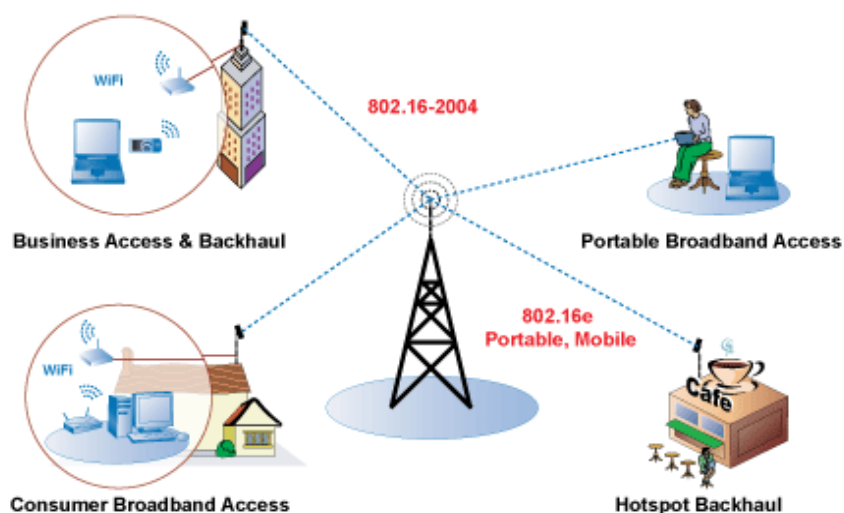




Πτυχιακή εργασία

## Προσομοιωτική μελέτη ασύρματων ευρυζωνικών δικτύων (IEEE 802.16 – WiMAX) μέσω του εργαλείου NS-2



Του φοιτητή  
Καρανίκα Γεωργίου  
Αρ. Μητρώου: 06/3114

Επιβλέπων καθηγητής  
Θωμάς Λάγκας

Θεσσαλονίκη 2011

## **1 ΠΡΟΛΟΓΟΣ**

Στα πλαίσια των προπτυχιακών σπουδών στο τμήμα Πληροφορικής του Αλεξάνδρειου Ανώτατου Τεχνολογικού Ιδρύματος Θεσσαλονίκης και καθώς ο κανονισμός ορίζει, ανατέθηκε η εκπόνηση πτυχιακής εργασίας ως αναπόσπαστο μέρος της ολοκλήρωσης των σπουδών. Η παρούσα πτυχιακή εργασία έχει τίτλο “Προσομοιωτική μελέτη ασύρματων ευρυζωνικών δικτύων (IEEE 802.16 – WiMAX) μέσω του εργαλείου NS-2” και επιβλέπων καθηγητής για την εκπόνησή της είναι ο Δρ. Θωμάς Λάγκας, Επιστημονικός Συνεργάτης του τμήματος Πληροφορικής του ΑΤΕΙ Θεσσαλονίκης.

Η πτυχιακή εργασία αυτή έρχεται να καλύψει την σύγχυση που υπάρχει στην επιστημονική κοινότητα όσον αφορά την χρήση του WiMAX στον NS-2 μιας και δεν υπάρχει κάποιο προ-εγκατεστημένο module στον προσομοιωτή, ενώ έχουν γραφεί και δημοσιοποιηθεί πολλές διαφορετικές υλοποιήσεις (modules) για την υποστήριξη του.

Οι πρώτες κινήσεις μετά την ανάθεση της πτυχιακής περιελάμβαναν την θεωρητική διερεύνηση τότε για το WiMAX όσο και για τον NS-2. Σχεδόν ταυτόχρονα δημιουργήθηκαν τα εικονικά μηχανήματα στα οποία στήθηκαν οι διάφορες υλοποιήσεις για WiMAX δίκτυα στον NS-2 και στην συνέχεια άρχισε η συγγραφή των σεναρίων για τα διάφορα modules. Έπειτα στα τελευταία βήματα κατατάσσονται η ανάλυση των αποτελεσμάτων των προσομοιώσεων καθώς και η συγγραφή της παρούσας εργασίας.

Συνοπτικά οπότε στην εργασία αυτή περιέχονται θεωρητικά κομμάτια τόσο για το WiMAX όσο και για τον NS-2. Στην πορεία γίνεται ανάλυση των modules που χρησιμοποιήθηκαν καθώς και των αποτελεσμάτων των σεναρίων που γράφτηκαν. Στο τέλος γίνεται η τελική σύγκριση μεταξύ των modules και στα παραρτήματα μπορεί να βρει κανείς τον κώδικα που χρησιμοποιήθηκε για κάθε προσομοίωση.

## **2 ΠΕΡΙΛΗΨΗ**

Μετά από μια σειρά εγκατάστασης και δοκιμής διαφόρων module για δίκτυα WiMAX στον προσομοιωτή δικτύων NS-2 (Network Simulator 2) πραγματοποιήθηκαν συγκρίσεις μεταξύ των modules οι οποίες είχαν σκοπό να αναδείξουν τα προτερήματα και μειονεκτήματα του κάθε module καθώς και να παρουσιάσουν το καλύτερο από αυτά.

Το σενάριο που δημιουργήθηκε αναπαριστά μια επικοινωνία μεταξύ ενός σταθμού βάσης και ενός σταθμού συνδρομητή στις περισσότερες περιπτώσεις ενώ υπάρχει και ένα σενάριο που αναπαριστά μια επικοινωνία Ad-hoc μεταξύ δύο σταθμών σε ένα mesh δίκτυο.

Στα διαγράμματα που βγαίνουν στο τέλος παρουσιάζονται σχεδόν όλα τα modules (λόγω τεχνικής αδυναμίας ένα module δεν παρουσιάζεται σε όλα τα διαγράμματα) και γίνονται συγκρίσεις όσον αφορά την αποτελεσματικότητα του καθενός ώστε να αναδειχθούν τα δυνατά και αδύνατα σημεία κάθε υλοποίησης.

