hoc δικτύων εντοπίζεται σε μία ευρεία γκάμα εφαρμογών.

- Στρατιωτικοί σκοποί και περιπτώσεις φυσικών καταστροφών.
- Εφαρμογές που αφορούν οχήματα (vehicular applications) για διαμοιρασμοί διαφημίσεων ή πληροφοριών σχετικών με την κυκλοφοριακή συμφόρηση και χώρους στάθμευσης.
- Κοινωνικά δίκτυα για διαμοιρασμό πληροφοριών που αφορούν τον τουρισμό και τα νέα της τοπικής κοινωνίας. Ωστόσο, σε αντίθεση με τα οχήματα, οι εφαρμογές αυτές είναι περισσότερο ενεργειακά ακριβές, αφού οι κόμβοι αφορούν έξυπνα κινητά με περιορισμένους πόρους.

Ένα MANET είναι ένα σύνολο από ασύρματους κινητούς υπολογιστές ή κόμβους που προωθούν μεταξύ τους τα πακέτα για να επικοινωνήσουν. Τα πρωτόκολλα δρομολόγησης (**routing protocols**), που έχουν σχεδιαστεί για MANET, όπως το DSR (**dynamic source routing**) ή το AODV (**ad hoc on demand distance vector**) βασίζονται στην υπόθεση ότι όλοι οι κόμβοι συνεργάζονται.

Για την επιτυχή όμως λειτουργία των Ad-Hoc δικτύων απαιτείται η διασφάλιση της ποιότητας της υπηρεσίας που παρέχεται. Είναι μείζονος σημασίας η εξασφάλιση της απρόσκοπτης χρήσης και συνεργασίας μεταξύ των κόμβων που σε ένα δίκτυο δίχως την παρουσία μίας κεντρικής οντότητας δεν είναι δεδομένη. Ενδέχεται να υπάρχουν κόμβοι, που προσπαθούν να επωφεληθούν της απουσίας αυτής και επιχειρούν να καταχραστούν τους κόμβους του δικτύου. Σε ορισμένες περιπτώσεις αρνούνται να προωθήσουν τα πακέτα που καταφθάνουν σε αυτούς προκειμένου να συντηρήσουν τους περιορισμένους τους πόρους (ενεργειακή κατανάλωση, επεξεργαστική ισχύς, μνήμη). Αυτοί οι κόμβοι αναφέρονται ως κακόβουλοι, δεν συμβάλλουν στη συνεργατικότητα του δικτύου και το υποβαθμίζουν. Για τον σκοπό αυτό, κρίνεται απαραίτητη η ανάπτυξη κινήτρων συνεργασίας (incentive mechanisms) που θα αποτρέπουν τους κακόβουλους κόμβους να αποκλίνουν και θα τους τιμωρούν ενώ θα επιβραβεύουν τους συνεργατικούς.

Στο πλαίσιο της διπλωματικής, θα γίνει παρουσίαση των μηχανισμών κινήτρων συνεργασίας σε Ad-Hoc δίκτυα, στη συνέχεια θα γίνει μία εισαγωγή στους υβριδικούς μηχανισμούς κινήτρων και θα παρουσιαστούν τα αποτελέσματα της προσομοίωσης του προτεινόμενου υβριδικού μηχανισμού ICARUS 2 που αποτελεί επέκταση του μηχανισμού ICARUS [4].

1.3 ΜΗΧΑΝΙΣΜΟΙ ΚΙΝΗΤΡΩΝ

Η διαδικασία της δρομολόγησης στα ασύρματα Ad-Hoc δίκτυα νέας γενιάς μπορεί να υποφέρει από διάφορα κενά ασφαλείας και να θέτει σε κίνδυνο της συνολική απόδοση του δικτύου. Η αιτία του κινδύνου αυτού αποτελεί η άγνωστη για το δίκτυο ταυτότητα των αυτόνομων συσκευών που το απαρτίζουν και κυρίως ο άμεσος ρόλος που παίζουν για τη δρομολόγηση των πακέτων.

Στις περισσότερες εφαρμογές υποθέτουμε ότι γνωρίζουμε τη ταυτότητα όλων των κόμβων του δικτύου, οι οποίοι είναι διατεθειμένοι να συμβάλλουν στη δρομολόγηση. Αυτή η υπόθεση είναι βάσιμη σε στρατιωτικές εφαρμογές ή σε περιπτώσεις φυσικών καταστροφών, ενώ σε άλλες όπως τα κοινωνικά δίκτυα είναι αβάσιμη.

Για την διασφάλιση της ποιότητας υπηρεσίας σε ένα Ad-Hoc δίκτυο είναι απαραίτητο να ωθούμε τους κόμβους σε συνεργατική συμπεριφορά. Οι κόμβοι τείνουν να έχουν αντικρουόμενα συμφέροντα και θέλουν να βλάψουν ή να καταχραστούν τους πόρους του δικτύου προς όφελός τους. Πιο συγκεκριμένα αυτή η επιβλαβής συμπεριφορά μπορεί να διαχωριστεί σε 2 κατηγορίες:

- Στους εγωιστές κόμβους, οι οποίοι έχουν ως σκοπό να μεγιστοποιήσουν το κύρος τους (ή να ελαχιστοποιήσουν τους χρησιμοποιούμενους τους πόρους) σε βάρος των υπόλοιπων κόμβων. Για παράδειγμα, ένας κόμβος μπορεί να αρνείται να δρομολογήσει τα πακέτα που καταφθάνουν από άλλους κόμβους, προκειμένου να μειώσει την ενεργειακή του κατανάλωση που απαιτείται για το κόστος της προώθησης, ενώ παράλληλα να απαιτεί από τους υπόλοιπους να δρομολογούν τα δικά του πακέτα.
- Στους κακόβουλους κόμβους, οι οποίοι έχουν ως στόχο να βλάψουν την ποιότητα υπηρεσίας του ολικού δικτύου, με τρόπους όπως κρυφή αλλαγή πινάκων δρομολόγησης, απόρριψη πακέτων κλπ.

Οι εγωιστές κόμβοι, λοιπόν, δεν είναι πρόθυμοι να συμβάλλουν θετικά στο δίκτυο δίχως την παρουσία ισχυρών κινήτρων ενώ παράλληλα χρησιμοποιούν τους τίμιους κόμβους προς όφελός τους. Γι' αυτό κρίνεται απαραίτητη η παροχή ενός ισχυρού μηχανισμού κινήτρων συνεργασίας που να ενθαρρύνουν τη συνεργασία μεταξύ των κόμβων και να αποτρέπουν την εμφάνιση εγωιστικών συμπεριφορών. Ένας αποδοτικός μηχανισμός δίνει κίνητρο στους κόμβους να εκδηλώνουν τη διαθεσιμότητα των πόρων τους, να προσφέρουν τις υπηρεσίες τους και την αποθηκευτική τους δυνατότητα.

Οι μηχανισμοί κινήτρων τυγχάνουν μεγάλης προσοχής από την ερευνητική κοινότητα και μπορούν να χωριστούν σε δύο μεγάλες κατηγορίες: τους μηχανισμούς βασισμένους στη φήμη και τους μηχανισμούς πίστωσης.

1.4 ΜΗΧΑΝΙΣΜΟΙ ΠΙΣΤΩΣΗΣ

Οι μηχανισμοί πίστωσης (Credit-based incentive mechanisms) χρησιμοποιούν την έννοια του εικονικού νομίσματος (virtual currency) για να ρυθμίσουν την ανταλλαγή δεδομένων μέσα στο δίκτυο.

Τα εικονικά νομίσματα δίνονται στους κόμβους που συμβάλλουν θετικά και πραγματοποιούν διάφορες εργασίες [9]. Πιο συγκεκριμένα, κάθε φορά που ένας κόμβος δρα ως δρομολογητής επιβραβεύεται με έναν αριθμό νομισμάτων, με τα οποία μπορεί να ενθαρρύνει τους υπόλοιπους να συνεργαστούν μαζί του στο μέλλον.

Η ασφάλεια παίζει ένα καίριο ρόλο στους μηχανισμούς πίστωσης. Για την αποφυγή πλαστών πληρωμών, έχουν δημιουργηθεί διάφορες τεχνικές. Από τη μία, υπάρχει η δυνατότητα αποθήκευσης των νομισμάτων που έχει συλλέξει ένας κόμβος σε ξεχωριστό chip στο hardware της συσκευής (πχ στην κάρτα SIM - Subscriber Identity Module στα κινητά που υποστηρίζουν τεχνολογία GSM), κάνοντάς το δύσκολο να το παρακάμψει ή να αλλοιώσει το περιεχόμενό του [4].

Από την άλλη, πιο διαδεδομένη είναι η ύπαρξη μίας διαπιστευμένης οντότητας που εκτελεί χρέη εικονικής τραπέζης και την οποία όλοι οι κόμβοι εμπιστεύονται για τη διαφύλαξη των νομισμάτων τους. Κάθε φορά που ένας κόμβος επιχειρεί να στείλει ένα πακέτο, η τράπεζα δεσμεύει έναν μέρος του λογαριασμού του, το οποίο και μοιράζει στους κόμβους που έδρασαν ως δρομολογητές. Έτσι, οι κόμβοι έχουν κίνητρο να συνεργαστούν προκειμένου να αυξήσουν τον αριθμό των νομισμάτων τους στην τράπεζα και να αποστέλλουν τα δικά τους μηνύματα.

1.5 ΜΗΧΑΝΙΣΜΟΙ ΦΗΜΗΣ

Οι μηχανισμοί φήμης (Reputation-based incentive mechanisms) είναι η δεύτερη μεγάλη κατηγορία μηχανισμών κινήτρων. Οι μηχανισμοί αυτοί ποσοτικοποιούν και μετράνε την εγωιστική ή μη συμπεριφορά των κόμβων για να περιορίσουν τη δράση των κακόβουλων.

Σε αυτή τη στρατηγική κινήτρων, κάθε κόμβος εκτιμά το ποσοστό συνεργασίας για κάθε έναν από τους υπόλοιπους, σχηματίζει ένα μέγεθος που αποκαλούμε φήμη και προσφέρει καλύτερες υπηρεσίες σε αυτούς με την υψηλότερη φήμη. Όσο πιο συνεργατικός είναι ένας κόμβος, τόσο μεγαλύτερη πρόσβαση θα έχει στους κόμβους του δικτύου ενώ όσο πιο εγωιστικά συμπεριφέρεται τόσο πιο πολύ απομονώνεται, τιμωρείται και τελικά αποκόπτεται από το υπόλοιπο δίκτυο. Οι μηχανισμοί φήμης συνήθως χρησιμοποιούν tit-for-tat (TFT) [8] αλγορίθμους. Σε μία TFT στρατηγική ένας κόμβος μπορεί να μειώσει την αλληλεπίδραση με έναν άλλον αν αντιληφθεί ότι ο δεύτερος παρεκτρέπει.

Η φήμη είναι ένα δυναμικό μέγεθος το οποίο ευρίσκεται συγκεντρώνοντας όλες τις παρατηρήσεις των μελών του δικτύου και συνδυάζοντάς τα. Βάσει της τιμής της φήμης που έχει ο κάθε κόμβος μπορεί να χαρακτηριστεί ως κακόβουλος ή όχι.

1.5.1 Απαιτήσεις

Σύμφωνα με τις προδιαγραφές που θέσαμε παραπάνω, οι προϋποθέσεις ενός μηχανισμού φήμης είναι [10]:

- Τα μέλη του δικτύου να συλλέγουν πληροφορίες για την συνολική τους εμπειρία με τα υπόλοιπα μέλη και προαιρετικά να τις διαμοιράζουν μεταξύ τους.
- Θα πρέπει να υπάρχει ένας επαρκής αριθμός συμβάντων (events) για να εκτιμήσουν τη συνολική εμπειρία και να αποφανθούν για τη συμπεριφορά.
- Οι προηγούμενες παρατηρήσεις και εμπειρίες επηρεάζουν τις μελλοντικές αποφάσεις.
- Θα πρέπει να υπάρχουν σημαντικά κίνητρα για τα μέλη ώστε να διαμοιράζουν τις εμπειρίες και πληροφορίες.

1.5.2 Ταξινόμηση μηχανισμών φήμης

Υπάρχουν δύο υποκατηγορίες στις οποίες μπορούν να ταξινομηθούν οι μηχανισμοί φήμης. Η πρώτη κατηγορία αφορά τους μηχανισμούς στους οποίους ο κάθε κόμβος διατηρεί πληροφορίες μόνο για τους γείτονες από τους οποίους έχει απόσταση ενός βήματος (**local reputation schemes**) [5]. Η φήμη των γειτονικών κόμβων μπορεί να ανακτηθεί μέσω των επιβεβαιώσεων (**acknowledgements**) στο MAC (**medium access control**) στρώμα. Στη δεύτερη κατηγορία εμπίπτουν οι μηχανισμοί όπου κάθε κόμβος διατηρεί πληροφορίες για όλους τους κόμβους του δικτύου (**global reputation schemes**). Η πληροφορία για τους μακρινούς κόμβους μπορεί να ανακτηθεί μέσω των από άκρη-σε-άκρη επιβεβαιώσεων (**end to end ACKS**) ή μέσω διαμοιρασμού των πινάκων φήμης με τους γείτονες.

Τα *local reputation schemes* είναι κατάλληλα για δίκτυα όπου η τοπολογία των δικτύων είναι σχετικά στατική και οι κόμβοι έχουν αρκετό χρόνο να τιμωρήσουν τους κακόβουλους. Αντίθετα, όταν η τοπολογία αλλάζει με μεγαλύτερη συχνότητα, ο μηχανισμός φήμης μπορεί να μην επιβάλλει την συνεργατικότητα αποτελεσματικά. Γι' αυτό και στα δίκτυα με δυναμική και ταχέως μεταβαλλόμενη συχνότητα έχουν αποτελεσματικότερη εφαρμογή οι *global reputation* μηχανισμοί.

1.6 ΚΟΙΝΑ ΠΡΟΒΛΗΜΑΤΑ ΜΗΧΑΝΙΣΜΩΝ ΚΙΝΗΤΡΩΝ

Βέβαια και στις δύο κατηγορίες μηχανισμών κινήτρων μπορούμε να συναντήσουμε κάποια κοινά προβλήματα. Αρχικά, συμπεραίνουμε ότι στη μελέτη συμπεριφοράς των κόμβων δε λαμβάνονται υπόψη κάποια τυχαία γεγονότα που μπορούν να συμβούν κατά την ώρα της μετάδοσης ή μη των πακέτων. Παραδείγματα τέτοιων γεγονότων είναι:

- η σύγκρουση ενός πακέτου με ένα άλλο στο μέσο διάδοσης (πχ αέρα).
- η καθυστέρηση μετάδοσης λόγω δυσκολίας εύρεσης επόμενου κόμβου.
- η απόρριψη ενός πακέτου λόγω μεγάλη συμφόρησης στην ουρά εξόδου ή εισόδου.

Σαν αποτέλεσμα, ένας κόμβος μπορεί να θεωρηθεί λανθασμένα ως εγωιστής παρότι είναι συνεργατικός. Επιπλέον, σοβαρά θέματα εγείρονται σχετικά με την εκτίμηση της συμπεριφοράς ενός κόμβου σε συνδυασμό με τη θέση του. Για παράδειγμα, ορισμένα μέλη του δικτύου μπορεί να θεωρηθούν κακόβουλα, καθώς λόγω της γεωγραφικής τους θέσης ή μη ύπαρξης γειτόνων δεν έχουν την ευκαιρία να δρομολογήσουν πακέτα. Τέτοια μέλη θα μπορούσε να είναι οι απομακρυσμένοι κόμβοι, οι οποίοι δεν έχουν αρκετούς γείτονες για να προωθήσουν πακέτα και γενικότερα οι κόμβοι τους οποίους τα πρωτόκολλα δρομολόγησης δεν αναδεικνύουν ως επόμενο βήμα.

Σε αυτά τα θέματα, καλείται ο σχεδιαστής του μηχανισμού κινήτρων να επέμβει και να δημιουργήσει συμβάσεις που τα επιλύουν με τη μικρότερη δυνατή κατανάλωση πόρων.

1.7 ΣΥΓΚΡΙΣΗ ΤΩΝ ΜΗΧΑΝΙΣΜΩΝ ΦΗΜΗΣ ΚΑΙ ΠΙΣΤΩΣΗΣ

Παρακάτω, παρατίθεται ο πίνακας 1.1 όπου παρουσιάζονται συνοπτικά τα πλεονεκτήματα και τα μειονεκτήματα των 2 μηχανισμών.

	Πλεονεκτήματα	Μειονεκτήματα
Μηχανισμοί φήμης	<ul style="list-style-type: none"> – Αντικειμενική αξιολόγηση καθότι γίνεται συλλογή πληροφοριών από μεγάλο δείγμα – Ανθεκτικοί στη διάδοση ψευδών πληροφοριών από μία μικρή ομάδα κακόβουλων κόμβων 	<ul style="list-style-type: none"> – Κόστος σε πόρους του δικτύου – Ευπαθείς στη διάδοση ψευδών πληροφοριών από μία μεγάλη ομάδα κακόβουλων κόμβων – Ανάγκη για ταυτοποίηση κόμβων (ιδιαίτερα περίπλοκη διαδικασία)
Μηχανισμοί πίστωσης	<ul style="list-style-type: none"> – Ιδιαίτερα αποτελεσματικοί στην προώθηση συνεργατικής συμπεριφοράς – Χρήσιμοι σε multi-hop δίκτυα που η δράση και η αμοιβή δεν πραγματοποιούνται ταυτόχρονα 	<ul style="list-style-type: none"> – Κόστος σε πόρους του δικτύου – Αδυναμία εύρεσης κακόβουλων και εγωιστών κόμβων – Πολυπλοκότητα του αλγορίθμου χρέωσης κόμβων – Ανάγκη για διαμοιρασμό credits σε κόμβους που έχουν ανάγκη λόγω της γεωγραφικής τους θέσης – Ανάγκη για μηχανισμό ανακατανομής πλούτου

Πίνακας 1.1 Πίνακας πλεονεκτημάτων – μειονεκτημάτων των δύο κατηγοριών μηχανισμών κινήτρων

1.8 ΠΕΡΙΓΡΑΦΗ ΜΗΧΑΝΙΣΜΩΝ ΚΙΝΗΤΡΩΝ ΣΥΝΕΡΓΑΣΙΑΣ

Σε αυτό το σημείο θα γίνει μία σύντομη περιγραφή των κυριότερων μηχανισμών κινήτρων συνεργασίας, παραθέτοντας τη λογική λειτουργίας τους.

1.8.1 CONFIDANT

Ο μηχανισμός φήμης *CONFIDANT* (Cooperation of Nodes: Fairness in Dynamic Ad hoc Networks) [6] εμπίπτει στην υποκατηγορία *global reputation scheme* και κάνει χρήση της τεχνικής διαμοιρασμού των πινάκων φήμης με τους γειτονικούς κόμβους. Βασίζεται στην υπόθεση ότι τα πακέτα από κόμβους οι οποίοι είναι κακόβουλοι ή εγωιστές δεν δρομολογούνται από τους συνεργατικούς. Εάν ένας κόμβος έχει εσφαλμένα κατηγορηθεί ή έχει πάψει να είναι κακόβουλος, τότε είναι εφικτή η επανένταξή του στο δίκτυο.

Πιο συγκεκριμένα αποτελείται από 4 λειτουργικά στοιχεία: α) του ελέγχου (**monitor**), β) του διαχειριστή φήμης που παρατηρεί όλες τις εγγραφές για τη φήμη γειτονικών κόμβων καθώς και απομακρυσμένων κόμβων, γ) του διαχειριστή διαδρομών (**path manager**) που λαμβάνει τις αποφάσεις δρομολόγησης και αποφεύγει τους κακόβουλους κόμβους, δ) του διαχειριστή εμπιστοσύνης (**trust manager**).

Κάθε κόμβος μπορεί να ανιχνεύσει εγωιστική συμπεριφορά από τους γείτονές του είτε καταγράφοντας το αποτέλεσμα της μετάδοσης είτε παρατηρώντας παρεκτροπές από το πρωτόκολλο που ακολουθείται. Από τη στιγμή που θα εντοπίσει μία κακή συμπεριφορά τότε ο διαχειριστής φήμης παρεμβαίνει και εκπέμπονται προειδοποιητικά μηνύματα (ALARM), από το διαχειριστή εμπιστοσύνης, στους φιλικούς κόμβους. Αντίστοιχα, όταν λαμβάνεται ένας προειδοποιητικό μήνυμα τότε τσεκάρεται η αυθεντικότητά του πριν πυροδοτηθεί μία ενέργεια. Το σύστημα αυτό προϋποθέτει την ύπαρξη μίας σχέσης εμπιστοσύνης μεταξύ των κόμβων. Ο διαχειριστής φήμης είναι υπεύθυνος να τροποποιήσει τις εγγραφές του για τον κακόβουλο κόμβο όταν υπάρχουν σαφείς ενδείξεις εγωιστικής συμπεριφοράς ή όταν αυτή η συμπεριφορά επαναληφθεί πολλές φορές ξεπερνώντας ένα κατώφλι, οπότε και είναι έτοιμος να επιβάλλει τις κυρώσεις. Ο κόμβος διαμορφώνει έναν μέγεθος που αποκαλούμε βαθμολογία (rating) και λαμβάνει υπόψη τη δικιά του εκτίμηση για τον κακόβουλο, τις εκτιμήσεις των γειτονικών κόμβων και τις εκτιμήσεις των ακόμα πιο μακρινών. Εάν το μέγεθος αυτό έχει διογκωθεί τότε δρα ο διαχειριστής διαδρομών, αποκλείει τις εγγραφές δρομολόγησης που περιλαμβάνουν τον κακόβουλο κόμβο και τον απομονώνει.

1.8.2 CORE

Ο μηχανισμός φήμης *CORE* (Collaborative Reputation Mechanism) [7] είναι επίσης *global reputation scheme*. Για τον υπολογισμό του πίνακα φήμης χρησιμοποιούνται οι εμπειρίες του κόμβου και οι εμπειρίες που συνέλλεξε από τους γείτονές του σε συνδυασμό με μία ειδική συνάρτηση. Αυτή η συνάρτηση χρησιμοποιείται για την εκτίμηση της συμπεριφοράς των άλλων κόμβων. Εάν η παρατηρούμενη συμπεριφορά είναι διαφορετική του αποτελέσματος της συνάρτησης αυτής τότε η βαθμολογία του κόμβου αλλάζει. Κάθε κόμβος ελέγχει τις συμπεριφορές των γειτόνων του ανάλογα με την ειδική συνάρτηση και συλλέγει πληροφορίες για τις διαδοχικές εκτελέσεις της. Αυτές οι πληροφορίες αποθηκεύονται στον πίνακα φήμης.

Κάθε εγγραφή στον πίνακα φήμης διαθέτει 4 πεδία: το αναγνωριστικό (id) του κόμβου, την υποκειμενική φήμη (**subjective reputation**) που αφορά τη φήμη που έχει συλλέξει ο κόμβος για τους γείτονές του δίνοντας μεγαλύτερη βαρύτητα στα στοιχεία που έχει συλλέξει στο παρελθόν, την έμμεση φήμη (**indirect reputation**) που αφορά τη φήμη που ο κόμβος λαμβάνει από το υπόλοιπο δίκτυο και τη λειτουργική φήμη (**functional reputation**) που αποτελεί το συνδυασμό των 2 προηγούμενων για διαφορετικές όμως διαδικασίες.

1.8.3 SORI

Ο μηχανισμός *SORI* (Secure and Objective Reputation-based Incentive) [8] είναι ένα σύστημα φήμης που σε αντίθεση με προηγούμενους μηχανισμούς, η φήμη του κάθε κόμβου διαδίδεται μονάχα στους γείτονές του. Υποθέτουμε ότι κάθε κόμβος γνωρίζει για τις μεταδόσεις που κάνουν οι γείτονές του. Η φήμη προκύπτει βάσει αντικειμενικών μετρήσεων και η διάδοση αυτής γίνεται από μία διάταξη αυθεντικότητας, που βασίζεται σε μονομερή hash tables.

Η λογική λειτουργίας του μηχανισμού βασίζεται σε τρία βασικά βήματα: την παρακολούθηση της συμπεριφοράς και το διαμοιρασμό της πληροφορίας αυτής στους γείτονες, την εύρεση φήμης και την τιμωρία των κακόβουλων κόμβων.

1.8.4 OCEAN

Ο μηχανισμός *OCEAN* (*Observation-based Cooperation Enforcement in Ad hoc Networks*) [9] είναι ένας μηχανισμός φήμης και εισάγει ένα ενδιάμεσο στρώμα ανάμεσα στο στρώμα MAC και δικτύου κατά OSI. Σε αντίθεση με τους προαναφερθείς μηχανισμούς, βασίζεται αποκλειστικά στην παρακολούθηση των γειτονικών κόμβων, χωρίς να γίνεται κάποια μετάδοση πληροφορίας αναμεταξύ τους.

Η φήμη ενός κόμβου διαμορφώνεται ανάλογα με κάποια θετικά και αρνητικά συμβάντα και καταγράφονται μέσω της αντίδρασης ενός γείτονα να προωθήσει ένα πακέτο. Η βαθμολογία αυτή αρχικοποιείται σε μία θετική τιμή όταν πέσει κάτω από ένα κατώφλι τότε προστίθεται σε μία τοπική, για τον κόμβο, λίστα με όλους τους προβληματικούς κόμβους.

Στη συνέχεια, βαθμολογούμε τις διαδρομές. Οι διαδρομές χαρακτηρίζοντας ως καλές ή κακές με κριτήριο αν περιλαμβάνουν κάποιον κόμβο, που βρίσκεται στη λίστα. Κάθε κόμβος λαμβάνει ένα πακέτο και τσεκάρει αν το επόμενο βήμα είναι σε μία κακή διαδρομή. Εάν όχι, τότε δρομολογεί το πακέτο και απαντάει στον αποστολέα ενώ σε κάθε άλλη περίπτωση απορρίπτει το πακέτο.

1.8.5 SPRITE

Ο μηχανισμός *SPRITE* (*Simple Cheat Proof Credit Based System for Mobile Ad hoc Networks*) [10] ανήκει στους μηχανισμούς πίστωσης. Δεν απαιτείται επιπλέον hardware chips για να αποτραπεί η υποκλοπή πιστωτικών μονάδων (**credits**) αλλά ενσωματώνει μία κεντρική οντότητα που παίζει το ρόλο της τραπέζης (**credit clearance service** ή **CCS**). Κάθε κόμβος που λαμβάνει ένα πακέτο αποθηκεύει την ψηφιακή απόδειξη του συγκεκριμένου πακέτου (**signed receipt**), η οποία είχε δημιουργηθεί από τον αποστολέα. Ο *SPRITE* προϋποθέτει ότι ο κάθε κόμβος διαθέτει ένα πιστοποιητικό δημοσίου κλειδιού, το οποίο του έχει αποσταλεί από μία κεντρική οντότητα πιστοποίησης (**Certified Authority** ή **CA**). Όταν ένας κόμβος έχει σύνδεση με το *CCS* τότε αποστέλλει όλες τις ψηφιακές αποδείξεις όλων των πακέτων που έχει λάβει. Όταν ένας κόμβος στέλνει ένα πακέτο τότε χάνει credits, επειδή οι άλλοι κόμβοι καταναλώνουν πόρους για να το δρομολογήσουν. Προκειμένου ένας κόμβος να διαθέτει αρκετά credits για να στέλνει πακέτα πρέπει να δρομολογεί για λογαριασμό των υπολοίπων.

Το *CCS* χρεώνει τον αποστολέα ανάλογα με τον αριθμό των ψηφιακών αποδείξεων και τον αριθμό των ενδιάμεσων κόμβων που απαιτήθηκαν για την δρομολόγηση των πακέτων στον παραλήπτη, αν και εφόσον ο παραλήπτης έχει αποστείλει τις ψηφιακές αποδείξεις.

Οι ψηφιακές αποδείξεις, που αποτελούν ενδείξεις συνεργατικότητας και ορθής δρομολόγησης, μπορεί να προκαλέσουν σημαντικές ευπάθειες στο μηχανισμό. Οι κόμβοι που λαμβάνουν ένα πακέτο λαμβάνουν και την ψηφιακό του αποτύπωμα, το οποίο θα μπορούσαν να διαμοιραστούν με κόμβους που δεν παίρνουν μέρος στη δρομολόγηση και να διεκδικήσουν με το ίδιο ψηφιακό αποτύπωμα το μερίδιο των credits από το CCS. Οι σχεδιαστές του μηχανισμού έχουν λάβει υπόψη αυτήν την ευπάθεια κάνοντας την κεντρική οντότητα ανθεκτική σε τέτοιου είδους συγκρούσεις, παραβλέποντας τις διπλές αποδείξεις που τους καταφθάνουν. Επιπλέον, η επιβράβευση των συνεργατικών κόμβων θα πρέπει να πραγματοποιείται ανεξάρτητα του αν η αναμετάδοση του πακέτου ήταν ανεπιτυχής, καθώς έχει δαπανήσει πόρους γι' αυτή την πράξη.

1.8.6 Token-Based Cooperation Enforcement

Αυτός ο μηχανισμός πίστωσης [11] προστατεύει τόσο την δρομολόγηση όσο και την προώθηση των πακέτων. Δεν προϋποθέτει καμία προϋπάρχουσα εμπιστοσύνη μεταξύ των γειτόνων ούτε και την παρουσία μίας κεντρικής οντότητας. Απομονώνει τους κακόβουλους κόμβους και επιστρατεύει την κρυπτογραφία για να είναι ανθεκτικός απέναντί τους. Ο μηχανισμός πίστωσης προκαλεί σημαντικό αριθμό πλεοναζόντων μηνυμάτων συγχρονισμού (**overhead**), το οποίο όμως μειώνεται αισθητά όσο οι κόμβοι συνεργάζονται.

Ο μηχανισμός λειτουργεί ως εξής: το σύστημα κρυπτογράφησης δημιουργεί ένα ιδιωτικό κλειδί, το οποίο διαμοιράζεται μεταξύ των κόμβων του δικτύου, με τον κάθε ένας όμως να κατέχει ένα μέρος αυτού. Επίσης, κάθε κόμβος διαθέτει ένα τεκμήριο (**token**), το οποίο προσυπογράφεται με το ιδιωτικό κλειδί του συστήματος. Οι γείτονες του κόμβου μπορούν να επιβεβαιώσουν το τεκμήριο αυτό. Οι κόμβοι χωρίς ένα έγκυρο τεκμήριο απομονώνονται επειδή όλοι οι τίμιοι γείτονές του δε θα είναι πρόθυμοι να αλληλοεπιδράσουν μαζί τους. Αξίζει να σημειωθεί ότι το τεκμήριο αυτό έχει ημερομηνία λήξης (**expiration date**) και οι κόμβοι οφείλουν να το ανανεώσουν. Οι κόμβοι συνεργατικά παρακολουθούν τους υπόλοιπους και προσπαθούν να ανιχνεύσουν εγωιστικές συμπεριφορές. Από τη στιγμή που ανιχνευθεί ένας κακόβουλος κόμβος, το τεκμήριο του εγωιστικού ανακαλείται από τους γείτονές του και μπλοκάρεται από τη σύνδεση στο δίκτυο. Αυτός ο μηχανισμός επιτρέπει στους συνεργατικούς κόμβους να συλλέγει credits και να ανανεώνει το τεκμήριό του με μικρότερη συχνότητα.

1.8.7 Ad hoc-VCG

Η ενεργειακή απόδοση είναι μία παράμετρος υψηλής σημασίας στα Ad hoc δίκτυα. Ένας κόμβος μπορεί να αρνηθεί να δρομολογήσει ένα πακέτο, εάν συνειδητοποιήσει ότι αυτό θα του κοστίσει σε επεξεργαστική ισχύ. Έτσι ο Ad hoc-VCG [12] μηχανισμό πίστωσης εκτιμά το κόστος ενέργειας c_i , το οποίο εκτιμάται για παράδειγμα σε ευρώ ανά watt και είναι ξεχωριστός για κάθε κόμβο. Εάν η προώθηση ενός πακέτου απαιτεί την εκπομπή ενός σήματος ίσου με p^{emit} watts, τότε ο κόμβος απαιτεί σαν πληρωμή $c_i \cdot p^{emit}$ ευρώ. Από την άλλη μεριά, εάν ο κόμβος δε λάβει τη πληρωμή που είναι επαρκής για να καλύψει το κόστος προώθησης, τότε αρνείται να το προωθήσει.

Από αυτήν την οπτική γωνία, η συμμετοχή στο δίκτυο είναι εθελοντική, με την παράμετρο c_i να εξαρτάται από το χρόνο, λαμβάνοντας υπόψη τις προτιμήσεις του κόμβου καθώς και την αυτονομία ισχύος που διαθέτει. Οι κόμβοι αποφασίζουν το επίπεδο ενεργειακής ισχύς μετάδοσης στους γείτονές τους μέσω μίας διαδικασίας σηματοδότησης και πρόσθετων πεδίων ελέγχου στα πακέτα. Αποστέλλουν ένα πακέτο με μεγάλη ισχύ για να δηλώσουν το επίπεδο ισχύος στην επικεφαλίδα (**header**) και λαμβάνουν ένα πακέτο με το επίπεδο ισχύος που κατέφθασε στο γείτονα. Οι κόμβοι, τέλος, ανταλλάσσουν το κόστος ενέργειας c_i με τους γείτονές τους.

