



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μηχανισμοί εξοικονόμησης ενέργειας σε
ασύρματα δίκτυα με χρήση μετρικών ποιότητας
σήματος

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Στα πλαίσια του Μεταπτυχιακού Διπλώματος

Ειδίκευσης (ΜΔΕ)

‘Επιστήμη και Τεχνολογία των Υπολογιστών’

του

ΝΙΚΟΛΑΟΥ ΣΤΑΘΟΠΟΥΛΟΥ

Επιβλέπων: Χ. Ι. Μπούρας
Καθηγητής ΤΜΗΥΠ

Πάτρα, Ιούλιος 2016



Πανεπιστήμιο Πατρών
Τμήμα Μηχανικών Η/Υ και Πληροφορικής

Μηχανισμοί εξοικονόμησης ενέργειας σε
ασύρματα δίκτυα με χρήση μετρικών ποιότητας
σήματος

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Στα πλαίσια του Μεταπτυχιακού Διπλώματος
Ειδίκευσης (ΜΔΕ)

‘Επιστήμη και Τεχνολογία των Υπολογιστών’

του

ΝΙΚΟΛΑΟΥ ΣΤΑΘΟΠΟΥΛΟΥ

Επιβλέπων: Χ. Ι. Μπούρας

Καθηγητής ΤΜΗΥΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....

Χ. Ι. Μπούρας
Καθηγητής ΤΜΗΥΠ

.....

Ιωάννης Γαροφαλάκης
Καθηγητής ΤΜΗΥΠ

.....

Κωνσταντίνος Μπερμπερίδης
Καθηγητής ΤΜΗΥΠ

Πάτρα, Ιούλιος 2016

ΝΙΚΟΛΑΟΣ ΣΤΑΘΟΠΟΥΛΟΣ

Διπλωματούχος Μηχανικός Η/Υ και Πληροφορικής Πανεπιστημίου Πατρών

© 2016 – All rights reserved

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Μπούρα για την καθοδήγηση και την στήριξη που μου παρείχε.

Επίσης ευχαριστώ ιδιαίτερα τους κ. Καπούλα και Στάμο για τις εποικοδομητικές συζητήσεις που με βοήθησαν στην εκπόνηση της διπλωματικής μου εργασίας.

Τέλος θα ήθελα να ευχαριστήσω την οικογένειά μου, για όλα τα χρόνια που με στήριξαν ψυχολογικά και με ενθάρρυναν να ακολουθήσω αυτό το μονοπάτι.

Περίληψη

Τα ασύρματα δίκτυα αποτελούν ένα σημαντικό κομμάτι την επικοινωνίας μεταξύ συσκευών και χρηστών. Η εξέλιξη των κινητών συσκευών έχει συμβάλει στην εγκαθίδρυση των ασύρματων δικτύων ως το βασικό μέσο για ανταλλαγή δεδομένων και πρόσβαση στο Internet. Παράλληλα, το μέγεθος των δεδομένων που ανταλλάσσονται μέσω των δικτύων τείνει να αυξάνεται, καθώς οι συσκευές στηρίζονται κατά ένα μεγάλο βαθμό στην πρόσβαση στο διαδίκτυο για να επιτελέσουν ορισμένες λειτουργίες. Αυτό έχει ως συνέπεια την αύξηση της κατανάλωσης ενέργειας, γεγονός το οποίο μειώνει τον χρόνο λειτουργίας των συσκευών, αφού η τεχνολογική εξέλιξη στα θέματα ενέργειας (κινητές πηγές ενέργειας) είναι περιορισμένη. Έτσι, θεωρείται πολύ σημαντική η έρευνα για την εύρεση τεχνικών και βελτιώσεων οι οποίες συμβάλλουν στην εξοικονόμηση ενέργειας στον τομέα της ασύρματης δικτύωσης.

Αντικείμενο της εργασίας αυτής είναι η ανάπτυξη και υλοποίηση μηχανισμών για την διαχείριση της ενέργειας μετάδοσης πακέτων σε ασύρματα δίκτυα, με απώτερο σκοπό την εξοικονόμηση ενέργειας στις κινητές συσκευές. Η βασική ιδέα της εργασίας είναι το γεγονός ότι σε ένα μεγάλο μέρος των μεταδόσεων που λαμβάνουν μέρος σε ένα ασύρματο δίκτυο, απαιτούν λιγότερη ενέργεια για την επιτυχής μετάδοση στον προορισμό από αυτή που σπαταλάται συνήθως, με άμεση συνέπεια την μείωση του χρόνου λειτουργίας των συσκευών.

Ο μηχανισμός που προτείνεται βασίζεται στην αξιοποίηση απλών μετρικών που χρησιμοποιούνται για την εκτίμηση της ποιότητας του καναλιού, όπως ο λόγος σήματος προς θόρυβο (SNR). Οι μηχανισμοί που παρουσιάζονται εφαρμόζονται τόσο σε κεντρικοποιημένα δίκτυα, όσο και σε μη κεντρικοποιημένα (ad hoc), ακολουθώντας διαφορετική τακτική.

Στα κεντρικοποιημένα δίκτυα, ο μηχανισμός υποστηρίζεται από ένα απλό σύστημα ανάδρασης μεταξύ του access point και των συνδεδεμένων κόμβων, με σκοπό την απόκτηση πληροφοριών σχετικά με την μετάδοση. Το access point στέλνει μηνύματα στους συνδεδεμένους κόμβους μέσω των οποίων παρέχει πληροφορίες που χρησιμοποιούνται από αυτούς, προκειμένου να προσαρμόσουν την ισχύ μετάδοσης. Στα μη κεντρικοποιημένα, ο μηχανισμός εκμεταλλεύεται τα πρωτόκολλα δρομολόγησης για την μεταφορά της πληροφορίας, προς αποφυγή επιβάρυνσης του δικτύου με επιπλέον μηνύματα. Αντικείμενο πειράματος είναι το πρωτόκολλο δρομολόγησης OLSR, στο οποίο γίνονται ορισμένες αλλαγές για γίνει δυνατή η χρήση του μηχανισμού.

Για την αξιολόγηση των μηχανισμών, έγιναν πειράματα με την χρήση του εξομοιωτή Network Simulator 3 (NS-3). Στα πειράματα υλοποιήθηκαν οι προαναφερθείσες περιπτώσεις πάνω σε δίκτυα κινητών κόμβων, με διαφορετικές παραμέτρους για τον μηχανισμό ή διαφορετικό

πλήθος κόμβων. Τα αποτελέσματα έδειξαν ότι επιτυγχάνεται η εξοικονόμηση ενέργειας σε μεγάλο βαθμό, επιμηκύνοντας τον χρόνο λειτουργίας των συσκευών, ενώ την ίδια στιγμή η ποιότητα του δικτύου δεν υποβαθμίζεται σε τέτοιο βαθμό ώστε να μην είναι αποδοτικό.

Στα πλαίσια της εργασίας αυτής έγιναν οι παρακάτω δημοσιεύσεις:

1. C. Bouras, V. Kapoulas, K. Stamos, N. Stathopoulos, N. Tavoularis, “Power Management and Rate Control Mechanism for Wi-Fi Infrastructure Networks”, 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), pp 731 - 736, 26-28 October 2015, Liverpool, UK
2. C. Bouras, V. Kapoulas, K. Stamos, N. Stathopoulos, “Evaluation of Power Control Mechanism on OLSR Routing Protocol”, 11th International Workshop on the Performance Analysis and Enhancement of Wireless Networks, 23-25 March 2016, Crans-Montana, Switzerland

Λέξεις Κλειδιά

Εξοικονόμηση ενέργειας, λόγος σήματος προς θόρυβο, παρεμβολή, ad hoc, δρομολόγηση.

Abstract

Nowadays, wireless networks play an important role in device communication. The evolution of mobile devices has contributed to the establishment of wireless networks as the most basic means to data exchange and Internet access. Meanwhile, the amount of data exchanged through networks tends to increase, since the devices depend greatly on Internet access to execute various tasks. This fact leads to increased energy consumption, which in turn leads to the reduction of devices' lifespan, because technological advancements in energy issues (mobile energy sources) are made at a slow pace. Thus, research for inventing new techniques or improvements that contribute to energy saving in wireless networks is considered to be a crucial task.

The purpose of this thesis is the development of mechanisms for transmission power management in wireless networks, whose goal is help mobile devices save energy. The basic idea of the thesis is the fact that a large amount of the transmissions that take place in a wireless network, often demand less energy for the successful deliverance to the destination than the amount wasted normally. As a coincidence, the devices' lifespan is reduced significantly.

The proposed mechanism utilizes simple metrics that are used to evaluate channel quality, such as Signal-to-Noise ratio (SNR). The presented mechanisms are applied not only on centralized networks, but in decentralized (ad hoc) ones as well, with different approach, though.

In centralized networks, the mechanism is supported by a plain feedback scheme between the access point and the connected nodes, in order to acquire information related to the transmissions. The access point send messages to the connected nodes, which contain information that is used by the nodes to adjust their transmission power. In decentralized networks the mechanism takes advantage of the routing protocols for transferring the necessary information, in order to avoid additional overhead in the network. The selected case subject where the mechanism is integrated is OLSR routing protocol, where certain modifications are made to make the use of the mechanism feasible.

For the evaluation of the mechanisms, experiments were conducted with the use of Network Simulator 3 tool (NS-3). In the experiments, the above cases were simulated, on different topologies with various mechanism parameters or number of nodes. The results showed that power saving is achieved to a great degree prolonging the lifespan of the devices, while the quality of the connection is not degraded to the extent that

communication is not feasible.

As part of the thesis, the following papers were published:

1. C. Bouras, V. Kapoulas, K. Stamos, N. Stathopoulos, N. Tavoularis, “Power Management and Rate Control Mechanism for Wi-Fi Infrastructure Networks”, 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), pp 731 - 736, 26-28 October 2015, Liverpool, UK
2. C. Bouras, V. Kapoulas, K. Stamos, N. Stathopoulos, “Evaluation of Power Control Mechanism on OLSR Routing Protocol”, 11th International Workshop on the Performance Analysis and Enhancement of Wireless Networks, 23-25 March 2016, Crans-Montana, Switzerland

Keywords

power management, Signal-to-noise ratio, interference, ad hoc, routing.

Περιεχόμενα

Ευχαριστίες	1
Περίληψη	3
Abstract	5
Περιεχόμενα	9
Κατάλογος Σχημάτων	12
Κατάλογος Πινάκων	13
1 Εισαγωγή	15
1.1 Αντικείμενο της διπλωματικής	16
1.1.1 Συνεισφορά	16
1.2 Οργάνωση του τόμου	17
2 Το IEEE 802.11n	19
2.1 Βασικά Στοιχεία	19
2.1.1 Τα στοιχεία	19
2.1.2 Τοπολογίες	20
2.2 Υπηρεσίες	21
2.2.1 Διανομή μηνυμάτων	21
2.2.2 Υπηρεσίες σύνδεσης	22
2.2.3 Ασφάλεια	24
2.2.4 Διαχείριση Φάσματος	24
2.3 Medium Access Control - MAC	24
2.3.1 Μορφή πλαισίου MAC	25
2.3.2 Μηχανισμός ανίχνευσης φέροντος (Carrier Sense)	26
2.3.3 MAC βασισμένο σε DCF	26
2.4 Φυσικό Επίπεδο	28
2.4.1 DSSS	29
2.4.2 OFDM	30

2.4.3	Επεκτάσεις στο 802.11n	31
3	Πρωτόκολλα δρομολόγησης	33
3.1	Εισαγωγή	33
3.2	Κατηγοριοποίηση των πρωτοκόλλων δρομολόγησης	34
3.2.1	Link State Routing (LSR) / Distance Vector Routing (DVR)	34
3.2.2	Proactive, Reactive and Hybrid δρομολόγηση	34
3.2.3	Επίπεδη / Ιεραρχική δομή	35
3.2.4	Source / Hop-by-hop routing	35
3.2.5	Μοναδικά/ Πολλαπλά μονοπάτια	35
3.3	Optimized Link State Routing Protocol (OLSR)	35
3.3.1	Διευθυνσιοδότηση κόμβων	36
3.3.2	Δομές πληροφορίας	36
3.3.3	Χρόνοι λήξης	37
3.3.4	Μηνύματα ελέγχου	37
3.3.5	Multipoint relaying	38
3.3.6	Χρήση πολλαπλών διεπαφών	40
3.3.7	Ανακάλυψη γειτόνων	41
3.3.8	Δήλωση κατάστασης σύνδεσης	44
3.3.9	Υπολογισμός μονοπατιού	45
3.3.10	Σύνοψη	46
4	Μετάδοση στα ασύρματα δίκτυα	49
4.1	Διάδοση	49
4.1.1	Διάδοση σε εσωτερικό χώρο	50
4.1.2	Διάδοση σε εξωτερικό χώρο	51
4.2	Λόγος σήματος προς θόρυβο	51
4.3	Μοντέλα σφάλματος	52
4.3.1	OFDM	52
4.4	Μετάδοση στα WLAN	54
5	Συγγενικές εργασίες	57
5.1	Εισαγωγή	57
5.2	Εξοικονόμηση ενέργειας στο 802.11n	57
5.2.1	Infrastructure WiFi	58
5.2.2	Ad hoc WiFi	58
5.3	Λοιπές εργασίες	59
6	Μηχανισμός προσαρμογής ισχύος μετάδοσης με το SNR ως κριτήριο απόφασης	63
6.1	Εισαγωγή	63
6.2	Βασική ιδέα	64

6.3	Επιλογή του SNR	64
6.4	Υπολογισμός ισχύος μετάδοσης	65
7	Υλοποίηση μηχανισμού εξοικονόμησης ενέργειας σε δίκτυα υποδο- μής	67
7.1	Εισαγωγή	67
7.2	Αρχιτεκτονική του μηχανισμού	67
7.3	Υλοποίηση του μηχανισμού	69
7.4	Ο προσομοιωτής δικτύων NS-3	71
7.5	Πειραματική αξιολόγηση	74
7.5.1	Αξιολόγηση ως προς την ισχύ μετάδοσης	76
7.5.2	Αξιολόγηση ως προς το packet loss	79
7.6	Σύνοψη και Συμπεράσματα	83
8	Υλοποίηση μηχανισμού εξοικονόμησης ενέργειας σε ad-hoc δίκτυα	85
8.1	Εισαγωγή	85
8.2	Αρχιτεκτονική του μηχανισμού	86
8.3	Πειραματική αξιολόγηση	88
8.4	Σύνοψη και Συμπεράσματα	92
9	Επίλογος	95
9.1	Σύνοψη	95
9.2	Μελλοντικές επεκτάσεις	96
	Βιβλιογραφία	98
A'	Κώδικες Υλοποίησης στον NS-3	103
A'.1	Εξαγωγή πιθανοτήτων επιτυχούς αποστολή πλαισίου για τα data rates του 802.11n	103
A'.2	Πείραμα για εξαγωγή SNR κόμβου ως προς την απόσταση από το Access Point	105
A'.3	Σενάριο προσομοίωσης του μηχανισμού του κεφαλαίου 7	112
A'.4	Σενάριο προσομοίωσης του μηχανισμού του κεφαλαίου 8	125

Κατάλογος Σχημάτων

2.1	Υποστηριζόμενες τοπολογίες του IEEE 802.11 WLAN	21
2.2	Γενική δομή πλαισίου MAC	25
2.3	Βασική πρόσβαση DCF MAC	27
3.1	Γενική μορφή πακέτου OLSR	37
3.2	Μορφή μηνύματος MID	41
3.3	Διαδικασία ανακάλυψης γειτόνων στο OLSR	41
3.4	Μορφή HELLO μηνύματος	42
3.5	Μορφή TC μηνύματος	44
3.6	Σύνοψη του OLSR. Πηγή [25]	47
4.1	Αστερισμός για την διαμόρφωση 16-QAM	53
4.2	Πιθανότητα επιτυχούς αποστολής πλαισίου για τα data rates του 802.11n.	53
4.3	SNR κινητού κόμβου ως προς την απόσταση από το access point.	54
5.1	Περιγραφή του μηχανισμού στο [29]. Πηγή:[29]	60
5.2	Δεξιά: Τα δίκτυα αλληλοκαλύπτονται, με αποτέλεσμα να ενισχύεται το exposed node problem. Αριστερά: Ίδανική περίπτωση με την εφαρμογή του [23]. Πηγή:[23]	60
5.3	Σύγκριση χρόνου ζωής μπαταρίας μεταξύ AODV και O-AODV. Πηγή:[7]	61
7.1	Περιγραφή του προτεινόμενου μηχανισμού σε δίκτυο υποδομής.	68
7.2	Εγγραφή κόμβου στην λίστα του μηχανισμού.	68
7.3	Αλληλεπιδράσεις στο μοντέλο OSI με την χρήση του μηχανισμού.	71
7.4	Οργάνωση του ns-3. Πηγή:[4]	72
7.5	Μέση ισχύς μετάδοσης για $offset = 1$	76
7.6	Μέση ισχύς μετάδοσης για $offset = 2$	77
7.7	Μέση ισχύς μετάδοσης για $offset = 3$	78
7.8	Μέση ισχύς μετάδοσης για $offset = 4$	79
7.9	Packet loss του πρώτου μηχανισμού για $offset = 1$	80
7.10	Packet loss του πρώτου μηχανισμού για $offset = 2$	81
7.11	Packet loss του πρώτου μηχανισμού για $offset = 3$	82
7.12	Packet loss του πρώτου μηχανισμού $offset = 4$	83

8.1	Δομή του HELLO πακέτου με την τροποποίηση του μηχανισμού.	87
8.2	Σύγκριση μέσης ισχύος μετάδοσης με και χωρίς την χρήση του μηχανισμού. .	90
8.3	Packet loss χωρίς την χρήση του μηχανισμού.	91
8.4	Packet loss με την χρήση του μηχανισμού.	92

Κατάλογος Πινάκων

2.1	Υποστηριζόμενοι ρυθμοί μετάδοσης στ 802.11n	31
2.2	Ρυθμοί κωδικοποίησης των ρυθμών μετάδοσης στο 802.11n	32
4.1	Παράμετροι για το πείραμα του Σχήματος 4.3.	54
6.1	Όρια SNR για επιτυχή μετάδοση	64
7.1	Χαρακτηριστικά πειραμάτων για την αξιολόγηση του πρώτου μηχανισμού.	74
7.2	Παράμετροι περιβάλλοντος για τα πειράματα του πρώτου μηχανισμού.	75
8.1	Χαρακτηριστικά πειραμάτων για την αξιολόγηση του μηχανισμού σε ad hoc δίκτυο.	88

Κεφάλαιο 1

Εισαγωγή

Οι ηλεκτρονικοί υπολογιστές έχουν γίνει ένα αναπόσπαστο κομμάτι της καθημερινότητας των ανθρώπων σε όλους τους τομείς. Από την εργασία μέχρι και την ψυχαγωγία, οι υπολογιστές παίζουν πρωτεύοντα ρόλο. Παλαιότερα, οι υπολογιστές χρησιμοποιούνταν σε τοπικό επίπεδο ως αυτόνομες μονάδες, οι οποίες πολύ συχνά ήταν αποκομμένες από το υπόλοιπο περιβάλλον. Η εξέλιξη των ευρυζωνικών τεχνολογιών όμως συνέβαλαν στην ευκολότερη πρόσβαση στο Διαδίκτυο, παρέχοντας νέες δυνατότητες στους χρήστες. Η πλοήγηση στο Διαδίκτυο είναι αν όχι η κυριότερη, μία από τις κυριότερες καθημερινές ασχολίες. Επιπλέον, ολοένα και περισσότερες εφαρμογές αξιοποιούν τα δίκτυα για την λειτουργία τους. Ένα παράδειγμα είναι η χρήση των cloud storages, όπου οι χρήστες μπορούν να ανταλλάζουν και να αποθηκεύουν αρχεία στο Διαδίκτυο.

Εκτός από την δυνατότητα για ευκολότερη πρόσβαση σε δίκτυα, ακόμη σημαντικότερο ρόλο έπαιξε η εξέλιξη των ασύρματων δικτύων. Επιτρέποντας στον χρήστη την πρόσβαση σε δίκτυα χωρίς την ύπαρξη κάποιου ενσύρματου μέσου και με υψηλές ταχύτητες μεταφοράς δεδομένων, τα ασύρματα δίκτυα έχουν ανατρέψει τα δεδομένα στους τομείς της δικτύωσης και επικοινωνίας των ηλεκτρονικών υπολογιστών. Η ευελιξία που παρέχει η ασύρματη επικοινωνία, έχει οδηγήσει στην ραγδαία αύξηση των ασύρματων δικτύων, καθώς απαιτούν λιγότερο εξοπλισμό και η διαδικασία εγκατάστασης είναι λιγότερο επίπονη (π.χ. απουσία καλωδίωσης μεταξύ του δρομολογητή και των συσκευών που συνδέονται στο δίκτυο).

Παράλληλα με τα ασύρματα δίκτυα, τεράστιες βελτιώσεις εμφανίστηκαν και στις κινητές συσκευές. Ο αριθμός των κινητών συσκευών αυξάνεται με ραγδαίους ρυθμούς σε σχέση με τις κλασικές στατικές[6]. Σημαντικό ρόλο για αυτή την αλλαγή έπαιξαν προφανώς τα ασύρματα δίκτυα, καθώς διευκολύνουν και επεκτείνουν τις δυνατότητες των συσκευών αυτών. Η υπολογιστική ισχύς που παρέχουν είναι αξιόλογη, πολλές φορές αντίστοιχη αυτής των σταθερών υπολογιστών, και η ευκολία μεταφοράς δεν είναι πρόβλημα, αφού η τεχνολογική εξέλιξη του υλικού έχει βοηθήσει στην κατασκευή φορητών υπολογιστών μικρού μεγέθους και βάρους.

Εκτός από τα κλασικά ασύρματα δίκτυα, τα ad hoc δίκτυα έχουν αρχίσει να γίνονται ολοένα πιο γνωστά στο κοινό. Λόγω της απουσίας υποδομών, τέτοια δίκτυα έχουν πάρα πολλές χρήσεις. Μερικές από αυτές συναντώνται στα δίκτυα οχημάτων, τα γνωστά VANETs,

στα μικρά τοπικά δίκτυα από smartphones (SPANs) και έχουν και στρατιωτική εφαρμογή καθώς η έλλειψη υποδομών δεν είναι αποδοτική σε αυτή τη περίπτωση.

Ένα σημαντικό πρόβλημα που αντιμετωπίζουν οι φορητές συσκευές, είναι η περιορισμένη ενέργεια που έχουν. Όντας κινητές, δεν έχουν κάποια σταθερή πηγή ενέργειας, με αποτέλεσμα ο χρόνος λειτουργίας τους να περιορίζεται σε μερικές μέρες ή και ώρες, ανάλογα με τον φόρτο εργασίας. Λαμβάνοντας υπ' όψη ότι η μεταφορά δεδομένων μέσω των ασύρματων δικτύων είναι μία από τις πιο συχνές λειτουργίες των σημερινών υπολογιστών, σε συνδυασμό με τον μεγάλο όγκο δεδομένων που μεταφέρεται, συμπεραίνεται ότι η ενέργεια που ξοδεύεται για αυτή την λειτουργία αποτελεί ένα αξιόλογο μέρος της συνολικής διαθέσιμης [8][26][18]. Η αύξηση της χρήσης του ασύρματου δικτύου από την συσκευή συνεπάγεται την μείωση του χρόνου λειτουργίας αυτής.

Μέσω της ανάλυσης των ασύρματων δικτύων μπορούν να εξαχθούν συμπεράσματα σχετικά με τα χαρακτηριστικά της επικοινωνίας που συντελείται, τα οποία μπορούν να οδηγήσουν σε εφεύρεση μηχανισμών οι οποίοι συμβάλλουν στην μείωση της ενέργειας που σπαταλάται για ασύρματη επικοινωνία. Για παράδειγμα, η απόσταση από το access point και το είδος κίνησης των κινητών συσκευών παίζει σημαντικό ρόλο στην ποιότητα της σύνδεσης, και μπορούν να εξαχθούν συμπεράσματα σχετικά το κατά πόσο είναι εφικτό να εξοικονομηθεί ενέργεια. Εναλλακτικά, προσαρμογές στον τρόπο λειτουργίας των δικτύων μπορούν να συμβάλλουν στην αυθαίρετη σπατάλη ενέργεια για αποστολή πακέτων.

1.1 Αντικείμενο της διπλωματικής

Αντικείμενο της διπλωματικής εργασίας είναι η μελέτη συγκεκριμένων χαρακτηριστικών της ασύρματης μετάδοσης και η υλοποίηση μηχανισμών, οι οποίοι αντιμετωπίζουν τις αρνητικές επιδράσεις των παρεμβολών που προκύπτουν μεταξύ των ασύρματων δικτύων, άλλα και μηχανισμών που προσαρμόζουν την ισχύ μετάδοσης των συσκευών με απώτερο σκοπό την μείωση της συνολικής κατανάλωσης ενέργειας από τις κινητές συσκευές για την μετάδοση των πακέτων πληροφορίας.

Βασική μετρική που μελετάται για την υλοποίηση των μηχανισμών είναι ο λόγος σήματος-προς-θόρυβο (SNR). Για την προσαρμογή της ισχύος μετάδοσης, το SNR χρησιμοποιείται ως μετρική που υποδηλώνει την ποιότητα τη μετάδοσης και θέτει ένα κάτω όριο βάσει του οποίου υπολογίζεται μία κατάλληλη τιμή για την ισχύ μετάδοσης των πακέτων. Σε αντίθεση με άλλου μηχανισμούς, οι μηχανισμοί αυτοί δρουν στο φυσικό επίπεδο του της επικοινωνίας, αφού στηρίζονται στην μεταβολή των μεγεθών που την χαρακτηρίζουν και έχουν άμεση επίδραση στην μετάδοση των πακέτων.

1.1.1 Συνεισφορά

Η συνεισφορά της διπλωματικής συνοψίζεται ως εξής:

1. Μελετήθηκε η συμπεριφορά του SNR σε περιβάλλοντα χωρίς και με παρεμβολές. Από αυτή την μελέτη αποδείχθηκε ότι το SNR μπορεί να θεωρηθεί ως ένα αξιόλογο και

αξιόπιστο κριτήριο για την λήψη αποφάσεων για την προσαρμογή ισχύος και ρυθμού μετάδοσης.

2. Υλοποιήθηκαν 2 αλγόριθμοι για να την προσαρμογή ισχύος μετάδοσης. Ο πρώτος αλγόριθμος εφαρμόστηκε σε infrastructure Wi-Fi δίκτυο ενώ ο δεύτερος αλγόριθμος σε δίκτυο ad hoc αξιοποιώντας το πρωτόκολλο δρομολόγησης OLSR.
3. Πραγματοποιήθηκαν πειράματα με την χρήση του εξομοιωτή Network Simulator-3(NS-3) [3], όπου διαπιστώθηκε ότι ο προτεινόμενος μηχανισμός για εξοικονόμηση ενέργειας καταφέρνει να μειώσει την κατανάλωση μέχρι και 90%, χωρίς να υπάρχουν απώλειες πακέτων. Επιπλέον, ο μηχανισμός προσαρμογής ρυθμού μετάδοσης βοηθά σημαντικά στην μείωση της απώλειας πακέτων εξαιτίας των παρεμβολών.

1.2 Οργάνωση του τόμου

Η οργάνωση της διπλωματικής εργασίας είναι ως εξής:

- Στο Κεφάλαιο 2 γίνεται η περιγραφή του προτύπου 802.11n το οποίο αποτελεί το βασικό στοιχείο για την ασύρματη επικοινωνία μεταξύ των συσκευών.
- Στο Κεφάλαιο 3 περιγράφονται τα πρωτόκολλα δρομολόγησης, τα οποία χρησιμοποιούνται στα ad hoc δίκτυα. Επίσης, αναλύεται το πρωτόκολλο δρομολόγησης OLSR το οποίο χρησιμοποιείται στον δεύτερο μηχανισμό της εργασίας.
- Στο Κεφάλαιο 4 μελετάται συνοπτικά η μετάδοση στα ασύρματα δίκτυα σε πολύ χαμηλό επίπεδο. Δίνεται έμφαση στην συμπεριφορά του SNR σε συνάρτηση με τους ρυθμούς μετάδοσης που καθορίζονται από το πρότυπο 802.11n
- Στο Κεφάλαιο 5 παρουσιάζονται εργασίες που έχουν γίνει στο παρελθόν και έχουν σχέση με την αντιμετώπιση παρεμβολών και την εξοικονόμηση ενέργειας και την προσαρμογή ρυθμού μετάδοσης στα ασύρματα δίκτυα.
- Στο Κεφάλαιο 6 παρουσιάζεται το μοντέλο πάνω στο οποίο στηρίζονται οι μηχανισμοί της εργασίας για εξοικονομήσουν ενέργεια κατά την μετάδοση των πακέτων.
- Στο Κεφάλαιο 7 περιγράφεται αναλυτικότερα ο μηχανισμός για τα infrastructure δίκτυα, ενώ στην συνέχεια παρουσιάζονται τα αποτελέσματα των πειραμάτων για αυτή την περίπτωση.
- Αντίστοιχα, στο Κεφάλαιο 8 περιγράφεται ο μηχανισμός εξοικονόμησης για τα ad hoc δίκτυα, μαζί με τα αντίστοιχα αποτελέσματα των πειραμάτων.
- Τέλος, στο Κεφάλαιο 9 γίνεται η σύνοψη της διπλωματικής εργασίας, όπου αναφέρονται τα αποτελέσματα και προτείνονται ιδέες για μελλοντικές εργασίες.

Κεφάλαιο 2

Το IEEE 802.11n

Το IEEE 802.11 είναι ένα σύνολο προτύπων για την επικοινωνία μεταξύ των υπολογιστών σε ένα ασύρματο τοπικό δίκτυο (Wireless LAN-WLAN), που αναπτύχθηκε από την IEEE. Το πεδίο εφαρμογής αυτών των προτύπων είναι να καθοριστούν ένα MAC και πολλές προδιαγραφές PHY για ασύρματη συνδεσιμότητα για σταθερούς ή κινούμενους σταθμούς (STAs). Όπως και με άλλα πρωτόκολλα βασισμένα στο IEEE 802 (π.χ. 802.3 και 802.5), η κύρια υπηρεσία του προτύπου 802.11 είναι να παραδώσει μονάδες δεδομένων της Υπηρεσίας MAC (MSDUs) μεταξύ ομότιμων LLC οντοτήτων.

Το 802.11n είναι η πιο κοινώς χρησιμοποιούμενη επέταση του 802.11. Παρόλο που το 802.11ac, που είναι το πιο πρόσφατο πρότυπο που έχει ανακοινωθεί από την IEEE, έχει ήδη ξεκινήσει να είναι διαθέσιμο στις σύγχρονες ασύρματες συσκευές, το 802.11n εξακολουθεί να χρησιμοποιείται ευρέως, καθώς οι αλλαγές που έφερε στο πρότυπο ήταν πολύ σημαντικές, όπως η υποστήριξη πολλαπλών εισόδων-πολλαπλών εξόδων (Multiple Input - Multiple Output (MIMO)).

Το κεφάλαιο αυτό παρέχει μια επισκόπηση των λειτουργιών που καθορίζονται από τα διάφορα IEEE 802.11 πρότυπα, δίνοντας ιδιαίτερη προσοχή στις πιο κοινές διαδικασίες και τα πιο δημοφιλή PHYs.

2.1 Βασικά Στοιχεία

Αυτή η ενότητα παρουσιάζει τα βασικά στοιχεία που υπάρχουν σε ένα IEEE 802.11 δίκτυο, τις πιθανές τοπολογίες, ανάλογα με τη διαμόρφωση και τις δυνατότητες των εν λόγω στοιχείων, και τις βασικές υπηρεσίες που υλοποιεί το επίπεδο MAC για να παρέχει ένα WLAN που επιτρέπει την κινητικότητα των σταθμών διαφανή σε υψηλότερα επίπεδα.

2.1.1 Τα στοιχεία

Το πιο βασικό στοιχείο σε ένα WLAN είναι ο σταθμός (STA). Κάθε συσκευή που περιέχει μία διασύνδεση με το ασύρματο μέσο, συμμορφούμενη με το IEEE 802.11 MAC και PHY, θεωρείται STA. Οι λειτουργίες του προτύπου 802.11 είναι υλοποιημένες ως μέρος του hardware

NIC, και ως κομμάτια λογισμικού που τρέχουν είτε στον kernel του λειτουργικού συστήματος (οδηγοί interface) ή στο user space(εφαρμογές). Ένας STA ονομάζεται QoS-STA αν υλοποιεί τις λειτουργίες QoS εισάγονται στο [1].

Ένα access point (AP) είναι μια οντότητα που υλοποιεί τις λειτουργίες ενός STA και εκτός αυτού, την παροχή πρόσβασης στις υπηρεσίες διανομής μέσω του ασύρματου μέσου, στους συνδεδεμένους STAs. Εάν ένα AP είναι παρόν στο δίκτυο, όλες οι επικοινωνίες μεταξύ δύο STAs γίνεται μέσω του AP, παρόλο που και οι δύο STAs βρίσκονται αρκετά κοντά, ώστε να επικοινωνήσουν άμεσα. Εάν χρησιμοποιείται πρόσβαση PCF, το AP είναι το σημείο συντονισμού, που έχει την ευθύνη να επιτελέσει το polling των PCF STAs.

Η βασική δομική μονάδα ενός IEEE 802.11 WLAN είναι γνωστή ως Basic Service Set (BSS). Ένα BSS είναι ένα σύνολο συνδεδεμένων STAs. Κατά κυριολεξία, η συμμετοχή σε ένα BSS δεν σημαίνει ότι είναι δυνατή η ασύρματη επικοινωνία με όλα τα άλλα μέλη του BSS. Ωστόσο, ένα BSS συχνά χρησιμοποιείται για να περιγράψει ένα AP και τους συνδεδεμένους STAs του. Αυτή η αρχιτεκτονική είναι επίσης γνωστή ως infrastructure BSS. Κάθε BSS προσδιορίζεται από ένα αναγνωριστικό BSS (BSSID). Σε ένα infrastructure BSS, η BSSID είναι η διεύθυνση MAC του AP. Σε αντίθετη περίπτωση, η BSSID είναι μια τοπικά χορηγούμενη διεύθυνση MAC που παράγεται από ένα 46-bit τυχαίο αριθμό.

Το σύστημα διανομής (DS) χρησιμοποιείται για τη διασύνδεση ενός συνόλου BSSs, συνήθως μέσω ενσύρματης υποδομής (π.χ. ένα Ethernet LAN). Το DS ή ραχοκοκαλιά μπορεί επίσης να κατασκευαστεί πλήρως ασύρματα μέσω του ασύρματου συστήματος διανομής (WDS).

2.1.2 Τοπολογίες

Χρησιμοποιώντας το BSSs όπως τα δομικά στοιχεία ενός IEEE 802.11 WLAN, δύο διαφορετικές τοπολογίες υποστηρίζονται, όπως απεικονίζεται στο Σχήμα 2.1: Ανεξάρτητη Basic (IBSS) και Extended Service Set (ESS).

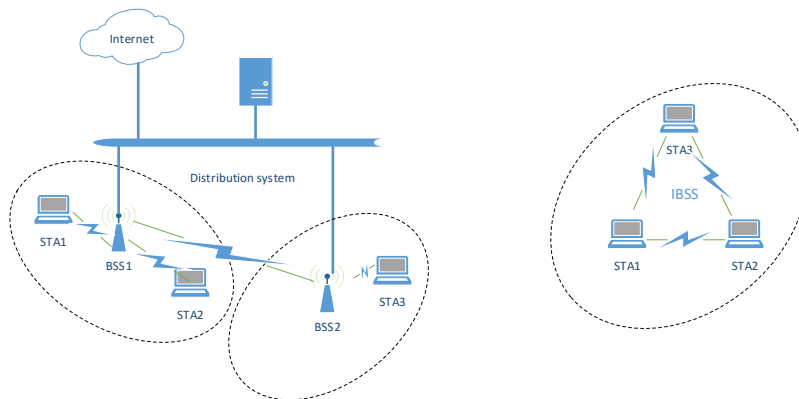
IBSS

Όταν ένα BSS σχηματίζει αυτοτελές δίκτυο χωρίς υποδομές και χωρίς πρόσβαση σε κάποιο διαθέσιμο DS, αυτό ονομάζεται IBSS, επίσης γνωστό ως δίκτυο ad hoc στο πρότυπο IEEE 802.11. Σε ένα IBSS, κάθε STA μπορεί να επιχειρήσει απευθείας επικοινωνία με οποιοδήποτε άλλο STA που ανήκει στο ίδιο BSS, χωρίς την απαίτηση της χρήσης ενός κεντρικού διαμεσολαβητή που ενεργεί ως AP.

Για τη σωστή λειτουργία σε αυτήν την κατάσταση, όλοι οι STAs θα πρέπει να είναι σε τέτοια απόσταση μεταξύ τους, ώστε η επικοινωνία να είναι εφικτή. Σε αντίθετη περίπτωση, θα πρέπει να εφαρμοστούν νέες λειτουργίες στο επίπεδο 2 ή το επίπεδο 3 (πρωτόκολλα δρομολόγησης), προκειμένου να είναι εφικτή η επικοινωνία μεταξύ όλων των ζευγών STAs.

ESS

Το ESS αποτελείται από πολλαπλά BSSs χρησιμοποιώντας ένα κοινό σύστημα διανομής. Ένα ESS θεωρείται από το LLC (Logical Link Control) υποεπίπεδο ως ένα ενιαίο μεγάλο BSS. Οι βασικές υπηρεσίες που παρέχονται από το κοινό σύστημα διανομής έχουν σκοπό τη μεταφορά πλαισίων μεταξύ των σταθμών που δεν είναι σε άμεση επικοινωνία με το άλλο. Δηλαδή, οι υπηρεσίες αυτές περιλαμβάνουν τη μεταφορά μεταξύ των APs μέσα σε ένα ESS. Το ESS παρέχει επίσης πρόσβαση για τους χρήστες ασύρματων εφαρμογών σε ένα σταθερό ενσύρματο δίκτυο, όπως ένα τοπικό δίκτυο (LAN) ή στο Διαδίκτυο. Επιπλέον, ένας STA μπορεί να μετακινηθεί από ένα BSS σε ένα άλλο (εντός της ίδιας ESS) με διαφάνεια στο LLC. Όλα τα APs στο ίδιο ESS ορίζονται με ένα κοινό αναγνωριστικό ESS (ESSID), το οποίο προσδιορίζει το δίκτυο. Ένα WDS μπορεί επίσης να χρησιμοποιηθεί για θέσεις που δεν είναι δυνατή η καλωδίωση ή για την επέκταση της περιοχής κάλυψης του BSS.



Σχήμα 2.1: Υποστηριζόμενες τοπολογίες του IEEE 802.11 WLAN

2.2 Υπηρεσίες

Το πρότυπο IEEE 802.11 ορίζει τις υπηρεσίες που θα πρέπει να εφαρμόζουν το DS και οι STAs. Στην ενότητα αυτή περιγράφονται εν συντομία αυτές τις υπηρεσίες, υποθέτοντας μια τοπολογία ESS.

2.2.1 Διανομή μηνυμάτων

Η κατανομή των μηνυμάτων δεδομένων μεταξύ STAs είναι η κύρια υπηρεσία που παρέχεται από ένα IEEE 802.11 δίκτυο. Στηρίζεται στην λειτουργία παράδοσης MSDU, η οποία είναι παρούσα σε όλους τους STAs.

Έστω η τοπολογία που περιγράφεται στο Σχήμα 2.1 και ένα μήνυμα από τον STA1 προς τον STA3. Η διανομή του μηνύματος έχει ως εξής: ο STA1 στέλνει το μήνυμα στο AP του, το AP δίνει το μήνυμα στο DS. Είναι η δουλειά του DS να παραδώσει αυτό το μήνυμα στον

κατάλληλο προορισμό, σε αυτή την περίπτωση, STA3 του AP. Τέλος, το AP εξόδου αποκτά πρόσβαση στο ασύρματο μέσο για να στείλει το μήνυμα στον τελικό προορισμό (STA3). Αυτή η λειτουργία μπορεί να απαιτήσει την διαβίβαση του πλαισίου μέσω διαφορετικών τεχνολογιών επιπέδου 2. Η προσαρμογή που απαιτείται για να παραδωθεί ένα πλαίσιο σε ένα μη-802.11 δίκτυο πραγματοποιείται από την υπηρεσία ολοκλήρωσης, που υπάρχει στα APs. Πώς το μήνυμα διανέμεται εντός του DS είναι πέρα από το πεδίο εφαρμογής του προτύπου. Ωστόσο, οι υπηρεσίες του IEEE 802.11 χρειάζεται να παρέχουν στο DS αρκετές πληροφορίες για να είναι σε θέση το δεύτερο να διαβιβάσει το μήνυμα προς το σωστό προορισμό του. Αυτές οι πληροφορίες παρέχονται από τις υπηρεσίες που σχετίζονται με την σύνδεση (association).

Στο προηγούμενο παράδειγμα, αποστέλλεται ένα μήνυμα σε ένα STA που ανήκει σε διαφορετικό BSS. Αν το μήνυμα προοριζόταν για το STA2, το AP εισόδου και εξόδου θα ήταν το ίδιο. Παρ'όλα αυτά, η υπηρεσία διανομής θα κληθεί ούτως ή άλλως. Επιπλέον, όλα τα μηνύματα δεδομένων που αποστέλλονται από οποιονδήποτε STA σε ένα infrastructure BSS πρέπει πάντα να αποστέλλονται στο AP του BSS, ακόμα κι αν τα STAs προέλευσης και προορισμού είναι σε εμβέλεια μεταξύ τους.

Με το WDS, η ασύρματη σύνδεση μεταξύ των APs χρησιμοποιεί την ίδια συσκευή με αυτή που χρησιμοποιήθηκε για την επικοινωνία με τους STAs. Τα ενιαία AP είναι φθηνότερα, αλλά οι κυψέλες και των δύο APs, καθώς και η ασύρματη σύνδεση μοιράζονται το ίδιο κανάλι, το οποίο μειώνει δραστικά την χωρητικότητα του δικτύου, δεδομένου ότι ορισμένα πλαίσια μπορούν να αποστέλλονται στο ασύρματο μέσο τρεις φορές: στο Σχήμα 2.1 ένα πλαίσιο που μεταδίδεται από το STA2 και απευθύνεται στο STA3 πρώτα αποστέλλεται από το STA2 στο AP του BSS1. Το AP χρησιμοποιεί το ίδιο κανάλι για να προωθήσει το πακέτο στο AP BSS2, το οποίο θα μεταδώσει τελικά πάλι στο STA3. Αυτή η αναποτελεσματικότητα μπορεί να μετριαστεί εάν τα APs έχουν δύο (ή περισσότερες) συσκευές που χρησιμοποιούν διαφορετικά κανάλια.

Η υπηρεσία χρονοπρογραμματισμού της QoS κίνησης είναι ενσωματωμένη στην υπηρεσία διανομής των ανάλογων δικτύων για την παροχή ενδο-BSS QoS μεταβιβάσεων πλαισίων.

2.2.2 Υπηρεσίες σύνδεσης

Πριν επιτραπεί στο STA να στείλει πλαίσια δεδομένων μέσω ενός AP, πρέπει πρώτα να συνδεθεί με το AP αυτό. Η λειτουργία αυτή παρέχει μία αντιστοιχία STA↔AP που χρησιμοποιείται από το DS για να πετύχει την υπηρεσία διανομής. Για αυτό τον λόγο, ένα STA θα πρέπει να είναι συνδεδεμένο με το πολύ 1 AP, καθώς το AP μπορεί να συνδεθεί με πολλά STAs ταυτόχρονα.

