

Ατομική Διπλωματική Εργασία

**ΜΙΜΟΥΜΕΝΟΙ ΤΗΝ ΦΥΣΗ ΣΤΗΝ ΣΧΕΔΙΑΣΗ ΕΥΡΩΣΤΩΝ
ΜΗΧΑΝΙΣΜΩΝ ΕΛΕΓΧΟΥ ΣΥΜΦΟΡΗΣΗΣ ΣΕ ΔΙΚΤΥΑ
ΑΙΣΘΗΤΗΡΩΝ**

Μαρίνα Λοΐζου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2010

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΙΜΟΥΜΕΝΟΙ ΤΗΝ ΦΥΣΗ ΣΤΗ ΣΧΕΔΙΑΣΗ ΕΥΡΩΣΤΩΝ ΜΗΧΑΝΙΣΜΩΝ ΕΛΕΓΧΟΥ ΣΥΜΦΟΡΗΣΗΣ ΣΕ ΔΙΚΤΥΑ ΑΙΣΘΗΤΗΡΩΝ

Μαρίνα Λοΐζου

Επιβλέπων Καθηγητής

Ανδρέας Πιτσιλλίδης

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των
απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του
Πανεπιστημίου Κύπρου

Μάιος 2010

Ευχαριστίες

Θα ήθελα να εκφράσω τις ευχαριστίες μου προς τον επιβλέπων καθηγητή μου κ. Ανδρέα Πιτσιλλίδη, καθηγητή στο Τμήμα Πληροφορικής του Πανεπιστημίου Κύπρου, ο οποίος με βοήθησε με την γνώση του και με καθοδηγούσε καθ' όλη τη διάρκεια διεκπεραίωσης της ατομικής διπλωματικής μου εργασίας. Θερμές ευχαριστίες οφείλω και στον κ. Παύλο Αντωνίου, ο οποίος είναι ο δημιουργός του αλγορίθμου που μελετάω και ο οποίος στάθηκε στο πλευρό μου από την πρώτη στιγμή και με βοήθησε να κατανοήσω κάθε πτυχή του αλγορίθμου του, καθώς επίσης και τα διάφορα εργαλεία που κλήθηκα να χρησιμοποιήσω. Μέσα από την καθοδήγηση τους κατάφερα να μάθω πολλά νέα πράγματα σχετικά με τα δίκτυα αισθητήρων τα οποία μου φάνηκαν ιδιαίτερος χρήσιμα αργότερα κατά την διάρκεια εκπόνησης της εργασίας μου. Μετά το τέλος του κάθε σεναρίου που μελετούσα είχα πάντοτε συνάντηση και τους δύο καθηγητές για παρουσίαση των αποτελεσμάτων μου και πάντοτε με στήριζαν, με διόρθωναν και με καθοδηγούσαν και χωρίς την δική τους υποστήριξη η διεκπεραίωση της διπλωματικής μου εργασίας θα ήταν πολύ πιο δύσκολη.

Περίληψη

Αυτή η διπλωματική εργασία μελετάει ένα εύρωστο αλγόριθμο που στόχος του είναι ο εντοπισμός περιοχών συμφόρησης σε ένα δίκτυο αισθητήρων. Ο αλγόριθμος αυτός, είναι εμπνευσμένος από την φύση (Swarm Intelligence) και μιμείται την συμπεριφορά των πουλιών που ταξιδεύουν σε κοπάδια και μπορεί να λειτουργεί και να προσαρμόζεται στα φυσικά εμπόδια, όπως ακριβώς θα λειτουργούσε κι ένα κοπάδι πουλιών. Η βασική ιδέα είναι τα πακέτα να δημιουργήσουν κοπάδια και να κινούνται προς την κατεύθυνση που βρίσκεται το sink, ενώ παράλληλα να μπορούν να αποφεύγουν περιοχές όπου υπάρχει συμφόρηση, είτε λόγω αυξημένου φορτίου είτε λόγω κατάρρευσης κάποιων κόμβων. Ο αλγόριθμος υποστηρίζει εξεύρεση της κατεύθυνσης του sink, έλεγχο συμφόρησης ο οποίος είναι βασισμένος στην κατάσταση των ουρών των κόμβων και στην κατάσταση του δικτύου (wireless channel loading), καθώς επίσης και διαχείριση του φόρτου, δηλαδή εξεύρεση εναλλακτικών μονοπατιών σε περίπτωση που υπάρχει μεγάλη συμφόρηση στο δίκτυο.

Μελετώντας αυτό τον αλγόριθμο έπρεπε να κατανοήσω κάποιες βασικές πτυχές του τρόπου λειτουργίας του και να καταφέρω στο τέλος να προσδιορίσω ποιες παράμετροι μπορούν να συμβάλουν στον έγκαιρο εντοπισμό περιοχών συμφόρησης και σε ποιο βαθμό. Ακολούθως, έκανα μια σύγκριση του αλγορίθμου με τρεις άλλους και παρουσίασα τα σχετικά αποτελέσματα.

Στην συνέχεια θα παρουσιάσω τα προβλήματα που προκαλούνται λόγω της συμφόρησης, τον τρόπο λειτουργίας του αλγορίθμου, τα αποτελέσματα διαφόρων σεναρίων και τέλος τα αποτελέσματα της σύγκρισης με άλλους διαδεδομένους αλγορίθμους.

Περιεχόμενα

Κεφάλαιο 1 - Εισαγωγή	1
1.1 Σκοπός Διπλωματικής Εργασίας.....	1
1.2 Μεθοδολογία που ακολουθήθηκε.....	2
1.3 Παρουσίαση Δομής Διπλωματικής Εργασίας	4
Κεφάλαιο 2 - Προβλήματα συμφόρησης στα δίκτυα αισθητήρων.....	5
2.1 Δίκτυα Αισθητήρων.....	5
2.2 Προβλήματα συμφόρησης	9
2.3 Προηγούμενη έρευνα σε θέματα συμφόρησης.....	12
2.4 Swarm Intelligence	15
Κεφάλαιο 3 - Flock-Based Congestion Control.....	17
3.1 Γενική Περιγραφή	17
3.2 Παράμετροι του αλγορίθμου	19
3.3 Εφαρμογή του αλγορίθμου	24
Κεφάλαιο 4 - Αξιολόγηση Απόδοσης Αλγορίθμου	29
4.1 Γενική Περιγραφή Σεναρίων	29
4.2 Μελέτη Παραμέτρων	31
4.3 Αλγόριθμοι Σύγκρισης.....	32
Κεφάλαιο 5 - Αξιολόγηση βασικής λειτουργίας αλγορίθμου.....	35
5.1 Ομάδα δημιουργίας πακέτων απέναντι από το sink	35
5.1.1 Ελαφριά συμφόρηση (Data Rate=25pkts/sec).....	36
5.1.2 Πρόβλημα Συμφόρησης (Data Rate=35pkts/sec).....	40
5.1.3 Σοβαρό πρόβλημα συμφόρησης (Data Rate=45pkts/sec).....	43
5.1.4 Παρατηρήσεις	46
5.1.5 Γραφική απεικόνιση κίνησης πακέτων	48
5.1.6 Σύγκριση αλγορίθμων.....	50
5.1.7 Σύνοψη.....	53
5.2 Ομάδα δημιουργίας πακέτων στο πλάι	54
5.2.1 Πρόβλημα Συμφόρησης (Data Rate=25pkts/sec)	56

5.2.2	Σοβαρό Πρόβλημα Συμφόρησης (Data Rate=35pkts/sec).....	58
5.2.3	Πολύ Σοβαρό πρόβλημα συμφόρησης (Data Rate=45pkts/sec)	62
5.2.4	Παρατηρήσεις.....	66
5.2.5	Γραφική απεικόνιση κίνησης πακέτων	67
5.2.6	Σύγκριση αλγορίθμων	69
5.2.7	Σύνοψη	71
5.3	Δύο ομάδες δημιουργίας πακέτων απέναντι από το sink	72
5.3.1	Ελαφριά Συμφόρηση (Data Rate=25pkts/sec)	73
5.3.2	Πρόβλημα Συμφόρησης (Data Rate=35pkts/sec)	75
5.3.3	Σοβαρό πρόβλημα συμφόρησης (Data Rate=45pkts/sec).....	78
5.3.4	Παρατηρήσεις.....	81
5.3.5	Γραφική απεικόνιση κίνησης πακέτων	83
5.3.6	Σύγκριση αλγορίθμων	86
5.3.7	Σύνοψη	88
Κεφάλαιο 6 - Αξιολόγηση ευρωστίας του αλγορίθμου		89
6.1	Κατάρρευση κόμβων στην μέση του δικτύου	89
6.1.1	Ελαφριά Συμφόρηση (Data Rate=25pkts/sec)	91
6.1.2	Πρόβλημα Συμφόρησης (Data Rate=35pkts/sec)	93
6.1.3	Σοβαρό πρόβλημα συμφόρησης (Data Rate=45pkts/sec).....	95
6.1.4	Παρατηρήσεις.....	97
6.1.5	Γραφική απεικόνιση κίνησης πακέτων	99
6.1.6	Σύγκριση αλγορίθμων	102
6.1.7	Σύνοψη	106
6.2	Κατάρρευση κόμβων δημιουργώντας καμπύλη	107
6.2.1	Γενικά αποτελέσματα.....	107
6.2.2	Γραφική απεικόνιση κίνησης πακέτων	108
6.2.3	Σύγκριση αλγορίθμων	111
6.2.4	Σύνοψη	113
6.3	Αντιμετώπιση συμφόρησης διαπερνώντας τους πλαϊνούς νεκρούς κόμβους.....	114
6.3.1	Γραφική απεικόνιση κίνησης πακέτων	115
6.3.2	Σύγκριση αλγορίθμων	118
6.3.3	Σύνοψη	120

Κεφάλαιο 7 - Συμπεράσματα	121
7.1 Συνολικά Συμπεράσματα	121
7.2 Μελλοντική Εργασία	124
Βιβλιογραφία	125

Λίστα Ακρωνυμίων

CAwR	Congestion Aware routing protocol
AODV	Ad Hoc on Demand Distance Vector protocol
PDR	Packet Delivery Ratio
CSMA	Carrier Sense Multiple Access
RREP	Route Reply
RREQ	Route Request
WSN	Wireless Sensor Network
GPS	Global Positioning System

Κεφάλαιο 1

Εισαγωγή

1.1	Σκοπός Διπλωματικής Εργασίας	1
1.2	Μεθοδολογία που ακολουθήθηκε.....	2
1.3	Παρουσίαση Δομής Διπλωματικής Εργασίας	4

1.1 Σκοπός Διπλωματικής Εργασίας

Τα τελευταία χρόνια έχει ανθίσει το ενδιαφέρον γύρω από τα δίκτυα αισθητήρων αφού έχει γνωστοποιηθεί η σημαντικότητα και χρησιμότητα τους. Λόγω της ιδιομορφίας τους όμως, τα διάφορα πρωτόκολλα που χρησιμοποιούνται με επιτυχία για τα παραδοσιακά ασύρματα δίκτυα φαίνεται να μην είναι αποτελεσματικά κι έτσι κρίστηκε αναγκαία η ανάπτυξη νέων πρωτοκόλλων που θα ικανοποιούν τους περιορισμούς των δικτύων αισθητήρων.

Μέσα από αυτή τη διπλωματική εργασία κλήθηκα να μελετήσω τα δίκτυα αισθητήρων και να προσπαθήσω να προσδιορίσω πως μπορεί να ανιχνευτεί το πρόβλημα της συμφόρησης. Είχα την ευκαιρία να μελετήσω διάφορους αλγόριθμους οι οποίοι είναι εμπνευσμένοι από την φύση και να μάθω αρκετά πράγματα για αυτή την προσέγγιση αφού όπως φαίνεται τέτοιου είδους πρωτόκολλα δίνουν αρκετά ικανοποιητικά αποτελέσματα και στο μέλλον είναι πολύ υποσχόμενα. Πιο συγκεκριμένα, μελέτησα ένα νέο αλγόριθμο ο οποίος μιμείται την συμπεριφορά των πουλιών καθώς κινούνται σε κοπάδια. Ο αλγόριθμος προσπαθεί να εντοπίσει το πρόβλημα της συμφόρησης στο δίκτυο και ταυτόχρονα προσπαθεί να αποφύγει τις προβληματικές περιοχές. Κάνοντας διάφορα πειράματα, τρέχοντας διάφορα σενάρια κατάφερα τελικά να προσδιορίσω

ποιες παράμετροι μπορούν να συμβάλουν στον εντοπισμό της συμμόρφωσης και σε ποιο βαθμό. Πιο πολλές λεπτομέρειες, τόσο για τα δίκτυα αισθητήρων και τη συμμόρφωση, όσο και για τον σχετικό αλγόριθμο θα παρουσιάσω στα επόμενα κεφάλαια.

1.2 Μεθοδολογία που ακολουθήθηκε

Για την ανάπτυξη της διπλωματικής μου εργασίας χρειάστηκε πρώτα να μελετήσω πολλά άρθρα σχετικά με τα δίκτυα αισθητήρων για να μπορέσω να κατανοήσω τον τρόπο λειτουργίας τους και τις σημαντικές διαφορές που έχουν από τα παραδοσιακά κινητά ασύρματα δίκτυα. Ακολούθως, μελέτησα σε βάθος τον αλγόριθμο ελέγχου συμμόρφωσης Flock-CC, ο οποίος μιμείται την συμπεριφορά των πουλιών, προκειμένου να κατανοήσω τις διάφορες πτυχές του για να μπορέσω αργότερα, τρέχοντας διάφορα σενάρια, να καθορίσω τις ιδανικές τιμές των διαφόρων παραμέτρων που χρησιμοποιούνται για έλεγχο συμμόρφωσης.

Στο πρώτο σενάριο που μελέτησα όλοι οι κόμβοι ήταν ενεργοποιημένοι και δημιουργούσαν πακέτα τα οποία και προωθούσαν προς το sink. Είδα ότι υπήρχαν σοβαρά προβλήματα συμμόρφωσης και πως στην καλύτερη περίπτωση οι κόμβοι έπρεπε να ξεκινούν να στέλλουν τα πακέτα με κάποια χρονική διαφορά ο ένας από τον άλλο. Βλέποντας λοιπόν, ότι δεν μπορούσε πλέον να βελτιωθεί το συγκεκριμένο σενάριο, αφού πολύ απλά το δίκτυο έφτανε σε οριακό σημείο, αποφασίσαμε πως έπρεπε να μελετήσω κάποια άλλα σενάρια όπου μια μικρή ομάδα κόμβων, η οποία βρίσκεται σε διαφορετικές θέσεις κάθε φορά, δημιουργεί τα πακέτα.

Συνολικά μελέτησα πέντε σενάρια και κατάφερα να καταλήξω σε κάποιες πολύ λογικές τιμές για τις παραμέτρους μου. Ο τρόπος με τον οποίο επεξεργάζομαι τα δεδομένα αναλύεται πιο κάτω. Χρησιμοποιώντας τον Μορφέα (server Πανεπιστημίου) έτρεξα όλους τους δυνατούς συνδυασμούς των παραμέτρων, κάθε ένα από αυτούς 10 φορές. Ακολούθως, χρησιμοποιώντας το μαθηματικό πρόγραμμα της Matlab, επεξεργάζομαι το αρχείο με τα αρχικά ακατέργαστα αποτελέσματα. Υπολόγιζα τον μέσο όρο του κάθε πιθανού συνδυασμού των παραμέτρων και αποθήκευα τις τιμές σε ένα νέο αρχείο. Από

αυτό το σημείο και μετά ξεκινούσε η μελέτη των αποτελεσμάτων για να φανεί ο καλύτερος συνδυασμός παραμέτρων. Αρχικά, χρησιμοποιούσα το πρόγραμμα Fluxviz το οποίο επέλεγε και παρουσίαζε τους 10 καλύτερους συνδυασμούς παραμέτρων. Σε περίπτωση που και για τα τρία data rates έβλεπα πως υπήρχε κάποια παράμετρος με την ίδια τιμή ως καλύτερη, το εκμεταλλευόμουν και διατηρούσα σταθερή την εκάστοτε παράμετρο για να μελετήσω τις υπόλοιπες. Πάντοτε έβλεπα γραμμή προς γραμμή το αρχείο εξόδου του προγράμματος Matlab για σιγουρευόμουν πως ήμουν στον σωστό δρόμο. Βάσει των όσων παρατηρούσα στο αρχείο επέλεγα τον τρόπο με τον οποίο θα παρουσίαζα τα αποτελέσματα, δηλαδή ποιες παραμέτρους θα διατηρούσα σταθερές προκειμένου να μελετήσω τις υπόλοιπες που δεν είχαν και τόσο ξεκάθαρα αποτελέσματα. Έπειτα, παρουσίαζα τα αποτελέσματα ξεκινώντας από ένα πιο γενικό επίπεδο και επιλέγοντας κάθε φορά τις καλύτερες περιπτώσεις κι απορρίπτοντας τις χειρότερες, κατέληγα σε μια γραφική όπου παρουσιάζονταν τα καλύτερα αποτελέσματα.

Αφού εντόπιζα τον καλύτερο συνδυασμό παραμέτρων χρησιμοποιούσα ένα πρόγραμμα στο Eclipse το οποίο μπορούσε να δημιουργήσει μια γραφική αναπαράσταση της κίνησης των πακέτων και του ποσοστού ενέργειας που απέμενε στον κάθε κόμβο μετά το τέλος της προσομοίωσης. Με αυτό το τρόπο μπορούσα να δω όχι μόνο τις περιοχές όπου παρουσιαζόταν πρόβλημα συμφόρησης, αλλά και τον τρόπο που επέλεγαν να κινηθούν τα πακέτα αναλόγως της κατάστασης του δικτύου και τον τρόπο με τον οποίο αναπροσαρμοζόταν ο αλγόριθμος έπειτα από κάποια απρόσμενη αλλαγή στην τοπολογία του δικτύου.

Τέλος, έχοντας καταλήξει στον καλύτερο συνδυασμό παραμέτρων έκανα μια σύγκριση του Flock-CC με κάποιους άλλους γνωστούς αλγορίθμους (No congestion, CAwR, AODV) και παρουσίαζα γραφικές παραστάσεις για το PDR, τις συνολικές καθυστερήσεις και την δαπανούμενη ενέργεια.

1.3 Παρουσίαση Δομής Διπλωματικής Εργασίας

Σε αυτό το σημείο θα παρουσιάσω περιληπτικά την δομή της διπλωματικής εργασίας ξεκινώντας από το δεύτερο κεφάλαιο όπου γίνεται μια αναφορά στα δίκτυα αισθητήρων, την σημασία που έχουν στις μέρες μας, την διαφορά που έχουν από τα παραδοσιακά δίκτυα και άλλες πολύ σημαντικές πληροφορίες. Αμέσως μετά, αναφέρομαι στο πρόβλημα της συμφόρησης και τις επιπτώσεις που μπορεί να επιφέρει σε ένα δίκτυο αισθητήρων και συνεπώς τους λόγους ανάπτυξης νέων πρωτοκόλλων που θα μπορούν να την εντοπίζουν και αν την αποφεύγουν με αποτελεσματικό τρόπο. Έπειτα, κάνω μια σύντομη έκθεση για το Swarm Intelligence από το οποίο είναι εμπνευσμένος ο αλγόριθμος που μελετώ. Το αμέσως επόμενο κεφάλαιο, που είναι το τελευταίο όσον αφορά το θεωρητικό μέρος, αφορά την περιγραφή του αλγορίθμου Flock-CC. Αναφέρω την όλη ιδέα, από που είναι εμπνευσμένος, εξηγώ τις διάφορες εξισώσεις που χρησιμοποιεί και τις παραμέτρους τους. Ακολούθως, ξεκινά η παρουσίαση των αποτελεσμάτων για τα πέντε σενάρια που μελέτησα. Τα σενάρια έχουν χωριστεί σε δύο ξεχωριστά κεφάλαια διότι στο ένα μελετούνται περισσότερο οι τιμές των παραμέτρων του αλγορίθμου ενώ στο άλλο γίνεται μια μελέτη σχετικά με την ευρωστία του αλγορίθμου χρησιμοποιώντας σενάρια όπου κάποιοι κόμβοι καταρρέουν. Το κάθε σενάριο περιλαμβάνει τα αποτελέσματα για τα τρία data rates, γενικές παρατηρήσεις για κάθε μια από τις παραμέτρους, μια σύγκριση του αλγορίθμου με κάποιους άλλους και τέλος, τα στιγμιότυπα της κίνησης των πακέτων. Η διπλωματική μου εργασία τελειώνει με μια παρουσίαση των συνολικών συμπερασμάτων και κάποιων προτάσεων για μελλοντική εργασία.

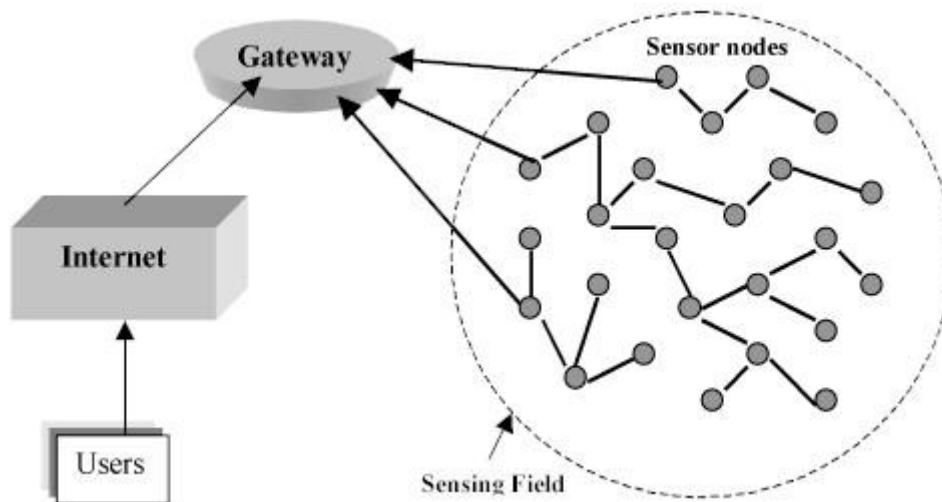
Κεφάλαιο 2

Προβλήματα συμφόρησης στα δίκτυα αισθητήρων

2.1	Δίκτυα Αισθητήρων	5
2.2	Προβλήματα συμφόρησης	9
2.3	Προηγούμενη έρευνα σε θέματα συμφόρησης	12
2.4	Swarm Intelligence.....	15

2.1 Δίκτυα Αισθητήρων

Τα δίκτυα αισθητήρων αποτελούν ένα σχετικά νέο θέμα για την επιστήμη της Πληροφορικής, γύρω από το οποίο διεξάγεται μεγάλη έρευνα. Χρησιμοποιούνται ευρέως σε πάρα πολλές εφαρμογές ζωτικής σημασίας όπως για παράδειγμα σε νοσοκομεία για παρακολούθηση ασθενών, μέτρηση ποσοστού μόλυνσης, θερμοκρασίας και πίεσης, παρακολούθηση δύσβατων περιοχών, έλεγχο υγρασίας σε τεράστιες εκτάσεις αγροκτημάτων, έλεγχο πυρκαγιάς ή πλημμύρας, έλεγχο κυκλοφοριακής συμφόρησης, ακόμη και για παρακολούθηση κάποιου πεδίου μάχης. Είναι απαραίτητο να μπορούν να λειτουργούν αποδοτικά, χωρίς προβλήματα, για εκτεταμένα χρονικά διαστήματα σε αναξιόπιστα περιβάλλοντα που μεταβάλλονται πολύ συχνά. [1]



(<http://www.acm.org/crossroads/xrds9-4/gfx/sn1.jpg>)

Σχήμα 2.1: Δομή δικτύου αισθητήρων

Στο σχήμα 2.1 παρουσιάζεται η δομή ενός απλού δικτύου αισθητήρων. Τα δίκτυα αισθητήρων χρησιμοποιούν μικροσκοπικά sensor nodes, τα οποία συνήθως έχουν χαμηλό κόστος, καταναλώνουν μικρά ποσοστά ενέργειας, έχουν περιορισμένο bandwidth για επικοινωνία και υπολογιστικές δυνατότητες. Αυτοί οι κόμβοι αλληλεπιδρούν με το περιβάλλον και κάνουν διάφορες μετρήσεις μέσω των αισθητήρων που διαθέτουν. Συνεργάζονται προκειμένου να καταφέρουν να φέρουν εις πέρας την αποστολή τους, επεξεργάζονται ελαφρώς τα δεδομένα κι ακολούθως στέλλουν τα αποτελέσματα τους σε ένα ή περισσότερα κεντρικά σημεία (sink, gateway) όπου θα τύχουν περαιτέρω επεξεργασίας. Έρευνες έχουν δείξει πως είναι απαραίτητο όπως οι κόμβοι εκτελούν τοπικούς υπολογισμούς, αφού αυτό συμβάλλει στην μεγαλύτερη εξοικονόμηση ενέργειας, παρά αν όλα τα ακατέργαστα δεδομένα στέλλονταν απευθείας στο sink. Το sink επικοινωνεί με κάποια εφαρμογή χρησιμοποιώντας είτε το διαδίκτυο και από εκεί γίνεται η αλληλεπίδραση με τους χρήστες. [9]

Τα sensor nodes τοποθετούνται πολύ κοντά το ένα στο άλλο και συνήθως οι θέσεις τους είναι τυχαίες. Αρκετές φορές μπορεί αφεθούν ελεύθερα από κάποιο αεροπλάνο όταν χρειάζεται να μελετηθεί κάποια δύσβατη περιοχή. Τοποθετούνται είτε μέσα στο φαινόμενο το οποίο μελετούν, είτε πολύ κοντά του. Συχνά, είναι αδύνατη και

ασύμφορη η αλλαγή της μπαταρίας κάποιου κόμβου κι έτσι προτιμάται απλά να αντικατασταθεί ο κόμβος με ένα νέο. Συνεπώς, η διάρκεια ζωής ενός sensor node εξαρτάται από την μπαταρία του. Έχουν δημιουργηθεί κάποια sensors τα οποία χρησιμοποιούν μια δευτερεύουσα μπαταρία η οποία επαναφορτίζεται αυτόματα εκμεταλλευόμενη κάποια ανανεώσιμη πηγή ενέργειας. Παρόλα αυτά η εξοικονόμηση ενέργειας εξακολουθεί να είναι ένα θέμα μείζονος.

Ένα δίκτυο αισθητήρων χρειάζεται να ικανοποιεί ένα σύνολο ιδιοτήτων που είναι απαραίτητες για την σωστή και αποδοτική λειτουργία του. Χρειάζεται να έχει οργανωτικές δυνατότητες, δηλαδή οι κόμβοι να μπορούν να αναπροσαρμόζονται ανάλογα με την κατάσταση που επικρατεί στο δίκτυο. Σε περίπτωση που καταρρεύσει κάποιος κόμβος, οι υπόλοιποι πρέπει να ενημερωθούν έγκαιρα για να βρεθούν εναλλακτικά μονοπάτια ή σε περίπτωση που οι κόμβοι είναι κινητοί κάποιος από αυτούς να μετακινηθεί προκειμένου να καλυφθεί το κενό που δημιουργήθηκε από την κατάρρευση του προηγούμενου κόμβου. Επίσης, ένα δίκτυο αισθητήρων πρέπει να έχει την δυνατότητα αποθήκευσης πληροφοριών και επικοινωνίας μέσω ασύρματου μέσου. Συνεπώς, χρειάζεται να υποστηρίζει πολλές λειτουργίες οι οποίες πρέπει να γίνονται με τον καλύτερο δυνατό τρόπο παρά τους περιορισμούς που υπάρχουν.

Κάποια από τα σημαντικότερα στοιχεία που χαρακτηρίζουν ένα δίκτυο αισθητήρων είναι η μικρή εμβέλεια στην οποία μπορούν να κάνουν broadcast πακέτα οι διάφοροι κόμβοι, η δρομολόγηση χρησιμοποιώντας πολλαπλά hops, η πυκνή τοποθέτηση των κόμβων και η συνεργασία για την επίτευξη του στόχου. Συχνά χρειάζεται να μπορούν να αντιμετωπίσουν διάφορα προβλήματα όπως την αλλαγή της τοπολογίας του δικτύου, την αναξιόπιστη επικοινωνία λόγω της χρήσης του ασύρματου μέσου, χάσιμο πακέτων λόγω έλλειψης χώρου στις ουρές των κόμβων είτε λόγω συγκρούσεων. Επίσης, οι περιορισμένες δυνατότητες όσον αφορά τα ενεργειακά αποθέματα, την ισχύ μετάδοσης, την μνήμη, και γενικά η χαμηλή υπολογιστική δύναμη των κόμβων δυσχεραίνουν το έργο ενός δικτύου αισθητήρων. Λόγω των πιο πάνω προβλημάτων τα δίκτυα αισθητήρων παρουσιάζουν αρκετές διαφορές από άλλα ad hoc ασύρματα δίκτυα και είναι ευπαθή σε διάφορα είδη σφαλμάτων τα οποία θα παρουσιάσω στην συνέχεια.

Ανάλογα με την τοπολογία του δικτύου και του πρωτοκόλλου δρομολόγησης που χρησιμοποιείται κάποιοι κόμβοι μπορεί να λάβουν μεγαλύτερο φορτίο δεδομένων συγκριτικά με άλλους και να προκληθούν προβλήματα συμφόρησης τα οποία εκλαμβάνονται είτε ως χάσιμο των πακέτων είτε ως αύξηση στις καθυστερήσεις. Αυτό είναι ένα σοβαρό ζήτημα διότι για κάποιες εφαρμογές ζωτικής σημασίας χρειάζεται εγγύηση πως τα πακέτα θα φθάσουν στον προορισμό τους και μάλιστα έγκαιρα. Επίσης, λόγω της αναξιопιστίας του ασύρματου μέσου και του μεταβαλλόμενου περιβάλλοντος, υπάρχει περίπτωση να απορριφθούν τα μεταδιδόμενα πακέτα ή να ληφθούν με κάποιο λάθος. Συχνά, διάφοροι κόμβοι μπορεί να καταρρεύσουν είτε λόγω κάποιας αλλαγής στο περιβάλλον είτε διότι έλειψαν τα ενεργειακά τους αποθέματα. Ως επακόλουθο μπορεί να είναι η καταστροφή των συνδέσεων ανάμεσα σε κάποιους κόμβους (link failures). Τέλος, έπειτα από προσθήκη ή καταστροφή κάπου συνδέσμου (link) τα δρομολόγια μπορεί να αλλάξουν και επομένως οι πληροφορίες που έχουν οι κόμβοι σχετικά με τους γείτονες τους να είναι παλιές. Εναπόκειται στο transport protocol κατά πόσον τα πακέτα που είχαν προωθηθεί από κάποιο παλιό δρομολόγιο θα απορριφθούν ή αν θα σταλούν αλλά με κάποια καθυστέρηση μέχρι να βρεθεί το δρομολόγιο προς τον προορισμό.

Λόγω των μεγάλων διαφορών ανάμεσα στα παραδοσιακά ασύρματα δίκτυα και τα WSN, δεν έχουν βρεθεί ιδανικοί αλγόριθμοι δρομολόγησης πακέτων και ελέγχου συμφόρησης. Επίσης, εξαιτίας των περιορισμένων δυνατοτήτων τους δεν μπορούν να χρησιμοποιηθούν αποδοτικά οι καθιερωμένοι αλγόριθμοι που χρησιμοποιούνται με επιτυχία στα ασύρματα κινητά δίκτυα. Οι διάφοροι κόμβοι καλούνται όχι μόνο να συλλέγουν πληροφορίες και να τις επεξεργάζονται αλλά να μπορούν να προωθούν και δεδομένα άλλων κόμβων. Επίσης, οι κόμβοι πρέπει να μπορούν να λειτουργούν δυναμικά για να αντεπεξέλθουν στις δύσκολες συνθήκες που μπορεί να προκύψουν. Χρειάζεται λοιπόν να αναπτυχθούν νέα πρωτόκολλα που να μπορούν να ξεπεράσουν τα προβλήματα των τοπολογικών αλλαγών ή κατάρρευσης κόμβων, καθώς επίσης και όλα τα προβλήματα που απορρέουν από την χρήση ασύρματου δικτύου.

Επιπλέον, οι αλγόριθμοι χρειάζεται να είναι εύρωστοι, ευέλικτοι και να μπορούν να προσαρμόζονται αυτόματα ανάλογα με τις συνθήκες που επικρατούν στο δίκτυο. Τα

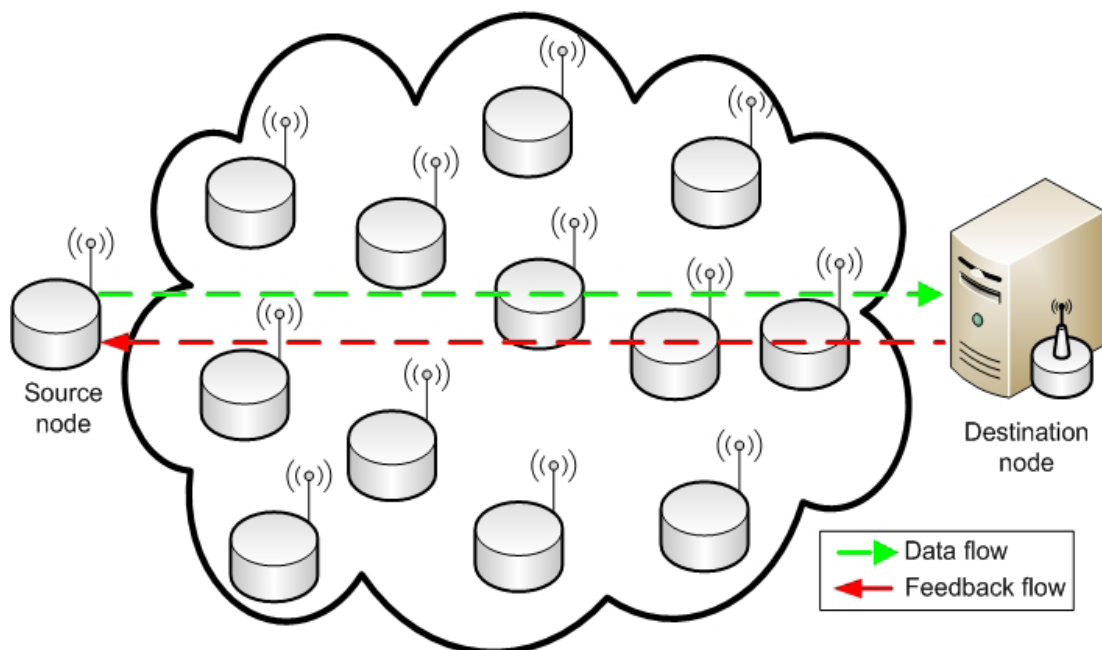
transport protocols αναμένουμε πως θα μειώνουν την συμφόρηση, το χάσιμο πακέτων, τις καθυστερήσεις και θα γίνεται δίκαιη δέσμευση όσων αφορά το bandwidth και θα υπάρχει εγγύηση για την μεταφορά των δεδομένων διότι πολλές φορές οι πληροφορίες που μεταφέρονται είναι ζωτικής σημασίας και πρέπει να φτάσουν στον προορισμό τους έγκαιρα. Γνωρίζοντας πως το κόστος μετάδοσης πληροφοριών απαιτεί μεγάλα ποσά ενέργειας, αναμένουμε πως οι αλγόριθμοι θα προσπαθούν να ελαχιστοποιήσουν την ανταλλαγή αχρείαστων πληροφοριών.

2.2 Προβλήματα συμφόρησης

Αρχικά θα ξεκινήσω με μια σύντομη περιγραφή του όρου συμφόρηση. Η συμφόρηση θεωρούμε πως προκαλείται όταν το traffic load που εισέρχεται στο δίκτυο είναι μεγαλύτερο από αυτό που εξέρχεται. Είναι ένα πάρα πολύ σημαντικό πρόβλημα για όλα τα δίκτυα αλλά κυρίως για τα WSN αφού προκαλεί σπατάλη ενέργειας, μείωση του throughput, αύξηση των συγκρούσεων και συνεπώς και των αναμεταδόσεων στο MAC layer, αύξηση των καθυστερήσεων και συχνά ακόμη και χάσιμο πληροφοριών και όλα αυτά έχουν ως επακόλουθο την μείωση της ποιότητας υπηρεσίας, την μείωση της συνολικής διάρκειας ζωής του δικτύου και την αποσύνθεση του σε μικρές ατομικές μονάδες. Στα δίκτυα αισθητήρων θεωρείται αναγκαία η επιτυχημένη κι έγκυρη αποστολή των δεδομένων. Στα παραδοσιακά ενσύρματα δίκτυα, θεωρούσαμε ως ένδειξη συμφόρησης το χάσιμο πακέτων και ο έλεγχος για συμφόρηση γινόταν μεταξύ του αποστολέα και του παραλήπτη. Δυστυχώς όμως αυτή η προσέγγιση προκαλεί αυξημένες καθυστερήσεις και ψηλά error rates και δεν είναι καθόλου αποτελεσματική αφού το ασύρματο περιβάλλον μπορεί από μόνο του να προκαλέσει χάσιμο πακέτων για διάφορους λόγους. [3]

Η συμφόρηση μπορεί να κατηγοριοποιηθεί σε δύο βασικούς τύπους, την συμφόρηση που προκαλείται στους κόμβους και την συμφόρηση στους διάφορους συνδέσμους (links). Όταν η εισερχόμενη ροή πακέτων είναι μεγαλύτερη από την εξερχόμενη τότε αναγκαστικά τα πακέτα αποθηκεύονται προσωρινά στις ουρές των κόμβων οι οποίες έχουν πεπερασμένη χωρητικότητα. Σε περίπτωση που γεμίσει η ουρά του κόμβου και

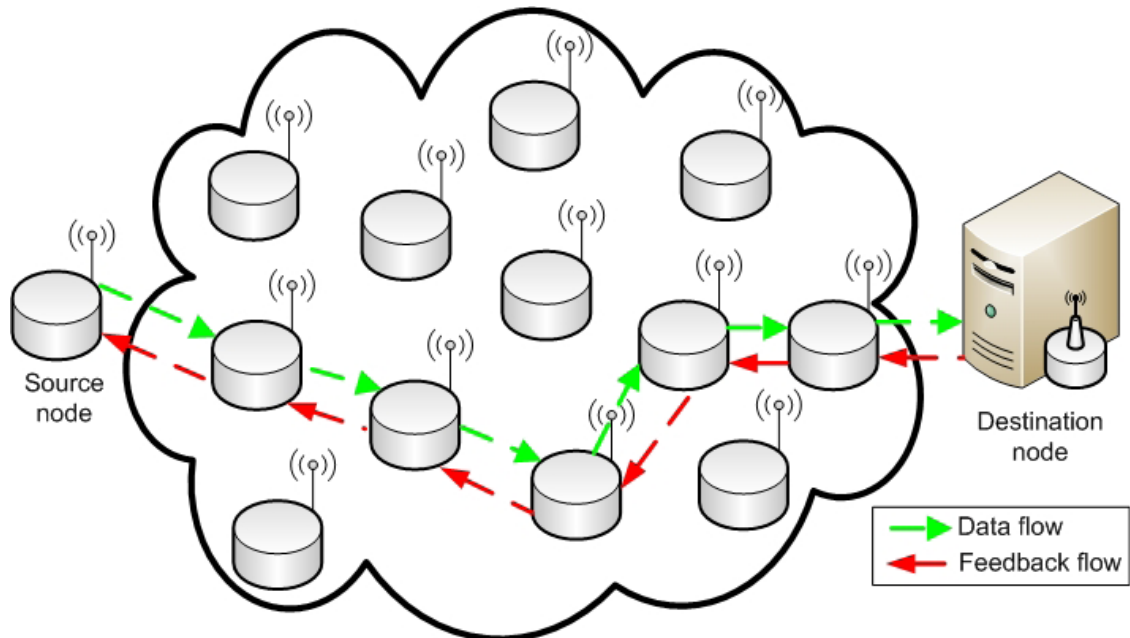
συνεχίσει να λαμβάνει πακέτα τότε παρουσιάζεται το πρόβλημα της υπερχειλίσης που οδηγεί στο χάσιμο των πακέτων και αυξάνει την καθυστέρηση λόγω ουράς (queuing delay). Συνεπώς, μπορούμε να θεωρήσουμε πως ένα κόμβος παρουσιάζει προβλήματα συμφόρησης όταν η εισερχόμενη ροή πληροφοριών είναι μεγαλύτερη της εξερχόμενης ή όταν ο αριθμός των πακέτων που μπαίνουν στην ουρά του κόμβου αυξάνεται σε σημείο που πλησιάζει να γεμίσει. Η συμφόρηση στους συνδέσμους οφείλεται στο περιορισμένο bandwidth των ασύρματων δικτύων, στην multi-hop φύση τους και κυρίως στην κοινή χρήση των links. Λόγω του ότι οι κόμβοι προσπαθούν να κερδίσουν την πρόσβαση σε ένα link όσο το δυνατό πιο σύντομα, προκαλούνται παρεμβολές με γειτονικούς κόμβους που προσπαθούν ταυτόχρονα να χρησιμοποιήσουν το κανάλι.



Σχήμα 2.2: Τεχνική End-to-End

Οι διάφορες τεχνικές που χρησιμοποιούνται για έλεγχο συμφόρησης μπορούν να ενταχθούν σε δύο βασικές κατηγορίες: (α) End-to-end και (β) Hop-by-hop. Ο end-to-end έλεγχος συμφόρησης (βλέπε σχήμα 2.2) εκτελείται κυρίως από τα πρωτόκολλα μεταφοράς που εφαρμόζονται στον αποστολέα και στον παραλήπτη. Οι κόμβοι στα άκρα του δικτύου μπορούν να αντιληφθούν πως υπάρχει πρόβλημα συμφόρησης μέσα από κάποιες πληροφορίες που συλλέγουν (round trip delay, jitter, χάσιμο πακέτων...)

από τον παραλήπτη. Οι ενδιάμεσοι κόμβοι δεν εμπλέκονται στην διαδικασία εντοπισμού της συμφόρησης και δεν αντιλαμβάνονται τι γίνεται.



Σχήμα 2.3: Τεχνική Hop-by-hop

Σε ένα hop-by-hop μοντέλο (βλέπε σχήμα 2.3), όλοι οι κόμβοι που περιλαμβάνονται στο μονοπάτι μεταξύ του αποστολέα και του παραλήπτη εμπλέκονται στην διαδικασία εντοπισμού προβλήματος συμφόρησης. Ο κάθε κόμβος λαμβάνει τοπικές αποφάσεις στηριζόμενες στις πληροφορίες που διαθέτει (π.χ. κατάσταση στις ουρές, channel load...). Το πλεονέκτημα αυτής της προσέγγισης έναντι του end-to-end, είναι πως επιτρέπει τον πολύ γρήγορο εντοπισμό του προβλήματος και συνεπώς της έγκαιρης αντιμετώπισης του. Αυτό με την σειρά του συμβάλλει στην μείωση των καθυστερήσεων και προπαντός στην εξοικονόμηση ενέργειας, κάτι που είναι πολύ σημαντικό για τα δίκτυα αισθητήρων. Επιπλέον, δεδομένου πως όλοι οι κόμβοι στέλλουν τα δεδομένα τους σε ένα μικρό αριθμό από κόμβους sink, συνεπάγεται πως χρησιμοποιώντας την end-to-end προσέγγιση αυξάνονται και οι υπολογιστικές πράξεις που χρειάζεται να διεκπεραιώσουν.

Λόγω της περιορισμένης δυνατότητας των δικτύων αισθητήρων είναι αναγκαίο να χρησιμοποιείται κάποια αποκεντρωμένη προσέγγιση ανίχνευσης και αποφυγής της

συμφόρησης η οποία θα μπορεί εύκολα, γρήγορα και αποτελεσματικά να ενημερώνει τους κόμβους για τις περιοχές που πρέπει να αποφύγουν. Πρέπει λοιπόν, μια τέτοια προσέγγιση να μπορεί να εφαρμόζεται στον κάθε κόμβο ξεχωριστά ο οποίος, δεδομένου των πληροφοριών που έχει για τους κόμβους που βρίσκονται μέσα στο πεδίο αντίληψης του, να μπορεί να επιλέγει κόμβους οι οποίοι έχουν τις καλύτερες προοπτικές για αποφυγή συμφόρησης. Γι' αυτούς τους λόγους φαίνεται πως οι τεχνικές τύπου hop-by-hop είναι καλύτερες για τα δίκτυα αισθητήρων.

2.3 Προηγούμενη έρευνα σε θέματα συμφόρησης

Στο παρελθόν οι έρευνες εστίαζαν κυρίως σε θέματα δρομολόγησης, τοπολογίας, εξοικονόμησης ενέργειας και MAC. Δεν είχε δοθεί τόση πολλή σημασία στην ποιότητα υπηρεσίας που προσφερόταν μέχρι που εμφανίστηκαν τα δίκτυα αισθητήρων με τις εφαρμογές ζωτικής σημασίας. Από τότε παρουσιάστηκε μεγάλη ανάπτυξη γύρω από τους μηχανισμούς ελέγχου ποιότητας υπηρεσίας για την αποφυγή προβλημάτων συμφόρησης. [2]

Μέσα από την έρευνα έχουν προταθεί διάφορες προσεγγίσεις ελέγχου συμφόρησης οι οποίες στηρίζονται στην διαχείριση του traffic (π.χ. αναπροσαρμογή του ρυθμού αποστολής δεδομένων ανάλογα με την κατάσταση του δικτύου και τις τοπολογικές αλλαγές, δρομολόγηση από πολλαπλά μονοπάτια...) και των πόρων του δικτύου. Οι πλείστες μέθοδοι στηρίζονται στον έλεγχο της εισερχόμενης ροής δεδομένων στο δίκτυο, κάτι το οποίο μπορεί να οδηγήσει στη μείωση της προσφερόμενης ποιότητας υπηρεσίας και πιθανότατα κατά τις χρονικές στιγμές που είναι πιο απαραίτητη.

Κάποιες άλλες προσεγγίσεις περιλαμβάνουν την δημιουργία ομάδων (clusters) κάτι το οποίο απαιτεί τον καθορισμό ρόλων στους διάφορους κόμβους και την εφαρμογή περαιτέρω μηχανισμών. Επίσης, οι περιοχές κοντά στις διάφορες ομάδες είναι πιθανόν να παρουσιάζουν προβλήματα συμφόρησης.

Η χρήση πολλαπλών μονοπατιών είναι μια πολύ καλή μέθοδος περιορισμού της συμφόρησης χωρίς να μειώνεται η ποιότητα υπηρεσίας. Το πρωτόκολλο Congestion Aware Routing, μπορεί να ανακαλύψει με δυναμικό τρόπο μια περιοχή συμφόρησης και να δρομολογήσει τα πακέτα υψηλής προτεραιότητας μέσα από την περιοχή ενώ τα πακέτα χαμηλής προτεραιότητας προωθούνται γύρω από την συγκεκριμένη περιοχή. Ένα από τα βασικά προβλήματα αυτής της προσέγγισης είναι η κατηγοριοποίηση των πακέτων σε υψηλή και χαμηλή προτεραιότητα.

Το πρωτόκολλο δρομολόγησης Biased Geographical Routing αντιμετωπίζει την συμφόρηση συνδυάζοντας τον έλεγχο στις ουρές κι ελέγχοντας τον ρυθμό αποστολής των πακέτων. Το μειονέκτημα του αλγορίθμου είναι πως δεν λαμβάνει υπόψη τις καταστάσεις φόρτου (load conditions) της ουράς του κάθε κόμβου και της κατάστασης του δικτύου κατά μήκος του μονοπατιού και συνεπώς οι τυχαία επιλεγμένες μεταβάσεις μπορεί να προκαλέσουν συμφόρηση είτε λόγω υπερχειλίσης των ουρών των κόμβων είτε λόγω συγκρούσεων σε επίπεδο MAC.

Είναι σημαντικό να αναφερθεί πως οι δύο προαναφερθέντες αλγόριθμοι, CAwR και Biased Geographical Routing, χρησιμοποιούν πληροφορίες σχετικά με την τοποθεσία των κόμβων τις οποίες λαμβάνουν από το σύστημα GPS και με αυτές τις πληροφορίες μπορούν να εντοπίσουν περιοχές με προβλήματα συμφόρησης. Το σύστημα GPS είναι κάπως ακριβό κι έτσι δεν μπορεί να χρησιμοποιηθεί για την κατασκευή μικρών και φθηνών αισθητήρων. Επίσης, αυτό το σύστημα μπορεί να δουλέψει σε ελεύθερο περιβάλλον δεδομένου πως δεν υπάρχουν διαφόρων ειδών εμπόδια.