1.9 ΥΒΡΙΔΙΚΟΙ ΜΗΧΑΝΙΣΜΟΙ ΚΙΝΗΤΡΩΝ

Όπως αναδείχθηκε παραπάνω οι μηχανισμοί φήμης και πίστωσης μπορούν να δώσουν κίνητρο στους κόμβους τους δικτύου να συνεργαστούν.

Σε ένα σύστημα φήμης, ωστόσο, οι έξυπνοι κακόβουλοι κόμβοι μπορούν εύκολα να διατηρήσουν τη φήμη τους πάνω από το κατώφλι προωθώντας πακέτα επιλεκτικά. Αυτοί οι κόμβοι μπορούν να επωφεληθούν των πόρων του δικτύου, εις βάρος των υπολοίπων δίχως να θεωρούνται εγωιστές. Αντίστοιχα σε συστήματα πίστωσης, παρότι παροτρύνουν τους κόμβους να είναι συνεργατικοί, αδυνατούν να παρέχουν έναν τρόπο να γνωρίζουν την ποιότητα υπηρεσίας που παρέχει ο καθένας τους. Επιπλέον, αποτυγχάνουν να τιμωρήσουν έναν πλούσιο αλλά εγωιστή κόμβο, ο οποίος έχοντας κερδίσει πολλές πιστωτικές μονάδες συνεργαζόμενος με το υπόλοιπο δίκτυο συμπεριφέρεται εγωιστικά από ένα σημείο και μετά.

Παρακάτω θα μελετηθούν οι πίνακες ανταπόδοσης (**pay off matrix**) σε συστήματα μηχανισμών φήμης και πίστωσης με βάση το μοντέλο της θεωρίας παιγνίων.

Σε μία αλληλεπίδραση μεταξύ δύο κόμβων, ένας κόμβος προωθεί ένα πακέτο για να δρομολογηθεί από έναν άλλο κόμβο. Θεωρούμε ότι το κόστος για να προωθήσεις ένα πακέτο είναι c και η ανταμοιβή για την προώθηση είναι p .

Ορισμός I: Ένα σύστημα λέμε ότι έχει έρθει σε ισορροπία κατά Nash (**Nash Equilibrium** ή **NE**) όταν η στρατηγική που θα επιλέξει κάθε κόμβος του δικτύου είναι η βέλτιστη γι' αυτόν ανεξαρτήτου του τι θα επιλέξουν οι υπόλοιποι κόμβοι.

Ορισμός II: Ένα αποτέλεσμα δεν είναι βέλτιστο κατά Pareto (**non-Pareto-optimal**) εάν υπάρχει άλλο αποτέλεσμα που θα μπορούσε να δώσει και στους 2 κόμβους του συστήματος υψηλότερη ανταμοιβή ή θα μπορούσε να δώσει στον έναν από τους 2 την ίδια ανταμοιβή αλλά στον άλλον υψηλότερη. Ένα αποτέλεσμα είναι Pareto-optimal αν δεν υπάρχουν άλλα τέτοια αποτελέσματα.

1.9.1 Θεωρία παιγνίων σε συστήματα φήμης

Στον πίνακα 1.2 φαίνονται η ανταπόδοση του κάθε κόμβου ανάλογα με τη στρατηγική που θα ακολουθήσει (**C** για **Cooperate** και **I** για **Incorporate**) [13].

		Node j	
		Cooperative	Non – cooperative
Node i	Cooperative	$(p - c, p - c)$	$P(C_i, I_j)$
	Non – cooperative	$P(I_i, C_j)$	$(0, 0)$

Πίνακας 1.2 Pay-off Matrix για συστήματα φήμης

$$\text{Όπου } P(C_i, I_j) = \begin{cases} (-c, p), & \text{αν } R_{I(j)} > R_t \\ (0, 0), & \text{αν } R_{I(j)} \leq R_t \end{cases} \quad (1) \quad \text{και} \quad P(I_i, C_j) = \begin{cases} (p, -c), & \text{αν } R_{I(i)} > R_t \\ (0, 0), & \text{αν } R_{I(i)} \leq R_t \end{cases} \quad (2)$$

Όταν και οι δύο κόμβοι συνεργάζονται τότε το κοινό τους όφελος είναι $p - c$. Όταν και οι δύο κόμβοι δεν συνεργάζονται τότε το όφελος και των δύο είναι 0, γιατί δεν έχουν κανένα κόστος και κανένα όφελος. Εάν ο κόμβος j είναι μη συνεργάσιμος αλλά η φήμη του κόμβου είναι πάνω από το κατώφλι R_t τότε ο μη-συνεργάσιμος j έχει όφελος p ενώ ο συνεργάσιμος i έχει κόστος $-c$. Μπορούμε να δούμε από τον παραπάνω πίνακα ότι εάν η φήμη των κόμβων είναι πάνω από το κατώφλι, τότε το σεν (I_i, I_j) που προσφέρει όφελος $(0, 0)$ είναι **NE** αλλά το σεν (C_i, C_j) είναι βέλτιστο κατά Pareto. Μόνο όταν η φήμη του ενός κόμβου είναι κάτω από το κατώφλι, τότε το σεν (C_i, C_j) κυριαρχεί του (I_i, I_j) , που σημαίνει ότι είναι και **NE** και βέλτιστο κατά Pareto.

Αυτό σημαίνει ότι τα συστήματα φήμης δεν μπορούν να παρέχουν κίνητρα για να ενθαρρύνουν τους κόμβους να είναι πιο συνεργατικοί. Απλώς τους προτρέπουν να μην είναι κακόβουλοι. Ένας έξυπνος κόμβος μπορεί εύκολα να εκμεταλλευτεί αυτήν την πολιτική προωθώντας ένα ποσοστό των πακέτων που καταφθάνουν κρατώντας τη φήμη του οριακά πάνω από ένα επίπεδο, υποβαθμίζοντας ωστόσο την ποιότητα των υπηρεσιών.

Επομένως τα συστήματα φήμης χρειάζονται συμπληρωματικές μεθόδους να παροτρύνουν τους κόμβους να είναι όσο πιο συνεργατικοί μπορούν και να ανταμείβουν ο ένας τον άλλον ανάλογα με το επίπεδο φήμης που κατέχουν.

1.9.2 Θεωρία παιγνίων σε συστήματα πίστωσης

Στα συστήματα πίστωσης εισάγουμε δύο νέα μεγέθη: το m_f που δηλώνει την αμοιβή για την προώθηση ενός πακέτου και το m_p το κόστος για την αποστολή ενός πακέτου [13]. Στον πίνακα 1.3 φαίνεται η ανταπόδοση του κάθε κόμβου ανάλογα με τη στρατηγική που θα ακολουθήσει.

		Node j	
		Cooperative	Non – cooperative
Node i	Cooperative	$(p - c - \Delta m, p - c - \Delta m)$	$P(C_i, I_j)$
	Non – cooperative	$P(I_i, C_j)$	$(0,0)$

Πίνακας 1.3 Pay-off Matrix για συστήματα πίστωσης

$$\text{Όπου } P(C_i, I_j) = \begin{cases} (-c + m_f, p - m_p), & \text{αν } V_j > 0 \\ (0,0), & \text{αν } V_j < 0 \end{cases} \quad (3)$$

$$\text{και } P(I_i, C_j) = \begin{cases} (p - m_p, -c + m_f), & \text{αν } V_i > 0 \\ (0,0), & \text{αν } V_i < 0 \end{cases} \quad (4)$$

Συγκεκριμένα, μεταξύ δύο κόμβων με τακτικές (C_i, I_j) , παρότι ο μη συνεργατικός κόμβος μπορεί να εξοικονομήσει το κόστος c , θα πρέπει όμως υποχρεωτικά να πληρώσει το κόστος m_p . Παράλληλα, ο συνεργατικός κόμβος θα χάσει από το κόστος p θα κερδίσει όμως το κόστος από την συνεργατική του συμπεριφορά m_c . Για την (C_i, C_j) στρατηγική, αφού και οι δύο κόμβοι συνεργάζονται, θα κερδίσουν p και θα χάσουν c για τη μετάδοση του πακέτου. Επίσης, θα πληρώσουν m_p για τη προώθηση του πακέτου αλλά θα κερδίσουν m_f για τη καλή τους συμπεριφορά. Γι' αυτό και το τελικό όφελος γι' αυτή τη στρατηγική θα είναι $(p - c - \Delta m, p - c - \Delta m)$ ορίζοντας ως $\Delta m = m_p - m_f$.

Μπορούμε να παρατηρήσουμε ότι η στρατηγική (C_i, C_j) είναι βέλτιστη κατά Pareto, αν και μόνο αν $p > m_p$ και $m_f > c$ [14]. Επίσης, προκύπτει ότι εάν ένας πλούσιος κακόβουλος κόμβος διατηρεί ένα πόσο πιστωτικών μονάδων μεγαλύτερο πάνω από το μηδέν, το κατώτατο όριο απόρριψης πακέτου είναι $\frac{qm_f + V - sm_p}{qm_f}$ όπου το q είναι ο αριθμός των πακέτων που ο κόμβος χρειάζεται να αποστείλει στις πρώτες q αλληλεπιδράσεις με το δίκτυο, s είναι ο αριθμός των μηνυμάτων που στάλθηκαν από έναν εγωιστή κόμβο και V είναι το συνολικό ποσό πιστωτικών μονάδων. Η παραπάνω συνθήκη δίνει 3 περιπτώσεις με εξαιρετικά υψηλό ρυθμό απόρριψης πακέτου στα συστήματα πίστωσης.

- 1) Εάν ένας κακόβουλος κόμβος έχει ένα αισθητό αριθμό credit V , τότε μπορεί να αρνηθεί να συνεργάζεται για ένα μεγάλο χρονικό διάστημα.
- 2) Εάν ένας κόμβος δεν έχει ανάγκη να στέλνει μηνύματα ($s = 0$), τότε επίσης θα απορρίπτει τα πακέτα των υπολοίπων.
- 3) Όταν $m_p \ll m_f$ που σημαίνει πολύ μικρότερο κόστος δημιουργίας και αποστολής πακέτων από την ανταμοιβή δρομολόγησης, το σύστημα οδηγείται σε κατάσταση υψηλού ρυθμού απόρριψης πακέτων.

Σύμφωνα με την παραπάνω ανάλυση, ένα σύστημα πίστωσης μπορεί να παρέχει κίνητρα για υψηλής συνεργατικότητας. Ωστόσο, αποτυγχάνει στην ανίχνευση κακής συμπεριφοράς από μερικούς έξυπνους κόμβους.

1.9.3 Θεωρία παιγνίων σε υβριδικούς μηχανισμούς

Αφού ξεχωριστά κάθε σύστημα κινήτρων δεν μπορεί να παρέχει έναν αυτόνομο αποδοτικό μηχανισμό για την μείωση των κακόβουλων κόμβων, δημιουργήθηκε η ανάγκη για μία νέα προσέγγιση, έναν υβριδικό μηχανισμό, ο οποίος θα μπορεί να συνδυάσει τα πλεονεκτήματα των άλλων δύο προκειμένου να απομονώσει τα μειονεκτήματα του καθενός. Στον υβριδικό μηχανισμό (**hybrid mechanism**), το κόστος που απαιτείται για την αποστολή ενός πακέτου εξαρτάται από την τιμή της φήμης του. Ένας κόμβος με υψηλή φήμη θα πληρώνει λιγότερα, ενώ ένας κόμβος με χαμηλή φήμη θα πληρώνει υψηλότερο τίμημα για την ίδια υπηρεσία. Αυτό αποθαρρύνει τους κόμβους με υψηλή φήμη να πάψουν να συνεργάζονται. Εντωμεταξύ, η φήμη του κόμβου εξακολουθεί να είναι το μέγεθος που θα διαχωρίζει τους συνεργατικούς από τους εγωιστές και θα βασίζεται σε ένα κατώφλι που θα προτρέπει τους πλούσιους κόμβους ή τους κόμβους που δεν επιθυμούν να αποστείλουν πακέτο να συνεργάζονται. Εάν η φήμη ενός κόμβου είναι κάτω από το κατώφλι, τότε ανεξαρτήτως του πλούτου που διαθέτει θα απορρίπτεται κάθε αίτημα αποστολής πακέτου και παράλληλα ο πλούτος θα μειωθεί ραγδαίως.

Γι' αυτό το λόγο, ο υβριδικός μηχανισμός μπορεί αποτελεσματικά να αποτρέψει εγωιστές κόμβους από τη διατήρηση της τιμής της φήμης τους μόλις πάνω από το κατώφλι.

		Node j	
		Cooperative	Non – cooperative
Node i	Cooperative	$P(C_i, C_j)$	$P(C_i, I_j)$
	Non – cooperative	$P(I_i, C_j)$	(0,0)

Πίνακας 1.4 Pay-off Matrix για υβριδικά συστήματα

$$\text{Όπου } P(C_i, C_j) = \left(p - c + \left(m_f - \frac{m_p}{R_i} \right), p - c + \left(m_f - \frac{m_p}{R_i} \right) \right). \quad (5)$$

$$P(C_i, I_j) = \begin{cases} \left(-c + m_f, \frac{p - m_p}{R_j}\right), & \text{αν } V_j > 0 \text{ και } R_{I(j)} > R_t \\ (0,0), & \text{αν } V_j \leq 0 \text{ ή } R_{I(j)} \leq R_t \end{cases} \quad (6)$$

$$\text{Και } P(I_i, C_j) = \begin{cases} \left(\frac{p - m_p}{R_j}, -c + m_f\right), & \text{αν } V_i > 0 \text{ και } R_{I(i)} > R_t \\ (0,0), & \text{αν } V_i \leq 0 \text{ ή } R_{I(i)} \leq R_t \end{cases} \quad (7)$$

Η στρατηγική (C_i, C_j) σε έναν υβριδικό μηχανισμό είναι NE και βέλτιστος κατά Pareto εάν το κόστος μετάδοσης c , η τρέχουσα τιμή της φήμης R_i και R_j και η τιμή m_f ικανοποιούν τη σχέση $m_f > c$ & $p > c$ [14]. Επιπλέον, μία υψηλή τιμή της φήμης, μπορεί να επιφέρει μεγαλύτερο κέρδος για τη συνεργατική συμπεριφορά και να προσφέρει υψηλότερο κίνητρο σε σχέση με τους μηχανισμούς πίστωσης.

1.9.4 HEAD

Ο υβριδικός μηχανισμός HEAD [15] (*A Hybrid Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks*) αποτελεί μία επέκταση του μηχανισμού φήμης OCEAN.

Ο μηχανισμός ταξινομεί τους μη-συνεργατικούς κόμβους σε 3 κατηγορίες: α) τους κακόβουλους που περιλαμβάνει τους κόμβους που σκόπιμα προσπαθούν να επιτεθούν στο σύστημα, β) τους παραπλανητικούς, στους οποίους οι κόμβοι λαμβάνουν μέρος στη διαδικασία εύρεσης διαδρομών αλλά αποτυγχάνουν στη δρομολόγηση των πακέτων και γ) τους εγωιστές, οι οποίοι προσπαθούν να διατηρήσουν τους πόρους τους και αρνούνται να προωθήσουν τα πακέτα των υπολοίπων ενώ αναμένουν πως οι άλλοι θα το πράξουν.

Κάθε κόμβος διατηρεί 2 ειδών λίστες: τη λίστα με τους ελαττωματικούς κόμβους στην οποία εγγράφονται οι κακόβουλοι ή παραπλανητικοί κόμβοι και τη λίστα με τους εγωιστές κόμβους που εγγράφονται οι εγωιστές. Πριν αποφασιστεί εάν θα προωθήσει ένα πακέτο, παρατηρεί τις λίστες και εάν ο γείτονας εμπίπτει σε τουλάχιστον μία από αυτές τότε το απορρίπτει.

Επιπλέον, κάθε κόμβος διατηρεί έναν πίνακα φήμης με τα εξής πεδία: την ταυτότητα (ID) κάθε γείτονα που παραμένει αμετάβλητη καθ' όλη την αλληλεπίδραση, τη φήμη που είναι μία αριθμητική τιμή που αυξάνεται ή μειώνεται ανάλογα με τα συμβάντα που προκύπτουν και στην περίπτωση που πέσει κάτω από ένα κατώφλι τότε προστίθεται ο γείτονας στη λίστα με τους ελαττωματικούς, έναν μετρητή που μπορεί να θεωρηθεί ένας λογαριασμός του γείτονα σε μία τράπεζα που φιλοξενείται στον τρέχων κόμβο, μία λίστα αποφυγής (**avoid-list**) που περιέχει

κόμβους που θεωρούνται από το γείτονα ως κακόβουλοι ή παραπλανητικοί και μία τελευταία λίστα (**nonchip-list**) που περιέχει όσους θεωρούνται εγωιστές σύμφωνα με το γείτονα.

Κάθε κόμβος εκπέμπει ένα μήνυμα με τη λίστα με τους ελαττωματικούς και κακόβουλους κόμβους. Κάθε φορά που ένας γείτονας λαμβάνει αυτό το μήνυμα αντικαθιστά την *avoid-list* και *nonchip-list* του αντίστοιχου κόμβου στο πίνακα φήμης με τις λίστες από το μήνυμα που έλαβε, χωρίς να το επανεκπέμπει. Επιπλέον, για να απαλείψει τη πιθανότητα λανθασμένης εκτίμησης κόμβου ως μη-συνεργατικό, κάθε ένα ορισμένο χρονικό διάστημα διαγράφει τους κόμβους από την ελαττωματική λίστα χωρίς όμως να τους αυξάνει τη αξιολόγηση. Με αυτόν τον τρόπο, τους δίνει τη δυνατότητα να επανορθώσουν ενώ στην περίπτωση που δεν το πράξουν επανεγγράφονται άμεσα στην ελαττωματική λίστα.

1.9.4.1 Διαδικασία ανακάλυψης διαδρομής

Παρακάτω παρουσιάζονται δύο μηχανισμοί σύμφωνα με τους οποίους ο HEAD πραγματοποιεί τη διαδικασία της ανακάλυψης των διαδρομών:

Ο πρώτος αφορά τη διαδικασία αίτησης ανακάλυψης.

Κόμβος ελέγχει τα πεδία του πακέτου αίτησης (RREQ)
 Εάν ο αποστολέας ανήκει στη λίστα με τους ελαττωματικούς
 Κόμβος απορρίπτει το πακέτο
 Εάν ο αποστολέας ανήκει στη λίστα με τους εγωιστές
 Κόμβος απορρίπτει το πακέτο
 Αλλιώς
 Κόμβος επαναμεταδίδει το πακέτο

Ο δεύτερος αφορά τη διαδικασία της απάντησης στη διαδικασία αίτησης.

Κόμβος ελέγχει τα πεδία του πακέτου απάντησης (RREP)
 Εάν ο προηγούμενος ή επόμενος κόμβος της διαδρομής βρίσκεται στη λίστα με τους ελαττωματικούς
 Κόμβος απορρίπτει το πακέτο
 Εάν ο προηγούμενος ή επόμενος κόμβος της διαδρομής βρίσκεται στη λίστα με τους εγωιστές
 Κόμβος απορρίπτει το πακέτο
 Αλλιώς
 Κόμβος προωθεί το πακέτο στο επόμενο βήμα

1.9.4.2 Διαδικασία προώθησης πακέτου

Κόμβος αφού λάβει το πακέτο δεδομένων, κόμβος ελέγχει τα πεδία του πακέτου δεδομένων

Εάν ο προηγούμενος ή επόμενος κόμβος της διαδρομής βρίσκεται στη λίστα με τους ελαττωματικούς Κόμβος απορρίπτει το πακέτο

Εάν ο προηγούμενος ή επόμενος κόμβος της διαδρομής βρίσκεται στη λίστα με τους εγωιστές Κόμβος απορρίπτει το πακέτο

Ελαττώνει κατά 1 το λογαριασμό του προηγούμενου κόμβου στον πίνακα φήμης

Αυξάνει κατά 1 το λογαριασμό του επόμενου κόμβου στον πίνακα φήμης

Αποθηκεύει το πακέτο στη μνήμη και ξεκινάει ένα χρονόμετρο

Εάν ο δεύτερος επόμενος κόμβος της διαδρομής ανήκει στη λίστα ελαττωματικών ή εγωιστών του επόμενου κόμβου

Ακυρώνει το χρονόμετρο και σταματάει τη διαδικασία

Εάν ανιχνεύσει τον επόμενο κόμβο να προωθήσει το πακέτο πριν τη λήξη του χρονομέτρου

Αυξάνει τη φήμη του επόμενου κόμβου

Αλλιώς

Μειώνει τη φήμη του επόμενου κόμβου

1.9.4.3 Ανάλυση της απόδοσης του μηχανισμού

Ο μηχανισμός HEAD καταφέρνει να είναι ιδιαίτερα αποτελεσματικός στην αντιμετώπιση των κακόβουλων κόμβων και στην ώθηση των εγωιστών σε συνεργατική συμπεριφορά. Ωστόσο, δημιουργείται σημαντική κίνηση πακέτων σηματοδοσίας λόγω των πλεοναζόντων μηνυμάτων για τη διάδοση της φήμης των κόμβων. Τέλος, απαιτείται περισσότερη τοπική μνήμη, αφού κάθε κόμβος διατηρεί εγγραφές για κάθε γείτονά του. Κάθε εγγραφή έχει επιπλέον δύο λίστες (*avoid-list* & *nonchip-list*). Δηλώνοντας ως B το μέσο όρο γειτόνων κάθε κόμβου στο δίκτυο το μέγεθος της κάθε εγγραφής είναι $3 + 2B$ ενώ το συνολικό μέγεθος του πίνακα είναι $B(3 + 2B)$.

Στη συνέχεια της διπλωματικής θα παρουσιαστεί αναλυτικά ο μηχανισμός φήμης DARWIN και ο προτεινόμενος υβριδικός μηχανισμός ICARUS.

2 ΜΗΧΑΝΙΣΜΟΣ DARWIN

Στο παρών κεφάλαιο θα παρουσιαστεί ο μηχανισμός φήμης Darwin, επέκταση του οποίου αποτελεί ο προτεινόμενος υβριδικός μηχανισμός.

2.1 ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

Αρχικά θεωρούμε ότι στο δίκτυο οι μη-συνεργατικοί κόμβοι αφορούν αποκλειστικά εγωιστές και όχι κακόβουλους κόμβους. Οι κόμβοι εξετάζονται σε ζευγάρια, ως παίγνιο δύο παικτών όπου ο ένας συμβολίζεται με i ενώ ο δεύτερος με $-i$. Το ζευγάρι αυτό των κόμβων επαναλαμβάνουν ένα παίγνιο όπου ο ένας στέλνει πακέτα στον άλλον και αποφασίζουν αν θα τα προωθήσουν ή όχι. Ο κάθε κόμβος αμείβεται για τη προσφορά του κατά a ενώ συμφωνούμε ότι το κανονικοποιημένο κόστος δρομολόγησης ορίζεται στην τιμή 1.

Στον πίνακα 2.1 εμφανίζεται ο πίνακας ανταπόδοσης για το συγκεκριμένο παίγνιο.

		<i>Node - i</i>	
		<i>Cooperative</i>	<i>Non - cooperative</i>
<i>Node i</i>	<i>Cooperative</i>	$(a - 1, a - 1)$	$(-1, a)$
	<i>Non - cooperative</i>	$(a, -1)$	$(0, 0)$

Πίνακας 2.1

Χωρίς απώλεια της γενικότητας, ο πίνακας 2.1 μπορεί να κανονικοποιηθεί σύμφωνα με τον τύπο $v_n = \frac{v+a}{2a-1}$ όπου v_n η κανονικοποιημένη τιμή και v η αντίστοιχη τιμή του πίνακα. Ο πίνακας 2.2 αποτελεί τον αντίστοιχο κανονικοποιημένο.

		<i>Node - i</i>	
		<i>Cooperative</i>	<i>Non - cooperative</i>
<i>Node i</i>	<i>Cooperative</i>	$(1, 1)$	$\left(-\frac{1}{2a-1}, \frac{2a}{2a-1}\right)$
	<i>Non - cooperative</i>	$\left(\frac{2a}{2a-1}, -\frac{1}{2a-1}\right)$	$(0, 0)$

Πίνακας 2.2

Εξαιτίας των συγκρούσεων, δεν είναι πάντα εφικτό να αναγνωριστεί εάν ένας κόμβος προώθησε ένα πακέτο ή όχι. Ορίζουμε το μέγεθος $p_e \in (0, 1)$ ως τη πιθανότητα ένα πακέτο να έχει προωθηθεί χωρίς να το γνωρίζει ο προηγούμενος κόμβος και η οποία θα είναι κοινή και για τους δύο. Παρατηρώντας το δίαυλο επικοινωνίας, ο κόμβος i υπολογίζει την αντιλαμβανόμενη

πιθανότητα απόρριψης του άλλου κόμβου $\hat{p}_{-i}^{(k)}$ τη χρονική στιγμή $k \geq 0$. Εδώ, αξίζει να σημειωθεί ότι ένα πακέτο θεωρείται ότι έχει απορριφθεί εάν είτε ο κόμβος $-i$ το απέρριψε είτε ο κόμβος i δεν αναγνώρισε τη προώθηση. Επομένως, μπορεί να εξαχθεί η εξής σχέση:

$\hat{p}_{-i}^{(k)} = p_{-i}^{(k)} + (1 - p_{-i}^{(k)})p_e = p_e + (1 - p_e)p_{-i}^{(k)}$ όπου $p_{-i}^{(k)}$ είναι η πιθανότητα ο κόμβος $-i$ να απορρίψει το πακέτο. Έτσι χρησιμοποιώντας το πίνακα 5, η μέση τιμή της ανταπόδοσης τη χρονική στιγμή k είναι:

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

Και αναδιατάσσοντας τους όρους προκύπτει:

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

Αφού ο κόμβος i δε γνωρίζει τη $p_{-i}^{(k)}$ δεν μπορεί να γνωρίζει το συνολικό του όφελος. Ωστόσο, χρησιμοποιώντας το τύπο αυτό μπορούμε να γνωρίσουμε εάν ο τρέχων κόμβος μπορεί να επωφεληθεί ακολουθώντας μία διαφορετική στρατηγική.

2.2 ΣΤΡΑΤΗΓΙΚΕΣ ΚΙΝΗΤΡΩΝ

Στη συνέχεια, θα αναπτυχθούν διάφορες στρατηγικές που έχουν προταθεί προκειμένου να δώσουν κίνητρα στους παίκτες του παιχνίσιου.

2.2.1 Tit For Tat

Μία εναλλακτική είναι η τεχνική Tit For Tat [16], η οποία είχε προτυποποιηθεί για τα ασύρματα δίκτυα [17] ως εξής:

$$\begin{aligned} \tilde{p}_i^{(0)}{}_{TFT} &= 0 \\ \tilde{p}_i^{(k)}{}_{TFT} &= \hat{p}_{-i}^{(k-1)} \text{ για } k \geq 1 \end{aligned}$$

που πρακτικά σημαίνει ότι ο κόμβος i ακολουθεί την τακτική που ο κόμβος $-i$ ακολούθησε την προηγούμενη χρονική στιγμή αλλά αποδείχτηκε ότι δε προσφέρει επαρκή κίνητρα στα ασύρματα δίκτυα.

2.2.2 Generous Tit for Tat

Το πρόβλημα με το TFT είναι ότι δε λαμβάνει υπόψη το γεγονός ότι δεν είναι πάντα εύκολο να αποφανθούμε εάν ένα πακέτο δρομολογήθηκε ή μη λόγω συγκρούσεων. Ένας τρόπος να απαλλαγούμε από αυτό το πρόβλημα είναι η χρήση ενός παράγοντα g που μπορεί να αποκαταστήσει τη συνεργασία μεταξύ των κόμβων. Αυτή η στρατηγική είναι γνωστή ως Generous TFT (GTFT) [18] και στα ασύρματα δίκτυα [17] έχει προτυποποιηθεί ως εξής:

$$\tilde{p}_i^{(0)}_{GTFT} = 0$$

$$\tilde{p}_i^{(k)}_{GTFT} = \max\{\hat{p}_{-i}^{(k-1)} - g, 0\} \text{ για } k \geq 1$$

Στο άρθρο [16] αποδεικνύεται ότι όταν δύο κόμβοι χρησιμοποιούν τη στρατηγική GTFT, τότε η συνεργασία μπορεί να επιτευχθεί μονάχα όταν $g = p_e$, μέγεθος το οποίο είναι αδύνατο να υπολογιστεί με ακρίβεια.

2.3 ΜΗΧΑΝΙΣΜΟΣ DARWIN

Στόχος είναι η ανάπτυξη ενός μηχανισμού που δε θα εξαρτάται από τον ακριβή υπολογισμό του μεγέθους p_e . Ο μηχανισμός DARWIN [19] βασίζεται σε μία διαφορετική στρατηγική κινήτρων γνωστή ως Contrite Tit For Tat (CTFT): ένας κόμβος που έκανε το λάθος και παρέκκλινε από τη συνεργατικότητα, θα μπορούσε να δηλώσει μετάνοια και να προσπαθήσει να διορθώσει το λάθος του αντί να τεθεί στο περιθώριο.

Η συγκεκριμένη στρατηγική βασίζεται στην καλή διαγωγή των κόμβων, που ορίζεται ως εξής: ένας κόμβος θεωρείται ότι έχει καλή διαγωγή στην αρχή του παιχνιδιού και συνεχίζει να έχει καλή διαγωγή όσο η στρατηγική CTFT το υποδεικνύει. Εάν ένας παίχτης έχει κακή διαγωγή μπορεί να επανέλθει συνεργαζόμενος με τον άλλο παίχτη. Τότε το CTFT ορίζει ότι ένας κόμβος θα πρέπει να συνεργάζεται εάν ο ίδιος έχει κακή διαγωγή είτε εάν ο άλλος κόμβος έχει καλή διαγωγή, διαφορετικά δρα εγωιστικά. Μαθηματικά αποτυπώνεται με τις ακόλουθες σχέσεις:

$$\tilde{p}_i^{(k)}_{DARWIN} = \left[\gamma \cdot (q_{-i}^{(k-1)} - q_i^{(k-1)}) \right]_0^1$$

$$q_i^{(k)} = \begin{cases} \left[\hat{p}_i^{(k)} - \tilde{p}_i^{(k)}_{DARWIN} \right]_0^1 & \text{για } k \geq 0 \\ 0 & \text{για } k = -1 \end{cases}$$

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

Το \hat{p}_i^k είναι η αντιλαμβανόμενη πιθανότητα απόρριψης και το \tilde{p}_i^k είναι η πιθανότητα απόρριψης DARWIN. Αν ισχύει $\hat{p}_i^k > \tilde{p}_i^k$ τότε ο κόμβος αντιλαμβάνεται ότι χάνονται περισσότερα πακέτα από αυτά που έπρεπε λόγω DARWIN. Το q_i^k μετράει αυτή τη λιποταξία ή αλλιώς δείχνει τη κατάσταση του κάθε κόμβου. Οι κόμβοι που είναι σε καλύτερη κατάσταση από αυτούς που συνεργάζονται τιμωρούν τον άλλο κόμβο με τη διαφορά της μεταξύ τους κατάστασης. Οι αντιλαμβανόμενες πιθανότητες απόρριψης των κόμβων ανταλλάσσονται μεταξύ των κόμβων. Έχουμε κάνει την παραδοχή ότι η πλειοψηφία των κόμβων δεν κλέβουν. Γι' αυτό βρίσκουμε την αντιλαμβανόμενη πιθανότητα ως το μέσο όσο των λαμβανόμενων πιθανοτήτων απόρριψης και έτσι τα ψέματα έχουν μικρή επίδραση.

Καταρχήν, παραδεχόμαστε ότι ισχύει $1 < \gamma < \frac{1}{p_e}$. Αποδεικνύεται ότι ο DARWIN είναι μία εύρωστη στρατηγική και ότι οι κόμβοι δεν έχουν λόγο να λιποτακτήσουν από την αμοιβαία συνεργασία σε μία μεγάλης διάρκειας αλληλεπίδραση. Για να ισχύει αυτό πρέπει το γ να παίρνει τη μικρότερη τιμή του συναρτήσει των α και p_e . Το να εκτιμήσουμε το α είναι αδύνατο άρα η ζητούμενη τιμή του γ μπορεί να προσεγγισθεί από τη σχέση $\gamma = \frac{1+p_e}{2 \cdot p_e}$. Όπως αποδεικνύεται διαφορετικές τιμές του α δεν έχουν μεγάλη επίδραση στην αποτελεσματικότητα του DARWIN.