Προκειμένου το BSS να υποστηρίξει την κινητικότητα των STAs, το DS διαθέτει επίσης την υπηρεσία επανασύνδεσης. Αυτή η υπηρεσία χρησιμοποιείται για να διατηρούνται ενημερωμένες οι αντιστοιχίσεις STA↔AP, καθώς τα STAs κινούνται από BSS σε BSS, εντός ενός ESS. Η υπηρεσίες σύνδεσης και επανασύνδεσης εκκινούνται πάντα από το STA. Ο STA στέλνει ένα αίτημα σύνδεσης (ή επανασύνδεσης) στο επιλεγμένο AP, και το AP απαντά με την αντίστοιχη απόκριση.

Η αποσύνδεση χρησιμοποιείται για τερματιστεί μία υπάρχουσα σύνδεση. Συνεπώς, οι πληροφορίες για την υπάρχουσα σύνδεση αφαιρούνται. Αντίθετα με την σύνδεση και την επανασύνδεση, η αποσύνδεση μπορεί να εκκινηθεί είτε από τους STAs είτε από το AP, και δεν μπορεί να απορριφθεί, δεδομένου ότι δεν είναι ένα αίτημα, αλλά μάλλον μια ειδοποίηση.

Σάρωση

Όπως εξηγήθηκε προηγουμένως, όταν ένας STA σε λειτουργία infrastructure είναι ενεργοποιημένος, χρειάζεται να εγκαθιδρύσει μία σύνδεση με ένα AP. Η διαδικασία σάρωσης επιτρέπει στο STA να αποκτήσει γνώση του περιβάλλοντος του, και ως εκ τούτου ο STA είναι σε θέση να επιλέξει το καταλληλότερο AP στην εμβέλεια του. Όλα τα APs ανακοινώνονται σε μια περιοχή με την περιοδική μετάδοση Beacon Frames. Αυτά τα πλαίσια διαχείρισης περιέχουν πληροφορίες σχετικά με τις δυνατότητες και την κατάσταση του AP, οι οποίες είναι χρήσιμες για τους πιθανούς πελάτες. Οι STAs συνήθως επιλεγουν το καταλληλότερο AP βασισμένοι στις μετρήσεις της ένδειξης ισχύος ληφθέντος σήματος (Received Signal Strength Indicator - RSSI) κατά την λήψη των Beacon Frames. Δηλαδή, οι STAs θα ζητήσουν μια σύνδεση με το AP του οποίου τα Beacon Frames λαμβάνονται με την περισσότερη ενέργεια κατά τη διάρκεια της διαδικασίας σάρωσης.

Η σάρωση μπορεί να είναι είτε ενεργητική ή παθητική. Σταθμοί που χρησιμοποιούν παθητική σάρωση ακούν κάθε κανάλι και να περιμένουν για τα Beacon Frames για να εντοπίσουν όλα τα APs εντός εμβέλειας. Αντίθετα με την ενεργή σάρωση, οι STAs δεν δημιουργούν κανένα πλαίσιο κατά τη διάρκεια μιας παθητικής σάρωσης. Αυτό λειτουργεί καλά για δίκτυα με λίγα κανάλια και σύντομα χρονικά διαστήματα εκπομπής Beacon Frame. Η ενεργή σάρωση είναι μια ταχύτερη διαδικασία, αλλά απαιτεί μια ανταλλαγή σηματοδότησης. Για να ικανοποιήσει τις κανονιστικές απαιτήσεις (η ενεργή σάρωση απαγορεύεται σε ορισμένες ζώνες συχνοτήτων και ρυθμιστικά πεδία), μπορεί να απαιτείται η αρχική σάρωση να είναι πάντα παθητική.

Η διαδικασία ενεργής σάρωσης έχει ως εξής. Για κάθε υποστηριζόμενο κανάλι, ο σταθμός που είναι πρόθυμος να συνδεθεί με ένα νέο AP εκπέμπει ένα πλαίσιο Probe Request. Κάθε AP απαντά με ένα Probe Response στη διεύθυνση του STA που δημιούργησε το Probe Request, οπότε το ζητούμενο ESSID ταιριάζει με το ESSID του AP.

Περιογωγή

Το DS πρέπει να υποστηρίζει την κινητικότητα των χρηστών μεταξύ των BSSs μέσα σε ένα ESS. Όταν ο STA αντιλαμβάνεται ότι η ποιότητα της επικοινωνίας με το τρέχον AP υποβαθμίζεται είτε λόγω της κινητικότητάς του είτε λόγω παρουσίας παρεμβολών, προσπαθεί να βρει ένα καλύτερο υποψήφιο AP μέσω μιας νέας σάρωσης. Αυτό ονομάζεται περιογωγή, η οποία έχει ως εξής.

Ένας σταθμός παρακολουθεί τα Beacon Frames που λαμβάνονται από το τρέχον AP του. Όταν η ποιότητα αυτών των Beacon Frames πέσει κάτω από το όριο αναζήτησης κυψελών ($10 < C_{STh} < 30dB$), ο STA ξεκινά μια ενεργή σάρωση (αν είναι δυνατόν) και στέλνει Probe Requests σε όλα τα διαθέσιμα κανάλια. Τα AP που λαμβάνουν την αίτηση θα στείλουν ένα

Probe Response. Όταν βρεθεί ένα AP του οποίου οι απαντήσεις βελτιώνουν την ποιότητα των Beacon Frames του τρέχοντος AP κατά τουλάχιστον ΔSNR (συνήθως $6 \leq \Delta SNR \leq 8dB$), ο STA ξεκινά μία αλλαγή κυψελών (επανασύνδεση). Εάν δεν βρεθεί ένας καλύτερος υποψήφιος, ο STA επιστρέφει στο κανάλι του τρέχοντος AP και η σάρωση επαναλαμβάνεται σε τακτά χρονικά διαστήματα.

2.2.3 Ασφάλεια

Ο σχεδιασμός των ενσύρματων τοπικών δικτύων βασίζεται στις φυσικές ιδιότητες των καλωδίων που παρέχουν ασφάλεια στο δίκτυο: η πιστοποίηση παρέχεται από την απαίτηση φυσικής σύνδεσης με τα μέσα. Αυτές οι υποθέσεις δεν είναι πλέον έγκυρες για IEEE 802.11 WLAN λόγω της ανοιχτής φύσης του μέσου. Για το λόγο αυτό, το πρότυπο IEEE 802.11 ορίζει 2 υπηρεσίες προκειμένου να καλυφθούν οι προαναφερθείσες υποθέσεις που κληρονομούνται από τα ενσύρματα LANs: την πιστοποίηση και την εμπιστευτικότητα δεδομένων στο δεύτερο επίπεδο. Για την υλοποίηση των υπηρεσιών αυτών έχουν δημιουργηθεί 2 βασικές τεχνικές: τις WEP και WPA.

2.2.4 Διαχείριση Φάσματος

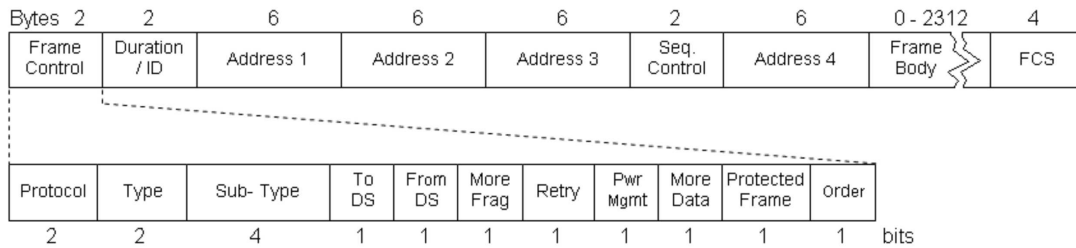
Προκειμένου να ικανοποιήσει τις απαιτήσεις σε διαφορετικά ρυθμιστικά πεδία, ειδικά για τη λειτουργία των WLANs στη ζώνη των 5 GHz, η IEEE εισήγαγε μια νέα τροποποίηση που κυκλοφόρησε από το 2003: το IEEE 802.11h. Καθορίζονται δύο υπηρεσίες: ο Έλεγχος Ισχύος Μετάδοσης (TPC) και η Δυναμική Επιλογή Συχνότητας (DFS).

Η υπηρεσία TPC έχει ως στόχο να μειώσει τις παρεμβολές με δορυφορικές συσκευές και το DFS χρησιμοποιείται για να αποφευχθεί η ομοκαναλική λειτουργία με συστήματα ραντάρ, καθώς και να εξασφαλιστεί η ομοιόμορφη αξιοποίηση των διαθέσιμων συχνοτικών καναλιών.

2.3 Medium Access Control - MAC

Όλοι οι IEEE 802.11 STAs σε ένα σύστημα μοιράζονται ένα κοινό ασύρματο μέσο. Το επίπεδο MAC ορίζει τους κανόνες που όλοι οι STAs πρέπει να τηρούν προκειμένου να μεταδώσουν σε αυτό το μέσο. Αυτό γίνεται με τη χρήση διαφόρων μηχανισμών πρόσβασης. Τα πρότυπα IEEE 802.11 καθορίζουν δύο μεθόδους πρόσβασης: την Distributed Coordination Function (DCF) και την Point Coordination Function (PCF). Επιπλέον, από τα τέλη του 2005, η ομάδα εργασίας IEEE 802.11 «E» κυκλοφόρησε ένα νέο πρότυπο, που ορίζει ένα σύνολο βελτιώσεων για το Quality of Service (QoS) των WLANs. Προσδιορίζονται οι διαδικασίες για τη διαχείριση ενός QoS δικτύου. Οι επεκτάσεις που εισάχθηκαν πηγάζουν από τους δύο προηγούμενους μηχανισμούς (DCF και PCF). Το νέο πρωτόκολλο MAC ονομάζεται Hybrid Coordination Function (HCF) και υπάρχει μόνο σε STAs με QoS, τα οποία να είναι ακόμη και σήμερα δυσέυρετα.

Το PCF και το HCF παρέχονται μέσω των υπηρεσιών της DCF, η οποία είναι η μόνη υποχρεωτική μέθοδος πρόσβασης. Η βασική μέθοδος πρόσβασης που χρησιμοποιείται από τα



Σχήμα 2.2: Γενική δομή πλαισίου MAC

Βασισμένα σε DCF MAC είναι η Carrier Sense Multiple Access with Collision Avoidance (CSMA / CA). Αυτή η μέθοδος βασίζεται στον ανταγωνισμό, ενώ η PCF στην ελεύθερη πρόσβαση. Οι τρεις τρόποι μπορούν να χρησιμοποιηθούν εναλλάξ στο χρόνο.

Η προαιρετική λειτουργία PCF επιτρέπεται μόνο στα infrastructure BSS. Στην περίπτωση αυτή, το AP ρωτάει διαδοχικά τους συνδεδεμένους STAs με την αποστολή μηνυμάτων polling. Επιπλέον, εάν το AP έχει δεδομένα έτοιμα για αποστολή σε ένα STA που ερωτάται, αυτά μπορούν να συμπεριληφθούν στο μήνυμα polling. Αν ο STA έχει δεδομένα για να στείλει στο το AP, αυτά μεταδίδονται στο μήνυμα απόκρισης. Με άλλα λόγια, η PCF είναι ένα πρωτόκολλο χωρίς ανταγωνισμό για το κανάλι και επιτρέπει στους σταθμούς να μεταδώσουν πλαίσια δεδομένων συγχρονισμένα, με τακτικές χρονικές καθυστερήσεις μεταξύ των μεταδόσεων πλαισίων δεδομένων. Η περίοδος απουσίας ανταγωνισμού έχει προτεραιότητα σε σχέση με την διαδικασία DCF. Με τον τρόπο αυτό, πακέτα που είναι ευαίσθητα στην καθυστέρηση (π.χ. φωνής ή βίντεο) μπορούν να έχουν υψηλότερη προτεραιότητα. Ωστόσο, το πρότυπο 802.11 είναι ασαφές στον καθορισμό του πρωτοκόλλου PCF. Σαν αποτέλεσμα, τα APs που εφαρμόζουν την PCF είναι σπάνια, ακόμα κι αν μερικά chipsets έχουν την PCF λειτουργικότητα ενσωματωμένη στο firmware. Επιπλέον, το Wi-Fi Alliance δεν περιλαμβάνει την PCF στο πρότυπο διαλειτουργικότητας του.

2.3.1 Μορφή πλαισίου MAC

Το IEEE 802.11 MAC ορίζει τρεις τύπους μηνυμάτων: δεδομένων, διαχείρισης και ελέγχου. Τα μηνύματα διαχείρισης χρησιμοποιούνται για την υποστήριξη των υπηρεσιών που περιγράφονται στο IEEE 802.11. Τα πλαίσια ελέγχου χρησιμοποιούνται για να υποστηρίξουν την παράδοση των μηνυμάτων δεδομένων και διαχείρισης. Τα πλαίσια δεδομένων μεταφέρουν την πληροφορία των υψηλότερων στρωμάτων. Η γενική μορφή πλαισίου MAC απεικονίζεται στο Σχήμα 2.2. Κάθε πλαίσιο αποτελείται από μια κεφαλίδα MAC, ένα σώμα πλαισίου μεταβλητού μήκους και μία ακολουθία πλαισίου ελέγχου (FCS), η οποία περιέχει ένα IEEE 32-bit CRC. Η κεφαλίδα MAC περιλαμβάνει διάφορα πεδία:

- **Frame Control:** αποτελείται από διάφορα υποπεδία και σημαίες που περιέχουν πληροφορίες για STAs σε λειτουργία εξοικονόμησης ενέργειας. Περιλαμβάνει επίσης πληροφορίες σχετικά με το τύπο πλαισίου, τον κατακερματισμό, κ.λ.π.

- Duration/ID: ποικίλει ανάλογα με τον τύπο του πλαισίου, αλλά συνήθως ορίζεται με το χρόνο (σε μs) που απαιτείται για να ολοκληρωθεί η τρέχουσα εκπομπή.
- Address Fields: χρησιμοποιείται για τον προσδιορισμό του BSSID, της διεύθυνσης πηγής, της διεύθυνσης προορισμού, της διεύθυνσης του εκπέμποντος STA και της διεύθυνσης του STA λήψης. Τα περιεχόμενα των πεδίων διεύθυνσης εξαρτώνται από τις τιμές των σημάτων To DS και From DS στο πεδίο Frame Control.
- Sequence Control: Χρησιμοποιείται για την αρίθμηση μίας μονάδας δεδομένων MAC (MSDU) και για την αναγνώριση κομματιών ενός συγκεκριμένου MSDU.
- QoS Control: οι STAs που υποστηρίζουν QoS, συμπεριλαμβάνουν επίσης ένα πεδίο 2 byte για τον προσδιορισμό της QoS συμπεριφοράς για το αντίστοιχο πλαίσιο.

Να σημειωθεί ότι τα πλαίσια δεν περιέχουν όλα τα πεδία που περιγράφονται παραπάνω. Τα τρία πρώτα πεδία και η FCS αποτελούν την ελάχιστη μορφή και εντοπίζονται παρούσα σε όλα τα πλαίσια. Τα υπόλοιπα πεδία βρίσκονται μόνο σε ορισμένους τύπους πλαισίων. Για παράδειγμα, η διεύθυνση 4 υπάρχει μόνο όταν χρησιμοποιείται WDS.

2.3.2 Μηχανισμός ανίχνευσης φέροντος (Carrier Sense)

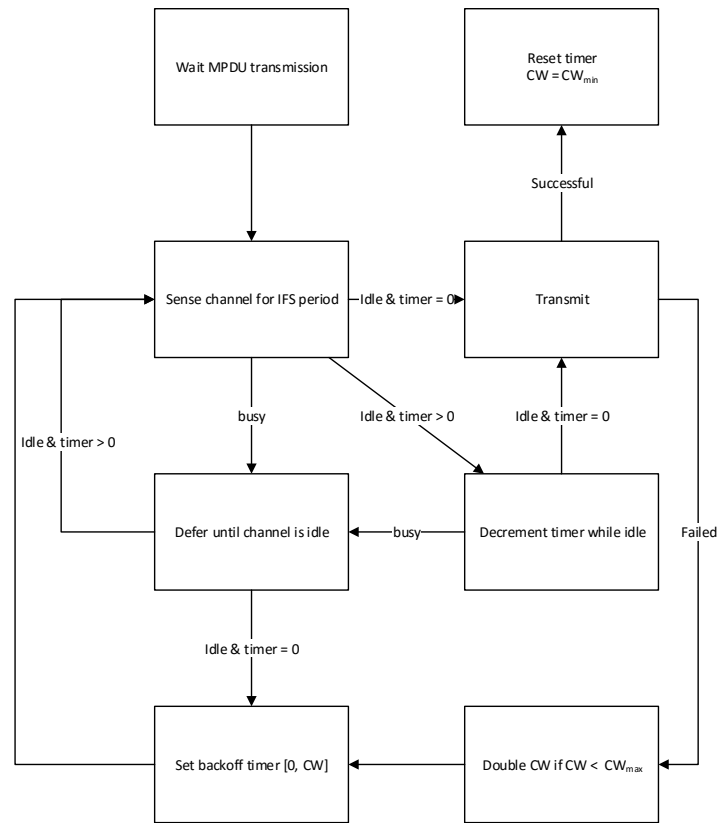
Για τον καθορισμό της κατάστασης του ασύρματου μέσου, το οποίο μπορεί να είναι είτε ελεύθερο ή απασχολημένο, μπορούν να χρησιμοποιηθούν φυσικές και εικονικές λειτουργίες ανίχνευσης φέροντος. Η εικονική ανίχνευση φέροντος αναφέρεται στο Network Allocation Vector (NAV). Το NAV διατηρεί μία πρόβλεψη της μελλοντικής κίνησης στο μέσο, βασισμένο στις πληροφορίες διάρκειας που ανακοινώνονται μέσω κάποιων πλαισίων.

Οι φυσικές λειτουργίες ανίχνευσης φέροντος παρέχονται από το PHY και εξαρτώνται από αυτό. Το φυσικό επίπεδο παρέχει αναγνώριση της κατάστασης του μέσου βασισμένο στην ανίχνευση ενέργειας της οποίας η ισχύς είναι μεγαλύτερη από ένα δεδομένο όριο P_{Th} . Οι τιμές του P_{Th} εξαρτώνται από την διαμόρφωση που χρησιμοποιείται και την ισχύ μετάδοσης. Αυτοί τιμές κυμαίνονται από -82 dBm (σήματα OFDM) έως -70 dBm (DSSS και η ισχύς μετάδοσης ≤ 50 mW).

2.3.3 MAC βασισμένο σε DCF

Όπως αναφέρθηκε προηγουμένως, η βασική μέθοδος πρόσβασης στο IEEE 802.11 WLANs είναι ένα DCF βασισμένο στο CSMA/CA και ένα τυχαίο χρονικό διάστημα υποχώρησης που ακολουθεί την κατάσταση δεσμευμένου καναλιού. Επιπλέον, όλες οι unicast μεταδόσεις χρησιμοποιούν πλαίσια θετικής επιβεβαίωσης (ACK). Αν δεν ληφθεί το ACK, τότε ο αποστολέας προγραμματίζει μία επαναμετάδοση του πλαισίου.

Οι κόμβοι ενός WLAN μοιράζονται το μέσο με ένα παρόμοιο τρόπο όπως οι κόμβοι ενός Ethernet δικτύου. Με το CSMA/CA, οι κόμβοι ανιχνεύουν το μέσο πριν από τη μετάδοση ενός πλαισίου. Εάν είναι απασχολημένο, περιμένουν έως ότου απελευθερωθεί. Χρησιμοποιούνται φυσικές και εικονικές λειτουργίες ανίχνευσης φέροντος για να προσδιορίζουν την



Σχήμα 2.3: Βασική πρόσβαση DCF MAC

κατάσταση του μέσου. Το CSMA/CA είναι σχεδιασμένο για να μειώσει την πιθανότητα σύγκρουσης, όταν πολλαπλοί STAs μοιράζονται το ίδιο κανάλι. Το σημείο όπου μια σύγκρουση είναι πιο πιθανό να συμβεί είναι ακριβώς μετά το μέσον τίθεται σε αδράνεια μετά από μια μετάδοση. Αυτό οφείλεται στο γεγονός ότι περισσότεροι από ένας STA μπορεί να περιμένουν το μέσο να τειθεί σε κατάσταση αδράνειας για να μεταδώσουν. Αυτές οι πιθανές συγκρούσεις ελαχιστοποιούνται χρησιμοποιώντας μια τυχαία διαδικασία υποχώρησης.

Βασική Πρόσβαση

Η βασική πρόσβαση του DCF φαίνεται στο Σχήμα 2.3 και λειτουργεί ως εξής: πριν ξεκινήσει μία μετάδοση, ο σταθμός ακούει το κανάλι για να διαπιστώσει αν είναι απασχολημένο. Αν το μέσο είναι αδρανές κατά την διάρκεια μιας χρονικής περιόδου που ονομάζεται DCF Interframe Space (DIFS), ο σταθμός μπορεί να μεταδώσει. Αν το μέσο είναι απασχολημένο, η μετάδοση καθυστερείται μέχρι να βρεθεί πάλι σε αδράνεια. Επιλέγεται ένα διάστημα εκθετικής υποχώρησης BO τέτοιο ώστε $BO \in [0, CW - 1]$ όπου CW είναι το παράθυρο ανταγωνισμού. Αυτή η διαδικασία καλείται πριν μεταδοθεί ένα πλαίσιο όταν το μέσο είναι απασχολημένο, και όταν ένας STA αντιληφθεί μία αποτυχημένη μετάδοση. Ο STA χρησιμοποιεί τον μηχα-

νισμό ανίχνευσης φέροντος για να προσδιορίσει αν υπάρχει δραστηριότητα σε κάθε σχισμή υποχώρησης. Ο μετρητής υποχώρησης μειώνεται όσο το κανάλι είναι αδρανές, σταματά όταν μία μετάδοση είναι σε εξέλιξη, και συνεχίζει όταν το κανάλι είναι αδρανές ξανά. Όταν λήξει ο μετρητής, ο σταθμός ξεκινά την μετάδοση. Η τιμή του CW είναι ορισμένη στην ελάχιστη τιμή του, CW_{min} στην πρώτη προσπάθεια μετάδοσης, και αυξάνεται κατά ακέραιες δυνάμεις του 2 σε κάθε αναμετάδοση, μέχρι να φτάσει μια προκαθορισμένη τιμή CW_{max} .

Μετά από κάθε επιτυχή λήψη unicast πλαισίου δεδομένων, ο δέκτης περιμένει για μία χρονική περίοδο που ονομάζεται Short Interframe Space (SIFS) και μεταδίδει ένα πλαίσιο ACK. Λογικά πρέπει $SIFS < DIFS$, έτσι το ACK έχει προτεραιότητα σε σχέση με τα άλλα πλαίσια, αποφεύγοντας περιττές αναμεταδόσεις. Τα ACKs δεν αποστέλλονται ως απάντηση σε broadcast/multicast πλαίσια.

Request to Send/Clear to Send (RTS/CTS)

Το πρωτόκολλο που περιγράφεται στην προηγούμενη υποενότητα ονομάζεται βασικός μηχανισμός ή μηχανισμός αμφίδρομης χειραψίας. Επιπλέον, η προδιαγραφή περιέχει επίσης ένα τεσσάρων κατευθύνσεων πρωτόκολλο ανταλλαγής πλαισίων, που ονομάζεται RTS / CTS, και λειτουργεί ως εξής: ένας σταθμός κερδίζει πρόσβαση στο κανάλι μέσω της διαδικασίας ανταγωνισμού που περιγράφηκε προηγουμένως, και στέλνει ένα ειδικό πλαίσιο που ονομάζεται Request to Send(RTS), αντί του πραγματικού πλαισίου δεδομένων. Ο δέκτης απαντά στέλνοντας ένα Clear to Send(CTS) πλαίσιο μετά από ένα διάστημα SIFS. Στη συνέχεια, ο σταθμός επιτρέπεται να ξεκινήσει τη μετάδοση πλαισίου δεδομένων μετά από ένα διάστημα SIFS. Οι πληροφορίες που περιέχονται σε μηνύματα RTS και CTS επιτρέπουν όλους τους STAs στην εμβέλεια να θέσουν το NAV τους για να υποδείξουν ότι το μέσο είναι απασχολημένο μέχρι το τέλος της αντίστοιχης μετάδοσης ACK.

Ο κύριος στόχος του RTS/CTS είναι η επίλυση του προβλήματος του κρυμμένου τερματικού. Ο μηχανισμός χρησιμοποιείται επίσης για την ελαχιστοποίηση των χαμένων περιόδων που προκαλείται από τις συγκρούσεις. Το πλαίσιο RTS είναι πολύ μικρότερο από τα πλαίσια δεδομένων. Λόγω του πρόσθετου overhead που προκαλείται από τα επιπλέον μηνύματα, αυτή η μέθοδος είναι συνήθως απενεργοποιημένη ή χρησιμοποιείται μόνο για την εξασφάλιση της μετάδοσης μεγάλων πλαισίων (το μέγεθος των οποίων υπερβαίνει ένα συγκεκριμένο όριο).

2.4 Φυσικό Επίπεδο

Το πρότυπο IEEE 802.11 ορίζει τέσσερις διαφορετικές τεχνικές μετάδοσης με τέσσερις διαφορετικές υλοποιήσεις PHY:

- Direct Spread Spectrum Sequence (DSSS)
- Orthogonal Frequency-Division Multiplexing (OFDM)
- Frequency-Hopping Spread Spectrum (FHSS)
- InfraRed (IR)

Οι πιο δημοφιλείς υλοποιήσεις είναι βασισμένες DSSS-CCK (Complementary Code Keying) και OFDM, οι οποίες χρησιμοποιούνται από τα IEEE 802.11a/b/g/n.

Το φυσικό επίπεδο χωρίζεται σε δύο υποεπίπεδα: το Physical Layer Convergence Procedure (PLCP) και το Physical Medium Dependent (PMD). Το PLCP ελαχιστοποιεί την εξάρτηση του επιπέδου MAC από το PMD με τη αντιστοίχιση MPDUs σε μια μορφή πλαισίου κατάλληλη για μετάδοση από το PMD. Υπό την καθοδήγηση του PLCP, το PMD παρέχει την ουσιαστική μετάδοση και λήψη PHY οντοτήτων μεταξύ δύο STAs μέσω του ασύρματου μέσου. Για την παροχή αυτής της υπηρεσίας, το PMD διασυνδέεται άμεσα με το ασύρματο μέσο και παρέχει διαμόρφωση και αποδιαμόρφωση των μεταδόσεων πλαισίων. Τα PLCP και PMD επικοινωνούν χρησιμοποιώντας απλοϊκές υπηρεσίες για να διαχειριστούν διέπει τις λειτουργίες μετάδοσης και λήψης. Κάθε υπόστρωμα PMD μπορεί να απαιτεί τον ορισμό ενός μοναδικού PLCP.

2.4.1 DSSS

Αυτή η ενότητα αναφέρει τις τεχνικές DSSS που χρησιμοποιούνται από το IEEE 802.11 στα 2.4 GHz. Η πρώτη προδιαγραφή του προτύπου ορίζει δύο βασικά rates, 1 και 2 Mbps. Αργότερα, προστέθηκαν υψηλότερα rates (5.5 και 11 Mbps), ως φυσική προέκταση του legacy DSSS με τη χρήση ενός συστήματος CCK. Αυτό το νέο καθεστώς παρείχε μια ομαλή μετάβαση σε υψηλότερα rates με DSSS, και επέτρεψε επίσης τη διαλειτουργικότητα με τις παλιές 1 και 2 Mbps συσκευές.

Legacy 802.11 DSSS

Το DSSS απλώνει ένα σήμα που μεταδίδεται σε ένα μεγαλύτερο εύρος φάσματος, ώστε να παρέχει ένα κέρδος εξάπλωσης από τις παρεμβολές στενής ζώνης. Οι τεχνικές DSSS απλώνουν τις μεταδόσεις περνώντας τα bit δεδομένων και μία ψευδοτυχαία ακολουθία από την XOR. Στο 802.11, αυτή η ακολουθία των bits είναι πάντα η 11-bit ακολουθία Barker: + 1-1 + 1 + 1-1 + 1 + 1 + 1-1-1-1. Δηλαδή, κάθε σύμβολο μεταδίδεται ως 11 bits που αντιπροσωπεύονται από την αρχική ακολουθία Barker ή την αντίστροφη της. Για όλους τους διαθέσιμους ρυθμούς δεδομένων (από 1 έως 11 Mbps), το chip rate είναι 11 Mcps, το οποίο οδηγεί σε ένα εύρος ζώνης καναλιού 22 MHz.

Για 1 Mbps, η ακολουθία Barker χρησιμοποιείται για να διαδοθεί ένα 1 Mbps σήμα με Differential BPSK (DBPSK). Για 2 Mbps, η ίδια ακολουθία είναι διαμορφωμένη με Differential QPSK (DQPSK). Με διαφορετικές διαμορφώσεις PSK, η πληροφορία μεταφέρεται από τη διαφορά φάσης μεταξύ των γειτονικών στοιχείων προς εκπομπή. Με τον τρόπο αυτό, η πολυπλοκότητα του συστήματος μειώνεται, δεδομένου ότι μια συνεκτική αναφορά φάση δεν είναι απαραίτητη. Αφ' ετέρου, το Bit Error Rate (BER) είναι υψηλότερο για ένα δεδομένο λόγο σήματος προς θόρυβο (SNR). Τα ακατέργαστα δεδομένα βασικής ζώνης είναι διαφορικά κωδικοποιημένα ως εξής: αν x_i είναι το κομμάτι που προορίζεται για μετάδοση και y_i είναι το bit που μεταδίδεται (διαφορικά κωδικοποιημένο), τότε $y_i = y_{i-1} \oplus x_i$.

High Rate DSSS

Η επέκταση αυτή ορίζει δύο επιπλέον PHY rates, στα 5.5 και 11 Mbps και είναι γνωστό ως High Rate DSSS (HR-DSSS). Ωστόσο, τα δεδομένα δεν εξαπλώνονται πια χρησιμοποιώντας την ακολουθία Barker. Αντί αυτού, χρησιμοποιείται το CCK, και προαιρετικά, το Packet Binary Convolutional Coding (PBC). Το CCK είναι μία παραλλαγή του M-ary Orthogonal Keying. Χρησιμοποιεί μια σειρά από συμπληρωματικούς κώδικες που προέρχονται από τα εισερχόμενα δεδομένα.

Για τα 11 Mbps, κάθε ομάδα 8 bits των εισερχόμενων δεδομένων είναι χωρισμένη σε 4 dibits. Το πρώτο dibit μετατρέπεται σε μία γωνία φάσης ϕ_1 , το οποίο είναι διαφορετικά κωδικοποιημένο σε κάθε διαδοχικές κωδικές λέξεις (Παρόμοια με τα 1 και 2 Mbps). Οι υπόλοιπες φάσεις (ϕ_2 έως ϕ_4) ορίζουν ένα σύνολο 64 πιθανών κωδικών των 8 chip και η ϕ_1 μπορεί να θεωρηθεί ως ένα επιπλέον φάση περιστροφής σε ολόκληρη τη λέξη κώδικα. Τα 8 chips αντιστοιχίζονται στον ίδιο QPSK αστερισμό και μεταδίδονται.

2.4.2 OFDM

Η τεχνική Orthogonal Frequency-Division Multiplexing (OFDM) αποτελεί τη βάση για το IEEE 802.11n που λειτουργεί στις ζώνες των 2.4 και 5 GHz. Η OFDM μαζί με το HR-DSSS, είναι οι πιο δημοφιλείς τεχνολογίες για τα WLAN.

Καταστάσεις λειτουργίας

Με το OFDM, η ροή δεδομένων υψηλού rate χωρίζεται σε 52 ρεύματα χαμηλότερου rate που μεταδίδονται ταυτόχρονα σε διαφορετικές υποφέρουσες. Έτσι, οι επιδράσεις της διασποράς χρόνου που προκαλείται από το multipath μειώνονται. Τέσσερις υποφέρουσες είναι οδηγοί που χρησιμοποιούνται ως σημείο αναφοράς για την παράβλεψη των αλλαγών συχνότητας ή φάσης. Οι οδηγοί παρεμποδίζονται με μια ακολουθία ψευδο-θορύβου μήκους 127. Οι υπόλοιπες 48 υποφέρουσες παρέχουν ξεχωριστά ασύρματα 'μονοπάτια' για την αποστολή των πληροφοριών με παράλληλο τρόπο. Τα πρότυπα καθορίζουν οκτώ διαφορετικούς τρόπους που παρέχουν 6-54 Mbps. Για όλες αυτές τις λειτουργίες, η διάρκεια συμβόλου είναι $4\mu s$. Επιπλέον, προκειμένου να μειωθεί διασυμβολική παρεμβολή, εισάγεται ένα διάστημα των 800 ns, που λέγεται Guard Interval (GI), σε κάθε σύμβολο OFDM. Το GI είναι μια επανάληψη του τέλους του συμβόλου στην αρχή. Αυτή η τεχνική, γνωστή ως κυκλικό πρόθεμα, χρησιμοποιείται για να διατηρήσει τις ημιτονοειδείς ιδιότητες που διευκολύνουν την εκτίμηση καναλιού στα multipath κανάλια. Ο χρόνος διάρκειας συμβόλου και του GI ορίζει το εύρος της υποφέρουσας στα 0.3125 MHz ($(4\mu s - 800ns) - 1$), καθώς και το συνολικό εύρος ζώνης που χρησιμοποιείται, 16.6 MHz για όλες τις λειτουργίες.

Η πιο αργή λειτουργία χρησιμοποιεί Binary Phase-Shift Keying (BPSK) και η ταχύτερη 64-Quadrature Amplitude Modulation (QAM). Με 48 υποφέρουσες και σύμβολα των $4\mu s$ και 1 έως 6 bits ανά υποφέρουσα, το bit rate κυμαίνεται από 12 έως 72 Mbps. Για να διορθωθεί η εξασθένιση στις υποφέρουσες, εφαρμόζεται Forward Error Correction (FEC) με τη βοήθεια συνελκτικών κωδικών. Ο ρυθμός κωδικοποίησης είναι μεταβλητός και ως εκ

Δείκτης MCS	Τύπος Διαμόρφωσης	Ρυθμός Μετάδοσης(Mbps)	
		Εύρος καναλιού 20 MHz	Εύρος καναλιού 40 MHz
0	BPSK	6.5	13.5
1	QPSK	13	27
2	QPSK	19.5	40.5
3	16-QAM	26	54
4	16-QAM	39	81
5	64-QAM	52	108
6	64-QAM	58.5	121.5
7	64-QAM	65	135

Πίνακας 2.1: Υποστηριζόμενοι ρυθμοί μετάδοσης στ 802.11n

τούτου οι οκτώ διαφορετικές επιτρεπτές καταστάσεις λειτουργίας είναι οι: 6.5, 13, 19.5, 26, 39, 52, 58.5 και 65 Mbps. Επιπλέον, με την εισαγωγή της τεχνικής MIMO (Multiple Input-Multiple Output), μπορεί να διπλασιαστεί το εύρος του καναλιού στα 40 MHz, αυξάνοντας παράλληλα τα rates. Τα υποστηριζόμενα rates για αυτές τις 2 περιπτώσεις φαίνονται στον Πίνακα 2.1 και ο αντίστοιχος ρυθμός κωδικοποίησης στο Πίνακα 2.2.

2.4.3 Επεκτάσεις στο 802.11n

Κατανομή καναλιών

Μία από τις απαιτήσεις για το πρότυπο 802.11n είναι η υποστήριξη της λειτουργίας του τόσο στην ζώνη των 2.4 GHz όσο και στην ζώνη των 5 GHz για την οποία πρωτίστως σχεδιάστηκε. Στη ζώνη των 2.4 GHz είναι διαθέσιμα 13 κανάλια, όπου οι κεντρικές συχνότητες του κάθε καναλιού σε βρίσκονται σε απόσταση 5 MHz μεταξύ τους, υπονοώντας υπάρχει χώρος για 3 μη επικαλυπτόμενα κανάλια σε ολόκληρη την ζώνη. Έτσι, η κεντρική συχνότητα του καναλιού i μπορεί να υπολογιστεί από την σχέση:

$$f_{c_i} = 2407 + 5 \cdot i(\text{MHz}) \quad (2.1)$$

όπου $i = 1, 2, 3, \dots, 13$. Ωστόσο, δεδομένου ότι η μετάδοση γίνεται μέσω καναλιών των 40 MHz, μόνο 2 μη επικαλυπτόμενα κανάλια είναι διαθέσιμα στη ζώνη των 2.4 GHz και η λειτουργία στην ζώνη των 5 GHz φαίνεται να προσφέρει περισσότερες δυνατότητες και χώρο σε σχέση με τη διαθεσιμότητα συχνοτήτων. Στη ζώνη των 5 GHz η πρώτη κεντρική συχνότητα ξεκινά στα 5000 MHz και παρέχει χώρο συχνοτήτων για μέχρι 200 κανάλια όλα με απόσταση 5 MHz. Συνεπώς οι κεντρικές συχνότητες δίνονται από:

$$f_{c_i} = 5000 + 5 \cdot i(\text{MHz}) \quad (2.2)$$

όπου $i = 1, 2, 3, \dots, 200$. Περισσότερα από 20 μη-επικαλυπτόμενα κανάλια είναι εφικτά σε αυτή την ζώνη.

MCS	Ρυθμός Κωδικοποίησης
0	1/2
1	1/2
2	3/4
3	1/2
4	3/4
5	2/3
6	3/4
7	5/6

Πίνακας 2.2: Ρυθμοί κωδικοποίησης των ρυθμών μετάδοσης στο 802.11n

Μέγιστη Ισχύς Μετάδοσης

Το πρότυπο 802.11n έχει μέγιστη επιτρεπόμενη ισχύ εξόδου η οποία είναι η ίδια όπως στα πρότυπα 802.11a/g, δηλαδή 100 mW ή 20 dB(στην Ευρώπη). Αυτό σημαίνει ότι σε περίπτωση που αλυσίδες πολλαπλής μετάδοσης είναι ενεργές, η συνολική ισχύς τους δεν πρέπει να υπερβαίνει αυτό το όριο.

MIMO

Η Multiple Input - Multiple Output είναι μία τεχνική που χρησιμοποιεί πολλαπλές κεραίες στον πομπό και το δέκτη επιτρέποντας την χωρική ποικιλομορφία, με αποτέλεσμα την αύξηση της πολυμορφίας, και την πολυπλεξία, με την οποία αυξάνονται τα data rates. Η πολυπλεξία χρησιμοποιεί τον πολυδιδευτικό χαρακτήρα του κανάλι για την αποστολή ανεξάρτητων δεδομένων πάνω από ανεξάρτητες διαδρομές. Με αυτό τον τρόπο MIMO αντιμετωπίζει το multipath fading του καναλιού και δέχεται οφέλη από αυτό με τέτοιο τρόπο ώστε να επιτευχθεί υψηλότερη φασματική απόδοση, υπό την προϋπόθεση ότι υπάρχει ακριβής γνώση του καναλιού στην μεριά του δέκτη. Συχνά, ένα trade-off πρέπει να γίνει μεταξύ των δύο τεχνικών: υψηλός ρυθμός μετάδοσης επιτυγχάνεται μέσω της χωρικής πολυπλεξίας και η αύξηση της αξιοπιστίας γίνεται μέσω της ποικιλομορφίας της μετάδοσης. Επιπλέον, η διασυμβολική παρεμβολή (Intersymbol Interference - ISI) και παρεμβολές από άλλους χρήστες μπορεί να μειωθεί κατά τη χρήση των κεραιών με έξυπνο τρόπο. Επιπλέον με το MIMO μπορεί να επιτευχθεί κέρδος συστοιχίας κεραιών, το οποίο είναι το αυξημένο λαμβανόμενο SNR που προκύπτει από το αποτέλεσμα συνδυασμού των σημάτων στο δέκτη. Τα οφέλη αυτά συμβάλλουν σε σημαντικά κέρδη απόδοσης. Τα πλεονεκτήματα της απόδοσης, έχουν από την άλλη πλευρά, κάποιο προστιθέμενο κόστος. Αυτά περιλαμβάνουν το κόστος για την ανάπτυξη των πολλαπλών κεραιών, τις απαιτήσεις ισχύος τους και υψηλότερη πολυπλοκότητα.

Κεφάλαιο 3

Πρωτόκολλα δρομολόγησης

Τα πρωτόκολλα δρομολόγησης αποτελούν ένα σημαντικό μέρος στην εξέλιξη των κινητών ad hoc δικτύων (Mobile Ad hoc Networks - MANETs). Στο κεφάλαιο αυτό αναλύεται η κατηγοριοποίηση αυτών και περιγράφεται το πρωτόκολλο δρομολόγησης OLSR το οποίο χρησιμοποιείται σε ένα από τους μηχανισμούς που παρουσιάζονται στην παρούσα εργασία.

3.1 Εισαγωγή

Στα infrastructure WLANs η δρομολόγηση των πακέτων δεν αποτελεί πρόβλημα λόγω της φύσης της επικοινωνίας. Οι κόμβοι επιτρέπεται να στείλουν δεδομένα μόνο στο AP, συνεπώς δεν υπάρχει ανάγκη για εύρεση μονοπατιού προς τον τελικό προορισμό. Στα ad hoc WLANs όμως, η δρομολόγηση αποτελεί το βασικότερο πρόβλημα. Τα πρωτόκολλα δρομολόγησης χρησιμοποιούνται για την ανακάλυψη μονοπατιών μεταξύ των κόμβων, για να επιτραπεί η επικοινωνία στο δίκτυο. Ο κύριος στόχος τους είναι να καθορίσουν ένα σωστό και αποτελεσματικό μονοπάτι ανάμεσα σε 2 κόμβους, έτσι ώστε τα μηνύματα να μπορούν να φτάσουν στον προορισμό τους σε αποδεκτά χρονικά πλαίσια. Τις τελευταίες 2 δεκαετίες, έχουν προταθεί πολλά πρωτόκολλα δρομολόγησης για MANETs, λόγω της αναγκαιότητας τους στα δυναμικά και συνεχώς μεταβαλλόμενα κινητά δίκτυα. Λόγω της δυναμικότητας αυτής, δεν είναι δυνατό να δημιουργηθούν πρωτόκολλα τέτοια ώστε να καλύπτουν όλες τις πιθανές περιπτώσεις επικοινωνίας. Κάθε πρωτόκολλο έχει τα δικά του πλεονεκτήματα και μειονεκτήματα, και μπορεί να προορίζονται για εξειδικευμένες περιπτώσεις.

Ο σχεδιασμός των πρωτοκόλλων δρομολόγησης είναι σχετικά δύσκολος λόγω των προκλήσεων που δημιουργούνται από τα δίκτυα. Πρώτον, στα κινήτα αυτά δίκτυα, η τοπολογία αλλάζει συνεχώς λόγω της κίνησης των κόμβων. Δεύτερον, εξαιτίας της ευμετάβλητης και απρόβλεπτης χωρητικότητας των συνδέσεων, το packet loss είναι ένα συχνό φαινόμενο. Επιπλέον, οι εκπομπές των κόμβων έχουν ως αποτέλεσμα την εμφάνιση προβλημάτων όπως αυτό του κρυμμένου τερματικού ή του εκτεθειμένου τερματικού (hidden/exposed terminal). Τέλος, πολύ σημαντικό πρόβλημα που κάνει αναγκαίο τον σχεδιασμό πρωτοκόλλων δρομολόγησης είναι η περιορισμένη ενέργεια που διαθέτουν οι κόμβοι για την λειτουργία τους, καθώς επίσης και το περιορισμένο εύρος ζώνης.