Τέλος, ο αλγόριθμος Traffic-Aware Dynamic Routing προσπαθεί να προωθήσει τα πακέτα γύρω από τις περιοχές με συμφόρηση και διασκορπίζει τα πακέτα που έχουν μεγάλες απαιτήσεις σε πολλαπλά μονοπάτια των οποίων οι κόμβοι δεν έχουν πολύ φόρτο, δεν παρουσιάζουν προβλήματα συμφόρησης και είναι αδρανείς (idle). Αυτός ο αλγόριθμος χρησιμοποιεί το βάθος και το κανονικοποιημένο μήκος της ουράς για να αναγκάσει τα πακέτα να αποφύγουν τα εμπόδια που προκαλούνται λόγω συμφόρησης και να κατευθυνθούν προς το sink. Ένα μειονέκτημα είναι πως αγνοεί τα προβλήματα που διέπουν το ασύρματο δίκτυο και τα οποία μπορούν να προκαλέσουν χάσιμο

πακέτων για εκτεταμένη χρονική περίοδο. Επιπλέον, οι κατασκευαστές του αλγορίθμου αξιολόγησαν το πρωτόκολλο υποθέτοντας ότι το MAC layer ήταν ιδανικό, δηλαδή ότι μπορεί να παρέχει σταθερό ασύρματο δίκτυο χωρίς να προκαλούνται συγκρούσεις, κάτι το οποίο δεν ευσταθεί στην πράξη.

Διάφορες μελέτες οι οποίες έχουν διεξαχθεί από τους C. Wan, S. Eisenman και A. Campbell, καθώς επίσης και από τους B. Hull, K. Jamieson και H. Balakrishnan, έχουν αποδείξει πως τα δίκτυα αισθητήρων που χρησιμοποιούν πρωτόκολλα Carrier Sense Multiple Access (CSMA), φαίνεται να έχουν μεγαλύτερο αριθμό χαμένων πακέτων λόγω του ανταγωνισμού για το ασύρματο μέσο παρά εξαιτίας των buffer drops και μάλιστα παρατήρησαν πως το χάσιμο πακέτων αυξάνεται πάρα πολύ με την αύξηση του load και χειροτερεύει στις περιοχές όπου υπάρχει πάρα πολλή κίνηση, δηλαδή κοντά στο sink ή στις περιοχές όπου παρατηρείται κάποιο φαινόμενο. Όταν οι κόμβοι ανιχνεύσουν κάποια αλλαγή στο φαινόμενο που παρακολουθούν, ξεκινούν να στέλλουν ταυτόχρονα όλοι τους πληροφορίες στο sink με αποτέλεσμα να χάνονται πακέτα σε εκείνη την περιοχή. Καθώς προσπαθούν όλοι οι κόμβοι να δεσμεύσουν το κανάλι η χωρητικότητα του φτάνει στο μέγιστο, κάτι το οποίο δεν ισχύει όμως για τις ουρές των κόμβων. Τα buffer drops προκαλούνται όταν πολλοί κόμβοι στέλλουν δεδομένα σε ένα κεντρικό σημείο (sink) το οποίο δεν προλαβαίνει να επεξεργαστεί όλα εκείνα τα δεδομένα που λαμβάνει με αποτέλεσμα να προκληθεί χάσιμο πακέτων. Έτσι όπως έχει παρατηρηθεί το χάσιμο πακέτων στις ουρές των κόμβων δεν μπορεί από μόνο του να θεωρηθεί ως αξιόπιστη ένδειξη προβλήματος συμφόρησης, χρειάζεται να λαμβάνονται υπόψη και άλλες παράμετροι.

2.4 Swarm Intelligence

Με την μεγάλη ανάπτυξη που έχει παρουσιαστεί γύρω από την Πληροφορική και την πολυπλοκότητα των νέων συστημάτων έχει παρουσιαστεί η ανάγκη ανάπτυξης νέων τεχνικών αφού όπως έχει φανεί οι παραδοσιακές μέθοδοι δεν μπορούν να ικανοποιήσουν τον μεγάλο αριθμό χρηστών και την διαφορετικότητα των διαφόρων συσκευών που είναι ενωμένες στα δίκτυα. Τα νέα συστήματα είναι αυξημένων απαιτήσεων και είναι αναγκαίο να μπορούν να είναι εύρωστα, ευέλικτα, αυτόνομα και αυτοδιοικούμενα. Έχουν διεξαχθεί πάρα πολλές έρευνες γύρω από τα σύνθετα συστήματα προκειμένου να προσδιοριστούν τα στοιχεία που επιτρέπουν στο σύστημα να αλληλεπιδρά με το περιβάλλον και τα μέλη που το απαρτίζουν να μπορούν να συνεργάζονται αποδοτικά για ένα κοινό σκοπό. Τα σύνθετα συστήματα στηρίζονται στην έρευνα φυσικών, βιολογικών μηχανισμών και διαδικασιών που επιτρέπουν την ανάπτυξη εύρωστων, αυτόνομων, ευέλικτων και αυτοδιοικούμενων πληροφορικών συστημάτων που αποτελούνται από απλούς αυτόνομους κόμβους. [4]

Ο αλγόριθμος που μελετώ είναι εμπνευσμένος από την φύση και πιο συγκεκριμένα από τον τρόπο κίνησης των πουλιών σε κοπάδια. Πολλοί άλλοι αλγόριθμοι είναι εμπνευσμένοι από την φύση και υπάρχουν πολλά παραδείγματα που μπόρεσαν να επιλύσουν αποτελεσματικά σύνθετα προβλήματα. Το Swarm Intelligence είναι ένα υπολογιστικό πρότυπο το οποίο είναι εμπνευσμένο από την φύση, και πιο συγκεκριμένα από την συμπεριφορά διαφόρων συνεργατικών οργανισμών όπως είναι τα μυρμηγκιά, οι μέλισσες, τα ψάρια και τα πουλιά. Μιμείται αυτούς τους οργανισμούς αφού ο τρόπος ζωής τους μπορεί να δώσει λύσεις σε διάφορα υπολογιστικά προβλήματα. Αυτές οι κοινωνίες εντόμων και πουλιών, ζουν αποκεντρωμένες και μπορούν να οργανώνονται ανάλογα με το συχνά τροποποιούμενο περιβάλλον όπου διαμένουν. Αυτά τα τρία χαρακτηριστικά είναι κοινά προβλήματα που προσπαθούμε εδώ και καιρό να αντιμετωπίσουμε και γι' αυτό το λόγο έχει δοθεί ιδιαίτερη έμφαση στην μελέτη των οργανισμών. Μέσα από αυτή την μελέτη έχουν αναπτυχθεί αρκετοί κατανεμημένοι αλγόριθμοι οι οποίοι έχουν εφαρμοστεί επιτυχώς σε διάφορα επιστημονικά προβλήματα. Επιπλέον, οι αλγόριθμοι του Swarm Intelligence μπορούν να εφαρμοστούν σε ένα σύνολο στατικών προβλημάτων αλλά και σε διάφορα άλλα

προβλήματα αφού είναι ευέλικτοι και εύρωστοι σε δυναμικά μεταβαλλόμενα περιβάλλοντα. [6,7]

Οι διάφορες κοινωνίες οργανισμών που έχω αναφέρει ονομαστικά πιο πάνω, μπορούν να συνεργάζονται σε εξαιρετικά καλό βαθμό προκειμένου να διεκπεραιώσουν τον κοινό τους σκοπό (κατασκευή της φωλιάς τους και συλλογή της τροφής τους). Παρόλο που είναι αυτόνομες οντότητες κι έχουν περιορισμένο πεδίο αντίληψης του περιβάλλοντος γύρω τους, μπορούν να συνεργάζονται άψογα εφαρμόζοντας μόνο ένα σύνολο απλών κανόνων συμπεριφοράς. Τα έντομα ανά πάσα στιγμή μπορούν να αφήσουν ίχνη κάποιας ουσίας την οποία εντοπίζουν οι υπόλοιποι οργανισμοί του ιδίου είδους και με αυτό τον τρόπο αλληλεπιδρούν, μεταβάλλουν το περιβάλλον στο οποίο διαμένουν και επικοινωνούν με τον δικό τους μοναδικό τρόπο. [3, 10]

Ο Flock-CC αλγόριθμος που μελετώ εστιάζει στην κίνηση των πουλιών σε κοπάδια όπου το κάθε πουλί έχει περιορισμένο πεδίο αντίληψης αλλά μπορεί να αλληλεπιδρά με τα άλλα πουλιά κι εφαρμόζοντας ένα απλό σύνολο κανόνων συμπεριφοράς μπορούν να οδηγούν το κοπάδι χωρίς προβλήματα. Οι κανόνες που ακολουθεί το κάθε πουλί είναι οι εξής:

1. Έλξη προς τους γείτονες για να διατηρείται συνοχή μεταξύ των μελών του κοπαδιού.
2. Απώθηση από γειτονικά πουλιά σε περίπτωση που βρίσκονται πάρα πολύ κοντά για να αποφευχθούν οι συγκρούσεις.
3. Ίδια ταχύτητα και κατεύθυνση με τα γειτονικά πουλιά.
4. Τυχαιότητα για να μπορούν να αντιμετωπίσουν τυχόν αλλαγή στο περιβάλλον όταν προκύψει κάποιο εμπόδιο.

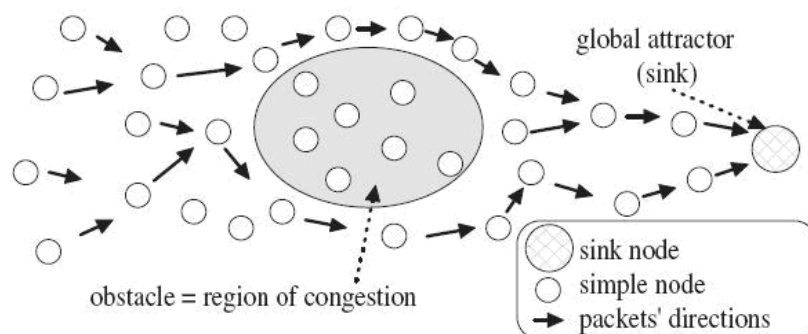
Κεφάλαιο 3

Flock-Based Congestion Control

3.1	Γενική περιγραφή	17
3.2	Παράμετροι του αλγορίθμου	19
3.3	Εφαρμογή του αλγορίθμου	24

3.1 Γενική Περιγραφή

Ο αλγόριθμος που μελετώ προσπαθεί να εντοπίσει περιοχές με συμφόρηση και να τις αποφύγει. Είναι ευέλικτος και μπορεί να ανταποκριθεί αυτόματα σε οποιαδήποτε αλλαγή προκύπτει στο δίκτυο. Εφαρμόζεται σε κάθε κόμβο ξεχωριστά, είναι απλός και προσπαθεί να μειώσει την ανταλλαγή αχρείαστων πληροφοριών, αυξάνοντας με αυτό τον τρόπο την συνολική διάρκεια ζωής του δικτύου. Είναι εμπνευσμένος από το Swarm Intelligence και πιο συγκεκριμένα από την συμπεριφορά των πουλιών που κινούνται σε κοπάδια.



Σχήμα 3.1: Κίνηση πακέτων προσπαθώντας να αποφύγουν το εμπόδιο [2]

Η βασική ιδέα του αλγορίθμου είναι η αποστολή των πακέτων προς το sink (global attractor) δημιουργώντας κοπάδια και αποφεύγοντας τα εμπόδια, δηλαδή τις περιοχές με προβλήματα συμφόρησης. Ταυτόχρονα, προσπαθεί να μοιράσει το φορτίο σε πολλαπλά μονοπάτια και να εκμεταλλευτεί όσο γίνεται καλύτερα ολόκληρο το δίκτυο. Προκειμένου να γίνει αυτό, χρειάζεται να εφαρμοστούν κάποιες δυνάμεις. Χρειάζεται με κάποιο τρόπο, τον οποίο θα αναπτύξω στην συνέχεια, τα πακέτα να έλκονται προς την κατεύθυνση του sink, ενώ παράλληλα πρέπει να μπορούν να «απωθούνται» και να «έλκονται» από γειτονικά πακέτα.

Τα πακέτα μπορούν να αναπαρασταθούν ως τα πουλιά τα οποία ταξιδεύουν σε κοπάδια σε ένα χώρο ο οποίος θεωρούμε πως είναι ένα δίκτυο αισθητήρων. Όπως τα πουλιά έχουν περιορισμένο πεδίο αντίληψης, με τον ίδιο τρόπο θεωρούμε πως τα πακέτα μπορούν να κινηθούν εντός ενός περιορισμένου πεδίου. Η κίνηση των πακέτων επηρεάζεται από μια ελκτική δύναμη προς το sink και από κάποιες ελκτικές και απωθητικές δυνάμεις μεταξύ των γειτονικών πακέτων. Συνοπτικά, ο αλγόριθμος υποστηρίζει εξεύρεση της κατεύθυνσης στην οποία βρίσκεται το sink, έλεγχο συμφόρησης και διαχείριση του traffic. Τα πακέτα στέλλονται προς το sink βήμα προς βήμα και κάθε φορά ο επόμενος κόμβος στον οποίο θα προωθηθεί το πακέτο επιλέγεται σύμφωνα με ένα desirability function, το οποίο λαμβάνει υπόψη την κατάσταση του ασύρματου καναλιού, την κατάσταση στις ουρές των κόμβων αλλά και την έλξη προς το sink. Ο αλγόριθμος προσπαθεί με αυτό τον τρόπο να επιλέξει το ιδανικότερο μονοπάτι προς το sink προκειμένου να έχει όσο το δυνατό χαμηλότερο αριθμό συγκρούσεων και χάσιμο πακέτων και όσο πιο ψηλό ποσοστό επιτυχημένης αποστολής πακέτων και εξοικονόμηση ενέργειας.

Για την υλοποίηση του αλγορίθμου χρειάστηκε να αναπροσαρμοστεί η προσέγγιση του Couzin από ένα μετρικό χώρο σε τοπολογικό (γράφος από κόμβους) για να μπορεί να ελεγχθεί η κατεύθυνση της κίνησης των πακέτων στο δίκτυο αισθητήρων. Τα πακέτα δημιουργούνται σε μια ομάδα κόμβων σε οποιοδήποτε σημείο μέσα στο δίκτυο και αποστέλλονται στο sink (το οποίο είναι μόνο ένα). Προσπαθώντας λοιπόν να δώσουμε στα πακέτα τις ιδιότητες των πουλιών καθώς διασχίζουν το δίκτυο, χρησιμοποιήσαμε

ελκτικές και απωθητικές δυνάμεις καθώς επίσης και μια ελκτική δύναμη προς την κατεύθυνση του sink.

Τα πακέτα λοιπόν:

1. Έλκονται σε γειτονικά πακέτα τα οποία βρίσκονται σε κόμβους των οποίων η ουρά είναι πιο άδεια (χαμηλό queue loading).
2. Απωθούνται από γειτονικά πακέτα τα οποία βρίσκονται σε κόμβους όπου παρουσιάζεται συμφόρηση στο κανάλι (ψηλό wireless channel contention).
3. Έλκονται προς την κατεύθυνση του sink, αφού κάθε φορά που πρόκειται να επιλεγεί ο επόμενος κόμβος στον οποίο θα προωθηθεί το πακέτο προτιμάται κάποιος ο οποίος βρίσκεται πιο κοντά στο sink.
4. Χρησιμοποιούν λίγη τυχειότητα για την επιλογή του δρομολογίου που θα ακολουθήσουν.

3.2 Παράμετροι του αλγορίθμου

Σε αυτή την ενότητα θα παρουσιάσω τις σημαντικές παραμέτρους που χρησιμοποιούνται από τον αλγόριθμο. Αρχικά, πρέπει να αναφέρω πως όλες οι παράμετροι που πρόκειται να ορίσω στη συνέχεια έχουν μελετηθεί και έχουν ληφθεί δείγματα σε διακριτά χρονικά διαστήματα T δευτερολέπτων και αξιολογούνται στον κάθε κόμβο ξεχωριστά, και περιοδικά (κάθε T secs) διαφημίζονται στους γειτονικούς κόμβους μέσω ενός control packet.

Ανίχνευση συμφόρησης: Όπως έχω ήδη αναφέρει, η συμφόρηση δεν μπορεί να ανιχνευθεί αξιολογώντας αποκλειστικά την κατάσταση στις ουρές των κόμβων. Χρειάζεται να χρησιμοποιηθούν και άλλες πληροφορίες που να υποδεικνύουν αποτελεσματικά πως υπάρχει ένδειξη συμφόρησης. Ο αλγόριθμος λαμβάνει υπόψη την κατάσταση στις ουρές των κόμβων αλλά και το loading του ασύρματου καναλιού. Στο ασύρματο μέσο συχνά παρατηρείται χάσιμο πακέτων λόγω υπερχειλίσης των ουρών των κόμβων, λόγω προβλημάτων στο link layer όπως είναι οι συγκρούσεις, και λόγω

προβλημάτων στην δρομολόγηση (π.χ. αδύνατη η επικοινωνία με τον επόμενο κόμβο). Λαμβάνοντας υπόψη λοιπόν τα παραπάνω συμπτώματα συμφόρησης και την κατάσταση στις ουρές των κόμβων, χρησιμοποιείται ένας node loading indicator $p_m(k)$ στον κάθε κόμβο n και υπολογίζεται ως εξής:

$$p_m(k) = \frac{P_m^{in}(k) + q_m(k-1) - P_m^{out}(k)}{P_m^{in}(k) + q_m(k-1)}$$

Με τον όρο $P_m^{in}(k)$ απεικονίζεται ο αριθμός των εισερχόμενων πακέτων και $P_m^{out}(k)$ ο αριθμός των επιτυχημένα εξερχόμενων πακέτων, η τιμή των οποίων υπολογίζεται κατά το τέλος της k -οστής περιόδου. Ο όρος $q_m(k-1)$ αναπαριστά το μέγεθος της ουράς για τον κόμβο n (πόσα πακέτα περιέχονται στην ουρά μετά το τέλος της προηγούμενης περιόδου δειγματοληψίας). Παρατηρούμε πως η τιμή αυτού του όρου υπολογίζεται κατά το τέλος της προηγούμενης περιόδου $(k-1)$ και είναι λογικό αφού θέλουμε να καθορίσουμε τον αριθμό των πακέτων που έχουν περάσει από τον κόμβο n κατά την διάρκεια της περιόδου δειγματοληψίας. Η παράμετρος $p_m(k)$, $(0 \leq p_m(k) \leq 1)$, αντιπροσωπεύει την διαφορά ανάμεσα στον συνολικό αριθμό πακέτων που πρέπει να εξυπηρετηθούν (δηλαδή τα πακέτα που έχουν καταφθάσει στον κόμβο εντός της περιόδου δειγματοληψίας, συν τα πακέτα που ήταν ήδη στην ουρά του από την προηγούμενη περίοδο), πλην τα πακέτα που έχουν μεταδοθεί επιτυχώς, διαιρούμενο με τον συνολικό αριθμό πακέτων που πρέπει να εξυπηρετηθούν. Όταν $p_m(k) \rightarrow 0$ σημαίνει πως πολύ λίγα πακέτα έχουν χαθεί (σχεδόν μηδέν) και η ουρά του κόμβου είναι σχεδόν άδεια. Όταν όμως $p_m(k) \rightarrow 1$, θεωρούμε πως ο κόμβος n παρουσιάζει προβλήματα συμφόρησης διότι έχει είτε αυξημένο αριθμό χαμένων πακέτων, είτε η ουρά του είναι σχεδόν γεμάτη.

Μια άλλη παράμετρος που είναι εξαιρετικά χρήσιμη είναι το $r_m(k)$. Αυτή η παράμετρος παρουσιάζει το useful link service rate για τον κάθε κόμβο n , συλλέγοντας πληροφορίες από το πρωτόκολλο MAC. Ο αλγόριθμος χρησιμοποιεί πρωτόκολλο τύπου CSMA (π.χ. 802.11) το οποίο χρησιμοποιεί αναμεταδόσεις (retransmissions) προκειμένου να αυξήσει την αξιοπιστία του ασύρματου καναλιού. Υπάρχει ένας μέγιστος αριθμός φορών για τις οποίες μπορεί να αναμεταδοθεί ένα πακέτο και σε περίπτωση που όλες οι φορές είναι αποτυχημένες τότε το πακέτο καταστρέφεται. Ο όρος $P_m^{out*}(k)$ αναπαριστά τον συνολικό αριθμό προσπαθειών μετάδοσης κάποιου

πακέτου στον κόμβο n , κατά την k -οστή περίοδο δειγματοληψίας και ισούται με τον αριθμό των επιτυχημένα εξερχόμενων πακέτων συν τον αριθμό των αναμεταδόσεων.

$$P_m^{out*}(k) = P_m^{out}(k) + retransmits$$

Το useful service rate για τον κόμβο n παρουσιάζεται ως ο λόγος του αριθμού των επιτυχημένα εξερχόμενων πακέτων ως προς τον ίδιο αριθμό μαζί με τις αναμεταδόσεις:

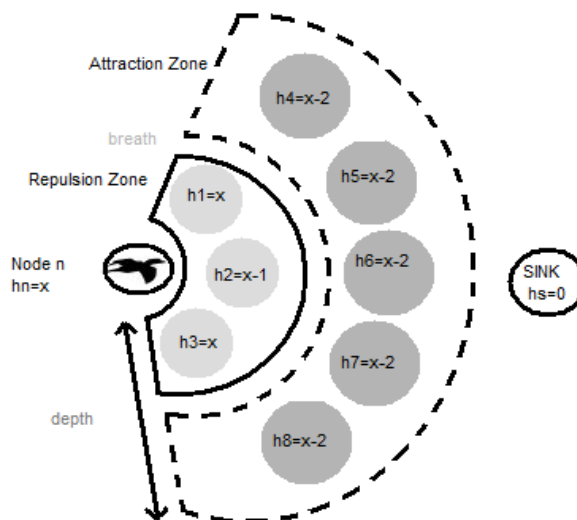
$$r_m(k) = \frac{P_m^{out}(k)}{P_m^{out*}(k)} = \frac{P_m^{out}(k)}{P_m^{out}(k) + retransmits}$$

Η παράμετρος $r_m(k)$ παίρνει τιμές μεταξύ 0 και 1. Όταν $r_m(k) \rightarrow 1$ σημαίνει πως το κανάλι δεν έχει συμφόρηση και ένα μεγάλο ποσοστό των πακέτων μεταδίδονται με επιτυχία (πολύ μικρός αριθμός αναμεταδόσεων). Όταν $r_m(k) \rightarrow 0$ σημαίνει πως το κανάλι παρουσιάζει προβλήματα συμφόρησης και ελάχιστος αριθμός πακέτων μεταδίδεται επιτυχώς και συχνά παρουσιάζεται και μεγάλος αριθμός αναμεταδόσεων.

Ανίχνευση κατεύθυνσης του sink: Για τον εντοπισμό της κατεύθυνσης του sink χρησιμοποιείται ο αριθμός των hop counts μεταξύ του κάθε κόμβου και του sink κι αυτό υποδηλώνεται με την παράμετρο $h_m(k)$. Περιοδικά, οι κόμβοι λαμβάνουν τα control packets από τα οποία μπορούν να μάθουν τον υπολογισμό για το hop count των γειτονικών τους κόμβων και με αυτό τον τρόπο μπορούν να εντοπίσουν την κατεύθυνση του sink. Όσο πιο μικρό είναι το hop count σημαίνει πως τόσο πιο κοντά στο sink βρίσκεται ο κόμβος. Επίσης, τα hop counts μπορούν να μεταβληθούν έπειτα από τοπολογικές αλλαγές ή από κατάρρευση κόμβων. Σε περίπτωση που περάσει ένα χρονικό διάστημα T_{lost} δευτερολέπτων και κάποιος κόμβος n δεν λάβει control packets από κάποιο γειτονικό του κόμβο, τότε θεωρεί πως ο συγκεκριμένος κόμβος έχει καταρρεύσει ή βρίσκεται εκτός του πεδίου αντίληψης του και συνεπώς θα πρέπει να επαναυπολογίσει τον πίνακα δρομολόγησης του. Παρομοίως, η παράμετρος του hop count επαναυπολογίζεται σε περίπτωση που η απομάκρυνση του γειτονικού κόμβου προκαλεί αλλαγή στην τιμή της. Οποιαδήποτε αλλαγή στις τιμές του hop count πρέπει να διαδοθεί σε όλο το δίκτυο (ακολουθώντας μια πορεία προς τα πίσω) και όσοι κόμβοι επηρεάζονται θα πρέπει να ανανεώσουν τις σχετικές μεταβλητές για το hop count.

Ζώνες έλξης και απώθησης: Θεωρούμε πως έχουμε ένα δίκτυο με N κόμβους οι οποίοι δημιουργούν πακέτα. Ο κάθε ένας από αυτούς τους κόμβους N έχει περιορισμένο

μέγεθος ουράς και το throughput του περιορίζεται από την χωρητικότητα του ασύρματου καναλιού. Ανάλογα με την τοποθεσία που βρίσκεται ο κάθε κόμβος, καθορίζεται και η τρέχουσα θέση του κάθε πακέτου i το οποίο φιλοξενείται στον εκάστοτε κόμβο. Το κάθε πακέτο i θεωρούμε πως γειτονεύει με κάποια άλλα πακέτα όταν αυτά βρίσκονται σε κόμβους εντός της εμβέλειας του συγκεκριμένου κόμβου που φιλοξενεί το πακέτο i . Αυτή η εμβέλεια μπορεί να διαχωριστεί σε δύο ζώνες, την ζώνη έλξης και την ζώνη απόθησης. Κάθε φορά που πρόκειται να επιλεγεί ο επόμενος κόμβος που θα φιλοξενήσει το πακέτο i , ο τρέχον κόμβος στον οποίο βρίσκεται το πακέτο χρειάζεται να κάνει κάποιους υπολογισμούς προκειμένου να επιλέξει τον ιδανικότερο κόμβο, ώστε να μειωθεί ή ακόμη καλύτερα να αποφευχθεί η συμφόρηση στους επόμενους κόμβους. Για να ληφθεί αυτή η απόφαση λαμβάνονται υπόψη οι δυνάμεις έλξης και απόθησης των γειτονικών πακέτων και η κατεύθυνση στην οποία βρίσκεται το sink. Ακολούθως, θα περιγράψω τη διαδικασία επιλογής του επόμενου κόμβου.



Σχήμα 3.2: Ζώνες έλξης και απόθησης

Στο σχήμα 3.2 παρουσιάζονται οι κόμβοι που συμπεριλαμβάνονται στην περιοχή απόθησης (ελαφρύ γκρίζο χρώμα) και οι κόμβοι που περιλαμβάνονται στην περιοχή έλξης (σκούρο γκρίζο χρώμα). Αυτή είναι η περιοχή γειτνίασης του κάθε πακέτου i που βρίσκεται στον κόμβο n . Το κωνικό σχήμα που βλέπουμε στο σχήμα 3.2 αποτελεί την περιοχή αντίληψης για το κάθε πακέτο i που φιλοξενείται από τον συγκεκριμένο κόμβο

n. Το σχήμα είναι κωνικό λόγω του περιορισμένου οπτικού πεδίου των πουλιών από τα οποία είναι εμπνευσμένος και ο αλγόριθμος. Όπως είναι λογικό, ένα πακέτο μπορεί να αντιληφθεί μόνο ένα μέρος των πακέτων που διακινούνται στο δίκτυο, αυτά με τα οποία γειτνιάζει. Το πεδίο αντίληψης πρέπει να έχει κλίση προς την πλευρά που βρίσκεται το sink προκειμένου να υλοποιηθεί η ελκτική δύναμη στην οποία είχα αναφερθεί προηγουμένως. Ο αλγόριθμος προσπαθεί ούτως ώστε σε κάθε βήμα τα πακέτα να βρίσκονται όλο και πιο κοντά στον προορισμό τους και γι' αυτό το λόγο το πεδίο αντίληψης περιλαμβάνει μόνο κόμβους με μικρότερο ή ίσο hop count από το sink, σε σχέση με την κόμβο n.

Όπως μπορούμε να παρατηρήσουμε και στο σχήμα 3.2 το βάθος και το πλάτος της κωνικής επιφάνειας παίζουν πολύ σημαντικό ρόλο για το πεδίο αντίληψης. Το βάθος προσδιορίζει τον αποδεκτό αριθμό από hops στον οποίο μπορεί να βρίσκεται κάποιος κόμβος και να εμπεριέχεται στο πεδίο αντίληψης. Είναι σημαντικό να αναφερθεί πως όλοι οι κόμβοι μπορούν να καταταχθούν σε τρεις κατηγορίες:

1. κόμβοι που βρίσκονται πιο κοντά στο sink σε σχέση με τον κόμβο n.
2. κόμβοι που έχουν την ίδια απόσταση σε hops από το sink σε σχέση με τον κόμβο n.
3. κόμβοι που βρίσκονται πιο μακριά από το sink σε σχέση με τον κόμβο n.

Το πλάτος της περιοχής καθορίζει ποια από τα προαναφερθέντα σύνολα θα συμπεριληφθούν στο πεδίο αντίληψης. Το πεδίο αντίληψης (βλέπε σχήμα 3.2) περιλαμβάνει μόνο τους κόμβους με μικρότερο ή ίσο hop count προς το sink σε σχέση με τον κόμβο n κι αφού το βάθος του πεδίου αντίληψης είναι ίσο με 2, λαμβάνονται υπόψη μόνο οι κόμβοι που βρίσκονται το πολύ 2 hops μακριά.

Όπως φαίνεται και στο σχήμα 3.2, οι κόμβοι που βρίσκονται ένα hop μικρότερο προς το sink σε σχέση με τον κόμβο n εμπεριέχονται στην ζώνη απώθησης και είναι οι κόμβοι εντός της ακτίνας μετάδοσης του κόμβου n που φιλοξενεί το πακέτο i, ενώ οι κόμβοι που βρίσκονται 2 hops πιο κοντά αποτελούν την ζώνη έλξης. Ο λόγος που επιλέχθηκε αυτή η δομή είναι πως σε περίπτωση που ο κόμβος n αισθανθεί κάποιο

γεγονός και ξεκινήσει την μαζική αποστολή πακέτων προς το sink, τότε είναι αναμενόμενο πως θα προκληθούν προβλήματα συμφόρησης στην περιοχή που βρίσκεται ένα hop από τον κόμβο n και όλοι οι κόμβοι θα ανταγωνίζονται για το ποιος θα κερδίσει την πρόσβαση στο ασύρματο μέσο. Γι' αυτό το λόγο αυτή η περιοχή έχει οριστεί ως η περιοχή απώθησης, διότι αφού το πακέτο θα απωθείται από τα πακέτα των κόμβων που βρίσκονται σε αυτή την περιοχή, θα μειώνεται και ο ανταγωνισμός για το μέσο και τις σχετικές συνέπειες. Επιπλέον, τα πακέτα που κατάφεραν με επιτυχία να μεταδοθούν από τους κόμβους που βρίσκονται στην ζώνη απώθησης προς την ζώνη έλξης (2 hops μακριά) έλκουν τα υπόλοιπα πακέτα που ακολουθούν. Το επίπεδο έλξης εξαρτάται πλέον από την κατάσταση στις ουρές των κόμβων που βρίσκονται ένα hop μακριά. Από την στιγμή που μόνο οι κόμβοι ένα hop μακριά βρίσκονται στην περιοχή μετάδοσης του εκάστοτε κόμβου n , είναι αναμενόμενο πως το κάθε πακέτο i , το οποίο πρόκειται να αποσταλεί σε ένα νέο κόμβο, μπορεί να μεταδοθεί μόνο σε κόμβους που βρίσκονται ένα hop μακριά από τον τρέχον κόμβο που το φιλοξενεί. Επίσης, το κάθε πακέτο μπορεί και αλληλεπιδρά με πακέτα που είτε βρίσκονται σε κόμβους ένα hop μακριά ή μόλις έχουν μεταδοθεί σε κόμβους δύο hops μακριά. Οι δυνάμεις έλξης και απώθησης που ασκούνται στα διάφορα πακέτα, υπολογίζονται από τον εμπλεκόμενο κόμβο που βρίσκεται ένα hop μακριά και γνωστοποιούνται στον κόμβο n , κάθε περίοδο k μέσω ενός broadcast control μηνύματος.

3.3 Εφαρμογή του αλγορίθμου

Σε αυτό το σημείο πρόκειται να περιγράψω τις εξισώσεις που χρησιμοποιούνται από τον αλγόριθμο και τον τρόπο με τον οποίο αναπαριστούνται οι σχετικές δυνάμεις στις οποίες έχω αναφερθεί προηγουμένως. Η δύναμη απώθησης είναι ανάλογη του $1 - r_m(k)$ και η ελκτική δύναμη του $1 - p_m(k)$. Αυτές οι δύο δυνάμεις χρησιμοποιούνται από μια συνάρτηση η οποία υπολογίζει πόσο «επιθυμητός» είναι κάποιος κόμβος (desirability function). Με τον όρο «επιθυμητός» εννοώ τον κόμβο ο οποίος παρουσιάζεται ως ο ιδανικότερος για να φιλοξενήσει το πακέτο, δηλαδή είναι ο κόμβος του οποίου η ουρά είναι σε πολύ κατάσταση, το ίδιο και το ασύρματο κανάλι. Στη

συνέχεια θα παρουσιάσω και τις υπόλοιπες ιδιότητες που υποστηρίζει ο αλγόριθμος, δηλαδή την έλξη προς την κατεύθυνση του sink και την εισαγωγή της τυχειότητας.

Desirability: Όταν ένας κόμβος πρόκειται να προωθήσει τα πακέτα του παρακάτω χρειάζεται να υπολογίσει πόσο επιθυμητός είναι ο κάθε ένας από τους κόμβους που βρίσκονται 1 hop μακριά του. Για να γίνει αυτό υπολογίζεται ένα διάνυσμα M διαστάσεων (M -dimensional desirability vector), το $\overrightarrow{D(k)}$, όπου M ο αριθμός των κόμβων που εμπεριέχονται στο πεδίο μετάδοσης του κόμβου n και $M \leq N$. Το κάθε στοιχείο του $D_m(k)$ του διανύσματος αντιπροσωπεύει πόσο επιθυμητός είναι ο κάθε κόμβος m , όπου $1 \leq m \leq M$ και υπολογίζεται μια φορά στην αρχή της περιόδου δειγματοληψίας. Επιπλέον, η τιμή αυτή χρησιμοποιείται για το κάθε πακέτο που πρόκειται να αποσταλεί εντός της συγκεκριμένης περιόδου. Η εξίσωση για το $D_m(k)$ ορίζεται ως εξής:

$$D_m(k) = a \cdot r_m(k) + (1 - a) \cdot (1 - p_m(k))$$

Η παράμετρος a καθορίζει την σημασία που θα παίξει η κατάσταση στις ουρές των κόμβων και η κατάσταση του ασύρματου μέσου και κυμαίνεται μεταξύ 0 και 1, συμπεριλαμβανομένου. Το $D_m(k)$ υπολογίζει την ελκτική τάση που έχουν τα πακέτα προς τον κόμβο m . Όσο μεγαλύτερη είναι η τιμή του, τόσο πιο επιθυμητός είναι ο συγκεκριμένος κόμβος. Χρειάζεται λοιπόν ψηλή τιμή στο $r_m(k)$, που αντιπροσωπεύει την απόθεση από κόμβους με αρκετά φορτωμένο ασύρματο κανάλι, και χαμηλό $p_m(k)$, δηλαδή έλξη προς τους κόμβους που έχουν καλή κατάσταση στις ουρές τους (χαμηλό φόρτο).

Σύμφωνα με όσα έχω αναφέρει έως τώρα, ένα πακέτο μπορεί απλά να προωθηθεί στον κόμβο m που είναι ο πιο επιθυμητός από τους κόμβους που βρίσκονται ένα hop μακριά από τον τρέχον κόμβο που φιλοξενεί το πακέτο. Αυτό όμως δεν λαμβάνει υπόψη όλες τις ιδιότητες που πρέπει να υποστηρίζει ο αλγόριθμος. Δηλαδή, δεν λαμβάνεται υπόψη η κατεύθυνση του sink και η τυχειότητα κι αυτό μπορεί να οδηγήσει το πακέτο σε κύκλους ανάμεσα σε κόμβους που δεν παρουσιάζουν προβλήματα συμφόρησης. Στην συνέχεια, θα παρουσιάσω τις δύο αυτές ιδιότητες.

Ελξη προς την κατεύθυνση του sink: Για να οριστεί η ελκτική δύναμη προς το sink χρειάζεται απλά να λαμβάνονται υπόψη οι κόμβοι με μικρότερο hop count με τον κόμβο που φιλοξενεί το πακέτο την δεδομένη χρονική στιγμή και ανάμεσα σε αυτούς τους κόμβους να γίνεται η επιλογή αυτού που είναι ο πιο επιθυμητός (έχει το μεγαλύτερο $D_m(k)$). Το μειονέκτημα αυτής της προσέγγισης είναι πως δεν λαμβάνονται υπόψη κόμβοι που παρόλο που δεν βρίσκονται στο κεντρικό μονοπάτι προς το sink, έχουν πιο ψηλή τιμή στο $D_m(k)$, δηλαδή χαμηλότερο φόρτο στις ουρές τους και λιγότερα προβλήματα στο ασύρματο κανάλι. Έτσι με αυτό τον τρόπο το «κοπάδι» που δημιουργείται είναι πιο στενό και ως λογικό επακόλουθο, αφού χρησιμοποιείται συνολικά μικρότερος αριθμός κόμβων, αυξάνεται το φόρτο στις ουρές τους και ο ανταγωνισμός μεταξύ τους για το ποιος κόμβος θα επικρατήσει και θα δεσμεύσει το κανάλι αλλά ταυτόχρονα σπαταλείται περισσότερη ενέργεια από αυτούς τους κόμβους μειώνοντας την συνολική διάρκεια ζωής του δικτύου. Σε αντίθετη περίπτωση, όπου γίνεται εκμετάλλευση ολόκληρου του δικτύου, σπαταλείται χαμηλότερο ποσό ενέργειας ανά κόμβο κι έτσι το δίκτυο παραμένει ζωντανό για μεγαλύτερη χρονική περίοδο. Για να αποφευχθούν λοιπόν αυτά τα προβλήματα, ο αλγόριθμος προνοεί και λαμβάνει υπόψη του και τους κόμβους που δεν βρίσκονται πιο κοντά στο sink όμως οι κόμβοι αυτοί μπορούν να επιλεγθούν με χαμηλότερη πιθανότητα χρησιμοποιώντας το $d_{im}(t)$ (discount factor).

Τυχαιότητα: Η τυχαιότητα μπορεί αρκετές φορές να οδηγήσει σε καλύτερα αποτελέσματα και καλύτερες λύσεις προβλημάτων γι' αυτό το λόγο χρησιμοποιείται και από τον αλγόριθμο κι εφαρμόζεται στο desirability function. Σε κάθε περίοδο δειγματοληψίας υπολογίζονται τα $D_m(t)$ υποψήφιων κόμβων. Προτού μεταδοθεί ένα πακέτο, χρειάζεται να διαφοροποιήσουμε ελαφρώς τα $D_m(t)$, και αφού γίνει αυτό επιλέγεται ο κόμβος με την μεγαλύτερη συνολικά τιμή. Η διαφοροποίηση γίνεται πολλαπλασιάζοντας την εξίσωση του desirability ενός κόμβου με ένα τυχαίο συντελεστή ο οποίος παίρνει την τιμή του έπειτα από εκτέλεση της κανονικής Γκαουσιανής διανομής (Gaussian distribution) με μέσο όρο 1 και τυπική απόκλιση α .

Η ανανεωμένη εξίσωση του desirability είναι η εξής:

$$D'_m(t) = g \cdot d_{im}(t) \cdot D_m(t)$$

Η παράμετρος g αναπαριστά τον τυχαίο συντελεστή που παίρνει τιμή από την Γκαουσιανή διανομή. Το $d_{im}(t)$ είναι το discount factor στο οποίο αναφέρθηκα προηγουμένως και το οποίο μειώνει την πιθανότητα επιλογής κάποιου κόμβου που δεν βρίσκεται πιο κοντά στο sink σε σχέση με τον κόμβο όπου φιλοξενείται το πακέτο i την δεδομένη χρονική στιγμή. Λόγω του ότι οι αποστάσεις αλλάζουν με την πάροδο του χρόνου, πιθανότατα λόγω κατάρρευσης κόμβων ή έπειτα από οποιαδήποτε μεταβολή στην τοπολογία του δικτύου, το discount factor εξαρτάται από το t , όπου $kT-T \leq t \leq kT$.

Φιλοσοφία αλγορίθμου: Για το κάθε πακέτο i που πρέπει να προωθηθεί παρακάτω, επιλέγεται ο πιο επιθυμητός κόμβος, δηλαδή αυτός με που έχει το μεγαλύτερο $D'_m(t)$ ή αλλιώς m^* όπου:

$$m^* = \underset{m}{\operatorname{argmax}} \{\overrightarrow{D'_i(t)}\}$$

$\overrightarrow{D'_i(t)}$: Είναι το ανανεωμένο διάνυσμα M -διαστάσεων που υποδηλώνει πόσο επιθυμητός είναι ο κάθε ένας από τους κόμβους m , όπου $M \leq N$ ο αριθμός των κόμβων που βρίσκονται εντός της περιοχής μετάδοσης του κόμβου που φιλοξενεί το πακέτο i .

Στη συνέχεια θα περιγράψω και κάποιες άλλες παραμέτρους στις οποίες δεν έχω αναφερθεί ακόμα και για σκοπούς απλοποίησης λαμβάνω υπόψη μόνο τους κόμβους με μικρότερο ή ίσο hop count προς το sink σε σχέση με τον κόμβο που φιλοξενεί το πακέτο. Το discount factor $d_{im}(t)$ παίρνει την τιμή 1 όταν ο κόμβος m βρίσκεται πιο κοντά στο sink από τον τρέχον κόμβο που φιλοξενεί το πακέτο, διαφορετικά ισούται με μια σταθερά e ($0 \leq e \leq 1$) της οποίας η τιμή εισάγεται από το πρόγραμμα. Η παράμετρος e καθορίζει κατά πόσον θα λαμβάνονται υπόψη και κόμβοι με ίσο hop count και με πιο απλά λόγια, προσδιορίζει πόσο θα «ανοίγει» το κοπάδι των πακέτων. Για να μπορέσουν να επιλεγθούν κόμβοι με ίσο hop count, πρέπει το e να παίρνει αρκετά ψηλή τιμή, αν και αυτό μπορεί να επιφέρει προβλήματα. Όπως έχω αναφέρει και προηγουμένως, η τυχαιότητα εμπλέκεται χρησιμοποιώντας την παράμετρο g στην εξίσωση $D'_{im}(t) = g \cdot d_{im}(t) \cdot D_m(t)$. Πιο συγκεκριμένα, η τιμή του g λαμβάνεται

έπειτα από εφαρμογή της Γκαουσιανής διανομής $G(1,u)$ με μέσο όρο την μονάδα και τυπική απόκλιση το u .

$$g \sim G(1,u)$$

Το u ισούται με $c \cdot (1-e)^2$ και συνεπώς το $\sqrt{u}$ ορίζεται ως μια γραμμική συνάρτηση του $(1 - e)$. Κατά την εκτέλεση των πειραμάτων μελετώ τις παραμέτρους e και c , οι οποίες είναι ορθογώνιες και μπορούν να βελτιστοποιηθούν ανεξάρτητα η μια από την άλλη, και μέσω αυτών υπολογίζεται η τιμή του u .

Μέσα από τα διάφορα σενάρια κλήθηκα να μελετήσω τις παραμέτρους a , e , c και T οι οποίες είναι και οι πιο σημαντικές αφού καθορίζουν τον τρόπο λειτουργίας του αλγορίθμου. Συνοψίζοντας, η παράμετρος a καθορίζει την επίδραση που έχει η κατάσταση στις ουρές των κόμβων ($p_m(k)$) και του ασύρματου καναλιού ($r_m(k)$) στην συνάρτηση του desirability. Η παράμετρος e προσδιορίζει ποιοι κόμβοι θα λαμβάνονται υπόψη, και πιο συγκεκριμένα αναπαριστά το bias έναντι στους κόμβους με ίσο hop count προς το sink. Η παράμετρος T που αντιστοιχεί στην περίοδο δειγματοληψίας καθορίζει το χρονικό διάστημα που μεσολαβεί μέχρι να λάβουν τα control packets οι κόμβοι και να υπολογίζουν και πάλι το desirability function. Τέλος, η παράμετρος c αναπαριστά την διαφοροποίηση του noise που εισάγουμε για να υποδείξουμε την τυχειότητα στον αλγόριθμο. Η επιλογή των τιμών για αυτές τις παραμέτρους έγινε μετά από πολλά πειράματα τα οποία πρόκειται να αξιολογήσω στην συνέχεια. [2]

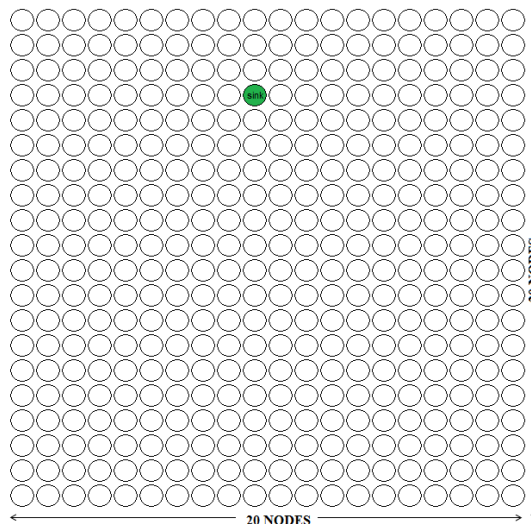
Κεφάλαιο 4

Αξιολόγηση Απόδοσης Αλγορίθμου

4.1	Γενική περιγραφή σεναρίων.....	29
4.2	Μελέτη παραμέτρων	31
4.3	Αλγόριθμοι σύγκρισης	32

Σε αυτό το κεφάλαιο θα περιγράψω τον τρόπο αξιολόγησης του αλγορίθμου Flock-CC τον οποίο μελέτησα χρησιμοποιώντας τον προσομοιωτή NS-2. Επίσης, θα κάνω και μια αναφορά στις μετρικές που επέλεξα να χρησιμοποιήσω και θα περιγράψω με λίγα λόγια τους αλγορίθμους με τους οποίους σύγκρινα τον Flock-CC.

4.1 Γενική Περιγραφή Σεναρίων



Σχήμα 4.1: Δίκτυο σε μορφή τετραγωνικού πλέγματος

Το δίκτυο αποτελείται από 400 κόμβους οι οποίοι είναι τοποθετημένοι σε ένα τετραγωνικό πλέγμα το οποίο μπορούμε να δούμε στο σχήμα 4.1. Η συγκεκριμένη

τοπολογία επιλέχθηκε διότι απλοποιεί τον αλγόριθμο και βοηθά στην καλύτερη κατανόηση και αξιολόγηση του. Θεωρούμε πως όλοι οι κόμβοι είναι οι ίδιοι με μέγιστη ακτίνα διάδοσης στα 50m. Η απόσταση ανάμεσα σε γειτονικούς κόμβους είναι η ίδια σε όλες τις περιπτώσεις και ισούται με $25 \cdot \sqrt{2}m$. Ο κάθε κόμβος μπορεί να έχει το πολύ 8 γειτονικούς κόμβους που βρίσκονται 1 hop μακριά. Το μέγεθος του control και του data packet έχει οριστεί σε 10 και 50 bytes αντίστοιχα. Η ουρά του κάθε κόμβου μπορεί να φιλοξενήσει το πολύ 50 πακέτα. Επίσης, χρησιμοποιείται το MAC πρωτόκολλο CSMA-based IEEE 802.11 με transmission rate ίσο με 2Mbps και χρησιμοποιεί την τεχνική του exponential backoff. Οι ανταλλαγές μηνυμάτων RTS και CTS έχει απενεργοποιηθεί για σκοπούς εξοικονόμησης ενέργειας.