Αποδεικνύεται με χρήση του ορισμού του DARWIN και της σχέσης $\hat{p}_i^k = p_e + (1 - p_e) \cdot p_i^{(k)}$ ότι αν δύο κόμβοι χρησιμοποιούν το DARWIN τότε η αμοιβαία συνεργασία επιτυγχάνεται στο μονοπάτι του σημείου ισορροπίας και έτσι ισχύει $p_i^{(k)} = p_{-i}^{(k)} = 0$ για κάθε $k \geq 0$. Αφού το DARWIN είναι καλύτερο από τις υπόλοιπες στρατηγικές επιτυγχάνει και καλύτερη ανταμοιβή για τους κόμβους.

2.3.1 Εφαρμογή του αλγορίθμου

Συμβολίζουμε με $N_i^{(k)}$ τους άμεσους γείτονες που ο κόμβος i έχει ανακαλύψει μέχρι τη χρονοθυρίδα k . Για κάθε κόμβο $j \in N_i^{(k)}$ ο κόμβος i διατηρεί δύο μετρητές, έναν για τον αριθμό των μηνυμάτων που εστάλησαν στον j στη χρονοθυρίδα k για προώθηση ($S_{ij}^{(k)}$) και έναν δεύτερο για τον αριθμό των μηνυμάτων που ο κόμβος j πραγματικά προώθησε στη χρονοθυρίδα k ($F_{ij}^{(k)}$). Στο τέλος του χρονικού αυτού διαστήματος υπολογίζει για κάθε γείτονά

του το λόγο $c_{ij}^{(k)} = \frac{F_{ij}^{(k)}}{S_{ij}^{(k)}}$ και τον οποίο ανταλλάσσει με όλους τους γείτονές του. Με τις τιμές που έχει συλλέξει υπολογίζει το μέσο λόγο συνδεσιμότητας:

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

Όπου $c_{ii}^{(k)} = 1$ για όλα τα k . Αξίζει να σημειωθεί ότι το μέγεθος αυτό η βαρύτητα με την οποία προσμετράται η γνώμη του κάθε κόμβου είναι ο λόγος αυτού που μετράει ο κόμβος i . Έτσι δεν επιτρέπεται σε έναν κακόβουλο κόμβο με διασπορά ψευδών στοιχείων να αυξήσει τη φήμη του, αφού όλοι οι υπόλοιποι κόμβοι έχουν κακή γνώμη για τον κακόβουλο και επομένως τα στοιχεία του μικρό αντίκτυπο στον υπολογισμό του $\hat{c}_j^{(k)}$. Με παρόμοιο τρόπο, ο κόμβος i θα υπολογίσει το μέσο λόγο συνδεσιμότητας $\hat{c}_i^{(k)}$ που οι άμεσοι γείτονές του αντιλαμβάνονται από αυτόν μέχρι τη χρονοθυρίδα k . Ορίζουμε ως $\hat{p}_j^{(k)} = 1 - \hat{c}_j^{(k)}$ και χρησιμοποιώντας τις εξισώσεις (1) και (2) υπολογίζεται η πιθανότητα απόρριψης που ο κόμβος i θα χρησιμοποιήσει για τον κόμβο j κατά τη χρονοθυρίδα $k + 1$.

Εφόσον απαιτείται $\gamma < p_e^{-1}$, χρειάζεται να υπολογίσουμε το μέγεθος p_e^{-1} . Μία ενδιαφέρουσα λύση είχε προταθεί πλημμυρίζοντας έναν κόμβο με ανώνυμα μηνύματα, αυξάνοντας ωστόσο ραγδαία την δικτυακή κίνηση. Αντίθετα, παρατηρούμε ότι το p_e είναι η πιθανότητα ένα τουλάχιστον τερματικό στο $N_i^{(k)}$ να μεταδίδει ταυτόχρονα με τον j . Έτσι, εκτιμάμε το p_e μετρώντας το ποσοστό του χρόνου που τουλάχιστον ένας κόμβος διαφορετικός του j μεταδίδει. Την εκτίμηση αυτή ονομάζουμε \hat{p}_{ej} . Στην περίπτωση το MAC layer χρησιμοποιεί ένα CSMA/CA πρωτόκολλο έχουμε ότι $\hat{p}_{ej} \geq p_e$ και επομένως $\gamma < \hat{p}_{ej}^{-1} \leq p_e^{-1}$.

2.3.2 Πλεονεκτήματα – Μειονεκτήματα

Όπως αποδεικνύεται από προσομοιώσεις το DARWIN τιμωρεί τους εγωιστικούς κόμβους με αρκετά χειρότερο λόγο προώθησης απέναντι σε αυτούς που συνεργάζονται. Μάλιστα όσο αυξάνεται η πιθανότητα με την οποία απορρίπτονται πακέτα τόσο πέφτει η απόδοσή τους. Επίσης ακόμα και με 90% εγωιστικούς κόμβους η απόδοση των κόμβων που συνεργάζονται είναι μεγαλύτερη των εγωιστών. Σε ποσοστά εγωιστικών κόμβων μέχρι 30% η διαφορά αυτή είναι πολύ μεγάλη. Ο λόγος προώθησης πακέτων παραμένει σχεδόν σταθερός ακόμα και σε υψηλούς ρυθμούς μετάδοσης, στους οποίους το p_e αυξάνεται με αποτέλεσμα να υπάρχουν λανθασμένες θετικές ενδείξεις. Αυτό είναι λογικό αφού δεν χρειάζεται στο DARWIN ακριβής υπολογισμός του p_e , για να επιτύχουμε πλήρη συνεργασία. Αυτό ενισχύεται από το γεγονός ότι ο μηχανισμός δεν

είναι ευαίσθητος στις παραμέτρους του συστήματος. Τέλος, αντιμετωπίζονται αποτελεσματικά τυχόν συμπαιγνίες κακόβουλων κόμβων [19].

3 ΠΡΟΤΕΙΝΟΜΕΝΟΣ ΥΒΡΙΔΙΚΟΣ ΜΗΧΑΝΙΣΜΟΣ ICARUS 2

3.1 ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ

Στο πλαίσιο αυτής της διπλωματικής έχει σχεδιαστεί και υλοποιηθεί ο υβριδικός μηχανισμός κινήτρων ICARUS 2 (hybrid incentive mechanism for cooperation stimulation in ad hoc networks 2) και αποτελεί έναν μηχανισμό που βασίζεται πάνω στο πρωτόκολλο ICARUS [4]. Ο ICARUS υιοθετεί τη χρήση μίας κεντρικής οντότητας παρόμοιας με το SPRITE, έναν μηχανισμό που βασίζεται σε credits. Ο βελτιωμένος μηχανισμός αποτελείται από μία διαφορετική υλοποίηση τόσο σε θεωρητικό επίπεδο όσο και στο τεχνικό μέρος. Ο στόχος είναι να βελτιωθεί η απόδοση του DARWIN χρησιμοποιώντας στοιχεία από τους μηχανισμούς πίστωσης και να βελτιώσει τη λειτουργία του μηχανισμού ICARUS.

Παρακάτω παρουσιάζεται ο πίνακας 3.1 με τους συμβολισμούς που χρησιμοποιούνται στον ICARUS 2.

N	A set containing all nodes
N_c	A set containing only cooperating nodes
$edp_i^{(k)}$	Estimated dropping probability of node i in time slot k
$p_i^{(k)}$	Expected dropping probability of node i in time slot k
$N_i^{(k)}$	A set containing the neighbors of node i in 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 sent from node i to node j 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_r	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
$cost_{i,z}$	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

Πίνακας 3.1 Πίνακας συμβολισμών για τον μηχανισμό ICARUS 2

3.2 ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΔΙΚΤΥΟΥ

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

Αρχικά θεωρούμε ότι ένας μέρος του δικτύου αποτελείται από εγωιστές κόμβους οι οποίοι αρνούνται να προωθήσουν τα πακέτα που τους στέλνονται. Οι κόμβοι αυτοί μπορούν να ταξινομηθούν στους εγωιστές και τους κακόβουλους.

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 άλλον και αποφασίζουν αν θα τα προωθήσουν ή θα τα απορρίψουν. Η αναμενόμενη πιθανότητα του κόμβου i σε μία χρονοθυρίδα k ορίζεται ως $p_i^{(k)}$. Ένα πακέτο θεωρείται απορριφθέν είτε αν ο $-i$ το απορρίψει είτε εάν δε χαθεί αλλά ο i δεν ενημερωθεί ποτέ για την αναμετάδοσή του.

Κάθε κόμβος στέλνει πακέτα περιοδικά, ενώ ο αποστολέας καθορίζεται δυναμικά και αλλάζεται ανά τακτά χρονικά διαστήματα. Οποτεδήποτε κάποιος κόμβους λαμβάνει ένα πακέτο αποστέλλει ένα πακέτο επιβεβαίωσης (το οποίο θα αναφέρουμε ως ack), το οποίο ακολουθεί τη διαδρομή ανάποδα μέχρι να φτάσει στον αποστολέα του πακέτου. Με τον τρόπο αυτό κάθε κόμβος ανανεώνει τα στατιστικά του που κρατάει μόνο όταν λάβει το πακέτο ack. Αυτό σημαίνει ότι ακόμα και αν ένα πακέτο φτάσει επιτυχώς στον προορισμό του, εάν για οποιοδήποτε λόγο χαθεί το ack τότε οι κόμβοι θα υποθέσουν ότι το πακέτο απορρίφθηκε από το γείτονά τους.

Μια σημαντική παραδοχή της εργασίας αυτής είναι ότι οι κόμβοι θα απορρίπτουν τα πακέτα ack με μία πιθανότητα της τάξης του 0.2%, μία τιμή αρκετά μικρή, αφού το μικρό τους μέγεθος δεν απαιτεί πολλούς πόρους για την προώθηση. Επιπλέον, το επόμενο βήμα αναφέρεται μέσα στο πακέτο ack και δεν απαιτείται μεγάλη ισχύς για την εύρεση του επόμενου κόμβου.

Συμβολίζουμε με $N_i^{(k)}$ τους γείτονες i τη χρονοθυρίδα k . Ο κόμβος i κρατάει τα εξής στατιστικά για κάθε γείτονα j :

- Τον αριθμό των πακέτων που στάλθηκαν στον j για προώθηση μέχρι τη χρονοθυρίδα k : $S_{ij}^{(k)}$,
- Τον αριθμό των πακέτων που προωθήθηκαν από τον j μέχρι τη χρονοθυρίδα k : $F_{ij}^{(k)}$.

Έπειτα ο i υπολογίζει το ποσοστό προώθησης του j ως

$$C_{ij}^{(k)} = \frac{F_{ij}^{(k)}}{S_{ij}^{(k)}}.$$

Το οποίο πρακτικά εκφράζει το κατά πόσο ένας κόμβος είναι συνεργάσιμος ή όχι. Το ποσοστό αυτό χρησιμοποιείται και στη δρομολόγηση. Στη συνέχεια οι κόμβοι ανταλλάσσουν μεταξύ τους αυτή τη πληροφορία και τελικά, όταν έχουν συγκεντρωθεί όλες οι τιμές, ο κόμβος i υπολογίζει το μέσο ποσοστό συνδεσιμότητας του γείτονα j ως εξής:

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

Πρέπει να σημειωθεί ότι ο μέσος σταθμίζεται με την εκτιμώμενο ποσοστό συνδεσιμότητας που μετράει ο i για το γείτονα m . Αυτό μας βοηθάει στο να αποφύγουμε επιθέσεις και διάδοση ψευδούς πληροφορίας. Με τον τρόπο αυτό ελαχιστοποιούμε τη σημασία της πληροφορίας που προέρχεται από αναξιόπιστους κόμβους.

3.3 ΔΡΟΜΟΛΟΓΗΣΗ

Στην ενότητα αυτή, θα περιγράψουμε πως πραγματοποιείται η δρομολόγηση στα πλαίσια αυτού του μηχανισμού. Η διαδικασία δρομολόγησης στον ICARUS 2 είναι η ίδια με το πρωτόκολλο ICARUS. Είναι όμως σημαντική η αναφορά της για την πλήρη κατανόηση της διαδικασίας εύρεσης νέων γειτόνων και διαδρομών.

Ως βέλτιστη διαδρομή θεωρούμε αυτή που περιέχει τον ελάχιστο αριθμό ενδιάμεσων κόμβων. Στην περίπτωση που εντοπιστούν δύο ή περισσότερες διαδρομές με τον ίδιο αριθμό ενδιάμεσων κόμβων, τότε προτιμάται αυτή που αντιστοιχεί στη μικρότερη συνολική απόσταση. Η συνολική απόσταση υπολογίζεται ως το άθροισμα όλων των βημάτων (hops). Κάθε κόμβος διατηρεί πληροφορία σχετικά με το ποιος κόμβος θα χρησιμοποιηθεί ως ενδιάμεσος ανάλογα με τον εκάστοτε προορισμό και όχι τη συνολική διαδρομή.

Πιο συγκεκριμένα, κάθε κόμβος διατηρεί δύο πίνακες στη βάση δεδομένων του. Ο πρώτος περιλαμβάνει τις εγγραφές για όλους τους γείτονές του και περιέχει πεδία όπως το id , τις συντεταγμένες του, την απόσταση του, φήμη και στατιστικά. Ο δεύτερος πίνακας περιλαμβάνει τις εγγραφές για τους επόμενους κόμβους και περιέχει πεδία όπως το id , τον ενδιάμεσο γείτονα που θα χρησιμοποιηθεί για τη δρομολόγηση, τον συνολικό αριθμό hops και τη συνολική απόσταση σε μέτρα.

Προκειμένου να αποφευχθεί η απώλεια πακέτων και να απομονωθούν οι εγωιστές κόμβοι, υποθέτουμε ότι οι κόμβοι δε θα προωθούν πακέτα σε εγωιστές κόμβους ακόμα και αν θεωρούνται βέλτιστοι ενδιάμεσοι σε αποθηκευμένες διαδρομές. Ο κόμβος i θεωρεί τον κόμβο

j εγλωιστή 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 εγλωιστές κόμβοι.

Στο πρώτο στάδιο εκκίνησης του συστήματος ο κόμβος A στέλνει μηνύματα αναγνώρισης γειτόνων (broadcast) και οι κόμβοι που λαμβάνουν τα μηνύματα αυτά απαντάνε με επιβεβαίωση, προσθέτοντας παράλληλα τον A στη λίστα των γειτόνων τους γεμίζοντας με εγγραφές τον πρώτο πίνακα. Στη συνέχεια ο κάθε κόμβος αναδημοσιεύει στους γείτονες του τη λίστα με τους απομακρυσμένους κόμβους, για τους οποίους δεν έχει προς το παρών καμία πληροφορία. Οι γείτονες αξιολογούν την όποια πληροφορία λάβουν και αν ανακαλύψουν ότι ο κόμβος διαθέτει μία καλύτερη διαδρομή για κάποιον απομακρυσμένο κόμβο, τότε ενημερώνουν την αντίστοιχη εγγραφή και τη νέα διαδρομή.

Η βασική ιδέα του αλγορίθμου δρομολόγησης είναι ότι οι κόμβοι δεν πρέπει να προωθούν πακέτα σε εγλωιστικούς κόμβους. Αν το επόμενο βήμα της διαδρομής αντιστοιχεί σε εγλωιστικό κόμβο, τότε πρέπει να βρεθεί εναλλακτική διαδρομή. Επιπλέον, προκειμένου να μειωθούν οι μη απαραίτητες αποστολές πακέτων και να αποφευχθεί το φαινόμενο του ring rong, θεωρούμε ότι ένα πακέτο δεν μπορεί να προωθηθεί σε έναν κόμβο από τον οποίο έχει ήδη περάσει. Επιπλέον, ο μέγιστος αριθμός επιτρεπόμενων βημάτων περιορίζεται σε h_{\max} . Μετά από αυτό τον αριθμό βημάτων το πακέτο αυτόματα απορρίπτεται. Το h_{\max} πρέπει να καθοριστεί από το σχεδιαστή του συστήματος, ανάλογα με την τοπολογία του δικτύου (αριθμός και πυκνότητα κόμβων), το μέσο αριθμό απαιτούμενων βημάτων, τον αναμενόμενο αριθμό εγλωιστικών κόμβων, κτλ.

Ας υποθέσουμε ότι έχουν προηγηθεί οι παραπάνω διαδικασίες και ο κόμβος A αναζητεί διαδρομή για τον κόμβο B . Αν το επόμενο βήμα της διαδρομής που υποδεικνύουν οι πίνακες δρομολόγησης του A δεν υπάρχει, τότε ρωτάει τους γείτονες για την προτεινόμενη διαδρομή και συγκρίνοντάς τις αποφασίζει για τη βέλτιστη. Αν το επόμενο βήμα υπάρχει και δεν αντιστοιχεί σε εγλωιστή κόμβο ή σε κόμβο που υπάρχει ήδη στη διαδρομή, τότε το πακέτο προωθείται.

Διαφορετικά αναζητείται εναλλακτική διαδρομή προς τον B από τους εναπομείναντες γειτονικούς κόμβους του A . Η βέλτιστη διαδρομή είναι αυτή που περιλαμβάνει τα λιγότερα βήματα και τη μικρότερη συνολική απόσταση. Εάν με αυτόν τον τρόπο εντοπιστεί μία εναλλακτική διαδρομή προς τον B , τότε το πακέτο προωθείται. Επιπλέον, εάν ο επόμενος κόμβος της παλιάς διαδρομής ήταν εγλωιστής, τότε η νέα διαδρομή αποθηκεύεται αντικαθιστώντας την παλιά. Από την άλλη πλευρά, στην περίπτωση που δεν εντοπιστεί εναλλακτική διαδρομή, τότε το πακέτο προωθείται στον αρχικό του προορισμό, αρκεί αυτός να μην περιλαμβάνεται ήδη στο μονοπάτι της διαδρομής (ανεξαρτήτως αν είναι εγλωιστικός ή όχι). Αν ανήκει ωστόσο στο μονοπάτι, αναζητείται εναλλακτική στους γείτονες του A , αλλά αυτή τη

φορά λαμβάνονται υπόψη και οι εγωιστές. Αν ο *A* αποτύχει πάλι να βρει διαδρομή, τότε το πακέτο τελικά απορρίπτεται.

Ο ψευδοκώδικας της εικόνας παρακάτω επεξηγεί τη διαδικασία δρομολόγησης.

```
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 stored 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
```

3.4 ΣΤΡΑΤΗΓΙΚΗ ΕΓΩΙΣΤΙΚΩΝ ΚΟΜΒΩΝ

Όπως και στο πρωτόκολλο ICARUS θεωρούμε ότι η συμπεριφορά των εγωιστών κόμβων είναι η ίδια ανεξαρτήτως του μηχανισμού αντιμετώπισής τους. Ένα συγκεκριμένο ποσοστό κόμβων θεωρείται εγωιστικό. Οι κόμβοι αυτοί αρνούνται να προωθήσουν πακέτα όσο αντιλαμβάνονται ότι μπορούν να χρησιμοποιούν τους πόρους του δικτύου προς όφελός τους. Μόλις ωστόσο αντιληφθούν ότι τα πακέτα τους πλέον δεν προωθούνται, θεωρούν τον εαυτό τους απομονωμένο και αρχίζουν να προωθούν μέχρι την άρση της απομόνωσης. Με άλλα λόγια, η απομόνωση από το δίκτυο αποτελεί κίνητρο για μεταστροφή και κατά συνέπεια μη εγωιστική συμπεριφορά.

Ως απομόνωση θεωρούμε την κατάσταση όπου ένας κόμβος αποτυγχάνει να έχει έστω και ένα προωθημένο πακέτο του για ένα συγκεκριμένο χρονικό διάστημα. Ο συνολικός αριθμός πακέτων που πρέπει να απορριφθούν προτού ένας εγωιστής κόμβος θεωρήσει τον εαυτό του απομονωμένο συμβολίζεται ως *IFN* (*Isolation Flag Nuber*).

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 Behavior Index*) για τον κόμβο i αφότου έχει σταλεί ένα πακέτο με σειριακό αριθμό r , ορίζεται ως:

$$SBI_i^{(r)} = f$$

Με βάση την προηγούμενη σχέση, οι εγωιστικοί κόμβοι θα ξεκινήσουν να συνεργάζονται μόνο όταν $SBI = 0$. Διαφορετικά, αν έστω και ένα πακέτο τους έχει προωθηθεί επιτυχώς και συνεπώς $SBI > 0$, θα συνεχίσουν να κλέβουν. Η στρατηγική επομένως των εγωιστικών κόμβων συνοψίζεται ως:

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

3.5 ΣΤΡΑΤΗΓΙΚΗ ΚΑΚΟΒΟΥΛΩΝ ΚΟΜΒΩΝ

Μία σημαντική προσθήκη του μηχανισμού ICARUS 2 σε σχέση με το πρωτόκολλο ICARUS είναι ο μηχανισμός εντοπισμού και απομόνωσης των κακόβουλων κόμβων, μία κατηγορία κόμβων για την οποία ο ICARUS δεν είχε καμία πρόβλεψη. Η αντιμετώπιση αυτής της κατηγορίας είναι εξίσου σημαντική με την εξάλειψη της εγωιστικής συμπεριφοράς αφού αποτελεί μία πολύ κοινή

απειλή για υποβάθμιση της ποιότητας υπηρεσιών του MANET. Χρήζει επομένως ξεχωριστής αντιμετώπισης και ενδεδειγμένης έρευνας για την καλύτερη αντιμετώπισή της από το σύστημα.

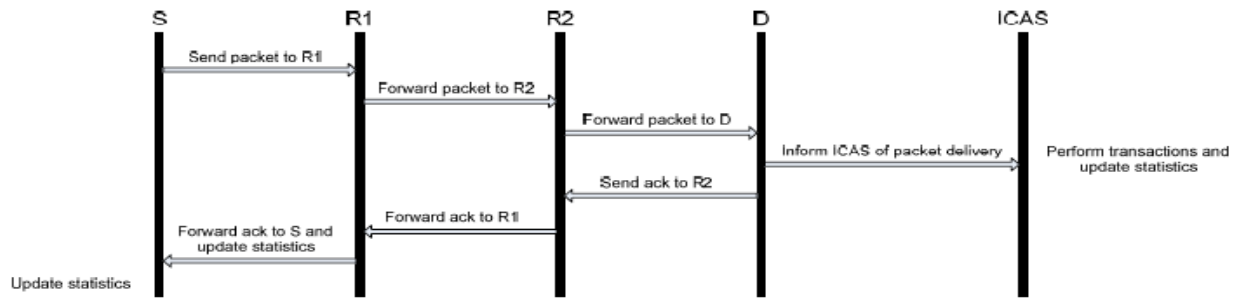
Ένα σαφώς μικρότερο ποσοστό κόμβων θεωρείται κακόβουλο. Οι κόμβοι αυτοί όχι μόνο αρνούνται να προωθήσουν πακέτα και *acks* αλλά συμμετέχοντας στον αλγόριθμο του *Darwin* αναμεταδίδουν ψευδείς τιμές για φήμες των γειτονικών τους κόμβων καθώς και αποστέλλουν ψεύτικες τιμές και αποδείξεις επιτυχημένης αποστολής πακέτων στην κεντρική τράπεζα. Επιπλέον, ενδέχεται να δρομολογούν επιλεκτικά πακέτα ώστε να βελτιώνουν λανθασμένα την άποψη των γειτόνων τους γι' αυτούς.

3.6 ΣΥΝΕΡΓΑΣΙΑ ΚΟΜΒΩΝ ΣΤΟΝ ICARUS 2

Στην ενότητα αυτή θα περιγραφούν οι μηχανισμοί συνεργασίας των συνεργατικών κόμβων κατά το μηχανισμό ICARUS 2, όπου όπως και στο ICARUS, εκμεταλλευόμαστε έννοιες από το μηχανισμό DARWIN όπως είναι η εκτίμηση πιθανότητας απόρριψης πακέτων και το ποσοστό συνδεσιμότητας. Στο ICARUS 2 εκμεταλλευόμαστε επίσης τη φήμη για τον πλήρη έλεγχο των *credits* στην ψηφιακή τράπεζα. Η ψηφιακή τράπεζα αυτή είναι μία ξεχωριστή και ανεξάρτητη οντότητα, το όνομα της οποίας παραμένει το ίδιο όπως στον ICARUS και ονομάζεται *ICAS* (*Icarus Credits Accounts Service*). Οι κύριοι στόχοι του ICAS είναι:

- Ταχύτερη και αυστηρότερη τιμωρία των εγωιστικών κόμβων σε σχέση με το DARWIN
- Βελτίωση του QoS για τους μη εγωιστικούς κόμβους
- Κίνητρα για συνεργαζόμενους κόμβους, ώστε να συνεχίσουν να προωθούν πακέτα
- Δικαιοσύνη για μακρινούς κόμβους
- Αποκλεισμός πιθανότητας οι εγωιστικοί κόμβοι να εκμεταλλευτούν μία μακρινή θέση για να πλουτίσουν
- Προστασία του συστήματος από κακόβουλους κόμβους που διαδίδουν ψευδείς πληροφορίες

Η παρακάτω εικόνα απεικονίζει τις δράσεις που λαμβάνουν χώρα μέχρι να παραδοθεί ένα πακέτο από την πηγή (**S**) στον προορισμό του (**D**) με δύο ενδιάμεσους κόμβους (**R1**, **R2**). Όπως μπορεί να παρατηρηθεί, κάθε κόμβος ανανεώνει τα στατιστικά του μόλις λάβει την επιβεβαίωση ότι ο επόμενος κόμβος έχει συνεργαστεί.



Εικόνα 3.1

Εδώ αξίζει να σημειωθεί ότι τίθεται θέμα επιπλέον overhead του δικτύου λόγω της ενημέρωσης του ICAS για την επιτυχημένη αποστολή πακέτων. Στο πλαίσιο της διπλωματικής προτείνεται αποστολή μίας ομάδας αποδείξεων ανα τακτά χρονικά διαστήματα.

3.6.1 Λειτουργίες του ICAS

Το ICAS έχει πρακτικά το ρόλο της τράπεζας, πραγματοποιώντας όλες τις ανταλλαγές *credits* (πληρωμές, κοστολόγηση, αμοιβές, κτλ) ως ένας αντικειμενικός παρατηρητής του δικτύου. Αυτό γίνεται προκειμένου να αποφευχθεί η κλοπή *credits* ή η διάχυση ψευδών πληροφοριών από κακόβουλους κόμβους. Οι κύριες λειτουργίες του ICAS είναι οι παρακάτω:

- Το ICAS διατηρεί τα *credits* κάθε κόμβου.
- Το ICAS αναθέτει ένα μικρό αριθμό *credits* m_0 σε όλους τους καινούργιους κόμβους.
- Όταν λαμβάνεται επιβεβαίωση επιτυχημένης 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 συνεργαζόμενων κόμβων. Οι κόμβοι δεν έχουν γνώση του ακριβούς ποσού των *credits* που τους απονέμονται καθώς και ούτε του συνολικού αριθμού *credits* που διαθέτουν στην εικονική τράπεζα.
- Το ICAS επιτρέπει ή απαγορεύει στους κόμβους να στείλουν τα πακέτα ανάλογα με τον αριθμό των *credits* που διαθέτουν.
- Το ICAS βοηθάει τους απομακρυσμένους κόμβους αν χρειάζεται
- Συλλέγει στατιστικά φήμης από όλους τους κόμβους και υπολογίζει τους εγωιστικές και κακόβουλους κόμβους.
- Δημοσιεύει τη λίστα με τους εγωιστές και κακόβουλους κόμβους σε όλους τους κόμβους ανά διαστήματα.

Οι επόμενες ενότητες επεξηγούν αναλυτικά τις λειτουργίες του ICAS. Κάθε κόμβος υπολογίζει το ποσοστό προώθησης $c_{ij}^{(k)}$ για όλους τους γείτονές του, με βάση την εξίσωση που έχει προαναφερθεί. Το ποσοστό αυτό το προωθούν στο ICAS, το οποίο στη συνέχεια υπολογίζει το μέσο ποσοστό συνδεσιμότητας $cp_{ICAS,i}^{(k)}$ για κάθε κόμβο, με βάση τη συγκεντρωτική πληροφορία. Υποθέτοντας ότι N είναι το σύνολο των κόμβων και $N_c \in N$ είναι ένα υποσύνολο που περιέχει μόνο τους συνεργάσιμους κόμβους,

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

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

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

Ως μέτρο ασφάλειας, το ICAS αγνοεί τις παρατηρήσεις των κόμβων 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 κόμβοι διαδώσουν ψευδή πληροφορία, το ICAS δε θα επηρεαστεί σημαντικά. Προβλήματα φυσικά θα παρουσιαστούν στην περίπτωση που ο αριθμός των κόμβων που ψεύδονται είναι υπερβολικά μέγεθος (π. χ. > 50%).

3.6.2 Τιμολόγηση στο ICARUS 2

Οι κόμβοι διαφοροποιούνται σε τρεις διαφορετικές καταστάσεις ανάλογα με τον αριθμό των credits που διαθέτουν στο ICAS. Αν ο αριθμός των credits που έχει ο κόμβος i τη χρονοθυρίδα k είναι $m_i^{(k)}$ και το κατώφλι των credits είναι S_t , 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 $m_i^{(k)} < 0$.

Σύμφωνα με τις παραπάνω καταστάσεις, ένας κόμβος χωρίς credits δεν έχει το δικαίωμα να αποστείλει πακέτα. Το ICAS είναι υπεύθυνο για να καταγράφει τα credits κάθε κόμβου και κατά συνέπεια να προσδιορίζει σε ποια κατάσταση βρίσκεται. Σε αυτό το σημείο γίνεται η παραδοχή ότι χωρίς εξουσιοδότηση από το ICAS οι κόμβοι δεν μπορούν να στείλουν πακέτα.

Η τιμολόγηση στο ICARUS 2 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 κάθε κόμβου i για την προώθηση του πακέτου z τη χρονοθυρίδα είναι

$$r_{i,z} = \begin{cases} 0.5 + 2.5 \cdot [cp_i^{(k)}]^2, & \text{αν } m_i^{(k)} \leq M_{thr1} \\ 0.5 + 0.5 \cdot [cp_i^{(k)}]^2, & \text{αν } m_i^{(k)} \leq M_{thr2} \\ 0.01 \cdot (M_{\max} - m_i^{(k)}), & \text{αν } m_i^{(k)} \leq M_{\max} \end{cases}$$

Όπου στη δικιά μας περίπτωση έχουμε θέσει $M_{thr1} = 600$, $M_{thr2} = 1000$, $M_{\max} = 1200$.

Έχουμε επισημάνει σε προηγούμενη ενότητα ότι ένας από τα σημαντικά προβλήματα των μηχανισμών πίστωσης είναι ο άκρατος πλουτισμός ορισμένων συνεργάσιμων κόμβων, οι οποίοι επωφελούνται της κατάστασης και αρχίζουν και δρουν εγωιστικά. Με το συγκεκριμένο σύστημα αμοιβών, αναστέλλεται ο άκρατος πλουτισμός μέχρι ο κόμβος να φτάσει το M_{\max} ενώ έχουμε θέσει και άλλα δύο κατώφλια credits. Όσο πιο πολύ αυξάνονται τα credits ενός κόμβου, τόσο πιο αργά ο κόμβος αυτός συλλέγει νέα credits λόγω της συνεργατικής του συμπεριφοράς. Στην προηγούμενη υλοποίηση ICARUS δεν προβλεπόταν ο περιορισμός αυτός με αποτέλεσμα η συσσώρευση μεγάλου αριθμού credits που μάλιστα αυξανόταν αναλογικά με το χρόνο.

Στο μηχανισμό ICARUS η ανταμοιβή των εγωιστών κόμβων γινόταν σύμφωνα με τη σχέση:

$$r_{i,z} = a + b \cdot [cp_i^{(k)}]^2, \text{ όπου } a = 0.5 \text{ και } b = 2.3$$

η οποία όμως δεν προβλέπει τον περιορισμό του πλουτισμού ορισμένων υπερ- συνεργατικών κόμβων.

Αφού ο αποστολέας υποχρεούται να ανταμείψει όλους τους κόμβους που λειτούργησαν ως ενδιάμεσοι, το συνολικό κόστος $cost_{i,z}$ για το πακέτο z που στάλθηκε από τον κόμβο i υπολογίζεται ως:

$$cost_{i,z} = \sum_{j \in N, j \in rI_{i,z}} r_{i,z}$$

Όπου $rl_{i,z}$ είναι μία λίστα όλων των κόμβων που προώθησαν το πακέτο z . Όπως φαίνεται και στην παραπάνω εικόνα, όλες οι ανταλλαγές των credits γίνονται αφού ένα πακέτο φτάσει στον προορισμό του και ο παραλήπτης ενημερώσει το ICAS. Κατά συνέπεια, αν το πακέτο δεν φτάσει ποτέ στον προορισμό του, τότε το ICAS δεν ενημερώνεται ποτέ και όσοι κόμβοι προώθησαν ήδη το πακέτο δεν ανταμείβονται. 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 υπερβολικοί πόροι για την εξακρίβωση του κάθε ισχυρισμού.