3.2 Κατηγοριοποίηση των πρωτοκόλλων δρομολόγησης

Τα πρωτόκολλα δρομολόγησης μπορούν να κατηγοριοποιηθούν σε διάφορες ομάδες, ανάλογα με το κριτήριο με το οποίο χαρακτηρίζονται. Το κριτήριο μπορεί να είναι ο τρόπος με τον οποίο αξιοποιούν χαρακτηριστικά του δικτύου ή η τακτική με την οποία δημιουργούνται τα μονοπάτια για την αποστολή των πακέτων.

3.2.1 Link State Routing (LSR) / Distance Vector Routing (DVR)

Τα πρωτόκολλα που βασίζονται στο Link State Routing (LSR) λαμβάνουν υπ' όψη μετρικές που χαρακτηρίζουν τις συνδέσεις, όπως το bandwidth, την καθυστέρηση (delay), την αξιοπιστία και τον φόρτο αυτών. Στο LSR, οι κόμβοι ανταλλάσσουν πληροφορίες για την δρομολόγηση με την μορφή πακέτων Link state packets (LSP), τα οποία περιέχουν πληροφορίες σχετικές με την σύνδεση για όλους του γείτονές τους. Καθώς αλλάζει η κατάσταση των συνδέσεων, τα LSPs εξαπλώνονται αμέσως σε όλο το δίκτυο. Λαμβάνοντας τα LSPs, κάθε κόμβος μπορεί να κατασκευάσει και να διατηρήσει την τοπολογία του δικτύου.

Από την άλλη μεριά, τα πρωτόκολλα που βασίζονται στο Distance Vector Routing (DVR) προσδιορίζουν το καλύτερο μονοπάτι ως προς το πόσο μακριά είναι ο προορισμός. Στο DVR, κάθε κόμβος διατηρεί ένα διάνυσμα που μπορεί να περιέχει το ID του προορισμού, τον επόμενο κόμβο και την απόσταση από κάθε προορισμό. Κάθε κόμβος ανταλλάσσει τέτοια διανύσματα με τους γείτονές του. Κατά την λήψη ενός τέτοιου διανύσματος, ο κόμβος κατασκευάζει νέα μονοπάτια και ενημερώνει το διάνυσμά του.

3.2.2 Proactive, Reactive and Hybrid δρομολόγηση

Ανάλογα με τον τρόπο που διατηρούνται τα μονοπάτια, τα πρωτόκολλα δρομολόγησης μπορούν να κατηγοριοποιηθούν ως proactive, reactive ή υβριδικά. Τα proactive πρωτόκολλα διαδίδουν συνεχώς και διατηρούν πληροφορίες για την δρομολόγηση. Συνεπώς, κάποια διαδρομή προς οποιοδήποτε προσβάσιμο κόμβο στο δίκτυο είναι πάντα διαθέσιμη, ανεξάρτητα από το αν χρειάζεται ή όχι. Αυτά τα πρωτόκολλα ανταποκρίνονται στις αλλαγές της τοπολογίας μέσω της διάδοσης ενημερώσεων σε όλο το δίκτυο.

Τα reactive πρωτόκολλα δημιουργούν διαδρομές μόνο όταν απαιτούνται από τον κόμβο που θέλει να στείλει κάποιο πακέτο. Όταν ο κόμβος αυτός θελήσει να στείλει δεδομένα προς ένα άλλο κόμβο αλλά δεν γνωρίζει κάποια κατάλληλη διαδρομή, ξεκινάει μία διαδικασία για την ανακάλυψη κατάλληλης διαδρομής. Αφού βρεθεί μία τέτοια διαδρομή, διατηρείται από κάποια κατάλληλη διαδικασία.

Τα υβριδικά πρωτόκολλα δρομολόγησης συνδυάζουν τα 2 παραπάνω είδη πρωτοκόλλων, αξιοποιώντας τα πλεονεκτήματα που προσφέρει το καθένα και να ξεπεραστούν τα μειονεκτήματά τους.

3.2.3 Επίπεδη / Ιεραρχική δομή

Στην επίπεδη δομή, όλοι οι κόμβοι συμμετέχουν στο δίκτυο στον ίδιο βαθμό και επιτελούν όλοι τις λειτουργίες για την δρομολόγηση. Για μικρά δίκτυα, η προσέγγιση αυτή είναι απλή στην υλοποίηση και αποδοτική. Στα μεγάλα δίκτυα όμως, οι πληροφορίες που πρέπει να μεταδοθούν στο δίκτυο σχετικά με την δρομολόγηση είναι πολλές και η πληροφορία δεν εξαπλώνεται αρκετά γρήγορα σε όλο το δίκτυο. Στα μεγάλα δίκτυα, αντί της επίπεδης δομής, χρησιμοποιείται ιεραρχική δομή για την αντιμετώπιση αυτών των προβλημάτων.

Στην ιεραρχική δομή, οι κόμβοι οργανώνονται σε συστάδες clusters. Στην συνέχεια, οι συστάδες ομαδοποιούνται σε μεγαλύτερες ομάδες (superclusters) και ούτω καθεξής. Κάποιος κόμβος μπορεί να έχει πληροφορίες για ολόκληρη την τοπολογία για το cluster του, οπότε μπορεί να χρησιμοποιηθεί proactive δρομολόγηση. Αν ο προορισμός βρίσκεται σε διαφορετικό cluster, μπορεί να χρησιμοποιηθεί δρομολόγηση μεταξύ των clusters. Σε αυτή την περίπτωση, μπορεί να χρησιμοποιηθεί reactive πρωτόκολλο ή κάποιος συνδυασμός, όπως είναι το πρωτόκολλο Zone-based Hierarchical Link State (ZHLS).

3.2.4 Source / Hop-by-hop routing

Κάποια από τα πρωτόκολλα δρομολόγησης περιέχουν ολόκληρο το μονοπάτι στην κεφαλίδα (header) του πακέτου δεδομένων. Έτσι, οι ενδιαμέσοι κόμβοι προωθούν τα πακέτα σύμφωνα με το μονοπάτι που υπάρχει στην κεφαλίδα. Ο μηχανισμός αυτός ονομάζεται source routing. Το πλεονέκτημα του είναι ότι οι ενδιαμέσοι κόμβοι προωθούν το πακέτο χωρίς να διατηρούν πληροφορίες σχετικά με την δρομολόγηση, αφού τα πακέτα περιέχουν όλες τις αποφάσεις σχετικά με αυτή. Από την άλλη μεριά, στο hop-by-hop routing, το μονοπάτι υπολογίζεται βήμα βήμα σε κάθε ενδιαμεσο κόμβο που φτάνει το πακέτο.

3.2.5 Μοναδικά/ Πολλαπλά μονοπάτια

Μερικά πρωτόκολλα δρομολόγησης διατηρούν μόνο ένα μονοπάτι για κάθε προορισμό. Αυτό οδηγεί στην υλοποίηση απλών πρωτοκόλλων και στην μείωση του χώρου που χρειάζεται για την αποθήκευση της πληροφορίας. Τα πρωτόκολλα μοναδικού μονοπατιού όμως, δεν είναι ικανά να εξισορροπήσουν την κίνηση του δικτύου. Άλλα πρωτόκολλα διατηρούν πολλαπλά μονοπάτια για κάθε προορισμό. Αυτά τα πρωτόκολλα έχουν το πλεονέκτημα ότι μπορούν να επανέλθουν πιο γρήγορα σε περίπτωση που κάποια σύνδεση καταρεύσει, κάνοντας τα πιο αξιόπιστα και ανθεκτικά. Επιπλέον, ο κόμβος που στέλνει το πακέτο μπορεί επιλέξει την καλύτερη διαδρομή ανάμεσα σε όλες τις πιθανές.

3.3 Optimized Link State Routing Protocol (OLSR)

Το πρωτόκολλο OLSR [5][3] ένα πρωτόκολλο που σχεδιάστηκε για τα MANETs. Το OLSR είναι ένα proactive πρωτόκολλο και χρησιμοποιεί την τεχνική Multipoint Relaying για τον έλεγχο του flooding της κίνησης στο δίκτυο. Το πρωτόκολλο αποτελείται από 2 σύνολα, το σύνολο με τις βασικές λειτουργίες, με το οποίο μπορεί να γίνει η δρομολόγηση

στο MANET, και το σύνολο βοηθητικών λειτουργιών, το οποίο συμβάλλει στην βελτίωση της επίδοσης του πρωτοκόλλου σε εξειδικευμένες περιπτώσεις. Στην ενότητα αυτή αναλύεται μόνο οι βασικές λειτουργίες.

3.3.1 Διευθυνσιοδότηση κόμβων

Το OLSR χρησιμοποιεί την διεύθυνση IP ως το μοναδικό αναγνωριστικό για τους όμβους που υπάρχουν στο δίκτυο. Καθώς το OLSR είναι σχεδιασμένο να λειτουργεί σε κόμβους που μπορεί να χρησιμοποιούν πολλαπλές διεπαφές δικτύου, κάθε κόμβος πρέπει να επιλέξει μία IP διεύθυνση που ορίζεται ως κύρια.

Το OLSR μπορεί να χρησιμοποιηθεί με IPv4 και με IPv6 διευθύνσεις. Στα πλαίσια του OLSR η μόνη διαφορά μεταξύ των 2 είναι το μέγεθος του πεδίου της διεύθυνσης που περιέχεται στα μηνύματα ελεγχου, το οποίο στην δεύτερη περίπτωση είναι μεγαλύτερο.

3.3.2 Δομές πληροφορίας

Το OLSR διατηρεί την κατάσταση των συνδέσεων χρησιμοποιώντας ένα μεγάλο πλήθος πινάκων που περιέχουν πληροφορίες. Αυτές οι δομές ενημερώνονται κατά την επεξεργασία των λαμβανόμενων μνημάτων ελέγχου και η πληροφορία που αποθηκεύεται χρησιμοποιείται κατά τη δημιουργία τέτοιων μνημάτων. Οι δομές που διατηρούνται σε κάθε κόμβο για την λειτουργία του πρωτοκόλλου είναι οι παρακάτω:

- **Multiple Interface Association Information Base:** Η δομή αυτή περιέχει πληροφορίες για τους κόμβους που χρησιμοποιούν περισσότερες από μία διεπαφές. Όλες οι διευθύνσεις των διεπαφών αποθηκεύονται εδώ.
- **Link Set:** Αυτή η δομή διατηρείται για την εκτίμηση της κατάστασης των συνδέσεων με τους γειτονικούς κόμβους. Είναι η μόνη βάση δεδομένων που εκτελεί λειτουργίες σε μη κύριες διευθύνσεις καθώς δουλεύει πάνω στις συνδέσεις κατά ζεύγη διεπαφών.
- **Neighbor Set:** Όλοι οι κόμβοι που απέχουν ένα βήμα καταγράφονται εδώ. Τα δεδομένα ενημερώνονται σύμφωνα με τις πληροφορίες που παρέχονται από το link set. Καταγράφονται όλοι οι γειτονικοί κόμβοι, συμμετρικοί και μη.
- **2-hop Neighbor Set:** Στην δομή αυτή αποθηκεύονται όλοι οι κόμβοι που είναι προσβάσιμοι μέσω ενός άμεσου (one-hop) γείτονα. Στην δομή αυτή αποθηκεύονται και οι κόμβοι που βρίσκονται στο Neighbor Set.
- **MPR Set:** Όλα τα επιλεγμένα MPR καταγράφονται σε αυτή την δομή.
- **MPR Selector Set:** Όλοι οι γείτονες που έχουν επιλέξει τον συγκεκριμένο κόμβο ως MPR καταγράφονται σε αυτό την δομή.
- **Topology Information Base:** Αυτή η δομή περιέχει την διαθέσιμη γνώση για όλες τις πληροφορίες σχετικές με την κατάσταση των συνδέσεων.

- **Duplicate Set:** Αυτή η δομή περιέχει πληροφορίες σχετικά με τα μηνύματα που προωθήθηκαν ή επεξεργάστηκαν πρόσφατα.

3.3.3 Χρόνοι λήξης

Οι περισσότερες πληροφορίες που διατηρούνται στις παραπάνω δομές, καταγράφονται μαζί με ένα χρονικό όριο. Αυτή η τιμή υποδεικνύει για πόσο χρόνο θεωρείται έγκυρη η πληροφορία που είναι αποθηκευμένη. Η τιμή της ορίζεται σύμφωνα με τον χρόνο εγκυρότητας που εξάγεται από το μήνυμα από το οποίο αποκτήθηκε η πληροφορία. Η χρήση μίας τέτοιας τιμής εγκυρότητας επιτρέπει την διαφορετική περίοδο αποστολής μηνυμάτων στους κόμβους του δικτύου. Σε περίπτωση που κάποια εγγραφή έχει λήξει, αφαιρείται από την αντίστοιχη δομή.

3.3.4 Μηνύματα ελέγχου

Τα μηνύματα ελέγχου του OLSR μεταδίδονται μέσω UDP στο port 698. Με την χρήση IPv4, η κίνηση γίνεται broadcast στο δίκτυο, όμως η διεύθυνση εκπομπής δεν είναι ορισμένη. Με την χρήση IPv6, απαιτείται η χρήση multicast διεύθυνσης.

Μορφή Πακέτων

Όλη η κίνηση που σχετίζεται με το OLSR, στέλνεται με την χρήση πακέτων OLSR. Τα πακέτα αυτά αποτελούνται από μία κεφαλίδα και ένα σώμα όπως φαίνεται στο σχήμα 3.1. Τα πεδία της κεφαλίδας είναι τα παρακάτω:

- **Packet Length:** Το μήκος πακέτου σε bytes, συμπεριλαμβανομένης της επικεφαλίδας.
- **Packet Sequence Number:** Ένας αριθμός ακολουθίας, που αυξάνεται κατά 1 κάθε φορά που εκπέμπεται ένα μήνυμα OLSR από αυτο τον κόμβο. Για κάθε διεπαφή διατηρείται διαφορετικός τέτοιος αριθμός, έτσι ώστε να υπάρχει διαχωρισμός ανάμεσα στις διεπαφές.

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Packet Length										Packet Sequence Number																					
Message Type					Vtime					Message Size																					
Originator Address																															
Time To Live					Hop Count					Message Sequence Number																					
MESSAGE																															
Message Type					Vtime					Message Size																					
Originator Address																															
Time To Live					Hop Count					Message Sequence Number																					
MESSAGE																															

Σχήμα 3.1: Γενική μορφή πακέτου OLSR

Το σώμα του πακέτου OLSR αποτελείται από ένα ή περισσότερα μηνύματα. Τα μηνύματα αυτά χρησιμοποιούν μία επιπλέον κεφαλίδα η οποία πρέπει οπωσδήποτε να περιέχεται για τον προσδιορισμό του μηνύματος. Τα πεδία σε αυτή την κεφαλίδα είναι τα παρακάτω:

- **Message Type:** Αποτελείται από ένα ακέραιο που προσδιορίζει τον τύπο του μηνύματος. Οι τύποι 0-127 είναι δεσμευμένοι από το OLSR, ενώ οι 128-255 μπορούν να χρησιμοποιηθούν για επεκτάσεις.
- **VTime:** το πεδίο αυτό υποδηλώνει τον χρόνο εγκυρότητας των πληροφοριών που περιέχονται στο μήνυμα.
- **Message Size:** Το μέγεθος του μηνύματος συμπεριλαμβανομένης της κεφαλίδας μηνύματος, σε bytes.
- **Originator Address:** Η κύρια διεύθυνση του κόμβου προέλευσης του μηνύματος.
- **Time To Live:** Ο μέγιστος αριθμός αλμάτων του συγκεκριμένου μηνύματος. Χρησιμοποιώντας αυτό το πεδίο μπορεί να ελεγχθεί ο βαθμός της εξάπλωσης του μηνύματος.
- **Hop Count:** Το πλήθος των προωθήσεων μέχρι την δεδομένη στιγμή.
- **Message Sequence Number:** Ένας αριθμός που αυξάνεται κατά 1 κάθε φορά που κάποιο μήνυμα μεταδίδεται από αυτό τον κόμβο.

Τύποι μηνυμάτων

Η κύρια λειτουργία του OLSR καθορίζει 3 τύπους μηνυμάτων. Όλη η λειτουργία του πρωτοκόλλου βασίζεται σε αυτούς τους 3 τύπους. Όμως, η δομή του πακέτου επιτρέπει την μετάδοση πολλών διαφορετικών πακέτων, που ικανοποιούν τις ανάγκες του δικτύου. Η βελτιστοποίηση του MPR κάνει την δυνατότητα της εξάπλωσης των μηνυμάτων ένα πολύ χρήσιμο εργαλείο για τα δίκτυα που απαιτούν την εξάπλωση των μηνυμάτων σε όλο το ad hoc δίκτυο.

3.3.5 Multipoint relaying

Το OLSR μεταφέρει τα μηνύματα με την χρήση flooding σε όλο το δίκτυο, για να διαδώσει την τοπολογία αυτού. Το flooding σημαίνει ότι όλοι οι κόμβοι επαναμεταδίδουν τα λαμβανόμενα πακέτα. Για την αποφυγή επαναλήψεων, ένας αριθμός ακολουθίας περιέχεται μέσα στο πακέτο. Αυτός ο αριθμός καταγράφεται από τους κόμβους που λαμβάνουν το πακέτο, για να εξασφαλίσουν ότι το πακέτο αναμεταδίδεται μόνο μία φορά. Αν κάποιος κόμβος λάβει ένα πακέτο με αριθμό μικρότερο ή ίσο με αυτόν που έχει καταγραφεί για τον αποστολέα του πακέτου, το πακέτο δεν επαναμεταδίδεται. Στα ενσύρματα δίκτυα προστίθενται άλλες βελτιστοποιήσεις όπως η μη επαναμετάδοση μέσω της διεπαφής μέσω της οποίας έφτασε το πακέτο. Σε ένα ασύρματο δίκτυο όμως, είναι σημαντικό οι κόμβοι να επαναμεταδίδουν τα πακέτα μέσω της διεπαφής από την οποία λήφθηκε το πακέτο, λόγω της φύσης του δικτύου. Αυτό έχει ως συνέπεια για τους επαναμεταδότες, οι οποίοι λαμβάνουν το ίδιο πακέτο λόγω της επαναμετάδοσης από τους γείτονες.

Multipoint relaying

Ο σκοπός του multipoint relaying είναι η μείωση του αριθμού των επαναμεταδόσεων των ίδιων πακέτων κατά την προώθηση αυτών. Αυτή η τεχνική περιορίζει το σύνολο των κόμβων που μπορούν να αναμεταδώσουν τα πακέτα. Το μέγεθος αυτού του υποσυνόλου εξαρτάται από την τοπολογία του δικτύου.

Για να επιτευχθεί το παραπάνω, επιλέγονται κάποιοι κόμβοι ως Multipoint relays (MPRs). Κάθε κόμβος υπολογίζει το δικό του σύνολο από MPRs, το οποίο αποτελείται από συμμετρικούς γείτονες, και μέσω αυτών υπάρχει πρόσβαση σε όλους τους 2-hop γείτονες. Αυτό σημαίνει ότι για κάθε κόμβο στο δίκτυο που μπορεί να επικοινωνήσει με κάποιο δεδομένο κόμβο μετά από 2 άλματα, θα πρέπει να υπάρχει ένα MPR τέτοιο ώστε ο πρώτος κόμβος να έχει μία συμμετρική σύνδεση με το MPR και το MPR με τον δεδομένο κόμβο.

Το OLSR επιτρέπει στους κόμβους να ανακοινώνουν την προθυμία τους να λειτουργήσουν ως MPR για τους γείτονές τους. Καθορίζονται 8 επίπεδα, ξεκινώντας από το χαμηλότερο WILL_NEVER, το οποίο υποδηλώνει ότι ο συγκεκριμένος κόμβος δεν μπορεί να επιλεγεί ως MPR, μέχρι το WILL_ALWAYS, που υποδηλώνει ότι ο κόμβος αυτός θα πρέπει πάντα να επιλέγεται ως MPR. Η προθυμία εξαπλώνεται στους κόμβους μέσω των μηνυμάτων HELLO και λαμβάνεται υπ' όψη κατά την επιλογή των MPR.

Προώθηση κίνησης OLSR

Η αναμετάδοση των μηνυμάτων είναι αυτό που κάνει το flooding εφικτό στα MANETs. Το OLSR καθορίζει ένα αλγόριθμο προώθησης που χρησιμοποιεί της πληροφορίες του MPR για την διάχυση των πακέτων. Βέβαια, δίνεται και η δυνατότητα να εφαρμοστούν διαμορφωμένοι τρόποι προώθησης από το πρωτόκολλο. Τα μηνύματα όμως που περιέχουν κάποιο τύπο που δεν είναι γνωστός από κάποιο δεδομένο κόμβο, αναμεταδίδονται με τον προκαθορισμένο αλγόριθμο. Ο αλγόριθμος περιγράφεται στα παρακάτω βήματα:

1. Αν η σύνδεση μέσω της οποίας έφτασε το πακέτο δεν είναι συμμετρική, το μήνυμα απορρίπτεται σιωπηλά. Για τον έλεγχο της κατάστασης της σύνδεσης γίνεται αναζήτηση στο link set.
2. Αν το TTL που περιέχεται στην κεφαλίδα του μηνύματος είναι 0, το μήνυμα απορρίπτεται σιωπηλά.
3. Αν το μήνυμα έχει ήδη προωθηθεί, τότε απορρίπτεται. Για τον έλεγχο του αν έχει ήδη προωθηθεί, γίνεται αναζήτηση στο duplicate set.
4. Αν το τελευταίο άλμα, έχει επιλέξει τον παρόν κόμβο ως MPR, τότε το μήνυμα προωθείται. Αν όχι, το μήνυμα απορρίπτεται. Για τον έλεγχο γίνεται αναζήτηση στο MPR selector set.
5. Αν το μήνυμα πρόκειται να προωθηθεί, το TTL αυτού μειώνεται κατά 1 και το πλήθος αλμάτων αυξάνεται κατά 1, πριν γίνει η εκπομπή του μηνύματος μέσω όλων των διεπαφών.

Το γεγονός ότι με αυτή την προσέγγιση όλοι οι άγνωστοι τύποι μηνυμάτων προωθούνται, καθιστά την διάχυση ειδικών μηνυμάτων δυνατή, ακόμα και αν αυτοί οι τύποι είναι γνωστοί μόνο σε ένα υποσύνολο των κόμβων.

Για να ελεγχθεί αν ένα μήνυμα έχει επαναμεταδοθεί, διατηρείται μία κρυφή μνήμη με μηνύματα που έχουν επεξεργαστεί και προωθηθεί πρόσφατα. Τα δεδομένα που αποθηκεύονται είναι τα ελάχιστα δυνατά για να αναγνωριστούν τα μηνύματα. Αυτό σημαίνει ότι το πραγματικό περιχόμενο των μηνυμάτων δεν αποθηκεύεται, αλλά μόνο η διεύθυνση προέλευσης, ο τύπος μηνύματος και ο ακολουθιακός αριθμός. Αυτά τα δεδομένα αποθηκεύονται για χρονικό διάστημα ίσο με `DUP_HOLD_TIME`, το οποίο είναι σηνύτως ίσο με 30s. Κάθε μήνυμα που λαμβάνει ο κόμβος, επεξεργάζεται και καταχωρείται στο `duplicate set`. Αν το μήνυμα προωθηθεί, η αντίστοιχη εγγραφή ενημερώνεται κατάλληλα, καταγράφοντας μέσω ποιων διεπαφών προωθήθηκε το μήνυμα. Μέσω της αναζήτησης στο `duplicate set` ο κόμβος μπορεί να εντοπίσει τα ήδη επεξεργασμένα και προωθημένα μηνύματα με κριτήριο την κάθε διεπαφή.

Για την αποφυγή συγκρούσεων πακέτων λόγω συγχρονισμού των μεταδόσεων, εισάγεται jitter κατά την προώθηση μηνυμάτων. Αυτό είναι ένα μικρο τυχαίο χρονικό διάστημα για το οποίο το μήνυμα αποθηκεύεται προσωρινά και αμέσως μετά προωθείται. Με την χρήση αυτής της τεχνικής, μπορεί να γίνει `riggybacking` των μηνυμάτων αφού μπορεί να υπάρχουν πολλαπλά μηνύματα που χρειάζεται να αποσταλούν. Όταν συμβαίνει αυτό, τα μηνύματα ενσωματώνονται σε ένα πακέτο `OLSR`.

Βελτιστοποίηση του link set

Λόγω της φύσης της επιλογής του MPR, μόνο οι κόμβοι που έχουν επιλεγεί ως MPR από κάποιο γείτονα μπορούν να ανακοινώσουν της κατάσταση των συνδέσεών τους. Ουσιαστικά, αυτοί οι κόμβοι πρέπει δηλώσουν τους MPR selectors στα μηνύματα link state. Όταν αυτή η πληροφορία διαχέεται σε όλους τους κόμβους του δικτύου, όλοι οι κόμβοι θα έχουν αρκετές πληροφορίες για να υπολογίσουν την συντομότερη διαδρομή προς όλους τους υπόλοιπους κόμβους. Η προκαθορισμένη ρύθμιση του OLSR είναι να επιτρέπει σε κάθε κόμβο να διαχέει μηνύματα αν είναι επιλεγμένος ως MPR από κάποιο γείτονά του, και μέσα σε αυτά καταγράφει μόνο τους κόμβους που τον έχουν ορίσει MPR.

3.3.6 Χρήση πολλαπλών διεπαφών

Οι κόμβοι που συμμετέχουν στην δρομολόγηση του OLSR μπορούν να χρησιμοποιήσουν περισσότερες από μία από τις διεπαφές τους. Για την γνωστοποίηση των πολλαπλών διεπαφών ενός κόμβου, διαδίδονται στο δίκτυο μηνύματα MID (Multiple Interface Declaration). Ένα τέτοιο μήνυμα είναι μία λίστα με τις διευθύνσεις που χρησιμοποιεί κάθε διεπαφή που λειτουργεί χρησιμοποιώντας το OLSR. Η μορφή του μηνύματος φαίνεται στο σχήμα 3.2.

Όταν ληφθεί ένα μήνυμα τύπου MID, ο κόμβος ενημερώνει την βάση σύμφωνα με τις πληροφορίες που περιέχονται στο μήνυμα. Όλες οι διεπαφές που αναγράφονται στην λίστα καταγράφονται στην κύρια διεύθυνση του κόμβου προέλευσης. Η κύρια διεύθυνση του κόμβου προέλευσης βρίσκεται στο πεδίο `originator`, βάσει του σχήματος 3.1. Όταν γίνεται προσθήκη

0										1										2										3							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1						
OLSR Interface Address																																					
OLSR Interface Address																																					
...																																					

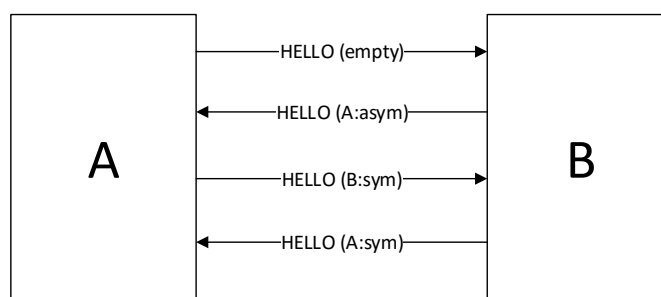
Σχήμα 3.2: Μορφή μηνύματος MID

μίας διαδρομής σε ένα κόμβο, το OLSR θα προσθέσει τις διαδρομές σε όλες τις διευθύνσεις των υπόλοιπων διεπαφών που λειτουργούν με OLSR.

Όλοι οι κόμβοι που τρέχουν με OLSR σε περισσότερες από μία διεπαφές, παράγουν μηνύματα MID σε τακτικά χρονικά διαστήματα. Τα μηνύματα αυτά διαχέονται στο δίκτυο με τον προκαθορισμένο αλγόριθμο που περιγράφηκε πιο πάνω.

3.3.7 Ανακάλυψη γειτόνων

Προφανώς, το OLSR χρειάζεται κάποιου είδους μηχανισμό για την ανίχνευση γειτόνων και της κατάσταση των συνδέσεων επικοινωνίας προς αυτούς. Τα μηνύματα HELLO εκπέμπονται σε τακτικά χρονικά διαστήματα για αυτό τον λόγο. Μία απλοποιημένη αναπαράσταση της ανακάλυψης γειτόνων με την χρήση τέτοιων μηνυμάτων φαίνεται στο σχήμα 3.3. Ο A εκπέμπει αρχικά ένα κενό μήνυμα HELLO. Στην συνέχεια, ο B λαμβάνει το μήνυμα και καταγράφει τον A ως ασύμμετρο γείτονα, επειδή δεν εντόπισε την διεύθυνσή του μέσα στο HELLO μήνυμα που δέχθηκε από τον A. Έπειτα, ο B στέλνει και αυτός ένα HELLO μήνυμα, δηλώνοντας τον A ως ασύμμετρο γείτονα. Όταν ο A λάβει το μήνυμα, εντοπίζει την διεύθυνσή του μέσα σε αυτό και έτσι θέτει τον B ως συμμετρικό γείτονα. Αυτή την φορά ο A περιέχει τον B στο μήνυμα που θα στείλει, επιτρέποντας στον B να καταχωρήσει και αυτός τον A ως συμμετρικό γείτονα.



Σχήμα 3.3: Διαδικασία ανακάλυψης γειτόνων στο OLSR

Τα μηνύματα HELLO εξυπηρετούν και άλλους σκοπούς εκτός από την ανακάλυψη της γειτονιάς ενός κόμβου. Δημιουργούνται και μεταδίδονται σε όλους τους άμεσους (1-hop) γείτονες για την ανίχνευση συνδέσεων, την ανίχνευση γειτόνων, την ανίχνευση 2-hop γειτόνων και την ανίχνευση των MPR selectors.

Στα μηνύματα HELLO οι όμβοι μεταδίδουν πληροφορίες για όλες τις γνωστές συνδέσεις και γείτονες. Επιπλέον δηλώνεται και ο τύπος των γειτόνων. Αυτό συμπεριλαμβάνει του κόμβους που έχουν επιλεγθεί ως MPR από τον κόμβο που στέλνει το μήνυμα. Οι καταγεγραμμένες συνδέσεις και γείτονες ομαδοποιούνται κατά τύπο σύνδεσης και γείτονα, για να βελτιστοποιηθεί το μέγεθος του πακέτου. Είναι πολύ σημαντικό να επισημανθεί ότι τα μηνύματα HELLO παράγονται ανά διεπαφή. Αυτό συμβαίνει επειδή τα μηνύματα αυτά χρησιμοποιούνται για την ανίχνευση συνδέσεων, η οποία πιθανόν να απαιτεί την χρήση πολλαπλών διευθύνσεων.

Η μορφή των μηνυμάτων HELLO φαίνεται στο σχήμα 3.4. Το μήνυμα περιέχεται στο σώμα του OLSR μηνύματος. Το byte link-code περιέχει πληροφορίες για την σύνδεση με τον γείτονα αλλά και το τύπο του γείτονα. Ο τύπος σύνδεσης περιγράφει την κατάσταση της σύνδεσης και ο τύπος γείτονα περιγράφει την κατάσταση του γείτονα, η οποία περιέχει πληροφορίες σχετικά με το MPR. Να σημειωθεί ότι μία σύνδεση μπορεί να θεωρηθεί μη συμμετρική ενώ ο γείτονας θεωρείται συμμετρικός, αν υπάρχουν πολλαπλές συνδέσεις προς αυτόν.

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved										Htime										Willingness											
Link Code					Reserved					Link Message Size																					
Neighbor Interface Address										Neighbor Interface Address																					
										...																					
Reserved										Htime										Willingness											
Link Code					Reserved					Link Message Size																					
Neighbor Interface Address										Neighbor Interface Address																					
										...																					

Σχήμα 3.4: Μορφή HELLO μηνύματος

Ανίχνευση συνδέσεων

Για την διατήρηση επικαιροποιημένων πληροφοριών σχετικά με τις συνδέσεις που υπάρχουν για ένα κόμβο, είναι απαραίτητη η επεξεργασία και αποθήκευση του link set. Στα μηνύματα HELLO ο κόμβος εκπέμπει όλες τις πληροφορίες σχετικά με τις συνδέσεις προς τους γείτονες που υπάρχουν για την διεπαφή από την οποία στέλνεται το μήνυμα. Όταν δηλώνονται συνδέσεις, χρησιμοποιείται η διεύθυνση IP των συγκεκριμένων διεπαφών. Όταν η κατάσταση του γείτονα ορίζεται ως μη προσβάσιμος μέσω της διεπαφής στην οποία δηλώνεται το μήνυμα HELLO, χρησιμοποιείται η κύρια διεύθυνση του γειτονικού κόμβου.

Κατά την λήψη ενός HELLO από κάποιο γείτονα, ο κόμβος ελέγχει αν το μήνυμα περιέχει την IP της διεπαφής μέσω της οποίας λήφθηκε. Στην συνέχεια ενημερώνεται το link set όπως περιγράφεται παρακάτω:

- Αν δεν υπάρχει εγγραφή στο link set, τότε δημιουργείται μία. Η IP προέλευσης ανακτάται από την κεφαλίδα IP του λαμβανόμενου πακέτου. Όταν δημιουργείται κάποια εγγραφή σύνδεσης, δημιουργείται επίσης και μία εγγραφή γείτονα, σε περίπτωση που δεν υπάρχει ήδη.
- Ενημερώνεται ένας μη συμμετρικός μετρητής σύμφωνα με τον χρόνο εγκυρότητας που λήφθηκε στο μήνυμα. Ο μετρητής αποφασίζει το χρονικό διάστημα για το οποίο η σύνδεση θα θεωρείται μη συμμετρική σε περίπτωση που λήξει ο συμμετρικός μετρητής.
- Αν η διεύθυνση της διεπαφής που λαμβάνει το μήνυμα εντοπίζεται μέσα στο HELLO μήνυμα, ο συμμετρικός μετρητής ενημερώνεται και η κατάσταση της σύνδεσης ενημερώνεται αν χρειάζεται. Επίσης, ενημερώνεται και η κατάσταση του εγγραφής για τον γείτονα σύμφωνα με την εγγραφή σύνδεσης.
- Τέλος ο χρόνος διατήρησης της εγγραφής ορίζεται να είναι η μέγιστη τιμή από τις τιμές μη συμμετρικού και συμμετρικού μετρητή.

Ανίχνευση γειτόνων

Η ανίχνευση γειτόνων καταχωρεί εγγραφές στην δομή 1-hop γειτόνων και χρησιμοποιεί μόνο την κύρια διεύθυνση των κόμβων. Οι εγγραφές γειτόνων σχετίζονται άμεσα με τις εγγραφές συνδέσεων. Όταν δημιουργείται μία εγγραφή σύνδεσης, γίνεται αναζήτηση στον πίνακα γειτόνων για την εύρεση της αντίστοιχης εγγραφής γείτονα. Να σημειωθεί ότι η εγγραφή αυτή πρέπει να καταχωρηθεί στην κύρια διεύθυνση του κόμβου. Αν δεν μπορεί να βρεθεί μία τέτοια εγγραφή, τότε δημιουργείται μία νέα εγγραφή γείτονα. Αυτό σημαίνει ότι ενώ ο κόμβος μπορεί να έχει πολλαπλές εγγραφές συνδέσεων προς τον ίδιο κόμβο, υπάρχει μόνο μία εγγραφή γείτονα.

Επιπλέον, η κατάσταση των εγγραφών γειτόνων ενημερώνεται παράλληλα με τις αλλαγές στο link set. Ένας κόμβος αποκαλείται συμμετρικός αν υπάρχει τουλάχιστον μία εγγραφή σύνδεσης στο link set που συνδέει κάποια διεπαφή με μία από τις διεπαφές των γειτόνων όπου ο συμμετρικός μετρητής δεν έχει λήξει. Όταν μία εγγραφή σύνδεσης διαγραφεί, διαγράφεται και η αντίστοιχη εγγραφή γείτονα σε περίπτωση που δεν υπάρχουν άλλες συνδέσεις για αυτόν.

Επιλογή 2-hop γειτόνων

Οι κόμβοι διατηρούν και μία δομή με όλους τους κόμβους που είναι προσβάσιμοι μέσω συμμετρικών γειτόνων. Αυτή η δομή ονομάζεται 2-hop neighbor set και χρησιμοποιείται για τον υπολογισμό των MPRs.

Όταν ληφθεί ένα μήνυμα HELLO από ένα συμμετρικό γείτον, όλοι οι αναφερόμενο συμμετρικοί γείτονες που δεν περιέχουν διευθύνσεις που ανήκουν στον κόμβο, προστίθενται ή

γίνεται η ενημέρωσή τους στο 2-hop set. Οι εγγραφές αυτού του σετ βασίζονται στις κύριες διευθύνσεις, οπότε για όλες τις εγγραφές στο HELLO αναζητείται η αντίστοιχη εγγραφή στο MID set.

Επιλογή MPR

Ο σχεδιασμός του flooding του MPR βασίζεται στην προϋπόθεση ότι όλοι οι κόμβοι έχουν καταγράψει ποιοι κόμβοι λειτουργούν ως MPR για αυτούς. Οι κόμβοι υποδεικνύουν τους γείτονες που έχουν επιλέξει για MPR χρησιμοποιώντας την τιμή MPR_NEIGH στο πεδίο τύπου γείτονα του μηνύματος HELLO.

Κατά την λήψη του HELLO, ο κόμβος ελέγχει τους κόμβους που περιέχονται στο μήνυμα για εγγραφές που ταιριάζουν με κάποια από τις διευθύνσεις του κόμβου. Αν βρεθεί τέτοια εγγραφή και ο γείτονας είναι τύπου MPR_NEIGH, τότε η εγγραφή ενημερώνεται ή δημιουργείται στο MPR selector set χρησιμοποιώντας την κύρια διεύθυνση του αποστολέα του HELLO.

3.3.8 Δήλωση κατάστασης σύνδεσης

Τα πρωτόκολλα που βασίζονται στο Link State Routing, βασίζονται στο flooding πληροφοριών σχετικές με τις συνδέσεις τους στο δίκτυο. Σε πρωτόκολλα όπως το ISIS, αυτές οι πληροφορίες έχουν περισσότερο σχέση με υποδίκτυα, καθώς τα πρωτόκολλα αυτά στηρίζονται στην συσσωμάτωση δικτύων. Το OLSR χρησιμοποιεί flat δρομολόγηση, οπότε η κατάσταση σύνδεσης που εκπέμπεται περιγράφει τις συνδέσεις προς τους γειτονικούς κόμβους. Αυτό γίνεται μέσω των μηνυμάτων Topology Control (TC). Η μορφή ενός τέτοιου μηνύματος φαίνεται στο σχήμα 3.5.

0					1					2					3						
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
ANSN										Reserved											
Advertised Neighbor Main Address																					
Advertised Neighbor Main Address																					
...																					

Σχήμα 3.5: Μορφή TC μηνύματος

Τα μηνύματα αυτά διαχέονται μέσω της βελτιστοποίησης των MPR. Αυτό γίνεται ανά τακτά χρονικά διαστήματα, άλλα τα μηνύματα TC παράγονται και άμεσα όταν εντοπίζονται αλλαγές στο σύνολο των MPR. Στο OLSR η διαδικασία του flooding βελτιστοποιείται με την χρήση των MPR.

Η λειτουργία των MPR παρουσιάζει 2 βελτιστοποιήσεις στην μετάδοση TC μηνυμάτων:

- **Βελτιστοποίηση μεγέθους:** Το μέγεθος των μηνυμάτων μειώνεται επειδή ο κόμβος μπορεί να δηλώσει μόνο τους MPR selectors σε κάθε μήνυμα. Ο παράγοντας αυτής της μείωσης σχετίζεται με την πυκνότητα του δικτύου.
- **Βελτιστοποίηση αποστολέα:** Οι κόμβοι που δεν έχουν συνδέσεις συνήθως δεν εκπέμπουν μηνύματα TC. Αποτελούν εξαίρεση μόνο οι κόμβοι που έχουν χάσει τους MPR selectors. Αυτοί οι κόμβοι παράγουν κενά μηνύματα TC για ένα δεδομένο χρονικό διάστημα για να ενημερωθούν οι κόμβοι του δικτύου. Εκτός από αυτή την ειδική περίπτωση, μόνο οι κόμβοι που έχουν επιλεγθεί ως MPR παράγουν TC μηνύματα. Αυτή η τακτική μειώνει το συνολικό overhead της κίνησης ελέγχου.

Advertised Neighbor Sequence Number(ANSN)

Το ANSN είναι ένας ακολουθιακός αριθμός που σχετίζεται με το neighbor set ενός κόμβου. Ο αριθμός όμως δεν αυξάνεται έπειτα από κάθε παραγωγή μηνύματος TC. Το ANSN αντιπροσωπεύει το πόσο επικαιροποιημένες είναι οι πληροφορίες που περιέχονται στο μήνυμα. Αυτό σημαίνει ότι όποτε κάποιος κόμβος εντοπίζει μία αλλαγή στον γείτονά του, το ANSN αυξάνεται.

Παραγωγή του topology set

Με την λήψη ενός μηνύματος TC, η δομή του TC ενημερώνεται ως εξής:

- Αν δεν υπάρχει κάποια εγγραφή για την διεύθυνση του κόμβου προέλευσης, δημιουργείται μία με χρόνο εγκυρότητας και ANSN σύμφωνα με την κεφαλίδα του TC μηνύματος.
- Αν υπάρχει εγγραφή, η οποία έχει μικρότερο ANSN από το ληφθέν, τότε η εγγραφή ενημερώνεται σύμφωνα με το μήνυμα.
- Αν υπάρχει εγγραφή, η οποία έχει ίσο ANSN με το ληφθέν, τότε ενημερώνεται ο χρόνος εγκυρότητας στην εντίστοιχη εγγραφή.

3.3.9 Υπολογισμός μονοπατιού

Ο προτεινόμενος αλγόριθμος για τον υπολογισμό κάποιου μονοπατιού είναι ο παρακάτω:

1. Αρχικά προστίθονται όλοι οι 1-hop γείτονες στον πίνακα δρομολόγησης.
2. Για κάθε συμμετρικό γείτονα, προστίθονται όλοι οι 2-hop γείτονες οι οποίοι:
 - Δεν έχουν ήδη προστεθεί στον πίνακα δρομολόγησης.
 - Έχουν μία συμμετρική σύνδεση προς τον γείτονα.

Αυτές οι εγγραφές προστίθονται με μήκος μονοπατιού 2 και με επόμενο άλμα τον άμεσο γείτονα.

3. Τότε, για κάθε κόμβο N στον πίνακα δρομολόγησης με πλήθος αλμάτων $n = 2$, προστίθονται όλες οι εγγραφές από το TC set, όπου:

- Ο κόμβος προέλευσης είναι ο N .
- Ο κόμβος προορισμού δεν έχει ήδη προστεθεί στον πίνακα δρομολόγησης.

Οι νέες εγγραφές προστίθενται με πλήθος αλμάτων $n + 1$.

4. Το n αυξάνεται κατά 1 και επαναλαμβάνεται το βήμα 3 μέχρι να μην υπάρχουν εγγραφές στον πίνακα δρομολόγησης με πλήθος αλμάτων $n + 1$.