Στα σενάρια που μελέτησα, η ομάδα δημιουργίας πακέτων βρισκόταν σε διάφορες θέσεις στο δίκτυο και οι θέσεις επιλέχθηκαν με τέτοιο τρόπο ώστε να καλυφθούν όσο το δυνατό καλύτερα οι σημαντικές περιπτώσεις και να φανεί η αποδοτικότητα του αλγορίθμου. Όταν η ομάδα δημιουργίας βρισκόταν ακριβώς απέναντι από το sink, μελετούσαμε την συμπεριφορά του αλγορίθμου για τοπολογίες όπου υπάρχουν πολλά εναλλακτικά μονοπάτια με τον συντομότερο αριθμό hops και τον τρόπο με τον οποίο τα αξιοποιεί. Στο σενάριο όπου είχαμε δύο ομάδες δημιουργίας πακέτων απέναντι από το sink, μελετούσαμε την ικανότητα του αλγορίθμου να αξιοποιεί τα εναλλακτικά μονοπάτια προκειμένου να αποφύγει την περιοχή συμφόρησης που προκαλείται από την ομάδα που βρίσκεται σε πιο κοντινή απόσταση. Στην περίπτωση όπου οι κόμβοι βρίσκονταν στο πλάι και τα πακέτα έπρεπε να οδηγηθούν μέσω της κεντρικής διαγωνίου, φάνηκε η ικανότητα του αλγορίθμου να αντεπεξέρχεται ακόμη και τοπολογίες όπου υπάρχει πολύ μικρός αριθμός δρομολογίων με τον συντομότερο αριθμό hops. Έπειτα, μέσα από τα σενάρια όπου μια σειρά κόμβων είχε καταρρεύσει στην μέση του δικτύου μελετούσαμε τον χρόνο ανάκαμψης του αλγορίθμου (εντοπισμό του προβλήματος και εξεύρεση εναλλακτικών μονοπατιών). Επιπλέον, με αυτό τον σενάριο δόθηκε ιδιαίτερη έμφαση στην σημασία της παραμέτρου T (περίοδος δειγματοληψίας). Στο άλλο σενάριο, όπου δύο κυρτές γραμμές κόμβων καταρρέουν μελετήθηκε η ευρωστία του αλγορίθμου, δηλαδή πόσο γρήγορα μπορεί να αντιληφθεί τις αλλαγές και να απεγκλωβίσει τα πακέτα που βρίσκονται στην εν λόγω προβληματική περιοχή. Με αυτό τον τρόπο μελετήθηκαν οι πιο σημαντικές

περιπτώσεις με τις οποίες φάνηκε η αξιοπιστία, ευρωστία και αποδοτικότητα του αλγορίθμου.

Δύο κοινές μετρικές απόδοσης για έλεγχο συμφόρησης είναι το ποσοστό επιτυχημένης αποστολής, οι συνολικές καθυστερήσεις και η ενέργεια που δαπανήθηκε. Αυτές τις μετρικές θα χρησιμοποιήσω κι εγώ στην συνέχεια κατά την αξιολόγηση των αποτελεσμάτων. Το ποσοστό επιτυχημένης αποστολής καθορίζεται ως ο λόγος του συνολικού αριθμού πακέτων που φθάνουν στο sink ως προς τον συνολικό αριθμό πακέτων που έχουν μεταδοθεί από την ομάδα των κόμβων που δημιουργούν τα πακέτα. Ενώ οι συνολικές καθυστερήσεις αναφέρονται στον συνολικό χρόνο που απαιτείται από ένα πακέτο να μεταδοθεί από τον κόμβο όπου δημιουργήθηκε στο sink.

4.2 Μελέτη Παραμέτρων

Για την αξιολόγηση των αποτελεσμάτων χρειάζεται να μελετήσω τις παραμέτρους a , e , c , T για τα τρία data rates. Έπειτα, θα παρουσιάσω μέσω κάποιων γραφικών αναπαραστάσεων την ικανότητα του αλγορίθμου να αναπροσαρμόζεται αναλόγως των συνθηκών που επικρατούν στο δίκτυο και τέλος θα κάνω μια σύγκριση του αλγορίθμου με κάποιους άλλους.

Για την παρουσίαση των αποτελεσμάτων που θα αναλύσω στην συνέχεια, έτρεξα όλους τους πιθανούς συνδυασμούς των παραμέτρων από 10 φορές την κάθε περίπτωση. Πιο κάτω παρουσιάζεται το πεδίο τιμών που έχω ορίσει για την κάθε μια από τις παραμέτρους μου:

Data Rate = [25 35 45]

A = [0.0 0.2 0.4 0.5 0.6 0.8 1.0]

C = [0.0 0.25 0.5 0.75 1.0]

E = [0.0 0.25 0.5 0.75]

T = [0.5 1.0 1.5 2.0]

Πρέπει να σημειωθεί πως επιλέχθηκαν οι συγκεκριμένες τιμές για το data rate προκειμένου να παρουσιαστούν αποτελέσματα όταν το φόρτο είναι ελαφρύ και δεν παρουσιάζεται πρόβλημα συμφόρησης (data rate=25pkts/sec), όταν υπάρχει μέτριο φόρτο στο δίκτυο και ελαφριά προβλήματα συμφόρησης (data rate=35pkts/sec) και τέλος όταν υπάρχει μεγάλο φόρτο, δηλαδή σοβαρό πρόβλημα συμφόρησης (data rate=25pkts/sec).

4.3 Αλγόριθμοι Σύγκρισης

No congestion:

Ένα πρωτόκολλο που δεν λαμβάνει υπόψη την συμφόρηση κι απλά στέλλει τα πακέτα από το πιο σύντομο μονοπάτι και σε περίπτωση που υπάρχουν πολλαπλά πιθανά μονοπάτια επιλέγει στην τύχη ένα από αυτά.

Congestion Aware Routing Protocol:

Το CAwR είναι ένα τυπικό πρωτόκολλο ελέγχου συμφόρησης το οποίο δρομολογεί τα πακέτα από πολλαπλά μονοπάτια επιλέγοντας κάθε φορά τον κόμβο με την λιγότερη συμφόρηση, δηλαδή με το μικρότερο μέγεθος ουράς. Στηρίζει την λειτουργία του στον εντοπισμό περιοχής με συμφόρηση και στον καθορισμό προτεραιότητας στα διάφορα πακέτα. Τα πακέτα που έχουν υψηλή προτεραιότητα δρομολογούνται μέσω της προβληματικής περιοχής. Τα πακέτα χαμηλής προτεραιότητας που δημιουργούνται έξω από την προβληματική περιοχή προωθούνται σε κόμβους που δεν παρουσιάζουν συμφόρηση ενώ αυτά που δημιουργούνται εντός της περιοχής προωθούνται προς τα έξω. Συνοψίζοντας, οι κόμβοι με προβλήματα συμφόρησης χρησιμοποιούνται μόνο για την μεταφορά πακέτων υψηλής προτεραιότητας. Συγκεκριμένα, για την σύγκριση με τον αλγόριθμο Flock-CC χρησιμοποιήθηκε ένα κάπως διαφοροποιημένο πρωτόκολλο όπου αρχικά δίνεται προτεραιότητα στο συντομότερο μονοπάτι και μόνο σε περίπτωση που υπάρχουν πολλαπλοί κόμβοι με τον μικρότερο αριθμό από hops προς το sink επιλέγεται ο κόμβος του οποίου η ουρά έχει την καλύτερη κατάσταση. [8]

Ad hoc On Demand Distance Vector (AODV) RFC3561:

Ο Ad hoc On Demand Distance Vector αλγόριθμος δρομολόγησης, είναι ένα πρωτόκολλο που σχεδιάστηκε για κινητά ad hoc δίκτυα. Δημιουργεί δρομολόγια μεταξύ των κόμβων μόνο όταν ζητηθεί από τους κόμβους που θέλουν να στείλουν πακέτα και τα διατηρεί για όσο χρονικό διάστημα χρησιμοποιούνται. Ο αλγόριθμος χρησιμοποιεί sequence numbers ως εγγύηση της εγκυρότητας των μονοπατιών. Παρέχει μηχανισμούς για αποφυγή δημιουργίας κύκλων των πακέτων και λειτουργεί καλά για μεγάλο αριθμό από κινητούς κόμβους. Ο AODV δημιουργεί τα μονοπάτια χρησιμοποιώντας τα μηνύματα Route Request (RREQ) και Route Reply (RREP). Όταν κάποιος κόμβος θέλει να στείλει κάποιο πακέτο και δεν έχει κάποιο μονοπάτι προς τον παραλήπτη κόμβο, τότε κάνει broadcast ένα RREQ σε όλο το δίκτυο. Όποιος κόμβος λάβει αυτό το πακέτο αναβαθμίζει τις πληροφορίες στον πίνακα με τα μονοπάτια σχετικά με τον αποστολέα και μπορεί να στείλει πίσω ένα RREP σε περίπτωση που είτε είναι ο ίδιος ο κόμβος προορισμού είτε γνωρίζει κάποιο μονοπάτι προς αυτόν. Σε αντίθετη περίπτωση κάνει και αυτός broadcast με την σειρά του το RREQ σε περίπτωση που είναι η πρώτη φορά που το έχει λάβει διαφορετικά το καταστρέφει και δεν το προωθεί παρακάτω.

Καθώς το RREP επιστρέφει πίσω προς τον αποστολέα, οι κόμβοι ενημερώνουν τους πίνακες τους σχετικά με το μονοπάτι προς τον προορισμό. Ενώσω το μονοπάτι παραμένει ενεργό οι κόμβοι θα συνεχίσουν να το διατηρούν στους πίνακες τους. Με τον όρο ενεργό εννοούμε πως περιοδικά στέλλονται δεδομένα από εκεί. Εάν ο κόμβος που στέλλει τα δεδομένα σταματήσει να στέλλει πακέτα και περάσει κάποιο συγκεκριμένο χρονικό διάστημα, τότε το μονοπάτι θα διαγραφεί από τους πίνακες δρομολόγησης των ενδιάμεσων κόμβων. Επίσης, χρησιμοποιεί τα Hello μηνύματα για να εντοπίζει τους ενεργούς γειτονικούς κόμβους τα οποία στέλλονται σε ένα τυχαίο σε διάστημα $0.75 \cdot T$ και $1.25 \cdot T$. [5]

Αλγόριθμος	Έλεγχος Συμφόρησης	Τρόπος Δρομολόγησης
Flock-CC	Κατάσταση στις ουρές των κόμβων, κατάσταση ασύρματου μέσου	Πολλαπλά μονοπάτια
No Congestion	Κανένας	Συντομότερο μονοπάτι
CAwR	Κατάσταση στις ουρές	Πολλαπλά μονοπάτια
AODV	Κανένας	Ένα μονοπάτι μόνο

Πίνακας 4.1: Κύρια χαρακτηριστικά των αλγορίθμων σύγκρισης

Κεφάλαιο 5

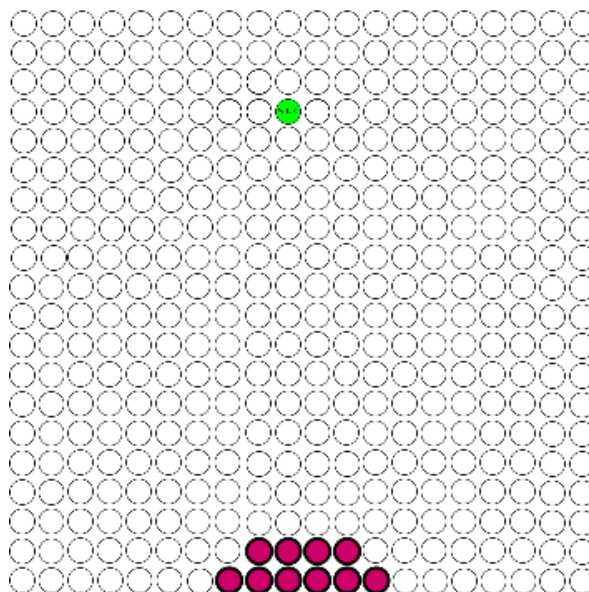
Αξιολόγηση βασικής λειτουργίας αλγορίθμου

5.1	Ομάδα δημιουργίας πακέτων απέναντι από το sink.....	35
5.2	Ομάδα δημιουργίας πακέτων στο πλάι	54
5.3	Δύο ομάδες δημιουργίας πακέτων απέναντι από το sink.....	72

Από αυτό το σημείο και μετά ξεκινά η αξιολόγηση των αποτελεσμάτων για τα σενάρια που επέλεξα να μελετήσω. Περισσότερες πληροφορίες σχετικά με τις παραμέτρους, τις τιμές που επέλεξα να χρησιμοποιήσω και τον λόγο εκλογής του κάθε σεναρίου μπορείτε να βρείτε στο κεφάλαιο 4.

5.1 Ομάδα δημιουργίας πακέτων απέναντι από το sink

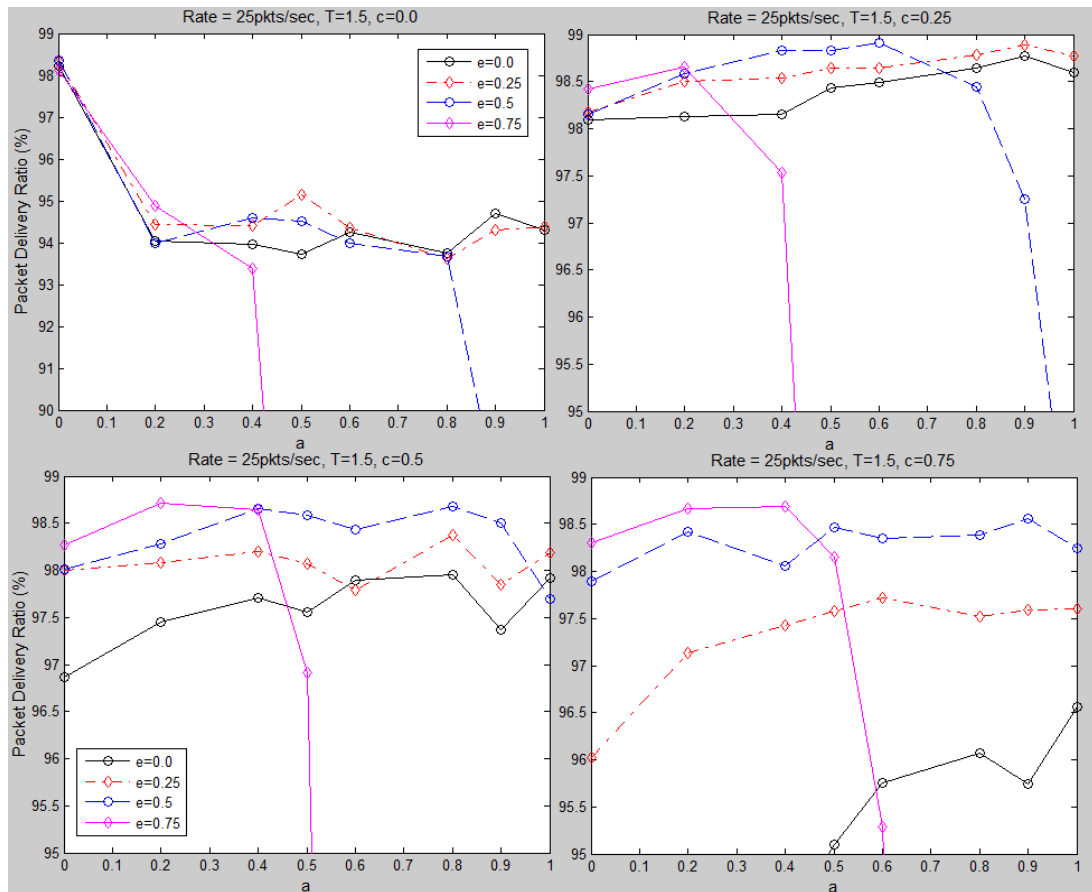
Σε αυτό το σενάριο έχουμε 10 ενεργοποιημένους κόμβους που στέλλουν δεδομένα προς το σημείο sink το οποίο βρίσκεται ακριβώς απέναντι τους (βλέπε σχήμα 5.1). Εκ πρώτης όψεως μπορούμε να παρατηρήσουμε πως υπάρχουν πολλαπλά εναλλακτικά μονοπάτια με την συντομότερη απόσταση προς το sink, κάτι το οποίο σαφώς μπορεί να συμβάλει στην αύξηση του ποσοστού επιτυχημένης αποστολής για τον οποιοδήποτε αλγόριθμο. Αναμένουμε λοιπόν πως ο αλγόριθμος θα λειτουργήσει με τέτοιο τρόπο ώστε να εκμεταλλευθεί τους ακρινούς κόμβους που δεν θα παρουσιάζουν προβλήματα συμφόρησης.



Σχήμα 5.1: Τοποθεσία κόμβων που δημιουργούν τα πακέτα

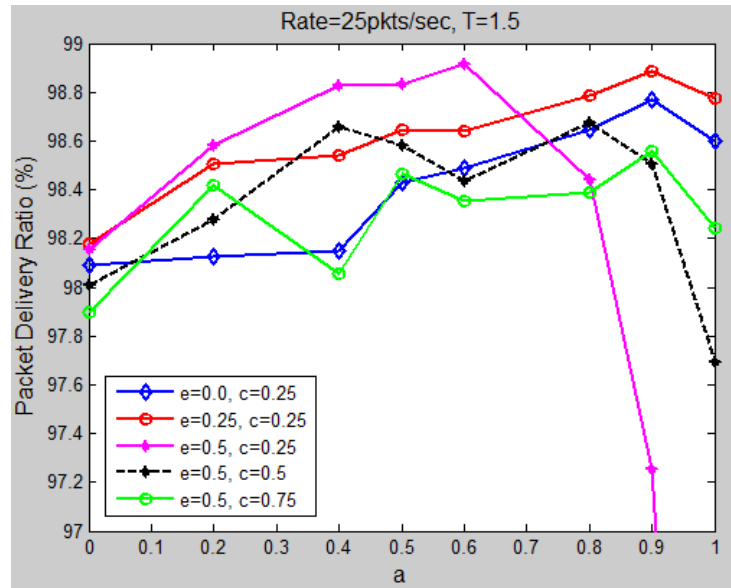
5.1.1 Ελαφριά συμφόρηση (Data Rate=25pkts/sec)

Για data rate ίσο με 25pkts/sec το δίκτυο θεωρείται πως δεν παρουσιάζει προβλήματα συμφόρησης. Μελετώντας τα αποτελέσματα παρατήρησα πως και για τα τρία data rates η ιδανικότερη τιμή για την παράμετρο T (sampling period) είναι το 1.5. Επίσης, παρατήρησα ότι τα καλύτερα αποτελέσματα είχαν το e μεταξύ 0.25 και 0.5, και το c ίσο με 0.25. Δηλαδή, επιτυγχάνονται καλύτερα αποτελέσματα όταν λαμβάνονται με μεγαλύτερη πιθανότητα οι κόμβοι με ίσο hop count και όταν χρησιμοποιείται λίγη τυχειότητα. Χρησιμοποίησα αυτή την πληροφορία για να μπορέσω να υπολογίσω τις τιμές της παραμέτρου a .



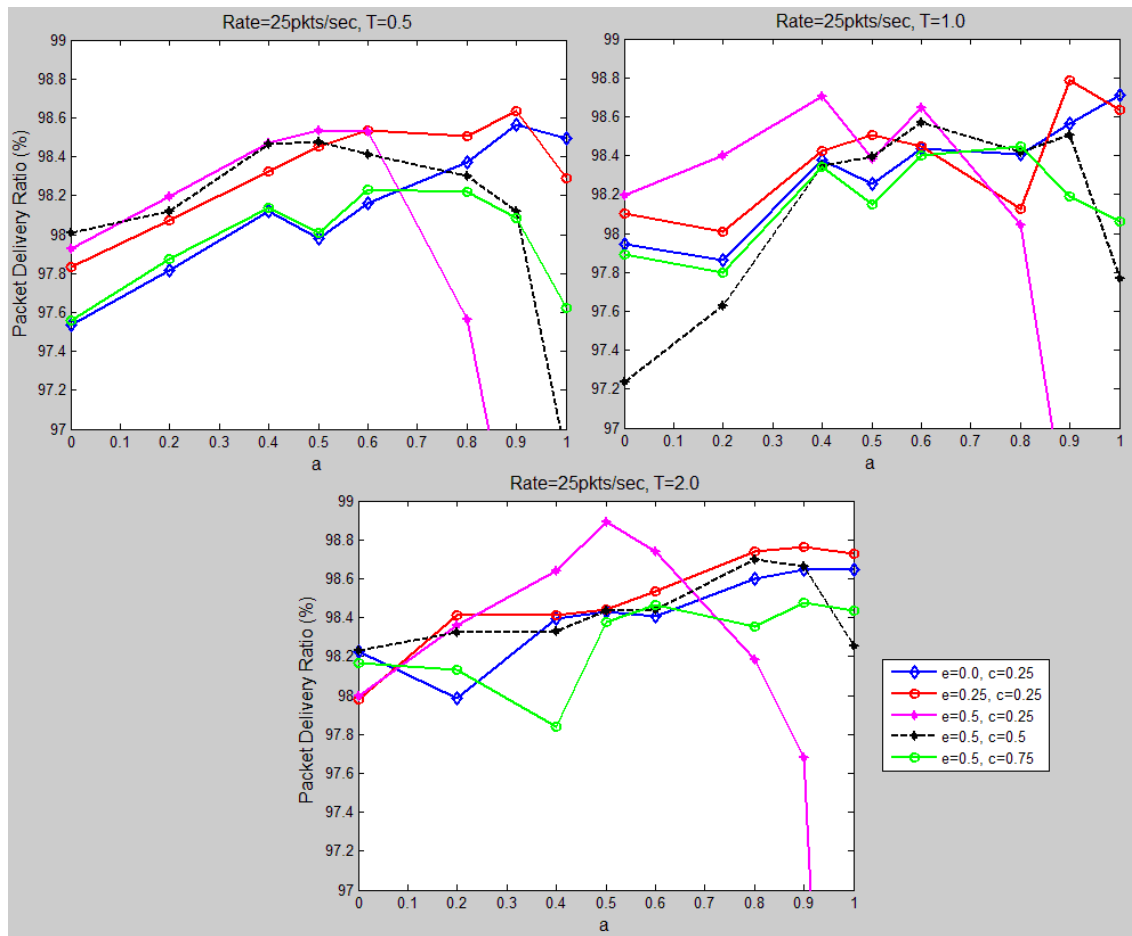
Σχήμα 5.2: Αποτελέσματα PDR (T=1.5)

Στο σχήμα 5.2 παρουσιάζονται τα αποτελέσματα του PDR για T ίσο με 1.5. Σε κάθε μια από αυτές τις γραφικές, διατηρείται σταθερή η παράμετρος c και μπορούμε να παρατηρήσουμε τα αποτελέσματα των παραμέτρων a και e. Καταρχάς, μπορούμε αμέσως να αποκλείσουμε τις τιμές 0.0, 0.75 για την παράμετρο c αφού τα αποτελέσματα του PDR είναι πολύ χαμηλά. Δεν παρουσιάζεται το c=1.0 διότι είχε εξίσου άσχημα αποτελέσματα. Επομένως, μελετώντας τα αποτελέσματα για c=0.25 και c=0.5 βλέπουμε πως είχαμε ψηλότερο PDR για e ίσο με 0.25 και 0.5. Από τη στιγμή που ο αλγόριθμος μιμείται την συμπεριφορά των πουλιών χρειάζεται να αναπαρασταθεί με κάποιο τρόπο η τυχαιότητα, κι αυτό γίνεται με την βοήθεια της παραμέτρου c. Όταν το c είναι πολύ μεγάλο ή ίσο με το μηδέν δεν έχουμε καλά αποτελέσματα και το PDR είναι πολύ χαμηλό. Για μεγάλες τιμές του c οι κόμβοι επιλέγουν με τυχαίο τρόπο τους κόμβους στους οποίους θα προωθήσουν τα πακέτα κι αυτό οδηγεί στα χαμηλά αποτελέσματα που είδαμε. Φαίνεται επίσης, πως όταν δεν χρησιμοποιείται καθόλου η τυχαιότητα και πάλι δεν έχουμε τόσο καλά αποτελέσματα.



Σχήμα 5.3: Καλύτερα αποτελέσματα PDR (T=1.5)

Στο σχήμα 5.3 παρουσιάζονται οι συνδυασμοί των παραμέτρων που δίνουν τα ψηλότερα ποσοστά PDR. Όπως μπορούμε να παρατηρήσουμε, τα καλύτερα αποτελέσματα δίνονται για $e=0.5$, $c=0.25$ και a μεταξύ 0.4 και 0.6. Πάρα πολύ καλά αποτελέσματα είχαμε επίσης για $e=0.25$. Όταν το $e=0$ τα πακέτα πηγαίνουν ευθεία προς το sink, μειώνοντας πολύ την ενέργεια των κόμβων στο συγκεκριμένο μονοπάτι (άρα μειώνεται και η συνολική διάρκεια ζωής του δικτύου) ενώ παράλληλα αυξάνεται πολύ και η συμφόρηση και οι καθυστερήσεις σε αυτή την περιοχή. Εάν το e ήταν μεγαλύτερο από 0.75 τότε θα είχαμε αντίστροφα αποτελέσματα, δηλαδή τα πακέτα μπορούν να επιλέξουν και κόμβους με ίσο hop count προς το sink με μεγάλη πιθανότητα κι αυτό μπορεί να προκαλέσει κύκλους των πακέτων ανάμεσα σε κόμβους που δεν έχουν συμφόρηση με αποτέλεσμα την μείωση του PDR.

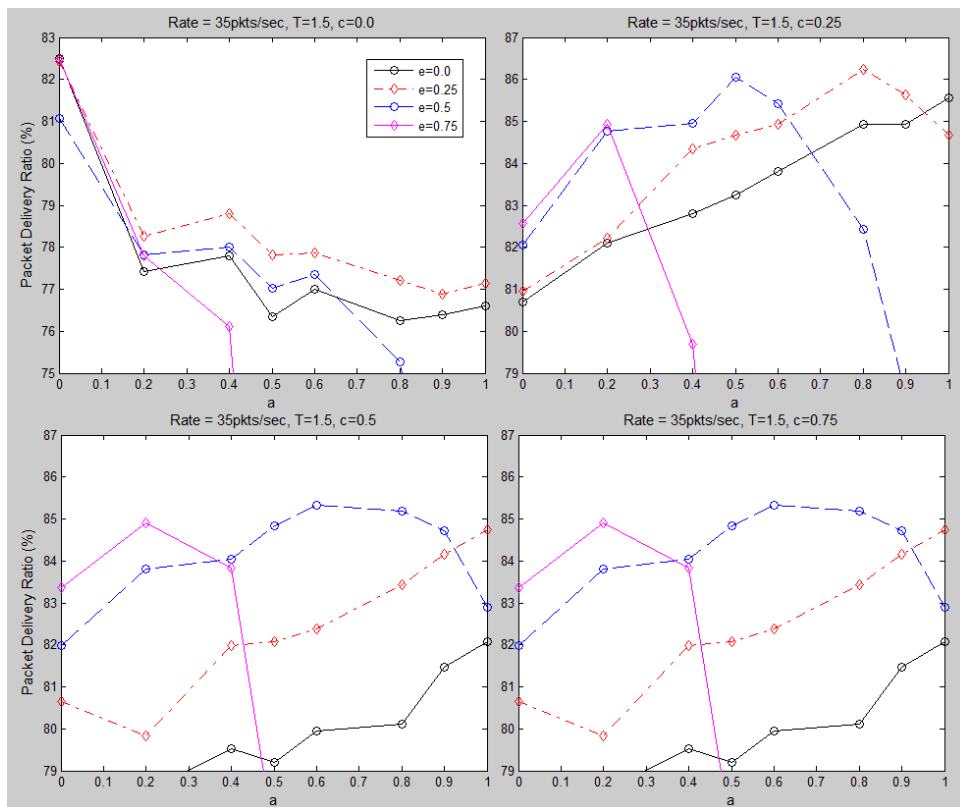


Σχήμα 5.4: Σύγκριση PDR για τις υπόλοιπες τιμές του T

Στο σχήμα 5.4 φαίνονται τα ίδια αποτελέσματα για τις υπόλοιπες τιμές του T. Παρατηρούμε πως για $T=0.5$ έχουμε χαμηλότερο PDR. Ελαφρώς χαμηλότερα αποτελέσματα είχαμε και για τιμή 2.0. Για $T=1.0$ μπορούμε να πούμε πως τα αποτελέσματα είναι πάρα πολύ κοντινά. Παρόλα αυτά προτιμάται η τιμή 1.5 για να μην πλημμυρίζει το δίκτυο με αχρείαστα δεδομένα σπαταλώντας με αυτό τον τρόπο τα ενεργειακά αποθέματα των κόμβων.

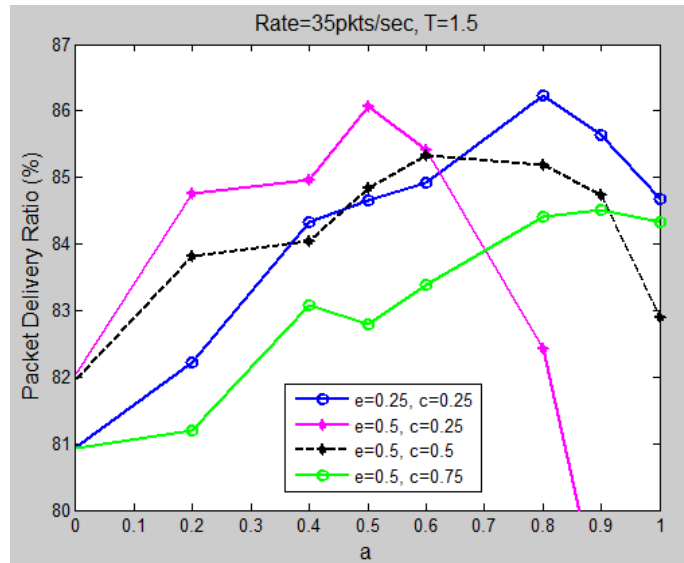
5.1.2 Πρόβλημα Συμφόρησης (Data Rate=35pkts/sec)

Εκ πρώτης όψεως φαίνεται πως τα καλύτερα αποτελέσματα για transmission rate=35pkts/sec είναι όταν $c=0.25$, $e=0.5$ και $T=1.5$, όπως και προηγουμένως. Θα κρατήσω σταθερή την παράμετρο T σε 1.5 και θα παρουσιάσω τα αποτελέσματα για τις υπόλοιπες παραμέτρους και στο τέλος θα κάνω μια σύγκριση με τις υπόλοιπες τιμές της περιόδου δειγματοληψίας.



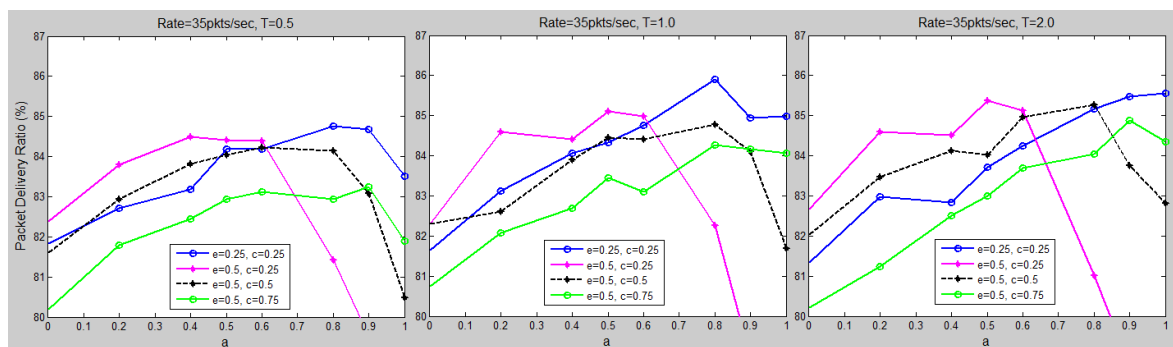
Σχήμα 5.5: Αποτελέσματα PDR ($T=1.5$)

Όπως παρατηρούμε στο σχήμα 5.5, τα καλύτερα αποτελέσματα είναι για c μεταξύ 0.25 και 0.75. Για $c=0.0$ υπήρχε πολύ μεγάλη μείωση στο packet delivery ratio. Κάτι το οποίο παρατηρήσαμε και προηγουμένως για rate ίσο με 25pkts/sec. Όπως μπορούμε να δούμε, η ιδανικότερη τιμή για το e είναι το 0.5 για όλες τις τιμές της παραμέτρου c .



Σχήμα 5.6: Συνολικά καλύτερα αποτελέσματα PDR (T=1.5)

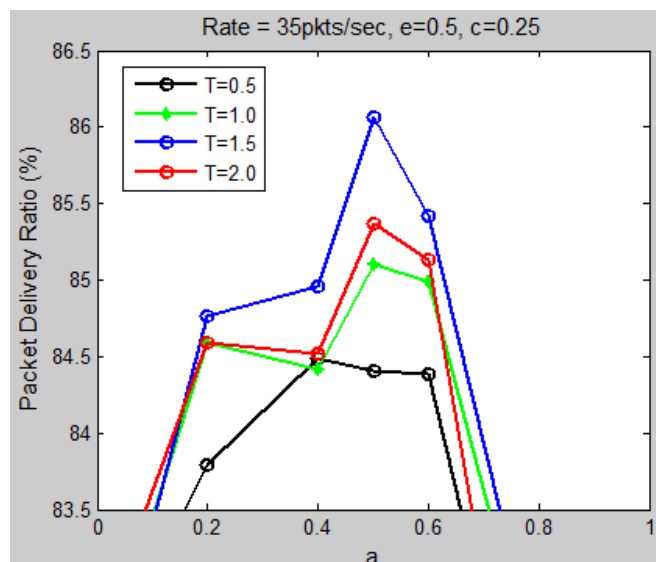
Στο σχήμα 5.6 παρουσιάζονται συνολικά τα καλύτερα αποτελέσματα. Όπως μπορούμε να παρατηρήσουμε τα καλύτερα αποτελέσματα έχουν το c και το e ίσο με 0.25 και 0.5 αντίστοιχα. Όσον αφορά την παράμετρο a δεν μπορούμε να απομονώσουμε μια μόνο τιμή ως την καλύτερη όμως παρόλα αυτά μπορούμε να δούμε ότι τα καλύτερα αποτελέσματα κυμαίνονται μεταξύ 0.4 και 0.8 όπου το PDR είναι μεταξύ 85% και 86%. Η τιμή 0.5 για την παράμετρο e είναι πολύ φυσιολογική αφού τα πακέτα προσπαθούν να ανοίξουν και να εκμεταλλευτούν περισσότερους κόμβους και όχι μόνο αυτούς που βρίσκονται στην κεντρική ευθεία μεταξύ της ομάδας αποστολής και του sink.



Σχήμα 5.7: Σύγκριση καλύτερων αποτελεσμάτων για τις υπόλοιπες τιμές του T

Στο σχήμα 5.7 παρουσιάζονται τα ίδια αποτελέσματα, δηλαδή οι καλύτερες περιπτώσεις για τις υπόλοιπες τιμές της παραμέτρου T. Το sampling period καθορίζει

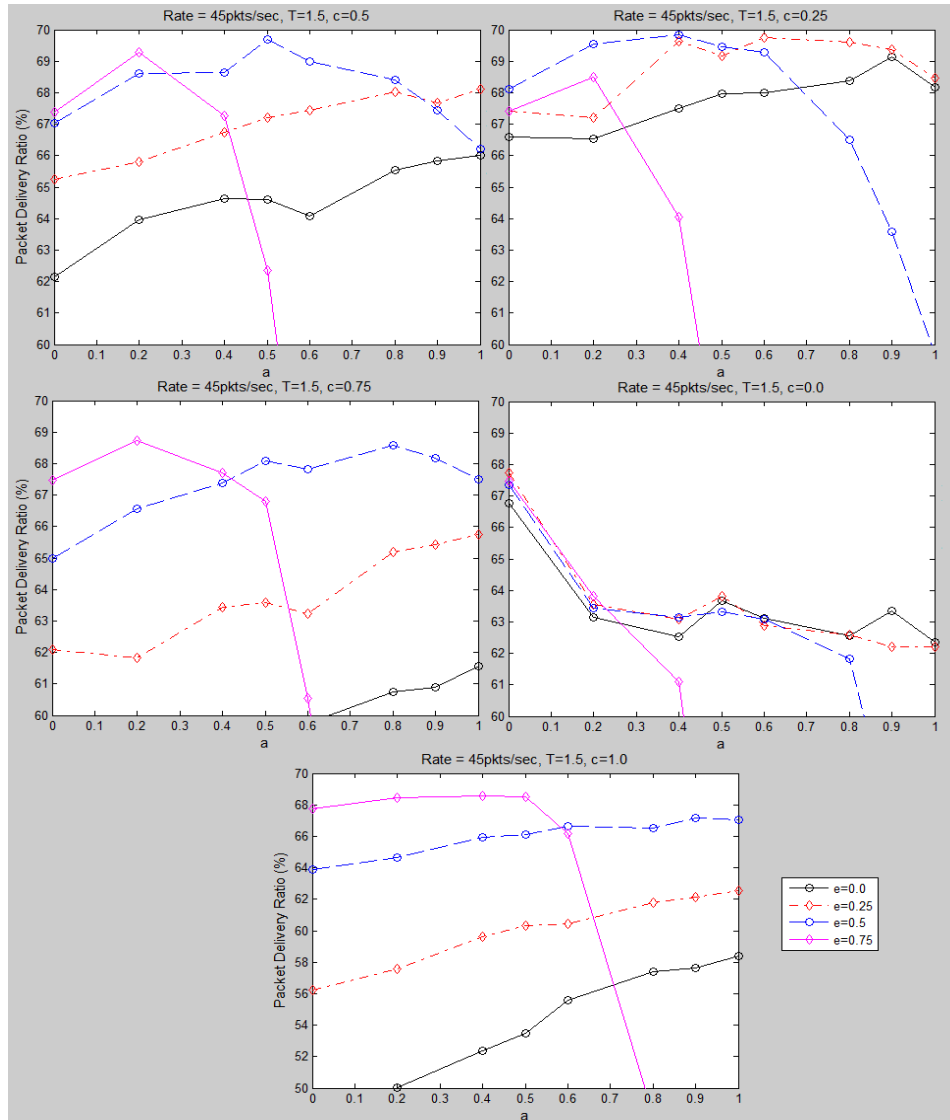
το πόσο σύντομα θα στέλλονται τα control packets. Δεδομένης της τοπολογίας του δικτύου και της θέσης των κόμβων που στέλλουν δεδομένα φαίνεται πως δεν χρειάζεται να λαμβάνουν τόσο σύντομα τα control packets για να μπορούν να υπολογίζουν περιοχές με συμφόρηση. Οι κόμβοι μπορούν να στείλουν τα δεδομένα τους εκμεταλλευόμενοι ολόκληρο το δίκτυο κι έτσι δεν προκύπτουν περιοχές με συμφόρηση με εξαίρεση φυσικά την περιοχή όπου βρίσκεται το sink. Έτσι οι κόμβοι μπορούν να λαμβάνουν τα control messages σε μεγαλύτερο sampling period γι' αυτό και φαίνεται πως τα καλύτερα αποτελέσματα είναι για T ίσο με 1.5. Το ίδιο αποτέλεσμα φαίνεται και στο σχήμα 5.8 όπου διατηρώ σταθερό το e σε 0.5 και το c σε 0.25 προκειμένου να δω τις διαφορές ανάμεσα στις τιμές του T .



Σχήμα 5.8: Σύγκριση PDR ως προς το T

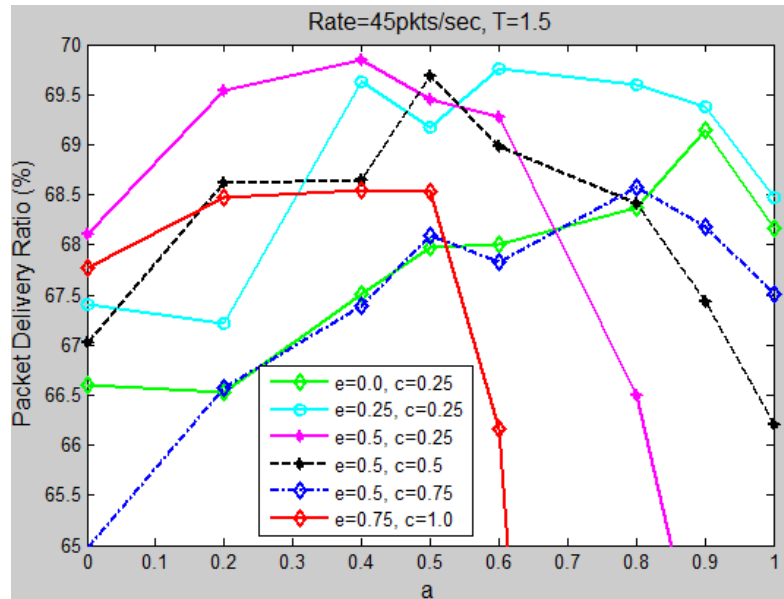
5.1.3 Σοβαρό πρόβλημα συμφόρησης (Data Rate=45pkts/sec)

Όπως και προηγουμένως, θα διατηρήσω σταθερό το T σε 1.5 και θα μελετήσω τις υπόλοιπες παραμέτρους και στο τέλος θα κάνω μια σύγκριση και για τις υπόλοιπες τιμές του sampling period.



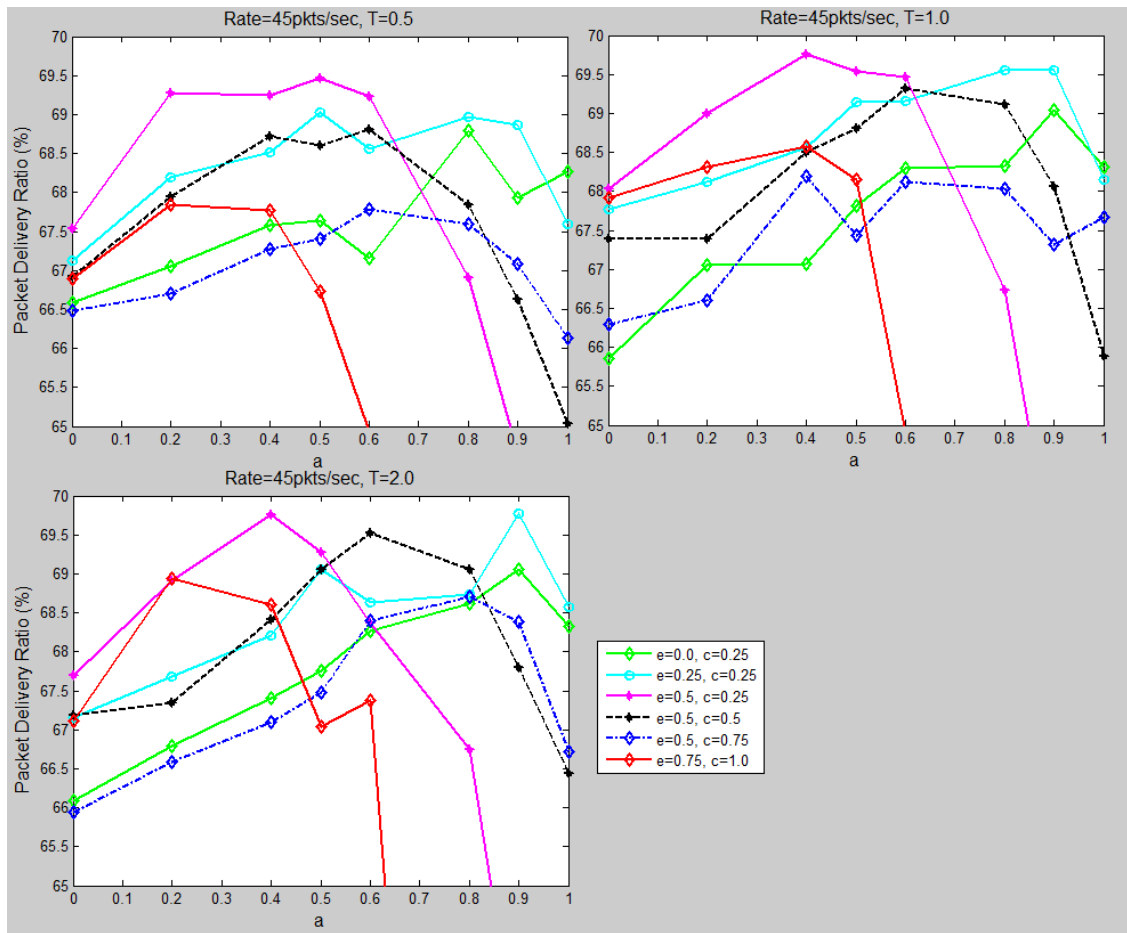
Σχήμα 5.9: Αποτελέσματα PDR για $T=1.5$

Στο σχήμα 5.9 παρατηρούμε τα αποτελέσματα για το PDR όταν $T=1.5$ για τις διάφορες τιμές της παραμέτρου c . Όπως μπορούμε να δούμε για $c=0.0$ και $c=1.0$ το PDR ήταν πολύ χαμηλότερο παρά στις άλλες περιπτώσεις. Εκ πρώτης όψεως, φαίνεται πως οι ιδανικότερες τιμές για την παράμετρο c είναι το 0.25 και το 0.5 το ίδιο και για το e .



Σχήμα 5.10: Συνολικά καλύτερα αποτελέσματα PDR

Στο σχήμα 5.10 παρουσιάζονται συνολικά τα καλύτερα αποτελέσματα του Packet Delivery Ratio. Όπως μπορούμε να δούμε, το ψηλότερο ποσοστό είναι περίπου 69.9%. Η ιδανικότερη τιμή για τις παραμέτρους e και c φαίνεται να είναι το 0.25-0.5. Όσον αφορά την παράμετρο a και πάλι δεν μπορούμε να ορίσουμε με μεγάλη ακρίβεια την ιδανικότερη της τιμή όμως φαίνεται να κυμαίνεται μεταξύ 0.2 και 0.8. Στην προκειμένη περίπτωση, για data rate ίσο με 45pkts/sec, τα καλύτερα αποτελέσματα μπορούμε να πούμε πως είναι για: **$a=0.4$, $e=0.5$, $c=0.25$** .



Σχήμα 5.11: Σύγκριση αποτελεσμάτων PDR για τις υπόλοιπες τιμές του T

Στο σχήμα 5.11 φαίνονται ακριβώς τα ίδια αποτελέσματα για τις υπόλοιπες τιμές του Sampling period. Όπως μπορούμε να δούμε, τα αποτελέσματα είναι πάρα πολύ κοντινά. Μπορούμε να διακρίνουμε όμως μια ελαφριά πτώση στο PDR όταν το T ισούται με 0.5. Τέλος, αφού για ψηλές τιμές του T είχαμε καλά αποτελέσματα είναι καλύτερα να επιλέξουμε μια από τις ψηλότερες τιμές, δηλαδή 1.5 ή 2.0, αφού με αυτό τον τρόπο δεν θα στέλλονται τόσο συχνά τα control packets και συνεπώς, δεν θα σπαταλούνται οι πόροι του δικτύου με αχρείαστο φόρτο.

5.1.4 Παρατηρήσεις

Περίοδος δειγματοληψίας (T): Παρατήρησα πως για όλα τα data rates η καλύτερη τιμή για την περίοδο δειγματοληψίας (T) ήταν το 1.5. Δεδομένης της τοπολογίας του δικτύου και της θέσης των κόμβων που στέλλουν δεδομένα, φαίνεται πως δεν χρειάζεται να λαμβάνουν τόσο σύντομα τα control packets για να μπορούν να υπολογίζουν περιοχές με συμφόρηση. Έτσι οι κόμβοι μπορούν να λαμβάνουν τα control messages μετά από μεγαλύτερο χρονικό διάστημα γι' αυτό και φαίνεται πως τα καλύτερα αποτελέσματα είναι για T ίσο με 1.5.

$$T=1.5$$

Παράμετρος e: Όσον αφορά την παράμετρο e είδαμε πως οι καλύτερες τιμές ήταν 0.25 και 0.5. Ήταν αναμενόμενο διότι θέλουμε να λαμβάνουμε υπόψη και κόμβους που έχουν ίση απόσταση σε hops από το sink. Όταν το e ισούται με το μηδέν τα πακέτα πηγαίνουν όλο ευθεία προς το sink, μειώνοντας τα ενεργειακά αποθέματα των κόμβων στο συγκεκριμένο μονοπάτι, προκαλώντας συμφόρηση και αυξάνοντας τις καθυστερήσεις. Σε αντίθετη περίπτωση, όπου το e παίρνει μεγάλες τιμές (>0.75), τα πακέτα επιλέγουν τυχαία μονοπάτια σε κόμβους που δεν έχουν προβλήματα συμφόρησης με αποτέλεσμα να δημιουργούνται κύκλοι. Έτσι καταλήγω στο συμπέρασμα πως οι ιδανικότερες τιμές για την παράμετρο e είναι το 0.25 και το 0.5.

$$E=\{0.25, 0.5\}$$

Παράμετρος c: Όσον αφορά την παράμετρο c φαίνεται καθαρά πως η ιδανική της τιμή είναι το 0.25. Από την στιγμή που υλοποιούμε ένα αλγόριθμο ο οποίος μιμείται την συμπεριφορά των πουλιών είναι αναμενόμενο πως πρέπει να εισάγεται μέσα το στοιχείο της τυχειότητας, το οποίο δεν θέλουμε να είναι πολύ μεγάλο. Όταν το c είναι πολύ μεγάλο βλέπουμε πως έχουμε μειωμένο ποσοστό επιτυχημένης αποστολής δεδομένων. Το ίδιο ισχύει όταν το $c=0$, διότι πλέον τα πακέτα πηγαίνουν απλά προς το sink ακολουθώντας συγκεκριμένα μονοπάτια.

$$C=0.25$$

Παράμετρος a: Σχετικά με την παράμετρο a , δεν μπορούμε να ορίσουμε με μεγάλη ακρίβεια την ιδανική της τιμή, ξέρουμε σίγουρα όμως πως δεν μπορεί να είναι ίση ούτε με το μηδέν αλλά ούτε και με το 1. Η κατάσταση του δικτύου αλλά και η κατάσταση στις ουρές των κόμβων είναι εξίσου σημαντικές. Αυστηρά μιλώντας φαίνεται πως το διάστημα καλών τιμών για τη συγκεκριμένη παράμετρο είναι μεταξύ 0.2 και 0.6. Υπήρχαν όμως και κάποιες περιπτώσεις στις οποίες η τιμή 0.8 έδωσε καλά αποτελέσματα. Και για τα τρία data rates οι διαφορές στο PDR για την μεταβλητή a ήταν πάρα πολύ μικρές και συγκρίνοντας τα αποτελέσματα τους μπορούμε να πούμε πως η ιδανικότερη τιμή είναι το 0.5.