Όσο αναφορά επιπλέον τους πολύ πλούσιους κόμβους, θα χρεώνονται επιπλέον από το ICAS για κάθε δικιά τους αποστολή πακέτου προκειμένου ο αριθμός των credits να κινείται σε λογικά με τους υπόλοιπους κόμβους πλαίσια. Έτσι όμοια με τις αμοιβές, η επιπλέον χρέωση εξαρτάται από το τετράγωνο του cp και η χρέωση προσαρμόζεται ως:

$$r_{i,z}^{adj} = \begin{cases} 2.6 \cdot r_{i,z}, & \text{αν } m_i^{(k)} > 5.7 \cdot m_0 \\ 1.3 \cdot r_{i,z}, & \text{αν } m_i^{(k)} > 2.5 \cdot m_0 \end{cases}$$

3.6.3 Εντοπισμός εγωιστικών κόμβων

Ο μηχανισμός ICARUS 2 χρησιμοποιεί ένα παρόμοιο μηχανισμό απόρριψης πακέτων με το ICARUS, σύμφωνα με τον οποίο αν ένας κόμβος j θεωρείται εγωιστής από τον κόμβο i , τότε ο τελευταίος απορρίπτει πακέτα με βάση την εκτιμώμενη πιθανότητα απόρριψης $edp_i^{(k)}$. Αν ο j δε θεωρείται εγωιστής, τότε ο i προωθεί πάντα τα πακέτα του. Ο εντοπισμός των εγωιστικών κόμβων επιτυγχάνεται με τους ακόλουθους τρόπους:

- Το ICAS μεταδίδει περιοδικά μία λίστα με όλους τους αναγνωρισμένους κακόβουλους και εγωιστές κόμβους. Αν ένας κόμβος εντοπίσει κάποιον γείτονά του σε αυτή τη λίστα, τότε τον προσθέτει στη 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 στη δική του τοπική λίστα.

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

3.6.4 Εντοπισμός κακόβουλων κόμβων

Ο μηχανισμός ICARUS 2 χρησιμοποιεί επίσης έναν μηχανισμό εντοπισμού των κακόβουλων κόμβων, στοιχείο που απουσίαζε από τη μελέτη σχεδιασμού του ICARUS. Στο πλαίσιο της διπλωματικής, υποθέτουμε ότι οι κακόβουλοι κόμβοι απορρίπτουν όλα τα πακέτα ενώ αναμεταδίδουν ψευδώς στους γείτονές τους ότι τα μετέδωσαν. Επιπλέον, στέλνουν ψευδείς τιμές φήμης στο ICAS.

Το ICAS αφού έχει αναπτύξει μία γνώση για τη μέση φήμη του κάθε κόμβου, ξεκινάει από ένα χρονικό διάστημα και μετά το μηχανισμό εντοπισμού κακόβουλων κόμβων. Σύμφωνα με το μηχανισμό αυτό τσεκάρει σε κάθε ανανέωση στατιστικών αν η νέα τιμή φήμης ενός κόμβου αποκλίνει από τη μέση τιμή και από την προηγούμενη τιμή που πρότεινε ο συγκεκριμένος γείτονας. Πιο συγκεκριμένα:

$$i = \begin{cases} \text{malicious, if } |cp_{ji}^k - \widehat{cp}_i^{k-1}| \geq 0.4 \\ \text{non malicious, if } |cp_{ji}^k - \widehat{cp}_i^{k-1}| < 0.4 \end{cases}$$

Όπου $cp_{ij}^{(k)}$ η άποψη του κόμβου i για τον κόμβο j τη χρονική θυρίδα k και \widehat{cp}_i^k η μέση τιμή φήμης του κόμβου i τη χρονική θυρίδα k .

Όταν ένας κόμβος θεωρηθεί κακόβουλος καταγράφεται στη λίστα κακόβουλων κόμβων του ICAS. Για να διαγραφεί, απαιτείται αλλαγή συμπεριφοράς του συγκεκριμένου κόμβου. Συγκεκριμένα για τις επόμενες M_k χρονικές θυρίδες θα πρέπει να επιδείξει μία κακόβουλη συμπεριφορά. Αν ικανοποιηθεί αυτή η συνθήκη, τότε αυτομάτως διαγράφεται από τη λίστα κακόβουλων κόμβων.

3.6.5 Απομακρυσμένοι κόμβοι

Οι περισσότεροι μηχανισμοί κινήτρων πάσχουν από το πρόβλημα της ανισότητας όσον αφορά τους απομακρυσμένους κόμβους, οι οποίοι λόγω της θέσης τους δεν μπορούν να συγκεντρώσουν αρκετά credits ή να βελτιώσουν τη φήμη τους προκειμένου να μπορούν να αποστέλλουν τα δικά τους πακέτα. Ένας δίκαιος μηχανισμός θα έπρεπε να επιτηρεί και ενισχύει τους κόμβους. Ο ICARUS 2 ωστόσο δεν ακολουθεί ακριβώς αυτή την προσέγγιση, αφού έτσι οι απομακρυσμένοι κόμβοι θα μπορούσαν δυνητικά να συγκεντρώσουν υπερβολικά πολλά credits χωρίς να προωθήσουν ποτέ κανένα πακέτο.

Στον προτεινόμενο αλγόριθμο, οι απομακρυσμένοι κόμβοι ενισχύονται από το δίκτυο και εν τέλει τους επιτρέπεται να στέλνουν πακέτα δωρεάν. Αυτό ωστόσο συμβαίνει μόνο στην περίπτωση της ολικής χρεωκοπίας τους. Με άλλα λόγια, τους επιτρέπεται να στέλνουν πακέτα

δωρεάν, αλλά όχι να πλουτίσουν μέσω αυτής της διαδικασίας. Έτσι εξασφαλίζουμε ότι οι κακόβουλοι κόμβοι δε θα εκμεταλλευτούν τη θέση τους για δικό τους όφελος.

Στο ICARUS 2 η βοήθεια που αναφέραμε προέρχεται από το ICAS. Οι απομακρυσμένοι κόμβοι εντοπίζονται από τον αριθμό των πακέτων που τους έχει ζητηθεί να προωθήσουν στο παρελθόν. Εδώ ως απομακρυσμένοι κόμβοι περιλαμβάνονται και οι κόμβοι οι οποίοι δεν βρίσκονται στις βέλτιστες διαδρομές δρομολόγησης άλλων κόμβων. Αν ο αριθμός τους 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 μία σταθερή τιμή $m_{e,0}$ 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:

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

```

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

```

Η παραπάνω εικόνα περιγράφει με ψευδοκώδικα το πώς ενισχύεται ένας απομακρυσμένος κόμβος i . Με $S_i^{(k)}$ συμβολίζουμε τον συνολικό αριθμό πακέτων που έχουν σταλεί για τον i για προώθηση μέχρι τη χρονοθυρίδα k . Προφανώς $S_i^{(k)} = \sum_{j \in N_i} S_{ji}^{(k)}$. Ο υποστηρικτικός μηχανισμός ICARUS ξεχωρίζει δύο κατηγορίες κόμβων: αυτούς που χρειάζονται ενίσχυση για να αποκτήσουν τον απαιτούμενο αριθμό 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 $S_i^{(k)} \leq k \cdot pf_1 \cdot cp_i^{(k)}$ και λαμβάνει credits μέχρι να εισέλθει στην κατάσταση 1 ή 2. Στην περίπτωση αυτή το ICAS επιτρέπεται δωρεάν αποστολή πακέτων. Η δεύτερη κατηγορία απομακρυσμένων κόμβων, για την οποία ισχύει η

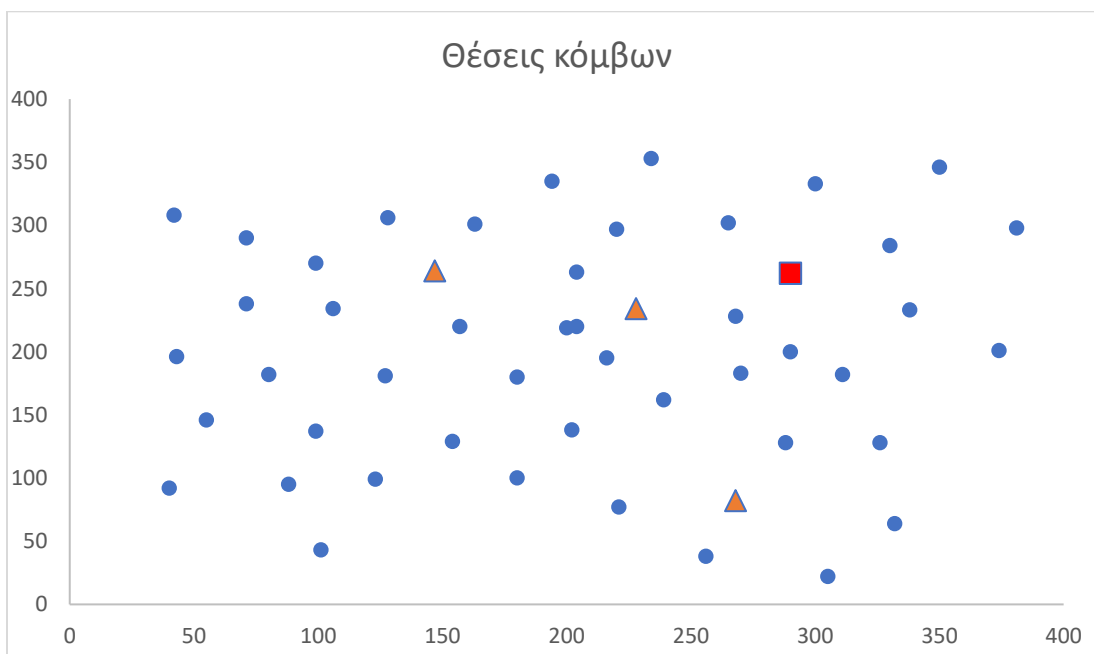
συνθήκη $S_i^{(k)} > k \cdot pf_1 \cdot cp_i^{(k)}$, λαμβάνει περισσότερα credits αλλά δεν της επιτρέπεται η δωρεάν αποστολή πακέτων.

4 ΠΡΟΣΟΜΟΙΩΣΗ ΜΗΧΑΝΙΣΜΩΝ DARWIN ΚΑΙ ICARUS 2

Στο πλαίσιο του 4^{ου} κεφαλαίου, θα αναπτυχθεί αναλυτικά ο σχεδιασμός και ο τρόπος υλοποίησης του υβριδικού μηχανισμού του οποίου η λειτουργία αναπτύχθηκε στον 3^ο κεφάλαιο. Είναι ένας μηχανισμός που συνδυάζει το σύστημα φήμης και πίστωσης παράλληλα για μεγαλύτερη αποτελεσματικότητα και αυξημένη ποιότητα υπηρεσιών.

4.1 ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΠΡΟΣΟΜΟΙΩΣΗΣ

Για τη μελέτη της αποτελεσματικότητας του υβριδικού μηχανισμού πραγματοποιήθηκε μία προσομοίωση σε μία τοπολογία 50 κόμβων οι οποίοι είναι διεσπαρμένοι σε ένα πλέγμα 400×400 τ.μ., φροντίζοντας να έχουμε πιο πυκνή κατανομή κόμβων στο κέντρο παρά στη περιφέρεια του χώρου. Ένας κόμβος θεωρείται γειτονικός εφόσον απέχει το πολύ 60μ από κάποιον άλλον.



Όπου με:

- Συμβολίζουμε τους συνεργατικούς κόμβους
- ▲ Συμβολίζουμε τους εγωιστές κόμβους
- Συμβολίζουμε τους κακόβουλους κόμβους

4.2 ΠΡΟΣΟΜΟΙΩΣΗ

Για να γίνει ξεκάθαρη η λειτουργία του αλγορίθμου, θα προηγηθεί μία σύντομη περιγραφή της λογικής λειτουργίας της προσομοίωσης.

Στο πρώτο στάδιο του αλγορίθμου ο κάθε κόμβος εκτελεί μία διαδικασία αρχικοποίησης κατά την οποία ευρίσκει τους γειτονικούς του κόμβους. Εν συνεχεία, ανταλλάσσει πίνακες δρομολόγησης με τους γειτονικούς του κόμβους μέχρις ότου αποκτήσει πληροφορίες δρομολόγησης για όλους τους απομακρυσμένους κόμβους του πλέγματος. Έπειτα, εγγράφεται στην εικονική τράπεζα *ICAS* όπου στέλνει πρόσθετες πληροφορίες σηματοδοσίας. Στο σημείο αυτό, αρχίζει η πραγματική ανταλλαγή πληροφορίας με τους υπόλοιπους κόμβους που απευθύνεται σε τυχαίους παραλήπτες. Στη προσομοίωση μας, έχουμε ορίσει προκαθορισμένους προορισμούς για κάθε κόμβο για ευκολία σύγκρισης αποτελεσμάτων και εξαγωγής στατιστικών. Εφόσον, ένας κόμβος παραλάβει το πακέτο δημιουργεί ένα *Ack* το οποίο ακολουθεί την αντίστροφη πορεία κατά τη διάρκεια της οποίας ανανεώνονται τα στατιστικά των κόμβων. Τέλος, ανά μία προκαθορισμένη περίοδο, οι κόμβοι ανταλλάσσουν μεταξύ τους τις τιμές φήμης που διαθέτουν και στέλνουν και τις ενημερώσεις αυτές και στο *ICAS*.

Το *ICAS* συλλέγει τις πληροφορίες αυτές και προσθέτει ή αφαιρεί τους κόμβους από τις λίστες εγωιστών και κακόβουλων κόμβων τις οποίες και αναδιανέμει στους κόμβους για να βελτιώσουν την απόδοσή τους. Επίσης, συλλέγει όλες τις αποδείξεις επιτυχημένης αποστολής πακέτων και ρυθμίζει ανάλογα τα *credits* του κάθε κόμβου.

Για την καλύτερα ρεαλιστική προσέγγιση ενός πραγματικού συστήματος, η προσομοίωση πραγματοποιήθηκε στη γλώσσα Java.

Πιο συγκεκριμένα, το συνολικό πρόγραμμα χωρίστηκε σε 3 διαφορετικά modules. Το πρώτο αφορούσε την λειτουργία των κόμβων του πλέγματος (*node module*), το δεύτερο τη λειτουργία του *ICAS* (*icas module*) και το τρίτο περιείχε κλάσεις και μεταβλητές που ήταν κοινές για τα δύο προηγούμενα modules (*common module*) και δεν παρείχε καμία επιπλέον λειτουργικότητα. Στόχος του παραπάνω σχεδιασμού ήταν η ανεξάρτητη λειτουργία του *ICAS*

από τους κόμβους καθώς και ο διαχωρισμός της κοινής τους μνήμης ώστε να ληφθούν υπόψη τα φαινόμενα που έχουν να κάνουν με την επικοινωνία των κόμβων με το ICAS και να μετρηθούν παράμετροι όπως χρόνος καθυστέρησης, συνολικό bandwidth σύνδεσης κλπ.

4.2.1 Maven: Σύστημα διαχείρισης

Για το συντονισμό και έλεγχο των παραπάνω modules χρησιμοποιήθηκε το σύστημα διαχείρισης λογισμικού (*software project management and comprehension tool*) που ονομάζεται *maven*. Η σημασία ύπαρξης ενός τέτοιου συστήματος σε μεγάλα προγράμματα λογισμικού είναι κομβική. Ο λόγος είναι ότι παρότι ο έλεγχος των εξαρτήσεων των modules σε μικρά projects είναι μία εύκολη διαδικασία, όταν ένα project λαμβάνει μεγαλύτερες διαστάσεις και αποτελείται από αρκετά επιμέρους κομμάτια κώδικα, τότε ο έλεγχος αυτός μπορεί να ξεφύγει πολύ εύκολα από τα χέρια του προγραμματιστή.

Το σύστημα διαχείρισης βοηθάει στη ξεκάθαρη διάκριση των διαφόρων modules αναμεταξύ τους και τον έλεγχο των dependencies του ενός από το άλλο. Στην προσομοίωση του ICAS το *node module* και το *icas module* είχαν εξάρτηση από το *common module*, το οποίο δεν είχε εξάρτηση από κανένα. Αυτό βοήθησε στην ανεξάρτητη ανάπτυξη του δύο μεγάλων *module*, τα οποία μπορούσαν να μεταγλωττιστούν (*compile*) και να εκτελεστούν (*execute*) ξεχωριστά, δίχως να έχει γνώση το ένα για το άλλο.

Συνολικά η ανάπτυξη ενός μεγάλου συστήματος κώδικα, που υποβοηθείται από την ύπαρξη ενός συστήματος διαχείρισης όπως το *maven*, αποτελείται από τα εξής βήματα:

- Τη συλλογή και ανάλυση των απαιτήσεων του ολικού project
- Το θεωρητικό σχεδιασμό της λύσης που θα πρέπει να υλοποιηθεί, σημείο στο οποίο αποφασίστηκε ο διαχωρισμός του συνολικού project σε τρία επιμέρους μικρότερα modules
- Την ανάπτυξη του κώδικα
- Το απαραίτητο testing για την ορθή λειτουργία του κώδικα
- Τη διατήρηση του μεγάλου όγκου κώδικα

4.3 ΠΕΡΙΓΡΑΦΗ ΚΛΑΣΕΩΝ

Σε αυτό το κομμάτι θα ακολουθήσει η περιγραφή των κλάσεων ανά module που έχουν υλοποιηθεί σε Java και οι οποίες περιγράφουν τη λειτουργία του αλγορίθμου.

4.3.1 Ανάλυση των υπηρεσιών

Ο κάθε κόμβος έχει δύο υπηρεσίες επικοινωνίας, που η κάθε μία είναι αφιερωμένη στο συγκεκριμένο τομέα. Η μία υπηρεσία προορίζεται μόνο για την επικοινωνία με όλους τους υπόλοιπους κόμβους του πλέγματος και η άλλη υπηρεσία προορίζεται αποκλειστικά για την επικοινωνία με το *ICAS*. Η επικοινωνία στη προσομοίωσή μας γίνεται πάνω από το πρωτόκολλο HTTP (Hyper Text Transfer Protocol) και τα μηνύματα που ανταλλάσσονται είναι αποκλειστικά σε μορφή JSON (JavaScript Object Notation).

Αυτό σημαίνει ότι έχουμε αρκετά μεγαλύτερο overhead επικοινωνίας, καθώς το μέγεθος των πεδίων των κατώτερων στρωμάτων του μοντέλου OSI είναι πολύ μεγαλύτερο από το μέγεθος των δεδομένων. Ωστόσο, όπως θα δούμε και παρακάτω, στα αποτελέσματα της προσομοίωσης, θεωρούμε ως overhead μονάχα το μέγεθος των επικεφαλίδων του πρωτοκόλλου HTTP (Http Headers) καθώς ένα πραγματικό σύστημα MANET δε θα υλοποιούταν στη κορυφή του πρωτοκόλλου HTTP αλλά σε ένα πιο ελαφρύ πρωτόκολλο με μικρότερο overhead επικοινωνίας.

Κάθε υπηρεσία, επομένως, είναι υπεύθυνη για την σύνδεση ενός κόμβου με μία άλλη οντότητα, είτε πρόκειται για κάποιον άλλο κόμβο είτε πρόκειται για το *ICAS*, μέσω του πρωτοκόλλου HTTP. Επομένως, η κάθε υπηρεσία έχει κάποιον HttpClient που υλοποιεί την πραγματική σύνδεση και έναν JSON μετατροπέα, ο οποίος μεταφράζει το πακέτο που θέλουμε να στείλουμε σε μία συμβολοσειρά με κατάλληλη μορφοποίηση σύμφωνα με τις JSON προδιαγραφές. Επιπλέον, προσθέτει την HTTP επικεφαλίδα

```
Content-Type: application/json; charset=utf-8
```

ώστε να ενημερώνει τον προορισμό ότι το σώμα του πακέτου HTTP είναι σε JSON μορφοποίηση. Κάθε υπηρεσία επικοινωνίας υλοποιείται στην κλάση **Communication Service**.

4.3.1.1 Υπηρεσία επικοινωνίας μεταξύ κόμβων

Κάθε κόμβος, κατά την αλληλεπίδρασή του με τους υπόλοιπους κόμβους, έχει πολύ συγκεκριμένες ενέργειες και λειτουργίες που επιτελεί. Αυτές μπορούν να συνοψιστούν ως εξής:

- Την ανακάλυψη γειτόνων, που πραγματοποιείται στο στάδιο εκκίνησης της προσομοίωσης

- Την ανταλλαγή των πινάκων δρομολόγησης που είναι μία διαδικασία που έπεται της ανακάλυψης γειτόνων με σκοπό την εύρεση διαδρομών για τους απομακρυσμένους κόμβους του πλέγματος
- Την ανταλλαγή πληροφοριών δρομολόγησης για έναν συγκεκριμένο κόμβο και την ανανέωση των πινάκων δρομολόγησης, η οποία είναι μία δυναμική διαδικασία κατά τη διάρκεια της προώθησης πακέτου ανάλογα με τους εγωιστές- κακόβουλους κόμβους του δικτύου.
- Την ανταλλαγή τιμών φήμης μεταξύ των γειτόνων στα πλαίσια της υλοποίησης του μηχανισμού φήμης *Darwin*
- Μία προαιρετική λειτουργία απενεργοποίησης ενός κόμβου όπου ενημερώνει τους γειτονικούς κόμβους για την απομάκρυνσή του από το δίκτυο
- Την πιο σημαντική λειτουργία αποστολής πακέτων πληροφορίας και *Ack*

Όλες αυτές οι λειτουργίες περιλαμβάνονται στην κλάση **NodeService** και είναι υπο-κλάση της **CommunicationService**.

Κάθε κόμβος τρέχει σε ένα ξεχωριστό thread στην προσομοίωση και διαθέτει το δικό του προγραμματιστή διαδικασιών (*Task Scheduler*) με μοναδική ευθύνη την αυτοματοποίηση των διαδικασιών.

Πρώτο task το οποίο καλείται να ολοκληρώσει είναι η ανακάλυψη γειτόνων, η οποία υλοποιείται στην κλάση **DiscoveryTask**. Εκεί καλείται η λειτουργία ανακάλυψης γειτόνων

```
nodeService.discoverNeighbors()
```

Το κάθε πακέτο εύρεσης γειτόνων υλοποιείται στην κλάση **DiscoveryPacket** και περιέχει διάφορες πληροφορίες του κόμβου αναζήτησης όπως είναι το *id* του και οι συντεταγμένες του (x, y). Η μορφή του πακέτου είναι

```
{
  nodeId: 6,
  sourceX: 75,
  sourceY: 107,
}
```

Σε ένα πραγματικό σύστημα αυτό το πακέτο θα ήταν ένα broadcast πακέτο το οποίο θα γινόταν αντιληπτό μόνο μέσα στην ακτίνα κάλυψης. Για την προσομοίωση της κατάστασης αυτής, το πακέτο εκπέμπεται σε όλους τους κόμβους, οι οποίοι υλοποιούν τον υπολογισμό της απόστασης, ανάλογα με τις δικές τους συντεταγμένες και εφόσον η απόσταση αυτή είναι μικρότερη από την ακτίνα, που έχουμε θεωρήσει εμείς ως αποδεκτή γειτνίαση, τότε απαντούν

με ένα καταφατικό μήνυμα στη πηγή του μηνύματος προσθέτοντάς τη στη λίστα γειτονικών κόμβων, ενώ στην αντίθετη περίπτωση με ένα αρνητικό μήνυμα. Ο υπολογισμός της απόστασης πραγματοποιείται στην κλάση **DistanceCalculator** ενώ η επαλήθευση της γειννίας υλοποιείται στην κλάση **NeighborValidator**.

Επόμενο task που εκπονείται είναι αυτό της ανταλλαγής πινάκων δρομολόγησης. Κατά τη διαδικασία αυτή ο κάθε κόμβος αποστέλλει τους πίνακες δρομολόγησης προς όλους τους γείτονές του ενώ μαθαίνει παράλληλα τους δικούς τους. Επαναλαμβάνεται δέκα φορές με ένα διάστημα παύσης ενός δευτερολέπτου και υλοποιείται στη διαδικασία

```
nodeService.exchangeRoutingTables()
```

με τη βοήθεια του πακέτου **RoutingPacket**. Το πακέτο αυτό περιλαμβάνει πληροφορίες του κόμβου, πληροφορίες (*distance*, *hops*) για τους γείτονες κόμβους και τους απομακρυσμένους κόμβους. Η μορφή του πακέτου είναι:

```
{
  id: 6,
  routingTable:
  [
    { id: 7,
      distance: 180,
      hops: 6
    },
    { id: 5,
      distance: 30,
      hops: 1
    }
  ]
}
```

Επόμενο task που λαμβάνει χώρα ανά τακτά χρονικά διαστήματα είναι η εκπόνηση του αλγορίθμου Darwin και περιγράφεται στην κλάση **DarwinUpdateTask** και η οποία καλεί τη διαδικασία

```
nodeService.exchangeDarwinInfo()
```

Το πακέτο **DarwinPacket** που ανταλλάσσεται μεταξύ των κόμβων έχει πολύ συγκεκριμένη δομή και περιέχει την ταυτότητα του κόμβου-πηγής, τον τρέχον γύρο του αλγορίθμου και πληροφορίες για τη φήμη που διαθέτει ο κόμβος για τους γείτονές του. Η μορφή του πακέτου είναι:

```

{
  id: 5,
  round: 6,
  neighborRatiolist:
  [
    { neighborId: 6,
      ratio: 0.91,
      pDarwinForNeighbor: 0.03,
      p: 0.042
    },
    { neighborId: 42,
      ratio: 0.32,
      pDarwinForNeighbor: 0.72,
      p: 0.54
    },
    { neighborId: 32,
      ratio: 0.49,
      pDarwinForNeighbor: 0.32,
      p: 0.32
    }
  ]
}

```

Μετά την ανταλλαγή των πληροφοριών τα πακέτα αποθηκεύονται στη μνήμη του κάθε κόμβου και καλείται ένα άλλο task όπου υπολογίζονται οι νέες τιμές φήμης στη διαδικασία:

```
node.executeDarwinAlgorithm()
```

Τέλος το ερώτημα (query) που κάνει ένας κόμβος σε γείτονά του για τη λήψη νέας διαδρομής δρομολόγησης για έναν απομακρυσμένο κόμβο καλείται στη διαδικασία

```
nodeService.exchangeRoutingInformationForNode(distantId)
```

και η απάντηση που λαμβάνει έχει την εξής μορφή

```

{  nodeId: 7,
   isFound: true,
   destinationId: 13,
   distance: 89,
   hops: 2 }

```

όπου το πεδίο `nodeId` δηλώνει τον κόμβο που απάντησε και το πεδίο `isFound` αν βρέθηκε πληροφορία για το `destination` κόμβο.

Στα πλαίσια της προσομοίωσης θεωρούμε ότι κάθε κόμβος τρέχει έναν διακομιστή διαδικτύου (web server) για τη δικτύωσή του με τις υπόλοιπες οντότητες. Αυτό ωστόσο δεν είναι εφικτό

αφού θα χρειαζόταν να στηθούν πενήντα μηχανές ή μία μηχανή όπου να τρέχει πενήντα web servers πράγμα που θα απαιτούσε πολλούς υπολογιστικούς πόρους. Επομένως στήθηκε ένας και μοναδικός web server όπου δρομολογούσε τα πακέτα στο αντίστοιχο thread του κόμβου προορισμού ανάλογα με μία επικεφαλίδα που εισήγαγε στις επικεφαλίδες του HTTP.

Η URL διεύθυνση στην οποία απηύθυναν τα HTTP αιτήματά τους οι κόμβοι με προορισμό άλλους κόμβους είναι η

`http://localhost:8081/{destinationNodeId}/{action}`

όπου `destinationNodeId` το *id* του κόμβου προορισμού, ακολουθούμενη από το αντίστοιχο ρήμα για την ενέργεια που θέλει να εκτελέσει.

Action Verb	Ενέργεια
Discovery	Ανακάλυψη γείτονα
Send	Αποστολή πακέτου δεδομένων
Darwin	Αποστολή πακέτου φήμης Darwin
RouteForNode/{dst}	Αναζήτηση πληροφορίας για νέα διαδρομή για τον κόμβο με <i>id = dst</i>
Routing	Ανταλλαγή πινάκων δρομολόγησης
SelfishBroadcast	Δημοσίευση λιστών εγωιστών κόμβων
Configure	Για εξωτερική παραμετροποίηση των κόμβων

Πίνακας 4.1 Πίνακας ρημάτων URL για το *node module*

Για παράδειγμα, για αποστολή πακέτου ανακάλυψης γειτόνων προς τον κόμβο 4 το HTTP URL θα είναι:

`http://localhost:8081/4/discovery`

Επιπλέον, αξίζει να σημειωθεί επίσης, ότι σε ένα ρεαλιστικό σενάριο δεν είναι δυνατό ένας κόμβος MANET συστήματος να τρέχει έναν web server, αφού απαιτεί αρκετούς υπολογιστικούς πόρους και παραπάνω τροφοδοσία, ενώ στην πλειοψηφία τους οι κόμβοι αποτελούν κινητές συσκευές μειωμένων χαρακτηριστικών που τους προσδίδουν αυτονομία και ευχρηστία π.χ. μικροαισθητήρες.

4.3.1.2 Υπηρεσία επικοινωνίας με το *ICAS*

Η επικοινωνία με το *ICAS* αποτελείται από τις εξής ενέργειες:

- Την εγγραφή του κόμβου στο *ICAS*, που έπεται της ανακάλυψης γειτόνων
- Την ενημέρωση του *ICAS* για τα στατιστικά και τις τιμές φήμης των γειτονικών κόμβων

- Την άδεια για την αποστολή πακέτου
- Την αποστολή των αποδείξεων επιτυχημένης αποστολής πακέτου
- Τη διαγραφή του κόμβου από το *ICAS*, σε περίπτωση απενεργοποίησής του

Οι λειτουργίες αυτές περιλαμβάνονται στην κλάση **IcasService** και είναι υποκλάση της **CommunicationService**.

Η εγγραφή στην ψηφιακή τράπεζα πραγματοποιείται με τη βοήθεια του **RegisterPacket** μέσα στη συνάρτηση

```
icasService.register()
```

Πέρα των βασικών πληροφοριών ο κόμβος αποστέλλει επίσης τον συνολικό αριθμό των γειτόνων του, σύμφωνα με τον οποίο το *ICAS* αποφασίζει αν πρόκειται για απομακρυσμένο κόμβο ή όχι. Η μορφή του πακέτου εγγραφής είναι η εξής:

```
{  
  id: 7,  
  x : 130,  
  y : 96,  
  neighbors: 3  
}
```

Η ενημέρωση του *ICAS* για τα στατιστικά είναι μία διαδικασία που επαναλαμβάνεται ανά μία συγκεκριμένη περίοδο και περιλαμβάνεται στην κλάση **UpdateBehaviorTask**, όπου καλείται

```
icasService.updateNeighborBehavior()
```

Η μορφή του πακέτου **BehaviorUpdate** μοιάζει αρκετά στο **DarwinPacket** και περιλαμβάνει επιπλέον πληροφορία για τον αριθμό πακέτων που δρομολογήθηκαν στο συγκεκριμένο γύρο. Η μορφή του είναι η εξής:

```

{
  id: 6,
  round: 7,
  relayedPackets: 14,
  totalNeighbors: 2,
  neighborList:
  [
    { neighId: 3,
      ratio: 0.82
    },
    { neighId: 15,
      ratio: 0.21
    }
  ]
}

```

Επόμενη λειτουργία είναι η μαζική ή μη αποστολή αποδείξεων στο *ICAS*. Η κάθε απόδειξη αποτελείται από ένα διάνυσμα όλων των κόμβων από τους οποίους έχει περάσει. Η μορφή του πακέτου είναι αρκετά απλή και δεν απαιτείται μία ξεχωριστή κλάση για να την περιγράψει.

```

[
  [4, 32, 15, 3, 2],
  [26, 29, 18, 2]
]

```

Στο συγκεκριμένο πακέτο παρατηρούμε τις δύο αποδείξεις που έλαβε ο κόμβος με $id = 2$. Ο αριθμός αποδείξεων ανά πακέτο είναι ένας αριθμός προς διερεύνηση στις προσομοιώσεις του επόμενου κεφαλαίου.

Η URL διεύθυνση στην οποία απηύθυναν τα HTTP αιτήματά τους οι κόμβοι με προορισμό το *ICAS* είναι η

`http://localhost:8080/{action}`

ακολουθούμενη από το αντίστοιχο ρήμα για την ενέργεια που θέλει να εκτελέσει.

<i>Action Verb</i>	<i>Ενέργεια</i>
Register	Εγγραφή στο ICAS
AskPermission	Άδεια για αποστολή πακέτου
DeliverySuccessful	Αποστολή αποδείξεων επιτυχημένης αποστολής πακέτου
UpdateNeighborBehavior	Ενημέρωση φήμης κόμβων

Πίνακας 4.2 Πίνακας ρημάτων URL για το *icas* module

Για παράδειγμα, για αποστολή πακέτου ανακάλυψης γειτόνων το HTTP URL θα είναι:

```
http://localhost:8080/register
```

Παρατηρούμε ότι τόσο ο web server για την εξυπηρέτηση επικοινωνία των κόμβων μεταξύ τους όσο και ο web server για την επικοινωνία με το ICAS τρέχουν στο ίδιο μηχάνημα αλλά σε διαφορετική πόρτα στο επίπεδο μεταφοράς δεδομένων (Transport Layer).