5. Για όλες τις εγγραφές στον πίνακα δρομολόγησης γίνεται αναζήτηση στο MID set για διευθύνσεις που απευθύνονται στον ίδιο κόμβο. Αν υπάρχουν τέτοιες διευθύνσεις, προστίθενται στον πίνακα δρομολόγησης με το ίδιο πλήθος αλμάτων.

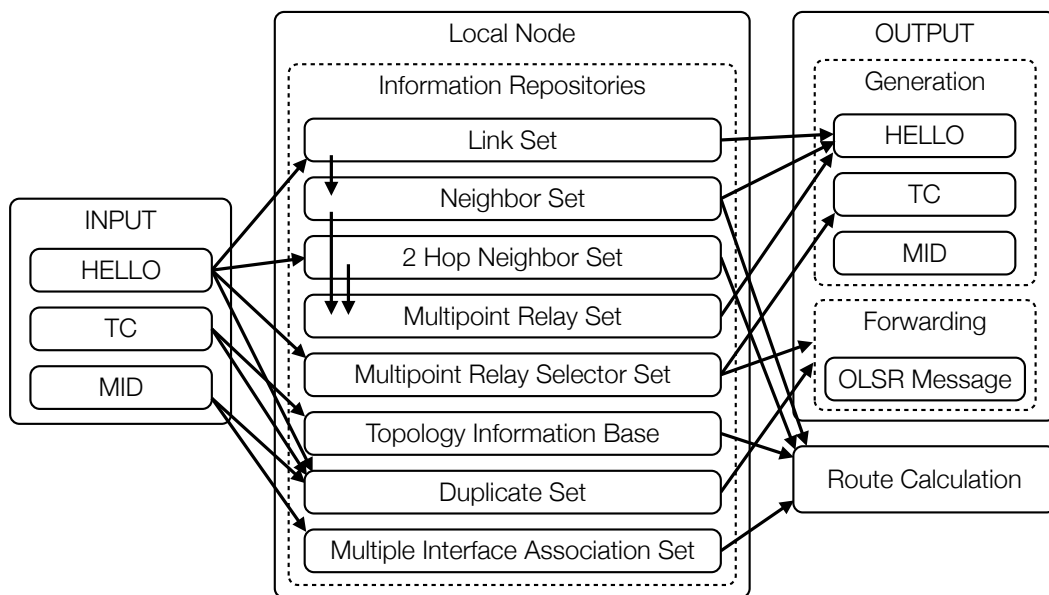
3.3.10 Σύνοψη

Οι λειτουργίες του OLSR μπορούν να χωριστούν σε 3 τμήματα: εντοπισμός γειτόνων, multipoint relaying, και link-state flooding. Το μεγαλύτερο μέρος της κίνησης ελέγχου δημιουργείται από τις δομές που διατηρούνται από το OLSR. Αυτές οι δομές ενημερώνονται δυναμικά βάσει των μηνυμάτων που λαμβάνονται.

Το σχήμα 3.6 δείχνει την δομή του OLSR και τις σχέσεις που έχουν μεταξύ τους όλα τα στοιχεία από τα οποία αποτελείται. Τα μηνύματα HELLO ενημερώνουν το link set το οποίο στην συνέχεια ενημερώνει το neighbor set και στην συνέχεια γίνεται ο υπολογισμός των MPR. Οι 2-hop γείτονες ενημερώνονται βάσει των HELLO μηνυμάτων. Τέλος, το MPR selector set ενημερώνεται σύμφωνα με τις πληροφορίες που περιέχονται στα HELLO μηνύματα. Τα λαμβανόμενα TC μηνύματα επιφέρουν αλλαγές στο topology set, ενώ τα MID επηρεάζουν την κατάσταση του MID set.

Κατά την παραγωγή HELLO μηνυμάτων γίνεται αναζήτηση στο link set, το neighbor set και το MPR set. Για την δημιουργία ενός TC μηνύματος, γίνεται αναζήτηση μόνο στο MPR selector set. Στην προώθηση κίνησης ελέγχου, χρησιμοποιούνται τα MPR selector set και το duplicate set.

Τέλος, ο υπολογισμός των διαδρομών βασίζεται σε πληροφορίες που ανακτώνται από το neighbor set, το 2-hop neighbor set, το TC set και το MID set.



Σχήμα 3.6: Σύνοψη του OLSR. Πηγή [25]

Κεφάλαιο 4

Μετάδοση στα ασύρματα δίκτυα

Για την βελτίωση των ασύρματων δικτύων, είναι απαραίτητη η κατανόηση των χαρακτηριστικών του καναλιού. Ορισμένες λειτουργίες, όπως η ανάθεση καναλιών, ή ο καθορισμός της ισχύος εκπομπής των κόμβων, δεν μπορούν να γίνουν χωρίς την γνώση πληροφοριών που αφορούν το κανάλι. Πληροφορίες που θεωρούνται μεγάλης σημασίας για τους μηχανισμούς που έχουν ως σκοπό την βελτίωση της απόδοσης των δικτύων είναι, για παράδειγμα, το path loss, ο σχεδιασμός του δέκτη, η διαμόρφωση του σήματος, οι παρεμβολές και ο θόρυβος που εισαγεται στο κανάλι.

Στο παρόν κεφάλαιο, μελετάται συνοπτικά η μετάδοση στα ασύρματα δίκτυα σε πολύ χαμηλό επίπεδο. Δίνεται έμφαση στην συμπεριφορά του SNR σε συνάρτηση με τους ρυθμούς μετάδοσης που καθορίζονται από το πρότυπο 802.11n.

4.1 Διάδοση

Σε ένα δέκτη, η ισχύς του σήματος στους ακροδέκτες της κεραίας εξαρτάται από την πυκνότητα ισχύος του ηλεκτρομαγνητικού κύματος που παράγεται από τον πομπό. Η λαμβανόμενη ισχύς P_{Rx} σε Watts γράφεται ως:

$$P_{Rx} = WA \quad (4.1)$$

όπου A είναι η περιοχή επίδρασης της κεραίας, και W είναι η πυκνότητα ισχύος του εκπεμπόμενου ηλεκτρομαγνητικού κύματος. Το W εξαρτάται από την ισχύ που παρέχεται στην κεραία εκπομπής (P_{Tx}), και την απόσταση d ανάμεσα στον πομπό και τον δέκτη. Τα A και W υπολογίζονται θεωρητικά από τους παρακάτω τύπους:

$$A = \frac{\lambda^2 G_{Rx}}{4\pi} \quad (4.2)$$

$$W = \frac{P_{Tx} G_{Tx}}{4\pi d^2} \quad (4.3)$$

όπου G_{Tx} και G_{Rx} είναι τα κέρδη κεραίας του πομπού και δέκτη αντίστοιχα, θεωρώντας πάντα ότι η κεραία εκπέμπει ισότροπα. Η εξίσωση μετάδοσης του Friis περιγράφει την σχέση

μεταξύ της απώλειας διάδοσης, της απόστασης μεταξύ των κεραιών και της συχνότητας που χρησιμοποιείται και είναι η ακόλουθη:

$$P_{Rx} = P_{Tx} G_{Tx} G_{Rx} \left(\frac{\lambda}{4\pi d} \right)^2 \quad (4.4)$$

Ωστόσο, αυτή η εξίσωση δεν είναι ακριβής σε πραγματικά περιβάλλοντα, όπου τα φαινόμενα multipath συναντώνται συχνά λόγω της παρουσίας εμποδίων και της απουσίας οπτικής επαφής (Line of Sight - LOS) των 2 στοιχείων. Για την περιγραφή διαφορετικών συνθηκών καναλιού, υπάρχει διαφορετικά μοντέλα για πιο ακριβή προσέγγιση της μετάδοσης. Η απώλεια διάδοσης, ή αλλιώς path loss (PL), ορίζεται ως η αναλογία της εκπεμπόμενης προς την ληφθείσα ισχύ και εκφράζεται σε dB . Η πιο απλή σχέση για τον υπολογισμό της απώλειας μονοπατιού είναι η

$$PL = P_{Tx_{dBm}} - P_{Rx_{dBm}} \quad (4.5)$$

όπου $P_{Tx_{dBm}}$ είναι η ισχύς μετάδοσης στον πομπό και $P_{Rx_{dBm}}$ η ισχύς με την οποία λαμβάνει ο δέκτης.

Η επόμενη εξίσωση περιγράφει το PL συναρτήσει της συχνότητας f σε MHz και της απόστασης d σε m:

$$PL_{dB} = 20 \log_{10} f + 20 \log_{10} d - 27.6 \quad (4.6)$$

4.1.1 Διάδοση σε εσωτερικό χώρο

Υπάρχει ένα ευρύ φάσμα εσωτερικών μοντέλων διάδοσης, λόγω της μεγάλης ποικιλίας των χαρακτηριστικών των κτιρίων. Διαφορετικά κτίρια μπορεί να έχουν διαφορετικά δομικά υλικά, μεγάλα ή μικρά δωμάτια, υψηλή ή χαμηλή οροφή, έπιπλα, κ.λ.π., και ως εκ τούτου διαφορετικά μοντέλα διάδοσης πρέπει να είναι εφαρμόζεται αναλόγως. Ένα μοντέλο που προτείνεται για την περιγραφή της διάδοσης σε εσωτερικό χώρο, είναι το μοντέλο λογαριθμικής απόστασης, το οποίο ορίζεται ως:

$$PL = PL_0 + 10\gamma \log_{10} \frac{d}{d_0} + X_\sigma \quad (4.7)$$

όπου

PL_0 είναι το μέτρο της απώλειας στην απόσταση αναφοράς d_0

d η απόσταση μεταξύ πομπού και δέκτη,

d_0 η απόσταση αναφοράς (συνήθως $1m$),

γ η σταθερά εξασθένησης,

X_σ είναι μία τυχαία μεταβλητή που ακολουθεί κάποια κατανομή.

Διαφορετικοί δέκτες τοποθετημένοι στην ίδια απόσταση από έναν πομπό θα λάβουν ένα διαφορετικό σήμα που οφείλεται στις διαφορετικές διαδρομές που διανύουν τα ηλεκτρομαγνητικά κύματα. Επιπλέον, το σήμα μεταβάλλεται στο χρόνο ως συνέπεια του δυναμικού περιβάλλοντος. Για αυτούς τους δύο λόγους, στην εξίσωση 4.7 προστίθεται η τυχαία μεταβλητή X_σ , η οποία ακολουθεί μία κανονική κατανομή με τυπική απόκλιση σ . Μετά από εμπειρικές

μετρήσεις, προέκυψε το συμπέρασμα ότι το σ κυμαίνεται μεταξύ 2 και 9 dB, ανάλογα με το περιβάλλον.

Η μεταβλητή X_σ παίζει σημαντικό ρόλο, καθώς σε αυτή αντικατοπτρίζεται η απόσβεση του σήματος λόγω του fading. Σε περίπτωση που δεν υπάρχει fading στο κανάλι, η μεταβλητή είναι ίση με 0. Σε περίπτωση που το κανάλι είναι slow fading, η τυχαία μεταβλητή μπορεί να ακολουθεί μία Gaussian κατανομή. Όταν το κανάλι είναι fast fading τότε ακολουθείται η κατανομή Rayleigh ή Rice. Στα πλαίσια της εργασίας, λαμβάνουμε υπ' όψη ότι το κανάλι δεν χαρακτηρίζεται από αυτό το φαινόμενο, οπότε $X_\sigma = 0$.

Σε αυτή την εξίσωση μπορεί να μεταβληθεί και η παράμετρος γ . Για ένα ανοικτό κτίριο, η προτεινόμενη τιμή του γ είναι $\gamma = 2.2$, για ένα ανοικτό χώρο γραφείου είναι $\gamma = 3.3$. Τέλος, για ένα χώρο γραφείων με μεγάλη πυκνότητα τοίχων και επίπλων, $\gamma = 4.5$.

4.1.2 Διάδοση σε εξωτερικό χώρο

Η εφαρμογή των WLAN σε εξωτερικούς χώρους χαρακτηρίζεται από ύψος της κεραίας λίγων μέτρων, με την απόσταση εκπομπής να κυμαίνεται από μερικές δεκάδες μέτρα μέχρι ένα ή δύο χιλιόμετρα. Τα πρότυπα IEEE 802.11 δεν προορίζονται για μεγάλες αποστάσεις ραδιοζεύξεων και περιορίζουν την ακτίνα των κυψελών σε όχι περισσότερο από 3 km.

Υπάρχουν διάφορα μοντέλα που προέρχονται από γνωστές εμπειρικές προσεγγίσεις που γίνονται στον τομέα της κινητής/κυψελωτής επικοινωνίας. Το μοντέλο Okumura-Hata περιορίζεται σε συχνότητες κάτω από 1500MHz, οι κεραίες τοποθετούνται σε υψόμετρο άνω των 30m. Άλλο γνωστό μοντέλο που χρησιμοποιείται για αυτού του τύπου την διάδοση είναι το COST231-Hata, το οποίο είναι για τις συχνότητες γύρω στα 2GHz. Στα πλαίσια της εργασίας αυτής δεν θα αναλυθούν περαιτέρω ούτε θα χρησιμοποιηθούν μοντέλα για διάδοση σε μεγάλη κλίμακα.

4.2 Λόγος σήματος προς θόρυβο

Κατά την ασύρματη μετάδοση δεδομένων, αντιμετωπίζονται δυσκολίες λόγω των χαρακτηριστικών του ασύρματου μέσου. Η ατμόσφαιρα, η οποία είναι το κανάλι μετάδοσης, προσθέτει θόρυβο στο αποστέλλόμενο σήμα, προκαλώντας έτσι πολλές φορές την αλλοίωση αυτού, και εν συνεχεία της αλλοίωση της πληροφορίας που μεταδίδεται. Ο λόγος σήματος-προς-θόρυβο ή SNR δείχνει πόσο πιο ισχυρό είναι το μεταδιδόμενο σήμα σε σχέση με τον θόρυβο του καναλιού. Ορίζεται ως:

$$SNR = \frac{P_{signal}}{P_{noise}}$$

Είναι καθιερωμένο ο λόγος αυτός να μετράται σε decibel (dB) οπότε για τον υπολογισμό του SNR οπότε η παραπάνω σχέση γίνεται:

$$SNR_{dB} = 10 \log_{10} \left(\frac{P_{signal}}{P_{noise}} \right) \quad (4.8)$$

ή

$$SNR_{dB} = P_{signal_{dB}} - P_{noise_{dB}} \quad (4.9)$$

Το SNR είναι ένα σημαντικό κριτήριο για την αξιολόγηση του καναλιού. Αποτελεί την κύρια ένδειξη τη ποιότητας του, και για αυτό συνδέεται με πολλά χαρακτηριστικά του ασύρματου δικτύου στο φυσικό επίπεδο, όπως την χωρητικότητα του καναλιού και την πιθανότητα σφάλματος κατά την μετάδοση. Και τα 2 προαναφερθέντα χαρακτηριστικά είναι ανάλογα του SNR, οπότε υψηλές τιμές αυτού υπονοούν καλύτερες επιδόσεις στο δίκτυο.

4.3 Μοντέλα σφάλματος

Η πιθανότητα ενός συμβόλου να ληφθεί λανθασμένα, παρουσία AWGN θορύβου δίνεται από την ακόλουθη έκφραση:

$$P_e = Q\left(\sqrt{\frac{d^2}{2N_0}}\right) \quad (4.10)$$

όπου d είναι η ελάχιστη Ευκλείδεια απόσταση μεταξύ δύο οποιωνδήποτε σημείων του αστερισμού της διαμόρφωσης. N_0 είναι η φασματική πυκνότητα θορύβου (σε W/Hz). Η συνάρτηση $Q(x)$ είναι ένας τρόπος για να εκφραστούν οι πιθανότητες για την κανονικές (Gaussian) τυχαίες μεταβλητές. Δεδομένου ότι $x \in \mathfrak{R}$, η $Q(x) \in [0, 1]$ ορίζεται ως η πιθανότητα ότι μια κανονική τυχαία μεταβλητή με μηδενική μέση τιμή και μονάδα διακύμανσης υπερβαίνει το x .

$$Q(x) = \frac{1}{2\pi} \int_x^\infty e^{-(t^2/2)} dt \quad (4.11)$$

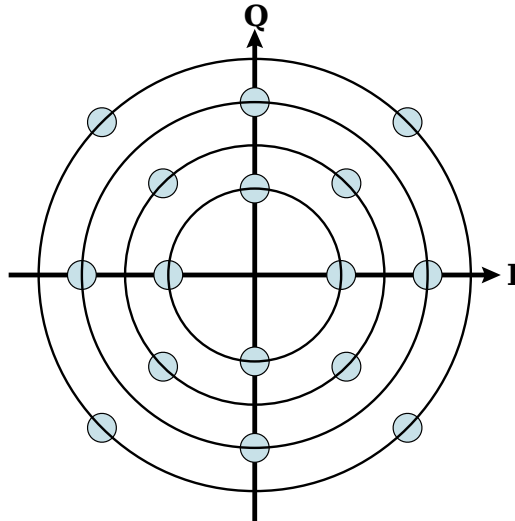
4.3.1 OFDM

Το πρότυπο 802.11n διαθέτει ένα σύνολο data rates τα οποία χρησιμοποιούν διαφορετικό τύπο διαμόρφωσης το καθένα, όπως φαίνεται στον πίνακα 2.1.

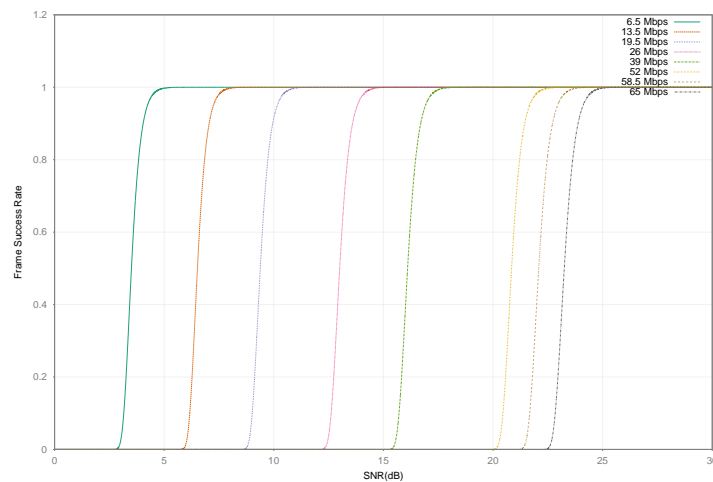
Στο Binart Phase-Shift Keying (BPSK), η φάση ενός σήματος σταθερού πλάτους αλλάζει κατά 180° , αντιστοιχίζοντας το αρχικό και το ολισθημένο σήμα στα 0 και 1. Τα σημεία τοποθετούνται σε απόσταση ίση με $\sqrt{E_b}$ από το κέντρο του επιπέδου στον αστερισμό. Η απόσταση d μεταξύ των 2 συμβόλων είναι $2\sqrt{E_b}$. Αντικαθιστώντας την ποσότητα αυτή στην εξίσωση 4.10, και λαμβάνοντας υπ' όψη ότι ένα σύμβολο του BPSK αντιστοιχεί σε 1 bit, το Bit Error Rate (BER) είναι ίσο με:

$$P_b^{BPSK} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (4.12)$$

Στις M-αδικές διαμορφώσεις PSK, το πλάτος του μεταδιδόμενου σήματος είναι σταθερό. Συνδυάζοντας διαφορετικές φάσεις και πλάτη, μπορούν να σχεδιαστούν περισσότερα σημεία του αστερισμού. Με άλλα λόγια, μπορούν να προστεθούν περισσότερα bits ανά σύμβολο καθώς μειώνεται η απόσταση μεταξύ συμβόλων. Αυτή είναι η βασική αρχή για τον άλλον τύπο



Σχήμα 4.1: Αστερισμός για την διαμόρφωση 16-QAM



Σχήμα 4.2: Πιθανότητα επιτυχούς αποστολής πλαισίου για τα data rates του 802.11n.

διαμόρφωσης που χρησιμοποιείται, το Quadrature Amplitude Modulation (QAM). Ουσιαστικά, οι διαμορφώσεις PSK είναι διαμορφώσεις QAM οι οποίες έχουν ένα σταθερό πλάτος. Συνεπώς τα 2-QAM και 4-QAM αντιστοιχούν στις διαμορφώσεις BPSK και QPSK.

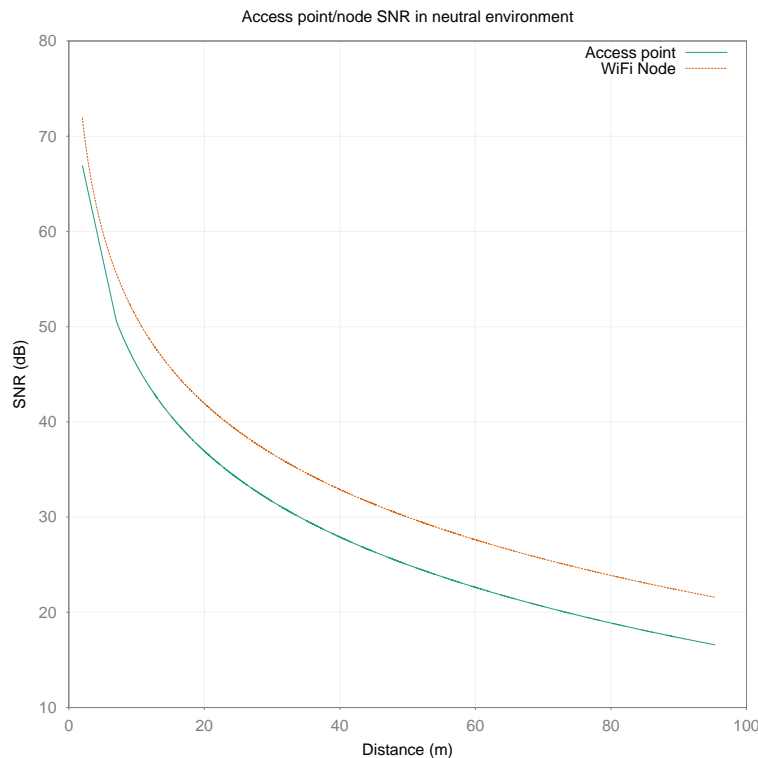
Στην εικόνα 4.2 φαίνεται το Packet Success Rate για κάθε data rate που διαθέτει το 802.11n συναρτήσει του SNR. Χρησιμοποιήθηκε το μοντέλο ns3::NistErrorRateModel του ns-3 για την εξαγωγή των αποτελεσμάτων, το οποίο αποτελεί μία πολύ καλή προσέγγιση στα πραγματικά δεδομένα[17]. Παρατηρούμε ότι όσο αυξάνεται το ο ρυθμός μετάδοσης, λόγω της χρήσης διαφορετικής διαμόρφωσης, απαιτείται μεγαλύτερο SNR για την επιτυχή αποστολή ενός πακέτου.

4.4 Μετάδοση στα WLAN

Για να δούμε την εξασθένιση του σήματος σε ένα WLAN, πραγματοποιήθηκαν πειράματα στον εξομοιωτή NS-3. Το πείραμα αποτελείται από το access point και ένα κόμβο που είναι συνδεδεμένο σε αυτό. Ο κόμβος αρχικά βρίσκεται κοντά στο access point και απομακρύνεται από αυτό με ταχύτητα που αντιπροσωπεύει ένα χρήστη εν ώρα περπατήματος. Οι παράμετροι του πειράματος φαίνονται στο Πίνακα 4.1. Τα αποτελέσματα φαίνονται στο Σχήμα 4.3. Στο πείραμα καταγράφεται το SNR του κόμβου και του access point.

Παράμετρος	Τιμή
Κανάλι	ns3::LogDistancePropagationLossModel
Reference Loss	40 dB
P_{Tx} Access point	25 dBm
P_{Tx} κόμβου	20 dBm
Data Rate	MCS 7 (65 Mbps)
Mobility Model	ns3::ConstantVelocityMobilityModel
Ταχύτητα	1.4 m/s
Packet Size	1500 bytes

Πίνακας 4.1: Παράμετροι για το πείραμα του Σχήματος 4.3.



Σχήμα 4.3: SNR κινητού κόμβου ως προς την απόσταση από το access point.

Από το σχήμα παρατηρείται η αντιστρόφως ανάλογη σχέση της απόστασης του κόμβου με το SNR. Μέχρι η απόσταση του κόμβου από το access point φτάσει περίπου τα 60 m, το SNR στην πλευρά του access point βρίσκεται πιο ψηλά από το όριο που ορίζει η καμπύλη για τα 65 Mbps στο Σχήμα 4.2. Με άλλα λόγια, για αυτές τις συνθήκες του καναλιού η ισχύς μετάδοσης του κόμβου είναι αρκετά μεγαλύτερη της απαραίτητης για να σταλεί το πακέτο χωρίς σφάλματα. Το γεγονός αυτό θα παίζει πρωταρχικό ρόλο στην μηχανισμό που προτείνεται στην παρούσα εργασία. Επιπλέον, μπορεί να καθοριστεί η ακτίνα του access point, η οποία είναι εξαρτημένη από την ισχύ εκπομπής αυτού. Για διάφορες τιμές του P_{Tx} η καμπύλη μετακινείται παράλληλα με τον άξονα του SNR.

Από την παραπάνω ανάλυση, μπορεί να εξαχθεί το συμπέρασμα ότι για κανάλια που είναι σχετικά καθαρά και δεν αντιμετωπίζουν σημαντικά προβλήματα λόγω fading, το path loss είναι η βασική αιτία εξασθένησης του σήματος και από αυτό εξαρτάται το SNR της μετάδοσης. Στην αντίθετη περίπτωση, τα χαρακτηριστικά του fading μπορεί να επηρεάσουν την μετάδοση σε μεγάλο βαθμό και η υλοποίηση μηχανισμών για εξοικονόμηση ενέργειας είναι εφικτοί, αλλά σχετικά δύσκολοι στην κατασκευή.

Κεφάλαιο 5

Συγγενικές εργασίες

5.1 Εισαγωγή

Η εξοικονόμηση ενέργειας είναι ένα μείζον ζήτημα στον τομέα των ασύρματων δικτύων. Η φύση των κινητών ή απομακρυσμένων στατικών κόμβων που υπάρχουν σε αυτά, κρίνουν απαραίτητη την υλοποίηση μηχανισμών, οι οποίοι μέσω διαφορετικής προσέγγισης του προβλήματος καταφέρνουν σημαντικά αποτελέσματα στην οικονομικότερη λειτουργία των δικτύων. Μελέτες για το πρόβλημα έχουν γίνει:

- Στα κλασσικά ασύρματα δίκτυα με υποδομές (Infrastructure WLANs)
- Στα ad hoc δίκτυα
- Στα ασύρματα δίκτυα αισθητήρων (Wireless Sensor Networks - WSNs)

Οι εργασίες αυτές παρέχουν λύσεις για το πρόβλημα, παρατηρώντας το από διαφορετικές σκοπιές. Αποδεικνύεται ότι εξοικονόμηση ενέργειας μπορεί να γίνει όχι μόνο στο φυσικό επίπεδο, αλλά και σε άλλα μέρη της ασύρματης επικοινωνίας, όπως το επίπεδο MAC, ή τον τρόπο λειτουργίας της ασύρματης συσκευής. Στο κεφάλαιο αυτό αναλύεται ο υπάρχων μηχανισμός ενέργειας που διαθέτει το 802.11n, και στην συνέχεια παραθέτονται σχετικές εργασίες από τις παραπάνω κατηγορίες ασύρματων δικτύων.

5.2 Εξοικονόμηση ενέργειας στο 802.11n

Το 802.11n [1][10] διαθέτει ήδη μία στοιχειώδη λειτουργία εξοικονόμησης ενέργειας (Power Save Mode - PSM). Οι κόμβοι μπορούν είτε να βρίσκονται σε κατάσταση εξοικονόμησης ενέργειας, είτε ενεργή. Στην πρώτη κατάσταση, ο πομποδέκτης είναι πρακτικά εκτός λειτουργίας. Η βασική ιδέα της προσέγγισης αυτής είναι ότι τα πλαίσια μπορούν να συσσωρευθούν και να αποσταλούν όλα μαζί στον παραλήπτη. Ανάλογα με την μορφή του δικτύου (Infrastructure/Ad hoc), ο μηχανισμός αυτός διαφοροποιείται σε μερικά σημεία για να μπορεί να αντεπεξέλθει στις συνθήκες επικοινωνίας.

5.2.1 Infrastructure WiFi

Στα Infrastructure δίκτυα, οι κόμβοι ενημερώνουν το access point σχετικά με την κατάσταση στην οποία βρίσκονται. Αυτό το γεγονός δηλώνει πως το access point πρέπει να είναι μόνιμα ενεργό και συνήθως συνδεδεμένο σε κάποια σταθερή πηγή ενέργειας. Το access point είναι υπεύθυνο για τις παρακάτω εργασίες:

- Εφόσον η κατάσταση κάθε κόμβου είναι γνωστή, μπορεί να αποφασιστεί αν ένα πλαίσιο μπορεί να σταλεί άμεσα, ή εναλλακτικά να αποθηκευτεί προσωρινά στην περίπτωση που ο ζητούμενος κόμβος βρίσκεται σε κατάσταση εξοικονόμησης ενέργειας.
- Η προσωρινή αποθήκευση των πακέτων όμως δεν αρκεί για την εύρυθμη λειτουργία του μηχανισμού. Συνεπώς, το access point ‘ανακοινώνει’ στους κόμβους περιοδικά, αν υπάρχουν πακέτα για αποστολή. Από την άλλη μεριά, οι κόμβοι γνωρίζοντας το πότε αποστέλλονται αυτά τα μηνύματα από το access point, μεταβαίνουν στην ενεργή κατάσταση για να ακούσουν το μήνυμα. Σε περίπτωση που υπάρχουν πλαίσια για αποστολή για αυτούς, παραμένουν ενεργή, για να ξεκινήσει η μετάδοση. Ειδικά, επιστρέφουν στην κατάσταση εξοικονόμησης.

Η ‘ανακοίνωση’ access point γίνεται με την χρήση του Traffic Indication Map (TIM). Το TIM είναι ένας χάρτης από bit που αποστέλλεται εντός των beacon frames. Οι κόμβοι είναι υποχρεωμένοι να ενεργοποιούνται κατά την αποστολή ενός τέτοιου frame, για να εξασφαλίσουν αν υπάρχουν πακέτα για αυτούς ή όχι. Μετά από εξέταση του TIM, αν υπάρχουν πλαίσια για αποστολή στον κόμβο, αυτός στέλνει ένα πλαίσιο τύπου PS-Poll στο access point για να ξεκινήσει η αποστολή των πλαισίων.

5.2.2 Ad hoc WiFi

Στα δίκτυα αυτά, όλοι οι κόμβοι είναι ισότιμοι, με αποτέλεσμα η διαδικασία ενημέρωσης να γίνεται περισσότερο πολύπλοκη σε σχέση με τα infrastructure δίκτυα. Εδώ χρησιμοποιείται το πλαίσιο ATIM (Ad hoc Traffic Indication Message) για την ενημέρωση των κόμβων. Μετά από κάθε μετάδοση beacon υπάρχει μία χρονική θυρίδα για την αποστολή τέτοιων μηνυμάτων. Κατά την διάρκεια της θυρίδας αυτής, οι κόμβοι ανταλλάσσουν ATIMs και στην συνέχεια ακολουθεί η διαδικασία που περιγράφηκε στην προηγούμενη περίπτωση.

Η προσέγγιση που ορίζεται από το πρότυπο 802.11n είναι αποτελεσματική, καθώς οι συσκευές μένουν για ένα μεγάλο χρονικό διάστημα ανενεργές εξοικονομώντας μεγάλο μέρος της διαθέσιμης ενέργειας που έχει η συσκευή. Ωστόσο, η προσωρινή αποθήκευση των πακέτων καθώς επίσης και ο τρόπος λειτουργίας των ασύρματων δικτύων με την εφαρμογή του CSMA/CA (Carrier sense multiple access with collision avoidance) και των μηνυμάτων RTS/CTS στα δίκτυα ad hoc, λειτουργεί εις βάρος του throughput του δικτύου και αυξάνει την καθυστέρηση delay σε αυτό [22].

5.3 Λοιπές εργασίες

Αρχετές εργασίες βασίζονται στο επίπεδο MAC και στον παραπάνω σχεδιασμό για να βελτιώσουν την βιωσιμότητα των συσκευών[29][19]. Στο [29], αναπτύσσεται ένας μηχανισμός παρόμοιος με αυτόν της Ενότητας 5.2, όπου ο κόμβος ενεργοποιείται άμεσα όταν θέλει να στείλει κάποιο πλαίσιο το οποίο πρέπει να σταλεί εγκαίρως. Ο μηχανισμός, κάνει τροποποιήσεις στα διαστήματα ενεργοποίησης των κόμβων για να ακούσουν τα μηνύματα TIM. Συγκεκριμένα, ο χρόνος που μένουν ανενεργοί ισούται με ένα πολλαπλάσιο του Beacon Interval (BI), αποτρέποντας έτσι τον κόμβο να σπαταλήσει ενέργεια για να ακούσει το TIM, ενώ δεν υπάρχει πακέτο αποθηκευμένο για αυτόν. Ο μηχανισμός εμφανίζεται γραφικά στο Σχήμα 5.1.

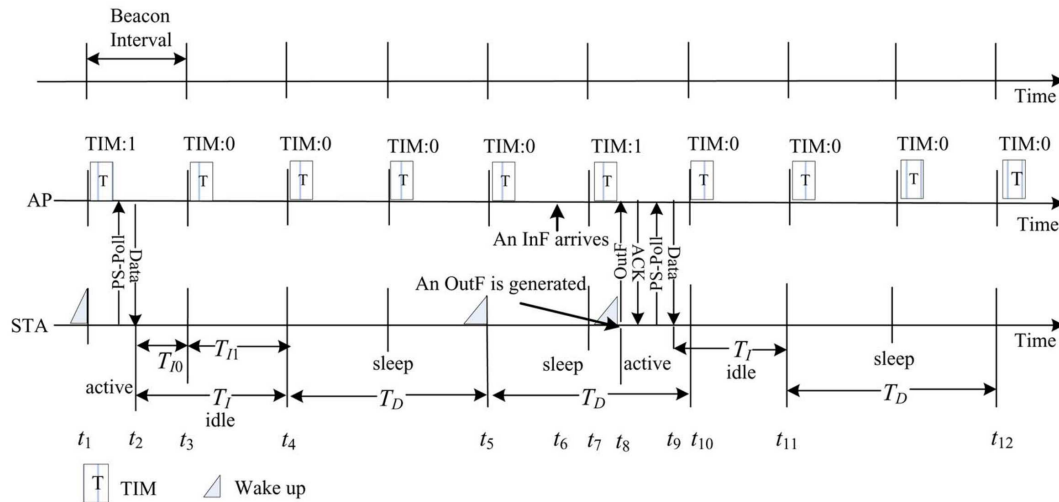
Στην περίπτωση που ο κόμβος θέλει να στείλει ένα πλαίσιο στο access point, βγαίνει άμεσα από την κατάσταση εξοικονόμησης και προχωρά στην αποστολή. Εδώ η διαδικασία χωρίζεται σε 2 περιπτώσεις:

1. Αν δεν υπάρχει πλαίσιο για λήψη από τον κόμβο, αυτός επιστρέφει στην κατάσταση εξοικονόμησης.
2. Αν υπάρχει κάποιο πλαίσιο για λήψη, αλλά δεν έχει ληφθεί επειδή ο κόμβος βρισκόταν σε κατάσταση εξοικονόμησης λόγω της τροποποίησης που έγινε, το access point κάνει piggyback το μήνυμα για την ύπαρξη τέτοιου πλαισίου, οπότε μετά την αποστολή του πλαισίου από τον κόμβο, γίνεται η λήψη του εισερχόμενου.

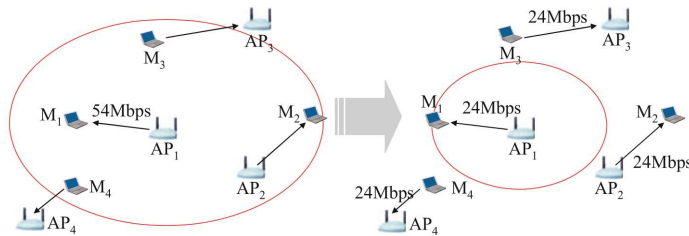
Ο μηχανισμός στο [29] καταφέρνει να μεγιστοποιήσει την ποσότητα ενέργειας που χάνεται ακόμη και από τις τακτικές αφυπνίσεις των κόμβων και επιπλέον βελτιώνει την απόδοση του δικτύου ως προς την καθυστέρηση και το throughput. Ωστόσο, όπως και στον κλασικό σχεδιασμό, το μέγεθος της μνήμης για την προσωρινή αποθήκευση των πλαισίων καθώς επίσης και η διαμορφωμένη περίοδος αδράνειας αποτελούν βασικά κριτήρια για την βελτιστοποίηση του.

Το [23] θεωρεί ως βασικό πρόβλημα την στατική ισχύ μετάδοσης που χαρακτηρίζει τα access points, η οποία μπορεί να φέρει στην επιφάνεια το πρόβλημα του εκτεθειμένου κόμβου (exposed node problem) και να προκαλέσει πτώση στο throughput λόγω των πυκνής παρουσίας δικτύων που πολύ πιθανόν να λειτουργούν στο ίδιο κανάλι. Για αυτό το λόγο, προτείνει ένα αλγόριθμο εξοικονόμησης ενέργειας σε συνδυασμό με ένα αλγόριθμο προσαρμογής ρυθμού μετάδοσης για τα access points, με τον οποίο μειώνεται οι διαμάχες για την δέσμευση του καναλιού, καθώς ελαχιστοποιούνται οι αλληλοκαλύψεις δικτύων χώρο, όπως φαίνεται στο Σχήμα 5.2.

Δημιουργείται ένας κεντρικοποιημένος αλγόριθμος, αποτελούμενος από έναν μία μονάδα ελέγχου, η οποία λαμβάνει πληροφορίες σχετικά με την κατάσταση του κάθε δικτύου από τα access points. Από τις πληροφορίες αυτές, δημιουργεί ένα σύνολο γράφων, οι οποίοι δηλώνουν την σχέση μεταξύ των access points. Δύο access points δεν αλληλοεπηρεάζονται, αν δεν βρίσκονται στον ίδιο γράφο. Συνεπώς εφαρμόζοντας τον μηχανισμό σε κάθε γράφο χωριστά, λαμβάνεται το βέλτιστο αποτέλεσμα ως προς το throughput στο δίκτυο. Ο αλγόριθμος στο [23]



Σχήμα 5.1: Περιγραφή του μηχανισμού στο [29]. Πηγή:[29]

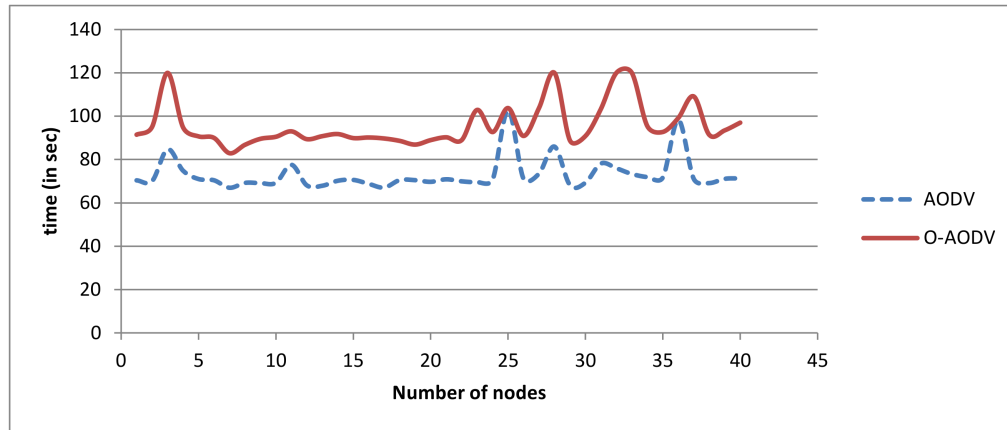


Σχήμα 5.2: Δεξιά: Τα δίκτυα αλληλοκαλύπτονται, με αποτέλεσμα να ενισχύεται το exposed node problem. Αριστερά: Ίδανική περίπτωση με την εφαρμογή του [23]. Πηγή:[23]

αποτελεί φέρει σημαντικά αποτελέσματα ως προς τις επιδόσεις του δικτύου, όμως δεν ωφελεί άμεσα τους κινητούς κόμβους, για τους οποίους η μείωση της σπαταλούμενης ενέργειας για μεταδόσεις είναι ζωτικής σημασίας. Επιπλέον, φαίνεται ότι η εξοικονόμηση ενέργειας μπορεί να αυξήσει τις δυνατότητες του δικτύου, ιδιαίτερα αν λάβουμε υπ' όψη ότι σε μη κεντρικοποιημένα δίκτυα, το πρόβλημα του ανταγωνισμού για την δέσμευση του καναλιού είναι πιο εμφανές.

Η εξοικονόμηση ενέργειας είναι εφικτή ακολουθώντας διαφορετικές προσεγγίσεις. Στα ad hoc δίκτυα, έχουν γίνει πολλές εργασίες για εξοικονόμηση ενέργειας μέσω της βελτιστοποίησης των πρωτοκόλλων δρομολόγησης [21][7][11][14][15][9]. Στο [7] παρουσιάζεται ένας αλγόριθμος που βελτιώνει πρωτόκολλο δρομολόγησης AODV (Ad hoc On Demand Distance Vector). Ο αλγόριθμος βασίζεται στην ιδέα ότι όταν κάθε κόμβος που προωθεί ένα RREQ μήνυμα, μπορεί να μην συμβάλλει στον καθορισμό του τελικού μονοπατιού, δηλαδή να μην είναι ένας ενδιάμεσος κόμβος. Συνεπώς, είναι πολύ πιθανό ο κόμβος να σπαταλήσει ενέργεια χωρίς να είναι αναγκαίο. Κάθε κόμβος γνωρίζει την κατάσταση της υπολειπόμενης ενέργειας που διαθέτει, καθώς επίσης και την πυκνότητα κόμβων στον περίγυρό του. Όταν ένας ενδιάμεσος κόμβος δέχεται ένα μήνυμα RREQ, ελέγχει τα όρια ThB και ThN , που είναι η υπολειπόμενη

ενέργεια και η πυκνότητα κόμβων γύρω του αντίστοιχα. Αν τα κριτήρια αυτά ικανοποιούνται ταυτόχρονα, το μήνυμα προωθείται. Σε περίπτωση που το μόνο πρώτο κριτήριο δεν ικανοποιείται, το μήνυμα δεν προωθείται, ενώ στην αντίστροφη περίπτωση το μήνυμα αποθηκεύεται προσωρινά. Ενδεικτικά αποτελέσματα και σύγκριση με το κλασσικό AODV φαίνονται στο Σχήμα 5.3.



Σχήμα 5.3: Σύγκριση χρόνου ζωής μπαταρίας μεταξύ AODV και O-AODV. Πηγή:[7]

Η βελτιωμένη έκδοση του AODV καταφέρνει να επιμηχύνει τον χρόνο ζωής της συσκευής έως και 50% στην καλύτερη περίπτωση. Εξοικονόμηση ενέργειας μπορεί να γίνει όχι μόνο με την προσαρμογή της ισχύος μετάδοσης, αλλά αξιοποιώντας και άλλα μέρη της ασύρματης επικοινωνίας. Το πρόβλημα με αυτά ωστόσο, είναι ότι η εφαρμογή τέτοιων μηχανισμών δεν είναι γενικευμένη, αλλά πρέπει να ικανοποιούνται συγκεκριμένες συνθήκες για να λειτουργήσουν σωστά.