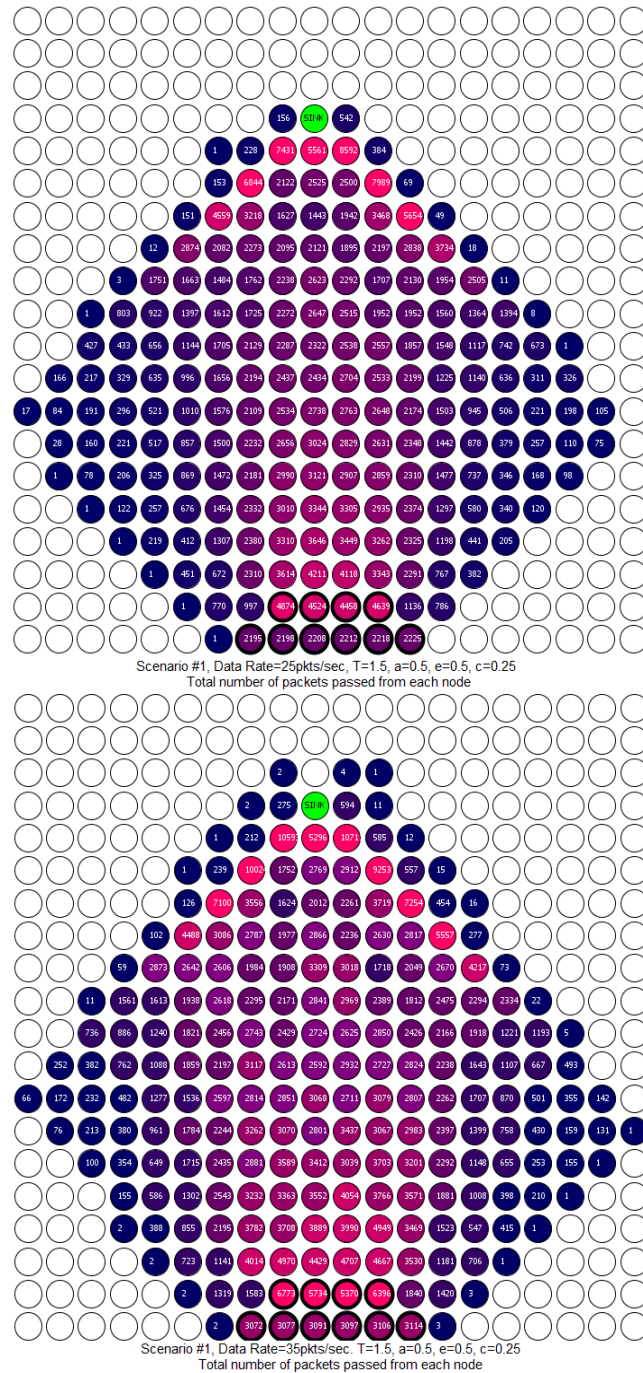
$A \in \{0.2, 0.4, 0.5, 0.6, 0.8\}$ γενικά.

Σύμφωνα με τα πιο πάνω και αυστηρά μιλώντας πάντοτε, αποφάσισα πως ο καλύτερος συνδυασμός τιμών για το σενάριο #1 για όλα τα data rates είναι ο εξής:

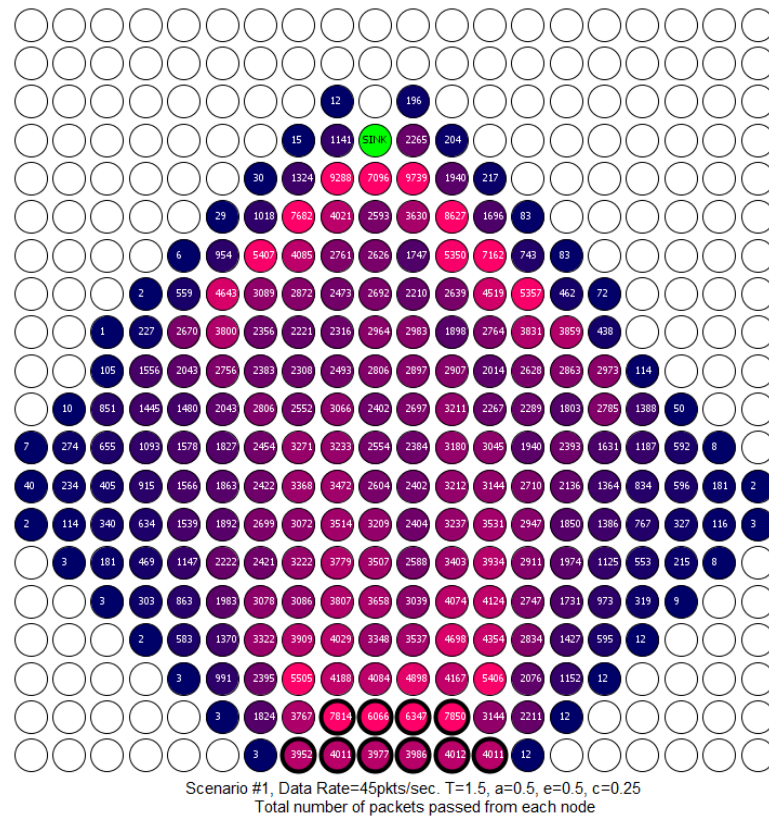
$T=1.5, A=0.5, E=0.5, C=0.25.$

5.1.5 Γραφική απεικόνιση κίνησης πακέτων

Στο σχήμα 5.12 και 5.13 παρουσιάζεται ο συνολικός αριθμός πακέτων που έχουν περάσει από τον κάθε κόμβο με τιμές παραμέτρων ίσες με: **T=1.5, A=0.5, E=0.5, C=0.25** για τα τρία data rates.



Σχήμα 5.12: Συνολικός αριθμός πακέτων που πέρασαν από τον κάθε κόμβο

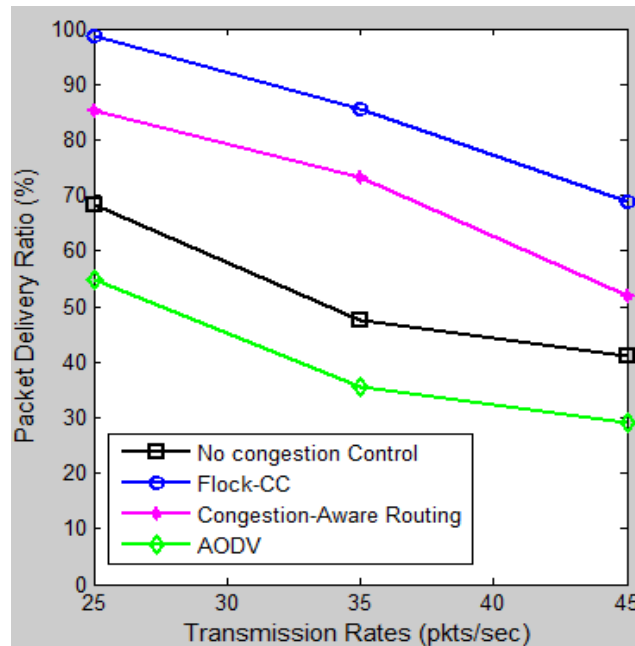


Σχήμα 5.13: Συνολικός αριθμός πακέτων που πέρασαν από τον κάθε κόμβο

Όπως μπορούμε να δούμε, για όλα τα data rates ο αλγόριθμος λειτουργεί αποδοτικά και το κοπάδι των πακέτων ανοίγει προκειμένου να αποφύγει τις περιοχές συμφόρησης. Στο σχήμα 5.13, το οποίο αναπαριστά την κίνηση των πακέτων για rate 45pkts/sec, η συμφόρηση προκαλείται στην σειρά των κόμβων που βρίσκονται στην ευθεία μεταξύ του sink και της ομάδας των κόμβων που δημιουργούν τα πακέτα. Παρατηρούμε πως καθώς αυξάνεται το data rate, και συνεπώς η συμφόρηση, το κοπάδι ανοίγει όλο και περισσότερο σε σημείο που ακόμη και οι ακραίοι κόμβοι και κόμβοι πίσω από το sink, λαμβάνουν πακέτα. Με αυτό τον τρόπο γίνεται ομοιόμορφη κατανομή των πακέτων, σχεδόν σε όλο το δίκτυο και συνεπώς αυξάνεται η συνολική διάρκεια ζωής του.

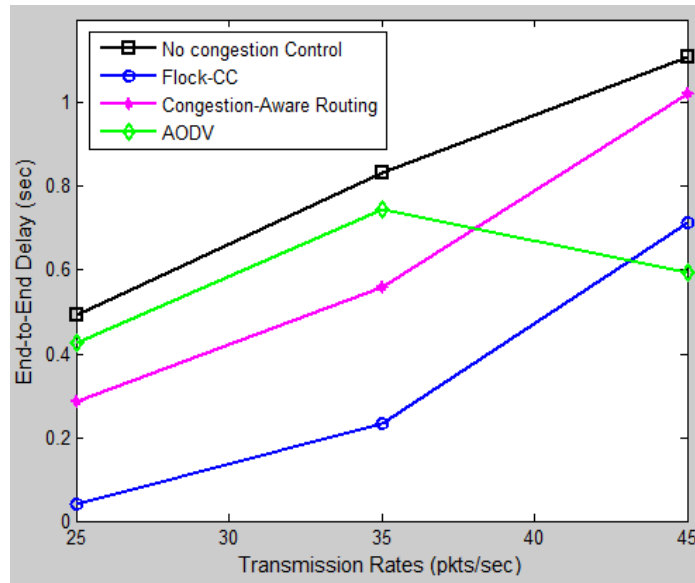
5.1.6 Σύγκριση αλγορίθμων

Και για τα τρία data rates χρησιμοποίησα τον ιδανικότερο συνδυασμό παραμέτρων ο οποίος είναι: $T=1.5$, $A=0.5$, $E=0.5$, $C=0.25$.



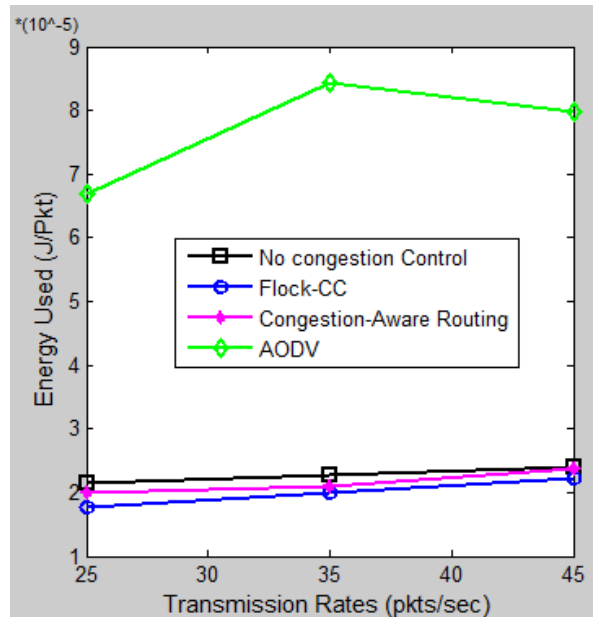
Σχήμα 5.14: Αποτελέσματα PDR

Στο σχήμα 5.14 παρουσιάζονται τα αποτελέσματα του PDR για τα τρία transmission rates για τους τέσσερις αλγορίθμους. Μπορούμε να δούμε πως υπάρχει τεράστια διαφορά ανάμεσα στα αποτελέσματα. Ο καλύτερος αλγόριθμος είναι αυτός που μελετάω αφού δίνει ποσοστά 98.82%, 85.57% και 68.76% για data rates ίσα με 25, 35 και 45 pkts/sec αντίστοιχα. Βλέπουμε μια διαφορά της τάξεως του 15% ανάμεσα στον αλγόριθμο Flock-CC και στον CAwR ο οποίος δίνει αποτελέσματα 85.13%, 73.27% και 51.84%. Έπειτα, ο No congestion control αλγόριθμος δίνει ποσοστά 68.40%, 47.53% και 41.11% για data rates ίσα με 25, 35 και 45pkts/sec αντίστοιχα. Τέλος, ο AODV δίνει το χαμηλότερο PDR το οποίο είναι ίσο με 54.71%, 35.55% και 29.19% για την ίδια ακολουθία data rates.



Σχήμα 5.15: Αποτελέσματα καθυστερήσεων

Στο σχήμα 5.15 παρουσιάζονται οι συνολικές καθυστερήσεις των επιτυχημένα αποσταλθέντων πακέτων. Μπορεί οι καθυστερήσεις γενικά να μην ήταν πάρα πολύ μεγάλες όμως ο Flock-CC αλγόριθμος είχε τον μικρότερο χρόνο για όλες τις περιπτώσεις σχεδόν. Για data rates ίσα με 25, 35 και 45pkts/sec, ο No Congestion Control αλγόριθμος δίνει καθυστερήσεις 0.493sec, 0.834sec και 1.11sec αντίστοιχα, ενώ ο CAwR δίνει 0.285sec, 0.558sec και 1.023sec. Ο Flock-CC δίνει 0.042sec, 0.235sec και 0.714sec. Τέλος, ο AODV έχει 0.427sec, 0.747sec και 0.595sec για rates ίσα με 25, 35 και 45pkts/sec. Το παράδοξο με τον AODV είναι πως είχε μικρότερες καθυστερήσεις για data rate ίσο με 45 παρά για 35pkts/sec. Η μόνη περίπτωση που κάποιος από τους αλγορίθμους σύγκρισης είχε χαμηλότερες καθυστερήσεις από τον Flock-CC, ήταν ο AODV για rate ίσο με 45pkts/sec και η διαφορά ήταν της τάξεως των 0.1sec, κόστος πολύ μικρό δεδομένου πως το PDR για την προκειμένη περίπτωση ήταν 68.76% για τον Flock-CC και μόλις 29.19% για τον AODV.



Σχήμα 5.16: Αποτελέσματα δαπανούμενης ενέργειας ανά πακέτο

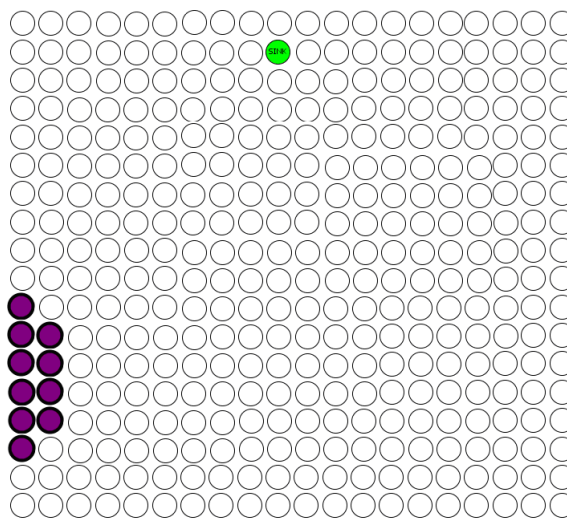
Στο σχήμα 5.16 παρουσιάζεται η ενέργεια που σπαταλήθηκε για το κάθε πακέτο. Υπολογίζεται ως το ποσό της συνολικής ενέργειας του δικτύου που σπαταλήθηκε δια τον συνολικό αριθμό πακέτων που κατάφεραν να παραδοθούν επιτυχώς στον προορισμό τους. Καταρχάς παρατηρούμε πως ο Flock-CC είχε την χαμηλότερη κατανάλωση η οποία ανέρχεται σε $1.77 \cdot 10^{-5}$, $1.99 \cdot 10^{-5}$ και $2.23 \cdot 10^{-5}$ Joules per delivered packet για data rates ίσα με 25, 35 και 45 pkts/sec αντίστοιχα. Πολύ κοντινά αποτελέσματα αν και ελαφρώς χειρότερα είχαν οι αλγόριθμοι CAwR και No Congestion Control. Όπως μπορούμε να δούμε ο AODV σπαταλά τετραπλάσιο ποσό ενέργειας συγκριτικά με τους άλλους αλγορίθμους, κι όπως είχαμε δει και προηγουμένως, έχει και το χαμηλότερο PDR.

5.1.7 Σύνοψη

Συνοψίζοντας, είδαμε πως για το συγκεκριμένο σενάριο, το οποίο είναι και το πιο απλό, ο αλγόριθμος Flock-CC έδωσε το ψηλότερο ποσοστό όσον αφορά το PDR και τις μικρότερες καθυστερήσεις και ποσό ενέργειας ανά πακέτο. Παρατηρήσαμε πως ο αλγόριθμος μπορεί να αντιληφθεί πως υπάρχουν πολλαπλά μονοπάτια και τα αξιοποιεί καταλλήλως. Είδαμε πως εκμεταλλεύτηκε ολόκληρο το πλάτος του δικτύου, μοιράζοντας με αυτό τον τρόπο το traffic load σε περισσότερους κόμβους. Επιπλέον, παρατηρήσαμε πως από την στιγμή που το σενάριο δεν είναι «προβληματικό», δηλαδή δεν υπάρχει και τόσο σοβαρό πρόβλημα συμφόρησης, η περίοδος δειγματοληψίας μπορεί να είναι μεγάλη έτσι με αυτό τον τρόπο επιτυγχάνεται μεγαλύτερη εξοικονόμηση ενέργειας αφού δεν σπαταλούνται οι πόροι του δικτύου για την αποστολή των μηνυμάτων ελέγχου.

5.2 Ομάδα δημιουργίας πακέτων στο πλάι

Σε αυτό το σενάριο έχουμε 10 ενεργοποιημένους κόμβους που στέλλουν δεδομένα προς το σημείο sink το οποίο βρίσκεται διαγώνια αριστερά τους (βλέπε σχήμα 5.17). Όπως θα αναμέναμε, τα πακέτα θα προσπαθήσουν να κινηθούν από το συντομότερο μονοπάτι ως επί το πλείστον όμως λόγω συμφόρησης κανονικά θα πρέπει να «ανοίξουν» αξιοποιώντας τους κόμβους δεξιά και αριστερά της κεντρικής διαγωνίου που δεν έχουν τόσο πολύ φόρτο.



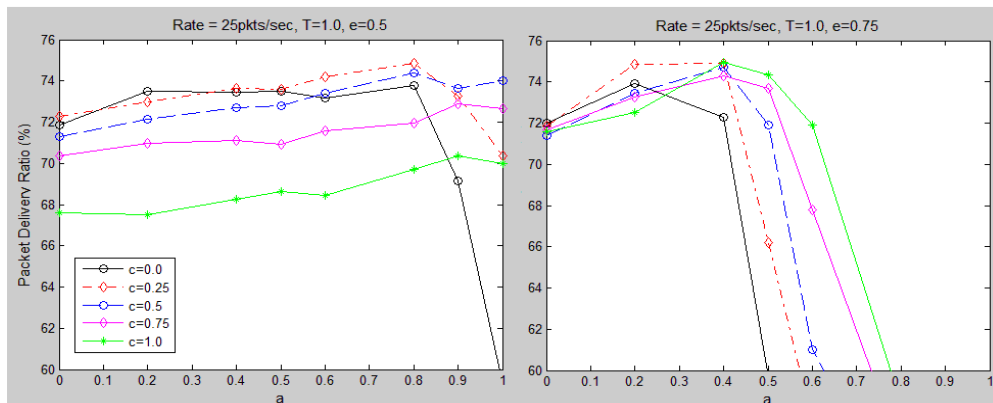
Σχήμα 5.17: Τοποθεσία ομάδας κόμβων δημιουργίας πακέτων

Γενικά παρατήρησα πως το ποσοστό επιτυχημένης αποστολής είναι πάρα πολύ χαμηλό και για τα 3 rates που χρησιμοποίησα. Πιο συγκεκριμένα, για data rate=25pkts/sec το ποσοστό επιτυχημένης αποστολής πακέτων ήταν ίσο με 75.13%, για data rate=35 pkts/sec ήταν 55.90% και τέλος για data rate=45pkts/sec ήταν 44.26%. Το δίκτυο έχει πάρα πολλή συμφόρηση για το λόγο ότι οι κόμβοι που δημιουργούν τα πακέτα βρίσκονται μόλις 8 hops μακριά από το sink στην καλύτερη περίπτωση, ενώ στο πρώτο σενάριο που βρίσκονταν ακριβώς απέναντι του ήταν 14 hops μακριά στην καλύτερη περίπτωση πάντα μιλώντας. Επίσης, στο προηγούμενο σενάριο οι κόμβοι είχαν την δυνατότητα να ανοίξουν πάρα πολύ και να εκμεταλλευτούν όλο το πλάτος του δικτύου και εξάλλου υπήρχαν πολύ περισσότερα μονοπάτια με τον ελάχιστο αριθμό hops τα οποία μπορούσαν να αξιοποιήσουν. Σε αυτή την περίπτωση όμως ο αριθμός των

μονοπατιών με τον ελάχιστο αριθμό hops είναι πάρα πολύ μικρός και τα πακέτα ακολουθούν κυρίως την κεντρική διαγώνιο.

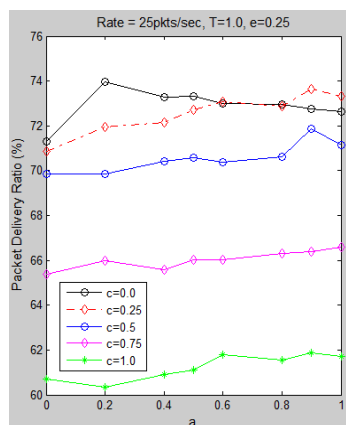
Εκ πρώτης όψεως, φάνηκε πως η καλύτερη τιμή για το e ήταν 0.5-0.75. Το e καθορίζει κατά πόσο θα λαμβάνουμε υπόψη τους κόμβους που έχουν ίσο hop count προς το sink. Επίσης, η παράμετρος c η οποία καθορίζει την τυχαιότητα, ήταν ίση με 0.5 και για τα 3 data rates. Επίσης, παρατήρησα πως για τις περιπτώσεις όπου υπήρχε συμφόρηση (data rates ίσα με 35 και 45pkts/sec), η καλύτερη τιμή για το T ήταν το 0.5. Βλέπουμε λοιπόν πως θέλουμε να έχουμε μικρό sampling period. Μελετώντας τα αποτελέσματα για data rate=25pkts/sec φαίνεται πως η καλύτερη τιμή είναι το 1.0 αν και για όλες τις τιμές είχαμε καλά αποτελέσματα (για τον ίδιο συνδυασμό τιμών παραμέτρων το T είχε καλά αποτελέσματα).

5.2.1 Πρόβλημα Συμφόρησης (Data Rate=25pkts/sec)



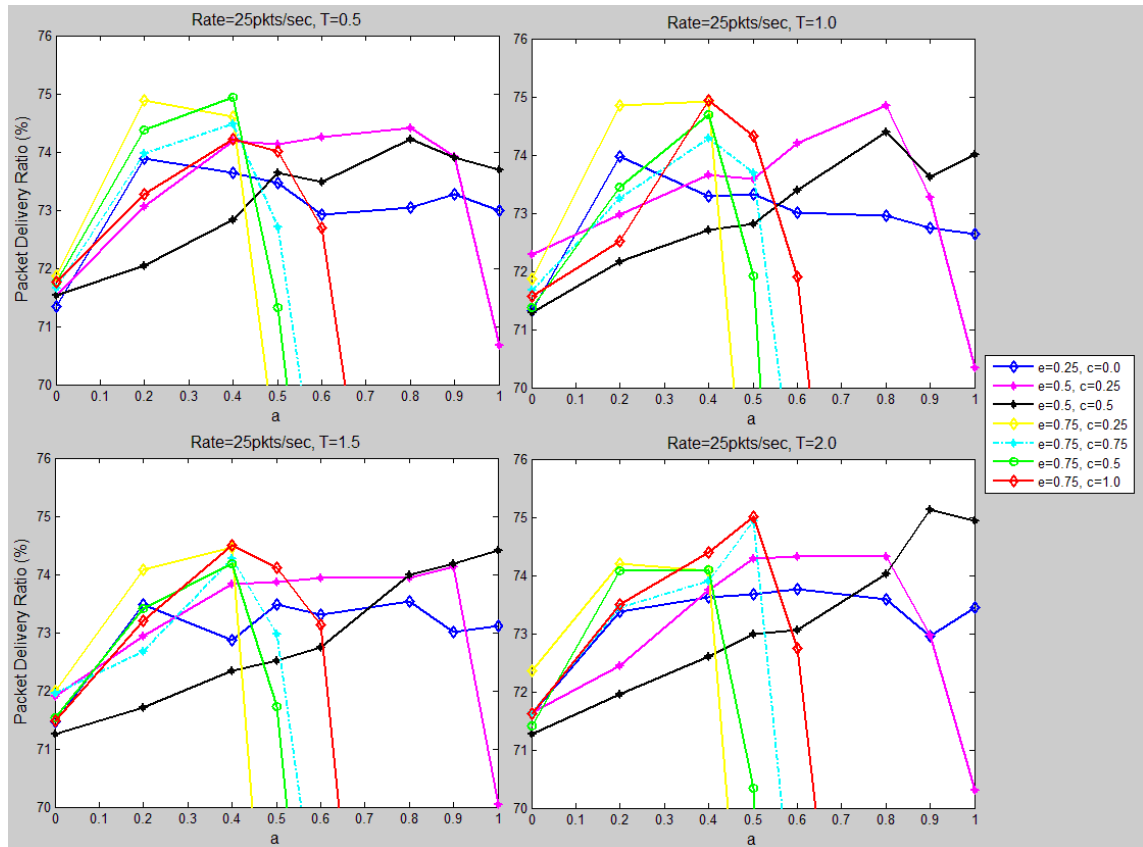
Σχήμα 5.18: Αποτελέσματα PDR

Στο σχήμα 5.18 παρουσιάζονται τα αποτελέσματα του PDR όταν διατηρούμε σταθερό το T σε 1.0 και το e σε 0.5 και 0.75. Όπως μπορούμε να δούμε, για $e=0.75$ έχουμε πολύ καλά αποτελέσματα για c μεταξύ 0.25 και 1.0, όταν το a είναι ίσο με 0.4. Καλά αποτελέσματα έχουμε επίσης όταν $e=0.5$ και το c κυμαίνεται μεταξύ 0.0 και 0.5 για a μεταξύ 0.2 και 0.8. Στο σχήμα 5.19 παρουσιάζονται τα ίδια αποτελέσματα όμως για $e=0.25$. Παρατηρούμε πως το PDR είναι πιο χαμηλό παρά για e μεταξύ 0.5 και 0.75 και τα πιο καλά αποτελέσματα είναι για c ίσο με 0.0 και 0.25. Πρέπει να αναφερθεί πως όταν το e ισούται με 0.75 μόνο για a ίσο με 0.4-0.5 έχουμε καλά αποτελέσματα, για όλες τις άλλες τιμές παρουσιάζεται τεράστια πτώση στο PDR γεγονός που το καθιστά κάπως αναξιόπιστο.



Σχήμα 5.19: Αποτελέσματα PDR ($T=1.0$, $e=0.25$)

Επειδή γενικά είχαμε κοντινά αποτελέσματα για την περίοδο δειγματοληψίας, θα παρουσιάσω συνολικά τους καλύτερους συνδυασμούς παραμέτρων συναρτήσει του T στο σχήμα 5.20.

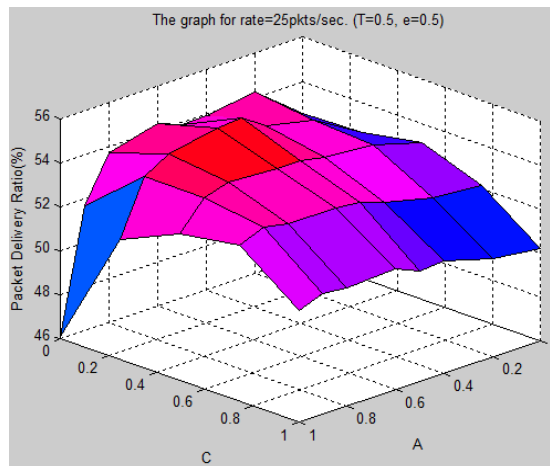


Σχήμα 5.20: Αποτελέσματα PDR ως προς το T

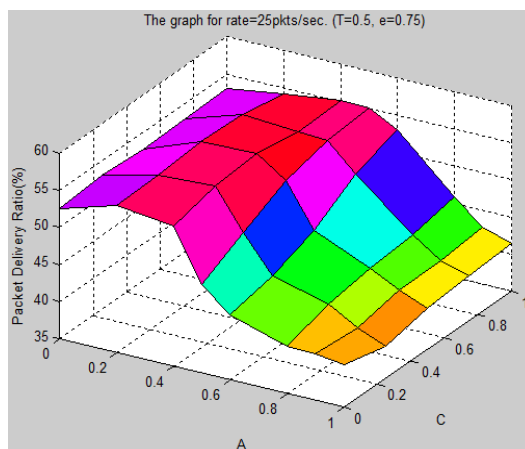
Όπως μπορούμε να δούμε τα αποτελέσματα είναι πάρα πολύ κοντινά. Και για τις 4 τιμές του T μπορούμε να ξεχωρίσουμε τις περιπτώσεις όπου το c ισούται με 0.0 και 0.25 και το e με 0.25 και 0.5 διότι είναι οι μόνοι συνδυασμοί παραμέτρων που δίνουν καλά αποτελέσματα και μια ομαλή γραφική απεικόνιση του PDR.

5.2.2 Σοβαρό Πρόβλημα Συμφόρησης (Data Rate=35pkts/sec)

Αρχικά μπορούμε να δούμε πως το PDR για data rate=35pkts/sec είναι πάρα πολύ χαμηλό, μόλις 55.90% γεγονός που υποδηλώνει πως υπάρχει πάρα πολύ μεγάλο πρόβλημα συμφόρησης. Μελετώντας τα αποτελέσματα των σεναρίων παρατήρησα πως για PDR μεταξύ 54-55%, η τιμή της παραμέτρου e ήταν πάντοτε το 0.5 - 0.75. Θα διατηρήσω λοιπόν σταθερή την παράμετρο e σε αυτές τις τιμές για να μπορέσω να δω την επίδραση των υπολοίπων παραμέτρων που δεν φαίνεται να έχουν και τόσο ξεκάθαρα αποτελέσματα. Επιπλέον, μελετώντας τα αποτελέσματα παρατήρησα πως για $T=0.5$ είχαμε καλύτερα αποτελέσματα σε σχέση με τις άλλες τιμές.

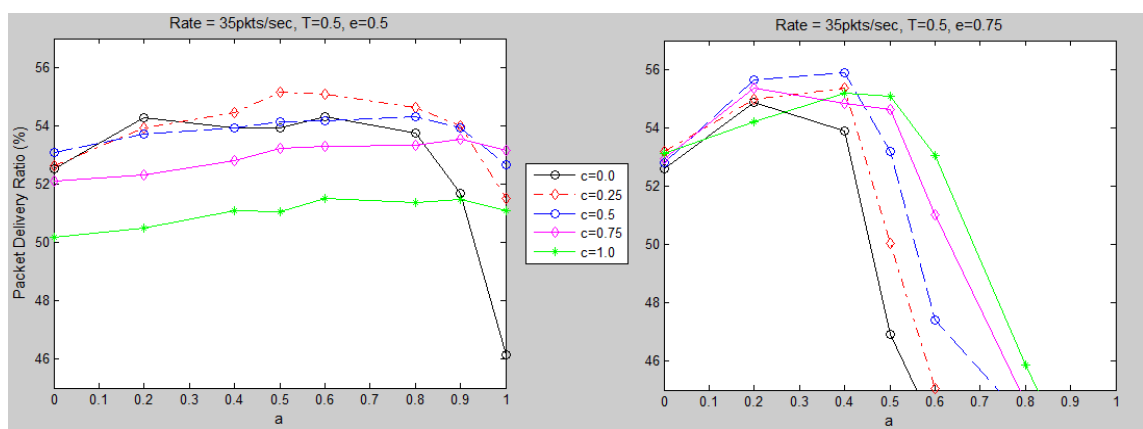


Σχήμα 5.21: Αποτελέσματα PDR ($e=0.5$)



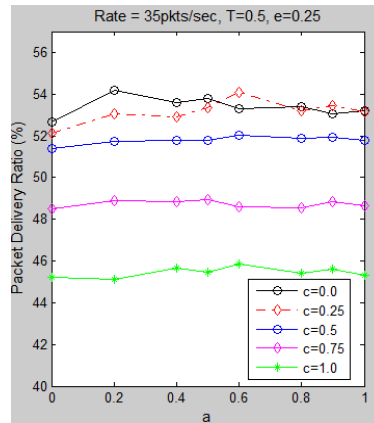
Σχήμα 5.22: Αποτελέσματα PDR ($e=0.75$)

Στα σχήματα 5.21 και 5.22 της προηγούμενης σελίδας παρουσιάζεται το PDR όταν $T=0.5$ και $e=0.5$ και $e=0.75$. Όπως μπορούμε να δούμε, όταν το e ισούται με 0.5 είχαμε πιο ομαλά αποτελέσματα, δηλαδή σχεδόν για όλες τις τιμές των a και c είχαμε ψηλό PDR. Από την άλλη, όταν το e ισούται με 0.75 βλέπουμε πως όταν το a είναι μεγαλύτερο από το 0.5 παρατηρείται μια τεράστια μείωση στο packet delivery ratio. Εκ' πρώτης όψεως, φαίνεται πως οι ιδανικότερες τιμές για το c είναι μεταξύ 0.25 και 0.5. Ακολουθώντας, θα παρουσιάσω εκτενέστερα τα αποτελέσματα για μπορέσουν να φανούν οι ιδανικές τιμές των παραμέτρων a και c .



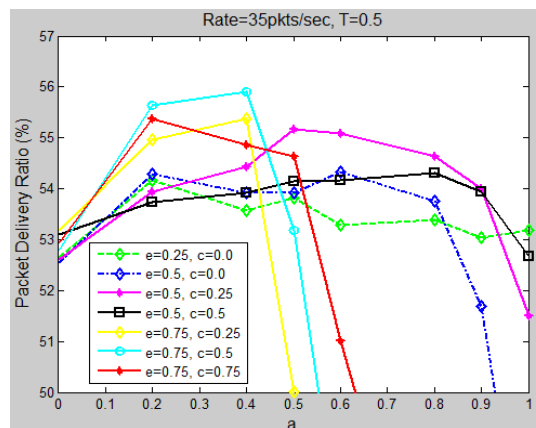
Σχήμα 5.23: Αποτελέσματα PDR για $T=0.5$ και e ίσο με 0.5 και 0.75

Στο σχήμα 5.23 παρουσιάζονται αναλυτικά τα αποτελέσματα για το packet delivery ratio όταν κρατάμε σταθερό το T σε 0.5 και το e σε 0.5 (αριστερά) και 0.75 (δεξιά). Όπως μπορούμε να δούμε, για $e=0.5$ είχαμε καλά αποτελέσματα όταν το c ισούται με 0.25 και η παράμετρος a κυμαίνεται μεταξύ 0.4 και 0.6 αυστηρά μιλώντας. Επίσης, πάρα πολύ καλά αποτελέσματα είχαμε όταν το e είναι ίσο με 0.75, το a μεταξύ 0.2 και 0.4, και το c μεταξύ 0.25 και 0.75.



Σχήμα 5.24: Αποτελέσματα PDR ($T=0.5$, $e=0.25$)

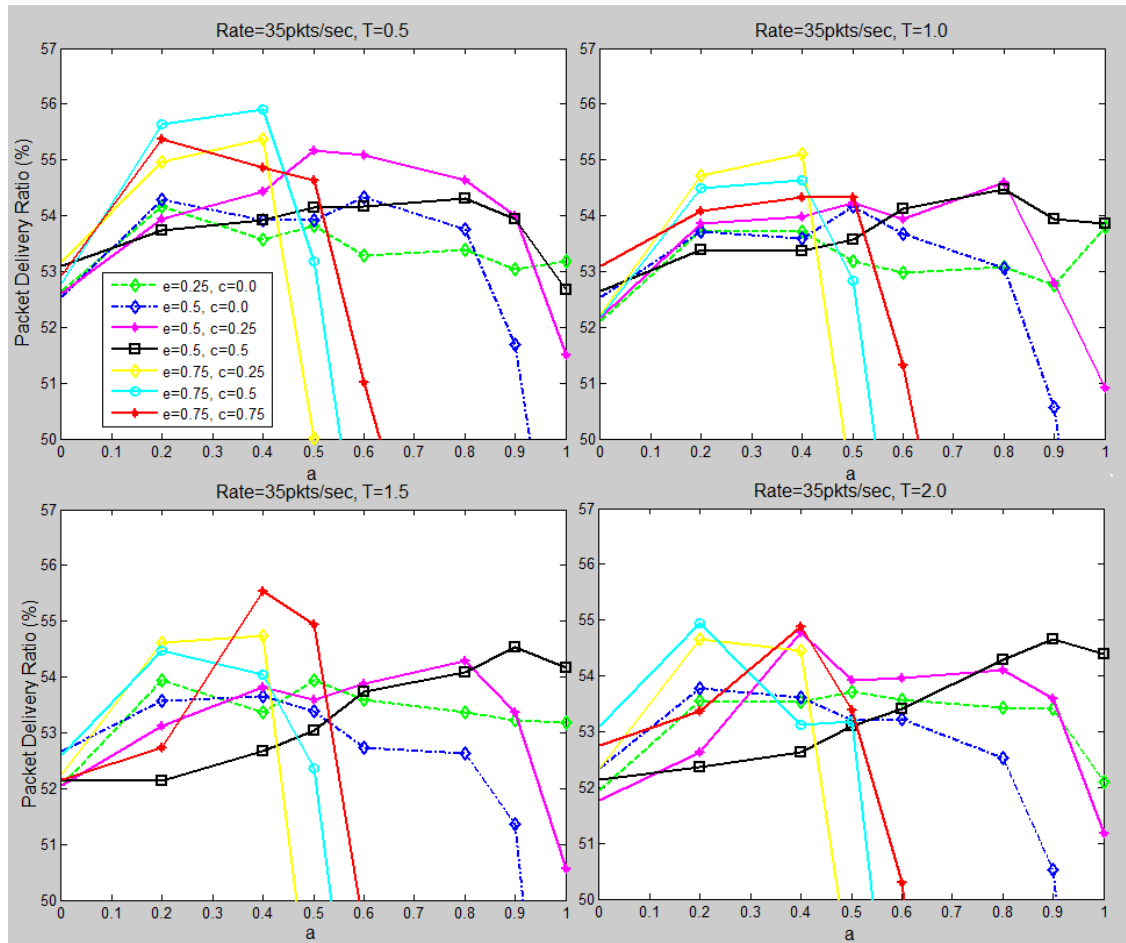
Στο σχήμα 5.24 φαίνονται τα αποτελέσματα του PDR για $T=0.5$ και $e=0.25$. Παρατηρούμε πως έχουμε χαμηλότερα αποτελέσματα σε σχέση με την προηγούμενη γραφική παράσταση όπου το e ισούται με 0.5 και 0.75. Παρόλα αυτά βλέπουμε καλά αποτελέσματα όταν το c ισούται με 0.0 και 0.25.



Σχήμα 5.25: Συνολικά αποτελέσματα PDR ($T=0.5$)

Στο σχήμα 5.25 απεικονίζονται τα καλύτερα αποτελέσματα συνολικά. Όπως μπορούμε να δούμε η μόνη περίπτωση όπου είχαμε ομαλή καμπύλη στο PDR για όλες τις τιμές της παραμέτρου a και αρκετά ψηλή ήταν για $e=0.5$ και $c=0.25$. Για αυτές τις τιμές η καλύτερες τιμές της παραμέτρου a είναι το 0.5 και το 0.6 και δίνουν PDR λίγο μεγαλύτερο ίσο με 55%.

Καλύτερος συνδυασμός παραμέτρων: $T=0.5$, $a=0.5$, $e=0.5$, $c=0.25$

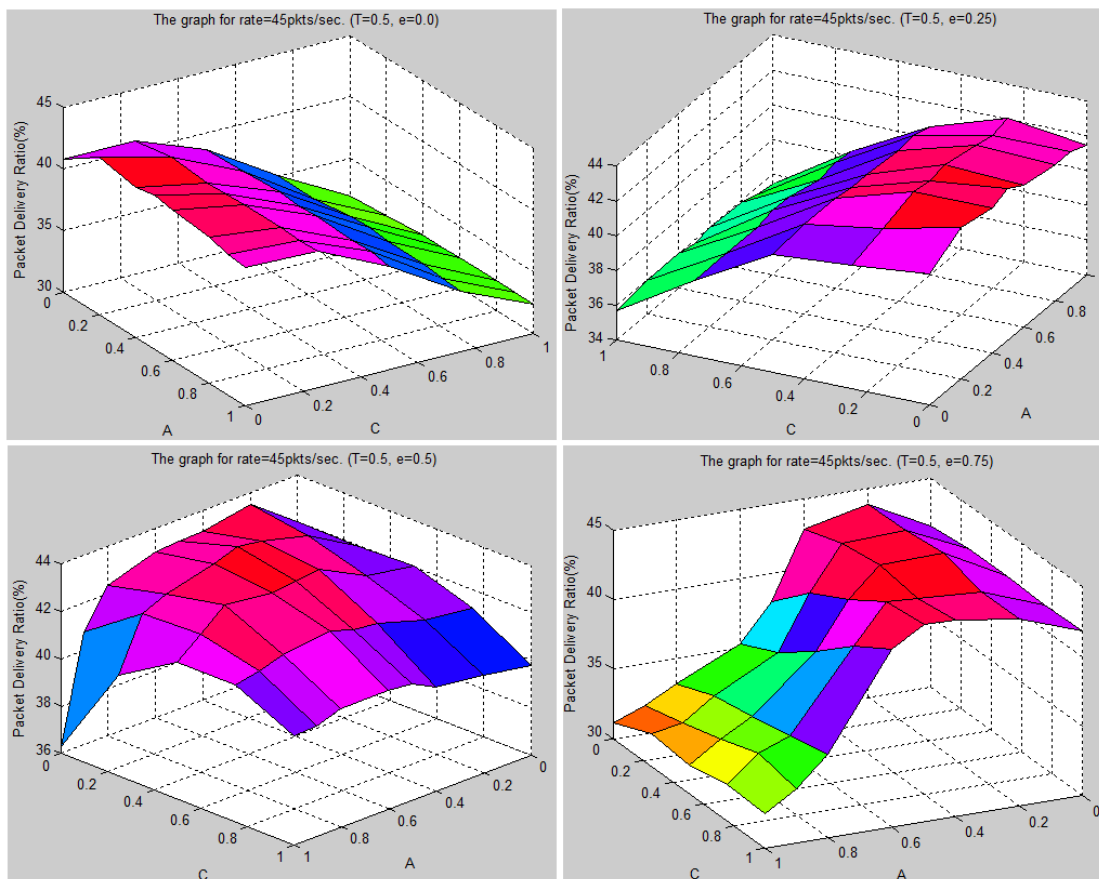


Σχήμα 5.26: Συνολικά αποτελέσματα PDR ως προς το T

Στο σχήμα 5.26 παρουσιάζεται μια σύγκριση των καλύτερων συνδυασμών παραμέτρων για τις τέσσερις τιμές του sampling period. Όπως μπορούμε να διακρίνουμε για τον καλύτερο συνδυασμό παραμέτρων, τον οποίο αναφέρω στην προηγούμενη σελίδα, παίρνουμε το ψηλότερο PDR όταν η περίοδος δειγματοληψίας ισούται με 0.5. Οι διαφορές βέβαια είναι μικρές, της τάξεως του 1% από τις άλλες περιπτώσεις.

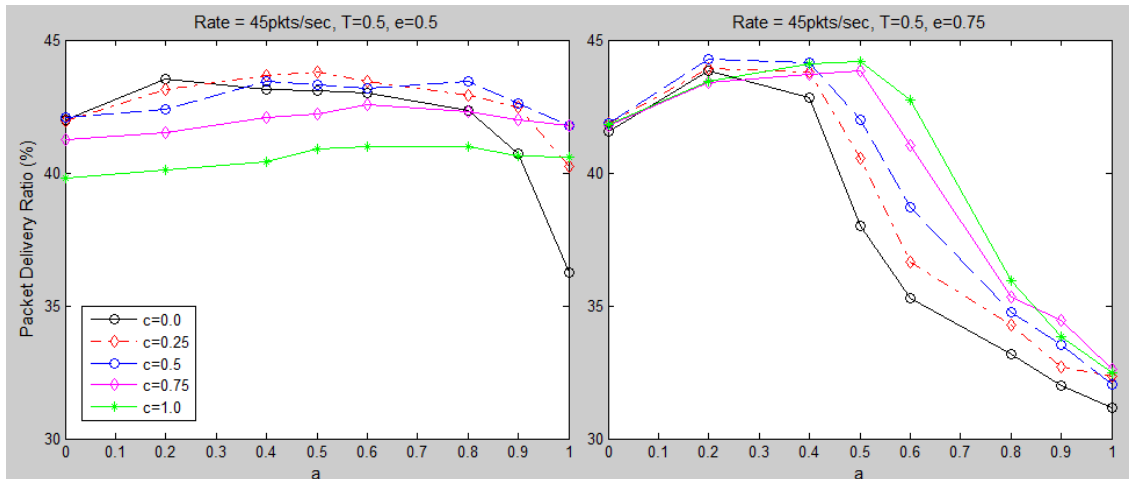
5.2.3 Πολύ Σοβαρό πρόβλημα συμφόρησης (Data Rate=45pkts/sec)

Αρχικά πρέπει να αναφέρω πως τα αποτελέσματα για data rate ίσο με 45pkts/sec ήταν πάρα πολύ χαμηλά. Υπάρχει σοβαρότατο πρόβλημα συμφόρησης στο δίκτυο. Το καλύτερο ποσοστό επιτυχημένης αποστολής πακέτων ανέρχεται μόλις στο 44.26%. Μελετώντας λοιπόν τα αποτελέσματα παρατήρησα πως η περίοδος δειγματοληψίας έδινε καλύτερα αποτελέσματα όταν ήταν ίση με 0.5. Αναμενόμενο αποτέλεσμα δεδομένου πως υπήρχε τόσο σοβαρό πρόβλημα συμφόρησης στο δίκτυο. Οι κόμβοι χρειάζεται να λαμβάνουν συχνά τα control messages για να ενημερώνονται για την κατάσταση των γειτονικών τους κόμβων και να μπορούν να υπολογίζουν με καλύτερο τρόπο που θα προωθήσουν τα πακέτα τους προκειμένου να αποφευχθούν οι προβληματικές περιοχές.



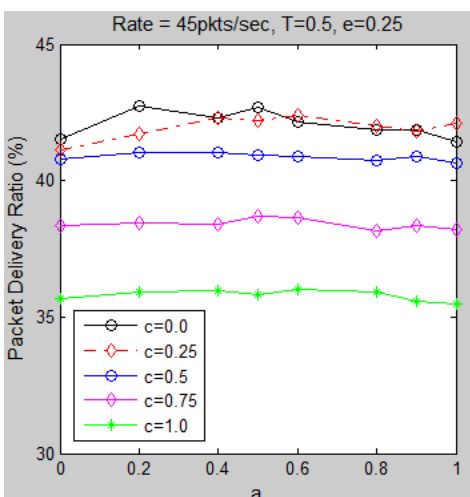
Σχήμα 5.27: Αποτελέσματα PDR (T=0.5)

Στο σχήμα 5.27 παρουσιάζεται το packet delivery ratio όταν $T=0.5$ και το e παίρνει τις τιμές 0.0, 0.25, 0.5 και 0.75. Όπως μπορούμε να δούμε τα καλύτερα αποτελέσματα είναι για e ίσο με 0.5 και 0.75, τα οποία και θα αναλύσω εκτενέστερα στην συνέχεια.