Το ICAS όπως και σε ένα ρεαλιστικό σενάριο έτσι και στην προσομοίωση τρέχει σε έναν ξεχωριστό και δικό του web server με τον οποίο έχουν επικοινωνία όλοι οι κόμβοι. Η θεώρηση ότι η επικοινωνία μαζί του πρέπει να γίνεται πάνω από HTTP πακέτα είναι σωστή, αφού στις περισσότερες περιπτώσεις δεν είναι απαραίτητο η ψηφιακή τράπεζα να βρίσκεται στον ίδιο φυσικό χώρο με τους κόμβους ώστε να χρησιμοποιούνται τεχνολογίες όπως BlueTooth ή Wifi. Επομένως θα είναι απαραίτητη η ύπαρξη ενός IP δικτύου, όπου η χρήση HTTP για μεταφορά δεδομένων είναι ενδεικνυόμενη.

4.3.2 Προσομοίωση κόμβων

Εδώ θα περιγραφούν οι κλάσεις που αφορούν το *node module*.

Node: αυτή η κλάση υλοποιεί τη πιο σημαντική οντότητα του συγκεκριμένου module, δηλαδή τον στοιχειώδη κόμβο και περιλαμβάνει κάποια βασικά στοιχεία και λειτουργίες που τον χαρακτηρίζουν. Κάθε κόμβος χαρακτηρίζεται από μία μοναδική ταυτότητα, η οποία είναι το *id* του καθώς και από τις μοναδικές του συντεταγμένες: (x, y) . Επιπλέον, διατηρεί μια λίστα με όλα τα *id* των κόμβων που καλείται να στείλει τα πακέτα και που για λόγους απλότητας θεωρούμε ότι είναι συγκεκριμένα καθ' όλη τη διάρκεια της προσομοίωσης και ορίζονται κατά την αρχικοποίηση του κόμβου. Παρέχει, επίσης, μία λίστα με όλους τους γειτονικούς κόμβους και μία λίστα με όλους τους απομακρυσμένους κόμβους και λίστες με όλους τους εγλωστές κόμβους. Τέλος, διατηρεί διάφορα στατιστικά που αφορούν τον συγκεκριμένο κόμβο αναφορικά με τα πακέτα που έστειλε επιτυχώς, έλαβε, απέρριψε και δρομολόγησε που χρησιμεύουν στην εξαγωγή μετρικών.

Neighbor: αυτή η κλάση περιγράφει το γείτονα ενός κόμβου. Ο λόγος ύπαρξης της κλάσης αυτής που ναι μεν αφορά έναν κόμβο αλλά δημιουργείται μία νέα κλάση για την ανεξαρτητοποίηση των κόμβων μεταξύ τους. Δηλαδή κάθε στιγμιότυπο *node* δεν έχει καμία γνώση για τα υπόλοιπα στιγμιότυπα και επομένως στις μεταβλητές που αυτά περιέχουν αλλά γνώση για το στιγμιότυπο της κλάσης *Neighbor*. Έτσι, παρέχεται ένα επιπλέον ασφάλειας μεταξύ των κόμβων. Η κλάση αυτή περιέχει πληροφορίες όπως το *id* του γείτονα κόμβου, τις συντεταγμένες του (x, y) , το λόγο συνδεσιμότητας cp_i και άλλες πληροφορίες όπως $p^i, p^{-i}, p_{DARWIN}^i, p_{DARWIN}^{-i}$ που είναι

απαραίτητες για την υλοποίηση του αλγορίθμου *Darwin* καθώς και διάφορες μετρητές στατιστικών.

Distant: αυτή η κλάση περιγράφει τους απομακρυσμένους κόμβους από έναν συγκεκριμένο κόμβο και περιλαμβάνει δηλαδή όσους δεν είναι γείτονες. Περιέχει πληροφορίες όπως το *id*, τα συνολικά βήματα που απαιτούνται για να προσεγγίσεις τον κόμβο αυτό από έναν αρχικό κόμβο, τη συνολική απόσταση και τον πρώτο ενδιαμέσο γείτονα κόμβο.

Link: η κλάση αυτή αναπαριστά τη ζεύξη σε μία τηλεπικοινωνιακή μετάδοση και χαρακτηρίζεται από τη μοναδική της ταυτότητα (*id*), τους κόμβους τους οποίους συνδέει και σαφώς το μήκος της, το οποίο ορίζει την καθυστέρηση διάδοσης. Για λόγους υλοποίησης διαθέτει τέσσερις ουρές *FIFO*, μία αποστολής και μία λήψης για κάθε κόμβο, οι οποίες αποθηκεύουν τα πακέτα που στέλνουν και λαμβάνουν τα άκρα της ζεύξης. Για να προσομοιώσουμε την καθυστέρηση διάδοσης, δημιουργούμε έναν μετρητή *timer*, ο οποίος ανά $t = \frac{u_c}{d}$, όπου u_c η ταχύτητα στο μέσο διάδοσης (αέρας) και d το μήκος της ζεύξης, και στην περίπτωση που υπάρχουν πακέτα σε μία εκ των δύο πακέτων αποστολής δίνει το σήμα αφαίρεσης πακέτου από την ουρά και αποστέλλεται το πακέτο στην ουρά λήψης του άλλου κόμβου. Κάθε ζεύξη είναι πεδίο της κλάσης **Neighbor** και μπορεί να προσπελαστεί μέσω της μεθόδου `neighbor.getLink()`. Η τοποθέτηση/ αφαίρεση των πακέτων από τις ουρές γίνεται μέσω των μεθόδων `link.addPacketToUpLink()`, `link.addPacketToDownLink()`, `link.removePacketFromDownLink()`.

Packet: η κλάση αυτή αναπαριστά το πακέτο πληροφορίας το οποίο αποστέλλεται από κόμβο σε κόμβο μέχρι τον τελικό προορισμό. Περιέχει στοιχεία όπως το χαρακτηριστικό του πακέτου (*id*), την πηγή (*src*), τον προορισμό (*dst*), ένα πεδίο *time – to – live (ttl)*, μία λίστα κόμβων που έχει ήδη περάσει (*pathlist*), ένα πεδίο *ack* που δηλώνει αν πρόκειται για πακέτο πληροφορίας ή όχι και τέλος η ουσιαστική πληροφορία που για εμάς ορίζεται σαν ένα *byte (data)*. Το πακέτο πληροφορίας μεταφέρεται πάνω στη ζεύξη ως το εξής μήνυμα:

```
{
  "id": 5,
  "src": 4,
  "dst": 3,
  "ttl": 6,
  "pathlist": [4,9,13,5,6],
  "ack": false,
  "data": 4
}
```

4.3.2.1 Αρχικοποίηση κόμβων

Κάθε κόμβος αρχικοποιείται με την έναρξη της προσομοίωσης όπου ορίζεται εξ' αρχής η μοναδική του ταυτότητα id , οι συντεταγμένες του (x, y) , οι τελικοί προορισμοί αποστολής πακέτων ενώ επιλέγεται αν πρόκειται για εγωιστή ή κακόβουλο κόμβο. Επιπλέον, αν ο κόμβος οριστεί ως κακόβουλος επιλέγεται και τι είδους ψευδείς πληροφορίες θα αρχίσει να διαδίδει. Οι ψευδείς πληροφορίες που μπορεί να διαδίδει μπορούν να διακριθούν είτε προς τους άλλους κόμβους είτε προς το *ICAS* και υλοποιούνται στις κλάσεις *DarwinLie* και *IcasLie* αντίστοιχα. Ο κώδικας αρχικοποίησης κόμβων παρατίθεται παρακάτω:

```
Node node = new Node(nodeId, NodePosition.x[nodeId], NodePosition.y[nodeId]);
node.setDestinationsIds(NodePosition.destinationNodes[nodeId % 13]);
node.setCheater(isCheater);
```

Ενώ αν πρόκειται για κακόβουλο κόμβο:

```
MaliciousNode maliciousNode = new MaliciousNode(nodeId,
                                                NodePosition.x[nodeId],
                                                NodePosition.y[nodeId]);
maliciousNode.tellsLie(new IcasLie())
               .tellsLie(new DarwinLie());
Node node = malNode;
```

Την αρχικοποίηση του κόμβου ακολουθεί η έναρξη λειτουργίας του, η οποία πραγματοποιείται με τη βοήθεια των *thread* που προσφέρει η Java. Για την δικιά μας ευκολία, δημιουργήσαμε την κλάση *NodeThread* που είναι υπο-κλάση της *Thread* της Java. Όπως αναφέρθηκε προηγουμένως επιλέχτηκε αυτή η τεχνική για να προσομοιώσουμε τα διαφορετικά περιβάλλοντα στα οποία υποτίθεται ότι τρέχουν οι διάφοροι κόμβοι του MANET συστήματός μας. Επιπλέον, θεωρήσαμε ότι κάθε κόμβος ανοίγει νέο *Thread* για την εύρεση και δρομολόγηση του νέου προορισμού και διαφορετικό *Thread* για την λήψη των πακέτων. Το μεν υλοποιείται στην κλάση *NodeRoutingThread*, του οποίου η λειτουργία ξεκινάει αφού έχουν υλοποιηθεί ορισμένα *tasks*, όπως περιγράφηκαν, και το δε είναι ενσωματωμένο στη λειτουργία του *web server*.

Η υλοποίηση του *web server* των κόμβων πραγματοποιείται με την ενσωμάτωση ενός ξεχωριστού εξωτερικού *module*, που ονομάζεται *Spring-Boot*, το οποίο είναι ευρέως διαδομένο εξαιτίας της ευκολίας ανάπτυξης και διαχείρισης μεγάλων συστημάτων λογισμικού με ιδιαίτερα οφέλη στην δημιουργία εφαρμογών διαδικτύου.

4.3.2.2 Αποστολή πακέτων δεδομένων

Η διαδικασία αποστολής πακέτου είναι η εξής. Κάθε μία σταθερή περίοδο δημιουργείται ένα αίτημα αποστολής πακέτου δεδομένων το οποίο αίτημα καταχωρείται σε μία ουρά αναμονής στο thread του κόμβου. Εάν δεν έχει καθυστερήσει η προώθηση του προηγούμενου πακέτου, τότε πραγματοποιείται ένα interrupt στο thread και γίνεται προσπάθεια να ικανοποιηθεί το νέο αίτημα που αφαιρείται από την ουρά. Εφόσον επεξεργαστεί το αίτημα, δημιουργείται το πακέτο δεδομένων ανάλογα με το προορισμό που έχει τεθεί και γίνεται προσπάθεια να βρεθεί το επόμενο βήμα- γείτονας που θα δρομολογήσει το πακέτο.

Επίσης στην ουρά αιτημάτων τοποθετείται και κάθε νέο πακέτο που καταφθάνει στον κόμβο μέσω των downlink ουρών των ζεύξεων.

Η διαδικασία πραγματοποιείται στη συνάρτηση `nodeRoutingThread.managePacket(Packet)`. Εκεί πέρα το πρώτο πράγμα που τσεκάρεται είναι αν πρόκειται για πακέτο *Ack*. Αρχικά θα αναλύσουμε την περίπτωση όπου πρόκειται για πακέτο δεδομένων. Οι συνθήκες μη απόρριψης ενός πακέτου δεδομένων είναι:

1. Ο συγκεκριμένος κόμβος να μην είναι εγωιστής
2. Να μην προέρχεται το πακέτο από κόμβο που βρίσκεται στη λίστα των εγωιστών
3. Εάν η τιμή *TTL* του πακέτου να μην έχει φτάσει στο 0
4. Ο συγκεκριμένος κόμβος να μην είναι κακόβουλος

Εφόσον ικανοποιηθούν και οι τέσσερις αυτές συνθήκες ταυτοχρόνως, τότε ο κόμβος αναζητά τον επόμενο κόμβο προώθησης. Αυτή η αναζήτηση γίνεται στην κλάση **Router** και πιο συγκεκριμένα στην μέθοδο `routePacket`. Η επεξήγηση της κλάσης αυτής θα γίνει αργότερα. Εφόσον βρεθεί νέος προορισμός, τότε ανανεώνονται ορισμένα στατιστικά μέσω της μεθόδου:

```
neighborStatsRecorder.recordPacket(Packet)
```

Αναλόγως με τον νέο προορισμό τοποθετείται στην uplink ουρά της αντίστοιχης ζεύξης σύνδεσης και εκπέμπεται προς το νέο προορισμό.

Η διαδικασία δρομολόγησης πακέτου μέσα στη μέθοδο `router.routePacket(Packet)` ακολουθεί τον αλγόριθμο που είχε εξηγηθεί στο προηγούμενο κεφάλαιο.

Εάν το πακέτο δεν είναι πακέτο *Ack*, τότε μειώνεται το πεδίο *TTL* μέσω της μεθόδου `packet.decrementTTL()` και αναζητείται ο επόμενος γείτονας. Εφόσον βρεθεί, προστίθεται στη λίστα πρόσβασης του πακέτου μέσω της μεθόδου `packet.addToPathList(neighborId)`. Εάν πρόκειται για πακέτο *Ack*, τότε ο επόμενος κόμβος μπορεί να βρεθεί μέσω της λίστας των

κόμβων που έχει ήδη δημιουργηθεί κατά τη διάρκεια αποστολής του πακέτου και πραγματοποιείται στο block κώδικα.

```
int nextNeighborIndex = packet.getPathList().indexOf(node.getId()) - 1;
long nextNeighborId = packet.getPathList().get(nextNeighborIndex);
Optional<Neighbor> nextNeighbor = node.findNeighborById(nextNeighborId);
```

Επιπλέον, για την απόρριψη ή μη των πακέτων επιβεβαίωσης από τους εγωιστές κόμβους εφαρμόζεται μία συγκεκριμένη πολιτική κάθε φορά την οποία ο εγωιστής κόμβος ανάλογα με την κατάστασή του ακολουθεί ή όχι. Η πολιτική αυτή περιγράφεται στην κλάση **DropAckPolicy**. Εφόσον την ακολουθεί, τότε απορρίπτει τα πακέτα σύμφωνα με μία πιθανότητα που ορίζεται από τη συγκεκριμένη πολιτική ενώ μπορεί να επιλέξει να μην την ακολουθεί αν αρχίσει να αντιλαμβάνεται ότι τα πακέτα του έχουν σταματήσει να προωθούνται από τους υπόλοιπους κόμβους. Η πολιτική αυτή μπορεί να περιγραφεί από τον εξής κώδικα:

```
public boolean drop(Packet p) {
    if (!isFollowingCurrentPolicy) {
        return false;
    } else if (node.isCheater()) {
        return new Random().nextDouble() <= 0.0001f;
    }
    return false;
}
```

Για την καταγραφή επίσης των στατιστικών γίνεται η αντίστοιχη διαδικασία ανεύρεσης του επόμενου κόμβου μέσω της λίστας των ήδη προσπελασμένων κόμβων του πακέτου. Εφόσον βρεθεί, ανανεώνονται οι μετρητές του αντίστοιχου γείτονα σύμφωνα με το παραπάνω κώδικα:

```
int nextNeighborIndex = packet.getPathList().indexOf(node.getId())-1;
long nextNeighborId = packet.getPathList().get(nextNeighborIndex);
Neighbor nextNeighbor = node.findNeighborById(nextNeighborId).get();
if (!packet.isAck()){
    if (neighbor.getId() != packet.getDestinationNodeId()){
        neighbor.incrementSentPacketCounter();
    }
    if (packet.getSourceNodeId != node.getId()){
        node.incrementRelayedPacketCounter();
    } else node.incrementSentPacketCounter();
} else {
    if (packet.getSourceNodeId() == node.getId()){
        node.incrementForwardedPacketCounter();
    }
} else {
    if (packet.getSourceNodeID() == node.getId()) {
        node.incrementForwardedPacketCounter();
    }
}
```

```

    }
}

```

4.3.2.3 Υλοποίηση αλγορίθμου Darwin

Κάθε κόμβος στα πακέτα Darwin που αποστέλλει υπολογίζει την τιμή cp_{ij} για κάθε γείτονά του, η οποία τιμή υπολογίζεται κάθε φορά που ανανεώνονται τα στατιστικά για τη λήψη ή μη ενός πακέτου μέσα στην κλάση **Neighbor**

```

private void updateConnectivityRatio() {
    if (packetsSent.get() == 0) connectivityRatio = 1.0f;
    else connectivityRatio = (1.0f * packetsForwarded.get() / packetsSent.get());
}

```

Η κλάση Darwin είναι υπεύθυνη για τους υπολογισμούς των νέων τιμών φήμης που διαθέτει ένας κόμβος για τους γείτονές του. Εφόσον συλλέξει όλα τα πακέτα από τους γείτονες σε κάθε γύρο καλείται η συνάρτηση `Darwin.computeDarwin(List<DarwinPacket>)`. Αρχικά υπολογίζει την τιμή $p_i^{(k)}$ για κάθε γείτονα μέσα στη μέθοδο `calculatePMinusI()` και υπολογίζει την τιμή $p_i^{(k)}$, η οποία είναι η τιμή φήμης που αντιλαμβάνονται οι γείτονές του για τον ίδιο τον κόμβο. Ο ακόλουθος κώδικας αποφασίζει για τον ορισμό ενός γείτονα ως εγωιστή ή όχι:

```

for (Neighbor neighbor: node.getNeighbors()){
    p_MinusI = neighbor.getPMinusI();
    p_DarwinI = neighbor.getDarwinI();
    p_DarwinMinusI = neighbor.getDarwinMinusI();
    q_I = DarwinUtils.normalizeValue(p_I - p_DarwinI);
    q_MinusI = DarwinUtils.normalizeValue(p_MinusI - p_DarwinMinusI);
    p_NewDarwinI = DarwinUtils.normalizeValue(gamma * (q_MinusI - q_I));
    neighbor.setDarwinI(p_NewDarwinI);
    DarwinUtils.DecideDarwinSelfishNode(node,neighbor);
}

```

όπου `DarwinUtils` είναι μία βοηθητική κλάση που παρέχει λειτουργικότητες που απαιτούνται για τον αλγόριθμο. Συγκεκριμένα παρέχονται οι συναρτήσεις:

```

public static double normalizeValue(double v){
    if (v>=1) return 1.0f;
    else if (v<=0) return 0.0f;
    else return v;
}

```

που υλοποιεί τη συνάρτηση $[x]_0^1$ και η συνάρτηση

```

public static void DecideDarwinSelfishNode(Node node, Neighbor neighbor) {
    if (neighbor.getDarwinI() >= SimulationParameters.EDP) {

```

```

        node.addDarwinSelfishNode(neighbor.getId());
    } else {
        node.removeDarwinSelfishNode(neighbor.getId());
    }
}

```

όπου ανάλογα με την τιμή κατωφλίου αποφασίζει αν ο συγκεκριμένος γείτονας είναι εγωιστής ή όχι.

4.3.2.4 Σταθερές του Node Module

Οι διάφορες τιμές συντεταγμένων των κόμβων του δικτύου βρίσκονται μέσα στην κλάση **NodePosition** όπου παρατίθενται οι ταυτότητες των κόμβων που θα πρέπει να θεωρούνται εγωιστές ή κακόβουλοι.

Επιπλέον, στην κλάση **NodeScheduledTask** υπάρχουν επιπλέον σταθερές που αφορούν τις περιόδους και τις καθυστερήσεις των διαφόρων task που απαιτούνται για τη λειτουργία του κάθε κόμβου. Αυτές οι τιμές είναι οι εξής και ορισμένες τίθενται προς διερεύνηση στις προσομοιώσεις του επόμενου κεφαλαίου:

Μεταβλητή	Τιμή(ms)	Περιγραφή
<i>DISCOVERY_PERIOD</i>	100000	Περίοδος ανακάλυψης γειτόνων
<i>DISCOVERY_INITIAL_DELAY</i>	0	Αρχική καθυστέρηση ανακάλυψης γειτόνων
<i>DARWIN_UPDATE_PERIOD</i>	60000	Περίοδος αποστολής πακέτων Darwin
<i>ICAS_UPDATE_PERIOD</i>	10000	Περίοδος αποστολής πακέτων φήμης στο ICAS
<i>PACKET_SENT_PERIOD</i>	2000	Περίοδος αποστολής πακέτων δεδομένων
<i>PACKET_SENT_INITIAL_DELAY</i>	500	Αρχική καθυστέρηση αποστολής πακέτων δεδομένων

Πίνακας 4.3 Πίνακας περιόδων και καθυστερήσεων του node module

4.3.3 Προσομοίωση ICAS

Η λειτουργία του *ICAS* ανήκει στο *Icas module*. Το *ICAS* δεν έχει αρχικά κάποια αρχικοποίηση, απλώς αναμένει τους κόμβους να αρχίσουν να εγγράφονται σε αυτό, έτσι ώστε να ξεκινήσει τις ενέργειές του.

Η βασική οντότητα που δημιουργείται κάθε φορά που εγγράφεται ένας νέος κόμβος είναι η **NodeEntity**, που περιέχει την ταυτότητά του *id*, τις συντεταγμένες του (*x, y*), τις πιστωτικές μονάδες του (*tokens*), το ποσοστό συνδεσιμότητάς του, την τρέχουσα κατάστασή του, τα πακέτα που έχει δρομολογήσει και τους συνολικούς του γείτονες.

Η κατάσταση του κόμβου, όπως περιεγράφηκε στο τρίτο κεφάλαιο, μπορεί να πάρει μία από τρεις διαφορετικές διακριτές τιμές. Η κατάσταση υλοποιείται στην ξεχωριστή κλάση **NodeStatus** και μπορεί να πάρει μία μονάχα από τιμές:

ANY_SEND, NEIGHBOR_SEND, NO_SEND

Επομένως, κάθε φορά που ένας κόμβος ζητάει άδεια για την αποστολή ενός πακέτου, το μόνο που ελέγχεται είναι η τρέχουσα κατάστασή του:

```
switch (node.getStatus()){
    case ANY_SEND:
        allow();
        break;
    case NEIGHBOR_SEND:
        if (areNeighbors) allow(); else deny();
        break;
    case NO_SEND:
    default:
        deny();
}
```

Τα tasks που καλείται να εκτελεί περιοδικά η ψηφιακή τράπεζα είναι τα εξής:

- Η εύρεση και δημοσίευση στους κόμβους της λίστας με τους εγωιστές και κακόβουλους κόμβους

```
Set<Long> selfishList = new HashSet<Long>();
Set<Long> maliciousList = new HashSet<Long>();

NodeEntity.NodeEntityList.stream()
    .filter(NodeEntity::isSelfish)
    .forEach(node -> selfishList.add(node.getId()));

NodeEntity.NodeEntityList.stream()
    .filter(NodeEntity::isMalicious)
    .forEach(node -> maliciousList.add(node.getId()));

publish(selfishList, maliciousList);
```

- Τον υπολογισμό της νέας κατάστασης (status) των κόμβων

Ο υπολογισμός της νέας κατάστασης των κόμβων υλοποιείται με το παρακάτω block κώδικα μέσα στην κλάση **NodeEntity**

```
public void updateNodeStatus() {
    if (tokens >= SimulationParameters.CREDIT_STATUS_THRESHOLD) {
        nodeStatus = NodeStatus.ANY_SEND;
    }
}
```

```

    } else if (tokens >= 0) {
        nodeStatus = NodeStatus.NEIGHBOR_SEND;
    } else nodeStatus = NodeStatus.NO_SEND;
}

```

Ενώ ο υπολογισμός του λόγου συνδεσιμότητας του κάθε κόμβου υπολογίζεται στο block κώδικα

```

public void updateConnectivityRatioIfNecessary() {
    if (neighborConnectivityRatio.size() == totalNeighbors) {
        double sum = 0.0f;
        int totalNonSelfishNeighbors = 0;
        for(Entry<Long, Double> neighborEntry : neighborConnectivityRatio.entrySet()){
            long neighborId = neighborEntry.getKey();

            // take into account only the non-selfish nodes
            if (!NodeEntity.GetNodeEntityById(neighborId).get().isSelfish()) {
                totalNonSelfishNeighbors++;
                sum += neighborEntry.getValue();
            }
        }
        if (totalNonSelfishNeighbors > 0) {
            nodeConnectivityRatio = sum / totalNonSelfishNeighbors;
        }
        isSelfish = nodeConnectivityRatio < 0.5f;
    }
}

```

- Την επιδότηση των απομακρυσμένων κόμβων

Η επιδότηση των απομακρυσμένων κόμβων είναι μία περιοδική διαδικασία, με μάλιστα αρκετά μικρή περίοδο. Παρατηρήθηκε ότι οι κόμβοι αυτοί τείνουν να ξεμένουν από credits αρκετά νωρίς στην προσομοίωση οπότε χρειάζονται σταδιακή ανανέωση των πιστωτικών μονάδων τους. Επιπλέον, κόμβοι που δεν συμμετέχουν στις βέλτιστες διαδρομές παρότι μη απομακρυσμένοι χρήζουν περαιτέρω υποστήριξης. Οπότε κριτήριο για την επιδότηση κόμβων δεν είναι πλέον ο αριθμός γειτόνων αλλά ο αριθμός των πακέτων που έχουν δρομολογήσει σε μία συγκεκριμένη χρονική περίοδο και υλοποιείται με τον παρακάτω κώδικα στην κλάση **DistantNodeAssist**.

```

double cpi = node.getNodeConnectivityRatio();
int me_th = (int) (SimulationParameters.CREDITS_INITIAL * Math.pow(cpi, 2));
if (!node.isSelfish() && node.getTokens() <= me_th
&& node.getRelayedPackets() <= k * SimulationParameters.DISTANT_NODES_THRESHOLD_SECOND * cpi
&& !node.isAllowedToSendPacketsForFree()) {
    int additionalTokens = 0;
    float sintelestis;
    if (node.getRelayedPackets() <= k *
        SimulationParameters.DISTANT_NODES_THRESHOLD_FIRST * cpi) {

```

```

        sintelestis = 1.5f;
    } else {
        sintelestis = 2.0f;
    }
    additionalTokens = (int) (sintelestis +
        SimulationParameters.CREDITS_EXTRA * Math.pow(cpi, 2));
    node.setTokens(node.getTokens() + additionalTokens);
}
boolean isAllowedToSendFree = !node.isSelfish() &&
    node.getStatus() != NodeStatus.ANY_SEND &&
node.getRelayedPackets() <= k * SimulationParameters.DISTANT_NODES_THRESOLD_SECOND * cpi &&
node.getTokens() < 0;
node.setAllowedToSendPacketsForFree(isAllowedToSendFree);

```

- Τον καταμερισμό των credits από τον κόμβο- πηγή στους κόμβους- δρομολογητές

Αυτή η διαδικασία υλοποιείται στην κλάση **Rewarder** όπου για κάθε receipt που λαμβάνει το *ICAS*, καλούνται με τη σειρά η παρακάτω διαδικασία ανταμοιβής κάθε δρομολογητή

```

public int rewardNode(NodeEntity node) {
    int initialTokens = node.getTokens();
    int reward = 0;
    if (initialTokens <= TOKEN_FIRST_THRESHOLD) {
        reward = (int) (0.5 + 2.5 * Math.pow(node.getNodeConnectivityRatio(), 2));
    } else if (initialTokens <= TOKEN_SECOND_THRESHOLD) {
        reward = (int) (0.5 + 0.5 * Math.pow(node.getNodeConnectivityRatio(), 2));
    } else {
        reward = (int) (0.01 * (TOKEN_MAX - initialTokens));
    }
    node.setTokens(initialTokens + reward);
    return reward;
}

```

Και αθροίζοντας τις ανταμοιβές χρεώνεται η πηγή σύμφωνα με τη διαδικασία

```

public void chargeNode(NodeEntity node, int totalRelayCost) {
    if (!node.isAllowedToSendPacketsForFree()) {
        int initialTokens = node.getTokens();
        if (initialTokens > SimulationParameters.CREDITS_INITIAL * 5.7f) {
            node.setTokens(initialTokens - (int) (2.6 * totalRelayCost));
        } else if (initialTokens > SimulationParameters.CREDITS_INITIAL * 2.5f) {
            node.setTokens(initialTokens - (int) (1.3 * totalRelayCost));
        } else node.setTokens(initialTokens - totalRelayCost);
    }
}

```

4.3.3.1 Σταθερές του Icas Module

Μέσα στην κλάση **IcasConstants** υπάρχουν ορισμένες σταθερές που ορίζουν την περίοδο και την αρχική καθυστέρηση των διεργασιών που περιεγραφήκαν παραπάνω, είναι ενδεικτικές και τίθενται προς διερεύνηση στις προσομοιώσεις του επόμενου κεφαλαίου.

Μεταβλητή	Τιμή(ms)	Περιγραφή
<i>SELFISH_UPDATE_INITIAL_DELAY</i>	10000	Αρχική καθυστέρηση αποστολής λίστας εγμιστών
<i>SELFISH_UPDATE_PERIOD</i>	20000	Περίοδος αποστολής λίστας εγμιστών
<i>NODE_UPDATE_INITIAL_DELAY</i>	10000	Αρχική καθυστέρηση ενημέρωσης κατάστασης κόμβων
<i>NODE_UPDATE_PERIOD</i>	20000	Περίοδος ενημέρωσης κατάστασης κόμβων
<i>DISTANT_ASSIST_INITIAL_DELAY</i>	3000	Αρχική καθυστέρηση παροχής βοήθειας απομακρυσμένων κόμβων
<i>DISTANT_ASSIST_PERIOD</i>	3000	Περίοδος παροχής βοήθειας απομακρυσμένων κόμβων

Πίνακας 4.4 Πίνακας περιόδων και καθυστερήσεων για το *icas module*

4.3.4 Εκτύπωση στατιστικών

Για την επαλήθευση της απόδοσης του αλγορίθμου, την εξαγωγή μετρικών και τη σύγκριση με το αλγόριθμο του Darwin και στα δύο modules υπήρχαν περιοδικές διεργασίες μοναδική ευθύνη των οποίων είναι η εκτύπωση τιμών είτε στην console της προσομοίωσης είτε σε ξεχωριστά αρχεία.

Στο node module η εκτύπωση των στατιστικών υλοποιείται στην κλάση **StatsTask**. Η κλάση αυτή καλείται κάθε δέκα δευτερόλεπτα και αποθηκεύει σε αρχεία τις εξής τιμές:

1. Ποσοστό προώθησης συνεργατικών κόμβων
2. Ποσοστό προώθησης εγμιστών κόμβων
3. Ποσοστό προώθησης απομακρυσμένων κόμβων
4. Συνολικά bytes για την αποστολή πακέτων σηματοδοσίας. Τα πακέτα σηματοδοσίας αποτελούν όλα τα πακέτα εξαιρουμένων των πακέτων δεδομένων και επιβεβαίωσης.

Το ποσοστό προώθησης της ομάδας κόμβων N ορίζεται ως το άθροισμα όλων των πακέτων για τα οποία έχουν λάβει επιβεβαίωση οι κόμβοι σε μία χρονικά περίοδο k προς το άθροισμα όλων των πακέτων που έχουν αποστείλει

$$cp_N^{(k)} = \frac{\sum_{j \in N} F^{(k)}}{\sum_{j \in N} S^{(k)}}$$

Στο icas module η εκτύπωση των στατιστικών υλοποιείται στην κλάση **PrintToken** η οποία εκτυπώνει κάθε δέκα δευτερόλεπτα το μέσο αριθμό token που διαθέτουν οι εγωιστές και συνεργατικοί κόμβοι. Ο λόγος που καταγράφεται ο αριθμός token και των συνεργατικών κόμβων είναι ο έλεγχος του υπέρμετρου πλουτισμού τους, ο οποίος θα μπορούσε στη συνέχεια να τους οδηγήσει σε εγωιστική συμπεριφορά και υποβάθμιση του δικτύου.

4.3.5 Κοινές παράμετροι και σταθερές

Ορισμένες παράμετροι του συστήματος αφορούν και τα δύο modules και γι' αυτό έχουν τοποθετηθεί στο κοινό domain module. Παρακάτω παρατίθεται οι σταθερές που βρίσκονται στην κλάση **SimulationParameters**.