Στις [13][12][27][20], ακολουθείται μία διαφορετική προσέγγιση για τα ασύρματα δίκτυα αισθητήρων, όπου λαμβάνεται ως κριτήριο χρησιμοποιείται το RSSI (Received Signal Strength Indication) ή άλλα στατιστικά μεγέθη που περιγράφουν την ποιότητα της μετάδοσης. Στο [20], δημιουργείται ένας αλγόριθμος για ανάδρασης, στο οποίο ο παραλήπτης υπολογίζει την βέλτιστη ισχύ μετάδοσης για τον αποστολέα και τον ενημερώνει. Αυτό σημαίνει ότι ο αποστολέας αυξομειώνει την ισχύ μετάδοσης, εξοικονομώντας ενέργεια όποτε είναι αυτό δυνατό. Η διαδικασία για τον αποστολέα και τον παραλήπτη περιγράφονται στον Αλγόριθμο 1.

Σε αυτή την θεώρηση, σημαντικό ρόλο στους υποκείμενους υπολογισμούς παίζει το path loss. Με άλλα λόγια, δίνεται μεγάλη βαρύτητα στα χαρακτηριστικά της μετάδοσης και όχι στις διαδικασίες που ορίζονται από τα πρωτόκολλα επικοινωνίας των ασύρματων δικτύων, όπως συμβαίνει στις προηγούμενες περιπτώσεις. Αυτό έχει ως αποτέλεσμα να εξοικονομείται ενέργεια δραστικά, καθώς για σχετικά μικρές αποστάσεις, οι κόμβοι εκπέμπουν με σημαντικά μικρότερη ισχύ (έως και 99% λιγότερη ισχύς) ενώ παράλληλα το FDR παραμένει αν όχι αναλλοίωτο, σε πολύ υψηλά επίπεδα. Παρόλα αυτά, RSSI ως κριτήριο απόφασης δεν αποτελεί ιδανική επιλογή καθώς μπορεί ορισμένες φορές να δίνει ανακριβείς ενδείξεις, βάσει των [24][16].

Συνοψίζοντας, η εξοικονόμηση ενέργειας είναι ένα πολύ σημαντικό ζήτημα στον τομέα

Algorithm 1 Περιγραφή αλγορίθμου από το [20]

- 1: **Source:**
 - 2: Listen for incoming packets for optimal transmit power
 - 3: Set the transmit power for this receiver
 - 4:
 - 5: **Destination:**
 - 6: Receive data packets from source.
 - 7: Extract the averaged RSSI of the packets from the source node.
 - 8: **if** sender is NOT already in the list of nodes for which the RSSI is known **then**
 - 9: use initial/current transmit power of Sender (20 dBm)
 - 10: **else**
 - 11: still need control packet from sender with record of the initial transmit power, Exit
 - 12: **end if**
 - 13: Calculate the new optimal transmit power $T_{x_{opt}}$
 - 14: Update the lookup table with the newly calculated transmit power $T_{x_{opt}}$
 - 15: Send the new $T_{x_{opt}}$ to sender if RSSI has changed significantly or there is a lack-of-data timeout
-

των ασύρματων δικτύων, του οποίου η επίλυση είναι δύσκολη. Η ενεργειακοί περιορισμοί, τα χαρακτηριστικά της μετάδοσης και του περιβάλλοντος, καθώς επίσης και οι ιδιότητες του ίδιου του δικτύου, δεν κάνουν εύκολη την δημιουργία στατικών ή γενικευμένων αλγορίθμων που θα προσφέρουν λύση στο πρόβλημα. Ίσως οι 2 σημαντικότερες κατηγορίες λύσεων που μπορούν να εφαρμοστούν σε γενικευμένες περιπτώσεις είναι αυτές που σχετίζονται με την λειτουργία του 802.11n, όπου επιδιώκεται η μεγιστοποίηση του χρόνου αδράνειας της συσκευής, και η δυναμική προσαρμογή ισχύος μετάδοσης, όπου για κάθε μετάδοση που γίνεται εξοικονομείται όση ενέργεια είναι δυνατή.

Κεφάλαιο 6

Μηχανισμός προσαρμογής ισχύος μετάδοσης με το SNR ως κριτήριο απόφασης

6.1 Εισαγωγή

Από όλες τις τεχνικές που χρησιμοποιούνται για την εξοικονόμηση ενέργειας, η προσαρμογή της ισχύος μετάδοσης ίσως να είναι η πιο αποδοτική και πρακτική από όλες αυτές που αναφέρθηκαν. Αν όμως δεν επιλεγθούν τα σωστά κριτήρια για την εφαρμογή της, μπορεί να οδηγήσει σε αρνητικά αποτελέσματα. Η πιθανή μείωση της ισχύος συνεπάγεται την μείωση της ακτίνας εκπομπής. Για ασύρματες συσκευές των οποίων η θέση είναι σταθερή αυτό δεν αποτελεί πρόβλημα, σε αντίθεση με τις κινητές όπου η αλλαγή θέσης μπορεί να θέσει τον τελικό προορισμό εκτός εμβέλειας και το πακέτο να χαθεί. Φαίνεται ότι υπάρχει ένα tradeoff μεταξύ κατανάλωσης ενέργειας και Packet Delivery Ratio (PDR). Βασικός στόχος τέτοιων τεχνικών είναι να ελαχιστοποιηθεί το tradeoff αυτό, έτσι ώστε οι απώλειες στην επίδοση να είναι αμελητέες.

Σε αυτό το κεφάλαιο παρουσιάζεται ο αλγόριθμος για εξοικονόμηση ενέργειας που προτείνεται στα πλαίσια της διπλωματικής εργασίας. Βασική μετρική που χρησιμοποιείται για την λήψη αποφάσεων είναι το SNR, το οποίο αξιοποιείται για το καθορισμό της ιδανικής τιμής ισχύος μετάδοσης. Βασικός στόχος του αλγόριθμου είναι η μείωση της κατανάλωσης ενέργειας των κόμβων του δικτύου, χωρίς όμως να υπάρχουν αρνητικές επιδράσεις στην επικοινωνία και χωρίς να χρειάζονται μεταβολές στο πρότυπο επικοινωνίας, δηλαδή το 802.11n.

Για την αξιολόγηση του αλγόριθμου, υλοποιήθηκαν 2 μηχανισμοί, οι οποίοι εφαρμόζονται σε διαφορετικά περιβάλλοντα. Ο πρώτος είναι σχεδιασμένος για εφαρμογή σε δίκτυα με υποδομή (Infrastructure Networks), ενώ ο άλλος μηχανισμός σε ad hoc δίκτυα, ενσωματώνοντας τον στο πρωτόκολλο δρομολόγησης OLSR [5]. Η υλοποιήσεις και τα αποτελέσματα για κάθε μηχανισμό παρουσιάζονται στα Κεφάλαια 7 και 8 αντίστοιχα.

6.2 Βασική ιδέα

Στα ασύρματα δίκτυα, ένα μεγάλο μέρος των μεταδόσεων γίνεται μεταξύ κόμβων των οποίων οι αποστάσεις είναι μικρές. Αυτό υπονοεί το γεγονός ότι ο κόμβος εκπέμπει με μεγαλύτερη ισχύ από την ελάχιστη δυνατή για να μεταδοθεί επιτυχώς το πακέτο. Ο αλγόριθμος προσπαθεί να υπολογίσει την βέλτιστη ισχύ εκπομπής για ένα τέτοιο κόμβο, χωρίς όμως να γνωρίζει την απόσταση από τον παραλήπτη. Επιπλέον, η απόσταση από τον παραλήπτη δεν είναι η μοναδική παράμετρος που πρέπει να λάβουμε υπόψη, καθώς το σήμα μπορεί να εξασθενεί και από άλλους παράγοντες όπως επιπρόσθετος θόρυβος το κανάλι λόγω παρεμβολών. Μία μετρική που περιγράφει όλα αυτά τα φαινόμενα, είναι το SNR.

6.3 Επιλογή του SNR

Στο [20] παρουσιάστηκε ένας αλγόριθμος προσαρμογής ισχύος, ο οποίος χρησιμοποιούσε ως κριτήριο απόφασης το RSSI. Η μετρική αυτή όμως δεν είναι τόσο αξιόπιστη [24] για να βοηθήσει στον υπολογισμό της βέλτιστης ισχύος. Από την άλλη μεριά, το SNR αποτελεί ένα κριτήριο που περιγράφει την γενική εικόνα το καναλιού, κάνοντας πιο εύκολη την επιλογή της κατάλληλης τιμής ισχύος. Βάσει του Σχήματος ;; μπορούμε να ορίσουμε ορισμένα κάτω όρια για τις επιτρεπτές τιμές του SNR, οι οποίες φαίνονται στον Πίνακα 6.1.

Δείκτης MCS	SNR (dB)
0 (6.5 Mbps)	5
1 (13 Mbps)	8
2 (19.5 Mbps)	12
3 (26 Mbps)	15
4 (39 Mbps)	18
5 (52 Mbps)	23
6 (58.5 Mbps)	24
7 (65 Mbps)	25

Πίνακας 6.1: Όρια SNR για επιτυχή μετάδοση

Χρησιμοποιώντας τον Πίνακα 6.1, μπορούμε κάθε φορά έχοντας ως δεδομένο το ρυθμό μετάδοσης των δεδομένων να ορίσουμε ένα SNR_{perf} , το οποίο προσδιορίζει το ελάχιστο SNR που πρέπει να έχει η μετάδοση για να είναι επιτυχής. Σύμφωνα με το [28], το SNR μπορεί να υποδείξει την πιθανότητα με την οποία η μετάδοση μπορεί να γίνει επιτυχώς. Το SNR από τον ορισμό του στην Εξίσωση 4.9, περιέχει την ισχύ λήψης του πακέτου P_{signal} , η οποία εμφανίζεται και στην Εξίσωση 4.5 για την απώλεια διάδοσης. Συνεπώς και η απόσταση μεταξύ πομπού και δέκτη λαμβάνεται υπ' όψη.

6.4 Υπολογισμός ισχύος μετάδοσης

Από την Εξίσωση 4.5 είναι

$$P_{Rx} = P_{Tx} - PL \quad (6.1)$$

βάσει του στόχου του προτεινόμενου αλγόριθμου, το SNR πρέπει να ισούται με κάποια από τις τιμές που παραθέτονται στον Πίνακα 6.1, οπότε αντικαθιστώντας την Εξίσωση 6.1 στην 4.9, έχουμε

$$SNR_{perf} = P_{Tx} - PL - P_{noise} \Rightarrow P_{Tx} = SNR_{perf} + PL + P_{noise} \quad (6.2)$$

Η Εξίσωση 6.2 αποτελεί τον τρόπο υπολογισμού της ιδανικής ισχύος μετάδοσης με βάση το SNR. Προφανώς, σε στατικά δίκτυα μπορεί να εφαρμοστεί με ασφάλεια, αφού το SNR παραμένει αμετάβλητος καθ' όλη την διάρκεια της επικοινωνίας. Εδώ, πρέπει να ληφθεί υπόψη ότι τα δεδομένα που λαμβάνει ο κάθε κόμβος σχετικά με την ισχύ μετάδοσης και τον θόρυβο, περιγράφουν την κατάσταση του καναλιού και την ποιότητα της λήψης σε προηγούμενο χρονικό διάστημα, και όχι στο παρόν. Οπότε, η υπολογισμένη ισχύς μετάδοσης είναι πολύ πιθανό να μην επιφέρει τα επιθυμητά αποτελέσματα, καθώς μπορεί να μην είναι επαρκής για την επιτυχή λήψη του πακέτου από τον παραλήπτη. Για αυτό τον λόγο, αυξάνουμε την ισχύ μετάδοσης με μία σταθερή ποσότητα *Offset*. Η επιλογή της ποσότητας αυτής παίζει σημαντικό ρόλο, καθώς όταν είναι μικρή, εξοικονομείται όσο το δυνατόν περισσότερη ενέργεια χωρίς να δίνεται βάρος στην ποιότητα του καναλιού, ενώ όσο αυξάνεται, αυξάνεται επίσης η ενέργεια αλλά μειώνεται το πακετ λωσς. Συνεπώς, η τελική εξίσωση που προκύπτει είναι η

$$P_{Tx} = SNR_{perf} + PL + P_{noise} + Offset \quad (6.3)$$

Από τις εξισώσεις, παρατηρούμε ότι για τον υπολογισμό του path loss απαιτείται η γνώση της παρούσας ισχύος μετάδοσης του αποστολέα και η ισχύ λήψης στον παραλήπτη. Συνεπώς για να μπορεί ο αποστολέας να υπολογίσει την βέλτιστη ισχύ μετάδοσης, πρέπει να γνωρίζει την ισχύ λήψης στο άλλο άκρο της μετάδοσης και τον θόρυβο. Επιπλέον, πρέπει να ληφθεί υπόψη ότι ο αποστολέας υπολογίζει την τιμή μετάδοσης βάσει των προηγούμενων τιμών των παραμέτρων της εξίσωσης. Λόγω αυτού, δίκτυα με αραιή κίνηση πολύ πιθανόν να παρουσιάζουν προβλήματα, καθώς τα διαστήματα αποστολής και κίνησης χωρίς να έχει ενημερωθεί ο αποστολέας για το P_{Rx} του παραλήπτη είναι μεγάλα, οπότε στην επόμενη μετάδοση ο αποστολέας θα χρησιμοποιεί παρωχημένα δεδομένα για τον υπολογισμό της ισχύος. Αντίθετα, σε δίκτυα με πυκνή κίνηση, συμπεριφέρεται σχεδόν ιδανικά, καθώς οι μεταβολές σε μικρά διαστήματα είναι αμελητέες και δεν επηρεάζουν αρνητικά τον υπολογισμό.

Κεφάλαιο 7

Υλοποίηση μηχανισμού εξοικονόμησης ενέργειας σε δίκτυα υποδομής

7.1 Εισαγωγή

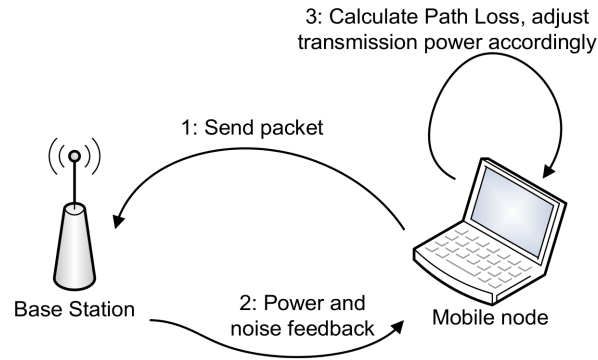
Στο κεφάλαιο αυτό παρουσιάζεται η πρώτη εφαρμογή του αλγορίθμου που παρουσιάστηκε στο Κεφάλαιο 6. Ο μηχανισμός είναι σχεδιασμένος για την εξοικονόμηση ενέργειας σε κεντροποιημένα δίκτυα. Στα δίκτυα αυτά, υπάρχει κάποιο access point ή σταθμός βάσης (base station), από όπου περνούν όλα τα πακέτα που δέχονται ή στέλνουν οι κόμβοι του δικτύου. Η μορφή των δικτύων αυτών, όπου το ένα άκρο είναι στατικό και δεν αλλάζει θέση, αλλά και ο ρόλος που παίζει στην λειτουργία του δικτύου, επιτρέπει ορισμένες αλλαγές στον αλγόριθμο για την καλύτερη επίδοση του.

Για την αξιολόγηση του μηχανισμού έγιναν πειράματα στον Network Simulator 3 (NS-3), όπου έγιναν μετρήσεις στην ισχύ μετάδοσης του κόμβου και το packet loss. Τα πειράματα δείχνουν ότι επιτυγχάνεται αισθητή μείωση της κατανάλωσης ενέργειας της συσκευής ενώ οι απώλειες πακέτων περιορίζονται σε ένα μικρό ποσοστό, το οποίο αυξάνεται ανάλογα με το ρυθμό μετάδοσης που έχει επιλεγεί από τον κόμβο.

7.2 Αρχιτεκτονική του μηχανισμού

Στο Σχήμα Στην περίπτωση αυτή, το δίκτυο αποτελείται από ένα access point και τους κόμβους οι οποίοι είναι συνδεδεμένοι σε αυτό. Υποθέτουμε ότι το access point είναι στατικό και συνδεδεμένο σε σταθερή πηγή ενέργειας. Για αυτό το λόγο, αλλά και για να μεγιστοποιήσουμε την εμβέλεια του δικτύου, θεωρούμε ότι η ισχύς μετάδοσης του είναι προκαθορισμένη και σταθερή με την μέγιστη τιμή ισχύος που μπορεί να πάρει. Κάθε φορά που το access point δέχεται ένα πακέτο από κάποιο κόμβο, καταγράφει την ισχύ λήψης του πακέτου και τον θόρυβο και τα αποθηκεύει σε μία λίστα με τους συνδεδεμένους κόμβους.

Για την ενημέρωση των κόμβων, το access point εκπέμπει περιοδικά ένα μήνυμα στο δίκτυο



Σχήμα 7.1: Περιγραφή του προτεινόμενου μηχανισμού σε δίκτυο υποδομής.

κάθε t_m δευτερόλεπτα, το οποίο περιέχει την αποθηκευμένη λίστα με τους κόμβους για τους οποίους το access point έχει καταγράψει την ισχύ λήψης και τον θόρυβο, όταν έλαβε κάποιο πακέτο από αυτούς. Κάθε εγγραφή αποτελείται από τρία πεδία, όπως στο σχήμα 7.2, την IP διεύθυνση του κόμβου, την ισχύ λήψης του τελευταίου πακέτου που έλαβε το access point από αυτό τον κόμβο και τον αντίστοιχο θόρυβο που διαβάστηκε στο κανάλι. Σε περίπτωση που υπάρχει εγγραφή σε αυτή την λίστα για αυτούς, οι κόμβοι ρυθμίζουν κατάλληλα την ισχύ μετάδοσης βάσει της Εξίσωσης 6.3. Σε διαφορετική περίπτωση εξακολουθούν να εκπέμπουν με την προκαθορισμένη μέγιστη ισχύ. Οι διαδικασίες για το access point και τους κόμβους περιγράφονται στους Αλγόριθμους 2 και 3 αντίστοιχα.

IP Address	P_{Tx}	P_{noise}
------------	----------	-------------

Σχήμα 7.2: Εγγραφή κόμβου στην λίστα του μηχανισμού.

Ο Αλγόριθμος 2 περιγράφει τις ενέργειες που κάνει το access point. Αποτελείται από 2 λειτουργίες, την Packet Received και NotifyNodes. Η πρώτη, είναι υπεύθυνη για την καταγραφή της ισχύος λήψης πακέτων, και η ενημέρωση της λίστας των κόμβων. Η δεύτερη επιτελεί την απλή λειτουργία της εκπομπής ενός πακέτου που περιέχει την λίστα με τους κόμβους και τις απαραίτητες πληροφορίες για να γίνει προσαρμογή ισχύος στους κόμβους.

Ο Αλγόριθμος 3, περιγράφει την διαδικασία που ακολουθείται από την πλευρά των κόμβων, όταν λαμβάνεται το πακέτο του μηχανισμού που περιέχει τις πληροφορίες σχετικά με την μετάδοση. Αφού λάβει το πακέτο, αναζητεί στην λίστα την εγγραφή που αντιστοιχεί σε αυτόν. Αν βρει μία τέτοια εγγραφή, υπολογίζει την νέα ισχύ μετάδοσης χρησιμοποιώντας τα δεδομένα που έλαβε. Στην αντίθετη περίπτωση, σημαίνει ότι ο κόμβος αυτός είτε δεν έχει στείλει ακόμη κάποιο πακέτο προς το access point, είτε βρίσκεται εκτός εμβέλειας του δεύτερου, με αποτέλεσμα κανένα πακέτο που στέλνει να μην λαμβάνεται. Οπότε, ο κόμβος ρυθμίζεται να στέλνει με την μέγιστη δυνατή ισχύ, δηλαδή με MAX_VALUE dBm.

Στην περίπτωση που ο κόμβος βρει την εγγραφή που του αντιστοιχεί στην λίστα, θα υπολογίσει την νέα ισχύ μετάδοσης βάσει της Εξίσωσης 6.3. Επιπλέον, ορίζεται μία τιμή ελάχιστης ισχύος, επειδή όταν ο κόμβος βρίσκεται πολύ κοντά στο access point, η ισχύς

Algorithm 2 Power management mechanism for access point

```

1: function PACKETRECEIVED(packet)
2:    $P_{Rx} = \text{measureRx}(\textit{packet});$ 
3:    $P_{noise} = \text{measureNoise}(\textit{packet});$ 
4:   if Node already exists in the list then
5:     UpdateList( $P_{Rx}, P_{noise}$ );
6:   else
7:     AddNewEntry( $P_{Rx}, P_{noise}$ );
8:   end if
9: end function
10: function NOTIFYNODES
11:   packet = GetList();
12:   BroadcastMessage(packet);
13: end function

```

λήψης μπορεί να είναι πολύ δυνατή, με αποτέλεσμα η υπολογισμένη τιμή να είναι πάρα πολύ μικρή ή αρνητική, το οποίο σημαίνει ότι η νέα ισχύς εκπομπής θα είναι πάρα πολύ μικρή και είναι πολύ πιθανόν να προκληθούν προβλήματα στην επικοινωνία. Για αυτό τον λόγο ορίζεται η τιμή *MIN_VALUE*, και χρησιμοποιείται όταν ο κόμβος υπολογίσει ισχύ εκπομπής μικρότερη της.

Αντίστοιχα, όταν ο κόμβος βρίσκεται πολύ μακριά από το access point η υπολογισμένη τιμή αυξάνεται, και πολλές φορές το αποτέλεσμα είναι μεγαλύτερο από την μέγιστη ισχύ με την οποία μπορεί να εκπέμψει ο κόμβος. Σε αυτή την περίπτωση, αγνοείται πάλι το αποτέλεσμα, και η ισχύς ρυθμίζεται στην μέγιστη δυνατή (δηλ. ίση με *MAX_VALUE*).

Σε κάθε άλλη περίπτωση, το αποτέλεσμα βρίσκεται εντός των αποδεκτών τιμών που μπορεί να πάρει ο κόμβος για να εκπέμψει, οπότε γίνεται προσαρμογή ισχύος με αυτή.

7.3 Υλοποίηση του μηχανισμού

Για την λειτουργία του, ο μηχανισμός απαιτεί στοιχεία από διαφορετικά επίπεδα του ISO/OSI. Στο φυσικό επίπεδο, γίνεται η μέτρηση της λήψης ισχύος, καθώς και ο υπολογισμός του θόρυβου του καναλιού. Οι εγγραφές που διατηρεί το access point για την λειτουργία του μηχανισμού περιέχουν και ένα πεδίο που περιέχει την IP διεύθυνση του αποστολέα, συνεπώς απαιτείται και εξαγωγή πληροφορίας από το επίπεδο δικτύου. Συνεπώς, η υλοποίηση του μηχανισμού πρέπει να ακολουθήσει το cross-layer design.

Cross-layer μηχανισμοί ονομάζονται οι μηχανισμοί που ξεφεύγουν από τα αυστηρά όρια που ορίζονται στο ISO/OSI, τα οποία δηλώνουν ότι δεδομένα που χειρίζονται από ένα επίπεδο πρέπει να παραμένουν σε αυτό. Με άλλα λόγια, τα ανώτερα στρώματα δεν γνωρίζουν καμία πληροφορία για τα κατώτερα, και αντίστροφα. Ο cross-layer μηχανισμός αγνοεί αυτό τον περιορισμό, και επιτρέπει στην μεταφορά πληροφορίας μεταξύ των στρωμάτων, έτσι ώστε να μπορούν να γίνουν βελτιώσεις στο υπάρχον σύστημα.

Algorithm 3 Power management mechanism for network nodes

```

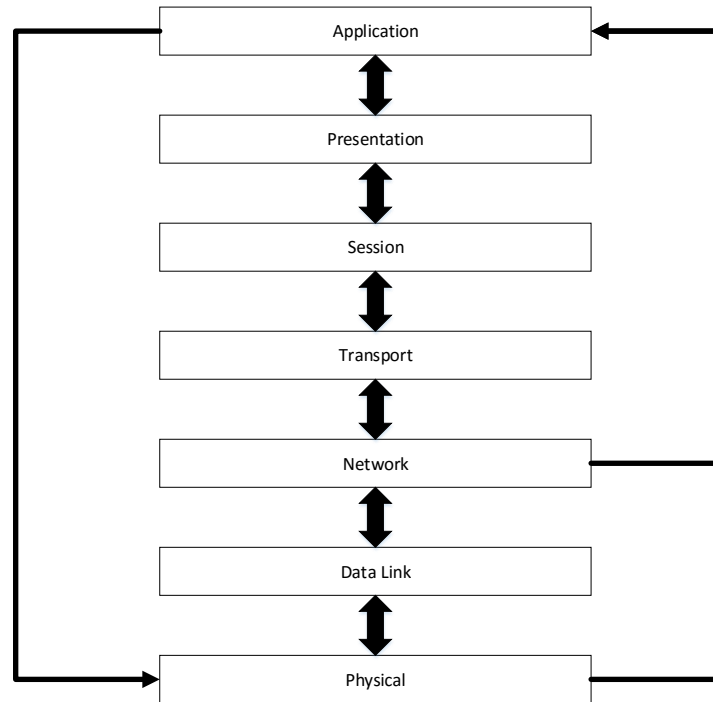
1: function BROADCASTMESSAGERECEIVED(packet)
2:   entry = GetEntryFromList();
3:   if entry  $\neq$  null then
4:      $P_{noise}$  = entry.GetNoise();
5:      $P_{Rx}$  = entry.GetRx();
6:      $P_{Tx_{new}}$  = CalculateTxPower( $SNR_{perf}$ ,  $P_{noise}$ ,  $P_{Rx}$ );
7:     if  $P_{Tx_{new}} < MIN\_VALUE$  then
8:       SetTxPower( $MIN\_VALUE$ );
9:     else if  $P_{Tx_{new}} > MAX\_VALUE$  then
10:      SetTxPower( $MAX\_VALUE$ );
11:     else
12:      SetTxPower( $P_{Tx_{new}}$ );
13:     end if
14:   else
15:     SetTxPower( $MAX\_VALUE$ );
16:   end if
17: end function

```

Ο μηχανισμός της παρούσας εργασίας περιέχει τις επικοινωνίες που εμφανίζονται στο Σχήμα 7.3 απαιτεί πληροφορίες από το φυσικό επίπεδο και από το επίπεδο δικτύου, οι οποίες μεταφέρονται στο επίπεδο εφαρμογών, όπου ουσιαστικά λαμβάνει μέρος ο μηχανισμός. Επιπλέον, το επίπεδο εφαρμογών πρέπει να επικοινωνεί με το φυσικό επίπεδο, προκειμένου να εφαρμοστεί η προσαρμογή ισχύος.

Η υλοποίηση σε πραγματικό περιβάλλον αποτελεί μία μεγάλη πρόκληση. Το λογισμικό που χρησιμοποιείται από τις κάρτες δικτύου συνήθως δεν είναι ελεύθερο για τροποποίηση, με αποτέλεσμα οι απαραίτητες αλλαγές στον κώδικα δεν μην είναι εφικτές. Παρόλα αυτά, στα λειτουργικά συστήματα Linux υπάρχουν διαθέσιμοι οι κώδικες των drivers μέσω του project Linux Wireless [2], οι οποίοι είναι υλοποιημένοι από την κοινότητα του λειτουργικού συστήματος, για να είναι δυνατή η χρήση των καρτών σε αυτή την οικογένεια λειτουργικών συστημάτων. Η υλοποιήσεις όμως είναι φτωχές, και έχουν ως στόχο την λειτουργία της κάρτας, χωρίς να δίνεται έμφαση σε επιπλέον λειτουργίες που μπορεί να υποστηρίζουν οι κάρτες, όπως την μέτρηση του θορύβου του καναλιού ή της ισχύος λήψης, που είναι σημαντική στον μηχανισμό της παρούσας εργασίας. Επιπλέον, όπου υποστηρίζονται οι επιπλέον αυτές λειτουργίες, δεν είναι ομοιόμορφες, δηλαδή δεν χρησιμοποιούν ίδιες διαδικασίες και υπολογισμούς για τις μετρήσεις, με αποτέλεσμα να υπάρχουν διαφορές στις τελικές τιμές που προβάλλονται από την διεπαφή της κάρτας δικτύου.

Από άποψη αξιολόγησης του μηχανισμού, η επανάληψη πειραμάτων είναι δύσκολη, αφού η κίνηση των κόμβων δεν είναι ίδια σε κάθε επανάληψη των πειραμάτων και υπάρχουν εξωτερικοί παράγοντες που μπορεί να επηρεάζουν τα αποτελέσματα διαφορετικά, κάνοντας έτσι μη συγκρίσιμα μεταξύ τους. Για αυτό, επιλέχθηκε να γίνει η αξιολόγηση του μηχανισμού με την



Σχήμα 7.3: Αλληλεπιδράσεις στο μοντέλο OSI με την χρήση του μηχανισμού.

εκτέλεση πειραμάτων στον προσομοιωτή δικτύων NS-3.

7.4 Ο προσομοιωτής δικτύων NS-3

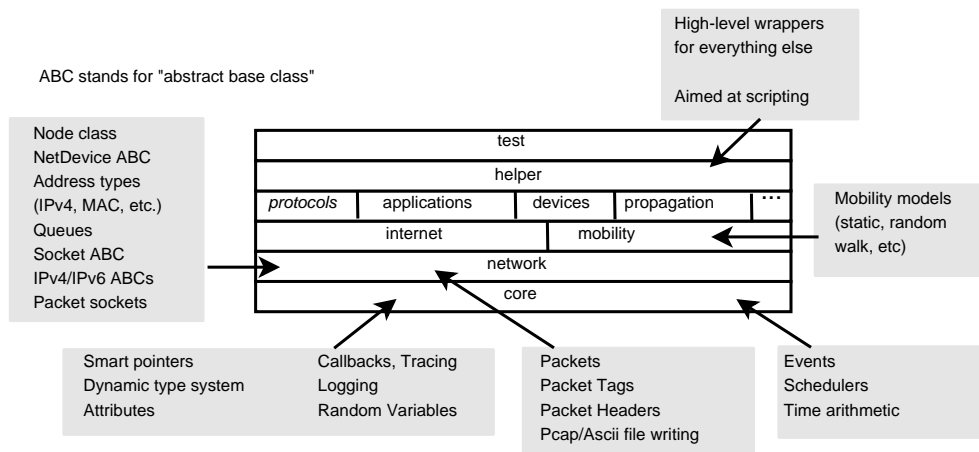
Ο NS-3 [3] είναι ένας προσομοιωτής δικτύου διακριτών γεγονότων με στόχο την έρευνα και την εκπαιδευτική χρήση. Η ανάπτυξη του ξεκίνησε το 2006 με την πιο πρόσφατη έκδοση 3.25 και είναι ένα ανοιχτού κώδικα πρόγραμμα ανάπτυξης. Παρέχει μοντέλα για κάθε επίπεδο του OSI, τα οποία είναι πιστά στην λειτουργία των αντίστοιχων πραγματικών, οδηγώντας έτσι στην παραγωγή ασφαλών και αξιόπιστων αποτελεσμάτων.

Ο πυρήνας προσομοίωσης καθώς επίσης και η βιβλιοθήκη μοντέλων που υποστηρίζονται είναι υλοποιημένες σε C++ και η ιεραρχία τους φαίνεται στο Σχήμα 7.4. Ο κώδικας χωρίζεται στις παρακάτω κατηγορίες:

- **Core:** Στην κατηγορία αυτή υλοποιούνται βασικά στοιχεία που χρησιμοποιούνται σε όλες τις υπόλοιπες μονάδες. Ορίζονται βασικές κλάσεις όπως η Object, η οποία αποτελεί την υπερκλάση όλων των οντοτήτων στον ns-3 και υλοποιούνται συστήματα για την διαχείριση των αντικειμένων, όπως οι Smart Pointers, τα Callbacks και το Attribute system.
- **Network:** Στην κατηγορία Network, αντιστοιχούνται οι κλάσεις που έχουν σχέση με τα δομικά στοιχεία ενός δικτύου τα οποία και περιγράφουν. Η βασική κλάση πάνω στην οποία τοποθετούνται όλες οι άλλες είναι η κλάση Node η οποία αντιπροσωπεύει

τους κόμβους που υπάρχουν στο δίκτυο. Ακολουθούν άλλες βασικές κλάσεις, όπως η Packet, η NetDevice και οι βάσεις για το IPv4.

- **Mobility/Internet:** Σε αυτό το επίπεδο, υλοποιούνται οι κλάσεις που έχουν σχέση με την τοπολογία των πειραμάτων καθώς επίσης υλοποιείται και το Internet Module το οποίο περιέχει τις υλοποιήσεις των στοιχείων που υπάρχουν στην διαδικτυακή επικοινωνία, στα χαμηλά επίπεδα του OSI. Δηλαδή, υλοποιούνται πρωτόκολλα όπως το ARP (Address Resolution protocol) αλλά και οποιοδήποτε συστατικό στοιχείο του IPv4/6.
- **Protocols/Applications/etc.:** Σε αυτή την κατηγορία, κατατάσσεται ο κώδικας που σχετίζεται με την υλοποίηση πρωτοκόλων δρομολόγησης, εφαρμογών καθώς επίσης και των φυσικών μέσων και των χαρακτηριστικών αυτού. Κατά την δημιουργία των πειραμάτων η πλειοψηφία των κλάσεων που χρησιμοποιούνται πηγάζουν από αυτή την κατηγορία.
- **Helpers:** Η κατηγορία αυτή περιέχει ένα πλήθος κλάσεων, οι οποίες χρησιμοποιούνται στην κατασκευή των πειραμάτων και στόχο έχουν την απλοποίηση της δημιουργίας αντικειμένων που έχουν κοινά χαρακτηριστικά. Για παράδειγμα, αντί να δημιουργηθούν οι κάρτες δικτύου ρητά με την δημιουργία στιγμιοτύπων μέσω ενός βρόχου, είναι δυνατό να χρησιμοποιήσουμε τον αντίστοιχο helper, μέσω του οποίου καθορίζουμε τα χαρακτηριστικά και στην συνέχεια με μία εντολή γίνεται η αυτόματη δημιουργία των αντικειμένων. Ο στόχος των κλάσεων αυτών είναι να βοηθήσει τον χρήστη/ερευνητή να αποφύγει τα προγραμματιστικά προβλήματα και δυσκολίες και να εστιάσει στα θέματα που θέλει να περιγράψει μέσω του πειράματος.



Σχήμα 7.4: Οργάνωση του ns-3. Πηγή:[4]

Βασικό στοιχείο για την δημιουργία των πειραμάτων αποτελεί η κλάση ns3::Node, η οποία αντιπροσωπεύει τους κόμβους του δικτύου. Διαισθητικά, η κλάση αυτή λειτουργεί ως ένα κουτί μέσα στο οποίο εκχωρούνται όλα τα υπόλοιπα στοιχεία που συμβάλλουν στην επικοινωνία. Ξεκινώντας από κάτω προς τα επάνω, ο κόμβος πρέπει να συνδεθεί με ένα κανάλι. Το κανάλι

αποτελεί μία ανεξάρτητη οντότητα πάνω στην οποία συνδέονται οι κόμβοι, και περιγράφει το φυσικό μέσο μέσω του οποίου γίνεται η επικοινωνία. Βασική κλάση είναι η `ns3::WifiChannel`, και σε αυτό καθορίζονται διάφορα χαρακτηριστικά, όπως το μοντέλο καθυστέρησης διάδοσης και το μοντέλο απώλειών διάδοσης. Επόμενο στοιχείο που πρέπει να δημιουργηθεί είναι το φυσικό επίπεδο, της οποίας βασική κλάση είναι η `ns3::WifiPhy`. Τέλος, απαραίτητο είναι να δημιουργηθεί ένα αντικείμενο της κλάσης `ns3::WifiMac`, το οποίο αποτελεί την υλοποίηση του Data Link Layer. Όλα τα παραπάνω στοιχεία συνδυάζονται για να δημιουργήσουν ένα αντικείμενο της κλάσης `ns3::WifiNetDevice`, το οποίο αποτελεί την συσκευή επικοινωνίας του κόμβου στο δίκτυο.

Για την διενέργεια πειράματος, πρέπει να δημιουργηθεί κάποιο αρχείο κώδικα C++, που ονομάζεται σενάριο. Στο σενάριο περιγράφονται τα χαρακτηριστικά του πειράματος χρησιμοποιώντας τις κλάσεις και τα εργαλεία που προσφέρει το ns-3. Εκτός από τα προαναφερθέντα, στον ns-3 δίνεται η δυνατότητα να ορίζουμε την τοπολογία του δικτύου και την κίνηση των κόμβων αυτού, σε περίπτωση που πρόκειται για πείραμα που περιέχει ασύρματα δίκτυα. Επιπλέον, καθορίζεται και η κίνηση που επιθυμείται στο δίκτυο, μέσω των κλάσεων εφαρμογών που ορίζονται από τον ns-3.

Ο σχεδιασμός και η οργάνωση του ns-3 είναι τέτοια που καθιστά την επέκταση του κώδικα των δυνατοτήτων που παρέχονται πολύ εύκολη, χρησιμοποιώντας τις βασικές κλάσεις που περιγράφουν τα διάφορα μέρη της ασύρματης επικοινωνίας, μπορούν να δημιουργηθούν νέα μοντέλα βάσει των αναγκών του ερευνητή, ή να γίνει επέκταση των υπάρχοντων. Παρόλα αυτά όμως, ορισμένες φορές είναι επίπονη διαδικασία η συγγραφή νέων στοιχείων, ειδικά αν πρόκειται για επεκτάσεις στα υπάρχοντα μέρη, και οι αλλαγές που γίνονται δεν επηρεάζουν τη υπόλοιπη λειτουργία των κλάσεων.

Για να δοθεί λύση σε αυτό το πρόβλημα, στο ns-3 είναι υλοποιημένο ένα υποσύστημα παρακολούθησης, το Tracing. Βασίζεται στα trace sources, τα οποία είναι οντότητες που μπορούν να σηματοδοτούν την παρουσία ενός γεγονότος. Για παράδειγμα, ένα trace source μπορεί να υποδείξει ότι ένα πακέτο λήφθηκε στο φυσικό επίπεδο. Τα trace sources όμως δεν παρέχουν καμία χρησιμότητα από μόνα τους. Για να αξιοποιηθούν, πρέπει να ο χρήστης να συνδέσει κάποια συνάρτηση στο trace source, έτσι ώστε όταν ενεργοποιείται, να γίνεται η κλήση της συνάρτησης. Μέσω της συνάρτησης, δίνεται η δυνατότητα να καταγραφούν δεδομένα, αλλά και να γίνουν ενέργειες στα αντικείμενα που υπάρχουν στο περιβάλλον μέσα από το οποίο γίνεται η κλήση του trace. Για παράδειγμα, όταν ενεργοποιείται το trace source που σημαίνει την λήψη ενός πακέτου στο φυσικό επίπεδο, μπορούμε να υλοποιήσουμε μία συνάρτηση που συνδέεται στο trace αυτό, και να καταγράψει την ισχύ λήψης του πακέτου και το SNR, όπως απαιτείται και από τον μηχανισμό της εργασίας.

Συμπεραίνεται ότι με την χρήση των trace sources μπορεί να γίνει επέκταση των υπάρχοντων στοιχείων πολύ εύκολα. Επιπλέον, η συλλογή δεδομένων δεν αποτελεί πρόβλημα για τον ερευνητή, ενώ αποφεύγονται τα προβλήματα συμβατότητας με νεότερες εκδόσεις, αφού οι αλλαγές/επεμβάσεις περιορίζονται στο αρχείο εκτέλεσης του σεναρίου, και όχι στον κύριο κώδικα του ns-3.

Στην παρούσα διπλωματική εργασία, οι μηχανισμοί που περιγράφονται υλοποιούνται προ-

γραμματιστικά με την χρήση trace sources, οι οποίοι παρεμβαίνουν στα αντίστοιχα επίπεδα του OSI, όπως περιγράφηκε στο Σχήμα 7.3. Οι κύριοι λόγοι επιλογής είναι προφανώς η ευκολία υλοποίησης, καθώς επίσης και η ευκολία αναπαραγωγής των πειραμάτων από τρίτους.

7.5 Πειραματική αξιολόγηση

Για τη αξιολόγηση του μηχανισμού, έγιναν πειράματα στον ns-3, των οποίων τα χαρακτηριστικά περιγράφονται στον Πίνακα 7.1. Από την ανάλυση του μηχανισμού στην Ενότητα 7.2, φαίνεται ότι σημαντικό ρόλο στην επίδοση του έχουν:

- Το data rate, αφού λόγω αυτού μετατοπίζεται το κάτω όριο του SNR.
- Η περίοδος αποστολής μηνυμάτων ενημέρωσης από το access point.
- Το offset κατά τον υπολογισμό της νέας ισχύος μετάδοσης.

Τα κριτήρια αξιολόγησης του μηχανισμού είναι η μέση ισχύς μετάδοσης, στην οποία αντικατοπτρίζεται ο στόχος του μηχανισμού, αλλά και το packet loss, από όπου μπορούν να εξαχθούν συμπεράσματα για την επιρροή του μηχανισμού στην ποιότητα μετάδοσης.

Χαρακτηριστικό	Τιμή
Πλήθος κόμβων	1
Συχνότητα δειγματοληψίας ισχύος μετάδοσης	0.1s
Συχνότητα ενημέρωσης κόμβων	0.5 : 0.5 : 2s
Offset μηχανισμού	1 : 1 : 4dBm
Ρυθμός Μετάδοσης (Data Rate)	MCS 0 – 7
Πλήθος πακέτων	9900
Ρυθμός αποστολής πακέτων	0.01s
Διάρκεια πειράματος	100s

Πίνακας 7.1: Χαρακτηριστικά πειραμάτων για την αξιολόγηση του πρώτου μηχανισμού.

Η δομή του μηχανισμού δεν επηρεάζεται από το πλήθος των κόμβων, συνεπώς τα πειράματα έγιναν με 2 κόμβους. Ο πρώτος κόμβος αντιπροσωπεύει το access point, ενώ ο δεύτερος έναν απλό κόμβο που συνδέεται στο access point. Πειράματα γίνονται σε όλα τα διαθέσιμα data rates που υπάρχουν στο 802.11n. Για κάθε rate έγιναν επαναλήψεις του πειράματος, όπου σε κάθε επανάληψη η περίοδος ενημέρωσης και το offset είναι διαφορετικό. Τα πειράματα ομαδοποιούνται βάσει της τιμής του offset, και σε κάθε ομάδα συγκρίνονται τα αποτελέσματα στα αντίστοιχα rates με διαφορετικές περιόδους ενημέρωσης.

Σχετικά με τον σχεδιασμό του σεναρίου, οι παράμετροι των στοιχείων που δημιουργούνται (κανάλι, συσκευές επικοινωνίας, κίνηση) διευκρινίζονται στον Πίνακα 7.2.