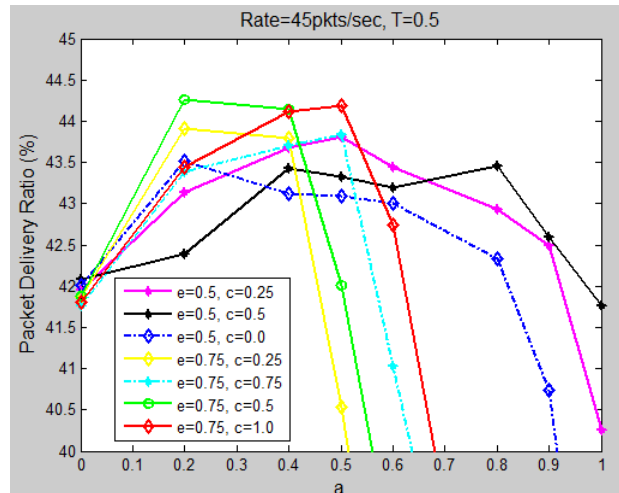
Σχήμα 5.28: Αποτελέσματα PDR $T=0.5$ και e ίσο με 0.5 και 0.75

Στο σχήμα 5.28 παρουσιάζονται τα αποτελέσματα του PDR για $T=0.5$ και e ίσο με 0.5 (δεξιά) και 0.75 (αριστερή). Όπως μπορούμε να δούμε, για $e=0.5$ σχετικά καλά αποτελέσματα είχαμε για c μεταξύ 0.0 και 0.5 και a μεταξύ 0.2 και 0.8. Ενώ για e ίσο με 0.75 είχαμε καλά αποτελέσματα μόνο για a μεταξύ 0.2 και 0.4 αυστηρά μιλώντας, παρόλο που για $c=0.25$ και $c=0.5$ παρατηρήθηκε μείωση για $a>0.4$.



Σχήμα 5.29: Αποτελέσματα PDR $T=0.5$, $e=0.25$

Στο σχήμα 5.29 παρουσιάζονται τα ίδια αποτελέσματα αλλά για e ίσο με 0.25 αυτή τη φορά. Παρατηρούμε πως δεν είχαμε και τόσο καλά αποτελέσματα. Η μόνη περίπτωση που αξίζει να ασχοληθούμε είναι για c ίσο με 0.0 και 0.25.

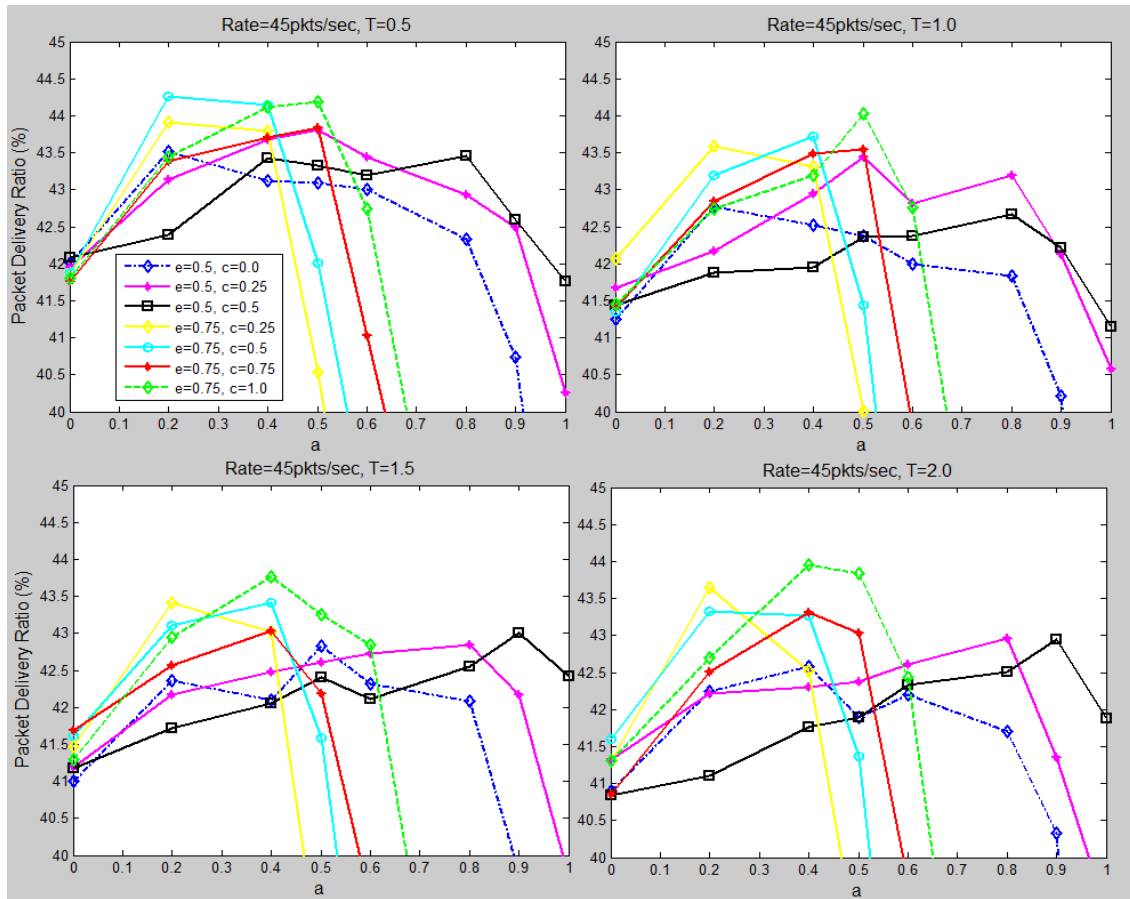


Σχήμα 5.30: Αποτελέσματα PDR (T=0.5)

Στο σχήμα 5.30 συνοψίζονται τα ψηλότερα αποτελέσματα. Όπως μπορούμε να διακρίνουμε, η πιο ομοιόμορφη καμπύλη δημιουργείται όταν $e=0.5$ και $c=0.25$ με βέλτιστες τιμές όταν η παράμετρος a είναι μεταξύ 0.4 και 0.6. Μπορούμε λοιπόν να επιλέξουμε ως τον καλύτερο συνδυασμό παραμέτρων το: **$T=0.5$, $a=0.5$, $c=0.5$, $c=0.25$** , συνδυασμός ο οποίος ήταν ο καλύτερος και προηγουμένως για data rate ίσο με 35pkts/sec.

Καλύτερος συνδυασμός παραμέτρων για data rate ίσο με 45pkts/sec:

$T=0.5$, $a=0.5$, $c=0.5$, $c=0.25$



Σχήμα 5.31: Αποτελέσματα PDR ως προς το T

Παραπάνω, στο σχήμα 5.31 παρουσιάζονται τα καλύτερα αποτελέσματα για τις διάφορες τιμές της περιόδου δειγματοληψίας. Παρατηρείται μια εμφανής μείωση στο PDR καθώς αυξάνεται το T. Λόγω της φύσης του σεναρίου, όπου είδαμε πως υπήρχε τεράστιο πρόβλημα συμφόρησης, οι κόμβοι χρειάζεται να λαμβάνουν γρηγορότερα τα μηνύματα ελέγχου για να ενημερώνονται για την κατάσταση των άλλων κόμβων και με αυτό τον τρόπο να μπορούν να επιλέγουν τον καταλληλότερο κόμβο αποφεύγοντας έτσι τις προβληματικές περιοχές.

5.2.4 Παρατηρήσεις

Περίοδος δειγματοληψίας (T): Είδαμε πως η καλύτερη τιμή για την παράμετρο T είναι το 0.5 και ήταν κάτι το οποίο αναμέναμε αφού θέλουμε οι κόμβοι να λαμβάνουν σύντομα νέες πληροφορίες από τους γείτονες τους και να τις αξιοποιούν καταλλήλως προκειμένου να αποφύγουν τις προβληματικές περιοχές.

$$T=0.5$$

Παράμετρος a : Παρατηρήσαμε πως τα καλύτερα αποτελέσματα για αυτή την παράμετρο ήταν μεταξύ 0.2 και 0.6 όμως αυστηρά μιλώντας, η καλύτερη τιμή ήταν το 0.5 για όλα τα data rates αφού οι διαφορές που είχαν ήταν πάρα πολύ μικρές. Δηλαδή, η κατάσταση στις ουρές των κόμβων, με την κατάσταση στο ασύρματο μέσο παίζουν εξίσου ρόλο για τον εντοπισμό της συμφόρησης.

$$A=0.5$$

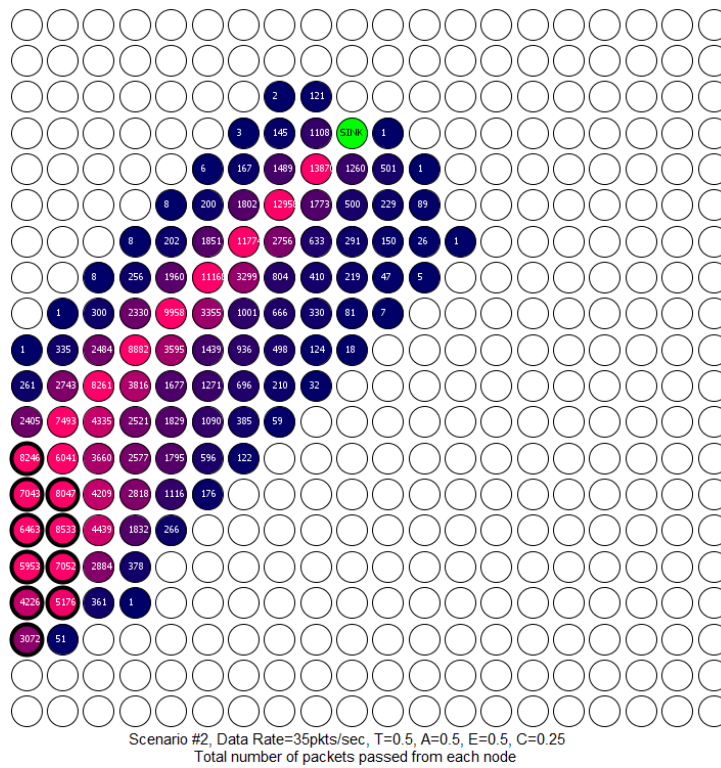
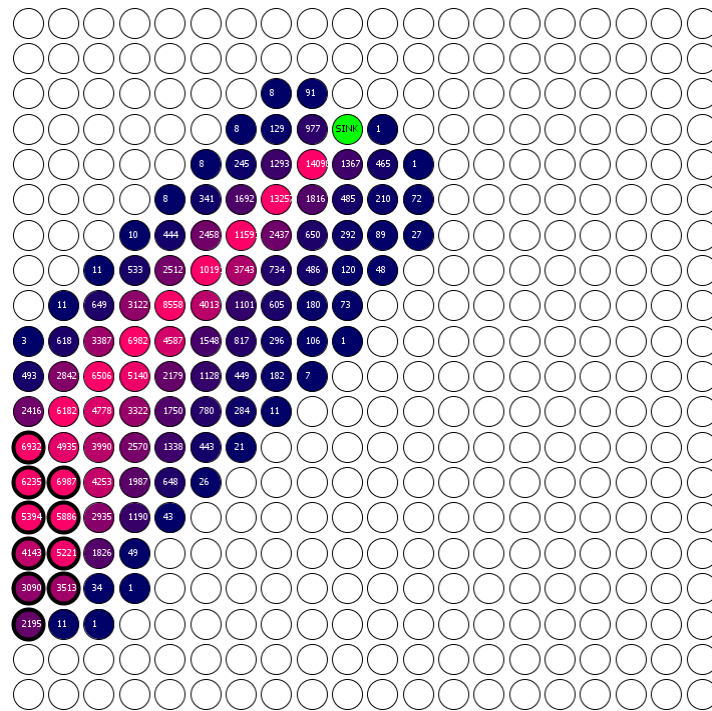
Παράμετρος e : Αυτή η παράμετρος ήταν πιθανότατα η πιο εύκολη στο να καθοριστεί. Για όλα τα data rates παρατηρήσαμε πως η καλύτερη τιμή ήταν το 0.5 και 0.75. Αναμενόμενο νοούμενου πως οι κόμβοι θέλουμε να λαμβάνουν υπόψη τους και τους γείτονες με ίσο hop count προς το sink. Το μονοπάτι μεταξύ του sink και της ομάδας δημιουργίας πακέτων ήταν πολύ στενό κι έτσι επιβάλλεται να λαμβάνονται σε μεγάλο βαθμό υπόψη και κόμβοι με ίσο hop count για να μπορέσει να ανοίξει το κοπάδι. Επέλεξα την τιμή 0.5 ως την καλύτερη διότι δίνει το ίδιο καλά αποτελέσματα με το 0.75 και οι διαφορές είναι απειροελάχιστες και για όλα τα άλλα σενάρια υπερίσχυσε. Επιπλέον, η τιμή αυτή έδινε πολύ ομαλά αποτελέσματα και ψηλό PDR σχεδόν για όλες τις τιμές της παραμέτρου a κάτι το οποίο δεν ίσχυε για $e=0.75$. Έτσι μπορούμε να την διατηρήσουμε σταθερή σε 0.5.

$$E=0.5$$

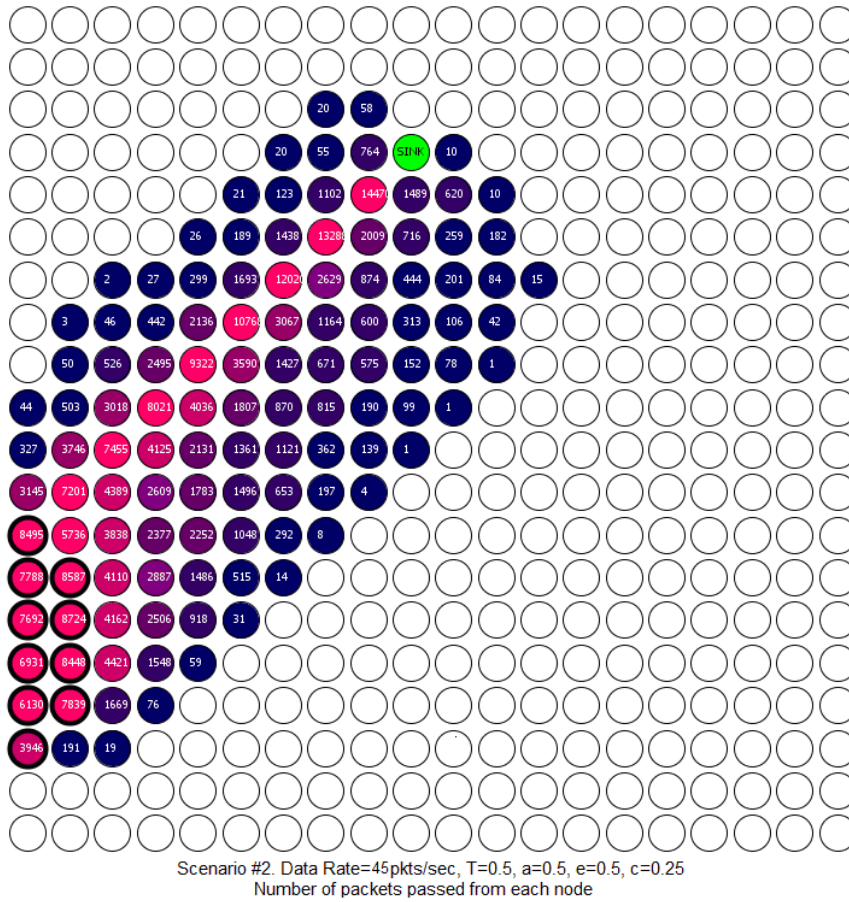
Παράμετρος c : Η παράμετρος c στην καλύτερη περίπτωση ήταν ίση με 0.25 για όλα τα data rates. Όπως και σε όλες τις άλλες περιπτώσεις φάνηκε πως έχουμε καλά αποτελέσματα αν χρησιμοποιείται λίγη τυχαιότητα.

$$C=0.25$$

5.2.5 Γραφική απεικόνιση κίνησης πακέτων



Σχήμα 5.32: Συνολικός αριθμός πακέτων που πέρασαν από τον κάθε κόμβο

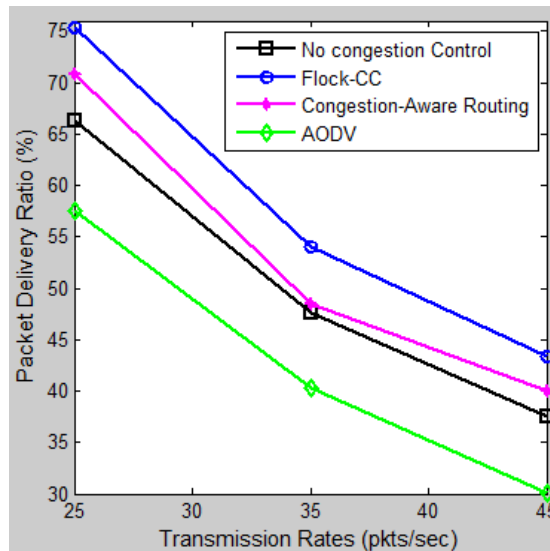


Σχήμα 5.33: Συνολικός αριθμός πακέτων που πέρασαν από τον κάθε κόμβο

Στα σχήματα 5.32 και 5.33, παρατηρούμε τον συνολικό αριθμό πακέτων που έχουν περάσει από τον κάθε κόμβο για τιμές παραμέτρων ίσες με $T=0.5$, $A=0.5$, $E=0.5$, $C=0.25$ για τις τρεις τιμές του data rate. Παρατηρούμε πως η συμφόρηση προκαλείται στην κεντρική διαγώνιο που ενώνει την ομάδα των κόμβων δημιουργίας πακέτων με το sink και η περιοχή όπου δημιουργούνται τα πακέτα. Σε αυτό το σενάριο δεν προκαλείται συμφόρηση στο sink όπως μπορούμε να δούμε.

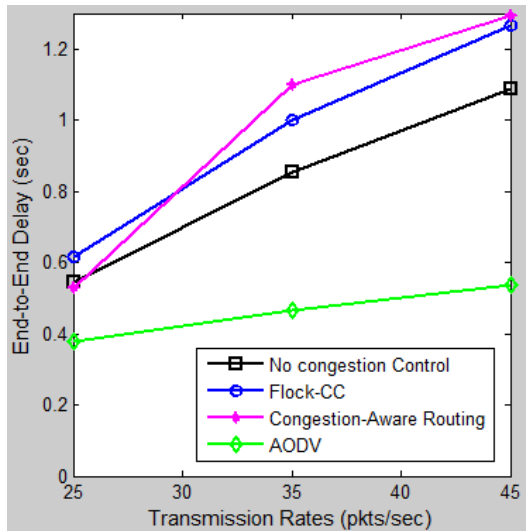
5.2.6 Σύγκριση αλγορίθμων

Χρησιμοποίησα τις καλύτερες τιμές για τις παραμέτρους μου ανά data rate οι οποίες ήταν: $T=0.5$, $A=0.5$, $E=0.5$, $C=0.25$.



Σχήμα 5.34: Αποτελέσματα PDR

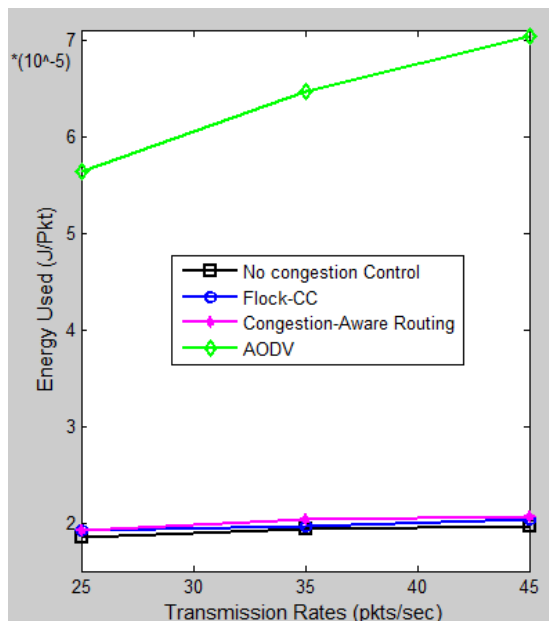
Στο σχήμα 5.34 απεικονίζεται το PDR για τους τέσσερις αλγορίθμους για όλες τις τιμές των data rates τις οποίες χρειάστηκε να μελετήσω. Όπως μπορούμε να δούμε, ο υπό μελέτη αλγόριθμος είχε τα καλύτερα αποτελέσματα από τους τέσσερις αλγορίθμους. Ο δεύτερος καλύτερος αλγόριθμος ήταν ο CAwR. Ακολούθως, τα χειρότερα αποτελέσματα είχαμε για τον αλγόριθμο που δεν χρησιμοποιούσε έλεγχο συμφόρησης. Τέλος, το χαμηλότερο PDR είχε ο AODV με διαφορά περίπου 15% από τον καλύτερο αλγόριθμο Flock-CC. Για όλα τα data rates παρατηρούμε πως ο αλγόριθμος μας έχει περίπου 5% μεγαλύτερο PDR από τον δεύτερο καλύτερο και 10% από τον τρίτο καλύτερο.



Σχήμα 5.35: Αποτελέσματα Καθυστερήσεων

Σε αυτή την γραφική παράσταση παρουσιάζονται οι καθυστερήσεις. Παρόλο που δεν είχαμε τόσο μεγάλες καθυστερήσεις για όλους τους αλγόριθμους φαίνεται πως ο AODV είχε τον μικρότερο χρόνο για όλα τα data rates ο οποίος ήταν 0.38sec, 0.47sec και 0.54sec για data rates ίσα με 25pkts/sec, 35pkts/sec και 45pkts/sec αντίστοιχα.

Ακολουθώντας, χαμηλότερες καθυστερήσεις είχε ο αλγόριθμος χωρίς έλεγχο συμφόρησης. Έπειτα, ο Flocking αλγόριθμος με χρόνους 0.62sec, 1sec και 1.27sec για κάθε ένα από τα data rates. Ο CAwR φαίνεται να έχει ελαφρώς χειρότερα αποτελέσματα από τον Flock-CC εκτός για data rate ίσο με 25pkts/sec όπου χρειάστηκε μόνο 0.53sec.



Σχήμα 5.36: Αποτελέσματα δαπανούμενης ενέργειας ανά πακέτο

Στο σχήμα 5.36 απεικονίζεται η συνολική ενέργεια που δαπανήθηκε ανά πακέτο. Παρατηρούμε πως οι αλγόριθμοι Flock-CC, CAwR και No Congestion Control δίνουν τα ίδια αποτελέσματα για όλα τα data rates τα οποία ισούνται με $\sim 1.9 \cdot 10^{-5}$ J/pkt. Όσον αφορά τον αλγόριθμο AODV μπορούμε και πάλι να δούμε πως δίνει τα χειρότερα αποτελέσματα και υπάρχει

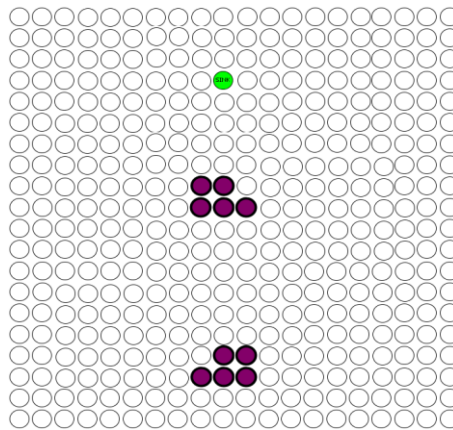
μεγάλη σπατάλη ενέργειας η οποία ισούται με $5.65 \cdot 10^{-5}$, $6.48 \cdot 10^{-5}$ και $7.05 \cdot 10^{-5}$ Joules per packet για data rates ίσα με 25, 35 και 45 pkts/sec αντίστοιχα.

5.2.7 Σύνοψη

Αυτό το σενάριο παρουσίασε αρκετά προβλήματα συμφόρησης τα οποία οφείλονται στην κοντινή θέση των κόμβων στο sink. Επίσης, λόγω της θέσης που βρίσκονται οι κόμβοι που δημιουργούν τα πακέτα, δεν υπάρχουν και πάρα πολλά εναλλακτικά μονοπάτια με τον ελάχιστο αριθμό hops για να σταλούν τα δεδομένα, σε αντίθεση με όλα τα άλλα σενάρια, κι έτσι τα πακέτα στέλλονται από την κεντρική διαγώνιο, προκαλώντας συμφόρηση σε εκείνη την περιοχή. Ίσως αν υπήρχε κάποια σειρά κόμβων πίσω από την ομάδα των 10 κόμβων που δημιουργούν τα πακέτα, τα αποτελέσματα να μην ήταν τόσο χαμηλά. Είδαμε πως το κοπάδι προσπάθησε να ανοίξει και να καλύψει μεγάλη περιοχή του δικτύου για να στέλλει τα πακέτα από πολλαπλά μονοπάτια. Παρόλα αυτά δεν ήταν αρκετό. Τα προβλήματα συμφόρησης προκαλούνταν κοντά στην περιοχή των κόμβων που έστελλαν τα δεδομένα κι όχι κοντά στο sink. Συγκριτικά με τους άλλους αλγορίθμους είδαμε πως έδινε το ψηλότερο PDR και είχε χαμηλές καθυστερήσεις και ποσό δαπανώμενης ενέργειας ανά πακέτο.

5.3 Δύο ομάδες δημιουργίας πακέτων απέναντι από το sink

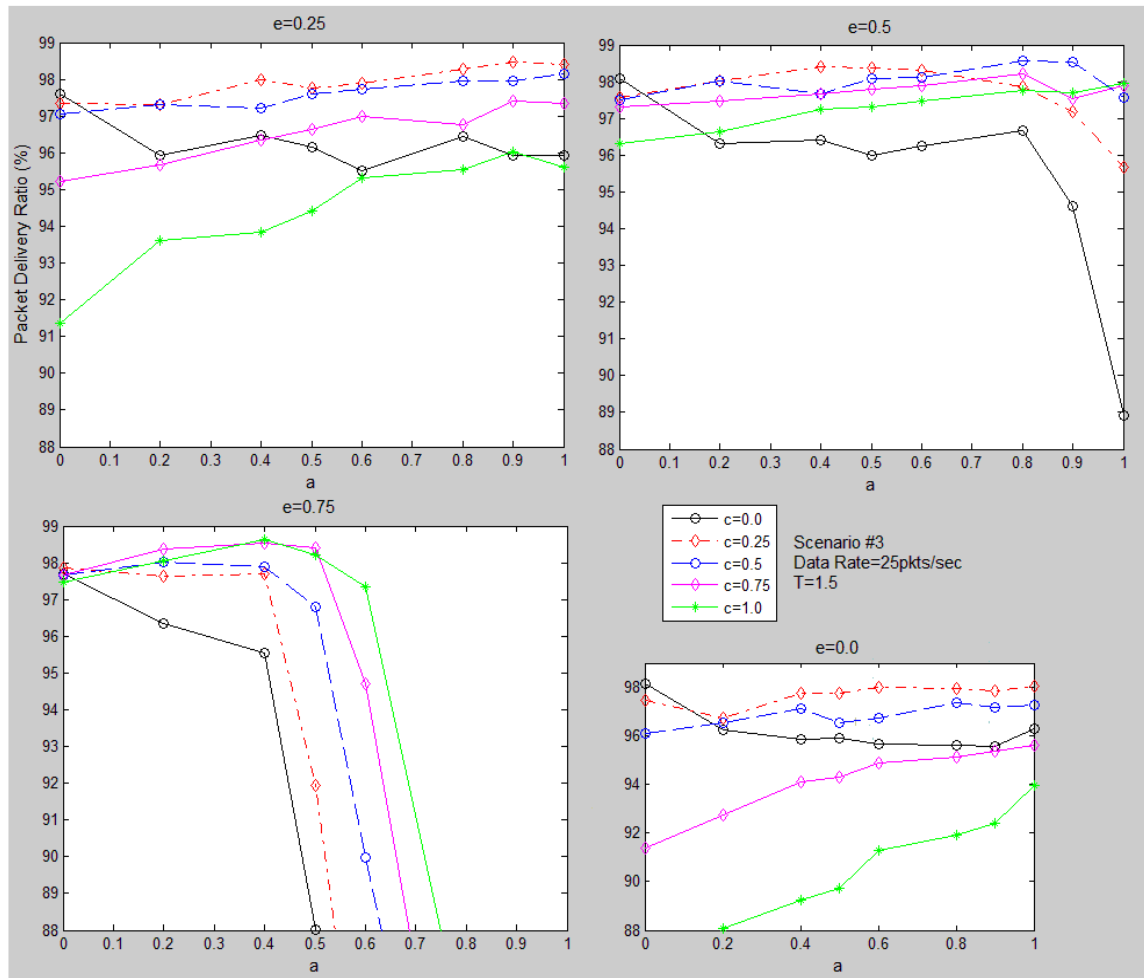
Σε αυτό το σενάριο έχουμε 5 κόμβους στην μέση του τετραγωνικού πλέγματος και ακόμη 5 κόμβους οι οποίοι βρίσκονται στην ίδια ευθεία με την πρώτη ομάδα όμως σε μεγαλύτερη απόσταση από το sink (βλέπε σχήμα 5.37). Η πρώτη ομάδα κόμβων, που βρίσκεται σε πιο κοντινή απόσταση από το sink, ξεκινάει την δημιουργία πακέτων στα πρώτα 10sec ενώ η δεύτερη ομάδα στα 52sec.



Σχήμα 5.37: Παρουσίαση κόμβων δημιουργίας πακέτων

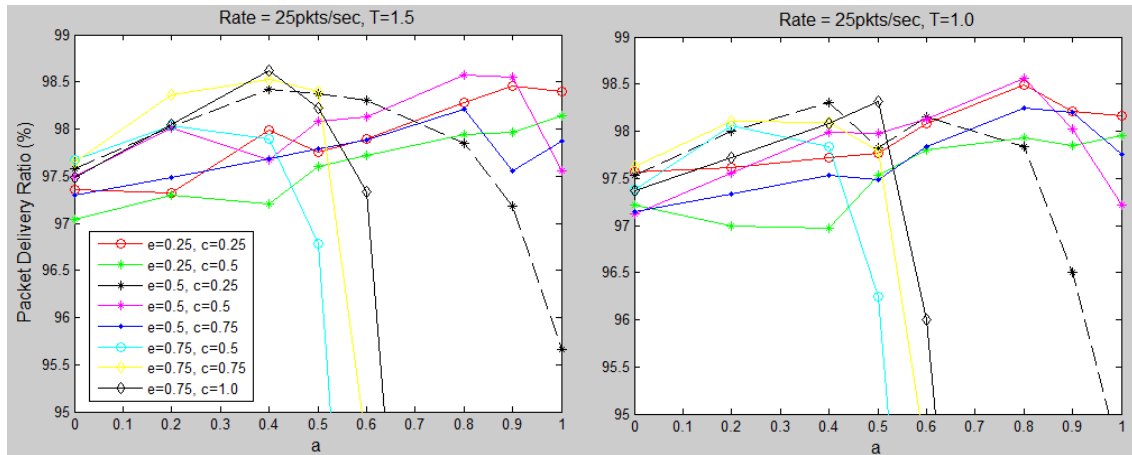
Αρχικά, μελετώντας τα αποτελέσματα του σεναρίου παρατήρησα πως καθώς αυξάνεται το data rate, και συνεπώς προκαλείται συμφόρηση στο δίκτυο, είχαμε ψηλότερο ποσοστό επιτυχημένης αποστολής πακέτων καθώς μειωνόταν η τιμή του sampling period (T). Δηλαδή, για data rate=25pkts/sec το ψηλότερο ποσοστό επιτυχημένης αποστολής (98.62%) είναι όταν το T ισούται με 1.5. Για data rate=35pkts/sec το ψηλότερο ποσοστό επιτυχημένης αποστολής είναι όταν το T ισούται με 1.0 και κυμαίνεται κάπου στα 87.85%. Τέλος, για data rate=45pkts/sec τα καλύτερα αποτελέσματα είναι για T=0.5 και το ποσοστό επιτυχημένης αποστολής ήταν ίσο με 77.21%. Καθώς αυξάνονται τα δεδομένα αυξάνεται και η συμφόρηση στο δίκτυο και συνεπώς χρειάζεται να λαμβάνονται πιο γρήγορα τα control packets για να μπορούν οι κόμβοι να αντιληφθούν έγκαιρα τις περιοχές όπου υπάρχει συμφόρηση και να βρουν εναλλακτικό μονοπάτι.

5.3.1 Ελαφριά Συμφόρηση (Data Rate=25pkts/sec)



Σχήμα 5.38: Αποτελέσματα PDR (T=1.5)

Το σχήμα 5.38 παρουσιάζει τα αποτελέσματα του PDR για $e=0.0$, $e=0.25$, $e=0.5$ και $e=0.75$. Αρχικά μπορούμε να δούμε πως όταν το e ισούται με 0.25 και 0.5 τα καλύτερα αποτελέσματα ήταν για c ίσο με 0.25 και 0.5 κι έδιναν 97-98% ποσοστό επιτυχημένης αποστολής πακέτων. Επιπλέον, όσον αφορά το $e=0$, οι ίδιες τιμές της παραμέτρου c δίνουν χαμηλότερα αποτελέσματα. Μετά παρατηρούμε πως για $e=0.75$ όταν το a είναι μεγαλύτερο από 0.5 υπάρχει δραματική μείωση στο packet delivery ratio. Όμως για a μικρότερο του 0.5 είχαμε αρκετά καλά αποτελέσματα για όλες τις τιμές του c εκτός από το 0.0.



Σχήμα 5.39: Αποτελέσματα PDR

Στο σχήμα 5.39 παρουσιάζονται συνολικά τα καλύτερα αποτελέσματα για data rate=25pkts/sec όταν η περίοδος δειγματοληψίας παραμένει σταθερή σε 1.5 και 1.0 αντίστοιχα. Όπως βλέπουμε, τα αποτελέσματα για όλες τις περιπτώσεις είναι πάρα πολύ κοντινά, σε σημείο που δεν μπορούμε να επιλέξουμε με μεγάλη ευκολία μια τιμή για την κάθε παράμετρο. Το σίγουρο είναι πως οι τιμές των παραμέτρων που συνδυάζονται για να δώσουν το μεγαλύτερο PDR είναι οι εξής:

$$e \in \{0.5, 0.75\}$$

$$c \in \{0.25, 0.5\}$$

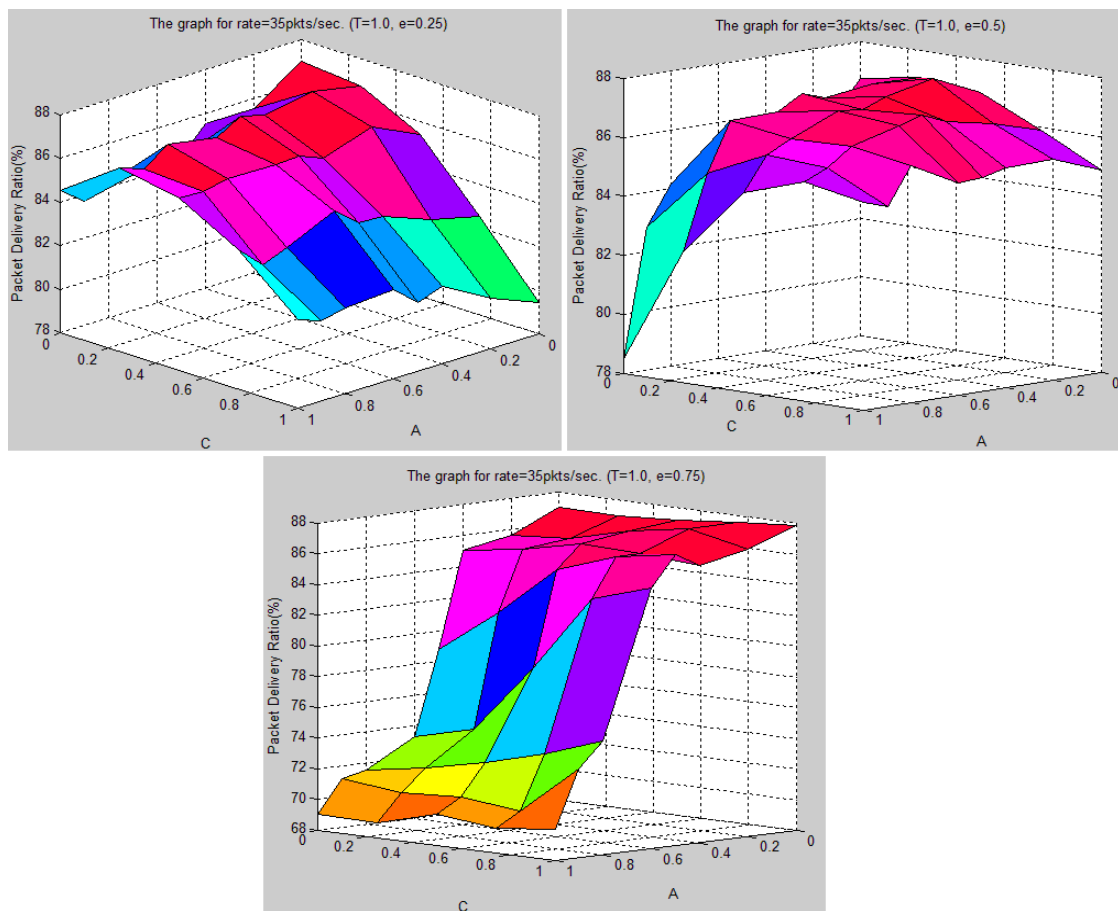
$$a \in \{0.2, 0.4, 0.5, 0.6, 0.8\}$$

Παρατηρούμε πως είχαμε αρκετά καλά αποτελέσματα για $e=0.75$ και $c=1.0$ για $T=1.5$ και $a=0.4$. Το γεγονός πως για όλες τις άλλες τιμές δεν είχαμε καλά αποτελέσματα μας οδηγεί στο συμπέρασμα πως δεν μπορούμε να εμπιστευθούμε αυτές τις τιμές. Επίσης, από όλες τις τιμές που βλέπουμε και στις δύο γραφικές παραστάσεις, τα πιο ομαλά αποτελέσματα είναι για $e=0.5$, $c=0.25$ και $a=0.5$.

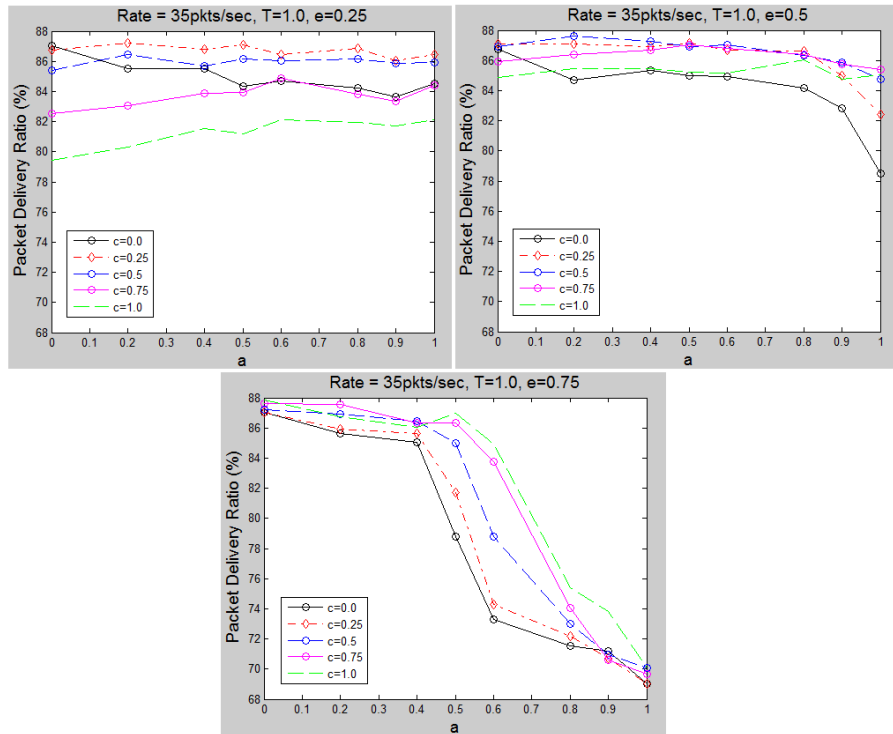
Το γεγονός πως δεν μπορούμε να καθορίσουμε με ακρίβεια ποια είναι η ιδανικότερη τιμή της κάθε παραμέτρου οφείλεται στο γεγονός πως το data rate είναι αρκετά μικρό ώστε να μην προκαλείται συμφόρηση στο δίκτυο. Γι' αυτό το λόγο δεν μπορεί να φανεί τόσο εύκολα η χρησιμότητα της κάθε παραμέτρου. Στην συνέχεια, θα μπορέσουμε να καθορίσουμε με μεγαλύτερη ακρίβεια την κάθε παράμετρο μέσω των data rates 35 και 45 packets per second.

5.3.2 Πρόβλημα Συμφόρησης (Data Rate=35pkts/sec)

Για data rate ίσο με 35pkts/sec αποφάσισα να κρατήσω σταθερή την περίοδο δειγματοληψίας σε 1.0, που ήταν και η καλύτερη τιμή, και το e σε 0.25, 0.5 και 0.75 και θα κοιτάξω πως επηρεάζονται οι άλλες παράμετροι κι έτσι σταδιακά θα μπορέσω να αποφανθώ ως προς το ποιες είναι οι ιδανικότερες τιμές. Ενδεικτικά, πιο κάτω στο σχήμα 5.40 παρουσιάζονται οι αντίστοιχες γραφικές παραστάσεις τις οποίες θα παρουσιάσω σε μεγαλύτερη λεπτομέρεια στην επόμενη σελίδα.



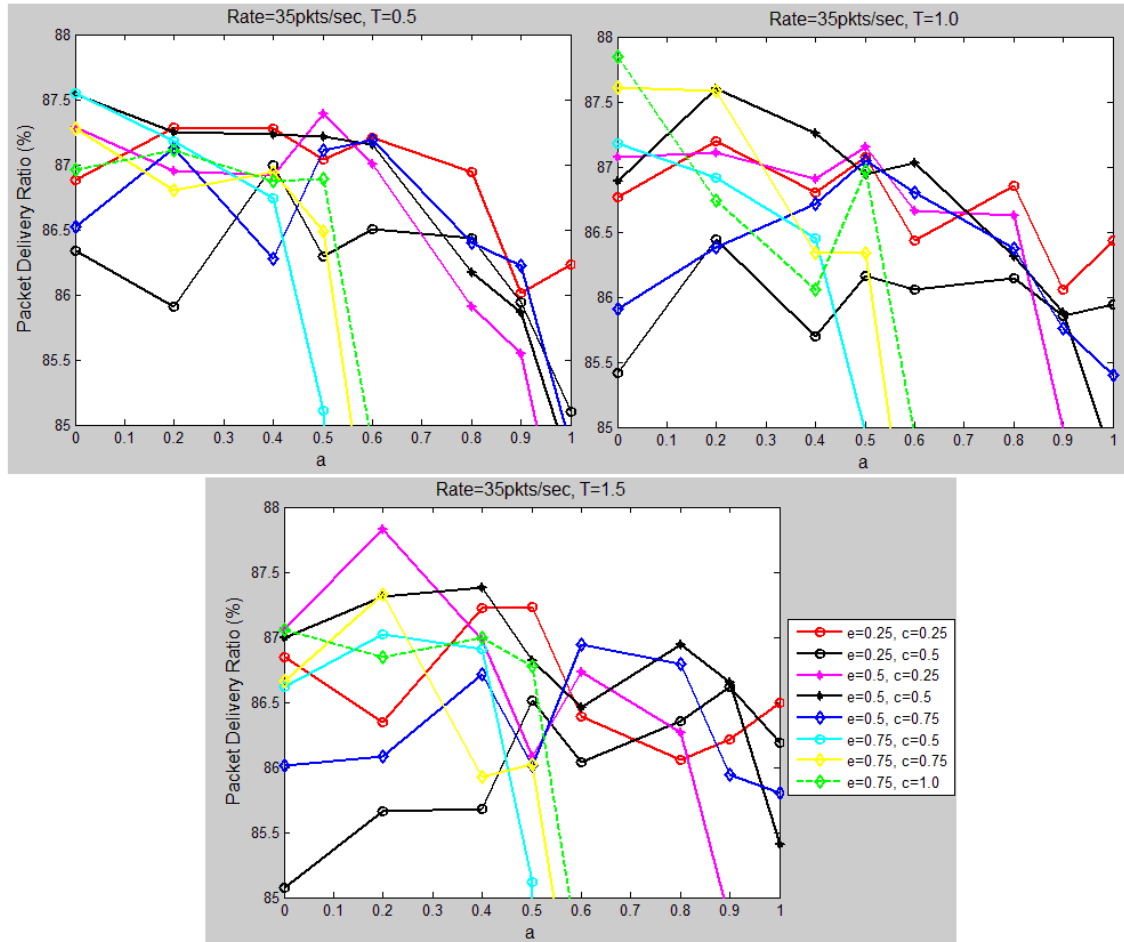
Σχήμα 5.40: Αποτελέσματα PDR (T=1.0)



Σχήμα 5.41: Αποτελέσματα PDR ($T=1.0$)

Στο σχήμα 5.41 παρουσιάζεται το PDR όταν το T διατηρείται σταθερό σε 1.0 και το e σε 0.25, 0.5 και 0.75 για κάθε μια από τις παραστάσεις. Παρατηρούμε πως για $e=0.75$ τα αποτελέσματα δεν είναι και τόσο καλά διότι αυξάνεται πάρα πολύ η πιθανότητα επιλογής κάποιου κόμβου που βρίσκεται σε ίση απόσταση προς το sink με τον αποστολέα κι αυτό οδηγεί τα πακέτα σε κύκλους ανάμεσα σε κόμβους που δεν έχουν συμφόρηση με αποτέλεσμα να αυξάνονται οι καθυστερήσεις και να χάνονται τα πακέτα.

Βλέπουμε πως όταν το e ισούται με 0.5 είχαμε πολύ καλά αποτελέσματα για όλες τις τιμές του c εκτός από το 1.0 και 0.0. Η παράμετρος c αναπαριστά την διαφοροποίηση που εισάγουμε για να υποδείξουμε την τυχαιότητα στον αλγόριθμο. Αν έχει μεγάλη τιμή τότε τα πακέτα μπορεί να καταλήξουν να στέλλονται τυχαία ενώ όταν ισούται με μηδέν ο αλγόριθμος παύει να έχει τυχαιότητα κάτι το οποίο και πάλι λειτουργεί αρνητικά. Όταν το e ισούται με 0.25 βλέπουμε πως δεν έχουμε καλά αποτελέσματα όταν το c είναι ίσο με 0.0, 0.75 και 1.0. Επιπλέον, για $e=0.5$ παρατηρείται μείωση στο packet delivery ratio όταν το a είναι μεγαλύτερο από το 0.6. Το ίδιο ισχύει για όλες τις μεγαλύτερες τιμές του e , σε όλα τα σενάρια.



Σχήμα 5.42: Αποτελέσματα PDR ως προς το T

Στο σχήμα 5.42 παρουσιάζονται τα καλύτερα αποτελέσματα συνολικά για data rate=35pkts/sec όταν η περίοδος δειγματοληψίας ισούται με 0.5, 1.0 και 1.5 αντίστοιχα. Παρατηρούμε πως η μεγαλύτερη τιμή του PDR ισούται με 87.65% και ήταν για **T=1.5**, **a=0.2**, **e=0.5**, **c=0.25** ή **T=1.0**, **a=0.2**, **e=0.5**, **c=0.5**.