<i>Μεταβλητή</i>	<i>Τιμή</i>	<i>Περιγραφή</i>
<i>CREDITS_INITIAL</i>	220	Αρχικός αριθμός credits ανά κόμβο
<i>DISTANT_NODES_THRESHOLD_FIRST</i>	1.1	1 ^ο Κατώφλι για τη βοήθεια απομακρυσμένων
<i>DISTANT_NODES_THRESHOLD_SECOND</i>	4	2 ^ο Κατώφλι για τη βοήθεια απομακρυσμένων
<i>CREDIT_STATUS_THRESHOLD</i>	4	Κατώφλι credit για τη κατάσταση κόμβου
<i>CON_RATIO_ICAS_THRESHOLD</i>	0.5	Ποσοστό συνδεσιμότητας για εγωιστές κόμβους
<i>IFN</i>	5	Αριθμός χαμένων πακέτων για μετάνοια κόμβου
<i>EDP</i>	0.85	Κατώφλι πιθανότητας απόρριψης πακέτου
<i>CREDITS_EXTRA</i>	28	Επιπλέον βοήθεια απομακρυσμένων κόμβων
<i>MAX_HOPS</i>	15	Μέγιστα βήματα ενός πακέτου δεδομένων
<i>MAX_NEIGHBOR_DISTANCE</i>	60	Μέγιστη απόσταση γείτονα
<i>CP_THRESHOLD</i>	0.4	Κατώφλι για εξαγωγή εγωιστών κόμβων
<i>TOTAL_CHEATING_NODES</i>	3	Αριθμός εγωιστών κόμβων
<i>CHEATING_PROBABILITY</i>	1.0	Πιθανότητα εμφάνισης εγωιστικής συμπεριφοράς
<i>PACKET_HEADER_BYTES</i>	60	Επικεφαλίδα πακέτων δεδομένων

Table 4.5 Πίνακας κοινών μεταβλητών icas και node module

4.3.6 Εφαρμογή Android για την εκτύπωση στατιστικών

Στα πλαίσια της διπλωματικής αναπτύχθηκε εφαρμογή σε λειτουργικό σύστημα Android, το οποίο βοήθησε στην ενημέρωση στατιστικών ορισμένων κόμβων, καθώς και στην παραμετροποίησή τους κατά τη διάρκεια της προσομοίωσης.

Στόχος της εφαρμογής ήταν να προσφέρει στον ερευνητή έγκυρα στατιστικά για διάφορες παραμέτρους του κόμβου και να ερευνήσει τη συμπεριφορά του συστήματος- δικτύου σε αλλαγές που αφορούν τη συμπεριφορά του κόμβου και την ταχύτητα απόκρισης σε αλλαγές της ενεργητικότητάς του.

Πιο συγκεκριμένα, προσφέρονται στατιστικά για τις εξής κατηγορίες:

- Πακέτα που εστάλησαν αλλά δεν έχει επιβεβαιωθεί η επιτυχημένη αποστολή τους
- Πακέτα που επιτυχώς εστάλησαν στον προορισμό τους και έχει ληφθεί επιβεβαίωση
- Πακέτα που ελήφθησαν στο συγκεκριμένο κόμβο
- Πακέτα που δρομολόγησε ο κόμβος
- Πακέτα που ο κόμβος αναγκάστηκε ή ηθελημένα απέρριψε


Επιπλέον παρέχονται πληροφορίες σχετικά με τους γείτονές του (απόσταση, θέση, πακέτα που εστάλησαν, ποσοστό συνδεσιμότητας) καθώς και για τους απομακρυσμένους κόμβους (συνολική απόσταση, βήματα που απαιτούνται για την προσέγγισή του και ενδιαμέσος κόμβος-γείτονας).

4.3.6.1 Εκτέλεση εφαρμογής

Για την εκτέλεση της εφαρμογής, αρκεί η χρήση του προγράμματος **Android Studio** που χρησιμοποιήθηκε για την ανάπτυξη της και ενός απλού κινητού android smartphone. Αρχικά θα χρειαστεί η εισαγωγή του υπάρχων κώδικα σε ένα νέο project μέσω των επομένων βημάτων:

File > New > Import Project > AndroidManifest.xml of your android project > OK

Για την εκτέλεση της εφαρμογής σε μία πραγματική συσκευή:

- Σύνδεση της συσκευής με το PC μέσω ενός απλού USB καλωδίου
- Ενεργοποίηση της επιλογής USB Debugging στη συσκευή
- Στο Android Studio: app module > Project > Run (ή επιλογή **Run**  στη γραμμή εργαλείων)
- Στο παράθυρο με τίτλο Select Deployment Target επιλέγουμε τη συσκευή μας και πατάμε OK.

4.3.6.2 Περιήγηση στις οθόνες της εφαρμογής

Στην έναρξη της εφαρμογής, υπάρχει η αρχική οθόνη όπου διερωτάται ο ερευνητής να εισάγει το *URL*, από το οποίο η εφαρμογή μέσω του πρωτοκόλλου HTTP θα αντλεί τα στατιστικά και θα εισάγει νέες παραμέτρους στο σύστημα.

Η εφαρμογή, στην περίπτωση που δεν είναι έγκυρο το *URL*, αρνείται στο χρήστη την πλοήγηση στην οθόνη των στατιστικών και τον ξαναρωτάει για μία έγκυρη διεύθυνση.

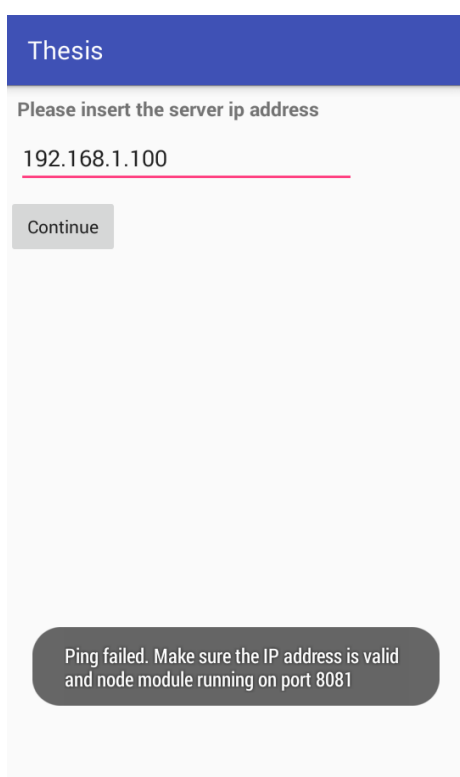
Η επαλήθευση της εγκυρότητας του *URL* γίνεται μέσω της διαδικασίας ελέγχου διαθεσιμότητας της συγκεκριμένης διεύθυνσης (ping). Αν το ping δεν επιστρέψει απάντηση, σημαίνει λάθος στην

πληκτρολόγηση της διεύθυνσης ή σφάλμα συνδεσιμότητας στη συγκεκριμένη πόρτα του στρώματος μεταφοράς (TCP).

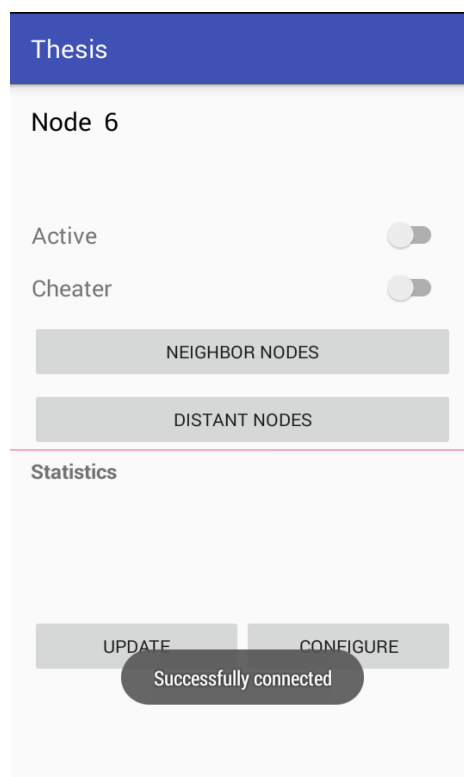
Για τη διευκόλυνση της διαδικασίας ανάπτυξης του software, έχουμε ορίσει ως έγκυρο *URL* το *URL* του web server στον οποίο τρέχει το node module δηλαδή το

<http://192.168.1.104:8081/>

Θεωρούμε ότι το node module τρέχει σε web server με διεύθυνση 192.168.1.104 οπότε λογικά στη διεύθυνση 192.168.1.100 εμφανίζει το εξής σφάλμα στα αριστερά ενώ στα δεξιά εμφανίζεται η οθόνη επιτυχημένης σύνδεσης:



Εικόνα 4.1: Αποτυχημένη σύνδεση με τον web server



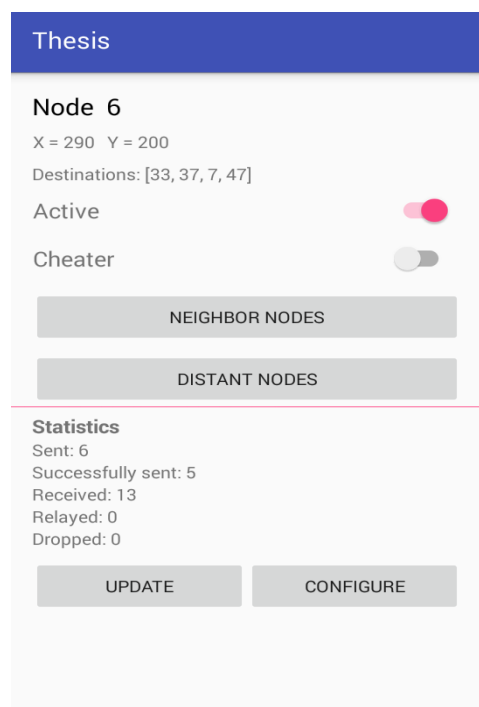
Εικόνα 4.2: Επιτυχημένη σύνδεση με τον web server

Στη συνέχεια μπορεί να δει κάποιος τη βασική οθόνη πλοήγησης με τα στατιστικά του κόμβου που διερευνούμε. Για τη διευκόλυνση σύγκρισης των αποτελεσμάτων μεταξύ τους ο κόμβος τον οποίο ερευνούμε είναι συνεχώς ο κόμβος με $id = 6$.

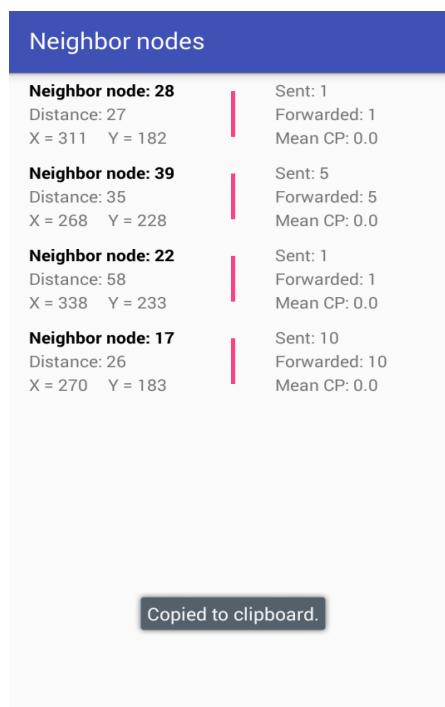
Υπάρχουν διακόπτες με τους οποίους μπορούμε να ενεργοποιήσουμε/ απενεργοποιήσουμε τον συγκεκριμένο κόμβο είτε να τον θέσουμε εγωιστή ή μη. Επιπλέον, υπάρχουν κουμπιά για την εμφάνιση στατιστικών των υπόλοιπων κόμβων.

Κάθε φορά που επιλέγεται η επιλογή UPDATE τότε ανανεώνονται όλα τα στατιστικά, ενώ με την επιλογή CONFIGURE ρυθμίζεται η νέα παραμετροποίηση του κόμβου. Τα πακέτα που ανταλλάσσονται μεταξύ της εφαρμογής Android και του web server περιγράφονται στην κλάση **NodeInfo** και περιέχει όλες τις απαραίτητες πληροφορίες για τη λειτουργία της εφαρμογής.

Παρακάτω παρατίθενται διάφορες οθόνες της εφαρμογής:



Εικόνα 4.3: Ενημέρωση στατιστικών κόμβου 6



Εικόνα 4.4: Εμφάνιση στατιστικών γειτονικών κόμβων

Distant nodes
Relay neighbor: 17
Distant Node: 5
Total hops: 2
Distance: 75
Relay neighbor: 39
Distant Node: 9
Total hops: 2
Distance: 106
Relay neighbor: 22
Distant Node: 14
Total hops: 2
Distance: 109
Relay neighbor: 22
Distant Node: 19
Total hops: 2
Distance: 75
Relay neighbor: 39
Distant Node: 1
Total hops: 3
Distance: 102
Relay neighbor: 39
Distant Node: 7
Total hops: 3
Distance: 106

Εικόνα 4.5: Εμφάνιση στατιστικών απομακρυσμένων κόμβων

5 ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΣΟΜΟΙΩΣΗΣ

5.1 ΠΑΡΑΜΕΤΡΟΙ ΠΡΟΣΟΜΟΙΩΣΗΣ

Προκειμένου να συγκριθεί η απόδοση του ICARUS 2 σε σχέση με αυτή του DARWIN πραγματοποιήθηκε μία σειρά προσομοιώσεων. Το σενάριο προσομοίωσης περιλαμβάνει 50 κόμβους τυχαία τοποθετημένους σε μία τοπολογία πλέγματος $400 \times 400 m^2$, από τους οποίους το 6% είναι εγωιστές και το 2% κακόβουλοι. Περίπου το 30% των κόμβων θεωρούνται απομακρυσμένοι, ενώ υποθέτουμε ότι κάθε κόμβος μπορεί να επικοινωνήσει απευθείας με όσους κόμβους βρίσκονται σε ακτίνα $60m$ (και συνεπώς τους θεωρεί γείτονές του). Η πιθανότητα να χαθεί ένα πακέτο δεδομένων ή επιβεβαίωσης είναι 0.2% και 0.02% αντίστοιχα. Η πιθανότητα απόρριψης πακέτων επιβεβαίωσης από τους εγωιστές κόμβους ορίζεται 0.01%. Η συνολική διάρκεια της προσομοίωσης είναι 350 δευτερόλεπτα.

Παρακάτω παρατίθενται οι παράμετροι προσομοίωσης οι οποίες αναφέρονται στην προηγούμενη ενότητα όπου γίνεται η αναλυτική περιγραφή του αλγορίθμου.

Μεταβλητές	Τιμή	Περιγραφή
m_0	220	Αρχικός αριθμός credits σε κάθε κόμβο
pf_1	1.1	1 ^ο κατώφλι για τη βοήθεια σε απομακρυσμένους κόμβους
pf_2	4	2 ^ο κατώφλι για τη βοήθεια σε απομακρυσμένους κόμβους
$cp_{ICAS,thr}$	0.5	Κατώφλι που χρησιμοποιείται από το ICAS για να αποφασίσει αν ένας κόμβος είναι εγωιστής ή μη
S_t	4	Κατώφλι credit για να αποαφισιστεί η κατάσταση ενός κόμβου
IFN	6	Αριθμός χαμένων πακέτων για τη μεταστροφή ενός εγωιστή κόμβου
edp_{th}	0.8	Πιθανότητα απόρριψης πακέτου για τους εγωιστές κόμβους
$m_{e,0}$	28	Αριθμός credit που δίνονται αρχικά σε απομακρυσμένους κόμβους
h_{max}	13	Μέγιστος αριθμός βημάτων για πακέτα δεδομένων
M_{max}	1200	Μέγιστος αριθμός credits που ένας κόμβος μπορεί να κατέχει
M_{thr1}	600	1 ^ο κατώφλι για την επιβράβευση δρομολογητών κόμβων
M_{thr2}	1000	2 ^ο κατώφλι για την επιβράβευση δρομολογητών κόμβων
P_{cheat}	1.0	Πιθανότητα εμφάνισης εγωιστικής συμπεριφοράς
$head_p$	60	Μέγεθος byte επικεφαλίδας πακέτων
rec_{size}	50	Αριθμός αποδείξεων ανά πακέτο αναφοράς επιτυχένων αποστολών

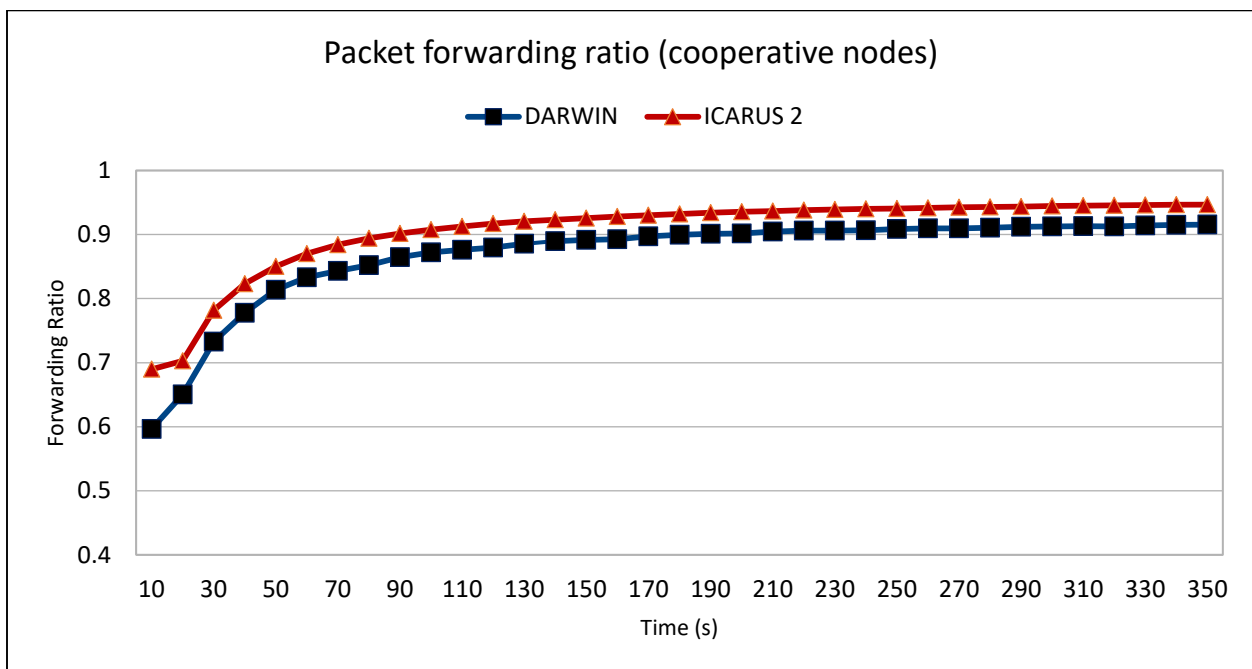
Πίνακας 5.1 Πίνακας μεταβλητών και τιμών

5.2 ΑΡΙΘΜΗΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΓΡΑΦΙΚΕΣ ΠΑΡΑΣΤΑΣΕΙΣ

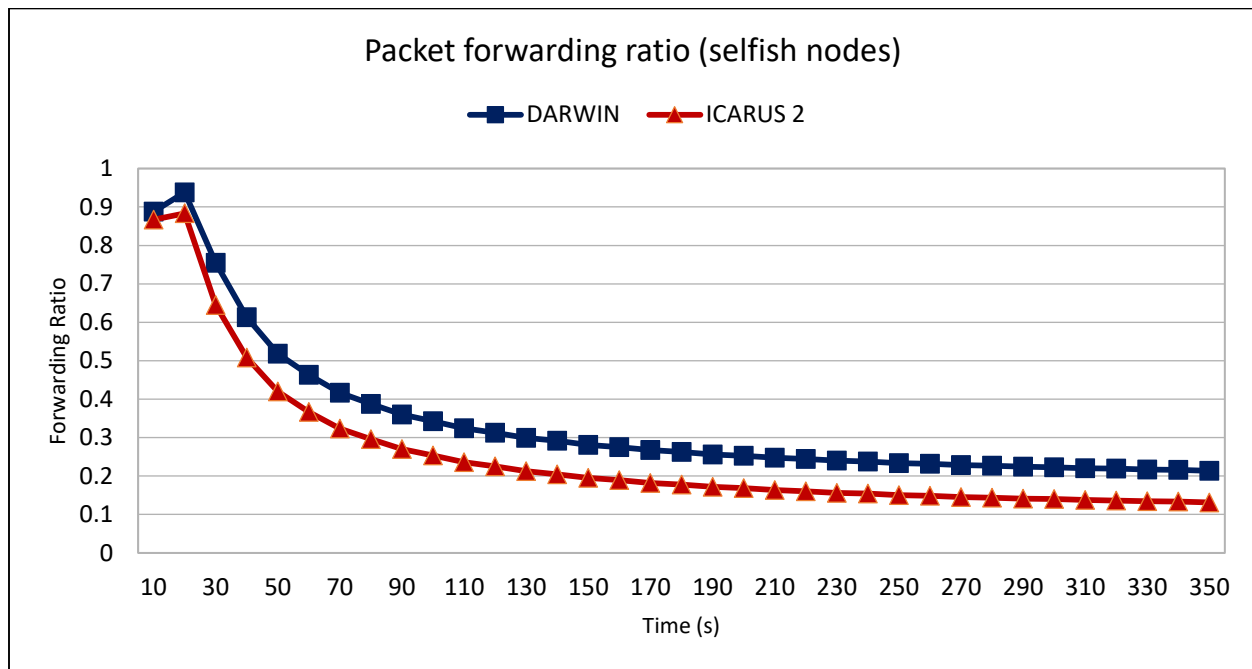
5.2.1 Σύγκριση του ICARUS 2 με το DARWIN

Συγκρίνεται η απόδοση του ICARUS 2 και του DARWIN με βάση την παρακίνηση της συνεργασίας και τον εντοπισμό εγωιστικών συμπεριφορών. Για το λόγο αυτό καταγράφεται τόσο το μέσο ποσοστό προώθησης για όλους τους κόμβους, που ορίζεται ως ο λόγος του αριθμού των πακέτων που μεταδόθηκαν επιτυχώς ως προς τον αριθμό των πακέτων που αποστάλθηκαν.

Οι εικόνες 5.1 και 5.2 απεικονίζουν τα ποσοστά προώθησης πακέτων για συνεργατικούς και μη- συνεργατικούς κόμβους. Σύμφωνα με την εικόνα 5.1 ο ICARUS 2 αποδίδει ελαφρώς καλύτερα (μέχρι 3%) για τους συνεργατικούς κόμβους από τα πρώτα κίολας στάδια της προσομοίωσης. Αντίστοιχα είναι και τα αποτελέσματα για τους εγωιστές κόμβους σε ένα ποσοστό που αγγίζει στο τέλος της προσομοίωσης το 3%, όπως δείχνει η εικόνα 5.2 . Παρατηρούμε ότι αρχικά οι εγωιστές κόμβοι και στις δύο προσομοιώσεις έχουν μία μικρή αύξηση του ποσοστού τους που οφείλεται στη καθυστέρηση της διαδικασίας της ανίχνευσης των εγωιστών κόμβων.



Εικόνα 5.1 Ποσοστό προώθησης για συνεργατικούς κόμβους



Εικόνα 5.2 Ποσοστό προώθησης πακέτων για εγωιστές κόμβους

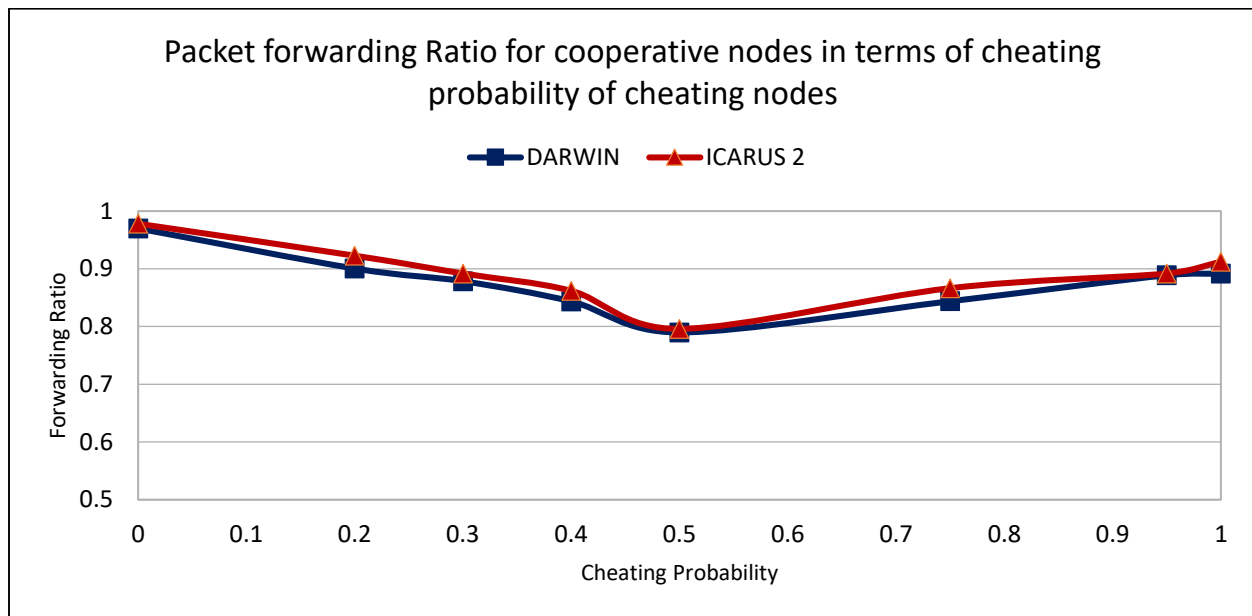
Στις άνωθεν προσομοιώσεις είχαμε διατηρήσει σταθερές τις περιόδους επανάληψης των διεργασιών τόσο των κόμβων όσο και του ICAS και προέκυψαν έπειτα από δέκα επαναλήψεις προσομοιώσεων και ο αριθμός εγωιστών κόμβων ήταν 4. Παρακάτω παρατίθενται οι χρονικές περιόδοι:

Πίνακας Σταθερών	
Σταθερά	Δευτερόλεπτα
<i>Discovery Period</i>	100
<i>Darwin Update Period</i>	25
<i>Icas Update Period</i>	10
<i>Packet Delivery Period</i>	5
<i>Selfish & Malicious List Broadcast Period</i>	20
<i>Node Status Update Period</i>	20
<i>Distant Node Assist Period</i>	10

Πίνακας 5.2 Πίνακας σταθερών προσομοίωσης

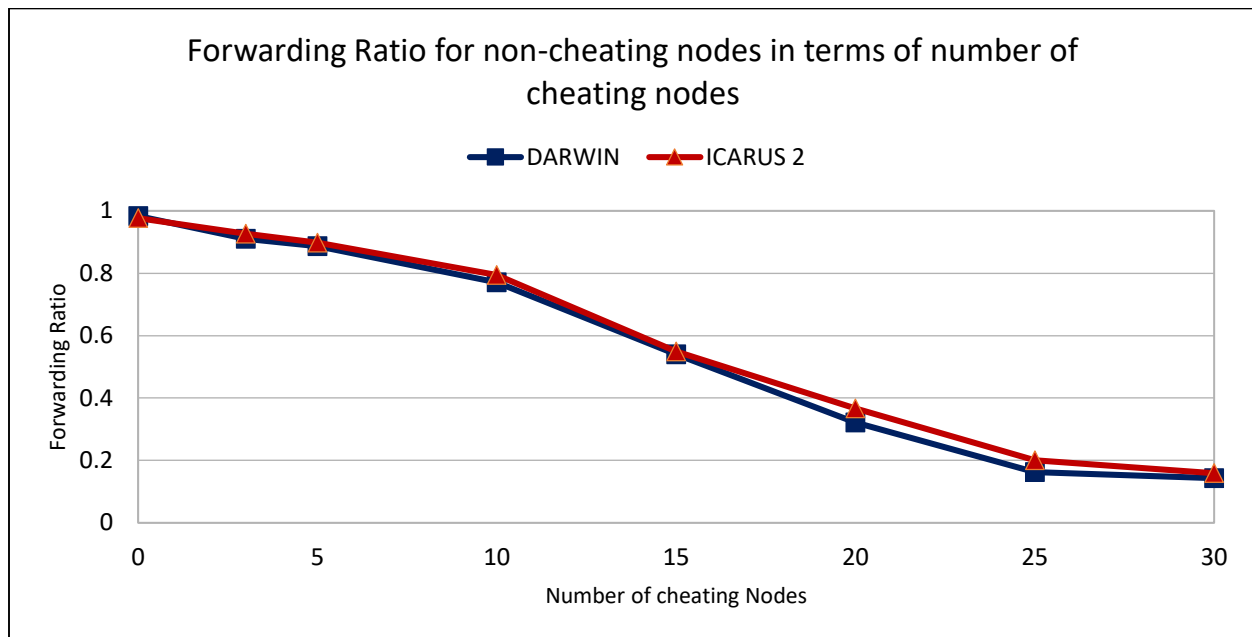
Στη συνέχεια και στην εικόνα 5.4 μελετάται το ποσοστό προώθησης πακέτων των συνεργατικών κόμβων με μεταβλητές τιμές cheating probability των εγωιστών κόμβων. Παρατηρούμε ότι η χειρότερη επίδοση έχουμε όταν η πιθανότητα αυτή ισούται με 0.5, που πρακτικά σημαίνει ότι οι εγωιστές κόμβοι απορρίπτουν 1 στα 2 πακέτα κάνοντας δυσκολότερη την ανίχνευσή του από το υπόλοιπο δίκτυο. Όταν η πιθανότητα μηδενίζεται τότε ουσιαστικά δεν παρουσιάζουν

εγωιστική συμπεριφορά και έχουμε το μεγαλύτερο ποσοστό προώθησης ενώ όταν αγγίζει το 1 τότε έχουμε τα ποσοστά προώθησης κοντά στο 93% (όπως είδαμε και παραπάνω) που διαμορφώνονται μετά την ανίχνευσή τους από το δίκτυο.



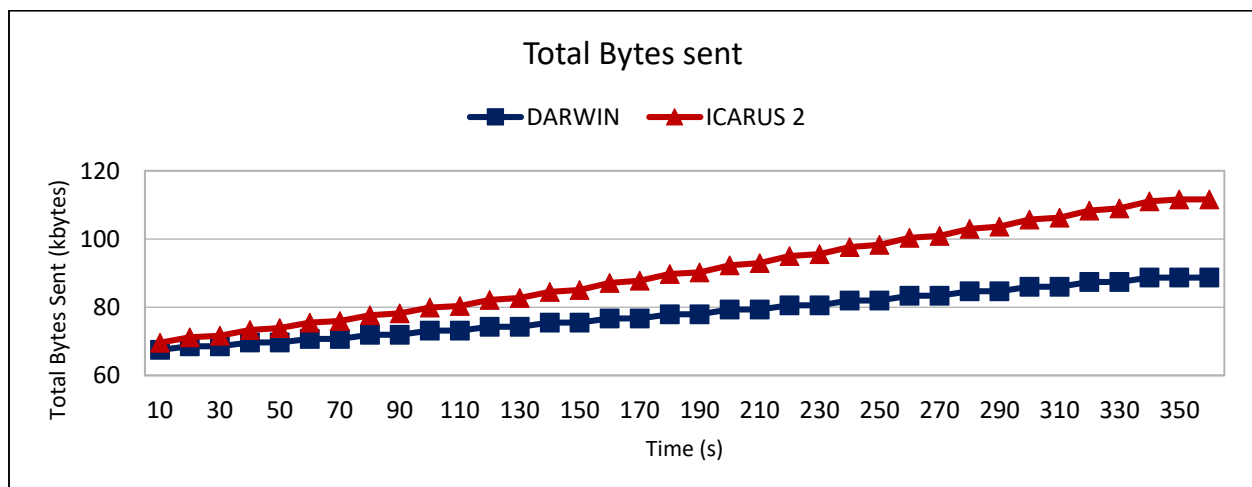
Εικόνα 5.3 Ποσοστό προώθησης πακέτων συνεργατικών κόμβων με μεταβλητή cheating probability

Το γράφημα της εικόνας 5.4 περιέχει το ποσοστό προώθησης των συνεργατικών κόμβων αναλόγως του συνολικού αριθμού των εγωιστών κόμβων που υπάρχουν στο πλέγμα. Εύκολα διαπιστώνεται ότι όσο μικρότερος είναι αυτός ο αριθμός τόσο μεγαλύτερο είναι το ποσοστό αυτό και τόσο πιο μικρή είναι η απόκλιση των δύο μηχανισμών φήμης. Όσο αυξάνεται, το ποσοστό πέφτει σημαντικά και αυξάνεται παράλληλα και η διαφορά τους.



Εικόνα 5.4 Ποσοστό προώθησης πακέτων συνεργατικών κόμβων με μεταβλητό αριθμό εγωιστικών κόμβων

Στο σχήμα της εικόνας 5.5 παρατηρούμε τα συνολικά bytes που στέλνει ένας συνεργατικός κόμβος. Αποτυπώνεται έτσι το αρνητικό αντίκτυπο που έχει ο μηχανισμός ICARUS 2 στη συνολική επίδοση του δικτύου αφού αυξάνεται κατά πολύ η χρησιμοποίησή του για τη μεταφορά δεδομένων από και προς την ψηφιακή τράπεζα και πρακτικά απαιτήθηκε 25% μεγαλύτερη χωρητικότητα.