Για το κανάλι χρησιμοποιήσαμε το μοντέλο λογαριθμικής απόστασης, με reference loss 40dB με διάδοση σταθερής ταχύτητας. Για το access point η ισχύς μετάδοσης είναι σταθερή στα 25dBm, υποθέτοντας ότι είναι συνδεδεμένο σε σταθερή πηγή ενέργειας. Ο ρυθμός

Παράμετρος	Χαρακτηριστικά
Κανάλι	ns3::LogDistancePropagationLossModel <i>ReferenceLoss = 40dB</i> ns3::ConstantSpeedPropagationDelayModel
Φυσικό επίπεδο	ns3::YansWifiPhy <i>TxPowerStart/TxPowerEnd(AccessPoint) = 25dBm</i> <i>TxPowerStart/TxPowerEnd(Node) = 5 - 20dBm</i>
Wifi Standard	<i>WIFI_PHY_STANDARD_80211n_2_4GHZ</i>
Data rate manager	<i>ns3 :: ConstantRateWifiManager</i>
MAC Layer	<i>AccessPoint : ns3 :: ApWifiMac</i> <i>QosSupported = false</i> <i>Node : ns3 :: StaWifiMac</i> <i>QosSupported = false</i>
Mobility	Access Point: <i>ns3 :: ConstantPositionMobilityModel</i> Node: <i>ns3 :: RandomWalk2dMobilityModel</i> <i>Bounds = (-60, 60, -60 - 60)</i> <i>Speed = 1.2 - 1.4m/s</i>

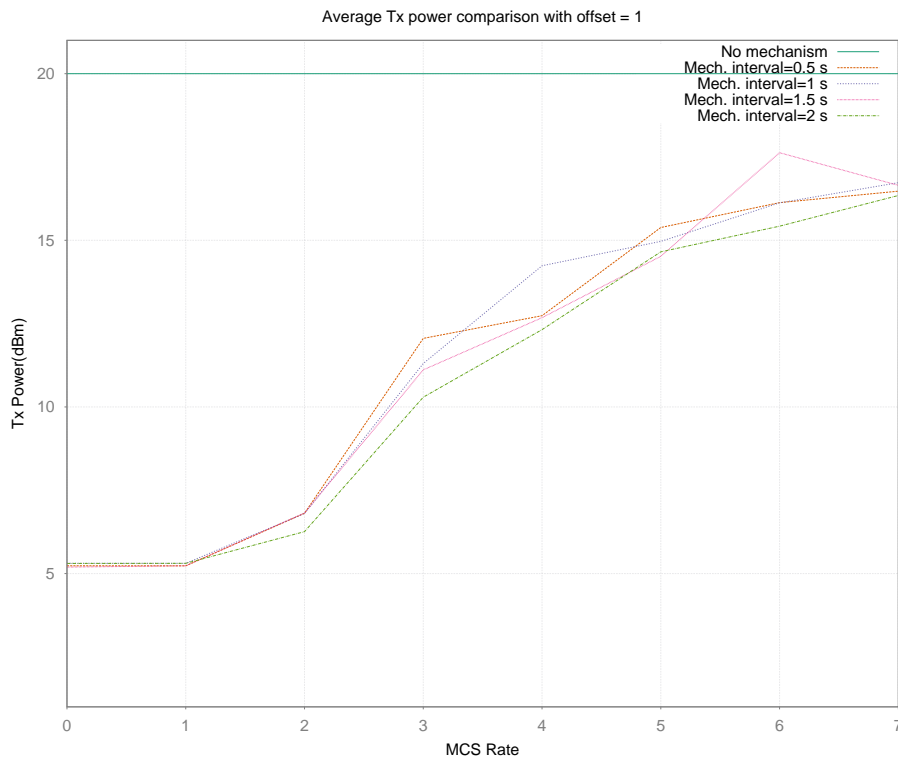
Πίνακας 7.2: Παράμετροι περιβάλλοντος για τα πειράματα του πρώτου μηχανισμού.

μετάδοσης που χρησιμοποιείται είναι σταθερός καθ' όλη την διάρκεια του πειράματος, και δεν χρησιμοποιείται κανένας αλγόριθμος προσαρμογής ρυθμού μετάδοσης. Το access point βρίσκεται σταθερά σε μία θέση, ενώ ο κόμβος κινείται τυχαία εντός της περιοχής του δικτύου.

7.5.1 Αξιολόγηση ως προς την ισχύ μετάδοσης

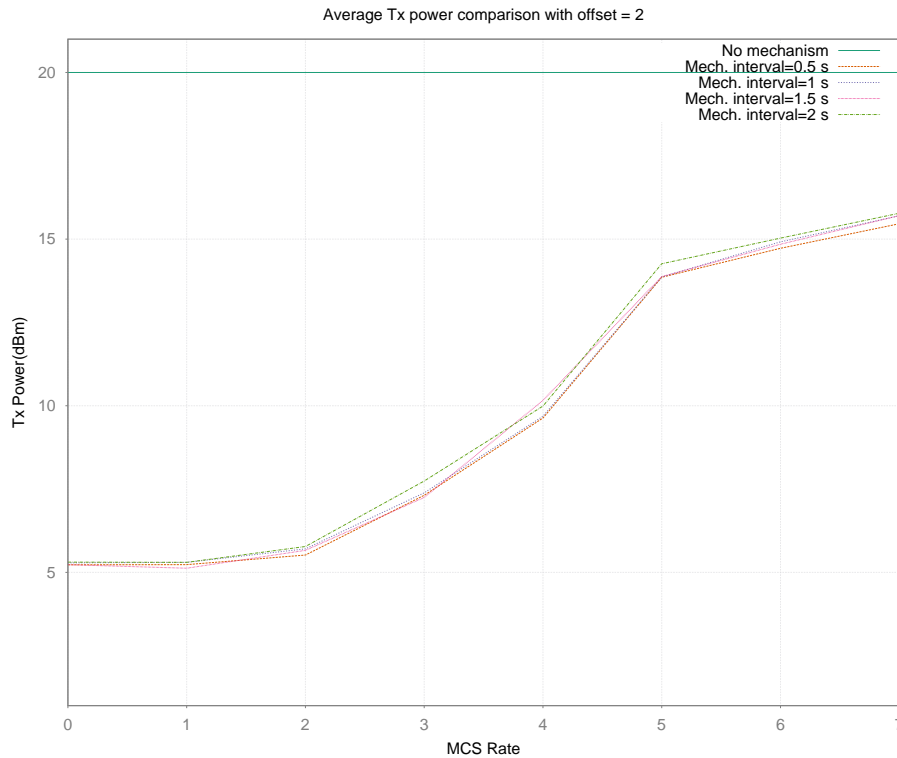
Στην υποενότητα αυτή παρατίθενται τα αποτελέσματα των πειραμάτων που αφορούν την ισχύ μετάδοσης του κόμβου. Οι γραφικές παραστάσεις προβάλλουν τα αποτελέσματα ως προς τον ρυθμό μετάδοσης, ενώ κάθε καμπύλη αντιπροσωπεύει διαφορετικό χρόνο ενημέρωσης.

Στο Σχήμα 7.5 φαίνονται τα αποτελέσματα όταν η τιμή του offset είναι 1, δηλαδή η υπολογισμένη ισχύς προσαυξάνεται κατά 1. Παρατηρούμε ότι στα 2 πρώτα rates, η περίοδος ενημέρωσης δεν προκαλεί διαφορές στην μέση ισχύ. Καθώς αυξάνεται ο ρυθμός μετάδοσης, παρατηρείται ότι υπάρχουν μικρές διαφορές μεταξύ των στιγμιοτύπων, ενώ μικρότερη κατανάλωση ενέργειας εμφανίζουν τα στιγμιότυπα του μηχανισμού που έχουν μεγαλύτερη περίοδο ενημέρωσης. Η μέγιστη ισχύς μετάδοσης που επιτυγχάνεται σε όλα τα πειράματα αυτής της ομάδας είναι τα 17.6dBm τα οποία σε σχέση με τα 20dBm , που είναι η μέγιστη δυνατή ισχύς στο κόμβου, επιτυγχάνουν να εξοικονομήσουν ένα μεγάλο ποσοστό της αρχικής ενέργειας. Συγκεκριμένα, κάνοντας τους υπολογισμούς για την μετατροπή των ποσοστών σε mW , προκύπτει ότι ενώ αρχικά η μέση ισχύς μετάδοσης ήταν στα $100mW$, με την χρήση του μηχανισμού φτάνει έως τα $57.5mW$. Συνεπώς, με την χρήση του μηχανισμού οι συσκευές μπορούν να εκμπεύουν με τουλάχιστον 43% λιγότερη ισχύ.



Σχήμα 7.5: Μέση ισχύς μετάδοσης για $offset = 1$

Στα Σχήματα 7.6-7.8 φαίνονται τα αποτελέσματα για τις ομάδες πειραμάτων με $offset = 2, 3$ και 4 αντίστοιχα.

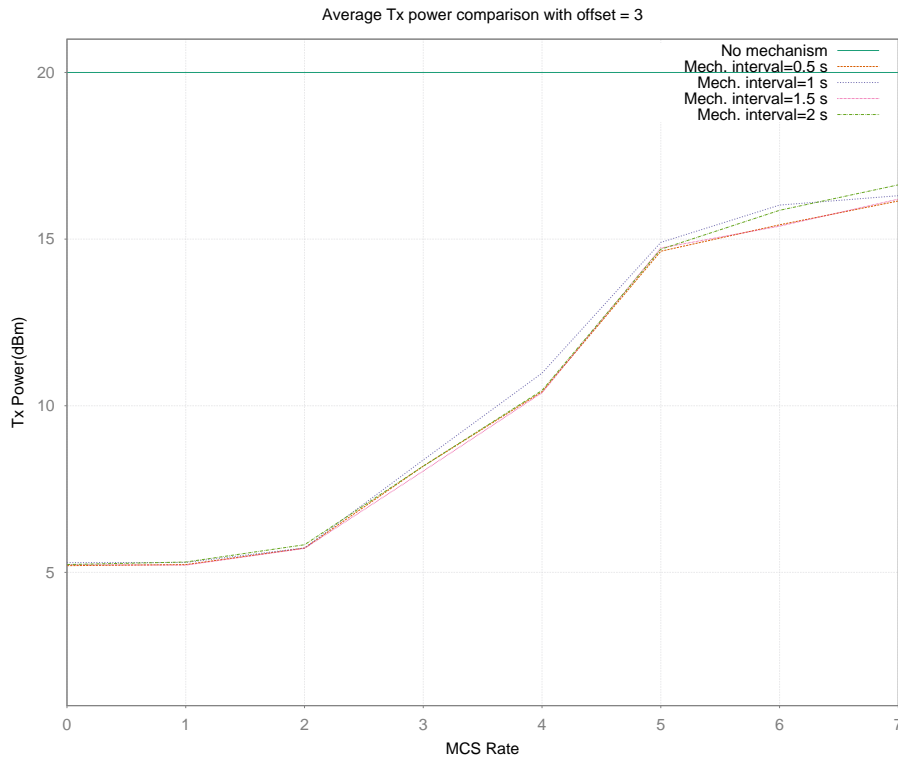


Σχήμα 7.6: Μέση ισχύς μετάδοσης για $offset = 2$

Άμεση παρατήρηση από τις γραφικές παραστάσεις είναι ότι η μέση ισχύς μετάδοσης είναι σχεδόν ίδια για όλες τις περιόδους ενημέρωσης. Ενώ αυξάνεται η τιμή προσαύξησης της υπολογισμένης ισχύος, η αντίστοιχη καμπύλη αυξάνονται ανάλογα. Στην ομάδα πειραμάτων με $offset = 2$, η μέγιστη ισχύς μετάδοσης είναι $15.7dBm$, το οποίο αντιστοιχεί σε 63% λιγότερη ισχύ. Αντίστοιχα, για $offset = 3$, η μέγιστη ισχύς είναι $16.6dBm$, που αντιστοιχεί σε μείωση 55%. Τέλος, για $offset = 4$, η μέγιστη ισχύς είναι $17.9dBm$, το οποίο αντιστοιχεί σε 39% μείωση ισχύος.

Για όλες τις ομάδες πειραμάτων, παρατηρούμε ότι για τα μικρά rates οι παράμετροι του μηχανισμού έχουν σημαντικό ρόλο. Ο κύριος παράγοντας που επηρεάζει τον μηχανισμό είναι το rate, το οποίο επειδή έχει πολύ μικρό όριο SNR, η υπολογισθείσα τιμή είναι μικρότερη από την ελάχιστη ισχύ μετάδοσης, με αποτέλεσμα η μέση ισχύς να παραμένει σε πάρα πολύ χαμηλά επίπεδα. Καθώς το rate αυξάνεται, η αύξηση της μέσης ισχύος είναι φυσιολογική, καθώς αυξάνεται το όριο του SNR, οπότε απαιτείται μεγαλύτερη ισχύς για την επιτυχή αποστολή πακέτων.

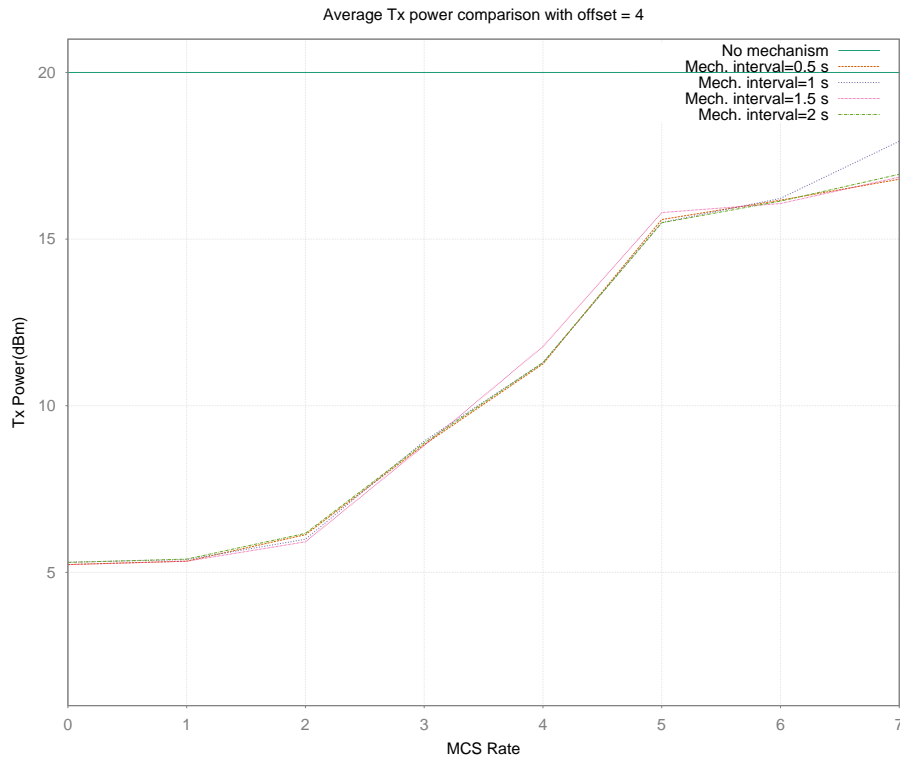
Συγκρίνοντας τις καμπύλες κάθε ομάδας μεταξύ τους, παρατηρούμε ότι όταν τις μικρότερες τιμές μέσης ισχύος παράγει ο μηχανισμός για περίοδο ενημέρωσης $0.5s$. Καθώς η ενημέρωση γίνεται με μεγάλη συχνότητα, η ακρίβεια της απαιτούμενης ισχύος για την εκπομπή των πακέτων είναι μεγαλύτερη. Επιπλέον, όταν η περίοδος ενημέρωσης είναι μεγάλη, αν ο κόμβος προσαρμόσει την ισχύ μετάδοσης σε ένα μεγάλο επίπεδο, θα συνεχίσει να εκπέμπει σε αυτό το επίπεδο για μεγαλύτερο χρονικό διάστημα, αφού ο χρόνος που θα λάβει το νέο μήνυμα για



Σχήμα 7.7: Μέση ισχύς μετάδοσης για $offset = 3$

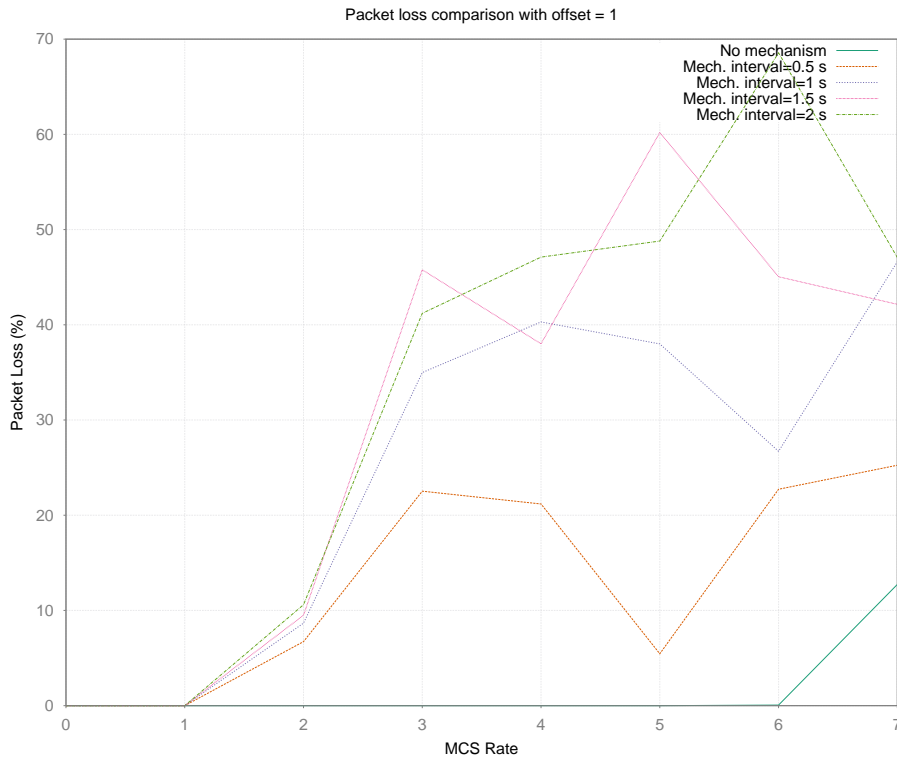
αναπροσαρμογή της ισχύος θα είναι μεγαλύτερος. Αυτό σημαίνει ότι περισσότερα πακέτα θα σταλούν χρησιμοποιώντας μεγαλύτερη ισχύ από την απαιτούμενη, άρα θα έχουν μεγαλύτερη κατανάλωση ενέργειας. Συμπεραίνεται λοιπόν ότι ο μηχανισμός καταφέρνει να εξοικονομήσει περισσότερη ενέργεια ανάλογο με την περίοδο ενημέρωσης των κόμβων.

Στο Σχήμα 7.5 παρατηρείται ότι τα αποτελέσματα είναι ασταθή και δεν συμβαδίζουν με τα συμπεράσματα που έχουν εξαχθεί. Για παράδειγμα, ενώ το αναμενόμενο αποτέλεσμα θα ήταν να εμφανίζεται η μεγαλύτερη μέση ισχύς στα πειράματα με ρυθμό μετάδοσης $65Mbps$, στα $58.5Mbps$ υπάρχει μία απότομη αύξηση της ισχύος. Αυτό είναι απόρροια της τεχνικής διόρθωσης που παρέχεται από τον μηχανισμό. Προφανώς, η προσαύξηση της ισχύος κατά 1 μονάδα δεν είναι αρκετή για την επιτυχή λήψη του πακέτου από το access point. Αυτό σημαίνει ότι το access point θα στείλει την ίδια εγγραφή στο μήνυμα ενημέρωσης περισσότερες από μία φορές, και ο κόμβος θα καταλάβει ότι δεν έστειλε με επαρκή ισχύ το πακέτο για να ληφθεί από το access point. Έτσι, ο κόμβος επαναφέρει στην μέγιστη δυνατή την ισχύ μετάδοσης, για να διασφαλίσει την αποστολή του πρώτου πακέτου μετά την αποτυχία, και να επανεκκινήσει την διαδικασία της προσαρμογής. Η παραπάνω εξήγηση υποδηλώνει ότι για $offset = 1$ η μετάδοση δεν είναι ασφαλής καθώς η υπολογισμένες τιμές βρίσκονται πολύ κοντά στα όρια του αποδεκτού SNR, οπότε υπάρχουν πάρα πολλές αποτυχίες λήψης. Συνεπώς, μεγάλο μέρος των πακέτων αποστέλλονται με την μέγιστη δυνατή ισχύ, αυξάνοντας την συνολική μέση τιμή.

Σχήμα 7.8: Μέση ισχύς μετάδοσης για $offset = 4$

7.5.2 Αξιολόγηση ως προς το packet loss

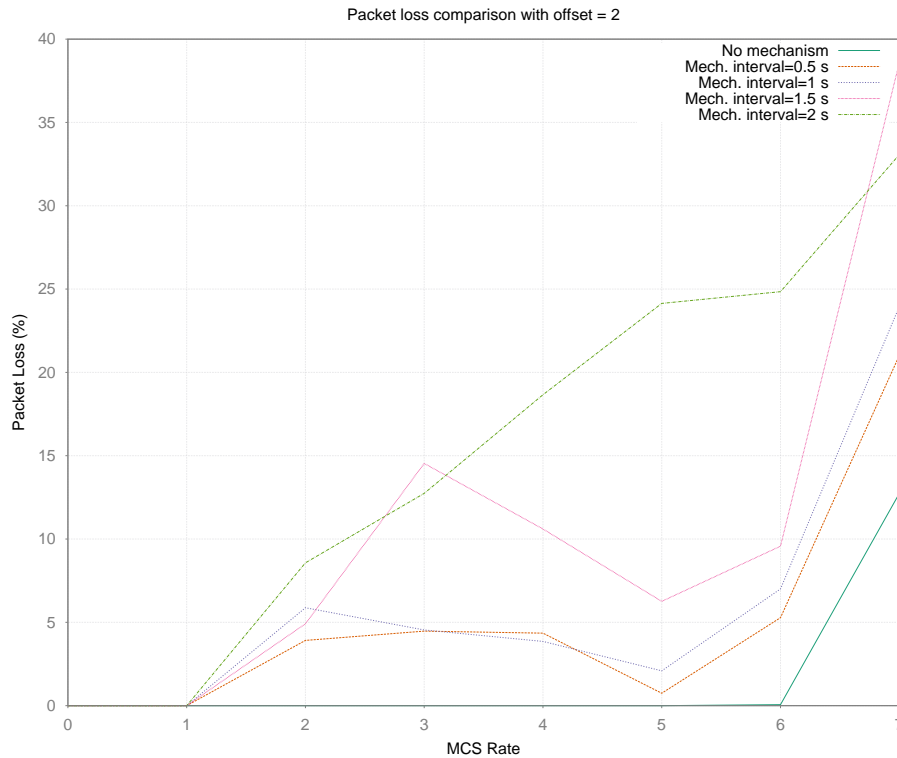
Σε αυτή την υποενότητα παρουσιάζονται τα αποτελέσματα της πειραματικής αξιολόγησης του μηχανισμού ως προς το packet loss. Τα πειράματα είναι ομαδοποιημένα όπως στην υποενότητα 7.5.1. Κατά την διάρκεια κάθε πειράματος, στέλνονται συνολικά 9900 πακέτα, με ρυθμό αποστολής 100 πακέτα ανά δευτερόλεπτο.



Σχήμα 7.9: Packet loss του πρώτου μηχανισμού για $offset = 1$

Στο Σχήμα 7.9 προβάλλονται τα αποτελέσματα για $offset = 1$. Άμεση παρατήρηση είναι ότι τα αποτελέσματα είναι ασταθή, καθώς δεν υπάρχει σταδιακή αύξηση του packet loss, όπου αυτό υπάρχει. Επιπλέον, τα πειράματα όπου η περίοδος ενημέρωσης είναι μεγαλύτερη από 1s, και μετά από το τρίτο rate παρουσιάζουν μεγάλο packet loss, στα πλαίσια το μη αποδεκτού, καθώς ξεπερνάει το 30%. Όταν δεν γίνεται χρήση του μηχανισμού, το packet loss είναι μηδενικό σε όλα τα rates, εκτός από το τελευταίο, όπου λόγω του μεγάλου απαιτούμενου SNR, όταν ο κόμβος βρίσκεται σε μεγάλη απόσταση από το access point, ακόμη και η μέγιστη ισχύς δεν είναι αρκετή. Συγκρίνοντας αυτά τα πειράματα, με αυτά για περίοδο ενημέρωσης 0.5s, τα οποία έχουν την καλύτερη απόδοση, υπάρχει μία διαφορά στο packet loss έως και 23% στην χειρότερη περίπτωση.

Για $offset = 1$, ο μηχανισμός δεν καταφέρνει να επιτύχει καλή απόδοση, καθώς η επαύξηση που γίνεται στην υπολογισμένη ισχύ αποδεικνύεται ότι δεν είναι επαρκής για να καλύψει τις απώλειες διάδοσης που προκαλούνται από την συνεχή κίνηση του κόμβου. Παρόλα αυτά, γίνεται πιο εμφανές σε σχέση με τα πειράματα της υποενότητας 7.5.1 ότι όταν μεγαλώνει η περίοδος ενημέρωσης, η διαδικασία προσαρμογής δεν επιφέρει τα επιθυμητά αποτελέσματα, λόγω της αραιής ενημέρωσης των κόμβων σχετικά με την ποιότητα της μετάδοσης.

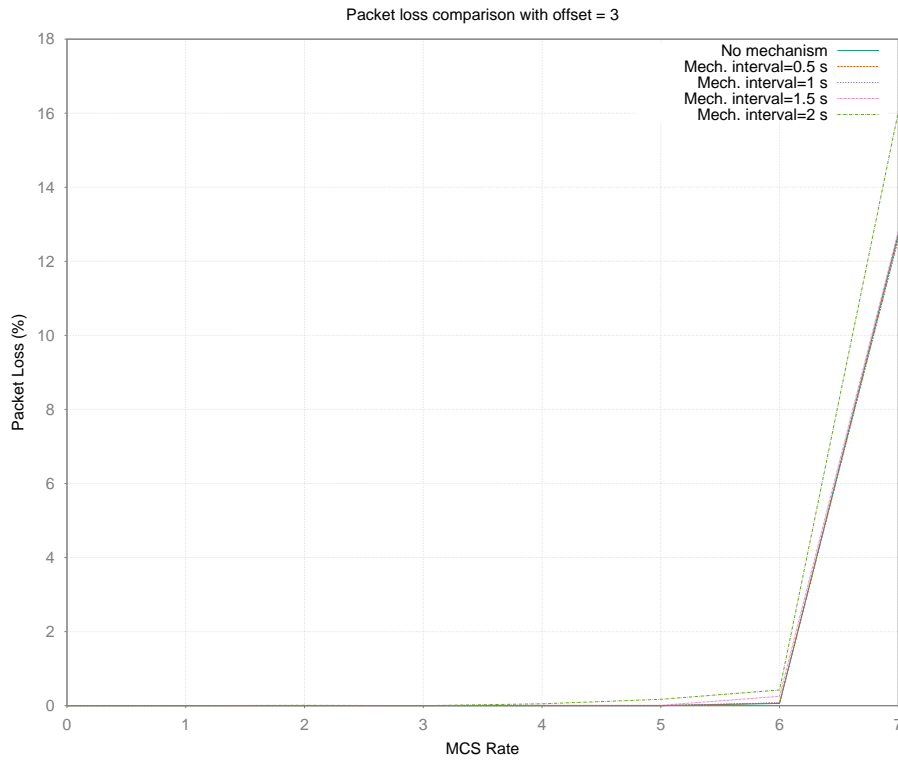


Σχήμα 7.10: Packet loss του πρώτου μηχανισμού για $offset = 2$

Στο Σχήμα 7.10, προβάλλονται τα αποτελέσματα για $offset = 2$. Ο μηχανισμός φαίνεται πως παρουσιάζει καλύτερη συμπεριφορά ανά περίοδο ενημέρωσης, με μόνη εξαίρεση όταν οι κόμβοι ενημερώνονται κάθε 2s. Σε αυτή την περίπτωση, υπάρχει μία σταδιακή αύξηση του packet loss, αυτή όμως η αύξηση φτάνει σε μη αποδεκτά επίπεδα ξεκινώντας από πολύ μικρό rate. Όσον αφορά τις υπόλοιπες περιόδους ενημέρωσης, η απόδοση τους ως προς το packet loss είναι ανεκτή, αφού κυμαίνεται σε τιμές κάτω από το 10%. Μοναδική εξαίρεση είναι στα 65 Mbps, όπου υπάρχουν απώλειες έως και 40% περίπου στην χειρότερη περίπτωση. Ο μηχανισμός δεν μπορεί να ικανοποιήσει το κριτήριο για επιτυχή αποστολή σε μεγάλες αποστάσεις, αφού το offset δεν είναι αρκετό για να καλύψει τα κενά που δημιουργούνται από την κίνηση του κόμβου και τον μεγάλο ρυθμό εξασθένησης του σήματος στο υψηλό αυτό rate.

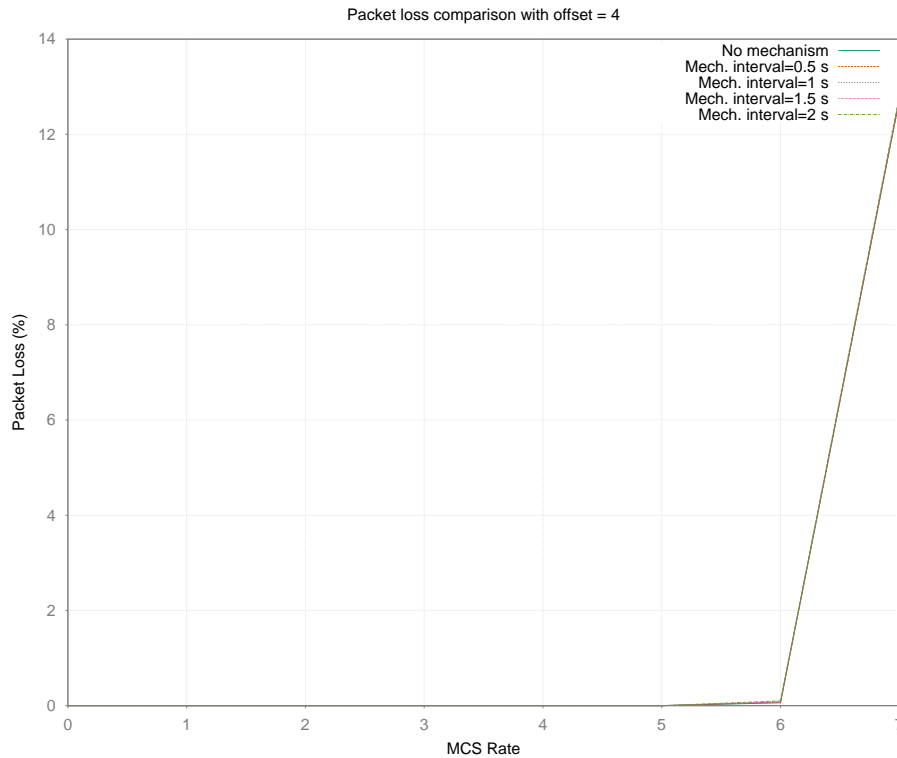
Στο Σχήμα 7.11 βρίσκονται τα αποτελέσματα της ομάδας για $offset = 3$. Σε αυτή την περίπτωση, όλες οι περιπτώσεις πειραμάτων, ανεξάρτητα της περιόδου ενημέρωσης, εμφανίζουν απόδοση όμοια αυτής χωρίς την χρήση του μηχανισμού η οποία δεν ξεπερνά το 1%. Και πάλι, μοναδική εξαίρεση είναι τα 65 Mbps όπου η αύξηση στο packet loss είναι πολύ μεγάλη σε σύγκριση με τα υπόλοιπα rates, αγγίζοντας το 16%. Επιπλέον, παρά την καλή επίδοση του μηχανισμού για όλα τα rates, φαίνεται ότι η αύξηση του χρόνου ενημέρωσης δεν ευνοεί το μηχανισμό. Η χρήση του μεγαλύτερου offset είναι η κρίσιμη παράμετρος που υπερκαλύπτει τις απώλειες της κίνησης του, αποφεύγοντας έτσι οποιοδήποτε πρόβλημα με την μετάδοση.

Στην τελευταία κατηγορία πειραμάτων το offset επιλέγεται να είναι ίσο με 4. παρατηρώντας το Σχήμα 7.12, αποδεικνύεται ότι ο μηχανισμός έχει ακριβώς την ίδια συμπεριφορά με την αρχική περίπτωση. Η αύξηση του packet loss στο τελευταίο rate οφείλεται στην απομάκρυνση του



Σχήμα 7.11: Packet loss του πρώτου μηχανισμού για $offset = 3$

κόμβου από το access point σε συνδυασμό με το μεγάλο SNR, καθιστώντας αναπόφευκτη την απώλεια πακέτων. Με άλλα λόγια, σε αυτή την ομάδα πειραμάτων ο μηχανισμός επιτυγχάνει να κάνει εξοικονόμηση ενέργειας, χωρίς να επηρεάσει καθόλου την μετάδοση στο δίκτυο.



Σχήμα 7.12: Packet loss του πρώτου μηχανισμού $offset = 4$

7.6 Σύνοψη και Συμπεράσματα

Στο παρόν κεφάλαιο παρουσιάστηκε ένας μηχανισμός για εξοικονόμηση ενέργειας σε δίκτυα υποδομής, ο οποίος προσαρμόζει την ισχύ μετάδοσης των πακέτων που εξέρχονται από τους κινητούς κόμβους του δικτύου. Ο μηχανισμός στηρίζεται στην αξιοποίηση μετρικών που περιγράφουν την ποιότητα λήψης των πακέτων και μετρικές που περιγράφουν το κανάλι μετάδοσης, οι οποίες χρησιμοποιούνται από τους κόμβους μέσω ανάδρασης. Στην συνέχεια αυτές οι πληροφορίες χρησιμοποιούνται στην λήψη αποφάσεων για την προσαρμογή της ισχύος μετάδοσης των μελλοντικών πακέτων, με σκοπό να εξοικονομηθεί όσο το δυνατόν περισσότερη ενέργεια από τις ασύρματες μεταδόσεις.

Η πειραματική αξιολόγηση έδειξε πως ο μηχανισμός καταφέρνει να μειώσει την εξοικονόμηση ενέργειας κατά τουλάχιστον 55% στο μεγαλύτερο διαθέσιμο rate του 802.11n ενώ μπορεί να φτάσει μέχρι και 95% ενώ μειώνεται το rate, χωρίς να υπάρχει αρνητική επίδραση στην επίδοση του δικτύου. Χωρίς να ληφθεί υπ' όψη η επίδοση του δικτύου, το κάτω όριο της εξοικονόμησης ενέργειας μπορεί να φτάσει έως και 63%, μειώνοντας όμως το ποσοστό επιτυχών αποστολών κατά 25%.

Από την μελέτη των πειραμάτων αποδείχθηκε ότι σημαντικοί παράγοντες στην ορθή λειτουργία του μηχανισμού ήταν η περίοδος ενημέρωσης των κόμβων για την εφαρμογή του μηχανισμού, καθώς επίσης και η παράμετρος επαύξησης της υπολογισμένης ισχύος μετάδοσης. Αυξάνοντας την παράμετρο επαύξησης, εισάγεται ένας πλεόνασμος στην ενέργεια που

καταναλώνεται, επιτυγχάνεται όμως η εξάλειψη ανεπιθύμητων φαινομένων, όπως αυτό της αποτυχίας αποστολής λόγω αύξησης της απώλειας μετάδοσης. Από την άλλη μεριά, η περίοδος επηρεάζει το τελικό αποτέλεσμα μηχανισμού, καθώς όσο πιο συχνή είναι η ενημέρωση των κόμβων, τόσο πιο επικαιροποιημένη είναι η ισχύς με την οποία στέλνονται τα πακέτα. Η παράμετρος αυτή παίζει σημαντικό ρόλο όταν η παράμετρος επαύξησης είναι μικρή, ενώ όσο αυξάνεται η δεύτερη, η σημαντικότητα της τείνει να εξαλείφεται, μέχρι να έχει σχεδόν μηδενική επίδραση στην επίδοση του μηχανισμού. Ανάλογα με τις ανάγκες και απαιτήσεις του δικτύου, μπορούν να επιλεγθούν κατάλληλες τέτοιες παράμετροι, έτσι ώστε να μεγιστοποιηθεί το ποσοστό εξοικονομούμενης ενέργειας από τους κόμβους.

Μοναδικό μεινέκτημα του μηχανισμού που παρουσιάστηκε είναι ότι προστίθεται overhead στο δίκτυο, καθώς τα πακέτα ενημέρωσης στέλνονται περιοδικά σε όλο το δίκτυο. Αυτό σημαίνει ότι αυξάνοντας την συχνότητα ενημέρωσης, το overhead μπορεί να αυξηθεί σε μεγάλο βαθμό.

Κεφάλαιο 8

Υλοποίηση μηχανισμού εξοικονόμησης ενέργειας σε ad-hoc δίκτυα

8.1 Εισαγωγή

Στο προηγούμενο κεφάλαιο παρουσιάστηκε η υλοποίηση ενός μηχανισμού μετάδοσης ενέργειας για ασύρματα δίκτυα υποδομής. Σε αυτά τα δίκτυα η επικοινωνία είναι κεντροποιημένη, αφού οι κόμβοι του δικτύου επικοινωνούν μόνο με το access point για να στείλουν πακέτα στο επιθυμητό προορισμό ακόμη και αν αυτός είναι στο ίδιο δίκτυο. Ο τρόπος λειτουργίας του δικτύου, όπου οι κόμβοι επικοινωνούν μόνο με ένα στοιχείο στο δίκτυο για τη μετάδοση πληροφορίας, κάνει εύκολη την υλοποίηση του μηχανισμού με την προσθήκη ενός μικρο overhead στο δίκτυο.

Εκτός από τα κλάσσικα δίκτυα όμως, υπάρχουν και τα ad-ηος δίκτυα, όπου οι κόμβοι είναι ισότιμοι και η επικοινωνία μεταξύ των κόμβων γίνεται χωρίς την παρουσία access point. Κάθε κόμβος επικοινωνεί με οποιοδήποτε άλλο στο δίκτυο, αρκεί να βρίσκεται εντός της εμβέλειάς του. Ένα από τα βασικά χαρακτηριστικά των ad-ηος δικτύων είναι η έκτασή τους δεν εξαρτάται από ένα συγκεκριμένο στοιχείο του δικτύου (π.χ. access point), αλλά μπορεί να μεγαλώσει ανάλογα με την τοπολογία του. Αυτό προφανώς σημαίνει ότι κάθε κόμβος δεν μπορεί να επικοινωνήσει με οποιοδήποτε άλλο άμεσα. Για αυτό, υπάρχουν διάφορα πρωτόκολλα δρομολόγησης, με τα οποία τα πακέτα διαβιβάζονται μεταξύ των κόμβων για να φτάσουν τον τελικό προορισμό. Η ευελιξία που προσφέρουν τα ad-hoc δίκτυα σε συνδυασμό με τα πρωτόκολλα δρομολόγησης, έχουν συμβάλει στην εξάπλωσή τους. Ενώ παλιότερα είχαν μόνο στρατιωτικές εφαρμογές, τώρα αναπτύσσονται ad-hoc δίκτυα στις εμπορικές και καθημερινές ασύρματες επικοινωνίες.

Για την ορθή λειτουργία των πρωτοκόλλων δρομολόγησης, είναι απαραίτητη η ανταλλαγή μηνυμάτων ελέγχου μεταξύ των κόμβων για την ανακάλυψη μονοπατιών προς απομακρυσμένους προορισμούς. Συνεπώς, η κατανάλωση ενέργειας των συσκευών αυξάνεται περισσότερο σε σχέση με τα infrastructure δίκτυα και η εφαρμογή τεχνικών για εξοικονόμηση ενέργειας

κρίνεται απαραίτητη.

Ο μηχανισμός του Κεφαλαίου 7 δεν μπορεί να εφαρμοστεί σε αυτούς τους τύπους δικτύων για να εξοικονομηθεί ενέργεια. Τα μηνύματα ενημέρωσης δεν στέλνονται από ένα μόνο κόμβο, αλλά κάθε κόμβος χωρίστα πρέπει να παίζει και τον ρόλο του access point για να λειτουργήσει σωστά ο μηχανισμός. Αυτό σημαίνει ότι ενώ αυξάνονται οι κόμβοι του δικτύου, ανάλογα αυξάνεται και το overhead στο δίκτυο. Βασική επίπτωση στην επίδοση του δικτύου είναι η μείωση του goodput, αφού αυξάνονται τα μηνύματα ελέγχου.

Σε αυτό το Κεφάλαιο, προτείνεται ένας μηχανισμός που βασίζεται στον αλγόριθμο του Κεφαλαίου 6 και αξιοποιεί το πρωτόκολλο δρομολόγησης OLSR. Συγκεκριμένα, τα HELLO πακέτα που αποστέλλονται από το πρωτόκολλο, έχουν διαμορφωθεί κατάλληλα έτσι ώστε να παρέχουν πληροφορίες στους κόμβους σχετικά με τους γείτονες τους, με στόχο να γίνεται η κατάλληλη προσαρμογή ισχύος κατά την αποστολή πακέτων προς αυτούς. Παράλληλα, τίθενται κανόνες και περιορισμοί στον τρόπο λειτουργίας του μηχανισμού, έτσι ώστε να μην επηρεάζεται η αρχική λειτουργία του πρωτοκόλλου δρομολόγησης.

Για την αξιολόγηση του αλγορίθμου έγιναν προσωμιώσεις στον NS-3, όπου μετρήθηκε η μέση ισχύ μετάδοσης, το packet loss και το throughput του δικτύου για διαφορετικά πλήθη κόμβων. Τα πειράματα αποδεικνύουν ότι επιτυγχάνεται μείωση της ισχύος μεταδοσης των κόμβων χωρίς να υπάρχουν απώλειες στην επίδοση του δικτύου, ενώ η λειτουργία του πρωτοκόλλου δεν παρεμποδίζεται από την εφαρμογή του μηχανισμού.

8.2 Αρχιτεκτονική του μηχανισμού

Στόχος του μηχανισμού που παρουσιάζεται είναι η ενσωμάτωση της τεχνικής εξοικονόμησης ενέργειας που αναλύεται στο Κεφάλαιο 6 στο πρωτόκολλο OLSR, με απώτερο σκοπό την αποδοτική χρήση της τεχνικής και την αξιοποίηση των υπάρχοντων χαρακτηριστικών του δικτύου. Βασική ιδέα πίσω από τον μηχανισμό είναι ότι τα περισσότερα πρωτόκολλα δρομολόγησης χρησιμοποιούν μηνύματα ελέγχου για την εύρεση μονοπατιών προς απομακρυσμένους κόμβους. Μέσω των μηνυμάτων ανακαλύπτουν τους γειτονικούς κόμβους στους οποίους πρέπει να στείλουν κάποιο πακέτο για να μεταβιβαστεί σε κάποιον άλλον κόμβο. Άρα, για την μετάδοση των πακέτων δεν μας ενδιαφέρει ο τελικός προορισμός, αλλά ο γειτονικός κόμβος στον οποίο πρέπει να σταλεί το πακέτο.