Γενικά μπορούμε να δούμε πως τα καλύτερα αποτελέσματα είναι για:

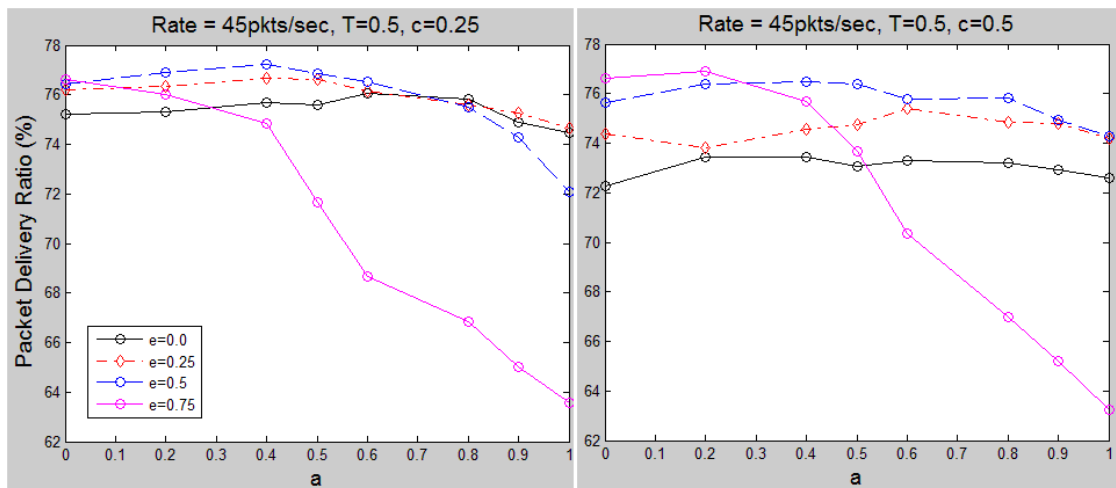
$$a \in \{0.2, 0.4, 0.5\}$$

$$e \in \{0.25, 0.5\}$$

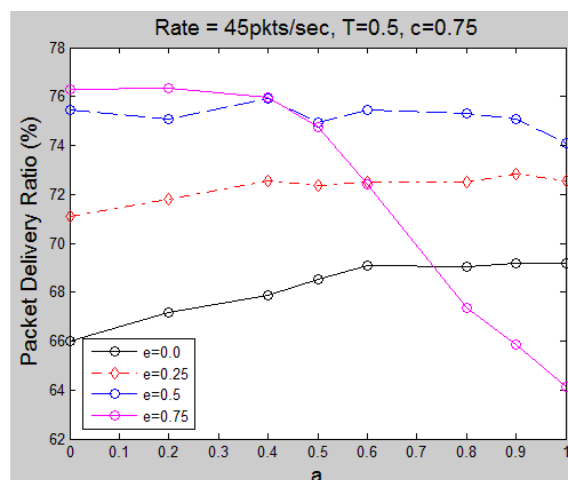
$$c \in \{0.25, 0.5\}$$

5.3.3 Σοβαρό πρόβλημα συμφόρησης (Data Rate=45pkts/sec)

Για data rate ίσο με 45pkts/sec είδαμε πως υπάρχει αρκετή συμφόρηση στο δίκτυο με αποτέλεσμα το μειωμένο PDR αφού στην καλύτερη περίπτωση ισούται με 77.21%. Τα καλύτερα αποτελέσματα έχουν το T ίσο με 0.5 αφού χρειάζεται να στέλλονται συχνά τα control messages για να ενημερώνουν τους κόμβους για τις αλλαγές στο δίκτυο, αν και για $T=1.0$ είχαμε και πολύ καλά αποτελέσματα. Όσον αφορά την παράμετρο c φαίνεται πως τα καλύτερα αποτελέσματα κυμαίνονται μεταξύ 0.25 και 0.5 ενώ για το e μεταξύ 0.25 και 0.75. Βάσει των όσων έχω ήδη αναφέρει θα διατηρήσω το sampling period σταθερό σε 0.5 και θα μελετήσω τις υπόλοιπες παραμέτρους.

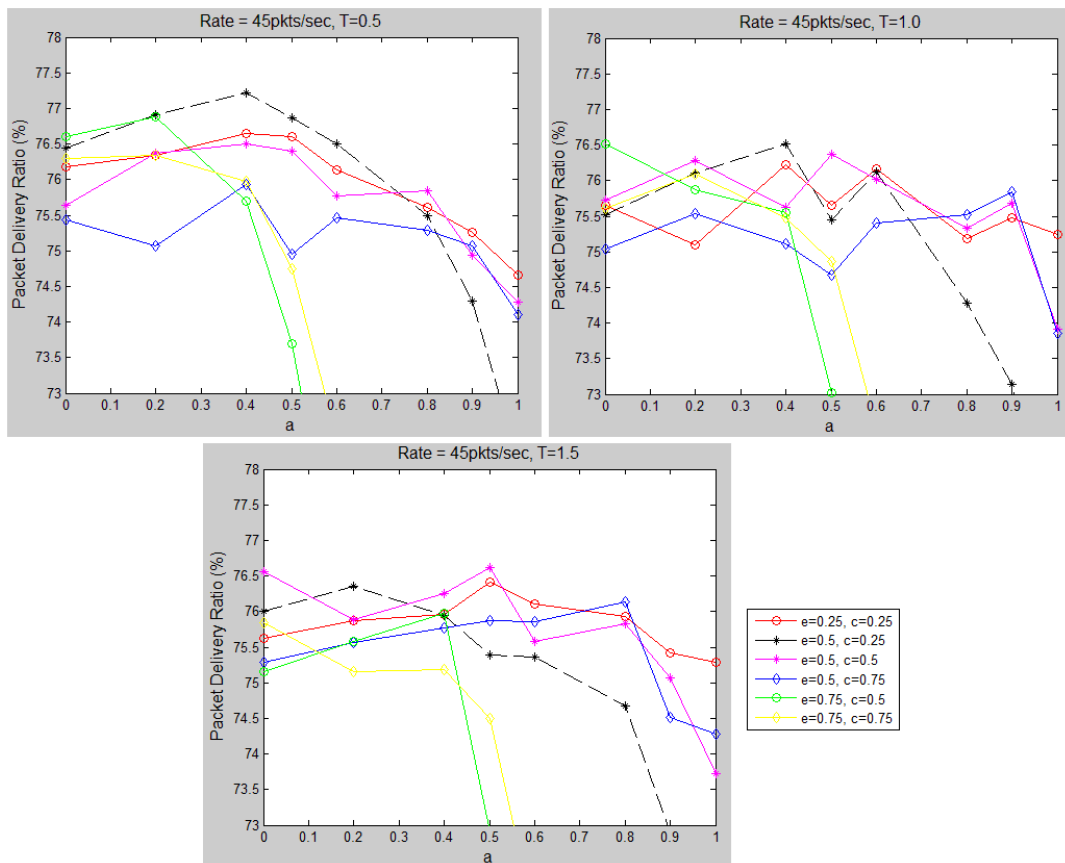


Σχήμα 5.43: Αποτελέσματα PDR ($T=0.5$, $c=0.25$ και $c=0.5$)



Σχήμα 5.44: Αποτελέσματα PDR ($T=0.5$ και $c=0.75$)

Στις γραφικές παραστάσεις στα σχήματα 5.43 και 5.44 φαίνεται το PDR για $T=0.5$, και $c=0.25$, $c=0.5$ και $c=0.75$. Όπως μπορούμε να διακρίνουμε από το σχήμα 5.44 όπου $c=0.75$, το PDR είναι χαμηλότερο σε σχέση με τις δύο άλλες παραστάσεις στο σχήμα 5.43, ανεξαρτήτως της τιμής των παραμέτρων e και a . Απορρίπτοντας την τελευταία γραφική παράσταση και κοιτάζοντας τις άλλες δύο, παρατηρούμε πως τα καλύτερα αποτελέσματα συνολικά είναι για e ίσο με 0.25 και 0.5 όταν το c ισούται με 0.25. Επίσης, όταν το $c=0.5$, τα καλύτερα αποτελέσματα φαίνεται να είναι για e ίσο με 0.25, 0.5 και 0.75. Σε αυτή την γραφική όμως διακρίνουμε πως όταν το $a=0.2$ και το $e=0.75$ είχαμε και πάλι καλά αποτελέσματα. Αυτές τις περιπτώσεις θα παρουσιάσω συνολικά στην συνέχεια. Αξίζει να σημειωθεί πως σύμφωνα με τις πιο πάνω τιμές φαίνεται πως η παράμετρος a έχει καλά αποτελέσματα όταν κυμαίνεται μεταξύ 0.2 και 0.6 ενώ πέραν του 0.6 παρατηρείται μείωση σε όλες τις περιπτώσεις.



Σχήμα 5.45: Αποτελέσματα PDR ως προς το T

Στο σχήμα 5.45 παρατηρούμε τις καλύτερες περιπτώσεις του PDR για data rate=45pkts/sec ως προς το T. Όπως μπορούμε να δούμε, ο καλύτερος συνδυασμός παραμέτρων είναι: **T=0.5, a=0.4, e=0.5, c=0.25** και μάλιστα με αρκετή διαφορά.

Παρατηρούμε γενικά αρκετά καλά αποτελέσματα για:

$A \in \{0.2, 0.4, 0.5, 0.6\}$

$E \in \{0.25, 0.5\}$

$C \in \{0.25, 0.5\}$

Καλύτερος συνδυασμός παραμέτρων:

T: 0.5

E: 0.5

C: 0.25

A: 0.4

Γενικά όμως είχαμε καλά αποτελέσματα για e μεταξύ 0.25 και 0.5, T μεταξύ 0.5 και 1.0, a μεταξύ 0.2 και 0.6 και τέλος, c μεταξύ 0.25 και 0.5.

5.3.4 Παρατηρήσεις

Περίοδος δειγματοληψίας (T): Σε αυτό το σενάριο, τα αποτελέσματα για το sampling period δεν έπαιζαν καθοριστικό ρόλο στην επιλογή των παραμέτρων. Αυτό οφείλεται στην μορφή του δικτύου, δηλαδή η θέση που βρίσκονταν οι κόμβοι που δημιουργούσαν τα δεδομένα, δεν προκαλούσε προβλήματα ώστε να παίζει τόσο σημαντικό ρόλο το διάστημα που μεσολαβούσε για να σταλούν τα control messages. Είναι οι άλλες παράμετροι που έπαιζαν πιο σημαντικό ρόλο. Από την στιγμή που το T είχε καλές τιμές μεταξύ 0.5 και 1.5 προτιμούμε την τιμή 1.5 για να μην γεμίζει το δίκτυο με control packets και να σπαταλείται ενέργεια. Για data rate ίσο με 45pkts/sec είχαμε ελαφρώς καλύτερα αποτελέσματα για $T=0.5$ όμως η διαφορά ήταν μοναχά 0.8% κι έτσι δεν έχει τόση σημασία και μπορούμε και σε αυτή την περίπτωση να χρησιμοποιήσουμε την τιμή 1.5.

$$T=1.5$$

Παράμετρος e : Όταν το e είναι πολύ μεγάλο, σημαίνει λαμβάνονται υπόψη κόμβοι με ίσο hop count προς το sink με μεγάλη πιθανότητα. Όταν το a είναι πάνω από 0.6 σημαίνει πως μας ενδιαφέρει περισσότερο η απώθηση από κόμβους με ψηλά φορτωμένο ασύρματο κανάλι. Αυτό όμως οδηγεί τα πακέτα σε κύκλους ανάμεσα σε κόμβους που δεν έχουν προβλήματα συμφόρησης κι έτσι αυξάνονται οι καθυστερήσεις και μειώνεται το PDR. Γενικά παρατηρούμε πως η παράμετρος e ήταν πάρα πολύ σημαντική και είχαμε πάρα πολύ καλά αποτελέσματα για τιμές μεταξύ 0.5 και 0.75 αυστηρά μιλώντας. Λόγω της τοπολογίας του δικτύου είναι θεμιτό τα πακέτα να ανοίξουν, και να μην πηγαίνουν απλά ευθεία προς το sink. Επίσης, ένεκα του γεγονότος πως όλοι οι κόμβοι που δημιουργούν τα δεδομένα βρίσκονται απέναντι από το sink (άλλοι πιο μπροστά και άλλοι πιο πίσω) υπάρχουν πάρα πολλά σύντομα μονοπάτια για να εκμεταλλευτούν κι έτσι η παράμετρος e δεν μπορεί να είναι μηδέν. Η ιδανικότερη τιμή ήταν το 0.5 όπως φάνηκε και για τα τρία data rates, τιμή αρκετά ικανοποιητική σχεδόν για όλα τα σενάρια.

$$E=0.5$$

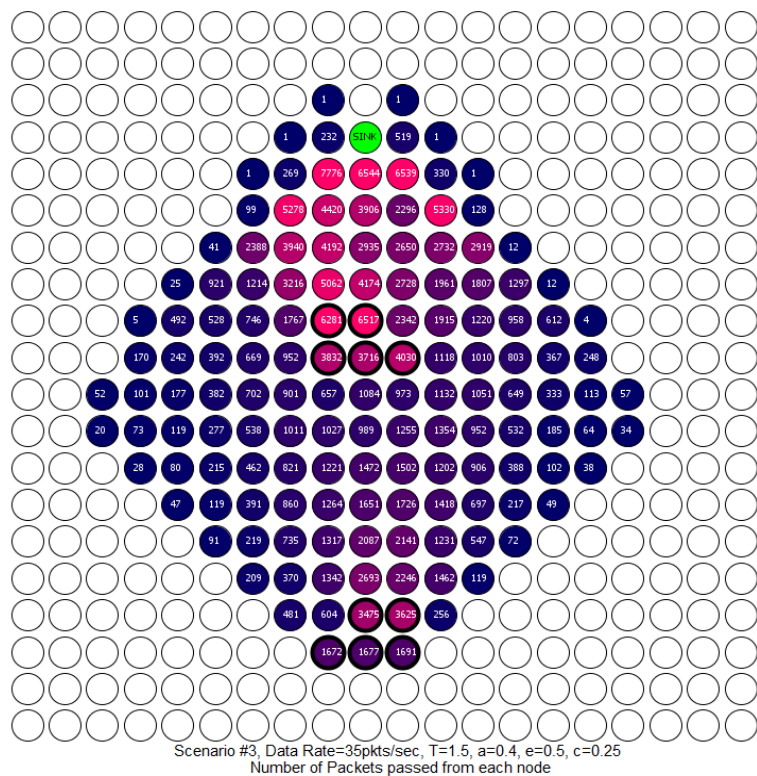
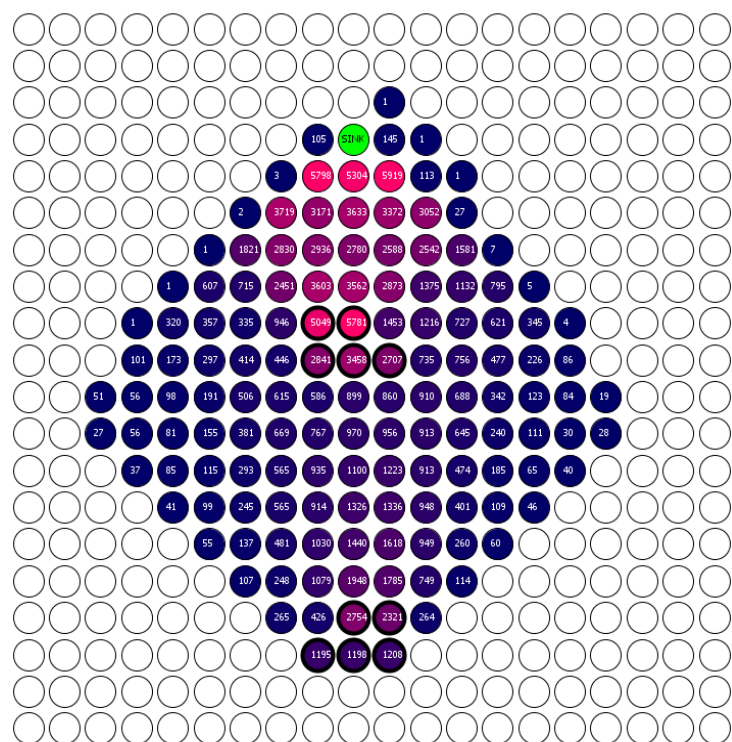
Παράμετρος c: Όσον αφορά την τυχειότητα φαίνεται πως τα καλύτερα αποτελέσματα κυμαίνονται μεταξύ 0.25 και 0.5 πάντοτε αυστηρά μιλώντας. Φαίνεται πως είναι επιθυμητό να υπάρχει κάποια δόση τυχειότητας ειδικά σε τέτοιου είδους τοπολογίες για να μην πηγαίνουν όλα τα πακέτα από το ίδιο μονοπάτι.

$$C=0.25$$

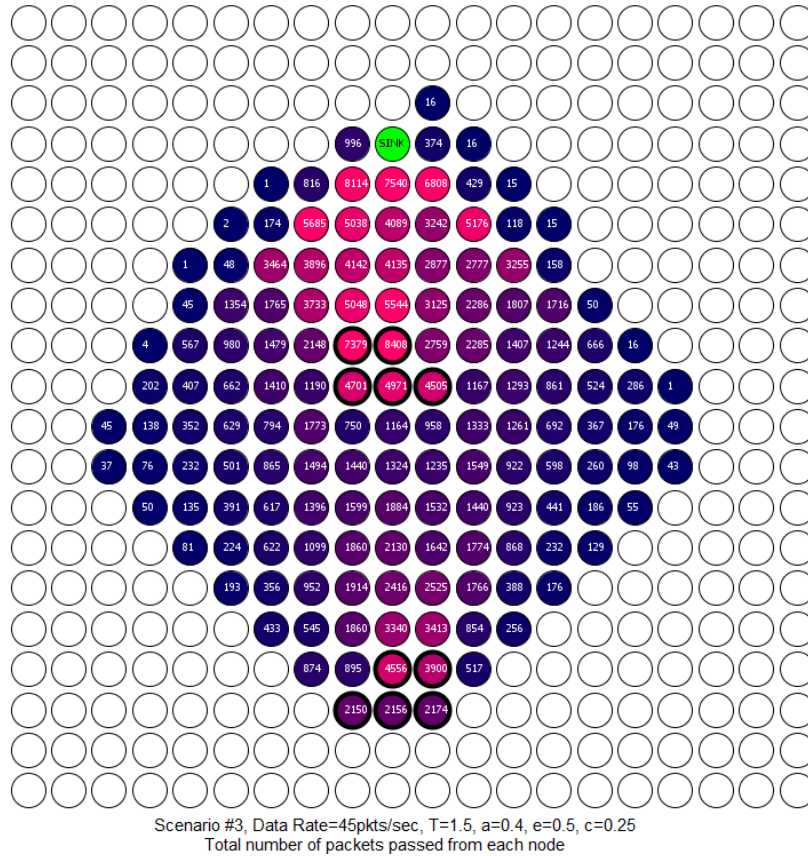
Παράμετρος a: Όσον αφορά την παράμετρο a φαίνεται πως τα καλύτερα αποτελέσματα κυμαίνονται μεταξύ 0.2 και 0.6 και ίσως η ιδανικότερη τιμή να είναι το 0.4. Βλέπουμε λοιπόν πως τόσο η κατάσταση του ασύρματου καναλιού, όσο και το φόρτο που έχει ο κάθε κόμβος είναι εξίσου σημαντικά. Απλά πρέπει να σημειωθεί πως για data rate ίσο με 25pkts/sec είχαμε καλύτερα αποτελέσματα για $a=0.2$ όμως η διαφορά ήταν μικρή (~0.8%) κι έτσι μπορούμε να χρησιμοποιήσουμε την τιμή 0.4 η οποία ήταν η καλύτερη για όλα τα άλλα data rates.

$$A=0.4$$

5.3.5 Γραφική απεικόνιση κίνησης πακέτων



Σχήμα 5.46: Συνολικός αριθμός πακέτων που πέρασαν από τον κάθε κόμβο

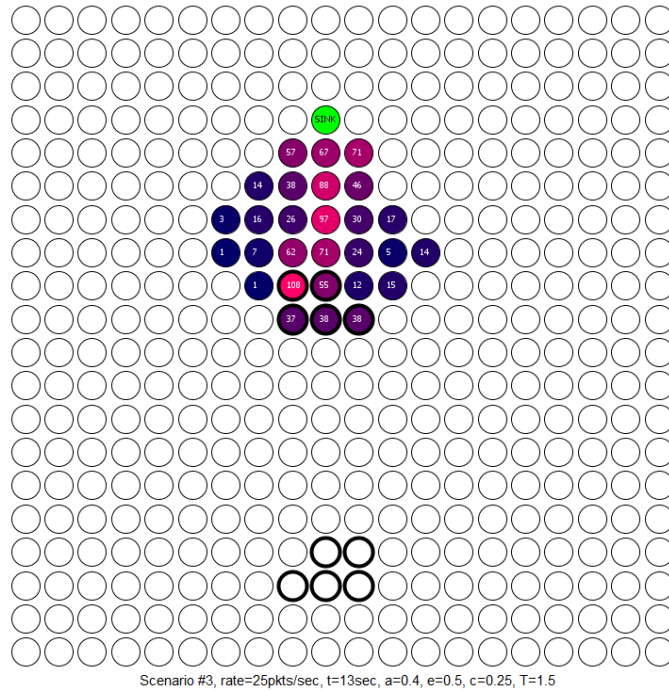


Σχήμα 5.47: Συνολικός αριθμός πακέτων που πέρασαν από τον κάθε κόμβο

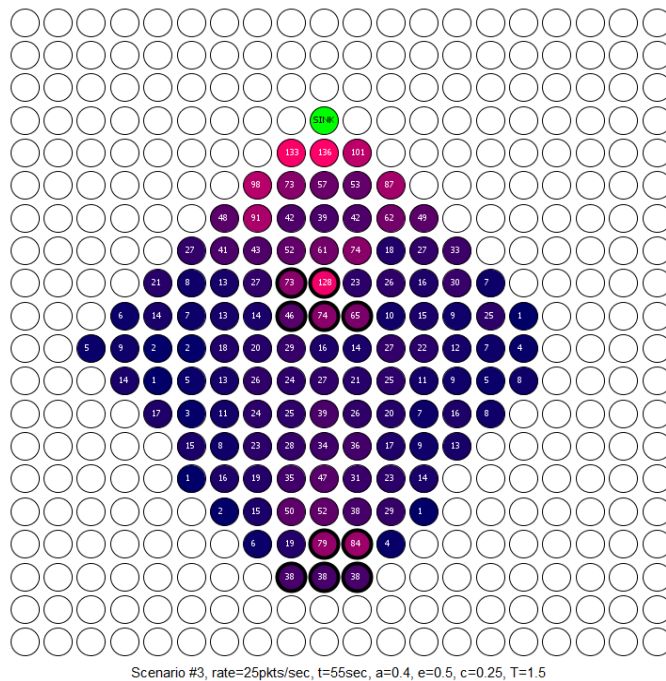
Στα σχήματα 5.46 και 5.47 παρουσιάζεται ο συνολικός αριθμός πακέτων που έχουν περάσει από τον κάθε κόμβο για τα τρία data rates. Αποφάσισα πως ο καλύτερος συνδυασμός παραμέτρων ήταν ο εξής:

$$T=1.5, a=0.4, e=0.5, c=0.25.$$

Για την ακρίβεια όσον αφορά το data rate ίσο με 45pkts/sec υπήρχε και πιο καλός συνδυασμός παραμέτρων όμως επειδή οι διαφορές στο PDR ήταν πολύ μικρές θεώρησα πως δεν χρειαζόταν να χρησιμοποιήσω εκείνο τον συνδυασμό.



Σχήμα 5.48: Στιγμιότυπο κίνησης πακέτων για τη χρονική στιγμή $t=13\text{sec}$

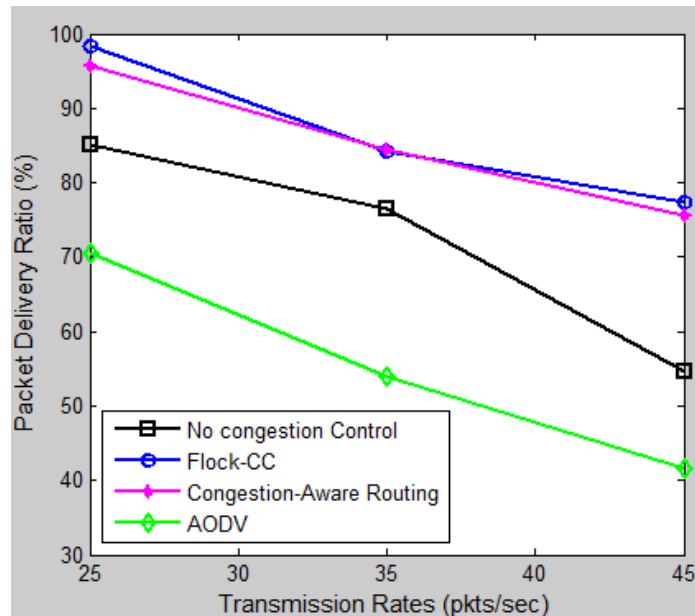


Σχήμα 5.49: Στιγμιότυπο κίνησης πακέτων για τη χρονική στιγμή $t=55\text{sec}$

Στις πιο πάνω εικόνες παρουσιάζονται ο αριθμός των πακέτων που καταφθάνουν σε κάθε κόμβο κατά τις χρονικές στιγμές των 13 και 55 δευτερολέπτων. Δηλαδή αμέσως μετά την ενεργοποίηση των 2 ομάδων κόμβων που δημιουργούν τα πακέτα.

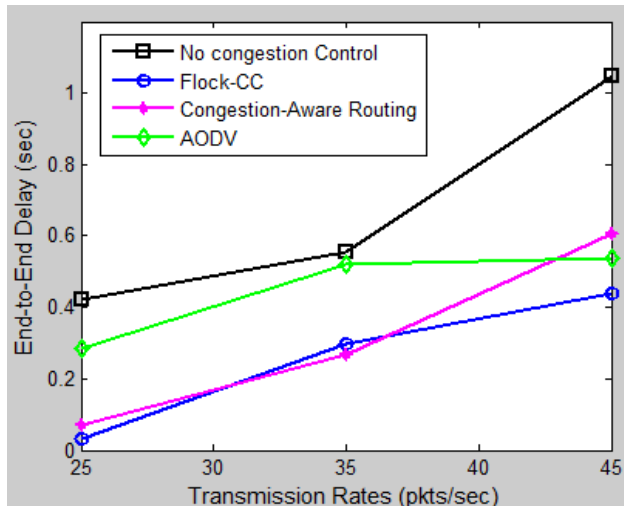
5.3.6 Σύγκριση αλγορίθμων

Για όλες τις τιμές του data rate ο αλγόριθμος Flock-CC χρησιμοποιεί τις τιμές: $T=1.5$, $a=0.4$, $e=0.5$, $c=0.25$



Σχήμα 5.50: Αποτελέσματα PDR

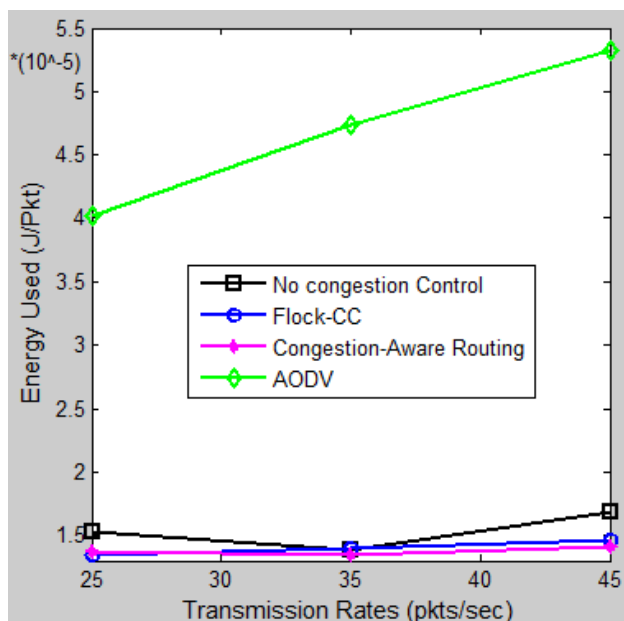
Στο σχήμα 5.50 παρουσιάζονται τα αποτελέσματα του PDR. Όπως μπορούμε να δούμε, όταν δεν εφαρμόζεται κάποια στρατηγική ελέγχου συμφόρησης και τα πακέτα στέλλονται απλά από το συντομότερο μονοπάτι, υπάρχει τεράστια μείωση στο PDR ακόμη και για data rate ίσο με 25pkts/sec. Συγκεκριμένα, το πρωτόκολλο No Congestion Control δίνει 85.15%, 76.43%, 54.58% για data rates 25, 35 και 45 pkts/sec αντίστοιχα. Ο αλγόριθμος Flock-CC είχε τα καλύτερα αποτελέσματα τα οποία ήταν 98.39%, 84.27% και 77.43%, ενώ ο CAwR είχε 95.59%, 84.39% και 75.67%. Τέλος, ο αλγόριθμος AODV είχε για ακόμη μια φορά το χαμηλότερο PDR το οποίο ήταν 70.51%, 54.03% και 41.67% για data rates ίσα με 25, 35 και 45pkts/sec αντίστοιχα.



Σχήμα 5.51: Αποτελέσματα καθυστερήσεων

Δίπλα φαίνεται το end-to-end delay για τους τέσσερις αλγορίθμους. Παρατηρούμε και πάλι πως ο πρώτος αλγόριθμος (no congestion) έχει τις μεγαλύτερες καθυστερήσεις οι οποίες είναι 0.4sec, 0.56sec, 1.05sec για data rates ίσα με 25, 35, και

45pkts/sec αντίστοιχα. Για τα ίδια data rates, ο CAwR χρειάστηκε 0.07sec, 0.27sec, 0.61sec ενώ ο Flock-CC αλγόριθμος που δίνει σαφώς καλύτερα αποτελέσματα χρειάστηκε μόλις 0.03sec, 0.3sec και 0.4sec. Τέλος, ο AODV χρειάστηκε 0.284sec, 0.522sec και 0.54sec για τα ίδια data rates.



Σχήμα 5.52: Αποτελέσματα δαπανώμενης ενέργειας ανά πακέτο

Στο σχήμα 5.52 παρουσιάζεται το πόσο της ενέργειας που δαπανήθηκε ανά πακέτο για τον κάθε αλγόριθμο. Παρατηρούμε πως οι αλγόριθμοι Flock-CC και CAwR δίνουν τα καλύτερα αποτελέσματα τα οποία κυμαίνονται στα $1.4 \cdot 10^{-5}$ J/pkt για όλα τα data rates. Ο αλγόριθμος No Congestion Control είχε ελαφρώς

χειρότερα αποτελέσματα για rates ίσα με 25 και 45 pkts/sec και τα οποία ισούνται με $1.53 \cdot 10^{-5}$ και $1.7 \cdot 10^{-5}$ J/pkt αντίστοιχα. Για μία ακόμη φορά ο AODV φαίνεται αν έχει την μεγαλύτερη ενεργειακή σπατάλη η οποία είναι 4-5.3 J/pkt για όλα τα data rates.

5.3.7 Σύνοψη

Σε αυτό το σενάριο μελετούσαμε την συμπεριφορά του αλγορίθμου έπειτα από εμφάνιση κάποιας περιοχής συμφόρησης μπροστά από την ομάδα δημιουργίας πακέτων, η οποία στην προκειμένη περίπτωση ήταν μια άλλη ομάδα δημιουργίας πακέτων. Κατ' ακρίβεια ο αλγόριθμος λειτούργησε για ακόμη μια φορά πολύ αποδοτικά. Με την ενεργοποίηση της πρώτης ομάδας των κόμβων που στέλλουν πακέτα, προκλήθηκε συμφόρηση λόγω της κοντινής τους απόστασης στο sink και ο αλγόριθμος μπόρεσε να το ανιχνεύσει και να αποστείλει τα πακέτα από εναλλακτικά μονοπάτια εκμεταλλευόμενος τους κόμβους στα πλάγια. Είχε και πάλι το ψηλότερο ποσοστό PDR μαζί με το πρωτόκολλο CAwR, είχε τις χαμηλότερες καθυστερήσεις και πολύ ικανοποιητικά αποτελέσματα όσον αφορά την χρήση των ενεργειακών αποθεμάτων. Η συμφόρηση προκλήθηκε στην περιοχή στο sink εξαιτίας την κοντινής απόστασης της πρώτης ομάδας κόμβων δημιουργίας πακέτων και δεν θα μπορούσε να αποφευχθεί με κανένα τρόπο.

Κεφάλαιο 6

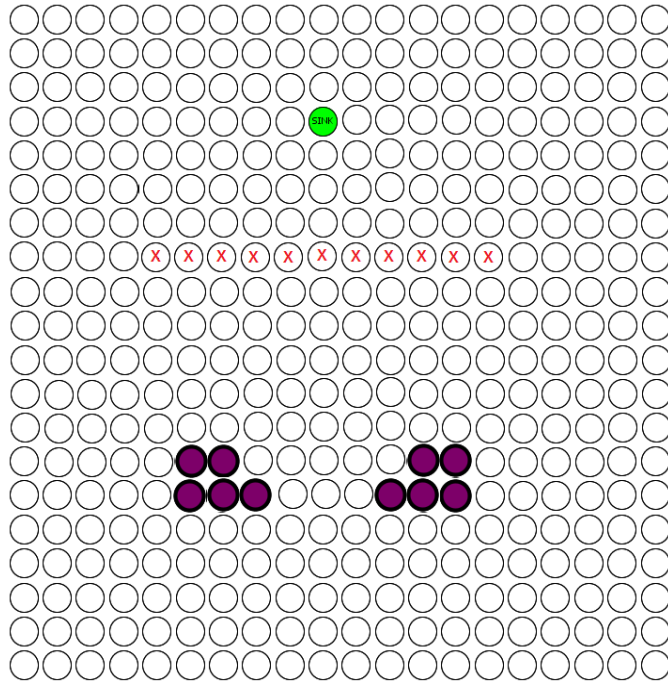
Αξιολόγηση ευρωστίας του αλγορίθμου

6.1	Κατάρρευση κόμβων σε ευθεία γραμμή	89
6.2	Κατάρρευση κόμβων δημιουργώντας καμπύλη.....	107
6.3	Αντιμετώπιση συμφόρησης διαπερνώντας τους πλαϊνούς νεκρούς κόμβους...	114

Σε αυτό το κεφάλαιο παρουσιάζονται τα αποτελέσματα δύο σεναρίων στα οποία καταρρέουν κάποιοι κόμβοι κατά τη διάρκεια της προσομοίωσης. Στο πρώτο από αυτά τα σενάρια η ομάδα των κόμβων που καταρρέει βρίσκεται σε ευθεία γραμμή στην μέση του δικτύου. Στο τελευταίο σενάριο καταρρέουν δύο γραμμές κόμβων οι οποίοι είναι τοποθετημένοι με τέτοιο τρόπο ώστε να δημιουργούν μια καμπύλη στην οποία είναι πολύ εύκολο να παγιδευτούν τα πακέτα. Σε αυτό το σενάριο χρησιμοποιήθηκαν οι τιμές που βρήκα ως καλύτερες για το πρώτο σενάριο (κατάρρευση σε ευθεία γραμμή).

6.1 Κατάρρευση κόμβων στην μέση του δικτύου

Σε αυτό το σενάριο έχουμε 10 ενεργοποιημένους κόμβους, χωρισμένους σε δύο ομάδες των 5, που στέλλουν δεδομένα προς το σημείο sink το οποίο βρίσκεται ακριβώς απέναντι τους (βλέπε σχήμα 6.1). Σε κάποια χρονική στιγμή μετά την έναρξη της προσομοίωσης (50sec), μια σειρά κόμβων καταρρέει στην μέση του δικτύου και τα πακέτα πρέπει να μπορέσουν να βρουν έγκαιρα νέο δρομολόγιο.

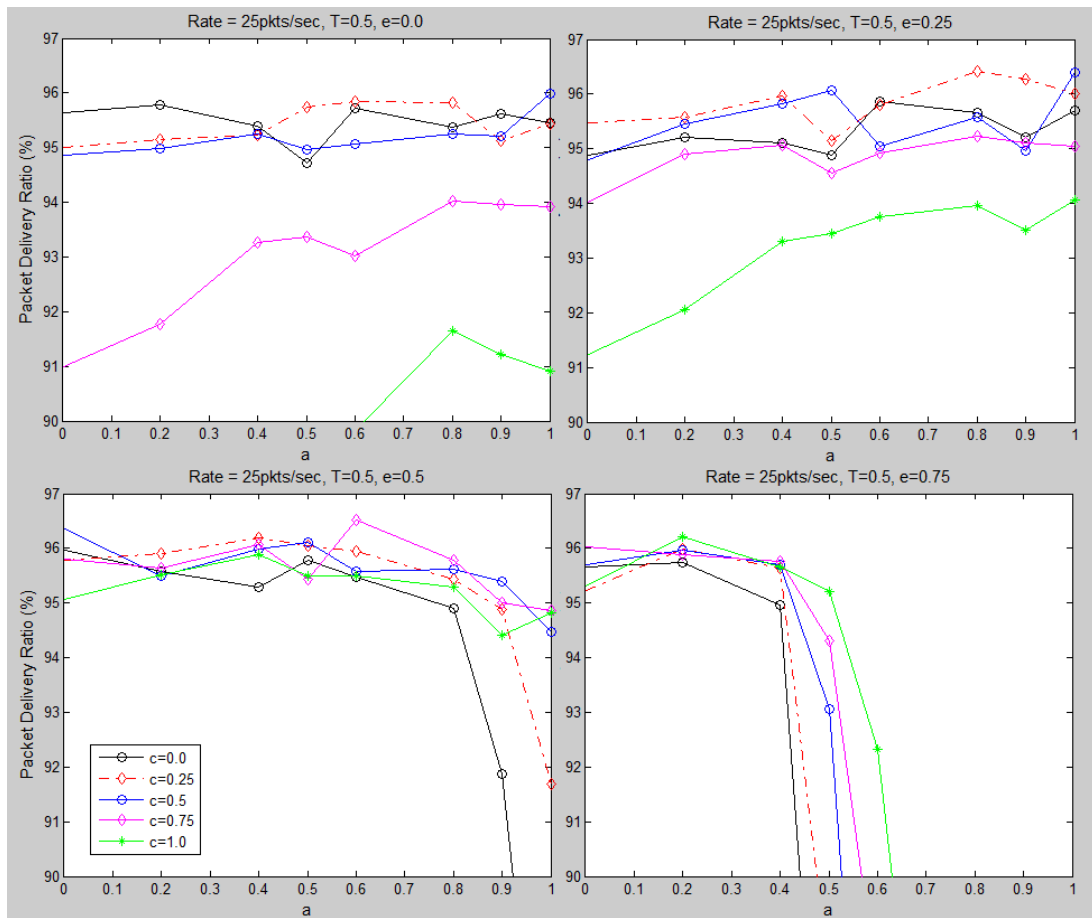


Σχήμα 6.1: Κόμβοι δημιουργίας πακέτων και κόμβοι που καταρρέουν

Μελετώντας τα αποτελέσματα φαίνεται πως καλύτερη τιμή για το PDR για data rate=25pkts/sec ήταν 96.51%, για 35pkts/sec ήταν 82.46% και τέλος για 45pkts/sec ήταν 67.01%. Βλέπουμε πως για 25pkts/sec δεν υπάρχει συμφόρηση στο δίκτυο, για 35pkts/sec προκαλείται ελαφριά συμφόρηση όμως για 45pkts/sec υπάρχει αρκετό πρόβλημα.

Σε αυτό το σενάριο αναμέναμε πως το sampling period θα έπαιζε καθοριστικό ρόλο διότι όσο πιο γρήγορα μπορέσουν οι κόμβοι να αντιληφθούν την περιοχή συμφόρησης τόσο πιο γρήγορα θα μπορούσαν να οργανωθούν προκειμένου να την αποφύγουν, όπως κι έγινε. Παρατήρησα πως σε όλες τις περιπτώσεις η καλύτερη τιμή για το T ήταν το 0.5 και μάλιστα με διαφορά. Η τιμή αυτή ήταν αναμενόμενη αφού θέλουμε να λαμβάνουμε σε σύντομο χρονικό διάστημα τα control packets για να μπορέσουν να ενημερωθούν έγκαιρα οι διάφοροι κόμβοι πως έχει καταρρεύσει μια σειρά κόμβων στη μέση του δικτύου και να προσπαθήσουν να στείλουν τα πακέτα από διαφορετικά δρομολόγια ώστε να αποφευχθεί η περιοχή όπου παρουσιάζεται πρόβλημα. Συνεπώς, θα κρατήσω σταθερή αυτή την παράμετρο προκειμένου να μελετήσω όλες τις υπόλοιπες για τις οποίες δεν υπάρχουν και τόσο ξεκάθαρα αποτελέσματα.

6.1.1 Ελαφριά Συμφόρηση (Data Rate=25pkts/sec)

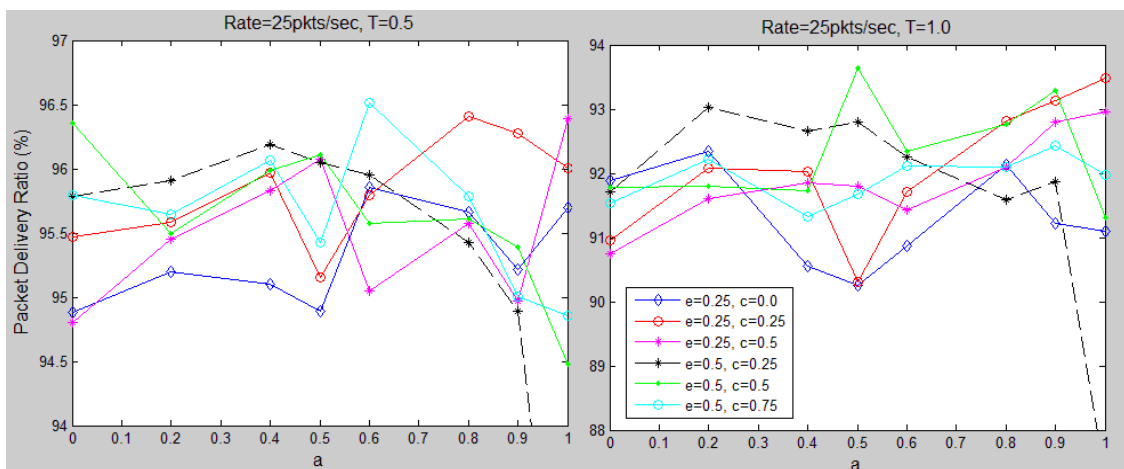


Σχήμα 6.2: Αποτελέσματα PDR (T=0.5)

Στο σχήμα 6.2 παρουσιάζονται τα αποτελέσματα του PDR συναρτήσει των παραμέτρων a και c . Σε κάθε μια από τις τέσσερις γραφικές παραστάσεις, η περίοδος δειγματοληψίας διατηρείται σταθερή σε 0.5 ενώ το e μεταβάλλεται. Όπως φαίνεται η ιδανικότερη τιμή για την παράμετρο e είναι το 0.25 και 0.5. Δεδομένης της θέσης των κόμβων που στέλλουν δεδομένα και των κόμβων που καταρρέουν, το e πρέπει σίγουρα να είναι μεγαλύτερο του μηδέν. Αν είναι πολύ μεγάλη η τιμή του τότε μπορεί να προκαλούνται κύκλοι στο δίκτυο αυξάνοντας τις καθυστερήσεις. Αν είναι πάρα πολύ μικρή η τιμή του, π.χ. 0.0 όταν οι κόμβοι προσπαθούν να υπολογίσουν σε ποιο κόμβο πρέπει να στείλουν τα πακέτα τους θα επιλέγουν μόνο τους κόμβους που έχουν μικρότερο hop count από τους ίδιους. Αυτό όμως περιορίζει πολύ τους εναλλακτικούς

κόμβους στους οποίους θα μπορούσαν να προωθηθούν τα πακέτα με αποτέλεσμα να αυξάνονται οι καθυστερήσεις και να χάνονται πακέτα.

Σχετικά με την παράμετρο c παρατηρούμε πως για κάθε μια από τις τιμές της παραμέτρου e , η μόνη τιμή που δίνει πάντοτε καλά αποτελέσματα για το c είναι το 0.25. Όταν το e κυμαίνεται μεταξύ 0.0 και 0.25 παρατηρούμε πως υπάρχει τεράστια μείωση στο PDR για c μεταξύ 0.75 και 1.0. Αυτό οφείλεται στην δημιουργία κύκλων των πακέτων ανάμεσα σε κόμβους που δεν παρουσιάζουν προβλήματα συμφόρησης.

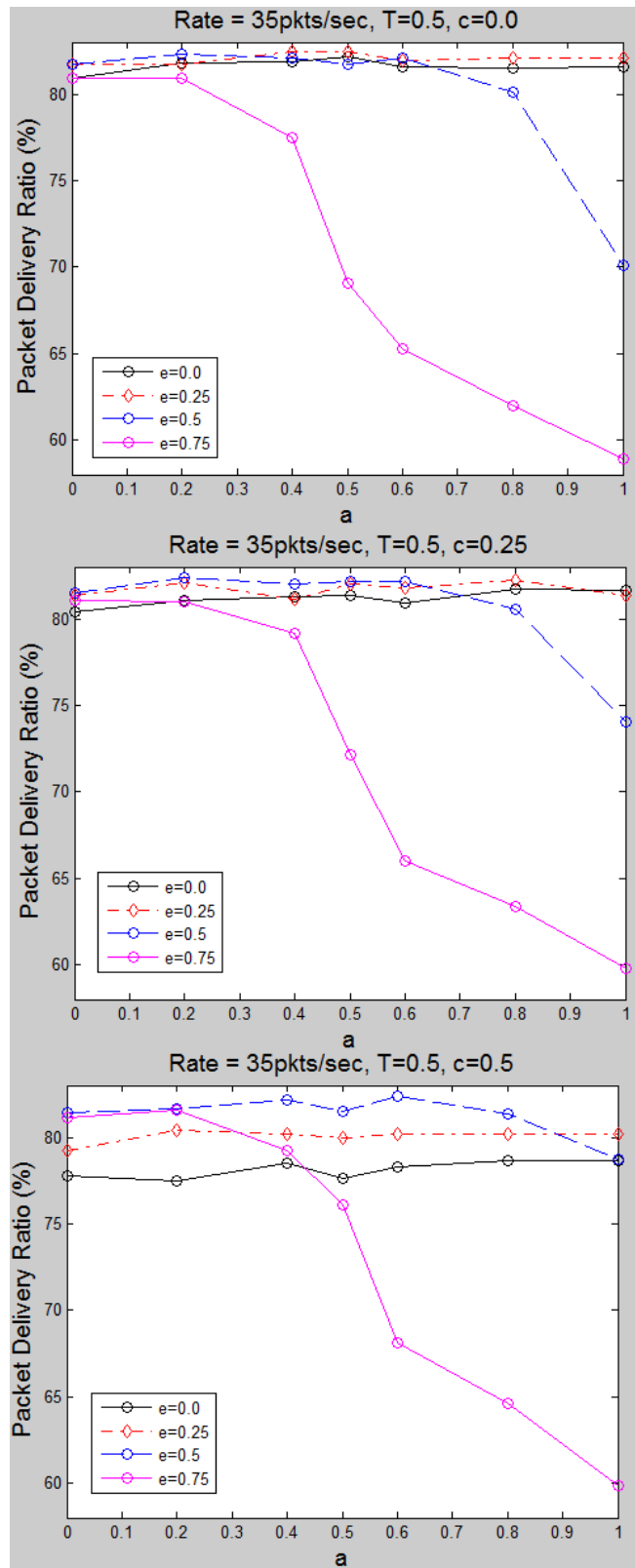


Σχήμα 6.3: Αποτελέσματα PDR ως προς το T

Στο σχήμα 6.3(α) παρουσιάζονται αναλυτικά οι καλύτερες περιπτώσεις. Όπως μπορούμε να ξεχωρίσουμε, όταν το e ισούται με 0.5 και το c με 0.25, το PDR παρουσιάζεται με μια ομαλή καμπύλη που δίνει πάρα πολύ καλά αποτελέσματα ειδικά για $a=0.4$. Ελαφρώς καλύτερα αποτελέσματα είχαμε για $a=0.6$, $e=0.5$, $c=0.75$ και εναλλακτικά $a=0.8$, $e=0.25$, $c=0.25$.