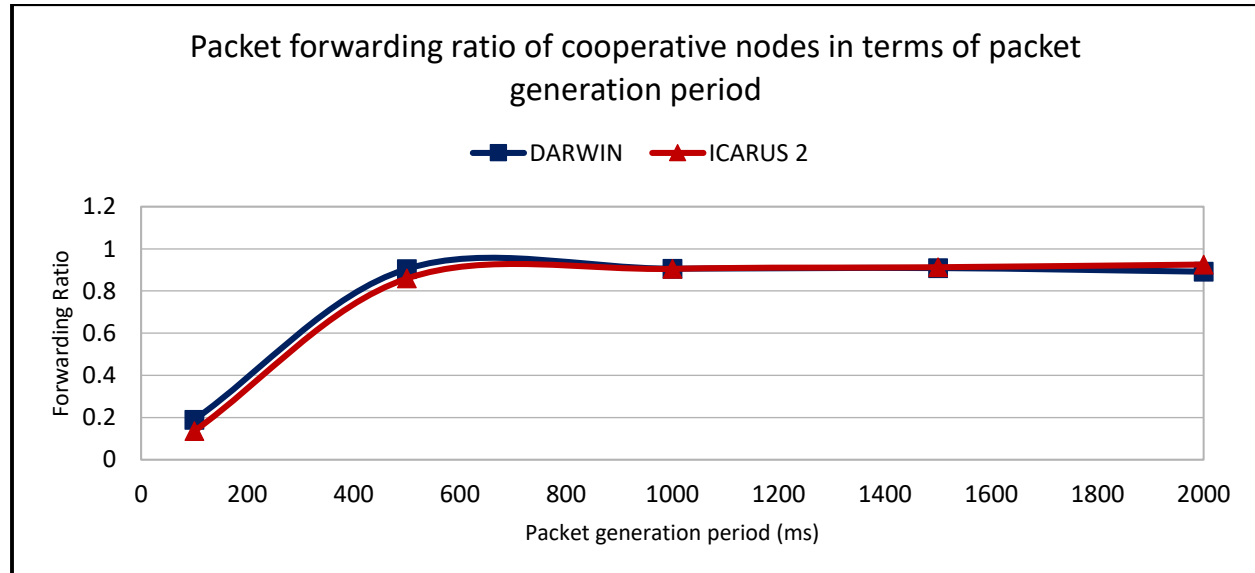


Εικόνα 5.5 Συνολικά bytes που απαιτήθηκαν για την αποστολή πακέτων σηματοδosis

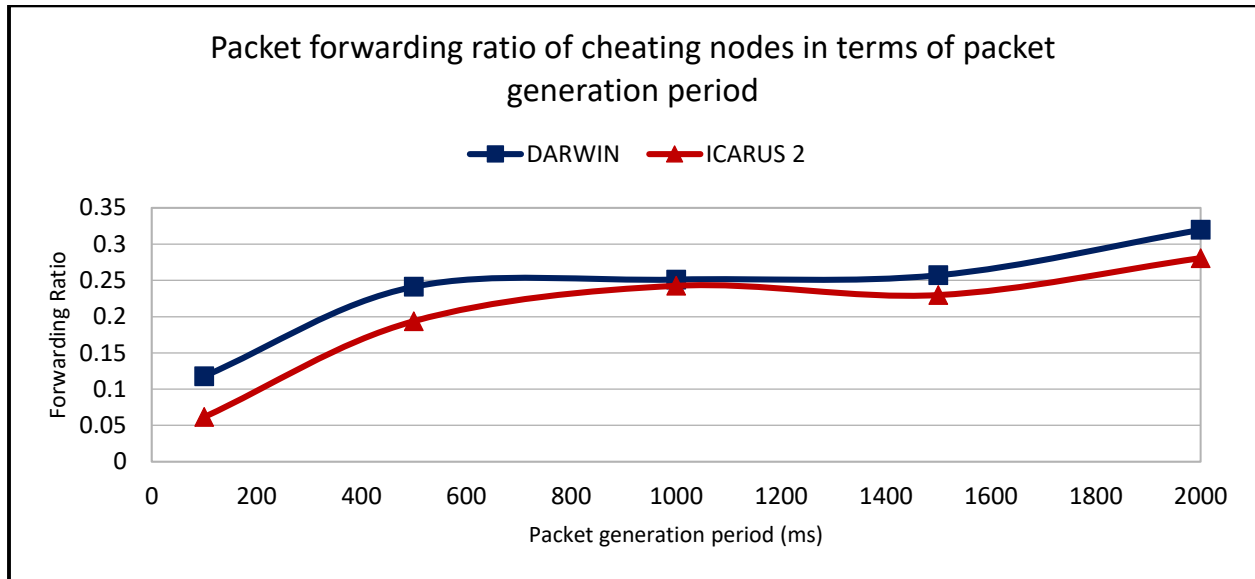
Παρατηρούμε επίσης την ραγδαία αύξηση της διαφοράς των δύο μηχανισμών καθώς η διάρκεια της προσομοίωσης αυξάνει. Επομένως, αποτελεί μείζονος σημασίας η επίλυση αυτής της προβληματικής συμπεριφοράς του μηχανισμού ώστε να μπορεί να είναι υλοποιήσιμος σε κινητές συσκευές- κόμβους με μειωμένα χαρακτηριστικά και μειωμένο κόστος.

Εν συνεχεία, ερευνούμε τη συμπεριφορά του δικτύου για διαφορετικές τιμές της περιόδου αποστολής πακέτων με τιμές: 0.1, 0.5, 1, 1.5, 2s, διατηρώντας τις υπόλοιπες τιμές σταθερές.

Στις παρακάτω εικόνες, απεικονίζεται το ποσοστό προώθησης των συνεργατικών και των εγωιστών κόμβων για τις διάφορες τιμές της μεταβλητής. Αξίζει να σημειωθεί ότι όταν η περίοδος είναι 0.1s τότε σημειώνεται ραγδαία μείωση του ποσοστού προώθησης. Αυτό οφείλεται στην υλοποίηση της προσομοίωσης και συγκεκριμένα στα πάρα πολλά HTTP μηνύματα που δέχεται τόσο ο web server των κόμβων όσο και του ICAS και τα οποία δεν μπορούν να εξυπηρετηθούν. Ωστόσο, αυτό δεν είναι ενδεικτικό ενός αληθινού συστήματος όπου θα χρησιμοποιούταν άλλη τεχνολογία για την υλοποίησή του. Παρατηρούμε ότι στους συνεργατικούς κόμβους δεν παίζει μεγάλο ρόλο η περίοδος αυτή όμως στους εγωιστές παρατηρείται μία μείωση του ποσοστού προώθησης κατά 3.5% και ο ICARUS 2 είναι μονίμως βέλτιστος του DARWIN. Στους συνεργατικούς κόμβους, για μικρές περιόδους παραγωγής πακέτων ο μηχανισμός ICARUS 2 είναι κατά 1.5% χειρότερος του DARWIN, ενώ στη συνέχεια το ποσοστό αυτό μειώνεται ώσπου γίνεται καλύτερος του DARWIN.

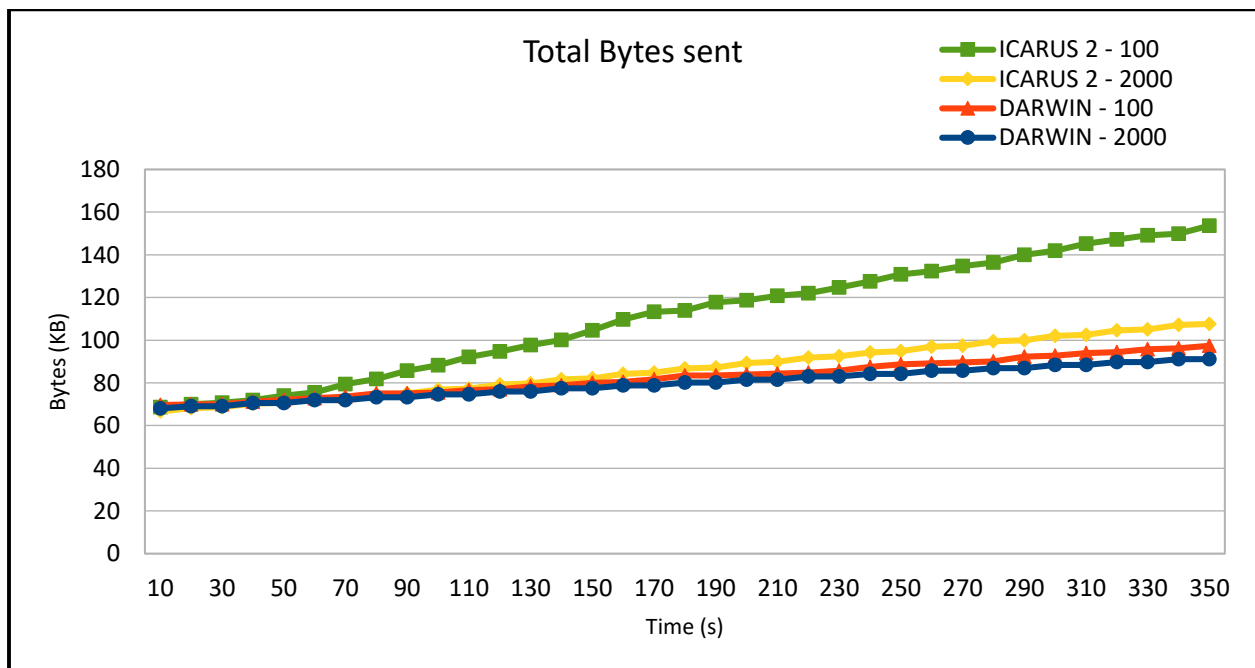


Εικόνα 5.6 Ποσοστό προώθησης συνεργατικών κόμβων για διάφορες τιμές περιόδου αποστολής πακέτων



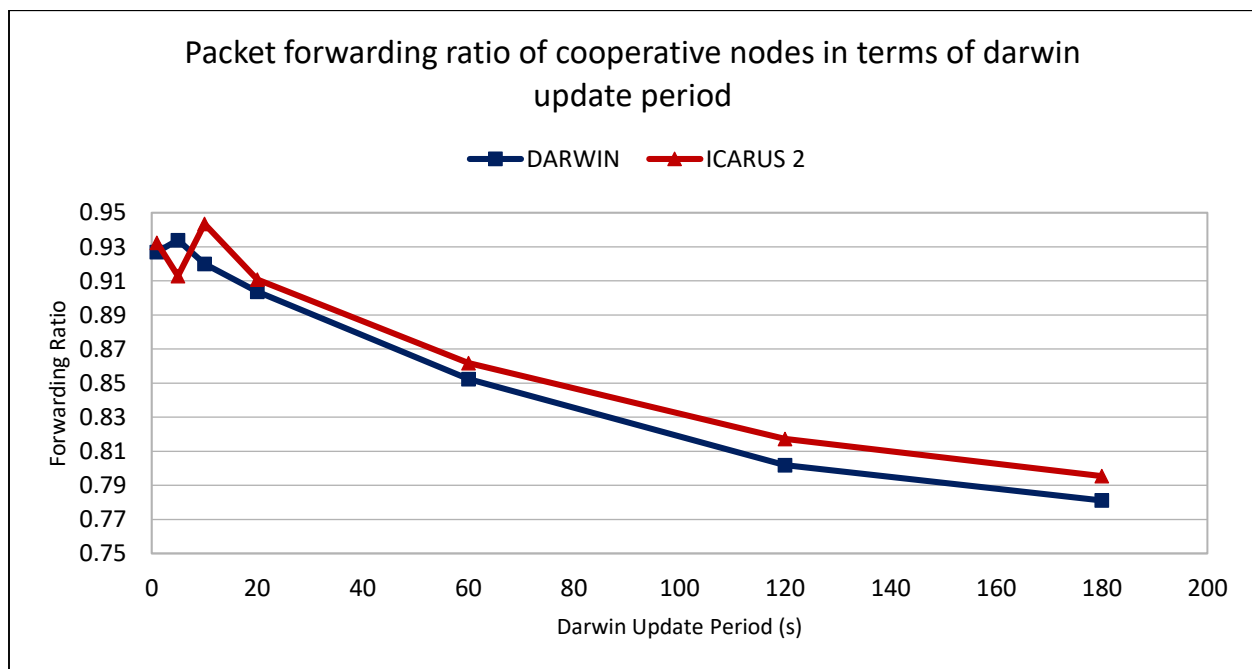
Εικόνα 5.7 Ποσοστό προώθησης εγωιστών κόμβων για διάφορες τιμές περιόδου αποστολής πακέτων

Στο παρακάτω γράφημα απεικονίζεται ο συνολικός αριθμός των kilobyte που αποστέλλει ένας συνεργατικός κόμβος κατά την προσομοίωση των δύο μηχανισμών για περιόδους αποστολής πακέτων 0.1s και 2s.

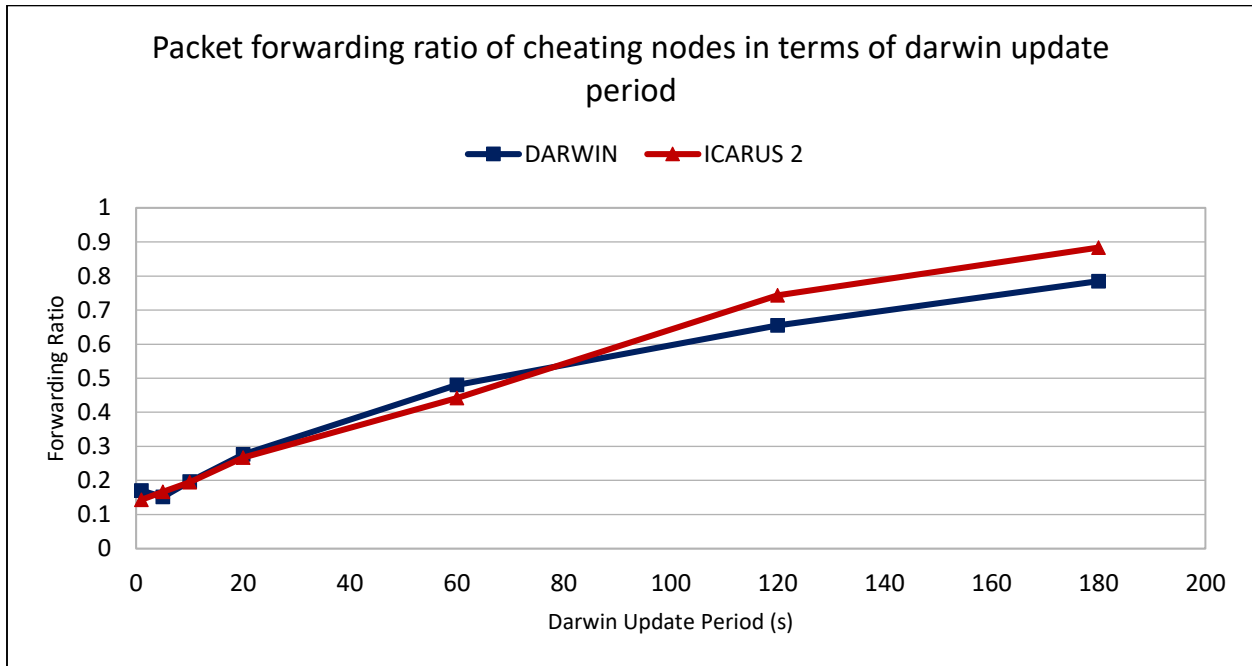


Εικόνα 5.8 Συνολικά bytes που απαιτήθηκαν για την αποστολή πακέτων σηματοδότησης για διαφορετικές τιμές περιόδου αποστολής πακέτων

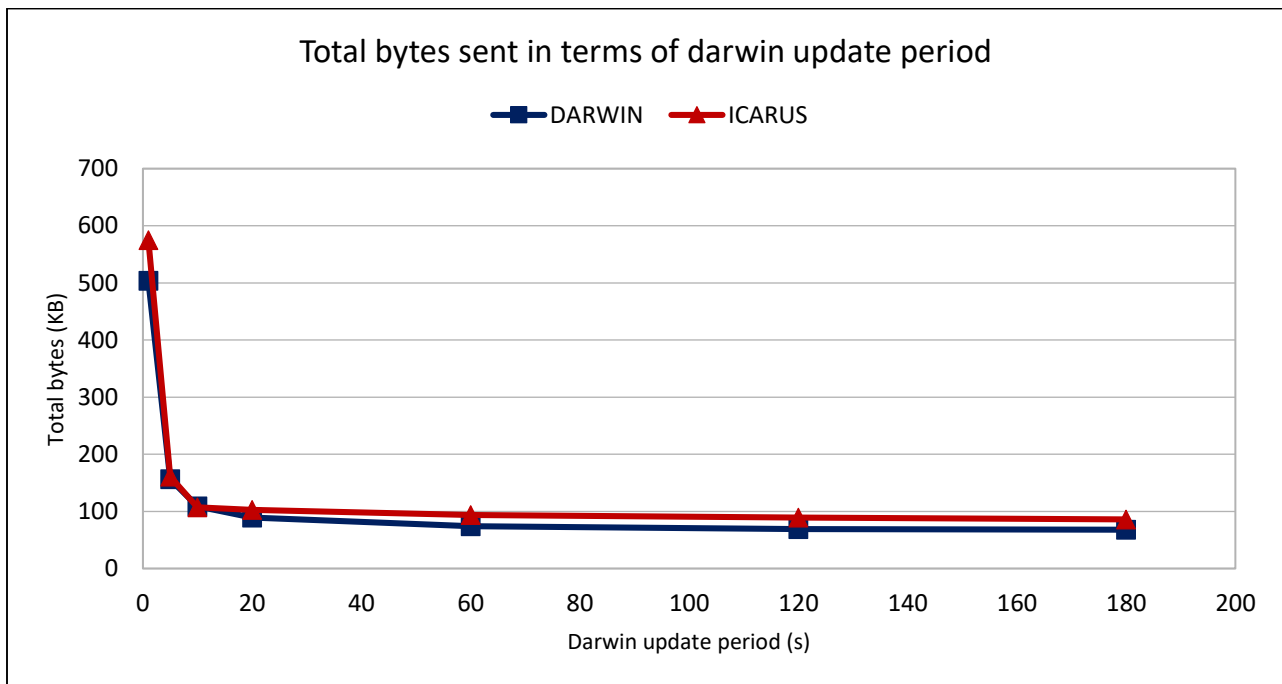
Στην πορεία, στις εικόνες 5.9 και 5.10 παρουσιάζονται η γραφική παράσταση του ποσοστού προώθησης των συνεργατικών και εγωιστών κόμβων για τους δύο μηχανισμούς σε συνάρτηση της περιόδου αποστολής των πακέτων DARWIN, τα οποία περιλαμβάνουν τις τιμές φήμες των γειτόνων και είναι απαραίτητα για την εκτέλεση του αλγορίθμου. Παρατηρούμε ότι όσο αυξάνεται η περίοδος τόσο πέφτει το ποσοστό προώθησης των συνεργατικών και αυξάνεται το ποσοστό προώθησης των εγωιστών και εξηγείται από το γεγονός ότι οι κόμβοι αργούν να αντιδράσουν στις παρατηρούμενες εγωιστικές συμπεριφορές των γειτόνων τους. Αξίζει να σημειωθεί ωστόσο ότι όσο αυξάνεται η περίοδος και μεν ο ICARUS 2 παραμένει καλύτερος του DARWIN για τους συνεργατικούς κατά 1.5% αλλά ο DARWIN έχει χαμηλότερα ποσοστά προώθησης για τους εγωιστές κατά 2.3%, κάτι που τον κάνει αποδοτικότερο από τον ICARUS 2 για τις υψηλές τιμές της περιόδου αποστολής πακέτων DARWIN κατά . Επίσης παρουσιάζεται και η γραφική απεικόνιση των συνολικών bytes που αποστέλλει ένας κόμβος σε συνάρτηση των τιμών της περιόδου.



Εικόνα 5.9 Ποσοστό προώθησης πακέτων συνεργατικών κόμβων για διάφορες τιμές της περιόδου αποστολής πακέτων DARWIN



Εικόνα 5.10 Ποσοστό προώθησης πακέτων εγωιστών κόμβων για διάφορες τιμές της περιόδου αποστολής πακέτων DARWIN

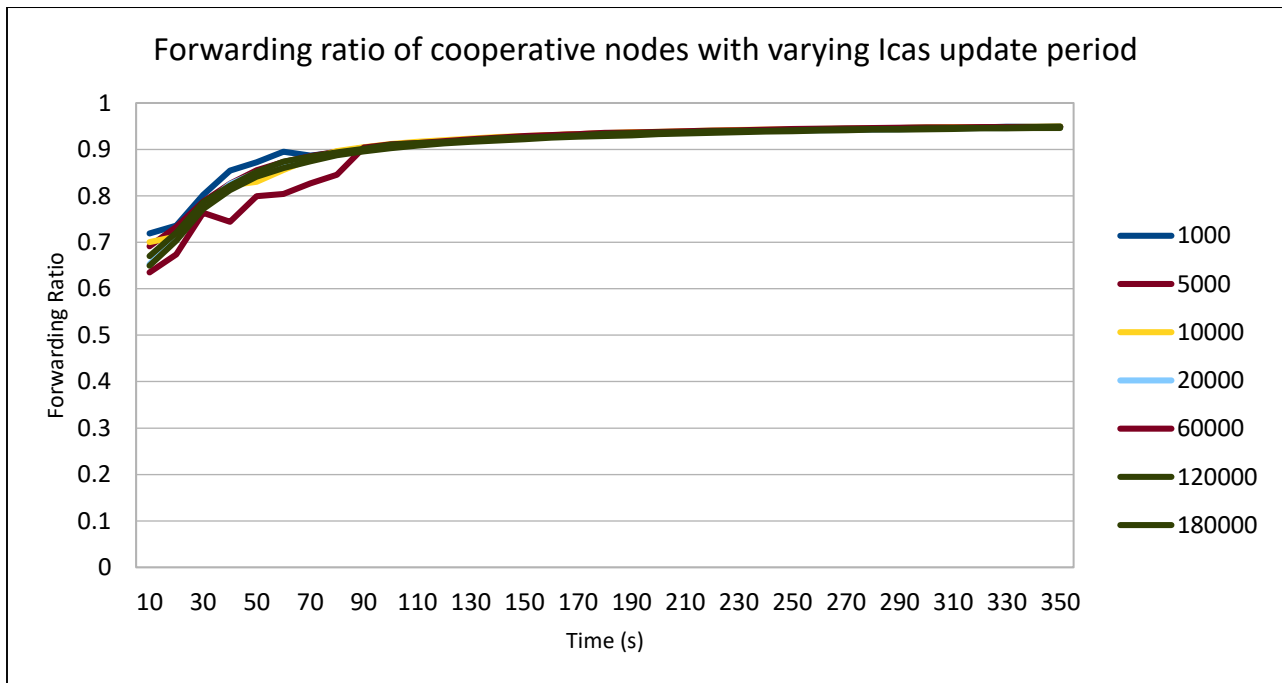


Εικόνα 5.11 Συνολικά bytes που απαιτήθηκαν για την αποστολή πακέτων σηματοδότησης για διαφορετικές τιμές της περιόδου αποστολής πακέτων DARWIN

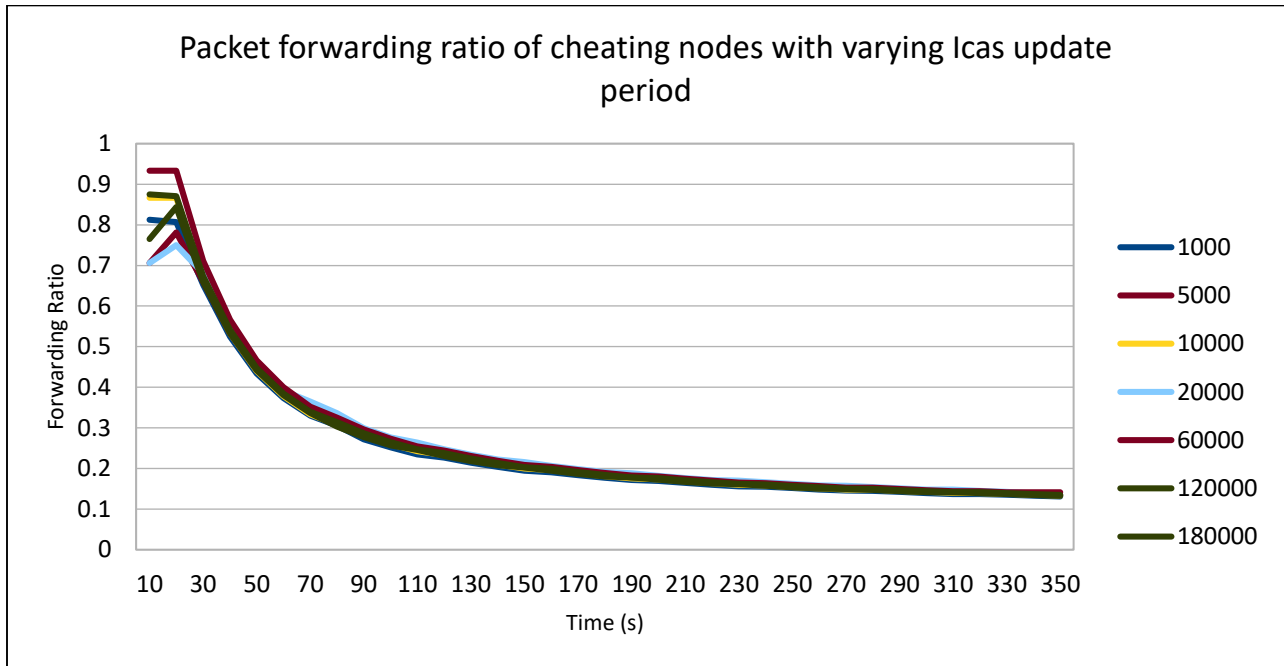
5.2.2 Αναλυτική μελέτη του μηχανισμού ICARUS 2

Στη συνέχεια γίνονται δύο προσομοιώσεις που αφορούν αποκλειστικά τη λειτουργία του ICARUS 2 μηχανισμού και τη βελτιστοποίησή του.

Αρχικά στην πρώτη προσομοίωση εκτελούμε το πείραμα 7 φορές για διαφορετικές τιμές περιόδους αποστολής πακέτων ενημέρωσης φήμης προς το ICAS. Εν συνεχεία συγκρίνουμε τα αποτελέσματα για τα ποσοστά προώθησης συνεργατικών και εγωιστών κόμβων αλλά δεν παρατηρείται ουσιαστική διαφορά αποτελεσμάτων για τις διάφορες τιμές, όπως παρατηρείται και στις εικόνες 5.12 και 5.13. Όμως ότι για μικρές τιμές της περιόδου ο αριθμός των bytes σηματοδότησης ανεβαίνει και για πολύ μεγάλες πέφτει το ποσοστό προώθησης των συνεργατικών, ιδανική περίοδος είναι αυτή των 20000ms.

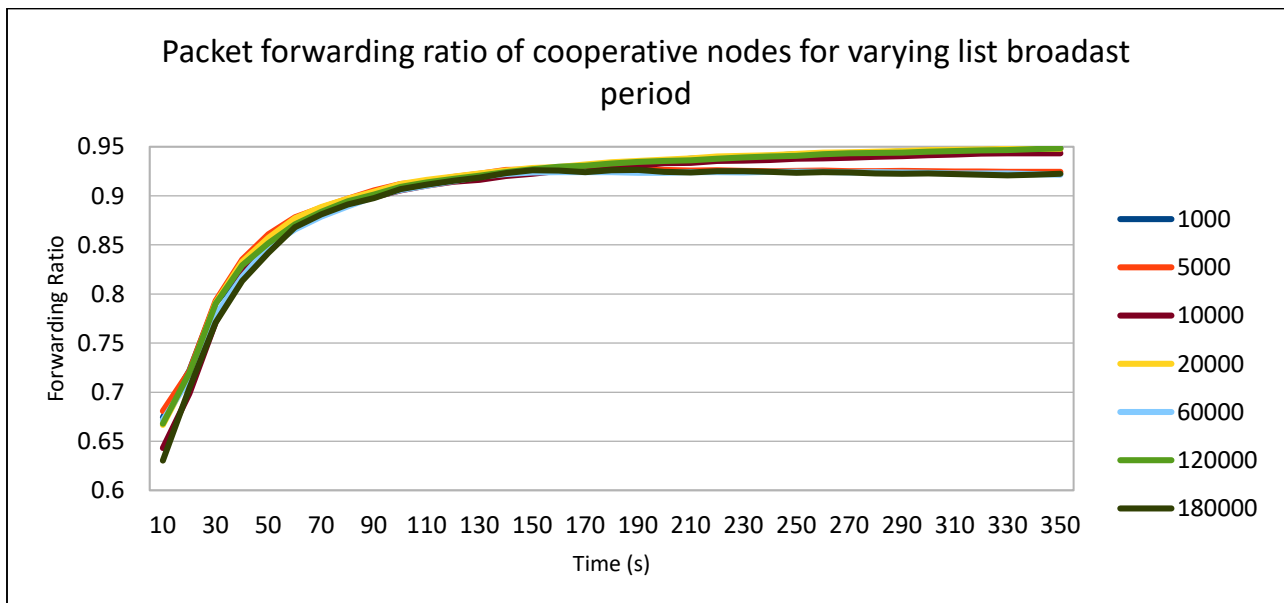


Εικόνα 5.12 Ποσοστό προώθησης πακέτων συνεργατικών κόμβων για διάφορες τιμές της περιόδου αποστολής πακέτων φήμης προς το ICAS

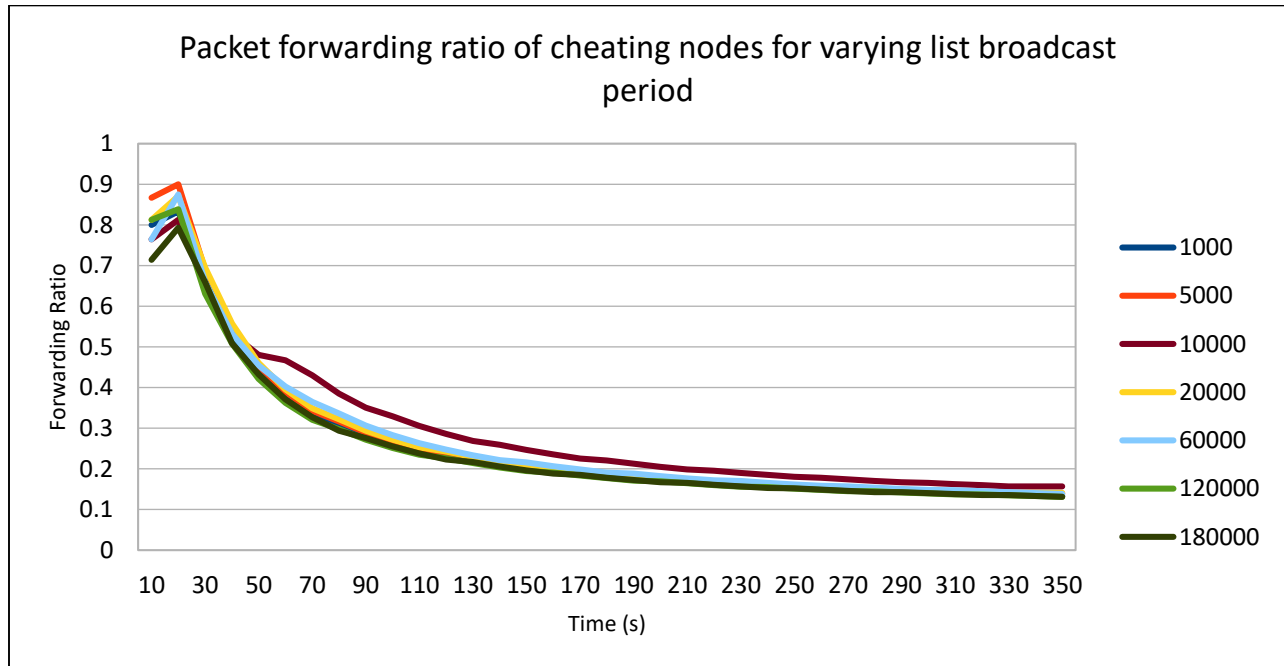


Εικόνα 5.13 Ποσοστό προώθησης πακέτων εγωιστών κόμβων για διάφορες τιμές της περιόδου αποστολής πακέτων φήμης προς το ICAS

Στις εικόνες 5.14 και 5.15 βλέπουμε τα αντίστοιχα ποσοστά προώθησης για τις διάφορες τιμές της περιόδου εκπομπής της λίστας συνεργατικών- εγωιστών.