Το OLSR χρησιμοποιεί HELLO μηνύματα για την ανακάλυψη γειτόνων και μονοπατιών προς άλλους κόμβους. Τα μηνύματα γίνονται broadcast στο δίκτυο και ακούγονται από τους γειτονικούς κόμβους, οι οποίοι βρίσκονται εντός εμβέλειας του αποστολέα. Τα περιεχόμενα του μηνύματος είναι η λίστα με τους κόμβους από τους οποίους έχει λάβει κάποιο HELLO μήνυμα. Για την ενσωμάτωση του μηχανισμού, γίνεται η προσθήκη επιπρόσθετων πεδίων σε κάθε εγγραφή της λίστας, που περιέχουν την ισχύ λήψης από τον κόμβο της εγγραφής καθώς επίσης και τον θόρυβο του καναλιού, όπως στο Σχήμα 8.1. Έτσι, όταν οι γειτονικοί κόμβοι λάβουν το HELLO πακέτο, ανατρέχοντας στην εγγραφή τους μπορούν να ανακτήσουν τα δεδομένα που είναι απαραίτητα για την προσαρμογή της ισχύος όταν χρειαστεί να στείλουν ένα πακέτο προς αυτό τον κόμβο.

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved										Htime										Willingness											
Link Code					Reserved					Link Message Size																					
TxPower										Noise Level																					
Neighbor Interface Address																															
Neighbor Interface Address																															
...																															
Link Code					Reserved					Link Message Size																					
TxPower										Noise Level																					
Neighbor Interface Address																															
Neighbor Interface Address																															

Σχήμα 8.1: Δομή του HELLO πακέτου με την τροποποίηση του μηχανισμού.

Για παράδειγμα, έστω ότι υπάρχουν 2 γειτονικοί κόμβοι A και B που βρίσκονται στην αρχική κατάσταση χωρίς καμία εγγραφή στον πίνακα δρομολόγησης. Αρχικά ο A θα εκμπέμψει ένα πακέτο HELLO, το οποίο θα λάβει ο B. Ο B, όταν λάβει το πακέτο, θα διαβάσει την ισχύ λήψης και τον θόρυβο, και θα τα καταχωρήσει στην εγγραφή που θα δημιουργήσει στο link set για τον A. Όταν έρθει η σειρά του B να στείλει το δικό του HELLO πακέτο, στα περιεχόμενα θα συμπεριλάβει την εγγραφή που έχει αποθηκεύσει για τον A, προσθέτοντας τις πληροφορίες σχετικά με την ισχύ λήψης και τον θόρυβο που αποθήκευσε νωρίτερα. Τέλος, όταν ο A λάβει το πακέτο, θα αναζητήσει την εγγραφή που του αντιστοιχεί και θα αποθηκεύσει τις τιμές που περιέχονται στον δικό του πίνακα δρομολόγησης. Έτσι, την επόμενη φορά που ο A θα στείλει κάποιο πακέτο δεδομένων προς τον B, θα χρησιμοποιήσει τις πληροφορίες αυτές για να προσαρμόσει κατάλληλα την ισχύ μετάδοσης. Όλες οι διαδικασίες που εκτελεί ο μηχανισμός περιγράφονται στον Αλγόριθμο

Από την παραπάνω περιγραφή, φαίνεται ότι εκτός από την τροποποίηση των HELLO πακέτων, απαιτούνται τροποποιήσεις στον πίνακα δρομολόγησης των που δημιουργείται από το πρωτόκολλο. Ένας κόμβος πρέπει να διατηρεί για κάθε εγγραφή/γείτονα τις παρακάτω πληροφορίες σε 4 πεδία:

1. Την ισχύ λήψης του HELLO πακέτου που έστειλε ο γειτονικός κόμβος.
2. Τον αντίστοιχο θόρυβο κατά την λήψη του πακέτου.
3. Την ισχύ λήψης με την οποία έλαβε ο γειτονικός κόμβος το HELLO πακέτο.
4. Τον αντίστοιχο θόρυβο.

Από τα πεδία αυτά, τα 1 και 2 χρησιμοποιούνται κατά την κατασκευή του HELLO πακέτου, ενώ τα 3 και 4 κατά την προσαρμογή της ισχύος μετάδοσης.

Πλεονέκτημα της αρχιτεκτονικής που προτείνεται είναι προφανώς η απουσία επιρόσθετων πακέτων ελέγχου για την εξοικονόμηση ενέργειας. Οι απαραίτητες πληροφορίες γίνονται piggyback στα πακέτου ελέγχου του OLSR. Έτσι αποφεύγεται επιβάρυνση του δικτύου με επιρόσθετη κίνηση και σπατάλη ενέργειας για την μετάδοση της κίνησης αυτής. Σημαντικό

ρόλο όμως παίζει η συχνότητα με την οποία γίνεται η ενημέρωση των κόμβων. Όσο μικρότερη είναι η συχνότητα εκπομπής HELLO πακέτων, οι κόμβοι διατηρούν παρωχημένες τιμές στο στον πίνακα δρομολόγησης, με αποτέλεσμα η πρόβλεψη για την ισχύ μετάδοσης να μην είναι ακριβής.

Από την ανάλυση των διαδικασιών που εκτελεί ο μηχανισμός, παρατηρείται ότι υπάρχει η απουσία του path loss από τις ανταλλασσόμενες πληροφορίες. Αυτό συμβαίνει διότι ο μηχανισμός είναι σχεδιασμένος έτσι ώστε να εξασφαλίζει ότι τα πακέτα HELLO, κατά την λήψη των οποίων γίνεται η μέτρηση της ισχύος, στέλνονται με την μέγιστη δυνατή ισχύ. Υποθέτουμε ότι η μέγιστη ισχύς των κόμβων είναι ίδια για όλους τους κόμβους, οπότε η πληροφορία που λείπει για τον υπολογισμό του path loss είναι ήδη γνωστή. Η αποστολή των HELLO πακέτων με την μεγαλύτερη δυνατή ισχύ είναι αναπόφευκτη, καθώς σε περίπτωση που μειωθεί, η εμβέλεια του κόμβου μικραίνει. Αυτό θα είχε ως συνέπεια την απώλεια πιθανών γειτόνων στο εγγύς περιβάλλον του κόμβου, και εν συνεχεία την μείωση των μονοπατιών προς απομακρυσμένους κόμβους. Για να μην υπάρξουν αρνητικές επιπτώσεις στην δρομολόγηση, θεωρούμε λοιπόν ότι τα HELLO μηνύματα θα εκπέμπονται με την μέγιστη δυνατή ισχύ.

8.3 Πειραματική αξιολόγηση

Για την αξιολόγηση του μηχανισμού έγιναν πειράματα στον ns-3 των οποίων οι παράμετροι φαίνονται στον Πίνακα 8.1.

Χαρακτηριστικό	Τιμή
Διαστάσεις χώρου	150 × 150m
Πλήθος κόμβων	15-50
Offset μηχανισμού	6
Ρυθμός Μετάδοσης (Data Rate)	MCS 0 – 7
Ρυθμός αποστολής πακέτων	500Kbps
Διάρκεια πειράματος	100s

Πίνακας 8.1: Χαρακτηριστικά πειραμάτων για την αξιολόγηση του μηχανισμού σε ad hoc δίκτυο.

Έγιναν πειράματα όχι μόνο για τη περίπτωση που χρησιμοποιείται ο μηχανισμός εξοικονόμησης, αλλά και πειράματα χωρίς την χρήση του μηχανισμού, έτσι ώστε να υπάρχει μία βάση αναφοράς σχετικά με την επίδοση του μηχανισμού. Οι κόμβοι του δικτύου κατανομούνται ομοιόμορφα στο δίκτυο και ινούνται τυχαία μέσα στον καθορισμένο χώρο. Για αυτό τον λόγο, είναι πολύ πιθανό να εμφανίζεται packet loss επειδή κάποιος κόμβος απομονώνεται ορισμένες φορές κατά την διάρκεια του πειράματος. Οι τοπολογίες σε κάθε ζεύγος πειραμάτων (χωρίς ή με τον μηχανισμό) είναι ίδιες, όπως επίσης και η κίνηση των κόμβων.

Κάθε πείραμα γίνεται με διαφορετικό πλήθος κόμβων, το οποίο σημαίνει ότι η πυκνότητα του δικτύου διαφέρει. Επιπλέον, όπως και στο Κεφάλαιο 7, τα πειράματα επαναλήφθηκαν για κάθε data rate που χρησιμοποιεί το 802.11n για να παρατηρηθεί το ποσοστό εξοικονόμησης

Algorithm 4 Power management mechanism for OLSR protocol

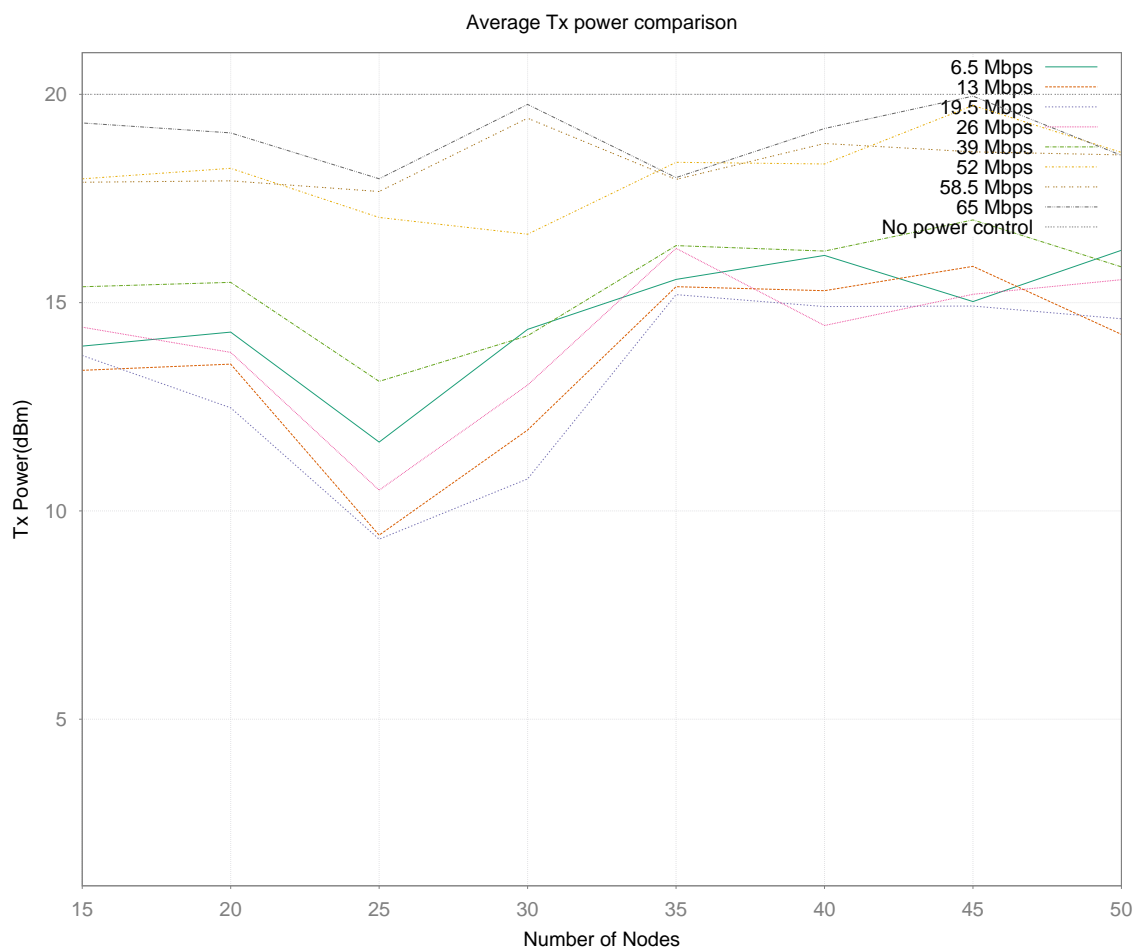
```

1: function PACKETRECEIVED(packet)
2:   source_ip = GetSourceIp(packet);
3:   rx_power = MeasureRxPower(packet);
4:   noise_level = MeasureNoise(packet);
5:   src_node_rx_power = ExtractRxFieldFromOLSRHeader(packet);
6:   src_node_noise_level = ExtractNoiseFieldFromOLSRHeader(packet);
7:   UpdateRoutingTableEntry(source_ip, rx_power, noise_level, src_node_rx_power, src_node_noise_level);
8: end function
9:
10: function SEND_HELLO_MESSAGE(hello_message, routing_table)
11:   for all entries in hello_message do
12:     entry_rx_power = GetRxPowerFromNode(entry_ip, routing_table);
13:     AddRxPowerToEntry(entry, entry_rx_power);
14:   end for
15:   Send_Message();
16: end function
17:
18: function SEND_PACKET(packet, data_rate)
19:   destination_ip = GetDestinationIp(packet);
20:   rx_power = GetRxPowerOfNode(destination_ip);
21:   noise = GetNoiseOfNode(destination_ip);
22:   snr_threshold = GetSnrThresholdForRate(data_rate);
23:
24:   ▷ Calculate Path Loss
25:   ▷  $P_{Tx}$  for HELLO is always 20
26:   path_loss = hello_tx_power - rx_power;
27:   noise = GetNoise();
28:   tx_power = snr_threshold + noise - path_loss;
29:
30:   ▷ Set limitations to Tx Power
31:   if tx_power < 5 then
32:     tx_power = 5;
33:   else if tx_power > 20 then
34:     tx_power = 20;
35:   end if
36:   SetTxPower(tx_power);
37: end function

```

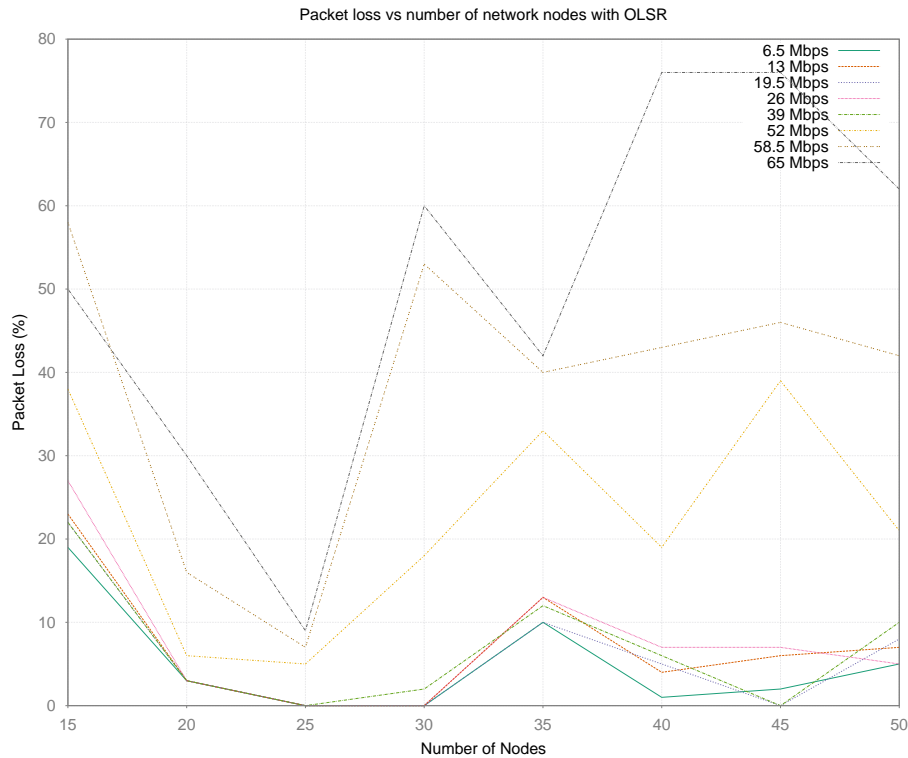
ενέργειας σε σχέση με το rate. Τα αποτελέσματα αναμένονται να είναι παρόμοια με αυτά του προηγούμενου κεφαλαίου, αλλά με μερικές διαφοροποιήσεις, καθώς το πρωτόκολλο δρομολόγησης έχει την δική του ανεξάρτητη συμπεριφορά όσον αφορά την επιλογή των διαδρομών, όπως περιγράφεται στο Κεφάλαιο 3.

Στο Σχήμα 8.2 φαίνεται η μέση ισχύς μετάδοσης των κόμβων ανά data rate και πλήθος κόμβων μέσα στο δίκτυο.



Σχήμα 8.2: Σύγκριση μέσης ισχύος μετάδοσης με και χωρίς την χρήση του μηχανισμού.

Από το σχήμα, φαίνεται ότι η ελάχιστη τιμή που πετυχαίνει ο μηχανισμός είναι στα 9 περίπου dBm, όταν το πλήθος των κόμβων είναι 25. Προφανώς η τοπολογία και η κίνηση των κόμβων ήταν τέτοια που η απόσταση ήταν μικρή και η επικοινωνία άμεση. Σε όλες τις υπόλοιπες περιπτώσεις, η μέση ισχύς φαίνεται να είναι σταθερή, παρουσιάζοντας προσauξήσεις 1-3 dBm. Για παράδειγμα, στην περίπτωση των 39 Mbps, βλέπουμε ότι η μέση ισχύς σε κάθε πείραμα που έγινε βρίσκεται πολύ κοντά στα 15 dBm, με μόνη εξαίρεση το σενάριο με τους 25 κόμβους. Η παρουσία των ενδιάμεσων κόμβων μειώνει την μέση ισχύ, παράλληλα όμως δεν μειώνεται υπερβολικά, καθώς το πρωτόκολλο δεν επιλέγει το επόμενο άλμα της διαδρομής με κριτήριο την εξοικονόμηση ενέργειας. Για παράδειγμα, μπορεί να υπάρχει μόνοπάτι προς τον κόμβο προορισμού μέσω ενός κόμβου που εξασφαλίζει μικρότερη κατανάλωση ισχύος,



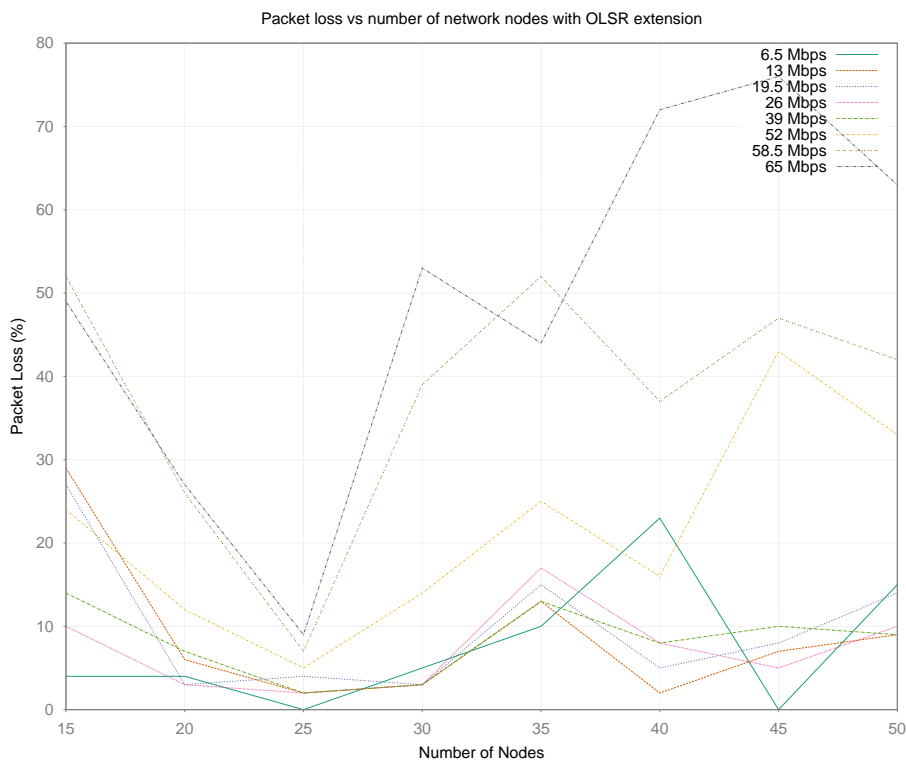
Σχήμα 8.3: Packet loss χωρίς την χρήση του μηχανισμού.

όμως ο κόμβος που επιλέγεται είναι κάποιος άλλος, που έχει επιλεγεί ως MPR σύμφωνα με το OLSR. Από την άλλη μεριά, η επιλογή του πιο κοντινού κόμβου, είναι πολύ πιθανό να επιβαρύνει τον ενδιαμέσο κόμβο, αφού θα πρέπει να μεταδώσει το πακέτο με μεγαλύτερη ισχύ για να φτάσει σε κάποιον απομακρυσμένο ενδιαμέσο κόμβο, ή τον τελικό προορισμό του.

Σε σύγκριση με την ισχύ μετάδοσης όταν δεν χρησιμοποιείται ο μηχανισμός, τα πακέτα μεταδίδονται με έως και 90% λιγότερη ισχύ, όπως στην περίπτωση με τους 25 κόμβους. Στην χειρότερη περίπτωση, ο μηχανισμός μεταδίδει τα πακέτα με 20% λιγότερη ισχύ. Όπως και στα πειράματα του προηγούμενου κεφαλαίου όσο αυξάνεται το data rate, η ενέργεια που μπορεί να σωθεί είναι λιγότερη, λόγω της ανάγκης για την επίτευξη μεγαλύτερου SNR.

Στα Σχήματα 8.3 και 8.4 φαίνεται το packet loss χωρίς και με τον μηχανισμό αντίστοιχα. Όπως προαναφέρθηκε, η τοπολογία και η κίνηση των κόμβων στα πειράματα είναι τυχαία, συνεπώς το packet loss είναι αναμενόμενο. Στην αρχική περίπτωση, όπου δεν χρησιμοποιείται ο μηχανισμός, παρατηρείται ότι για τα μικρά rates το packet loss είναι ικανοποιητικά μικρό, ενώ όσο αυξάνεται το rate η επίδοση του δικτύου επιδεινώνεται. Συμπεραίνεται ότι συνδέσεις που ικανοποιούν οριακά τις προϋποθέσεις για ορθή μετάδοση δεν είναι αρκετές για την σωστή μετάδοση των πακέτων. Όταν το δίκτυο αποτελείται από μικρό πλήθος κόμβους, το packet loss είναι μεγάλο εξαιτίας της μικρής πυκνότητας του δικτύου.

Με την χρήση του μηχανισμού, οι απώλειες που υπάρχουν στο δίκτυο κυμαίνονται στα ίδια επίπεδα με την κανονική περίπτωση. Υπάρχει μία αύξηση 4-5 % στα χαμένα πακέτα. Προφανής λόγος για αυτές τις απώλειες είναι ότι με την χρήση του συγκριμένου μηχανισμού,



Σχήμα 8.4: Packet loss με την χρήση του μηχανισμού.

η ισχύς που υπολογίζεται για την μετάδοση τείνει να είναι πολύ κοντά στο όριο για την επιτυχή αποστολή πακέτων. Σε συνδυασμό με την αυξομείωση της απόστασης μεταξύ των κόμβων που επικοινωνούν, αυξάνεται η πιθανότητα να χαθεί ένα πακέτο κατά την αποστολή του. Παρόλα αυτά, οι απώλειες που παρουσιάζονται βρίσκονται σε ανεχτά επίπεδα, καθιστώντας τον μηχανισμό εφαρμόσιμο.

8.4 Σύνοψη και Συμπεράσματα

Σε αυτό το κεφάλαιο παρουσιάστηκε ένας μηχανισμός για την εξοικονόμηση ενέργειας για ad-hoc ασύρματα δίκτυα τα οποία χρησιμοποιούν το OLSR για πρωτόκολλο δρομολόγησης. Όπως και στο Κεφάλαιο 7, ο μηχανισμός βασίζεται στην ανταλλαγή πληροφοριών μεταξύ των κόμβων για την απόκτηση πληροφοριών σχετικά με την ισχύ λήψης των σημάτων. Για την υλοποίηση αυτού του μηχανισμού ανάδρασης, αξιοποιήθηκε η αρχιτεκτονική του πρωτοκόλλου δρομολόγησης, στην οποία ανταλλάσσονται μηνύματα μεταξύ των κόμβων ανα τακτά χρονικά διαστήματα. Έτσι αποφεύχθηκε η επιβάρυνση του δικτύου με επιπρόσθετα μηνύματα τα οποία θα μείωναν την ωφέλιμη χωρητικότητα του καναλιού. Τα αποτελέσματα έδειξαν ότι με την χρήση του μηχανισμού η ισχύς μετάδοσης μειώθηκε σε 20 έως και 90%, ανάλογα με τα χαρακτηριστικά του δικτύου, όπως ο ρυθμός μετάδοσης και οι συνθήκες του καναλιού. Οι απώλειες από την μείωση της ισχύος δεν είναι μεγάλες και οι διαφορές τους από την αρχική περίπτωση είναι πολύ μικρές.

Ο μηχανισμός προσαρμόστηκε κατάλληλα για την ενσωμάτωση του μόνο στο πρωτόκολλο δρομολόγησης OLSR, η βασική ιδέα όμως που περιγράφεται από αυτόν μπορεί να ενσωματωθεί και στα υπόλοιπα proactive πρωτόκολλα, η πλειοψηφία αυτών βασίζεται στην εκ των προτέρων γνώση των μονοπατιών που θα ακολουθηθούν. Από την άλλη μεριά, η ενσωμάτωση αυτού σε reactive πρωτόκολλα, δεν αναμένεται να είναι αποδοτική, καθώς το μονοπάτι υπολογίζεται μόνο όταν ζητηθεί. Λόγω αυτού, αν το μονοπάτι είναι πολύ μεγάλο, οι πληροφορίες που απαιτούνται θα καθυστερήσουν να φτάσουν σε κάθε κόμβο κατά την επιστροφή του μονοπατιού, το οποίο αν συνδυαστεί με το γεγονός ότι οι κόμβοι μπορεί να κινούνται, θα οδηγήσει σε λανθασμένο υπολογισμό ισχύος. Για αυτούς του τύπους δικτύων ο μηχανισμός χρειάζεται να αξιοποιήσει επιπλέον μετρικές και πληροφορίες για να καταφέρει εξοικονόμηση, ακολουθώντας αυτή την κατεύθυνση.

Κεφάλαιο 9

Επίλογος

9.1 Σύνοψη

Τα ασύρματα δίκτυα αποτελούν πια τον πιο διαδεδομένο τρόπο για ασύρματη επικοινωνία μεταξύ των συσκευών και των χρηστών. Η χρήση κινητών συσκευών συνέβαλε στην ανάγκη για την ανάπτυξη νέων τεχνικών στην ασύρματη επικοινωνία και την συνεχή βελτίωση αυτών, με σκοπό την επίτευξη μεγάλων ρυθμών μετάδοσης δεδομένων και την εξασφάλιση της καλής ποιότητας αυτών. Ωστόσο, το σημαντικότερο πρόβλημα που αντιμετωπίζουν οι κινητές συσκευές είναι η περιορισμένη ενέργεια που διαθέτουν για να λειτουργήσουν. Ο άριστος ρυθμός εξέλιξης των τεχνολογιών ενέργειας και των ασύρματων επικοινωνιών έχει ως αποτέλεσμα την δραματική μείωση του χρόνου λειτουργίας των συσκευών, ειδικά όταν η συσκευή χρησιμοποιεί για μεγάλα χρονικά διαστήματα το ασύρματο δίκτυο. Έτσι, υπάρχει μεγάλη ανάγκη για την εύρεση τεχνικών οι οποίες μπορούν να μειώσουν την ενέργεια που σπαταλάται για την ασύρματη επικοινωνία, έτσι ώστε να επιμηκυνθεί ο χρόνος λειτουργίας των συσκευών.

Έχουν προταθεί πολλές μέθοδοι για την εξοικονόμηση ενέργειας στα ασύρματα δίκτυα. Κάποιες από αυτές, βασίζονται σε διάφορα στρώματα του OSI, όπως το MAC layer στο οποίο έχουν γίνει τροποποιήσεις στον χρονοπρογραμματισμό διάφορων διαδικασιών που λαμβάνουν μέρος, έτσι ώστε να σπαταληθεί λιγότερη ενέργεια για την ορθή λειτουργία της επικοινωνίας. Άλλες τεχνικές, επιζητούν την εξοικονόμηση ενέργειας μέσω της προσαρμογής της ισχύος μετάδοσης των συσκευών, σύμφωνα με διάφορα κριτήρια ή μέσω της τροποποίησης των πρωτοκόλλων δρομολόγησης που χρησιμοποιούνται στα ασύρματα δίκτυα.

Στην παρούσα διπλωματική εργασία, παρουσιάστηκε ένας μηχανισμός προσαρμογής ισχύος μετάδοσης με στόχο την εξοικονόμηση ενέργειας. Βασικό κίνητρο πίσω από αυτό τον μηχανισμό είναι το γεγονός ότι σε ένα μεγάλο μέρος των μεταδόσεων που λαμβάνουν μέρος, υπάρχει πλεόνασμα ενέργειας το οποίο σπαταλάται και δεν επιφέρει κάποια βελτίωση στην επίδοση του δικτύου. Συγκεκριμένα, μελετήθηκε η συμπεριφορά του SNR ως προς τους ρυθμούς μετάδοσης που υποστηρίζονται από τα ασύρματα δίκτυα. Ο μηχανισμός ορίζει για κάθε ρυθμό μετάδοσης ένα κάτω όριο για το SNR, το οποίο εξασφαλίζει ότι κάθε πακέτο θα φτάσει επιτυχώς στον προορισμό του. Οι κόμβοι ανταλλάσσουν πληροφορίες ανα τακτά χρονικά διαστήματα σχετικά με την ισχύ λήψης των πακέτων, έτσι ώστε ο μηχανισμός να μπορεί να κάνει τους

απαραίτητους υπολογισμούς. Ο μηχανισμός χρησιμοποιεί το υπολογισμένο path loss και το οριακό SNR για να καθορίσει την κατάλληλη ισχύ μετάδοσης. Επιπλέον, προτάθηκε και μία υλοποίηση για ad-hoc δίκτυα, όπου γίνεται τροποποίηση του πρωτοκόλλου OLSR. Στη υλοποίηση αυτή, αξιοποιήθηκε η παρουσία των μηνυμάτων HELLO που ανταλλάσσονται μεταξύ των κόμβων, για την μεταφορά των πληροφοριών που απαιτούνται από τον μηχανισμό.

Τα αποτελέσματα έδειξαν ότι με την χρήση του μηχανισμού οι κινητές συσκευές μπορούν να χρησιμοποιήσουν έως και 90% λιγότερη ενέργεια. Βασικά κριτήρια που καθορίζουν το κάτω όριο ισχύος εκπομπής είναι ο ρυθμός μετάδοσης, αφού η αύξηση αυτού απαιτεί μεγαλύτερο SNR για την ορθή επικοινωνία των κόμβων. Όσον αφορά την επίδοση του δικτύου, ο μηχανισμός παρουσιάζει παρόμοια συμπεριφορά με την περίπτωση όπου δεν χρησιμοποιείται ο μηχανισμός, κάτω από συγκεκριμένες τιμές των παραμέτρων του μηχανισμού. Σε ορισμένες περιπτώσεις όμως, ο μηχανισμός αδυνατεί να προσαρμόσει κατάλληλα την ισχύ, οδηγώντας σε απώλειες, οι οποίες δεν κάνουν τον μηχανισμό μη εφαρμόσιμο.

Βασικό μειονέκτημα του μηχανισμού είναι προφανώς η ανάγκη και πρόσφατες πληροφορίες σχετικά με την ισχύ λήψης των κόμβων. Από τα πειράματα αποδείχθηκε ότι όταν οι κόμβοι κινούνται, απαιτείται η συχνότερη αποστολή μηνυμάτων για την ενημέρωση του μηχανισμού. Συνεπώς, εμφανίζεται μεγάλο overhead το οποίο επιβαρύνει το δίκτυο ως προς το throughput αλλά και τους κόμβους ως προς την κατανάλωση ενέργειας.

9.2 Μελλοντικές επεκτάσεις

Στο μέλλον, μπορούν να γίνουν μελέτες παρόμοιων μηχανισμών για την εξοικονόμηση ενέργειας, οι οποίοι θα χρησιμοποιούν εναλλακτικές μετρικές για την αξιολόγηση της σύνδεσης. Σημαντική θα ήταν και η μελέτη της εφαρμογής του μηχανισμού σε κανάλια τα οποία έχουν διαφορετικά, πιο περίπλοκα χαρακτηριστικά, όπως fading ή interference. Σημαντική μελλοντική βελτίωση θα ήταν η επέκταση του μηχανισμού στα ad-hoc δίκτυα με τέτοιο τρόπο ώστε τα RTS/CTS μηνύματα να αποστέλλονται με την ίδια ισχύ με την οποία θα σταλεί και το πακέτο. Αυτή η τροποποίηση μπορεί να οδηγήσει σε αύξηση του throughput στα δίκτυα αυτά, καθώς οι κόμβοι που θα βρίσκονται εκτός εμβέλειας θα μπορούν να εκπέμψουν. Επιπρόσθετα, για την αποφυγή του hidden node problem, χρήσιμη θα ήταν η προσθήκη ενός μοντέλου πρόβλεψης συγκρούσεων λόγω εμβέλειας, έτσι ώστε να λειτουργήσει ορθά το δίκτυο.

Βιβλιογραφία

- [1] 802.11-2012 - IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Technical Report IEEE Std 802.11™-2012, IEEE-Inst.
- [2] Linux wireless wiki. <https://wireless.wiki.kernel.org/>. Accessed: 2016-06-06.
- [3] Network simulator 3. <https://www.nsnam.org/>. Accessed: 2016-06-06.
- [4] *ns-3 Manual*.
- [5] Optimized link state routing protocol (olsr), 2003.
- [6] M. Afanasyev, T. Chen, G. M. Voelker, and A. C. Snoeren. Usage patterns in an urban wifi network. *IEEE/ACM Transactions on Networking*, 18(5):1359–1372, 2010.
- [7] S. P. Bhatsangave and V. R. Chirchi. Article: Oaodv routing algorithm for improving energy efficiency in manet. *International Journal of Computer Applications*, 51(21):15–22, August 2012. Full text available.
- [8] A. Carroll and G. Heiser. An analysis of power consumption in a smartphone. In *USENIX annual technical conference*, volume 14. Boston, MA, 2010.
- [9] G. Ferrari, S. A. Malvassori, M. Bragalini, and O. K. Tonguz. Physical layer-constrained routing in ad-hoc wireless networks: A modified aodv protocol with power control.
- [10] M. S. Gast. *802.11 Wireless Networks: The Definitive Guide, Second Edition*. O’Reilly Media, Inc., 2005.
- [11] M. Heni and R. Bouallegue. Power control in reactive routing protocol for mobile ad hoc network. *CoRR*, abs/1205.1657, 2012.
- [12] S. Jin, J. Fu, and L. Xu. *AsiaSim 2012: Asia Simulation Conference 2012, Shanghai, China, October 27-30, 2012. Proceedings, Part II*, chapter The Transmission Power Control Method for Wireless Sensor Networks Based on LQI and RSSI, pages 37–44. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

- [13] J. Kim, S. Chang, and Y. Kwon. Odtpc: On-demand transmission power control for wireless sensor networks. In *2008 International Conference on Information Networking*, pages 1–5, Jan 2008.
- [14] A. Pal, B. Soibam, and A. Nasipuri. A distributed power control and routing scheme for rechargeable sensor networks. In *Southeastcon, 2013 Proceedings of IEEE*, pages 1–5, April 2013.
- [15] K. Pappa, A. Athanasopoulos, E. Topalis, and S. Koubias. Implementation of power aware features in aodv for ad hoc sensor networks. a simulation study. In *2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007)*, pages 1372–1375, Sept 2007.
- [16] A. T. Parameswaran, M. I. Husain, and S. Upadhyaya. Is RSSI a Reliable Parameter in Sensor Localization Algorithms - An Experimental Study. 2009.
- [17] G. Pei and T. R. Henderson. Validation of ofdm error rate model in ns-3. *Boeing Research Technology*, pages 1–15, 2010.
- [18] G. P. Perrucci, F. H. Fitzek, and J. Widmer. Survey on energy consumption entities on the smartphone platform. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pages 1–6. IEEE, 2011.
- [19] E. Rozner, V. Navda, R. Ramjee, and S. Rayanchu. Napman: Network-assisted power management for wifi devices. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, MobiSys '10*, pages 91–106, New York, NY, USA, 2010. ACM.
- [20] A. Sheth and R. Han. An implementation of transmit power control in 802.11 b wireless networks.
- [21] H. Simaremare, A. Syarif, A. Abouaissa, R. F. Sari, and P. Lorenz. Energy consumption analysis of modified aodv routing protocol under random waypoint and reference point group mobility models. In *Advanced Computer Science and Information Systems (ICACSIS), 2012 International Conference on*, pages 47–51, Dec 2012.
- [22] P. Swain, S. Chakraborty, S. Nandi, and P. Bhaduri. Performance modeling and evaluation of ieee 802.11 ibss power save mode. *Ad Hoc Netw.*, 13:336–350, Feb. 2014.
- [23] S. Tang, H. Yomo, A. Hasegawa, T. Shibata, and M. Ohashi. Joint transmit power control and rate adaptation for wireless lans. *Wireless Personal Communications*, 74(2):469–486, 2014.
- [24] A. Vlavianos, L. K. Law, I. Broustis, S. V. Krishnamurthy, and M. Faloutsos. Assessing link quality in ieee 802.11 wireless networks: which is the right metric? In *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, pages 1–6. IEEE, 2008.

-
- [25] Wikipedia. Optimized link state routing protocol — wikipedia, the free encyclopedia, 2016.
- [26] C. Yoon, D. Kim, W. Jung, C. Kang, and H. Cha. Appscope: Application energy metering framework for android smartphone using kernel activity monitoring. In *Presented as part of the 2012 USENIX Annual Technical Conference (USENIX ATC 12)*, pages 387–400, 2012.
- [27] H. Yu, L. Zhong, and A. Sabharwal. Power management of mimo network interfaces on mobile systems. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20(7):1175–1186, July 2012.
- [28] J. Zhang, K. Tan, J. Zhao, H. Wu, and Y. Zhang. A practical snr-guided rate adaptation. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE, 2008.
- [29] Y. Zhu, S. Luan, V. C. M. Leung, and K. Chi. Enhancing timer-based power management to support delay-intolerant uplink traffic in infrastructure IEEE 802.11 wlans. *IEEE Trans. Vehicular Technology*, 64(1):386–399, 2015.