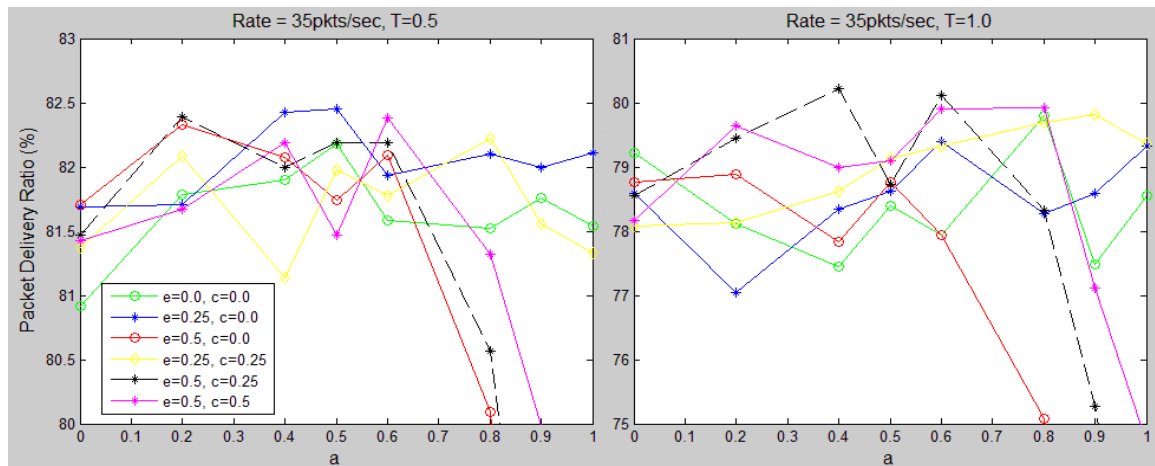
Πιο πάνω στο σχήμα 6.3(β) παρουσιάζεται η ίδια γραφική παράσταση με την διαφορά πως η περίοδος δειγματοληψίας διατηρείται σταθερή σε 1.0. Όπως μπορούμε να παρατηρήσουμε το ποσοστό του PDR είναι πολύ πιο χαμηλό σε σχέση με την προηγούμενη παράσταση. Στην καλύτερη περίπτωση φτάνει μόλις το 94% ενώ για $T=0.5$ ήταν περίπου 96.5%.

6.1.2 Πρόβλημα Συμφόρησης (Data Rate=35pkts/sec)



Σχήμα 6.4: Αποτελέσματα PDR

Στο σχήμα 6.4 παρουσιάζεται το PDR για data rate=35pkts/sec. Η παράμετρος T διατηρείται σταθερή σε 0.5 αφού ήταν η καλύτερη τιμή για όλες τις περιπτώσεις. Σε κάθε μια από αυτές τις παραστάσεις μεταβάλλεται η τιμή της παραμέτρου c και βλέπουμε το ποσοστό επιτυχημένης αποστολής σε σχέση με τις παραμέτρους a και e . Όπως φαίνεται η καλύτερη τιμή για την παράμετρο e είναι το 0.5 και το 0.25. Ενώ η χειρότερη τιμή είναι το 0.75. Όσον αφορά την παράμετρο a φαίνεται πως τα καλύτερα αποτελέσματα είναι για τιμές μεταξύ 0.2 και 0.6.



Σχήμα 6.5: Αποτελέσματα PDR ως προς το T

Στο σχήμα 6.5(α) φαίνονται οι καλύτερες περιπτώσεις που δίνουν τα ψηλότερα αποτελέσματα του PDR όταν διατηρείται σταθερό το T σε 0.5. Όπως μπορούμε να δούμε πάρα πολύ καλά αποτελέσματα είχαμε όταν $a \in \{0.4, 0.5\}$ για $e=0.25$ και $c=0.0$. Επίσης, εξίσου καλά αποτελέσματα είχαμε για $a=0.2$, $e=0.5$, $c=0.25$.

120

Στο σχήμα 6.5(β) παρουσιάζονται τα ίδια αποτελέσματα με το σχήμα 6.6 με μόνη διαφορά πως η περίοδος δειγματοληψίας διατηρείται σταθερή σε 1.0 αντί 0.5. Όπως μπορούμε να δούμε, υπάρχει πάρα πολύ μεγάλη διαφορά ανάμεσα στις δύο γραφικές παραστάσεις. Όταν το T παραμένει σταθερό σε 1.0 το PDR είναι πολύ πιο χαμηλό παρά όταν το T ήταν ίσο με 0.5 αφού η καλύτερη τιμή ανέρχεται μόλις στο 80%.

Καλύτερα αποτελέσματα για data rate=35 pkts/sec είναι:

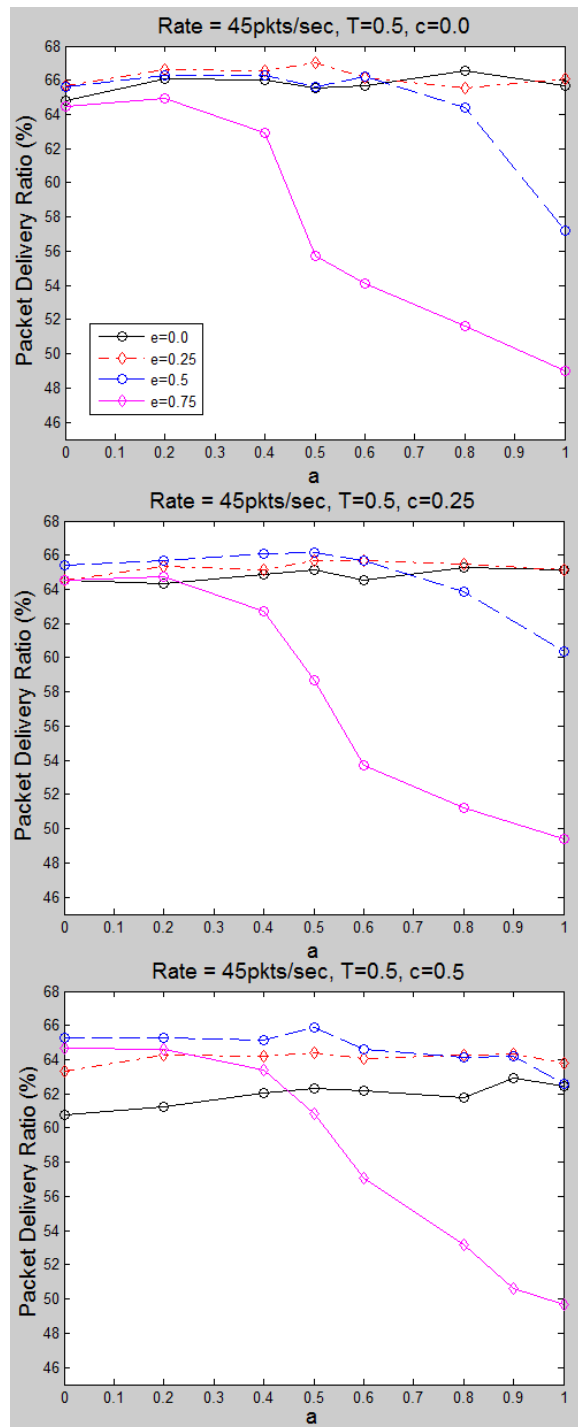
$$T=0.5$$

$$A \in \{0.2, 0.4, 0.5, 0.6\}$$

$$E \in \{0.25, 0.5\}$$

$$C \in \{0.0, 0.25\}$$

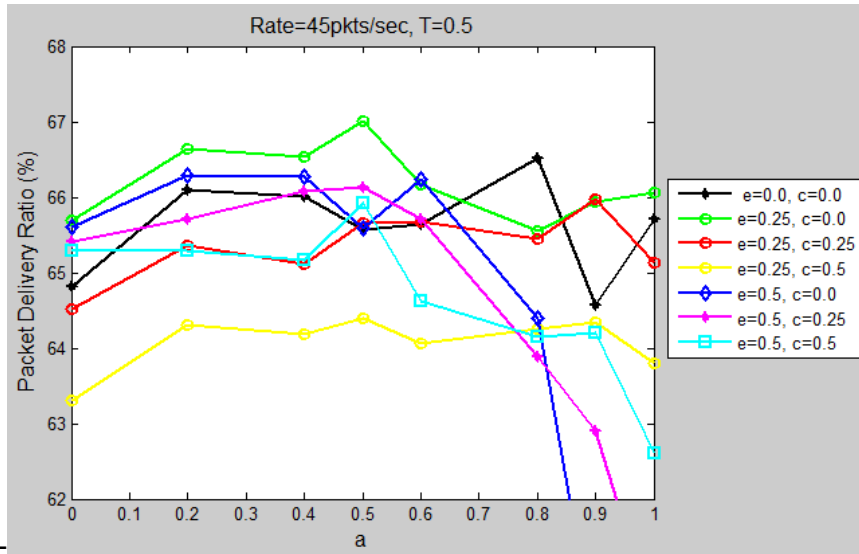
6.1.3 Σοβαρό πρόβλημα συμφόρησης (Data Rate=45pkts/sec)



Σχήμα 6.6: Αποτελέσματα PDR (T=0.5)

Σε αυτές τις γραφικές παραστάσεις παρουσιάζεται το ποσοστό επιτυχημένης αποστολής πακέτων για data rate=45%. Διατηρώ το T σταθερό σε 0.5 αφού ήταν η καλύτερη τιμή. Επίσης, τα καλύτερα αποτελέσματα ήταν όταν το c παίρνει τιμές 0.0, 0.25 και 0.5 έτσι σε αυτές τις γραφικές παραστάσεις θα μελετήσω την επίδραση των παραμέτρων a και e. Γενικά το ψηλότερο PDR ήταν 67.01%. Όσον αφορά την τρίτη παράσταση όπου το c ισούται με 0.5 φαίνεται πως έχουμε χαμηλότερο PDR με σχέση με τις άλλες δύο περιπτώσεις. Τα καλύτερα αποτελέσματα είναι όταν το c ισούται με μηδέν ή με 0.25. Μελετώντας τις δύο πρώτες γραφικές παραστάσεις βλέπουμε πως και στις δύο περιπτώσεις η ιδανικότερη τιμή για την παράμετρο a είναι το 0.5, όμως αρκετά καλά αποτελέσματα είχαμε και για 0.2, 0.4 και 0.6. Επίσης, η καλύτερη τιμή για την παράμετρο e φαίνεται να είναι το 0.25 και το 0.5. Όταν το a είναι πολύ μεγάλο

και το e επίσης, βλέπουμε μια πάρα πολύ μεγάλη μείωση στο ποσοστό επιτυχημένης αποστολής, πράγμα το οποίο φάνηκε και στις άλλες περιπτώσεις όπου το data rate ήταν χαμηλότερο.



Σχήμα 6.7: Συνολικά καλύτερα αποτελέσματα PDR

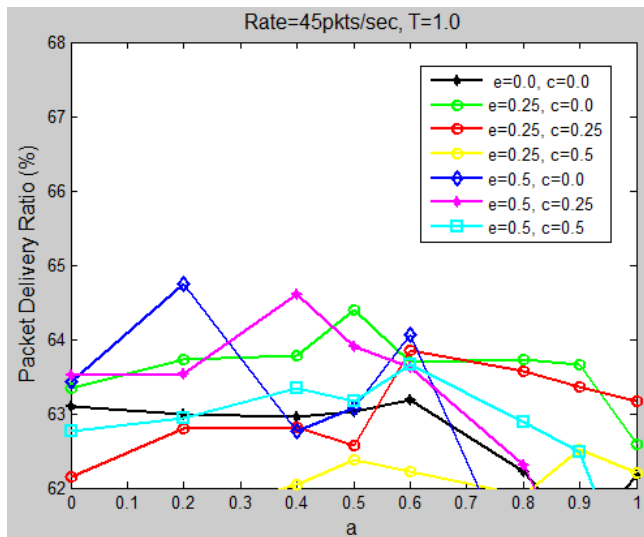
Στο σχήμα 6.7 απεικονίζονται συνολικά οι καλύτερες περιπτώσεις για το PDR.

Όπως μπορούμε να δούμε, τα καλύτερα αποτελέσματα είναι:

$$A \in \{0.2, 0.4, 0.5, 0.6\}$$

$$E \in \{0.25, 0.5\}$$

$$C \in \{0.0, 0.25\}$$



Σχήμα 6.8: Συνολικά καλύτερα αποτελέσματα PDR για T=1.0

Σε αυτή την γραφική παράσταση παρατηρούμε ακριβώς τα ίδια αποτελέσματα για T=1.0 αντί 0.5 που ήταν προηγουμένως. Μπορούμε να δούμε πως υπάρχει αρκετά μεγάλη διαφορά για όλες τις τιμές των παραμέτρων. Αφού

ακόμη και η καλύτερη δυνατή τιμή ανέρχεται μόλις στο 75% εν συγκρίσει με προηγουμένως που ήταν 77%.

6.1.4 Παρατηρήσεις

Περίοδος δειγματοληψίας (T): Το συγκεκριμένο σενάριο ήταν αρκετά διαφορετικό από τα υπόλοιπα και αναμέναμε πως θα έπαιζε πάρα πολύ σημαντικό ρόλο η παράμετρος T, όπως κι έγινε. Για όλα τα data rates η καλύτερη τιμή για το sampling period ήταν το 0.5 με αρκετή διαφορά. Σε αυτό το σενάριο είχαμε δύο ομάδες απέναντι από το sink που έστελλαν δεδομένα. Σε κάποια φάση μια σειρά κόμβων στην μέση του δικτύου είχε καταρρεύσει. Το T είναι μικρό για να μπορούν να λαμβάνουν συχνά τα control packets οι διάφοροι κόμβοι και να μπορούν να αντιληφθούν έγκαιρα πως υπάρχει μια περιοχή με προβλήματα συμφόρησης και να στείλουν τα δεδομένα τους από άλλα μονοπάτια. Επίσης, είδαμε πως ακόμη και για τον καλύτερο συνδυασμό παραμέτρων με $T=0.5$, η αλλαγή του T σε 1.0 επέφερε τεράστια μείωση στο PDR.

$$T=0.5$$

Παράμετρος e: Όπως έχει φανεί η ιδανικότερη τιμή για αυτή την παράμετρο είναι το 0.5 για όλα τα data rates που έχω μελετήσει. Όταν το e ισούται με το μηδέν, οι κόμβοι που προσπαθούν να βρουν τον επόμενο κόμβο στον οποίο θα στείλουν τα δεδομένα τους, λαμβάνουν υπόψη τους μόνο κόμβους με μικρότερο hop distance από το sink. Αυτό όμως οδηγεί τους κόμβους στο να στέλλουν τα πακέτα τους στην κεντρική διαγώνιο, δηλαδή πάνω στην περιοχή όπου έχει προκύψει πρόβλημα με κάποιους κόμβους. Επίσης, όταν η τιμή του e είναι πολύ μεγάλη τα πακέτα οδηγούνται σε κύκλους. Γι' αυτό το λόγο τα καλύτερα αποτελέσματα ήταν για e ίσο με 0.5.

$$E=0.5$$

Τυχαιότητα (c): Η παράμετρος c φαίνεται καθαρά πως η ιδανική της τιμή είναι το 0.25. Όπως και στα προηγούμενα σενάρια, έτσι και σε αυτό είδαμε πως το c είναι καλύτερα να παίρνει τιμή μεγαλύτερη του μηδενός η οποία όμως να μην είναι πολύ μεγάλη. Η εισαγωγή του τυχαίου στοιχείου στην επιλογή κάποιου μονοπατιού μπορεί τελικά να επιφέρει καλά αποτελέσματα.

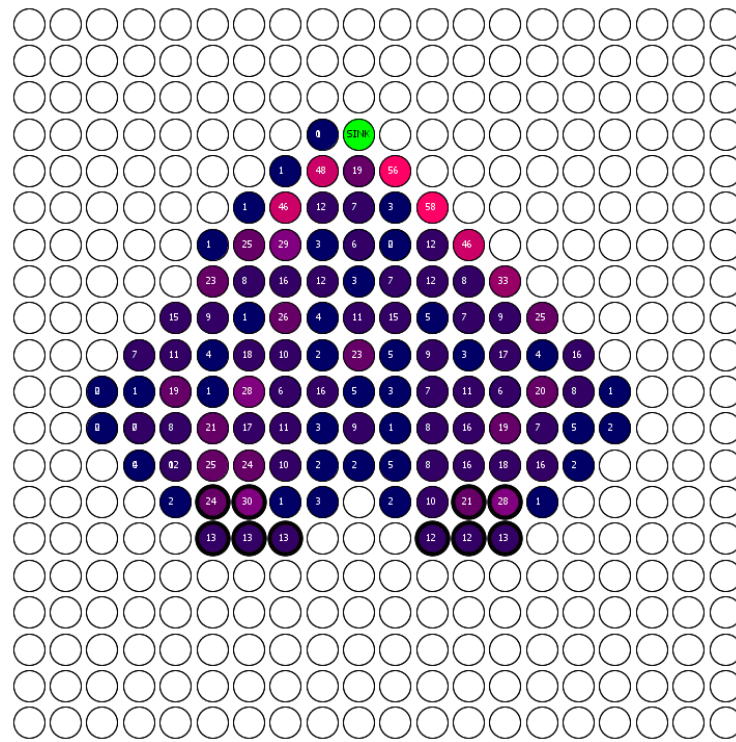
$$C=0.25$$

Παράμετρος a: Η ιδανικότερη τιμή για αυτή την παράμετρο είναι μεταξύ 0.2 και 0.6. Γενικά θεωρούνται εξίσου σημαντικά η κατάσταση της ουράς του κάθε κόμβου και το useful service ratio και στο σενάριο #1 είχαν βγει και πάλι αντίστοιχα αποτελέσματα.

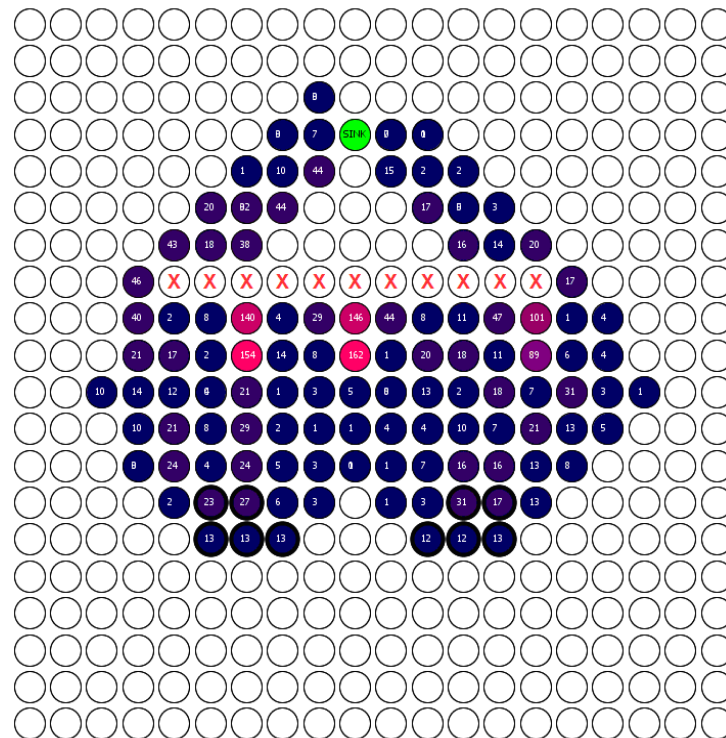
$$A=0.4 - 0.5$$

Στο σενάριο #1, όπου είχαμε 10 κόμβους να δημιουργούν πακέτα απέναντι από το sink στην άλλη άκρη του δικτύου, πήραμε παρόμοια αποτελέσματα. Η παράμετρος a έπαιρνε τιμές μεταξύ 0.4 και 0.6. Η παράμετρος e κυμαίνεται μεταξύ 0.25 και 0.5 και η παράμετρος c ήταν ίση με 0.25. Το sampling period ήταν διαφορετικό ($T=1.5$) λόγω του ότι ήταν διαφορετικοί οι κόμβοι που έστελλαν τα δεδομένα και δεν υπήρχαν οι 11 κόμβοι στην μέση του δικτύου οι οποίοι είχαν καταρρεύσει.

6.1.5 Γραφική απεικόνιση κίνησης πακέτων

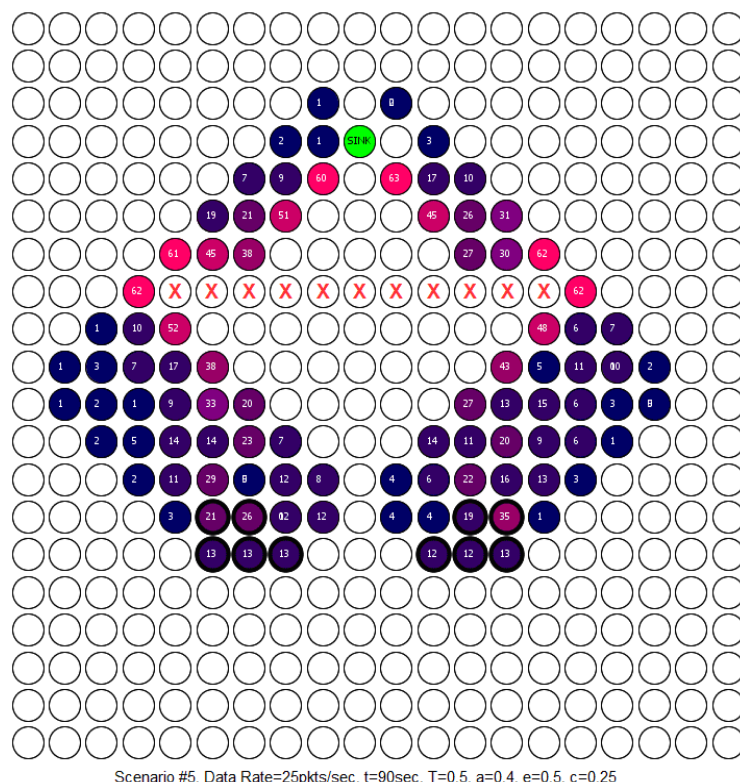


Scenario #5, Data Rate=25pkts/sec, T=0.5, a=0.4, e=0.5, c=0.25, t=13sec



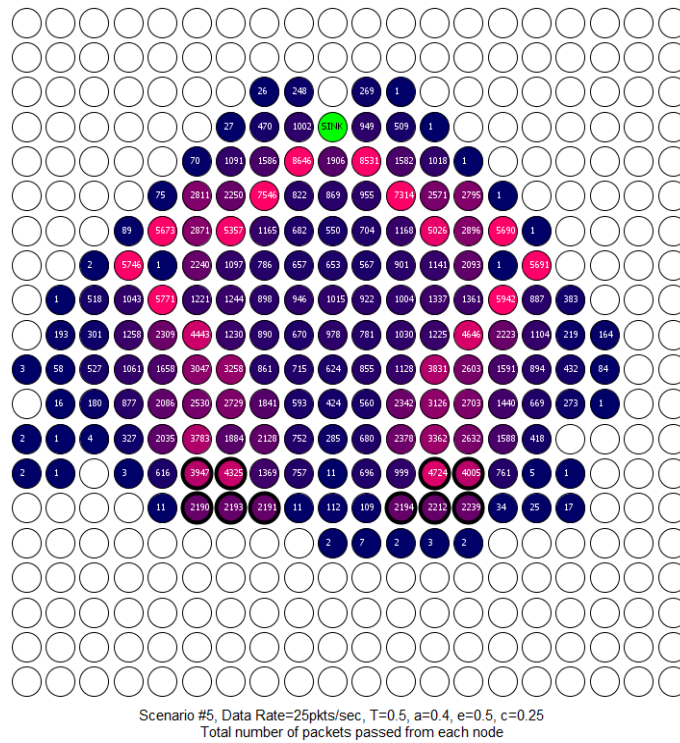
Scenario #5, Data Rate=25pkts/sec, t=53sec, a=0.4, e=0.5, c=0.25, T=0.5

Σχήμα 6.9: Συνολικός αριθμός πακέτων ανά κόμβο τις χρονικές στιγμές 13sec, 53sec



Σχήμα 6.10: Συνολικός αριθμός πακέτων ανά κόμβο την χρονική στιγμή των 90sec

Στα σχήματα 6.9 και 6.10 παρουσιάζονται στιγμιότυπα από τον αριθμό πακέτων που φτάνουν σε κάθε κόμβο κατά τις χρονικές στιγμές 13sec, 53sec και 90sec, δηλαδή λίγο πριν καταρρεύσουν οι ενδιάμεσοι κόμβοι, αμέσως μετά την κατάρρευση και μετά από λίγο διάστημα. Οι τιμές των παραμέτρων που χρησιμοποιήσα για data rate ίσο με 25pkts/sec ήταν $a=0.4$, $e=0.5$, $c=0.25$ και $T=0.5$. Παρατηρούμε πως μετά την κατάρρευση των κόμβων χρειάστηκε λίγο διάστημα προτού το αντιληφθούν οι κόμβοι και ξεκινήσουν να προωθούν τα πακέτα προς τα άκρα του δικτύου προκειμένου να αποφύγουν την περιοχή συμφόρησης. Παρατηρούμε πως προκαλείται συμφόρηση στην διαγώνιο μεταξύ των κόμβων που στέλλουν δεδομένα και των δύο κόμβων που βρίσκονται δεξιά και αριστερά της σειράς των κόμβων που έχουν καταρρεύσει. Όπως φαίνεται οι κόμβοι μιμούνται ορθά την κίνηση των πουλιών και αποφεύγουν την περιοχή όπου υπάρχει πρόβλημα αλλά επίσης βλέπουμε πως στέλλουν τα δεδομένα τους από πολλαπλά μονοπάτια.

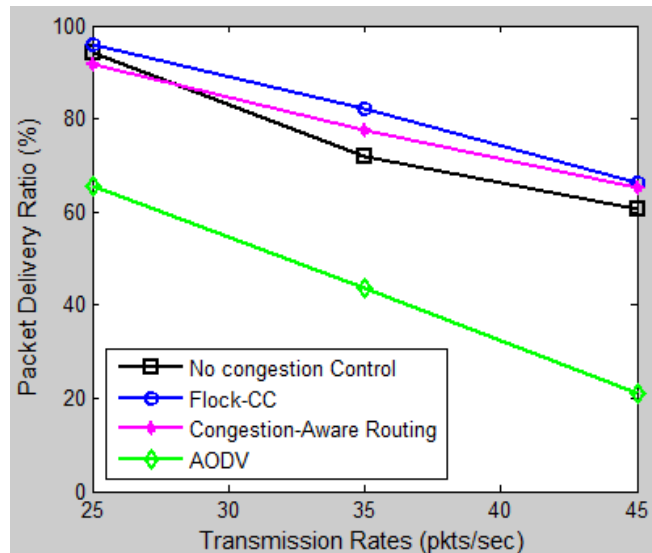


Σχήμα 6.11: Συνολικός αριθμός πακέτων που πέρασαν συνολικά από τον κάθε κόμβο

Στο σχήμα 6.11 παρουσιάζεται ο συνολικός αριθμός πακέτων που έχουν περάσει από τον κάθε κόμβο για data rate ίσο με 25pkts/sec. Όπως μπορούμε να παρατηρήσουμε, το κοπάδι των πακέτων έχει ανοίξει αρκετά και μάλιστα υπάρχουν πακέτα που έχουν σταλεί στους ακραίους αριστερά κόμβους. Επιπλέον, όπως μπορούμε να παρατηρήσουμε, η συμφόρηση προκαλείται στους κόμβους που δημιουργούν τα πακέτα και στις δύο διαγώνιους που ενώνουν την ομάδα των κόμβων δημιουργίας πακέτων με τους ακραίους κόμβους δεξιά κι αριστερά της ομάδας των κόμβων που έχουν καταρρεύσει. Επίσης, βλέπουμε πως λόγω της μεγάλης δυσκολίας που είχε αυτό το σενάριο στην αποστολή των πακέτων προς το sink, κάποια πακέτα είχαν σταλεί πίσω από τους κόμβους που τα δημιουργούν. Σε γενικές γραμμές όμως ο αλγόριθμος λειτούργησε πάρα πολύ καλά, μπόρεσε να εντοπίσει γρήγορα στην περιοχή συμφόρησης μετά την κατάρρευση των κόμβων και να βρει εναλλακτικά μονοπάτια διατηρώντας ψηλό το PDR.

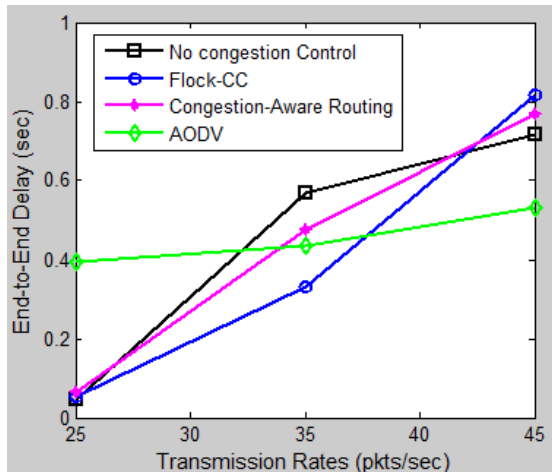
6.1.6 Σύγκριση αλγορίθμων

Χρησιμοποίησα τις καλύτερες τιμές για τις παραμέτρους μου ανά data rate οι οποίες ήταν: $A=0.4$, $E=0.5$, $C=0.25$, $T=0.5$.



Σχήμα 6.12: Αποτελέσματα PDR

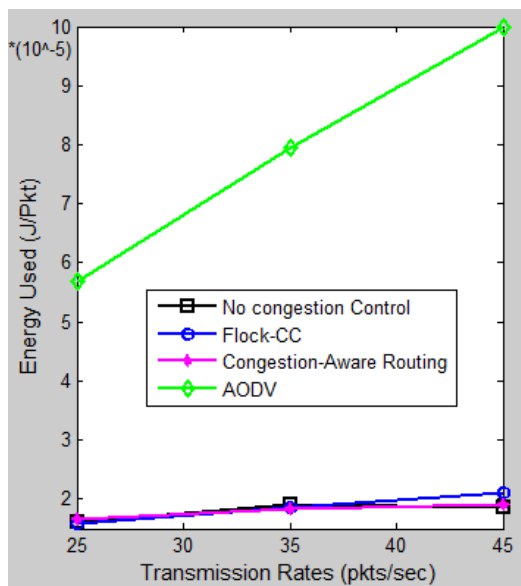
Πιο πάνω (σχήμα 6.12) παρουσιάζεται το PDR για τους τέσσερις αλγορίθμους για όλα τα data rates. Όπως μπορούμε να δούμε, ο Flocking αλγόριθμος που μελετάω έχει τα καλύτερα αποτελέσματα για όλα τα data rates. Πιο συγκεκριμένα, για rates ίσα με 25, 35 και 45pkts/sec το PDR ήταν ίσο με 95.98%, 82.40% και 65.41% αντίστοιχα. Το No Congestion Control protocol είχε PDR ίσο με 94%, 71.75% και 60.43% για την ίδια σειρά transmission rates. Το CAwR παρουσιάζει αποτελέσματα ίσα με 91.81%, 77.48%, 65.36%. Τέλος, ο AODV είχε το χαμηλότερο PDR και ήταν 65.48%, 43.71% και 21.06% για rates ίσα με 25, 35 και 45pkts/sec. Βλέπουμε πως η μεγαλύτερη διαφορά ανάμεσα στα τρία αυτά πρωτόκολλα είναι για rate=35pkts/sec όπου ο αλγόριθμος flocking υπερέχει με διαφορά πέραν του 5% από το congestion-aware routing protocol, πέραν του 10% από το πρωτόκολλο χωρίς έλεγχο συμφόρησης και πέραν του 52% για τον AODV.



Σχήμα 6.13: Αποτελέσματα καθυστερήσεων

Δίπλα στο σχήμα 6.13 παρουσιάζονται οι καθυστερήσεις. Βλέπουμε πως για rate ίσο με 25pkts/sec ο AODV χρειάστηκε 0.4sec ενώ όλοι οι υπόλοιποι αλγόριθμοι δίνουν χρόνο περίπου 0.05sec. Για data rate ίσο με 35pkts/sec ο No congestion control

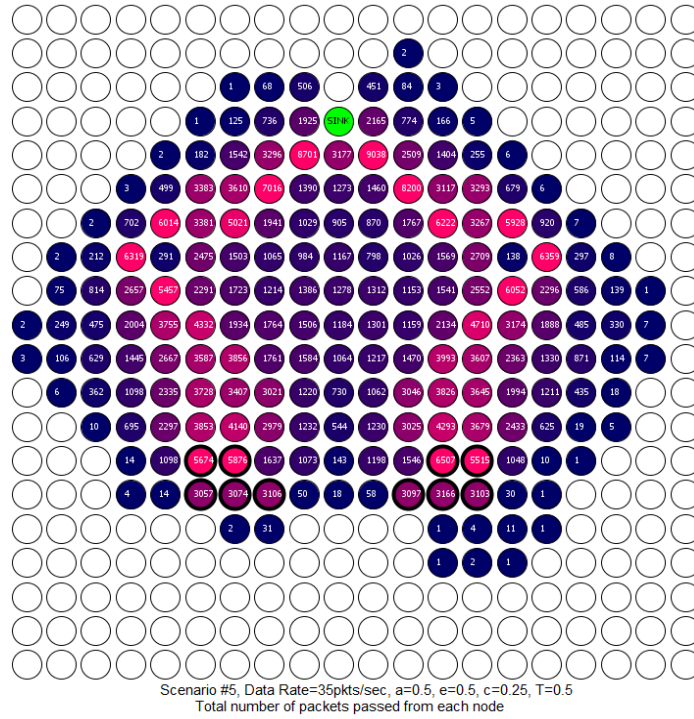
αλγόριθμος δίνει χρόνο 0.57sec, ο CAwR 0.48sec, ο Flock-CC 0.33sec ενώ ο AODV 0.44sec. Τέλος, για data rate ίσο με 45pkts/sec ο No Congestion χρειάζεται 0.72sec, ο CAwR 0.77sec, ο Flock-CC 0.82sec και τέλος ο AODV 0.53sec.



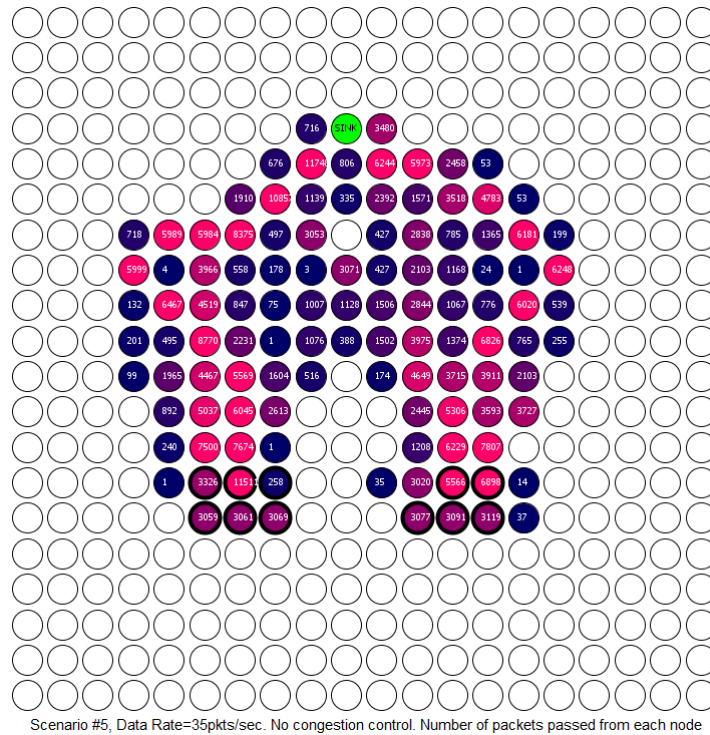
Σχήμα 6.14: Αποτελέσματα δαπανούμενης ενέργειας

Στο σχήμα 6.14 παρουσιάζεται το ποσό της ενέργειας που σπαταλήθηκε για το κάθε πακέτο. Παρατηρούμε πως τα αποτελέσματα όλων των αλγορίθμων πλην του AODV είναι περίπου τα ίδια και είναι ~1.5-1.9J/pkt. Όπως πάντα ο AODV φαίνεται να μην μπορεί να αποδώσει και όχι μόνο έχει αυξημένες καθυστερήσεις και χαμηλό PDR αλλά

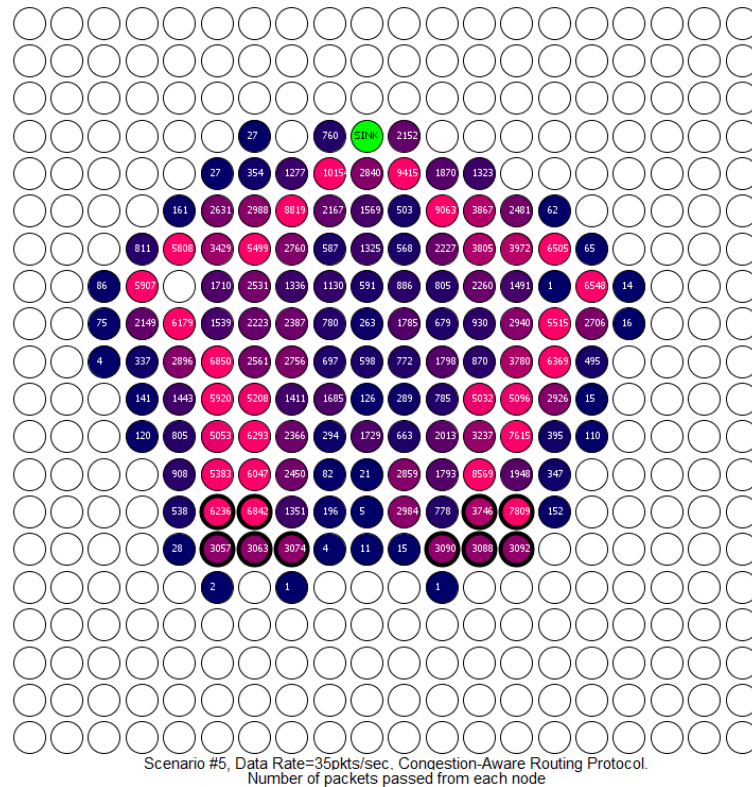
σπαταλά και μεγάλα ποσά ενέργειας για το κάθε πακέτο.



Σχήμα 6.15: Συνολικός αριθμός πακέτων που πέρασαν από τον κάθε κόμβο για τον αλγόριθμο Flock-CC.



Σχήμα 6.16: Συνολικός αριθμός πακέτων που πέρασαν από τον κάθε κόμβο για τον αλγόριθμο No Congestion Control.



Σχήμα 6.17: Συνολικός αριθμός πακέτων που πέρασαν από τον κάθε κόμβο για τον αλγόριθμο CAwR.

Στα σχήματα 6.15, 6.16 και 6.17 παρουσιάζεται ο συνολικός αριθμός πακέτων που έχουν περάσει από τον κάθε κόμβο στην περίπτωση που το data rate ήταν ίσο με 35pkts/sec. Οι τιμές των παραμέτρων για τον υπό εξέταση αλγόριθμο είναι: $\mathbf{a=0.5}$, $\mathbf{e=0.5}$, $\mathbf{c=0.25}$, $\mathbf{T=0.5}$. Στο σχήμα 6.15 φαίνεται η κίνηση των πακέτων για το πρωτόκολλο Flock-CC. Βλέπουμε μια ομοιόμορφη εξάπλωση των πακέτων προκειμένου να βρεθούν εναλλακτικά μονοπάτια και να γίνει όσο το δυνατό καλύτερη εκμετάλλευση του δικτύου. Στο σχήμα 6.16 φαίνεται η περίπτωση όπου δεν χρησιμοποιείται έλεγχος συμφόρησης και τα πακέτα προωθούνται στο συντομότερο μονοπάτι. Είναι εμφανής η διαφορά. Τα πακέτα στέλλονται μόνο από τα δεξιά κι αριστερά άκρα της σειράς των κόμβων που έχουν καταρρεύσει και όλα τα πακέτα αποστέλλονται μέσω δύο μονοπατιών τα οποία είναι χρωματισμένα με έντονο ροζ χρώμα. Τέλος, στο σχήμα 6.17 απεικονίζονται τα αποτελέσματα του CAwR. Έχουμε σαφώς καλύτερη κάλυψη σε σχέση με τον αλγόριθμο που δεν λαμβάνει υπόψη την συμφόρηση, όμως και πάλι υστερεί σε σχέση με τον υπό μελέτη αλγόριθμο.

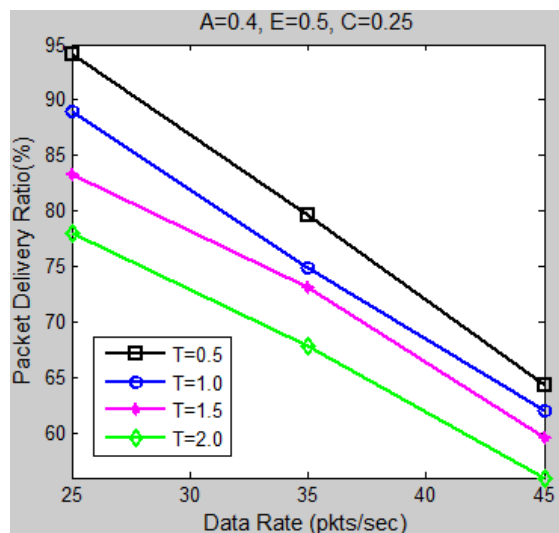
6.1.7 Σύνοψη

Αυτό το σενάριο είχε πάρα πολύ μεγάλο ενδιαφέρον αφού αποδεικνύει την ικανότητα του αλγορίθμου να αναπροσαρμόζεται έπειτα από εντοπισμό κάποιας προβληματικής περιοχής. Ακόμη και σε αυτή την περίπτωση, όπου μια μεγάλη σειρά κόμβων είχε καταρρεύσει, ο Flock-CC μπόρεσε να αντιληφθεί την αλλαγή και μετά από μικρή χρονική περίοδο προωθούσε τα πακέτα προς τους πλαϊνούς κόμβους αποφεύγοντας την ευθεία προς το sink. Είχε το μεγαλύτερο ποσοστό επιτυχημένης αποστολής από όλους τους αλγορίθμους αν και το CAwR πρωτόκολλο έδινε εξίσου καλά αποτελέσματα. Όπως και στις άλλες φορές ο χρόνος καθυστέρησης ήταν σχετικά μικρός, το ίδιο και το ποσό της ενέργειας που δαπανήθηκε. Σύμφωνα με τις προσδοκίες μας, σε αυτό το σενάριο καθοριστικό ρόλο έπαιξε η περίοδος δειγματοληψίας. Όσο πιο μικρή ήταν η τιμή τόσο καλύτερα ήταν και τα αποτελέσματα και μάλιστα με πάρα πολύ μεγάλη διαφορά. Χρειάστηκε να πάρει μικρή τιμή ώστε να μπορέσουν να ενημερωθούν πολύ γρήγορα οι κόμβοι για το πρόβλημα και το κοπάδι να μοιραστεί και να στέλλονται τα δεδομένα γύρω από την προβληματική περιοχή.

6.2 Κατάρρευση κόμβων δημιουργώντας καμπύλη

Σε αυτό το σενάριο έχουμε μια ομάδα κόμβων που δημιουργεί πακέτα και κατά την χρονική στιγμή των 40 δευτερολέπτων δύο σειρές κόμβων καταρρέουν δημιουργώντας μια καμπύλη κι εγκλωβίζοντας μέσα τα πακέτα. Ο λόγος που καταρρέουν δύο σειρές κόμβων είναι πως σε αντίθετη περίπτωση που θα κατάρρεε μια γραμμή μόνο, τα πακέτα θα μπορούσαν να προωθηθούν ενδιάμεσα από αυτούς τους κόμβους ακολουθώντας τις διαγώνιους κι έτσι το σενάριο δεν θα είχε τα αναμενόμενα αποτελέσματα.

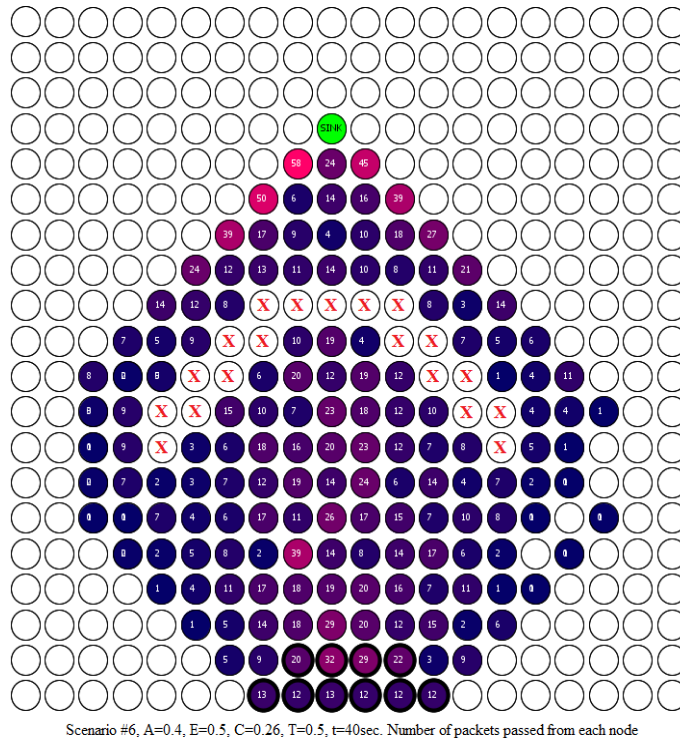
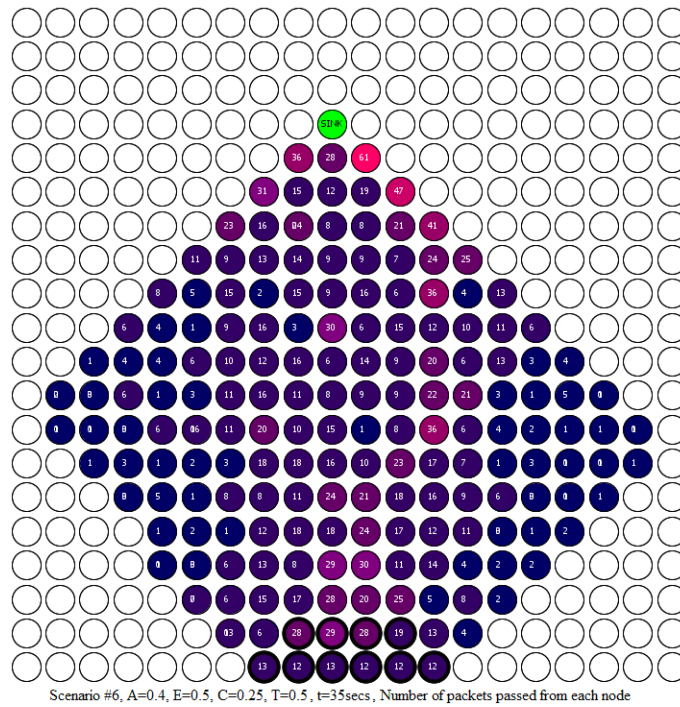
6.2.1 Γενικά αποτελέσματα



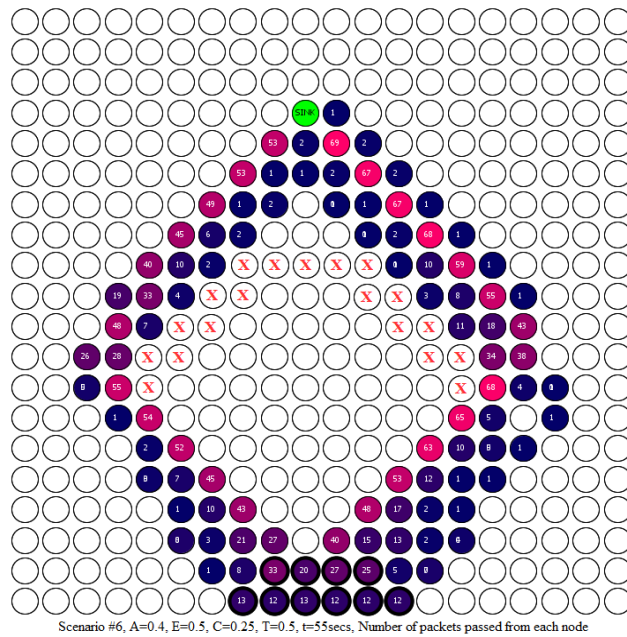
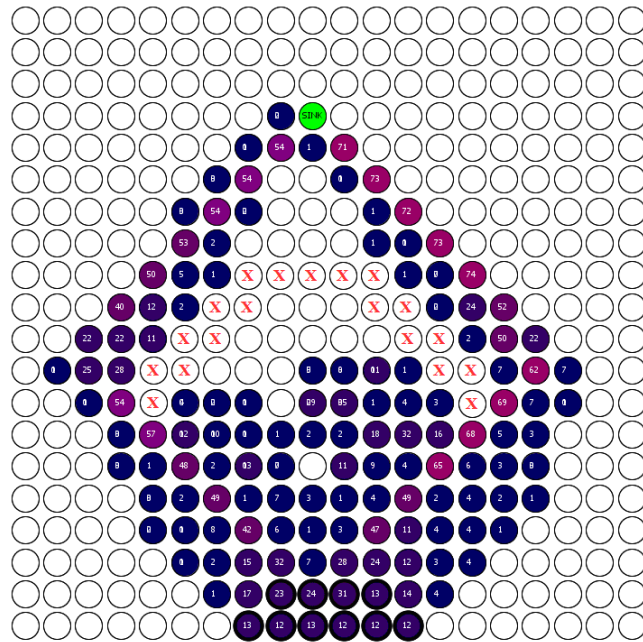
Σχήμα 6.18: Αποτελέσματα PDR

Στο σχήμα 6.18 παρουσιάζεται το PDR όταν οι παράμετροι a , e , και c διατηρούνται σταθερές σε 0.4, 0.5 και 0.25 αντίστοιχα. Βλέπουμε τα αποτελέσματα για τα τρία data rates διαφοροποιώντας κάθε φορά την περίοδο δειγματοληψίας. Αρχικά πρέπει να αναφερθεί πως αυτό το σενάριο είχε ελαφρώς χαμηλότερα αποτελέσματα (~2%) από το προηγούμενο όπου η σειρά των κόμβων που είχε καταρρεύσει ήταν σε ευθεία σειρά. Παρατηρούμε πως αυξάνοντας το T μειώνεται το PDR και η διαφορά ανάμεσα στα αποτελέσματα για T ίσο με 0.5 και 2.0 είναι πολύ μεγάλη, περίπου ~17%.