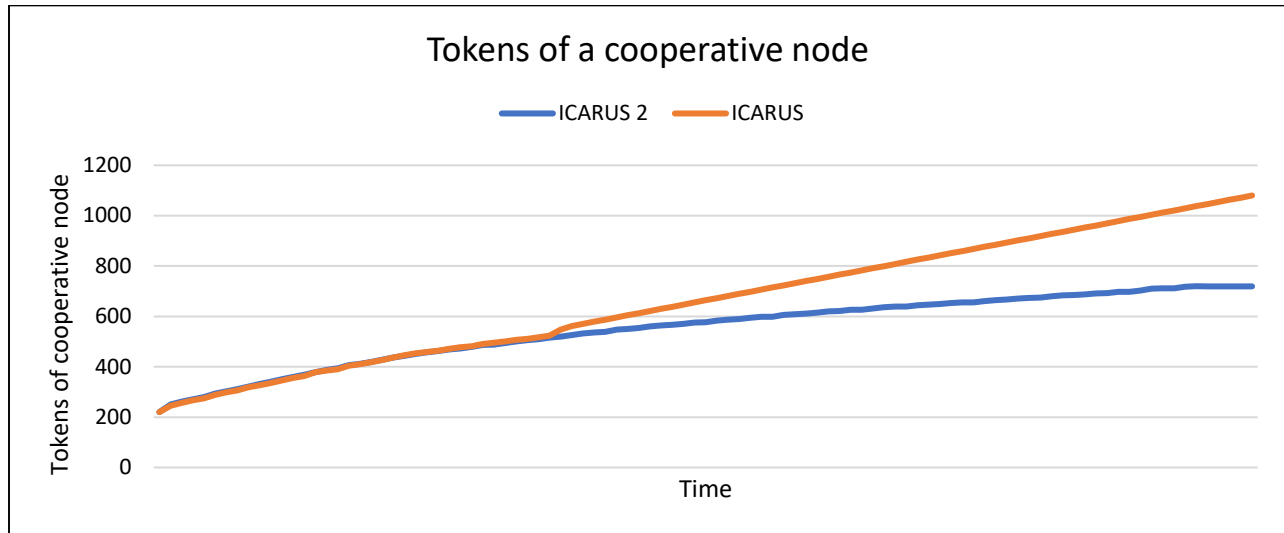
Εικόνα 5.15 Ποσοστό προώθησης πακέτων συνεργατικών κόμβων για διάφορες τιμές της περιόδου εκπομπής λίστας εγωιστών



Εικόνα 5.16 Ποσοστό προώθησης πακέτων εγωιστών κόμβων για διάφορες τιμές της περιόδου εκπομπής λίστας εγωιστών

Σε αυτό το σημείο αξίζει να τονιστεί ότι στην περίπτωση πολύ μεγάλης περιόδου εκπομπής λίστας εγωιστών το ποσοστό προώθησης μπορεί να κρατιέται σε ένα αρκετά καλό επίπεδο και αυτό οφείλεται στο γεγονός ότι ο μηχανισμός περιλαμβάνει και τη διαδικασία ανταλλαγής πακέτων DARWIN, με τα οποία οι κόμβοι αποφασίζουν εσωτερικά για τους εγωιστές γείτονες-κόμβους.

Στην επόμενη προσομοίωση που πραγματοποιήθηκε, θελήσαμε να συγκρίνουμε το μηχανισμό επιβράβευσης του βελτιωμένου αλγορίθμου σε σχέση με το μηχανισμό επιβράβευσης που χρησιμοποιήθηκε στην αρχική του εκδοχή. Το κύριο πρόβλημα που χρίζει αντιμετώπισης είναι ο περιορισμός του άκρατου πλουτισμού των συνεργατικών κόμβων που θα μπορούσε να τους ωθήσει σε εγωιστική συμπεριφορά, σε σημείο που να υποβάθμιζαν κατά πολύ την ποιότητα υπηρεσιών του ασύρματου δικτύου και που θα ήταν αρκετά δύσκολο για την ψηφιακή τράπεζα να εντοπίσει την αιτία του προβλήματος. Στην προσομοίωση παρατηρούμε άμεσα αυτήν την διαφορά αφού στο νέο μηχανισμό ο ρυθμός αύξησης των tokens ενός κόμβου μειώνεται όσο αυξάνεται ο τραπεζικός του λογαριασμός μέχρι να φτάσει ένα συγκεκριμένο κατώφλι και το οποίο δεν μπορεί να ξεπεράσει. Σε αντίθεση, ο παλιός μηχανισμός δεν κάνει αυτήν την πρόβλεψη και δεν μπορεί να περιορίσει τον υπέρμετρο πλουτισμό ενός κόμβου. Η προσομοίωση έγινε σε ένα διάστημα 1000 δευτερολέπτων.

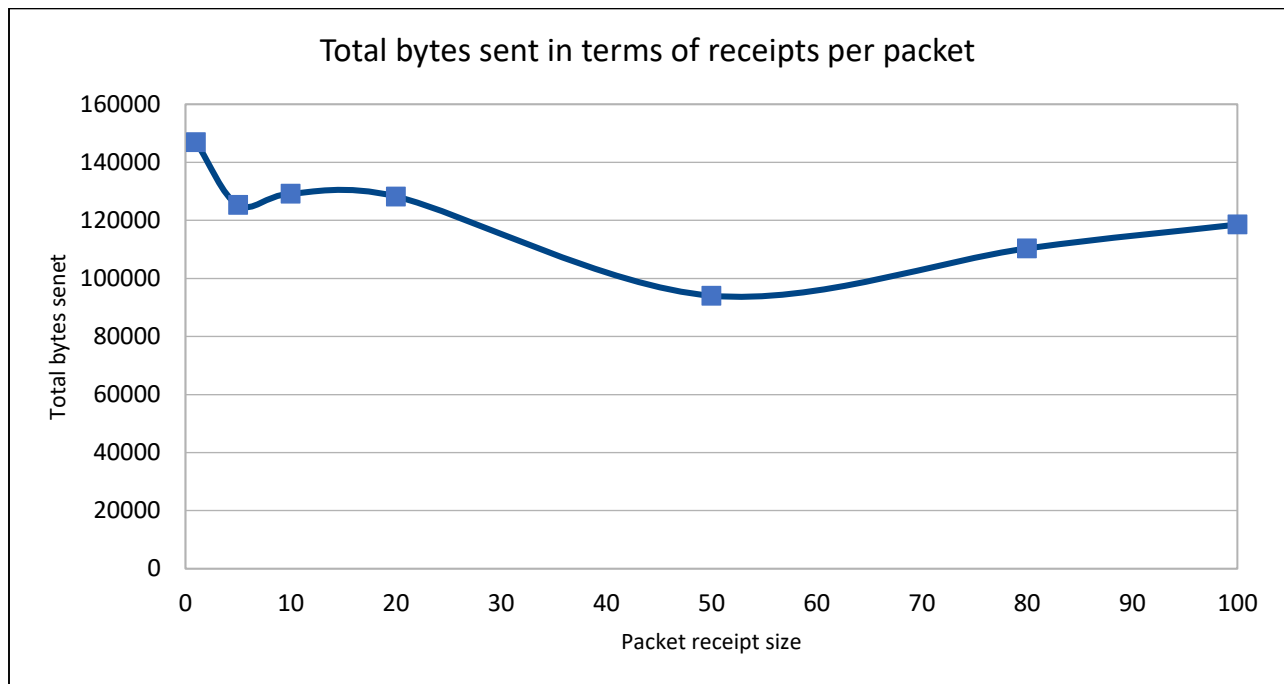


Εικόνα 5.17 Αριθμός των tokens ενός συνεργατικού κόμβου στο ICAS με το νέο και το παλαιό μηχανισμό

Η άνοδος των tokens ενός συνεργατικού κόμβου με τον παλαιό μηχανισμό είναι γραμμική ενώ με το νέο μηχανισμό είναι λογαριθμική.

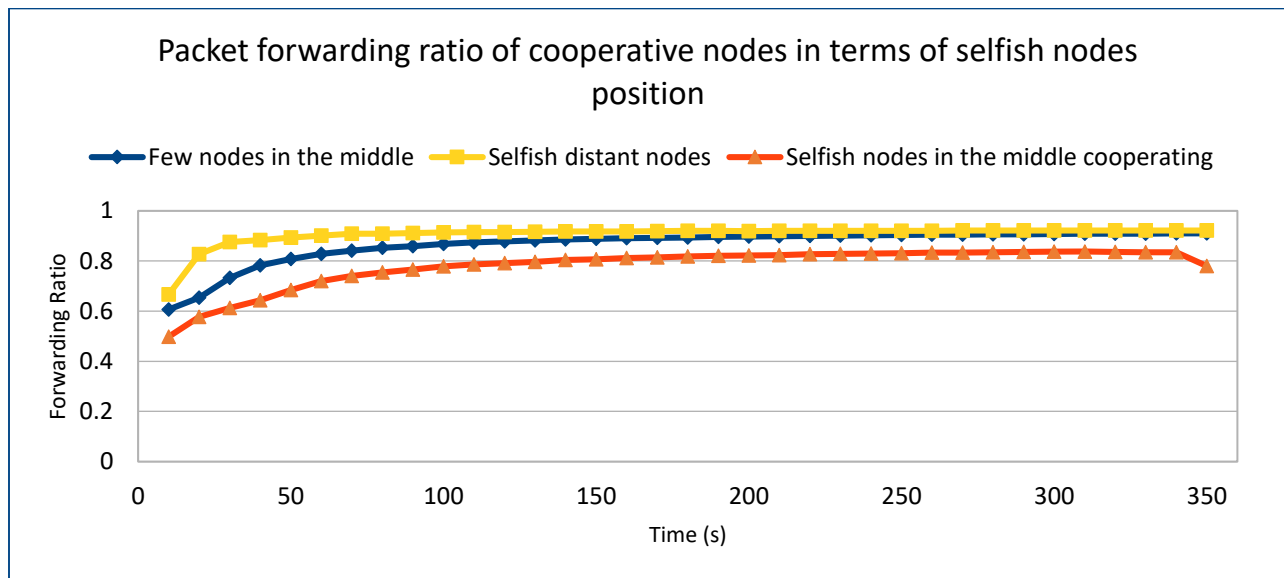
Στη συνέχεια, ερευνούμε ένα εξίσου σημαντικό ζήτημα. Το μεγαλύτερο πρόβλημα του ICARUS είναι η πρόσθετη επιβάρυνση του δικτύου λόγω της επικοινωνίας του κάθε κόμβου με την ψηφιακή τράπεζα [24], το οποίο αντιμετωπίζεται ως ένα βαθμό από τον ICARUS 2. Ένα αρκετά σημαντικό overhead προέρχεται από τις ειδοποιήσεις που στέλνει κάθε κόμβους με το που λαμβάνει ένα πακέτο προκειμένου να επιβραβευτούν οι δρομολογητές κόμβοι και να χρεωθεί ο αποστολέας. Σε ένα όμως πλέγμα 50 κόμβων και με περίοδο αποστολής ανά 2 δευτερόλεπτα, γίνεται αμέσως αντιληπτό ότι πρόκειται για ένα τεράστιο όγκο πακέτων τα οποία πρέπει να αποσταλούν στο ICAS. Γι' αυτό το λόγο, κρίθηκε σημαντικό η ομαδοποίηση των αποδείξεων που αποστέλλονται με σκοπό την μείωση του συνολικού αριθμού πακέτων που εκπέμπονται. Αν θεωρήσουμε ότι S το μέγεθος της ψηφιακής απόδειξης αποστολής και H το μέγεθος της επικεφαλίδας του πακέτου, τότε για ομαδοποίηση N αποδείξεων σε ένα πακέτο έχουμε μέγεθος πακέτου $N \times S + H$ ενώ για αποστολή ξεχωριστού πακέτου ανά απόδειξη έχουμε $N \times (S + H)$ το οποίο είναι σαφώς υψηλότερο αφού εν γένει $N > 1$.

Παρακάτω παρατίθεται η γραφική παράσταση των αποτελεσμάτων της προσομοίωσης, όπου γίνεται αντιληπτή η μείωση του αριθμού των bytes αναλόγως του αριθμού των αποδείξεων ανά πακέτο. Αξίζει να σημειωθεί ότι από ένα όριο και μετά η ποιότητα της υπηρεσίας πέφτει αφού χρειάζονται να συλλεχθούν αρκετές αποδείξεις για την αποστολή ενός πακέτου και αυτό δημιουργεί πρόσθετη ανάγκη για επικοινωνία με το ICAS προκειμένου να έρθει το δίκτυο σε ισορροπία και να διατηρείται η ποιότητα των υπηρεσιών. Ο αριθμός των αποδείξεων ανά πακέτο που τέθηκαν υπό έρευνα είναι: 1, 5, 10, 20, 50, 80, 100.



Εικόνα 5.18 Συνολικά bytes που απαιτήθηκαν για την αποστολή πακέτων σηματοδοσίας για διαφορετικό αριθμό αποδείξεων ανά πακέτο αναφοράς επιτυχημένης αποστολής

Στην επόμενη προσομοίωση, αποτελέσματα της οποίας παρουσιάζονται στην εικόνα 5.19, ερευνάται η επίδραση της τοπολογίας των εγμιστών κόμβων στη συνολική επίδοση του δικτύου. Τα πρώτο σενάριο που εκτελέστηκε είναι η παρουσία ορισμένων εγμιστών κόμβων στο κέντρο της τοπολογίας, το οποίο μοιάζει αρκετά με το σενάριο που έχει εκτελεστεί μέχρι τώρα στις προσομοιώσεις. Το δεύτερο σενάριο αφορά απομακρυσμένους κόμβους που έχουν εγωιστική συμπεριφορά και η οποία, όπως ήταν λογικό, έχει τη μικρότερη επίδραση στο δίκτυο παρουσιάζοντας βελτίωση κατά 4% σε σχέση με το πρώτο. Στο τρίτο σενάριο, η τοπολογία αφορά εγωιστές κόμβους στη μέση της τοπολογίας οι οποίοι κιάλας συνεργάζονται για την υποβάθμιση του δικτύου και έχει τη μέγιστη επίδραση στο δίκτυο, παρουσιάζοντας μία πτώση κατά 13%.

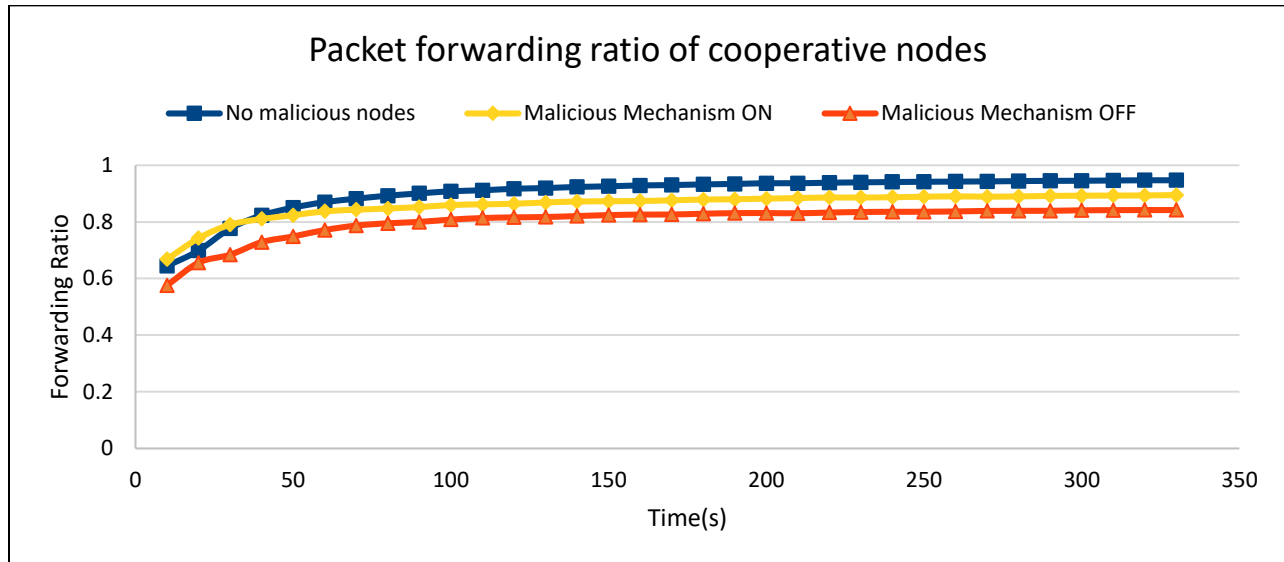


Εικόνα 5.19 Ποσοστό προώθησης πακέτων συνεργατικών κόμβων για διάφορες τοπολογίες εγωιστών κόμβων

Στην επόμενη προσομοίωση, δίνεται έμφαση σε ένα πολύ σημαντικό μερίδιο των κόμβων, τους κακόβουλους κόμβους. Στις προηγούμενες προσομοιώσεις θεωρήσαμε ότι δεν υπάρχουν κακόβουλοι κόμβοι, οι οποίοι να υπονομεύουν την ποιότητα υπηρεσιών του δικτύου. Στην εικόνα 5.19 απεικονίζεται το ποσοστό προώθησης των συνεργατικών κόμβων για τις περιπτώσεις:

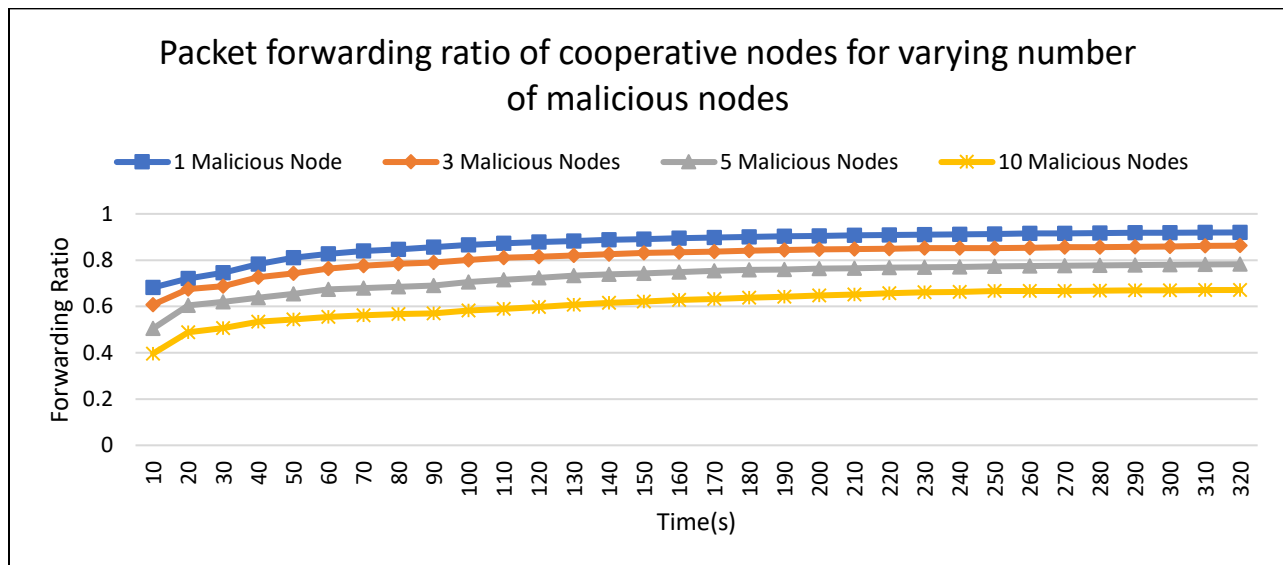
- Που δεν υπάρχουν κακόβουλοι κόμβοι
- Όπου υπάρχουν κακόβουλοι κόμβοι αλλά δεν υπάρχει μηχανισμός αντιμετώπισής τους
- Όπου υπάρχουν κακόβουλοι κόμβοι και ο μηχανισμός αντιμετώπισής τους είναι ενεργοποιημένος

Στην περίπτωση που ο μηχανισμός είναι ενεργοποιημένος τότε παρουσιάζεται βελτίωση 6% σε σχέση με όταν είναι απενεργοποιημένος. Επίσης, σημαντικό είναι να τονιστεί ότι όταν δεν υπάρχουν κακόβουλοι κόμβοι παρουσιάζεται αύξηση της απόδοσης του δικτύου κατά 11% σε σχέση με την περίπτωση όπου υπάρχουν αλλά δεν αντιμετωπίζονται.



Εικόνα 5.20 Ποσοστό προώθησης πακέτων συνεργατικών κόμβων αναλόγως της κατάστασης των κακόβουλων κόμβων

Τέλος στην προσομοίωση της εικόνας 5.21, βλέπουμε τη συμπεριφορά του μηχανισμού αντιμετώπισης εγωιστών κόμβων για μεταβλητό αριθμό κακόβουλων κόμβων. Παρότι ο εντοπισμός των εγωιστών κόμβων γίνεται επιτυχημένα από το ICAS βλέπουμε σημαντική διαφορά όταν ο αριθμός αυτός αυξάνεται. Συγκεκριμένα, όταν το 20% των κόμβων παρουσιάζουν κακόβουλη συμπεριφορά τότε παρατηρείται πτώση της απόδοσης του δικτύου κατά 27% σε σχέση με την περίπτωση 2% κακόβουλων κόμβων και 12% σε σχέση με την περίπτωση 10% κακόβουλων κόμβων.



Εικόνα 5.21 Ποσοστό προώθησης πακέτων συνεργατικών κόμβων για διάφορο αριθμό κακόβουλων κόμβων

5.2.3 Συμπεράσματα

Στο πλαίσιο αυτής της ενότητας, θα παρατεθούν τα συμπεράσματα που απορρέουν από τις γραφικές παραστάσεις της ενότητας 5.2.1 και 5.2.2.

Ο προτεινόμενος υβριδικός μηχανισμός ICARUS 2 αποδείχτηκε στις περισσότερες προσομοιώσεις αρκετά καλύτερος του μηχανισμού φήμης DARWIN και αποδίδεται στην δυνατότητά του να έχει καλύτερο έλεγχο και εποπτεία πάνω στη συμπεριφορά των κόμβων καθώς και να τιμωρεί αυστηρότερα τους εγωιστές και κακόβουλους κόμβους. Επιπλέον, συνδυάζει τα πλεονεκτήματα των μηχανισμών φήμης και πίστωσης. Ο ICARUS 2 αποτελεί έναν ιδιαίτερα αποτελεσματικό μηχανισμό δίχως να παρουσιάζει ιδιαίτερες αδυναμίες στα γενικότερα πλαίσια των μηχανισμών κινήτρων συνεργασίας.

Σε γενικές γραμμές, οι περίοδοι των διαφόρων διεργασιών οφείλουν να μην λαμβάνουν υψηλές τιμές, καθώς τότε το σύστημα αργεί να αντιδράσει στις μεταβολές του περιβάλλοντος- δικτύου και στις αλλαγές των συμπεριφορών των κόμβων. Από την άλλη μεριά, χαμηλές τιμές περιόδων θα οδηγούσαν σε ένα συνωστισμένο δίκτυο υψηλής χρησιμοποίησης του εύρους ζώνης του και συνολικά υψηλότερης κατανάλωσης ενέργειας και πόρων. Επιπλέον, η επικοινωνία μεταξύ των κόμβων μεταξύ τους και με το ICAS θα ήταν μία αρκετά συχνή διαδικασία και θα στοιχίζε οικονομικά στους κόμβους αυτούς καθ' αυτούς.

Στον πίνακα 5.2 αποτυπώνονται οι ιδανικές τιμές περιόδων προκειμένου να έχουμε βέλτιστη χρησιμοποίηση του εύρους ζώνης του ασύρματου αδόμητου δικτύου, συνυπολογίζοντας και το συνολικό κόστος μετάδοσης πακέτων δεδομένων και σηματοδοσίας.

Πίνακας Σταθερών	
Σταθερά	Δευτερόλεπτα
<i>Discovery Period</i>	100
<i>Darwin Update Period</i>	60
<i>Icas Update Period</i>	10
<i>Packet Delivery Period</i>	2
<i>Selfish & Malicious List Broadcast Period</i>	20
<i>Node Status Update Period</i>	20
<i>Distant Node Assist Period</i>	3

Πίνακας 5.2 Πίνακας σταθερών προσομοίωσης

Όσο αναφορά τη σύγκριση με την προηγούμενη εκδοχή του ICARUS [4], παρατηρούμε μία βελτίωση στην αξιοποίηση του εύρους ζώνης που διατίθεται, αφού μετριέται και αξιολογείται σε επίπεδο byte ο όγκος της πληροφορίας που εκπέμπεται. Μία σημαντική προσθήκη είναι η εισαγωγή της των αποδείξεων επιτυχημένης αποστολής πακέτου που οδήγησε σε μείωση της συνολικής εκπεμπόμενης πληροφορίας κατά 30%. Ο ιδανικός αριθμός αποδείξεων ανά πακέτο αναφοράς προς το ICAS καθορίζεται στην τιμή 50.

Τέλος, μελετώνται και διερευνώνται οι κακόβουλοι κόμβοι, για τους οποίους δεν υπήρχε πρόβλεψη, και προτείνεται ένας μηχανισμός αντιμετώπισής τους. Παρατηρούμε ότι για διασφάλιση ποιότητας υπηρεσίας άνω του 90% το ποσοστό κακόβουλων κόμβων οφείλει να μην ξεπερνάει το 2- 4%. Σε διαφορετική περίπτωση, απαιτείται η ανάπτυξη ενός πιο έγκυρου μηχανισμού εντοπισμού και εξάλειψής αυτής της κατηγορίας των κόμβων.

6 ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ

Στο πλαίσιο της διπλωματικής αναπτύχθηκε ο νέος βελτιωμένος υβριδικός μηχανισμός κινήτρων ICARUS 2 που έχει ως σκοπό να ενθαρρύνει τη συνεργατική συμπεριφορά σε ένα ασύρματο αδόμητο δίκτυο. Ως μελλοντική έρευνα μπορεί να θεωρηθεί η εφαρμογή του αλγορίθμου σε πραγματικές υποδομές και τερματικά, με έμφαση στην πολυπλοκότητα που επιφέρει οι αλγόριθμος. Ειδικότερα, σημαντικοί παράμετροι που πρέπει να εξεταστούν διεξοδικά σε αυτά τα σενάρια είναι η συχνότητα με την οποία οι κόμβοι ανανεώνουν τα στατιστικά τους, αλλά και η ελαστικότητα ως προς την παρεχόμενη ποιότητα υπηρεσίας από τους χρήστες. Οι παράμετροι αυτοί μπορούν να επηρεάσουν δραματικά το αποτέλεσμα και πρέπει να καθορίζονται πάντα με βάση το σύστημα υπό εξέταση.

Επιπλέον, η απόδοση του αλγορίθμου πρέπει να εξεταστεί για περισσότερους τύπους υπηρεσιών και εφαρμογών καθώς και σε πιο σύνθετα και ετερογενή περιβάλλοντα. Μία ταξινόμηση των υπηρεσιών που η εφαρμογή του αλγορίθμου ICARUS 2 θα ήταν ιδανική, ανάλογα με την ποιότητα δικτύου που απαιτείται και που μπορεί ο μηχανισμός να προσφέρει, θα βοηθούσε στην μείωση της καθυστέρησης για την ευρεία χρησιμοποίησή του σε πραγματικά δίκτυα νέας γενιάς.

Είναι πλέον εμφανές, ότι οι μηχανισμοί κινήτρων θα αποτελέσουν θεμελιώδη λίθο των δικτύων νέας γενιάς. Αποτελεί μεγάλη πρόκληση η ανάπτυξη νέων αλγορίθμων που μπορούν να επιτυχώς να υποκινήσουν τη συνεργασία μεταξύ οντοτήτων δικτύων, παρέχοντας κίνητρα αλλά και διατηρώντας τη δικαιοσύνη. Σε τέτοιες περιπτώσεις είναι ζωτική η μελέτη θεμάτων ασφαλείας και προστασίας των δικτύων από κακόβουλες οντότητες. Συνεπώς, θα πρέπει να ερευνηθούν οι υπάρχοντες μηχανισμοί αντιμετώπισης κακόβουλων στοιχείων, όπως αυτός που παρουσιάστηκε τα πλαίσια της διπλωματικής και να αναπτυχθούν νέοι μέθοδοι ταυτοποίησης και επιτήρησης. Όλα αυτά θα πρέπει να επιφέρουν φυσικά τον ελάχιστο δυνατό φόρτο στο δίκτυο.

7 ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Cisco, «Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012-2016,» Cisco, February 2013.
- [2] Kanupriya Singh, Sanjeev Thakur and Somya Singh. Article: Comparison of 3G and LTE with other Generation. *International Journal of Computer Applications* 121(6):42-47, July 2015.
- [3] «Mobility Report. Tech. rep. Ericsson, November 2015 (cit. on pp. 2, 73)».
- [4] Dimitris E. Charilas, Konstantinos D. Georgilakis, and Athanasios D. Panagopoulos. 2012. ICARUS: hybrid incentive mechanism for cooperation stimulation in ad hoc networks. *Ad Hoc Netw.* 10, 6 (August 2012), 976-989.
- [5] Marias, G. F., Georgiadis, P., Flitzanis, D. and Mandalas, K. (2006), Cooperation enforcement schemes for MANETs: a survey. *Wirel. Commun. Mob. Comput.*, 6: 319–332.
- [6] Sonja Buchegger and Jean-Yves Le Boudec. 2002. Performance analysis of the CONFIDANT protocol. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing (MobiHoc '02)*. ACM, New York, NY, USA, 226-236.
- [7] Michiardi P., Molva R. (2002) Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks. In: Jerman-Blažič B., Klobučar T. (eds) *Advanced Communications and Multimedia Security*. IFIP — The International Federation for Information Processing, vol 100. Springer, Boston, MA.
- [8] Qi He, Dapeng Wu and P. Khosla, "SORI: a secure and objective reputation-based incentive scheme for ad-hoc networks," 2004 IEEE Wireless Communications and Networking Conference (IEEE Cat. No.04TH8733), 2004, pp. 825-830 Vol.2.
- [9] M. B. Sorav Bansal, «Observation-based Cooperation Enforcement in Ad hoc Networks,» Stanford Technical Report, 2003.
- [10] S. Zhong, J. Chen and Y. R. Yang, "Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks," IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428), 2003, pp. 1987-1997 vol.3.
- [11] Hao Yang, Xiaoqiao Meng, and Songwu Lu. 2002. Self-organized network-layer security in mobile ad hoc networks. In *Proceedings of the 1st ACM workshop on Wireless security (WiSE '02)*. ACM, New York, NY, USA, 11-20.
- [12] E. S. Anderegg L, «Ad-hoc-VCG: a truthful and cost-efficient routing protocol for mobile ad-hoc networks with selfish agents,» σε *Proceedings of 9th Annual International Conference on Mobile Computing and Networking*, September 2003.

- [13] Z. Li and H. Shen, "Analysis of a Hybrid Reputation Management System for Mobile Ad hoc Networks," 2009 Proceedings of 18th International Conference on Computer Communications and Networks, San Francisco, CA, 2009, pp. 1-6.
- [14] M. Felegyhazi, J. P. Hubaux and L. Buttyan, "Nash equilibria of packet forwarding strategies in wireless ad hoc networks," in IEEE Transactions on Mobile Computing, vol. 5, no. 5, pp. 463-476, May 2006.
- [15] J. Guo, H. Liu, J. Dong and X. Yang, "HEAD: A hybrid mechanism to enforce node cooperation in mobile ad hoc networks," in Tsinghua Science and Technology, vol. 12, no. S1, pp. 202-207, July 2007.
- [16] Axelrod, R. (1981). The Emergence of Cooperation among Egoists. *American Political Science Review*, 75(2), 306-318
- [17] Fabio Milan, Juan José Jaramillo, and R. Srikant. 2006. Achieving cooperation in multihop wireless networks of selfish nodes. In *Proceeding from the 2006 workshop on Game theory for communications and networks (GameNets '06)*. ACM, New York, NY, USA, Article 3.
- [18] J. Wu and R. Axelrod. How to cope with noise in the iterated prisoner's dilemma. *The Journal of Conflict Resolution*, 39(1):183–189, Mar. 1995.
- [19] Juan José Jaramillo and R. Srikant. 2007. DARWIN: distributed and adaptive reputation mechanism for wireless ad-hoc networks. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking (MobiCom '07)*. ACM, New York, NY, USA, 87-98.
- [20] M. S. Chaturvedi and Sanjay Srivastava, "Reputation based Cooperation Enforcement Mechanism in Wireless Ad hoc Networks Using Game Theory".
- [21] Vitor G. Rolla and Marilia Curado. 2015. A Simple Survey of Incentive Mechanisms for User-Provided Networks. *Wirel. Pers. Commun.* 83, 4 (August 2015), 2579-2591.
- [22] D. Chatzopoulos and P. Hui, "Asynchronous reputation systems in device-to-device ecosystems," 2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), Coimbra, 2016, pp. 1-3.
- [23] D. E. Asher, A. Zaldivar, B. Barton, A. A. Brewer and J. L. Krichmar, "Reciprocity and Retaliation in Social Games With Adaptive Agents," in IEEE Transactions on Autonomous Mental Development, vol. 4, no. 3, pp. 226-238, Sept. 2012.
- [24] Normalia Samian, Zuriati Ahmad Zukarnain, Winston K.G. Seah, Azizol Abdullah, and Zurina Mohd Hanapi. 2015. Cooperation stimulation mechanisms for wireless multihop networks. *J. Netw. Comput. Appl.* 54, C (August 2015), 88-106.