Παράρτημα Α΄

Κώδικες Υλοποίησης στον NS-3

Α΄.1 Εξαγωγή πιθανοτήτων επιτυχούς αποστολή πλαισίου για τα data rates του 802.11n

```
#include "ns3/core-module.h"
#include "ns3/yans-error-rate-model.h"
#include "ns3/nist-error-rate-model.h"
#include "ns3/gnuplot.h"

#include <fstream>
#include <vector>
#include <cmath>

using namespace ns3;

int main (int argc, char *argv [])
{
    uint32_t FrameSize = 2000;
    std::ofstream nistfile ("logs/DataRatesFigures/nist-frame-
        success-rate-n.plt");
    std::vector <std::string> modes;

    modes.push_back ("HtMcs0");
    modes.push_back ("HtMcs1");
    modes.push_back ("HtMcs2");
    modes.push_back ("HtMcs3");
    modes.push_back ("HtMcs4");
    modes.push_back ("HtMcs5");
    modes.push_back ("HtMcs6");
```

```

modes.push_back ("HtMcs7");

CommandLine cmd;
cmd.AddValue ("FrameSize", "The frame size", FrameSize);
cmd.Parse (argc, argv);

Gnuplot nistplot = Gnuplot ("nist-frame-success-rate-n.eps");
WifiTxVector txVector;

Ptr <NistErrorRateModel> nist = CreateObject<
    NistErrorRateModel> ();

for (uint32_t i = 0; i < modes.size (); i++)
{
    std::cout << modes[i] << std::endl;
    Gnuplot2dDataset nistdataset (modes[i]);
    txVector.SetMode (modes[i]);

    for (double snr = 0; snr <= 30.0; snr += 0.001)
    {
        double ps = nist->GetChunkSuccessRate (WifiMode (modes
            [i]), txVector, std::pow (10.0, snr / 10.0),
            FrameSize * 8);
        if (ps < 0 || ps > 1)
        {
            //error
            return 0;
        }
        nistdataset.Add (snr, ps);
    }

    nistplot.AddDataset (nistdataset);
}

nistplot.SetTerminal ("postscript eps color enh \\"Times-
    BoldItalic\");
nistplot.SetLegend ("SNR(dB)", "Frame Success Rate");
nistplot.SetExtra ("set xrange [-5:30]\n\
set yrange [0:1.2]\n\
set style line 1 linewidth 5\n\

```

```

set style line 2 linewidth 5\n\
set style line 3 linewidth 5\n\
set style line 4 linewidth 5\n\
set style line 5 linewidth 5\n\
set style line 6 linewidth 5\n\
set style line 7 linewidth 5\n\
set style line 8 linewidth 5\n\
set style increment user”

);

nistplot.GenerateOutput (nistfile);
nistfile.close ();
}

```

Α'.2 Πείραμα για εξαγωγή SNR κόμβου ως προς την απόσταση από το Access Point

```

#include <string.h>

#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/applications-module.h"
#include "ns3/wifi-module.h"
#include "ns3/mobility-module.h"
#include "ns3/ipv4-global-routing-helper.h"
#include "ns3/internet-module.h"
#include "ns3/netanim-module.h"
#include "src/core/model/double.h"
#include "ns3/flow-monitor-helper.h"
#include "ns3/propagation-loss-model.h"
#include "src/wifi/model/interference-helper.h"
#include "src/wifi/model/yans-wifi-phy.h"
#include "src/network/Utils/ipv4-address.h"
#include "src/wifi/model/snr-tag.h"
#include "src/stats/model/gnuplot.h"
#include "src/wifi/helper/yans-wifi-helper.h"
#include "src/core/model/integer.h"
#include "src/core/model/log.h"
#include "ns3/olsr-helper.h"
#include "src/network/helper/application-container.h"

```

```

#include "ns3/wave-mac-helper.h"

#define LOG_DIR "logs/DistanceCase/"
#define TIME 60

using namespace ns3;

NSLOG_COMPONENT_DEFINE("Distance Wifi");

/* Metrics variables */
std::map<double, double> snr;
std::map<double, double> snrAp;

NodeContainer wifiNodes;
NodeContainer wifiApNode;

void CreateDatasets() {
    std::ofstream File;
    std::string outputFile;
    std::map<double, double>::iterator it;

    outputFile.append(LOG_DIR).append("snrDistance.dat");
    File.open(outputFile.c_str(), std::ios::trunc);
    File << "#X\tY" << std::endl;

    for (it = snr.begin(); it != snr.end(); ++it)
        File <<" " << it->first << "\t" << it->second << std::
            endl;
    File.close();

    outputFile.clear();
    outputFile.append(LOG_DIR).append("snrApDistance.dat");
    File.open(outputFile.c_str(), std::ios::trunc);
    File << "#X\tY" << std::endl;

    for (it = snrAp.begin(); it != snrAp.end(); ++it)
        File <<" " << it->first << "\t" << it->second << std::
            endl;
    File.close();
}

```



```

}

/*Measure SNR on Access Point*/
void measureApRx(Ptr< const Packet > packet , uint16_t
channelFreqMhz , uint16_t channelNumber , uint32_t rate ,
WifiPreamble preamble , WifiTxVector txVector , struct mpduInfo
aMpdu , struct signalNoiseDbm signalNoise){
Ptr<MobilityModel> model1 = wifiNodes.Get(0)->GetObject<
MobilityModel>();
Ptr<MobilityModel> model2 = wifiApNode.Get(0)->GetObject<
MobilityModel>();

double d = model1->GetDistanceFrom(model2);

snrAp.insert(std::pair<double , double>(d , signalNoise.signal
));
}

```

```

/*Measure Rx on node side*/
void measureRx(Ptr< const Packet > packet , uint16_t
channelFreqMhz , uint16_t channelNumber , uint32_t rate ,
WifiPreamble preamble , WifiTxVector txVector , struct mpduInfo
aMpdu , struct signalNoiseDbm signalNoise){
Ptr<MobilityModel> model1 = wifiNodes.Get(0)->GetObject<
MobilityModel>();
Ptr<MobilityModel> model2 = wifiApNode.Get(0)->GetObject<
MobilityModel>();

double d = model1->GetDistanceFrom(model2);

snr.insert(std::pair<double , double>(d , signalNoise.signal )
);
}

```

```

/*Set position of a node*/
static void
SetPosition(Ptr<Node> node , Vector position) {
Ptr<MobilityModel> mobility = node->GetObject<MobilityModel>
();
}

```

```

        mobility->SetPosition(position);
    }

    int main(int argc, char *argv[]) {

        LogComponentEnable("Distance Wifi", LOG_LEVEL_ALL);

        CommandLine cmd;
        cmd.Parse(argc, argv);

        /* Disable RTS/CTS all the time. */
        Config::SetDefault("ns3::WifiRemoteStationManager::
            FragmentationThreshold", StringValue("1500"));
        Config::SetDefault("ns3::WifiRemoteStationManager::
            RtsCtsThreshold", StringValue("1500"));
        Config::SetDefault("ns3::WifiRemoteStationManager::
            NonUnicastMode", StringValue(WifiPhy::GetHtMcs7().
            GetUniqueName()));
        Config::SetDefault("ns3::LogDistancePropagationLossModel::
            ReferenceLoss", DoubleValue(40.0));

        /* Creating Nodes. */
        NSLOG_INFO("Creating Wi-Fi Network.");
        wifiApNode.Create(1);
        wifiNodes.Create(1);

        /* Channel Creation.
         * Light Speed Propagation
         * Log Distance Propagation Model with Reference Path Loss
           38 dB */
        YansWifiChannelHelper channel = YansWifiChannelHelper::
            Default();
        channel.AddPropagationLoss("ns3::
            NakagamiPropagationLossModel", "m0", DoubleValue(200));
        Ptr<YansWifiChannel> ch = channel.Create();

        /* PHY Layer Creation*/
        YansWifiPhyHelper phyAp = YansWifiPhyHelper::Default();
        phyAp.Set("TxPowerStart", DoubleValue(25.0));
    }

```

```

phyAp.Set("TxPowerEnd", DoubleValue(25.0));
phyAp.SetChannel(ch);

YansWifiPhyHelper phySta = YansWifiPhyHelper::Default();
phySta.Set("TxPowerStart", DoubleValue(20.0));
phySta.Set("TxPowerEnd", DoubleValue(20.0));
phySta.SetChannel(ch);

/*Set Wifi mode to 802.11n*/
WifiHelper wifi;
wifi.SetStandard(WIFI_PHY_STANDARD_80211n_2_4GHZ);

WifiHelper wifiAp;
wifiAp.SetStandard(WIFI_PHY_STANDARD_80211n_2_4GHZ);

HtWifiMacHelper mac = HtWifiMacHelper::Default();
Ssid ssid = Ssid("Shire");

/*Initialize rate manager. We assume that the nodes start
transmitting using 65 Mbps data rate.*/
wifiAp.SetRemoteStationManager("ns3::ConstantRateWifiManager",
    "DataMode", StringValue(WifiPhy::GetHtMcs7().
    GetUniqueName()), "ControlMode", StringValue(WifiPhy::
    GetHtMcs7().GetUniqueName()));
wifi.SetRemoteStationManager("ns3::ConstantRateWifiManager",
    "DataMode", StringValue(WifiPhy::GetHtMcs7().
    GetUniqueName()), "ControlMode", StringValue(WifiPhy::
    GetHtMcs7().GetUniqueName()));

/*Initialize Net Devices*/
NetDeviceContainer apDevice;
NetDeviceContainer staDevices;

mac.SetType("ns3::ApWifiMac", "Ssid", SsidValue(ssid));
apDevice = wifiAp.Install(phyAp, mac, wifiApNode);

mac.SetType("ns3::StaWifiMac", "Ssid", SsidValue(ssid), "
ActiveProbing", BooleanValue(false));
staDevices = wifi.Install(phySta, mac, wifiNodes);

```

```

/* Setting Motion */
MobilityHelper mobility;
mobility.SetMobilityModel("ns3::
    ConstantPositionMobilityModel");

/* Set AP at the center of the plane*/
mobility.Install(wifiApNode);
SetPosition(wifiApNode.Get(0), Vector(0.0, 0.0, 0.0));
mobility.SetMobilityModel("ns3::
    ConstantVelocityMobilityModel");
mobility.Install(wifiNodes);
Ptr<ConstantVelocityMobilityModel> velocityModel =
    DynamicCast<ConstantVelocityMobilityModel>(wifiNodes.Get
        (0)->GetObject<MobilityModel>());
velocityModel->SetPosition(Vector(2.0, 0.0, 0.0));
velocityModel->SetVelocity(Vector(1.4, 0.0, 0.0));

/* Internet stack*/
InternetStackHelper stack;
stack.Install(wifiApNode);
stack.Install(wifiNodes);

Ipv4AddressHelper address;
address.SetBase("10.1.3.0", "255.255.255.0");
Ipv4InterfaceContainer wifiNodesInterfaces;
Ipv4InterfaceContainer apNodeInterface;
apNodeInterface = address.Assign(apDevice);
wifiNodesInterfaces = address.Assign(staDevices);

/* Applications*/
u_int t = TIME;
uint16_t port = 50000;
Ptr<UniformRandomVariable> var = CreateObject<
    UniformRandomVariable> ();

Config::SetDefault("ns3::TcpSocket::SegmentSize",
    UIntegerValue(1500));
Config::SetDefault("ns3::TcpSocket::RcvBufSize",
    UIntegerValue(900000));

```

```

Config :: SetDefault ("ns3 :: TcpSocket :: SndBufSize" ,
    UIntegerValue(900000));

Address sinkLocalAddress(InetSocketAddress(Ipv4Address ::
    GetAny() , port));

/* Install sinks in all Wi-Fi Nodes*/
PacketSinkHelper sink("ns3 :: UdpSocketFactory" ,
    sinkLocalAddress);
ApplicationContainer sinkApp = sink.Install(wifiNodes);
sinkApp.Add(sink.Install(wifiApNode));
sinkApp.Start(Seconds(1.0));
sinkApp.Stop(Seconds(t + 1));

/* Install applications in AP and STAs*/
ApplicationContainer serverApp;

/* Access Point → Node*/
OnOffHelper server("ns3 :: UdpSocketFactory" , Ipv4Address ::
    GetAny ());
server.SetAttribute("OnTime" , StringValue("ns3 ::
    UniformRandomVariable [Min=1.2|Max=1.6]"));
server.SetAttribute("OffTime" , StringValue("ns3 ::
    UniformRandomVariable [Min=1.2|Max=1.6]"));
AddressValue remoteAddress(InetSocketAddress(
    wifiNodesInterfaces.GetAddress(0) , port));
server.SetAttribute("Remote" , remoteAddress);
server.SetAttribute("DataRate" , DataRateValue(DataRate("2
    Mbps")));
server.SetAttribute("PacketSize" , UIntegerValue(1000));
double value = var->GetValue(1.0 , 5.0);
NS_LOG_INFO("Access Point → Node start time: " << value);
server.SetAttribute("StartTime" , TimeValue(Seconds(value)));
value = var->GetValue(t , t + 1);
NS_LOG_INFO("Access Point → Node stop time: " << value);
server.SetAttribute("StopTime" , TimeValue(Seconds(value)));

serverApp = server.Install(wifiApNode.Get(0));

```

```

remoteAddress.Set(InetSocketAddress(apNodeInterface.
    GetAddress(0), port));
server.SetAttribute("Remote", remoteAddress);
value = var->GetValue(1.0, 5.0);
NSLOG_INFO("Node —> Access Point start time: " << value);
server.SetAttribute("StartTime", TimeValue(Seconds(value)));
value = var->GetValue(t - 2, t - 1);
NSLOG_INFO("Node —> Access Point stop time: " << value);
server.SetAttribute("StopTime", TimeValue(Seconds(value)));
serverApp.Add(server.Install(wifiNodes.Get(0)));

Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

Config::ConnectWithoutContext("/NodeList/0/DeviceList/*/ns3
    ::WifiNetDevice/Phy/MonitorSnifferRx", MakeCallback(&
    measureApRx));
Config::ConnectWithoutContext("/NodeList/1/DeviceList/*/ns3
    ::WifiNetDevice/Phy/MonitorSnifferRx", MakeCallback(&
    measureRx));

Ptr<FlowMonitor> flowMonitor;
FlowMonitorHelper flowHelper;
flowMonitor = flowHelper.InstallAll();

Simulator::Stop(Seconds(t));
Simulator::Run();
Simulator::Destroy();

flowMonitor->SerializeToXmlFile(LOG_DIR + std::string("
    flow_monitor.xml"), true, true);
CreateDatasets();

return 0;
}

```

A'3 Σενάριο προσομοίωσης του μηχανισμού του κεφαλαίου 7

```

#include <string.h>

#include "ns3/flow-monitor-module.h"
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/applications-module.h"
#include "ns3/wifi-module.h"
#include "ns3/mobility-module.h"
#include "ns3/ipv4-global-routing-helper.h"
#include "ns3/internet-module.h"
#include "ns3/netanim-module.h"
#include "src/core/model/double.h"
#include "ns3/flow-monitor-helper.h"
#include "ns3/propagation-loss-model.h"
#include "src/wifi/model/interference-helper.h"
#include "src/wifi/model/yans-wifi-phy.h"
#include "src/network/utls/ipv4-address.h"
#include "src/wifi/model/snr-tag.h"
#include "src/stats/model/gnuplot.h"
#include "src/wifi/helper/yans-wifi-helper.h"
#include "src/core/model/integer.h"
#include "src/core/model/log.h"
#include "ns3/olsr-helper.h"
#include "src/network/helper/application-container.h"
#include "ns3/wave-mac-helper.h"

#define LOG_DIR "logs/TpcCase/"
#define TIME 100

using namespace ns3;

NS_LOG_COMPONENT_DEFINE("Tpc Wifi");

std::map<Ipv4Address, signalNoiseDbm> nodeList;
std::map<std::string, double> snrThresholds;
std::vector<double> txMeasurements;
NodeContainer wifiNodes;
NodeContainer wifiApNode;
std::map<Ipv4Address, double> recentReadings;
std::string mcsRate("HtMcs0");
uint32_t numNodes = 1;

```

```

bool raEnabled = false;
bool mEnabled = true;
double notifyInterval = 1.0;
double mOffset = 1.0;

void RecordData(Ptr<FlowMonitor> monitor){
    std::ofstream File;
    std::string outputFile;
    uint32_t totalPackets, lostPackets = 0;
    double packetLoss;

    NSLOG_INFO("———Experiment statistics ——");

    FlowMonitorHelper flowmon;
    Ptr<Ipv4FlowClassifier> classifier = DynamicCast<
        Ipv4FlowClassifier> (flowmon.GetClassifier ());
    std::map<FlowId, FlowMonitor::FlowStats> stats = monitor->
        GetFlowStats ();

    for (std::map<FlowId, FlowMonitor::FlowStats>::
        const_iterator iter = stats.begin (); iter != stats.end
        ()); ++iter)
    {
        //Ipv4FlowClassifier::FiveTuple t = classifier->FindFlow
            (iter->first);
        totalPackets += iter->second.txPackets;
        lostPackets += iter->second.lostPackets;

        //      NSLOG_UNCOND("Flow ID: " << iter->first << " Src Addr
            " << t.sourceAddress << " Dst Addr " << t.destinationAddress
            );
        //      NSLOG_UNCOND("Throughput: " << iter->second.rxBytes *
            8.0 / (iter->second.timeLastRxPacket.GetSeconds()-iter->
            second.timeFirstTxPacket.GetSeconds()) / 1024 << " Kbps
            ");
        //      NSLOG_UNCOND("Packet loss %= " << ((iter->second.
            txPackets-iter->second.rxPackets)*1.0)/iter->second.txPackets
            );
    }

    packetLoss = (lostPackets*1.0)/totalPackets;

```



```

outputFile.append(LOG_DIR);
if(mEnabled){
    std::ostringstream str;
    str << notifyInterval << "-" << mOffset;
    std::string str = str.str();
    outputFile.append("packetLoss_").append(str).append(".
        dat");
}
else
    outputFile.append("packetLoss_no_mechanism.dat");

File.open(outputFile.c_str(), std::ios::app);
File << " " << mcsRate.at(mcsRate.size() - 1) << "\t" <<
    packetLoss * 100 << std::endl;
File.close();

outputFile.clear();
outputFile.append(LOG_DIR);
if(mEnabled){
    std::ostringstream str;
    str << notifyInterval << "-" << mOffset;
    std::string str = str.str();
    outputFile.append("txPower_").append(str).append(".dat")
        ;
} else
    outputFile.append("txPower_no_mechanism.dat");
double sum = 0;
for(uint32_t i = 0; i < txMeasurements.size(); i++)
    sum += txMeasurements.at(i);

File.open(outputFile.c_str(), std::ios::app);
File << " " << mcsRate.at(mcsRate.size() - 1) << "\t" << sum
    /((int)txMeasurements.size()) << std::endl;
File.close();
}

static Ipv4Address GetPacketSource(Ptr<const Packet> packet) {
    Ipv4Address src_ip;

    Ptr<Packet> q = packet->Copy();

```

```

PacketMetadata::ItemIterator metadataIterator = q->BeginItem
    ();
PacketMetadata::Item item;
while (metadataIterator.HasNext()) {
    item = metadataIterator.Next();

    // If we want to have an ip header
    if (item.tid.GetName() == "ns3::Ipv4Header") {
        Callback<ObjectBase*> constr = item.tid.
            GetConstructor();
        NS_ASSERT(!constr.IsNull());

        // Ptr<> and DynamicCast<> won't work here as all
        // headers are from ObjectBase, not Object
        ObjectBase *instance = constr();
        NS_ASSERT(instance != 0);

        Ipv4Header* ipv4Header = dynamic_cast<Ipv4Header*> (
            instance);
        NS_ASSERT(ipv4Header != 0);

        ipv4Header->Deserialize(item.current);

        // The ipv4Header can now obtain the source of the
        // packet
        src_ip = ipv4Header->GetSource();

        // Finished, clear the ip header and go back
        delete ipv4Header;
        return src_ip;
        break;
    }
}
return Ipv4Address::GetAny();
}

/*Measure current TxPower Levels for each node*/
void MeasureTx(){
    for(uint32_t i = 0; i < wifiNodes.GetN(); i++){
        Ptr<Node> node = wifiNodes.Get(i);

```

```

    Ptr<NetDevice> device = node->GetDevice(0);
    Ptr<WifiNetDevice> wifiDevice = DynamicCast<
        WifiNetDevice> (device);
    Ptr<YansWifiPhy> phy = DynamicCast<YansWifiPhy>(
        wifiDevice->GetPhy());

    txMeasurements.push_back(phy->GetTxPowerStart());
}

Simulator::Schedule(Seconds(0.1), &MeasureTx);
}

/*Measure SNR on Access Point*/
static void measureApRx(Ptr< const Packet > packet, uint16_t
    channelFreqMhz, uint16_t channelNumber, uint32_t rate,
    WifiPreamble preamble, WifiTxVector txVector, struct mpduInfo
    aMpdu, struct signalNoiseDbm signalNoise) {
    if(mEnabled && packet->GetSize() > 1000){
        Ipv4Address sourceAddress = GetPacketSource(packet);
        if(sourceAddress != Ipv4Address::GetAny()){
            std::map<Ipv4Address, signalNoiseDbm>::iterator it =
                nodeList.find(sourceAddress);
            if (it != nodeList.end())
                it->second = signalNoise;
            else
                nodeList.insert(std::pair<Ipv4Address,
                    signalNoiseDbm>(sourceAddress, signalNoise));
        }
    }
}

static void PacketReceived(Ptr<Socket> socket) {
    if(mEnabled){
        Ptr<Packet> p = socket->Recv();
        Ptr<Node> node = socket->GetNode();
        NotifyHeader header;
        p->RemoveHeader(header);
        std::map<Ipv4Address, signalNoiseDbm> nodeList = header
            .GetData();
    }
}

```

```

Address addr;
socket->GetSockName (addr);
InetSocketAddress iaddr = InetSocketAddress::ConvertFrom
    (addr);

std::map<Ipv4Address, signalNoiseDbm>::iterator it =
    nodeList.find(iaddr.GetIpv4());
std::map<Ipv4Address, double>::iterator it2 =
    recentReadings.find(iaddr.GetIpv4());

if (it != nodeList.end()){
    signalNoiseDbm stats = it->second;
    double snrTarget = snrThresholds.at(mcsRate);
    Ptr<NetDevice> device = node->GetDevice(0);
    Ptr<WifiNetDevice> wifiDevice = DynamicCast<
        WifiNetDevice> (device);
    Ptr<YansWifiPhy> phy = DynamicCast<YansWifiPhy>(
        wifiDevice->GetPhy());

    double pathLoss = phy->GetTxPowerStart() + phy->
        GetTxGain() - stats.signal;
    double newTxPower = snrTarget + pathLoss + stats.
        noise + mOffset;
//      NS_LOG_INFO("Node " << node->GetId() << ": Tx=" <<
// phy->GetTxPowerStart() <<
//      "\tRx=" << stats.signal <<
//      "\tPathLoss=" << pathLoss <<
//      "\tNoise=" << stats.noise <<
//      "\tTargetSNR=" << snrTarget <<
//      "\tSNR=" << stats.signal - stats.noise <<
//      "\tTxPower=" << newTxPower);
    if(newTxPower < 5){
        phy->SetTxPowerEnd(5.0);
        phy->SetTxPowerStart(5.0);
    } else if (newTxPower > 20){
        phy->SetTxPowerEnd(20);
        phy->SetTxPowerStart(20);
    } else {
        phy->SetTxPowerEnd(newTxPower);
        phy->SetTxPowerStart(newTxPower);

```

```

    }

    if(it2 != recentReadings.end()){
        if(it2->second == it->second.signal){
            NSLOG_INFO("[Node " << socket->GetNode()->
                GetId() << "] Signal value is the same
                after 2 AP broadcasts. Reseting TxPower
                to 20 dBm.");
            phy->SetTxPowerEnd(20);
            phy->SetTxPowerStart(20);
        }
        it2->second = it->second.signal;
    }else{
        recentReadings.insert(std::pair<Ipv4Address ,
            double>(iaddr.GetIpv4(), it->second.signal));
    }
}
else
    NSLOG_INFO("Node " << socket->GetNode()->GetId() <<
        ": not in the list yet");
}
}
}

```

```

static void NotifyNodes(Ptr<Socket> socket, Time pktInterval) {

    NSLOG_INFO("——STATUS TABLE——");
    for (std::map<Ipv4Address, signalNoiseDbm>::iterator it=
        nodeList.begin(); it!=nodeList.end(); ++it){
        std::ostringstream oss;
        std::string ip;

        it->first.Print(oss);
        ip = oss.str();
        NSLOG_INFO(ip <<"\t"<< it->second.signal <<"\t" << it->
            second.noise);
    }
    uint32_t size = sizeof(nodeList);

    Ptr<Packet> p = Create<Packet>(size);
    NotifyHeader header;
}

```

```

    header.SetData(nodeList);
    p->AddHeader(header);
    socket->Send(p);
    Simulator::Schedule(pktInterval, &NotifyNodes, socket,
        pktInterval);
}

static void GenerateTraffic(Ptr<Socket> socket, uint32_t pktSize
,
    uint32_t pktCount, Time pktInterval) {
    if (pktCount > 0) {
        socket->Send(Create<Packet>(pktSize));
        Simulator::Schedule(pktInterval, &GenerateTraffic,
            socket, pktSize, pktCount - 1, pktInterval);
    } else {
        socket->Close();
    }
}

/*Set position of a node*/
static void SetPosition(Ptr<Node> node, Vector position) {
    Ptr<MobilityModel> mobility = node->GetObject<MobilityModel>
        ();
    mobility->SetPosition(position);
}

int main(int argc, char *argv[]) {

    LogComponentEnable("Tpc Wifi", LOG_LEVEL_ALL);

    snrThresholds.insert(std::pair<std::string, double>("HtMcs0"
        , 7));
    snrThresholds.insert(std::pair<std::string, double>("HtMcs1"
        , 10));
    snrThresholds.insert(std::pair<std::string, double>("HtMcs2"
        , 13));
    snrThresholds.insert(std::pair<std::string, double>("HtMcs3"
        , 17));
    snrThresholds.insert(std::pair<std::string, double>("HtMcs4"
        , 20));
    snrThresholds.insert(std::pair<std::string, double>("HtMcs5"

```

```

    , 25));
snrThresholds.insert(std::pair<std::string, double>("HtMcs6"
    , 26));
snrThresholds.insert(std::pair<std::string, double>("HtMcs7"
    , 27));

CommandLine cmd;
cmd.AddValue("nodes", "Number of nodes in the network",
    numNodes);
cmd.AddValue("mcs", "Data rate MCS index for network nodes",
    mcsRate);
cmd.AddValue("ra", "Rate adaptation en", raEnabled);
cmd.AddValue("mInterval", "Mechanism notification interval
    in seconds", notifyInterval);
cmd.AddValue("mEnabled", "True to enable mechanism, false
    otherwise", mEnabled);
cmd.AddValue("mOffset", "Mechanism offset value in dBm",
    mOffset);
cmd.Parse(argc, argv);

/* Disable RTS/CTS all the time. */
Config::SetDefault("ns3::WifiRemoteStationManager::
    FragmentationThreshold", StringValue("2200"));
Config::SetDefault("ns3::WifiRemoteStationManager::
    RtsCtsThreshold", StringValue("2200"));
Config::SetDefault("ns3::LogDistancePropagationLossModel::
    ReferenceLoss", DoubleValue(40.0));

/* Creating Nodes. */
NS_LOG_INFO("Creating Wi-Fi Networks.");
wifiApNode.Create(1);
wifiNodes.Create(numNodes);

/* Channel Creation.
    * Light Speed Propagation
    * Log Distance Propagation Model with Reference Path Loss
    38 dB */
YansWifiChannelHelper channel = YansWifiChannelHelper::
    Default();
//channel.AddPropagationLoss("ns3::
    NakagamiPropagationLossModel", "m0", DoubleValue(150));

```

```

Ptr<YansWifiChannel> ch = channel.Create();

/* PHY Layer Creation*/
YansWifiPhyHelper phy = YansWifiPhyHelper::Default();
phy.Set("CcaMode1Threshold", DoubleValue(0.0));

phy.Set("TxPowerStart", DoubleValue(25.0));
phy.Set("TxPowerEnd", DoubleValue(25.0));
phy.SetChannel(ch);

/*Set Wifi mode to 802.11n*/
WifiHelper wifi;
wifi.SetStandard(WIFI_PHY_STANDARD_80211n_2.4GHZ);

HtWifiMacHelper mac = HtWifiMacHelper::Default();
Ssid ssid = Ssid("Shire");

if (raEnabled) {
    NS_LOG_INFO("Rate adaptation enabled, mcs parameter is
        omitted");
    wifi.SetRemoteStationManager("ns3::MinstrelHtWifiManager
        ");
} else {
    NS_LOG_INFO("Rate set to " << mcsRate);
    wifi.SetRemoteStationManager("ns3::
        ConstantRateWifiManager", "DataMode", StringValue(
        mcsRate), "ControlMode", StringValue(mcsRate), "
        NonUnicastMode", StringValue(mcsRate));
}

/*Initialize Net Devices*/
NetDeviceContainer apDevice;
NetDeviceContainer staDevices;

mac.SetType("ns3::ApWifiMac", "Ssid", SsidValue(ssid), "
    QosSupported", BooleanValue(false));
apDevice = wifi.Install(phy, mac, wifiApNode);

phy.Set("TxPowerStart", DoubleValue(20.0));
phy.Set("TxPowerEnd", DoubleValue(20.0));
mac.SetType("ns3::StaWifiMac", "Ssid", SsidValue(ssid), "

```



```

    ActiveProbing", BooleanValue(false), "QosSupported",
    BooleanValue(false));
staDevices = wifi.Install(phy, mac, wifiNodes);

/* Setting Motion */
MobilityHelper mobility;
mobility.SetMobilityModel("ns3::
    ConstantPositionMobilityModel");

/* Set AP at the center of the plane*/
mobility.Install(wifiApNode);
setPosition(wifiApNode.Get(0), Vector(0.0, 0.0, 0.0));

mobility.SetPositionAllocator("ns3::
    RandomDiscPositionAllocator",
    "X", DoubleValue(0.0),
    "Y", DoubleValue(0.0),
    "Rho", StringValue("ns3::UniformRandomVariable[Min
        =0|Max=60]"));
mobility.SetMobilityModel("ns3::RandomWalk2dMobilityModel",
    "Bounds", StringValue("-60|60|-60|60"),
    "Distance", DoubleValue(60),
    "Speed", StringValue("ns3::UniformRandomVariable[Min
        =1.2|Max=1.6]"));
mobility.Install(wifiNodes);

/* Internet stack*/

InternetStackHelper stack;
stack.Install(wifiApNode);
stack.Install(wifiNodes);

Ipv4AddressHelper address;
address.SetBase("10.1.3.0", "255.255.255.0");

Ipv4InterfaceContainer wifiNodesInterfaces;
Ipv4InterfaceContainer apNodeInterface;
apNodeInterface = address.Assign(apDevice);
wifiNodesInterfaces = address.Assign(staDevices);

```

```

/* Applications */
u_int t = TIME;
Ptr<UniformRandomVariable> rng = CreateObject<
    UniformRandomVariable>();

/* Access point broadcast app */
TypeId tid = TypeId::LookupByName("ns3::UdpSocketFactory");
Ptr<Socket> apSocket = Socket::CreateSocket(wifiApNode.Get
    (0), tid);
InetSocketAddress local = InetSocketAddress(Ipv4Address::
    GetAny(), 1);
apSocket->Bind(local);
InetSocketAddress remote = InetSocketAddress("
    255.255.255.255", 80);
apSocket->SetAllowBroadcast(true);
apSocket->Connect(remote);

Simulator::ScheduleWithContext(apSocket->GetNode()->GetId(),
    Seconds(0.1225), &NotifyNodes,
    apSocket, Seconds(notifyInterval));

/* Set up sockets for nodes */
for (uint32_t i = 0; i < wifiNodes.GetN(); i++) {
    Ptr<Socket> nodeSocket = Socket::CreateSocket(wifiNodes.
        Get(i), tid);
    InetSocketAddress local = InetSocketAddress(
        wifiNodesInterfaces.GetAddress(i), 80);
    nodeSocket->Bind(local);
    nodeSocket->SetRecvCallback(MakeCallback(&PacketReceived
        ));
    InetSocketAddress remoteAp = InetSocketAddress(
        apNodeInterface.GetAddress(0), 1);
    nodeSocket->Connect(remoteAp);

    Simulator::ScheduleWithContext(nodeSocket->GetNode()->
        GetId(),
        Seconds(rng->GetValue(1.0, 2.0)), &
        GenerateTraffic,
        nodeSocket, 1500, 9900, Seconds(0.01));
}

```

```

}

Config::ConnectWithoutContext("/NodeList/0/DeviceList/*/ns3
    ::WifiNetDevice/Phy/MonitorSnifferRx", MakeCallback(&
    measureApRx));
Simulator::Schedule(Seconds(0.1), &MeasureTx);
Ptr<FlowMonitor> flowMonitor;
FlowMonitorHelper flowHelper;
flowMonitor = flowHelper.InstallAll();

Simulator::Stop(Seconds(t));
Simulator::Run();
Simulator::Destroy();

flowMonitor->SerializeToXmlFile(LOG_DIR + std::string("
    flow_monitor.xml"), true, true);

RecordData(flowMonitor);

return 0;
}

```

Α'.4 Σενάριο προσομοίωσης του μηχανισμού του κεφα- λαίου 8

```

#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/applications-module.h"
#include "ns3/mobility-module.h"
#include "ns3/wifi-module.h"
#include "ns3/olsr-module.h"
#include "ns3/internet-module.h"
#include "ns3/flow-monitor-helper.h"
#include "ns3/energy-module.h"
#include "ns3/wifi-net-device.h"
#include "ns3/constant-rate-wifi-manager.h"

#include <iostream>
#include <vector>
#include <sstream>

```

```

#define OFFSET 6
#define LOG_DIR "logs/"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE("PowerCtrlOLSR");

std::vector<double> TransmissionPower [100];
uint32_t rateMcs_, numNodes_;
std::vector<int> snrForRates;
uint32_t totalPacketSent = 0;
uint32_t olsrPackets = 0;
uint32_t otherPackets = 0;
uint32_t flowPacketsTx = 0;
uint32_t flowPacketsRx = 0;
uint32_t flowPacketsDropped = 0;
double duration = 100;
Ptr<FlowMonitor> flowMonitor;

void GetPacketStatistics() {
    std::map<FlowId, FlowMonitor::FlowStats> stats = flowMonitor
        ->GetFlowStats();

    for (std::map<FlowId, FlowMonitor::FlowStats>::
        const_iterator iter = stats.begin(); iter != stats.end();
        ++iter) {
        flowPacketsDropped += iter->second.lostPackets;
        flowPacketsTx += iter->second.txPackets;
        flowPacketsRx += iter->second.rxPackets;
    }
}

void ExportTransmissionPower() {

    std::ofstream logs;

    GetPacketStatistics();
}

```

```

NS_LOG_INFO("——PHY LAYER STATISTICS——\n");
NS_LOG_INFO("Total Packets Sent: " << totalPacketSent);
NS_LOG_INFO("Total Packets for Apps/Other " << otherPackets)
    ;
NS_LOG_INFO("Total Packets for OLSR " << olsrPackets << "\n"
    );

NS_LOG_INFO("——FLOW MONITOR STATISTICS——\n");
NS_LOG_INFO("Total Packets Sent: " << flowPacketsTx);
NS_LOG_INFO("Total Packets Received: " << flowPacketsRx);
NS_LOG_INFO("Total Packets Dropped " << flowPacketsDropped);

std::ostream str1;
str1 << "TxPower-"<< rateMcs_ << ".dat";
std::string outputFile;
outputFile.append(LOG_DIR).append(str1.str());
logs.open(outputFile.c_str(), std::ios::app);

uint32_t activeNodes = numNodes_;
//Calculate Average Transmission Per Node
double averageTransmissionPerNode[numNodes_];
for(uint32_t i = 0; i < numNodes_; i++){
    std::vector<double> vect = TransmissionPower[i];
    averageTransmissionPerNode[i] = 0;
    if(vect.size() == 0){
        activeNodes--;
        continue;
    }
    for (std::vector<double>::iterator it = vect.begin(); it
        != vect.end(); ++it) {
        averageTransmissionPerNode[i] += *it;
    }
    averageTransmissionPerNode[i] =
        averageTransmissionPerNode[i]/vect.size();
}

double averageTransmission = 0;
for(uint32_t i = 0; i < numNodes_; i++){
    averageTransmission += averageTransmissionPerNode[i];
}

```

```

NSLOG_INFO(" Average Transmission Power: " <<
    averageTransmission / activeNodes);
logs << numNodes_ << " " << averageTransmission /
    activeNodes << std::endl;
logs.close();

outputFile.clear();
std::ostringstream str2;
str2 << "PacketLoss-"<<rateMcs_<<" .dat";
outputFile.append(LOG_DIR).append(str2.str());
logs.open(outputFile.c_str(), std::ios::app);
logs << numNodes_ << " " << (double)((flowPacketsDropped
    *100)/flowPacketsTx) << std::endl;
logs.close();

outputFile.clear();
std::ostringstream str3;
str3 << "Throughput-"<<rateMcs_<<" .dat";
outputFile.append(LOG_DIR).append(str3.str());
logs.open(outputFile.c_str(), std::ios::app);
logs << numNodes_ << " " << (flowPacketsRx*8*1000)/(duration
    *1000) << std::endl;
logs.close();

}

```

```

std::string GetPhyMode(uint32_t mcsRate){
    std::string phyMode;
    switch(mcsRate){
        case 0:
            phyMode = WifiPhy::GetOfdmRate6_5MbpsBW20MHz().
                GetUniqueName();
            break;
        case 1:
            phyMode = WifiPhy::GetOfdmRate13MbpsBW20MHz().
                GetUniqueName();
            break;
        case 2:

```

```

        phyMode = WifiPhy::GetOfdmRate19_5MbpsBW20MHz().
            GetUniqueName();
        break;
    case 3:
        phyMode = WifiPhy::GetOfdmRate26MbpsBW20MHz().
            GetUniqueName();
        break;
    case 4:
        phyMode = WifiPhy::GetOfdmRate39MbpsBW20MHz().
            GetUniqueName();
        break;
    case 5:
        phyMode = WifiPhy::GetOfdmRate52MbpsBW20MHz().
            GetUniqueName();
        break;
    case 6:
        phyMode = WifiPhy::GetOfdmRate58_5MbpsBW20MHz().
            GetUniqueName();
        break;
    case 7:
        phyMode = WifiPhy::GetOfdmRate65MbpsBW20MHz().
            GetUniqueName();
        break;
    }
    return phyMode;
}

std::vector<std::string> split(std::string str, char delimiter)
{
    std::vector<std::string> internal;
    std::stringstream ss(str); // Turn the string into a stream.
    std::string tok;

    while (getline(ss, tok, delimiter)) {
        internal.push_back(tok);
    }

    return internal;
}

Ipv4Header GetPacketHeader(Ptr<Packet> packet) {

```

```

Ptr<Packet> q = packet->Copy();
Ipv4Header header;
PacketMetadata::ItemIterator metadataIterator = q->BeginItem
    ();
PacketMetadata::Item item;
while (metadataIterator.HasNext()) {
    item = metadataIterator.Next();
    // If we want to have an ip header
    if (item.tid.GetName() == "ns3::Ipv4Header") {
        Callback<ObjectBase*> constr = item.tid.
            GetConstructor();
        NS_ASSERT(!constr.IsNull());

        // Ptr<> and DynamicCast<> won't work here as all
        // headers are from ObjectBase, not Object
        ObjectBase *instance = constr();
        NS_ASSERT(instance != 0);

        Ipv4Header* ipv4Header = dynamic_cast<Ipv4Header*> (
            instance);
        NS_ASSERT(ipv4Header != 0);

        ipv4Header->Deserialize(item.current);
        header = *ipv4Header;

        delete ipv4Header;
        break;
    }
}
return header;
}

```

```

void SetTxPower(std::string context, Ptr<const Packet> packet,
    WifiTxVector txVector, Ptr<YansWifiPhy> phy) {
    totalPacketSent++;
    std::vector<std::string> sep = split(context, '/');
    int nodeId = std::atoi(sep[2].c_str());
    Ptr<Node> node = NodeList::GetNode(nodeId);

    Ptr<ns3::Ipv4RoutingProtocol> rp(node->GetObject<Ipv4>()
        ->GetRoutingProtocol());
}

```



```

Ptr<ns3::olsr::RoutingProtocol> proto = Ipv4RoutingHelper::
    GetRouting <ns3::olsr::RoutingProtocol> (rp);
std::vector<ns3::olsr::RoutingTableEntry> entries = proto->
    GetRoutingTableEntries();
Ptr<Packet> p = packet->Copy();

Ipv4Header header = GetPacketHeader(p);
std::ostream osd;
header.GetDestination().Print(osd);

/* If it's an ARP or OLSR set to Max Power */
if(osd.str() == "102.102.102.102" || osd.str() == "
    10.1.1.255"){
    phy->SetTxPowerEnd(20.0);
    phy->SetTxPowerStart(20.0);
    olsrPackets++;
else{

otherPackets++;
for(std::vector<ns3::olsr::RoutingTableEntry>::
    const_iterator iter = entries.begin(); iter!=entries.end
    ()); iter++){
    if(iter->destAddr == header.GetDestination()) {
        if(iter->rxPowerNeigh != 0) {
            uint32_t snr = snrForRates.at(phy->WifiModeToMcs
                (txVector.GetMode()));
            double pathLoss = 20 - iter->rxPowerNeigh;
            double txPower = snr + phy->m_noise + pathLoss +
                OFFSET;
            //double realTxPower = txPower;
            if(txPower > 20)
                txPower = 20;
            if(txPower < 5)
                txPower = 5 + OFFSET;
            phy->SetTxPowerEnd(txPower);
            phy->SetTxPowerStart(txPower);
            TransmissionPower[nodeId].push_back(txPower);
            //NS_LOG_INFO("[Node "<<nodeId<<"] Calculated
                Power: "<< realTxPower << "\tPower Set to "
                <<txPower << "\tPathLoss Calculated:" <<
                pathLoss << "\tRxFromNeighbor:" << iter->

```

```

        rxPowerNeigh <<"\tNoise:" << phy->m_noise );
    }else {
        phy->SetTxPowerEnd(20.0);
        phy->SetTxPowerStart(20.0);
        TransmissionPower [ nodeId ] . push_back (20.0);
    }
}
}
}
}

```

```

Ptr<Socket> SetupPacketReceive(Ipv4Address addr, Ptr<Node> node)
{
    TypeId tid = TypeId::LookupByName("ns3::UdpSocketFactory");
    Ptr<Socket> sink = Socket::CreateSocket(node, tid);
    InetSocketAddress local = InetSocketAddress(addr, 9);
    sink->Bind(local);

    return sink;
}

```

```

int main(int argc, char *argv[]) {

//    LogComponentEnable("PowerCtrlOLSR", LOG_LEVEL_ALL);
//    LogComponentEnable("YansWifiPhy", LOG_LEVEL_INFO);
LogComponentEnable("PowerCtrlOLSR", LOG_LEVEL_INFO);
    /*Initialize rates*/
    snrForRates.push_back(7);
    snrForRates.push_back(10);
    snrForRates.push_back(13);
    snrForRates.push_back(17);
    snrForRates.push_back(20);
    snrForRates.push_back(25);
    snrForRates.push_back(26);
    snrForRates.push_back(27);

    /*Initialize parameters*/
bool mobile = true;
    uint32_t numNodes = 20; // Number of Nodes
int rateMcs = 0; //Mcs rate

```

```

/* Initialize CMD to enable visualization */
CommandLine cmd;
cmd.AddValue("numNodes", "Num of nodes ", numNodes);
cmd.AddValue("DataRate", "Mcs Code for rate", rateMcs);
cmd.AddValue("mobile", "Mobility on the nodes", mobile);
cmd.Parse(argc, argv);

numNodes_ = numNodes;
rateMcs_ = rateMcs;

NS_LOG_INFO("Creating Ad Hoc Network with " << numNodes << "
            nodes ... ");
std::string phyMode = GetPhyMode(rateMcs);
NS_LOG_INFO("Rate Selected: " << phyMode);

/* Disable Fragmentation */
Config::SetDefault("ns3::WifiRemoteStationManager::
    FragmentationThreshold", StringValue("2200"));
Config::SetDefault("ns3::WifiRemoteStationManager::
    RtsCtsThreshold", StringValue("500"));
Config::SetDefault("ns3::WifiRemoteStationManager::
    NonUnicastMode", StringValue(WifiPhy::
    GetOfdmRate58.5MbpsBW20MHz().GetUniqueName()));
Config::SetDefault("ns3::LogDistancePropagationLossModel::
    ReferenceLoss", DoubleValue(40));

/* Create the nodes */
NodeContainer nodes;
nodes.Create(numNodes);

/* Create the channel */
YansWifiChannelHelper wifiChannel = YansWifiChannelHelper::
    Default();

/* Create PHY Layer */
YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default();
wifiPhy.Set("TxPowerLevels", UIntegerValue(1));
wifiPhy.Set("TxGain", DoubleValue(0));
wifiPhy.Set("RxGain", DoubleValue(0));
wifiPhy.Set("TxPowerStart", DoubleValue(20.0));
wifiPhy.Set("TxPowerEnd", DoubleValue(20.0));

```

```

wifiPhy.SetChannel(wifiChannel.Create());

/* Create Wifi Layer*/
WifiHelper wifi = WifiHelper::Default();
wifi.SetStandard(WIFI_PHY_STANDARD_80211n_2_4GHZ);
wifi.SetRemoteStationManager("ns3::ConstantRateWifiManager",
    "DataMode", StringValue(phyMode),
    "ControlMode", StringValue(phyMode));

/* Create MAC Layer*/
HtWifiMacHelper mac = HtWifiMacHelper::Default();
mac.SetType("ns3::AdhocWifiMac", "Ssid", SsidValue(Ssid("
    Test Net")));

/* Create Devices*/
NetDeviceContainer devices = wifi.Install(wifiPhy, mac,
    nodes);

MobilityHelper mobility;
mobility.SetPositionAllocator("ns3::
    RandomRectanglePositionAllocator",
    "X", StringValue("ns3::UniformRandomVariable [Min=0|
    Max=150]"),
    "Y", StringValue("ns3::UniformRandomVariable [Min=0|
    Max=150]"));
if(mobile){
mobility.SetMobilityModel("ns3::RandomWalk2dMobilityModel",
    "Mode", StringValue("Time"),
    "Time", StringValue("15s"),
    "Speed", StringValue("ns3::UniformRandomVariable [Min
    =1.2|Max=1.4]"),
    "Bounds", StringValue("0|150|0|150"));
}
mobility.InstallAll();

/*Routing Installation*/
OlsrHelper olsr;
olsr.Set("HelloInterval", TimeValue(Seconds(2)));

```

```

/* Create Internet Stack */
InternetStackHelper internet;
internet.SetRoutingHelper(olsr);
internet.InstallAll();

/* IP Stack Install */
Ipv4AddressHelper ipv4;
NS_LOG_INFO("Assigning IP Addresses.");
ipv4.SetBase("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer interfaces = ipv4.Assign(devices);

/* Generate Traffic */
OnOffHelper onoff1("ns3::UdpSocketFactory", Address());
onoff1.SetAttribute("OnTime", StringValue("ns3::
    ConstantRandomVariable[Constant=1.0]"));
onoff1.SetAttribute("OffTime", StringValue("ns3::
    ConstantRandomVariable[Constant=0.0]"));
onoff1.SetAttribute("PacketSize", UIntegerValue(1000));
onoff1.SetAttribute("DataRate", DataRateValue(DataRate("500
    kb/s")));

ApplicationContainer apps;

Ptr<Socket> sink = SetupPacketReceive(interfaces.GetAddress
    (0), nodes.Get(0));
AddressValue remoteAddress(InetSocketAddress(interfaces.
    GetAddress(0), 9));
onoff1.SetAttribute("Remote", remoteAddress);
apps = onoff1.Install(nodes.Get(nodes.GetN() - 1));
apps.Start(Seconds(6.0));
apps.Stop(Seconds(duration));

NS_LOG_INFO("Number of applications in the network: " <<
    apps.GetN());

std::ostringstream oss;
oss.str("");
oss.clear();

```

```

oss << "NodeList/*/DeviceList/0/$ns3::WifiNetDevice/Phy/
TxStart";
Config::Connect(oss.str(), MakeCallback(&SetTxPower));

FlowMonitorHelper flowHelper;
flowMonitor = flowHelper.InstallAll();
Simulator::Schedule(Seconds(duration), &
ExportTransmissionPower);

Simulator::Stop(Seconds(duration));
Simulator::Run();
Simulator::Destroy();

flowMonitor->SerializeToXmlFile(LOG_DIR + std::string("flow
.xml"), true, true);
return 0;
}

```