6.2.2 Γραφική απεικόνιση κίνησης πακέτων



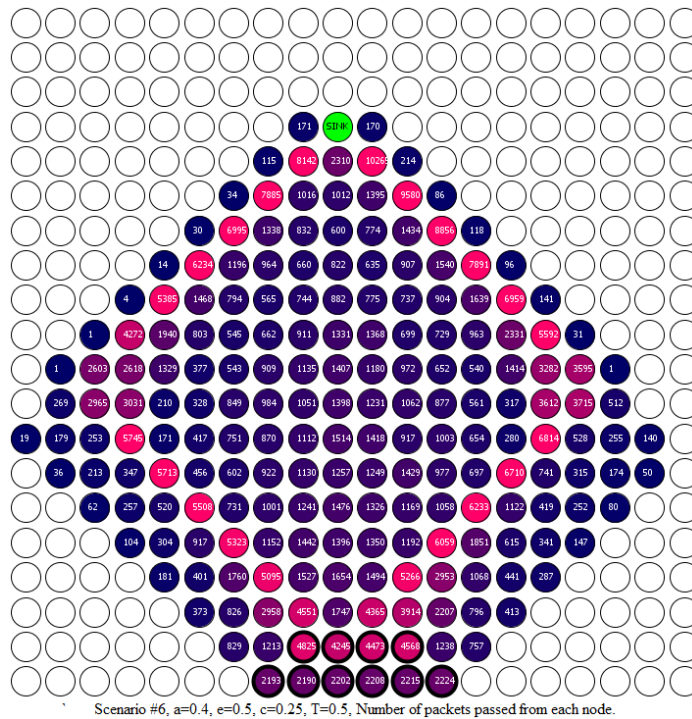
Σχήμα 6.19: Συνολικός αριθμός πακέτων που πέρασαν από τον κάθε κόμβο τις χρονικές στιγμές των 35sec και 40sec.



Σχήμα 6.20: Συνολικός αριθμός πακέτων που πέρασαν από τον κάθε κόμβο τις χρονικές στιγμές των 50sec και 55sec.

Στα σχήματα 6.19 και 6.20 παρατηρούμε τον συνολικό αριθμό πακέτων που έχουν περάσει από τον κάθε κόμβο για data rate ίσο με 25pkts/sec. Στην εικόνα 6.19(α) βλέπουμε πως οι κόμβοι δεν έχουν καταρρεύσει ακόμη κι έτσι τα πακέτα ακολουθούν τις ευθείες προς το sink ανοίγοντας όμως και προς τα πλάγια για να εκμεταλλευτούν ολόκληρο το δίκτυο. Ακολούθως, στα 40 δευτερόλεπτα, οι κόμβοι μόλις έχουν

καταρρεύσει και αρκετά πακέτα έχουν εγκλωβιστεί στην καμπύλη που δημιουργείται από την ομάδα των κόμβων που έχουν καταρρεύσει. Στην τρίτη εικόνα παρουσιάζεται το στιγμιότυπο για την χρονική στιγμή 50secs και παρατηρούμε πως ήδη έχει γίνει αντιληπτό το πρόβλημα και τα πακέτα έχουν σταματήσει να προωθούνται προς την προβληματική περιοχή. Τέλος, στην τελευταία εικόνα παρουσιάζεται το στιγμιότυπο για 55secs και όπως μπορούμε να δούμε οι κόμβοι πλέον έχουν ξεπεράσει το πρόβλημα πλήρως και προωθούν τα πακέτα όλα προς τα πλάγια.



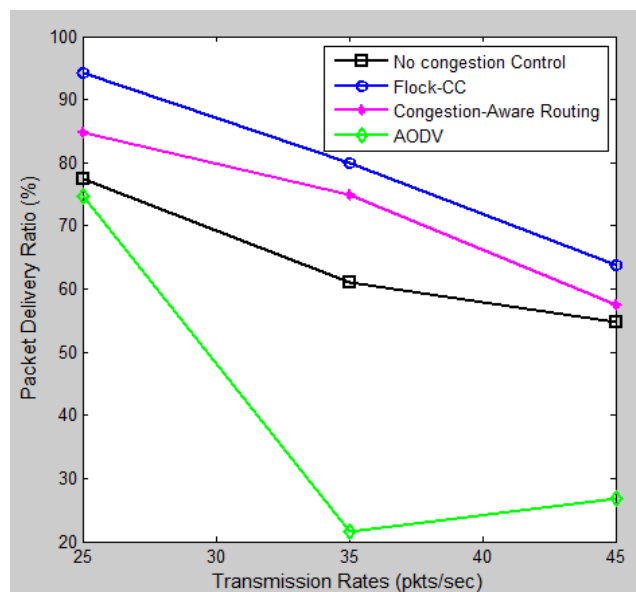
Σχήμα 6.21: Συνολικός αριθμός πακέτων που πέρασαν από τον κάθε κόμβο

Στο σχήμα 6.21 παρουσιάζεται ο συνολικός αριθμός πακέτων που έχουν περάσει από τον κάθε κόμβο. Όπως βλέπουμε ο αλγόριθμος εκμεταλλεύεται ολόκληρο το πλάτος του δικτύου και λειτουργεί αποδοτικά όπως και σε όλα τα προηγούμενα σενάρια. Παρατηρούμε πως υπάρχει συμφόρηση στις δύο κεντρικές διαγώνιους γύρω από την ομάδα των κόμβων που έχουν καταρρεύσει, όμως αυτό δεν αποτελεί πρόβλημα αφού συνολικά το ποσοστό επιτυχημένης αποστολής ήταν αρκετά ψηλό ακόμη και γι' αυτό το δύσκολο σενάριο.

6.2.3 Σύγκριση αλγορίθμων

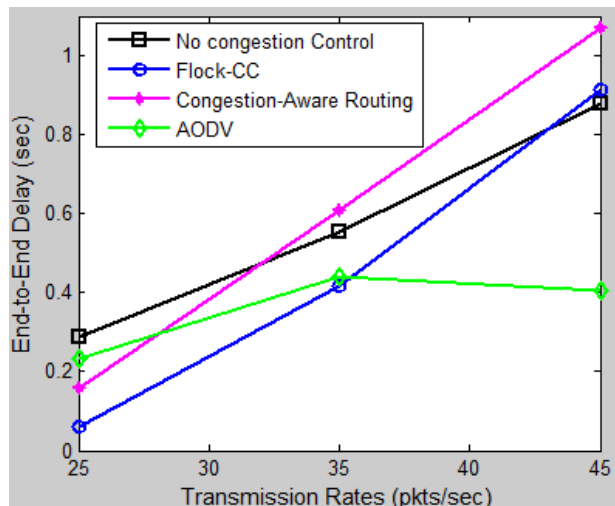
Τιμές Παραμέτρων αλγορίθμου Flock-CC: $A=0.4$, $E=0.5$, $C=0.25$, $T=0.5$

Τιμές για το Hello_Interval του αλγορίθμου AODV: 0.375 – 0.625. Για να μπορούν τα αποτελέσματα να είναι συγκρίσιμα με αυτά του Flock-CC, τα Hello messages πρέπει να στέλλονται σε κάποια χρονική στιγμή με μέση τιμή το 0.5, αλλά για σκοπούς βελτίωσης τα μηνύματα στέλλονται ανάμεσα στο διάστημα των $0.75 \cdot T$ και $1.25 \cdot T$ δευτερολέπτων.



Σχήμα 6.22: Αποτελέσματα PDR

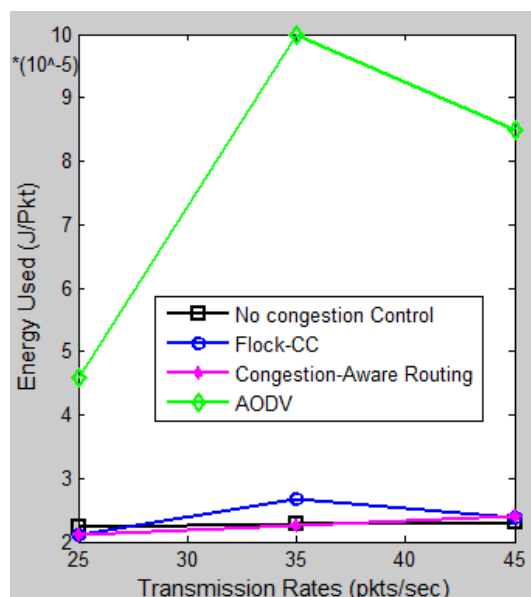
Στο σχήμα 6.22 παρουσιάζονται τα αποτελέσματα του PDR που δίνουν οι διάφοροι αλγόριθμοι για όλα τα data rates. Μπορούμε να διακρίνουμε την εμφανή διαφορά του Flock-CC αφού επιτυγχάνει το ψηλότερο PDR με διαφορά $\sim 10\%$ για όλα τα data rates συγκριτικά με τον δεύτερο καλύτερο CAwR. Ακολούθως, χειρότερα αποτελέσματα φαίνεται να έχει το No congestion control protocol με διαφορά περίπου $\sim 20\%$ για data rates ίσα με 25 και 35 pkts/sec και $\sim 10\%$ για 45pkts/sec από τον Flock-CC. Όσον αφορά τον AODV φαίνεται πως δεν μπορεί να ανταποκριθεί σε τέτοιου είδους σενάρια αφού δίνει υπερβολικά χαμηλά αποτελέσματα. Το παράξενο είναι πως για data rate ίσο με 45pkts/sec δίνει ελαφρώς καλύτερα αποτελέσματα παρά για 35pkts/sec.



Σχήμα 6.23: Αποτελέσματα καθυστερήσεων

Το σχήμα 6.23 αναπαριστά τις συνολικές καθυστερήσεις. Μπορούμε να δούμε πως ο αλγόριθμος Flock-CC δίνει τα καλύτερα αποτελέσματα. Για data rate ίσο με 25pkts/sec χρειάστηκε μόλις 0.06sec για PDR ίσο με 94% ενώ όλοι οι άλλοι

αλγόριθμοι χρειάζονται 0.16-0.3sec με PDR ~20% χαμηλότερο συγκριτικά με αυτό το Flock-CC. Εξίσου καλά αποτελέσματα είχαμε για data rate ίσο με 35pkts/sec. Για 45pkts/sec οι καθυστερήσεις είχαν αυξηθεί σε ~0.9sec για τον Flock-CC και τον No congestion, 1.07sec για τον CAwR και 0.405 για τον AODV.



Σχήμα 6.24: Αποτελέσματα δαπανούμενης ενέργειας ανά πακέτο

Στο διπλανό σχήμα απεικονίζεται η συνολική ενέργεια που δαπανείται για το κάθε πακέτο που καταφέρνει να αποσταλθεί επιτυχώς. Βλέπουμε πως οι αλγόριθμοι Flock-CC, CAwR και No Congestion έχουν εξίσου καλά αποτελέσματα για rates ίσα με 25 και 45 pkts/sec και ισούνται με $2.2 \cdot 10^{-5}$ και $2.4 \cdot 10^{-5}$ J/pkt αντίστοιχα. Για data rate

ίσο με 35pkts/sec ο Flock-CC έχει ελαφρώς χειρότερα αποτελέσματα ($2.86 \cdot 10^{-5}$ J/pkt), μηδαμινή διαφορά δεδομένου πως έχει ψηλότερο PDR. Τέλος, ο AODV έχει με μεγάλη διαφορά χειρότερα αποτελέσματα.

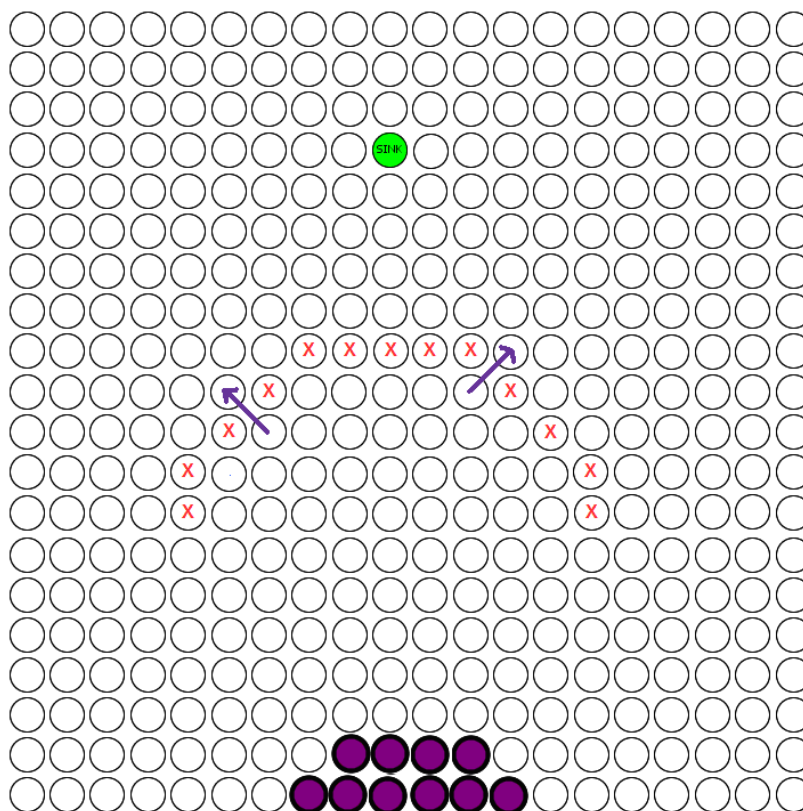
6.2.4 Σύνοψη

Σε αυτό το σενάριο μπορέσαμε να δούμε την ευρωστία του αλγορίθμου. Όπως φάνηκε ο αλγόριθμος κατάφερε σε σύντομο χρονικό διάστημα να αντιληφθεί το πρόβλημα της συμφόρησης και να απεγκλωβίσει τα πακέτα από την καμπύλη και να τα δρομολογεί έξω από την προβληματική περιοχή. Το ποσοστό του PDR ήταν ικανοποιητικά ψηλό και συγκρίνοντας τα αποτελέσματα με τους άλλους αλγορίθμους φάνηκε πως υπερτερεί και μάλιστα με διαφορά. Συνοψίζοντας, δίνει το ψηλότερο PDR, τις χαμηλότερες καθυστερήσεις και μια πολύ ικανοποιητική χρήση των ενεργειακών αποθεμάτων του δικτύου.

6.3 Αντιμετώπιση συμφόρησης διαπερνώντας τους πλαϊνούς νεκρούς κόμβους

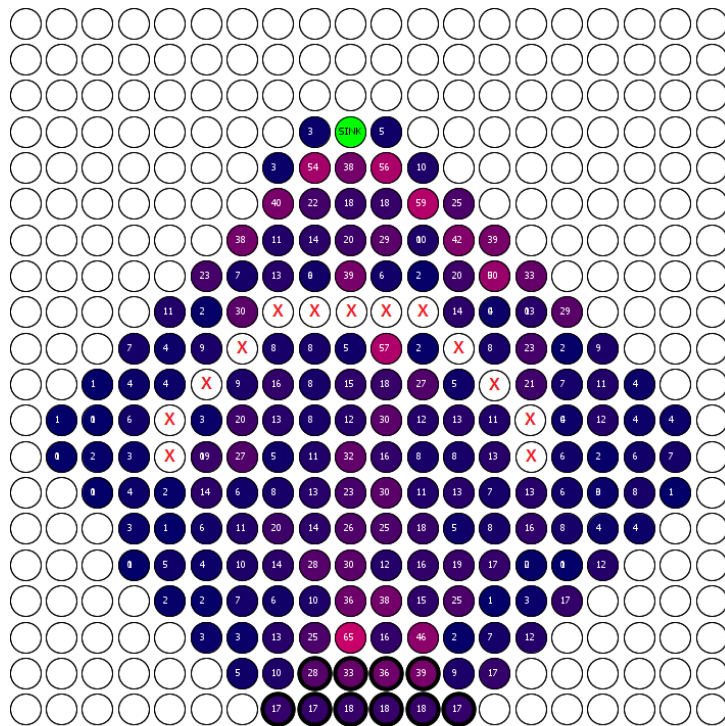
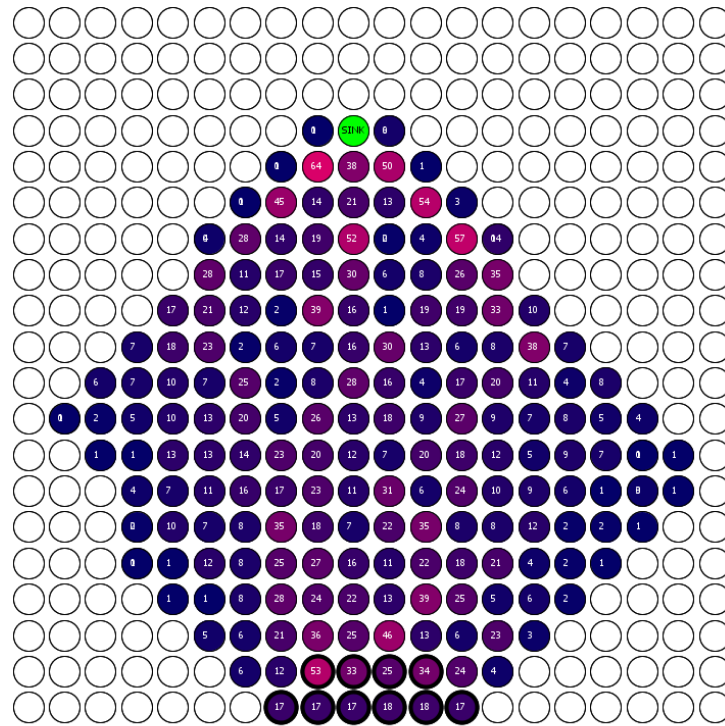
Αυτό το σενάριο αποτελεί επέκταση του προηγούμενου και στοχεύει στη παρουσίαση της ευρωστίας του αλγορίθμου. Όπως και προηγουμένως, κάποιοι κόμβοι καταρρέουν δημιουργώντας μια καμπύλη, όμως στην προκειμένη περίπτωση μόνο μια σειρά κόμβων καταρρέει στα πλάγια εν αντιθέσει με πριν που ήταν δύο οι σειρές. Σκοπός αυτής της αλλαγής είναι να φανεί πως όντως τα πακέτα θα προωθηθούν ακολουθώντας διαγώνια μονοπάτια (βλέπε σχήμα 6.25) και θα διαπεράσουν το πρόβλημα στους κόμβους που καταρρέουν μέσα από τα διάφορα κενά.

Λόγω έλλειψης χρόνου δεν έγινε μελέτη όλων των τιμών των παραμέτρων και χρησιμοποιήθηκαν οι τιμές: $T=0.5$, $a=0.4$, $e=0.5$, $c=0.25$ τις οποίες έκρινα ως τις ιδανικότερες για σενάρια όπου καταρρέουν κόμβοι.

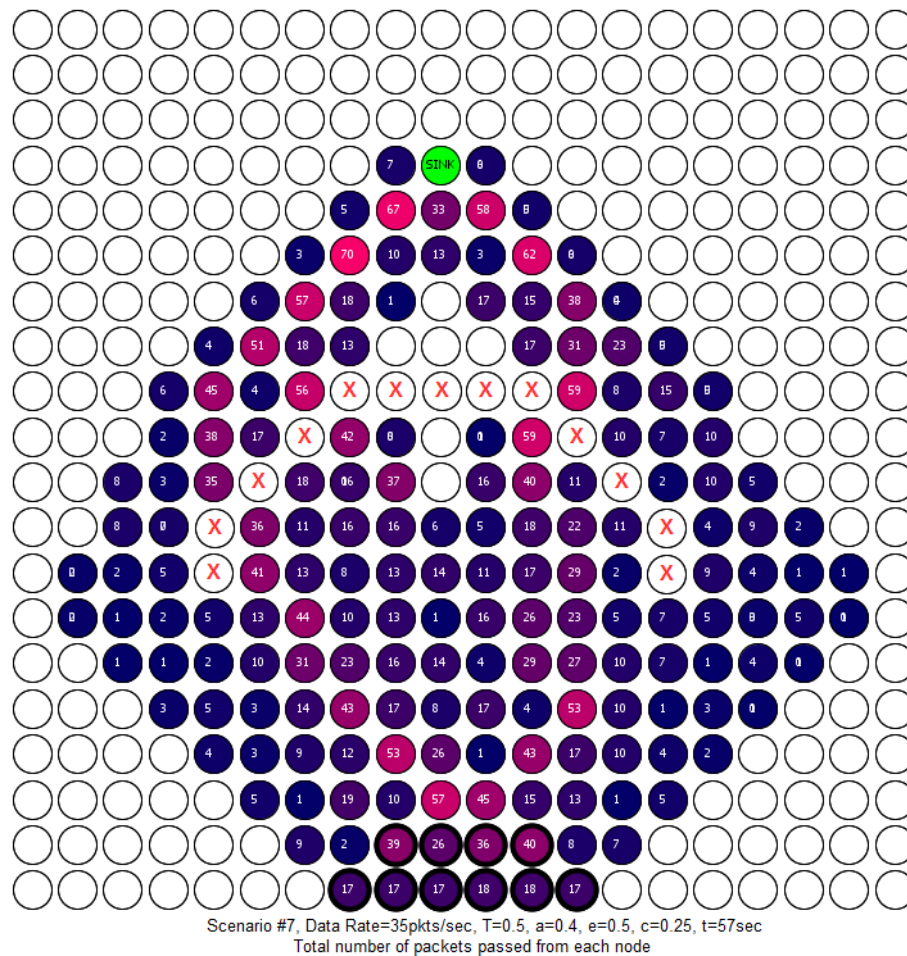


Σχήμα 6.25

6.3.1 Γραφική απεικόνιση κίνησης πακέτων

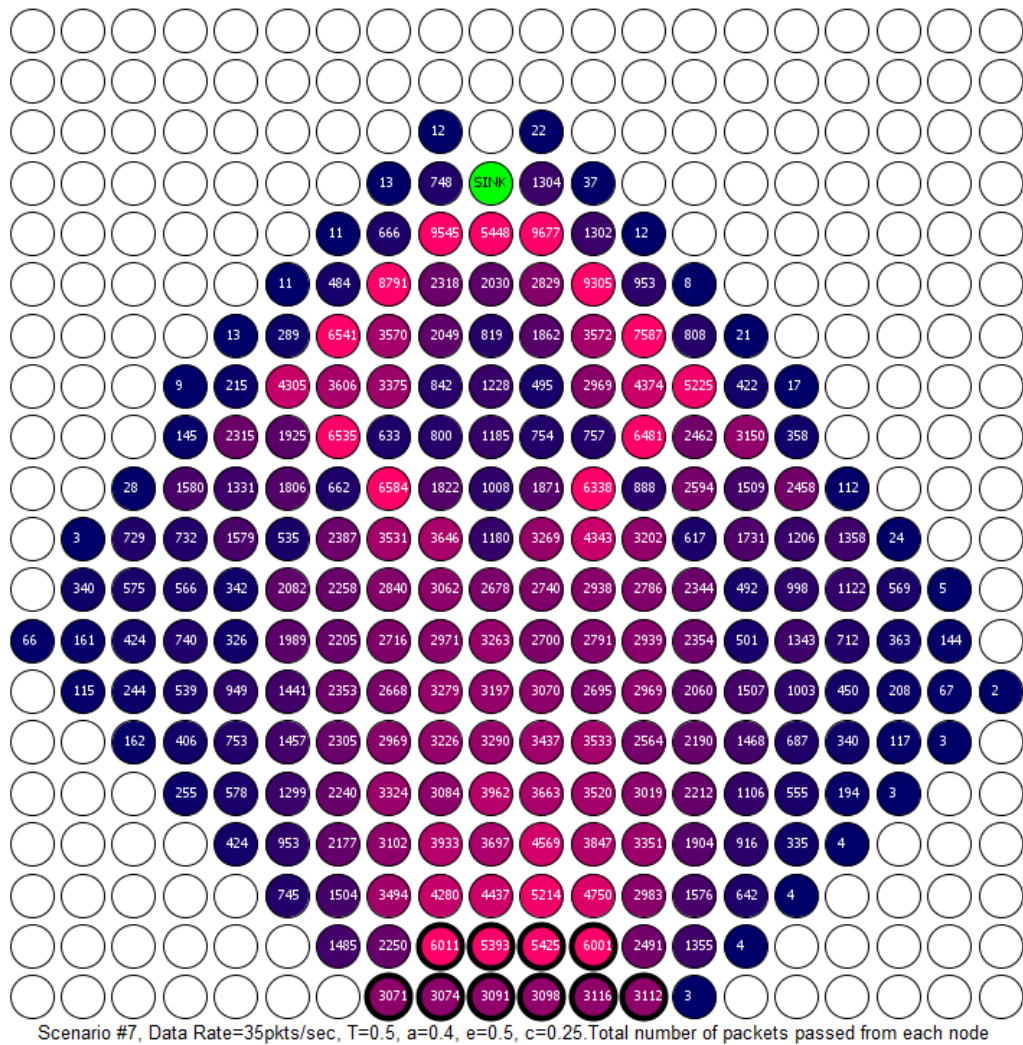


Σχήμα 6.26: Συνολικός αριθμός πακέτων που έχουν περάσει από τον κάθε κόμβο κατά τις χρονικές στιγμές των 30sec και 40sec.



Σχήμα 6.27: Συνολικός αριθμός πακέτων που έχουν περάσει από τον κάθε κόμβο κατά τις χρονικές στιγμές των 57sec.

Στο σχήμα 6.26 και 6.27 παρουσιάζεται ο συνολικός αριθμός πακέτων που έχουν περάσει από τον κάθε κόμβο κατά τις χρονικές στιγμές των 30, 40 και 57 δευτερολέπτων, δηλαδή πριν, κατά την διάρκεια και μετά την κατάρρευση των κόμβων. Όπως μπορούμε να διακρίνουμε τα πακέτα καταφέρνουν και ξεπερνούν το εμπόδιο στέλλοντας τα πακέτα διαγώνια ανάμεσα στους νεκρούς πλάγιους κόμβους (βλέπε σχήμα 6.27). Η μόνη περιοχή που φαίνεται να έχει αδειάσει είναι μπροστά από τους κόμβους που έχουν καταρρεύσει σε ευθεία γραμμή. Αυτό οφείλεται στο γεγονός πως τα πακέτα δεν μπορούν να διαπεράσουν τους κόμβους που έχουν καταρρεύσει.

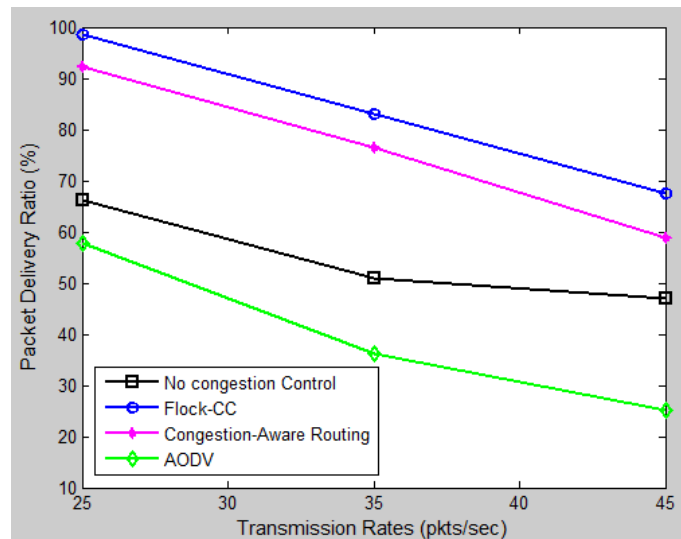


Σχήμα 6.28: Συνολικός αριθμός πακέτων που έχουν περάσει από τον κάθε κόμβο

Στο σχήμα 6.28 απεικονίζεται ο συνολικός αριθμός πακέτων που έχουν περάσει από τον κάθε κόμβο κατά την διάρκεια της προσομοίωσης. Παρατηρούμε πως η περιοχή συμφόρησης προκαλείται κοντά στο sink και πιο συγκεκριμένα στις διαγώνιους που ενώνουν το sink με τους δύο ακρινούς κόμβους της ευθείας γραμμής των κόμβων που έχουν καταρρεύσει. Επίσης, πρόβλημα φαίνεται να παρουσιάζεται και στην περιοχή όπου δημιουργούνται τα πακέτα. Γενικά όμως παρατηρούμε πως ο αλγόριθμος αντιμετωπίζει πολύ καλά την κατάσταση και το κοπάδι των πακέτων έχει ανοίξει πάρα πολύ, αφού υπάρχουν πακέτα που έχουν σταλεί πίσω από το sink και μέχρι και το δεξιό και αριστερό άκρο του δικτύου μοιράζοντας ομοιόμορφα το φορτίο δεδομένων.

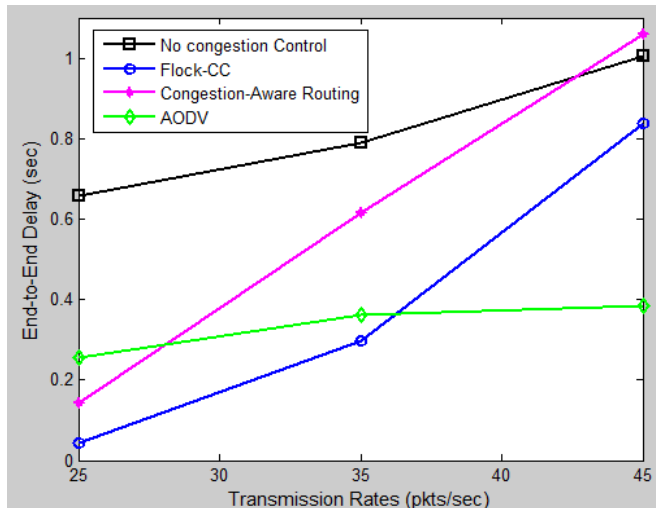
6.3.2 Σύγκριση αλγορίθμων

Τιμές παραμέτρων για τον αλγόριθμο Flock-CC: $T=0.5$, $a=0.4$, $e=0.5$, $c=0.25$



Σχήμα 6.29: Αποτελέσματα PDR

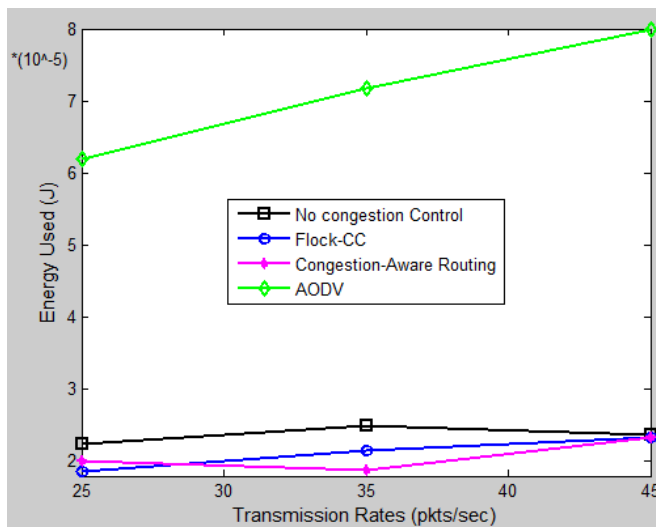
Στο σχήμα 6.29 παρουσιάζεται το ποσοστό επιτυχημένης αποστολής για όλα τα data rates που μελετήθηκαν για όλους τους αλγορίθμους. Παρατηρούμε πως ο Flock-CC δίνει γενικά τα καλύτερα αποτελέσματα με διαφορά περίπου 10% από το δεύτερο καλύτερο πρωτόκολλο CAwR. Συγκεκριμένα, για data rates ίσα με 25, 35 και 45 pkts/sec δίνει ποσοστά ίσα με 98.57%, 82.96% και 67.55% αντίστοιχα, εν αντιθέσει με τον CAwR που δίνει τιμές 92.29%, 76.32%, 58.87%. Το πρωτόκολλο No Congestion Control δίνει χαμηλότερα αποτελέσματα με μεγάλη διαφορά αφού το PDR ανέρχεται μόλις στα 66.22%, 50.79% και 47.02% για την ίδια ακολουθία data rates. Ο AODV φαίνεται να μην μπορεί να ανταποκριθεί αφού έχει τα χαμηλότερα αποτελέσματα και μάλιστα με διαφορά πέραν του 40% συγκριτικά με τον καλύτερο Flock-CC.



Σχήμα 6.30: Αποτελέσματα καθυστερήσεων

Στο σχήμα 6.30 παρουσιάζονται οι συνολικές καθυστερήσεις. Καταρχάς παρατηρούμε πως ο Flock-CC δίνει τις μικρότερες καθυστερήσεις σε σχέση με τους αλγορίθμους CAwR και No congestion Control οι οποίες

ανέρχονται σε 0.045, 0.3 και 0.84 δευτερόλεπτα για data rates ίσα με 25, 35 και 45 pkts/sec αντίστοιχα. Ο AODV δίνει τιμές ίσες με 0.255, 0.361 και 0.385 δευτερόλεπτα. Μπορεί να δίνει καλύτερες καθυστερήσεις για data rates ίσα με 45pkts/sec όμως το ποσοστό επιτυχημένης αποστολής του είναι τόσο χαμηλό που η μικροσκοπική αυτή διαφορά εκμηδενίζεται.



Σχήμα 6.31: Αποτελέσματα δαπανώμενης ενέργειας ανά πακέτο

Στο σχήμα 6.31 παρουσιάζονται η συνολική ενέργεια που σπαταλήθηκε ανά πακέτο. Καταρχάς παρατηρούμε πως ο αλγόριθμος AODV απαιτεί την μεγαλύτερη ενεργειακή

κατανάλωση ανά πακέτο και μάλιστα χρειάζεται σχεδόν τριπλάσιο ποσό σε σχέση με τους άλλους αλγορίθμους. Για data rates ίσα με 25 και 45 pkts/sec παρατηρούμε πως ο Flock-CC δίνει την καλύτερη ενεργειακή κατανάλωση. Για data rate ίσο με 35pkts/sec φαίνεται πως ο CAwR δίνει ελαφρώς καλύτερα αποτελέσματα από τον Flock-CC, όμως η διαφορά είναι ασήμαντη και ειδικά λαμβάνοντας υπόψη το γεγονός πως το PDR είναι περίπου 10% χαμηλότερο από τον δεύτερο.

6.3.3 Σύνοψη

Σε αυτό το σενάριο παρατηρήσαμε τον τρόπο λειτουργίας του αλγορίθμου σε ένα σενάριο που αποτελεί επέκταση της περίπτωσης που παρουσιάζεται στο υποκεφάλαιο 6.2. Όπως φάνηκε ο αλγόριθμος μπορεί να λειτουργήσει πολύ αποδοτικά σε αντίθεση με όλους τους άλλους αλγορίθμους. Συγκεκριμένα, μετά την κατάρρευση των κόμβων ο Flock-CC κατάφερε να στείλει τα πακέτα διαγώνια διαπερνώντας με ευκολία το εμπόδιο που παρουσιάστηκε και αξιοποίησε μεγάλο μέρος του δικτύου όπως πάντα. Οι άλλοι αλγόριθμοι όμως δεν αντέδρασαν με τον ίδιο τρόπο και έστελλαν τα πακέτα τους μόνο από τα συντομότερα μονοπάτια αυξάνοντας κατά πολύ την συμφόρηση και σπαταλώντας μεγαλύτερα ποσά ενέργειας από κάποιους κόμβους.

Κεφάλαιο 7

Συμπεράσματα

7.1	Συνολικά Συμπεράσματα	121
7.2	Μελλοντική Εργασία.....	124

7.1 Συνολικά Συμπεράσματα

Μέσα από αυτή την διπλωματική εργασία μελέτησα ένα νέο αλγόριθμο ελέγχου συμφόρησης για τα δίκτυα αισθητήρων, ο οποίος μιμείται την συμπεριφορά των πουλιών καθώς κινούνται σε κοπάδια. Στόχος μου ήταν ο προσδιορισμός των παραμέτρων που συμβάλλουν στον εντοπισμό της συμφόρησης και της τιμής τους. Μέσα από κάποια σενάρια, τα οποία παρουσιάστηκαν προηγουμένως στα κεφάλαια 5 και 6, κατάφερα να παρουσιάσω τα πλεονεκτήματα και τις ιδιότητες του αλγορίθμου και να κάνω και μια σύγκριση με κάποιους άλλους. Το κάθε ένα από τα σενάρια βοήθησε στον έλεγχο της συμπεριφοράς του αλγορίθμου σε διάφορες καταστάσεις, όταν υπάρχουν πολλά εναλλακτικά μονοπάτια, όταν υπάρχει ελάχιστος αριθμός από συντομότερα δρομολόγια, όταν προκύψει κάποιο πρόβλημα στο δίκτυο και καταρρεύσουν κάποιοι κόμβοι προκαλώντας συμφόρηση σε μια περιοχή. Μέσα από την κάθε περίπτωση έγινε καλύτερη κατανόηση των διαφόρων παραμέτρων και της χρησιμότητάς τους.

Μετά από πολύ σκληρή μελέτη όλων των σεναρίων που παρουσίασα προηγουμένως, κατέληξα στις ιδανικές τιμές για τις παραμέτρους που μελετούσα. Πρέπει να αναφέρω πως προκειμένου να αποφανθώ ποιες ήταν οι ιδανικότερες τιμές χρειάστηκε να περάσω από διάφορα επίπεδα αποκλείοντας σταδιακά κάποιες από αυτές και στηριζόμενη και λίγο στα αποτελέσματα των προηγούμενων σεναρίων μπόρεσα τελικά να προσδιορίσω

τις καλύτερες τιμές οι οποίες είναι ικανοποιητικές για όλα τα data rates και όλα τα σενάρια. Αξίζει να σημειωθεί πως φαινομενικά υπήρχαν κάποιες άλλες τιμές οι οποίες μπορεί να έδιναν καλά αποτελέσματα και ίσως και καλύτερα σε ορισμένες περιπτώσεις, όμως η διαφορά ήταν πάντοτε απειροελάχιστη. Στηριζόμενη σε αυτό προσπάθησα να βρω τις τιμές που για όλες τις πιθανές περιπτώσεις δίνουν τουλάχιστον ένα αξιοπρεπές ποσοστό επιτυχημένης αποστολής κάτι το οποίο κατάφερα με επιτυχία.

Συνολικά καλύτερες τιμές:

E=0.5

C=0.25

A=0.4-0.5

**T=1.5 για τα σενάρια που δεν παρουσιάζουν
ιδιαίτερο πρόβλημα συμφόρησης και 0.5 σε
αντίθετη περίπτωση.**

Σχετικά με την παράμετρο δειγματοληψίας παρατηρήθηκε πως όταν το δίκτυο υποφέρει από προβλήματα συμφόρησης είναι καλύτερα να παίρνει μικρή τιμή. Το ίδιο ισχύει και στις περιπτώσεις όπου προκύπτει κάποια αλλαγή στο περιβάλλον έπειτα από κατάρρευση κόμβων. Με αυτό τον τρόπο οι κόμβοι ενημερώνονται έγκαιρα για τις αλλαγές και προσπαθούν να βρουν εναλλακτικά μονοπάτια. Σε αντίθετη περίπτωση όπου το δίκτυο είναι υγιές μπορεί να χρησιμοποιηθεί μια κάπως μεγαλύτερη τιμή αφού με αυτό τον τρόπο εξοικονομείται περισσότερη ενέργεια.

Η παράμετρος e έδειξε πως δεν μπορεί να είναι μηδενική αλλά ούτε και πάρα πολύ μεγάλη. Αυτή η παράμετρος παίζει καθοριστικό ρόλο στο ποιοι κόμβοι θα συμπεριλαμβάνονται στο πεδίο αντίληψης και συνεπώς συμβάλλει στο πόσο θα ανοίξει το κοπάδι. Είδαμε πως όταν παίρνει μεγάλες τιμές το PDR μειώνεται διότι τα πακέτα καταλήγουν σε κύκλους ανάμεσα σε κόμβους που δεν παρουσιάζουν προβλήματα

συμφόρησης. Όταν είναι μηδενική, το κοπάδι πακέτων είναι πολύ στενό, και δεν γίνεται μεγάλη εκμετάλλευση του δικτύου, με όλα τα αρνητικά επακόλουθα.

Πολλές μελέτες έδειξαν πως η εισαγωγή του τυχαίου στοιχείου σε ένα αλγόριθμο μπορεί να επιφέρει καλύτερα αποτελέσματα, κάτι το οποίο φάνηκε επίσης μέσα από την έρευνα μου. Η τυχαιότητα (randomness) πρέπει να παίρνει μικρή τιμή, αλλά όχι μηδενική. Αυτό βοηθά στην επιλογή του επόμενου κόμβου με κάπως τυχαίο τρόπο και συνεπώς συμβάλλει στην καλύτερη κατανομή του traffic αφού τα πακέτα θα επιλέγουν διαφορετικά μονοπάτια.

Ίσως το πιο δύσκολο σημείο της διπλωματικής μου εργασίας, ήταν ο προσδιορισμός της παραμέτρου α . Από την πρώτη στιγμή φάνηκε πως τόσο η κατάσταση στις ουρές ήταν σημαντική όσο και η κατάσταση του ασύρματου μέσου. Το δύσκολο μέρος ήταν να οριστεί πόσο σημαντικό ήταν το κάθε ένα από αυτά τα στοιχεία, δηλαδή που πρέπει να δίνεται περισσότερη βαρύτητα. Στα διάφορα σενάρια που κλήθηκα να μελετήσω βρήκα διάφορα αποτελέσματα τα οποία με βοήθησαν στο να περιορίσω το πεδίο τιμών που μπορεί να πάρει η παράμετρος (0.2 – 0.6). Κάνοντας όμως μια σύγκριση όλων των σεναρίων, όλων των data rates βασιζόμενη στο συγκεκριμένο πεδίο τιμών κατάφερα να περιορίσω ακόμη περισσότερο το εύρος σε 0.4 και 0.5. Βλέπουμε πως αφού η παράμετρος α ισούται με 0.4 ή 0.5, συμπεραίνουμε πως η κατάσταση του ασύρματου μέσου και η κατάσταση στις ουρές των κόμβων παίζουν εξίσου σημαντικό ρόλο στον εντοπισμό της συμφόρησης.

Μέσα από τις διάφορες συγκρίσεις με του άλλους αλγορίθμους φάνηκε πως ο Flock-CC δίνει το μεγαλύτερο PDR, σχετικά μικρές καθυστερήσεις και ίσως τα καλύτερα αποτελέσματα για την σπατάλη ενέργειας. Παρουσιάζοντας τις γραφικές αναπαραστάσεις φάνηκε πόσο αποδοτικός και πόσο εύρωστος είναι ο αλγόριθμος αφού όποτε αντιλαμβανόταν πως μπροστά υπήρχε περιοχή με πρόβλημα, άνοιγε προς τα πλάγια προκειμένου να την αποφύγει, κάτι το οποίο δεν έκαναν οι άλλοι αλγόριθμοι. Τέλος, σε κάποια από τα σενάρια παρουσίασα και τον συνολικό αριθμό πακέτων που πέρασαν από τον κάθε κόμβο για όλους τους αλγορίθμους και είδαμε πως κανένας τους

δεν εκμεταλλευόταν σε τόσο μεγάλο βαθμό το δίκτυο όσο ο Flock-CC κι έτσι το πρόβλημα της συμμόρφωσης φαινόταν πολύ πιο έντονο.

Συνοψίζοντας, ο σκοπός αυτής της διπλωματικής εργασίας και η προσέγγιση που ακολουθήσαμε επέφερε το ζητούμενο αποτέλεσμα. Καταλήξαμε σε ένα πάρα πολύ καλό συνδυασμό παραμέτρων που μπορεί να εντοπίσει την συμμόρφωση και μάλιστα να την αποφύγει. Σίγουρα ο Flock-CC είναι ένας καλός αλγόριθμος, πολύ υποσχόμενος γύρω από τον οποίο θα μπορούσε να διεξαχθεί περισσότερη έρευνα.

7.2 Μελλοντική Εργασία

Μέσα από αυτή την διπλωματική εργασία εστίασαμε την αξιολόγηση του αλγορίθμου σε σταθερή τετράγωνη τοπολογία και στο μέλλον θα ήταν καλά να γίνει μελέτη του αλγορίθμου για τυχαίες τοπολογίες. Επίσης, οι παράμετροι (a , e , c , T) θα μπορούσαν να διερευνηθούν και σε διαφορετικά σενάρια για να φανεί κατά πόσον οι τιμές που όρισα δίνουν ικανοποιητικά αποτελέσματα ή αν υπάρχει κάποιος καλύτερος συνδυασμός. Επιπλέον, θα μπορούσαν να εφαρμοστούν και διαφορετικές τεχνικές όσον αφορά την εξεύρεση της κατεύθυνσης του sink και αντί της κανονικής Γκαουσιανής διανομής θα μπορούσε να χρησιμοποιηθεί κάποια άλλη συνάρτηση που θα δίνει ακόμη καλύτερα αποτελέσματα. Τέλος, θα ήταν καλό να γίνει σύγκριση με κάποιους άλλους αλγορίθμους που είναι εξειδικευμένοι για τα δίκτυα αισθητήρων (π.χ. AntHocNet ή ο AntNet [7]).

Βιβλιογραφία

- [1] Akyildiz Ian, Su Weilian, Sankarasubramaniam Yogesh, and Cayirci Erdal Georgia Institute of Technology, “A survey on Sensor Networks”, IEEE Communications magazine, August 2002.
- [2] Antoniou Pavlos, Pitsillides Andreas, Blackwell Tim, Engelbrecht Andries, Michael Loizos, “Congestion Control in Wireless Sensor Networks based on the Bird Flocking Behavior”, Journal on selected areas in communications, Vol. X, No. X, March 2009.
- [3] Antoniou Pavlos and Pitsillides Andreas, “Wireless Sensor Network Control: Drawing Inspiration from Complex Systems”, Poster paper, Corfu, June Tutorial & PhD Poster Session, Med-Hoc-Net 2007, Ionian Academy, Corfu, 12 - 15 June 2007.
- [4] Antoniou Pavlos, Pitsillides Andreas, Blackwell Tim, Engelbrecht Andries, “Employing the Flocking Behavior of Birds for Controlling Congestion in Autonomous Decentralized Networks”, IEEE Press Piscataway, NJ, USA, 2009, pp. 1753-1761.
- [5] C. Perkins, E. Royer and S. Das, “Ad hoc On-demand Distance Vector (AODV) Routing”, RFC 3561, Copyright © The Internet Society, July 2003.
- [6] Chonggang Wang and Kazem Sohraby, University of Arkansas Bo Li, The Hong Kong University of Science and Technology, Mahmoud Daneshmand, AT&T Labs Research Yueming Hu, South China Agricultural University, “A Survey of Transport Protocols for Wireless Sensor Networks”, IEEE Network, Vol. 20, May/June 2006, pp. 34-40.

- [7] Falko Dressler, Ozgur B. Akan, “A Survey on Bio-inspired Networking”, Elsevier Computer Networks 54(6) (2010), pp. 881-900.
- [8] Kumar R., Rowaihy H., Cao G., Anjum F., Yener A. and La Porta T., “Congestion Aware Routing in Sensor Networks”, Technical Report NAS-TR-0036-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, 2006.
- [9] “MiND2C – Mimicking Nature for Designing Robust Congestion Control Mechanisms in Self-Organized Autonomous Decentralized Networks”, Deliverable D7, May 31, 2009.
- [10] Meisel Michael, Pappas Vasileios, Zhang Lixia, “A taxonomy of biologically inspired research in computer networking”, Computer Networks 54 (2010), 16 December 2009, pp. 901-916.