### **3 ABSTRACT**

Through a series of different installations and testing of several WiMAX modules for the network simulator NS2 (Network Simulator 2), comparisons have been made between them which were intended to show the advantages and disadvantages of each module and present which one is the best.

The simulation scenario that has been used represents a communication between a base station and a mobile station in most occasions but there is also one scenario that represents an Ad-Hoc communication between two stations in a mesh network.

In the simulation charts almost all of the modules are represented (there has been a technical difficulty in one of the modules that didn't allow the representation of one of the modules in all of the charts) and comparisons are being made regarding the effectiveness of each one of them.

## 4 ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

<b>1 ΠΡΟΛΟΓΟΣ</b>	<b>2</b>
<b>2 ΠΕΡΙΛΗΨΗ</b>	<b>3</b>
<b>3 ABSTRACT</b>	<b>4</b>
<b>4 ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ</b>	<b>5</b>
<b>5 ΠΕΡΙΕΧΟΜΕΝΑ ΕΙΚΟΝΩΝ</b>	<b>11</b>
<b>6 ΠΕΡΙΕΧΟΜΕΝΑ ΠΙΝΑΚΩΝ</b>	<b>16</b>
<b>1 ΕΙΣΑΓΩΓΗ</b>	<b>17</b>
1.1 ΓΕΝΙΚΑ	17
1.2 ΑΝΤΙΚΕΙΜΕΝΟ ΚΑΙ ΣΚΟΠΟΣ ΤΗΣ ΠΑΡΟΥΣΗΣ ΕΡΓΑΣΙΑΣ	17
1.3 ΜΕΘΟΔΟΛΟΓΙΑ ΚΑΙ ΔΟΜΗ ΕΡΓΑΣΙΑΣ	18
<b>2 WIMAX</b>	<b>21</b>
2.1 ΕΙΣΑΓΩΓΗ	21
2.2 Το ΠΡΟΤΥΠΟ IEEE 802.16-2004	21
2.2.1 Φυσικό Επίπεδο	22
2.2.1.1 Επιπλέον Στοιχεία OFDM-OFDMA [3]	24
2.2.1.2 Δομή Συμβόλων και υπο-Κανάλια Προδιαγραφής OFDMA [3],[4]	26
2.2.1.3 Άλλα Χαρακτηριστικά [1],[3]	28
2.2.2 Υπο - Επίπεδο MAC	29
2.2.2.1 Γενικά [1]	29
2.2.2.1.1 Υπο-επίπεδο Σύγκλισης (Convergence sublayer (CS))	30
2.2.2.1.2 Common part sublayer (CPS)	31
2.2.2.1.3 Υπο-επίπεδο Ασφαλείας(Privacy sublayer)	31
2.2.2.2 Βασικές Λειτουργίες Υπο-Επιπέδου MAC [3],[4]	31
2.2.2.3 Ροή Υπηρεσιών (Service Flow) [1]	35
2.2.2.4 Υποστήριξη QoS (Quality of Service) στο 802.16	38
2.2.2.5 Αιτήσεις Σύνδεσης και Bandwidth	41
2.2.2.6 Προγραμματισμός Ποιότητας Επικοινωνίας (QoS) [1],[2]	42
2.3 IEEE 802.16E-2005	46
2.3.1 Γενικά	46
2.3.2 Φυσικό Επίπεδο	47
2.3.2.1 Λεπτομέρειες Μεταβλητού OFDMA(Scalable-OFDMA) [3]	47
2.3.2.2 Άλλοι Μηχανισμοί Φυσικού Επιπέδου	48
2.3.3 Υπο - Επίπεδο MAC	51

2.3.3.1 Υποστήριξη QoS (Quality of Service).....	51
2.3.3.2 Υποστήριξη Handover/Handoff [1].....	53
<b>3 Ο ΠΡΟΣΟΜΟΙΩΤΗΣ NS-2.....</b>	<b>55</b>
3.1 ΕΙΣΑΓΩΓΗ.....	55
3.2 Η ΧΡΗΣΙΜΟΤΗΤΑ ΔΥΟ ΔΙΑΦΟΡΕΤΙΚΩΝ ΓΛΩΣΣΩΝ [5],[7].....	55
3.3 ΕΠΙΣΚΟΠΗΣΗ ΤΟΥ ΠΡΟΣΟΜΟΙΩΤΗ [5],[7].....	56
<b>4 NIST MODULE.....</b>	<b>58</b>
4.1 ΕΙΣΑΓΩΓΗ.....	58
4.2 DOCUMENTATION [11].....	58
4.2.1 Υπόστρωμα σύγκλισης πακέτων.....	58
4.2.2 Υποστρώμα MAC.....	59
4.2.2.1 Δομή του προσθέτου MAC.....	59
4.2.3 Διευθυνσιοδότηση και σύνδεση.....	61
4.2.4 Μορφή των PDUs της MAC.....	62
4.2.5 Κατασκευή και μετάδοση των PDU του επιπέδου MAC.....	63
4.2.5.1 Κατακερματισμός.....	64
4.2.6 Υπηρεσίες Χρονοπρογραμματισμού.....	64
4.2.6.1 Χρονοπρογραμματιστές.....	64
4.2.7 Υποστήριξη του PHY στο επίπεδο MAC.....	65
4.2.8 Είσοδος σε δίκτυο και αρχικοποίηση.....	66
4.2.9 Τηλεμετρία.....	67
4.2.10 Διαδικασίες παράδοσης του επιπέδου MAC.....	68
4.2.10.1 Σάρωση.....	68
4.2.11 Η δομή του frame.....	70
4.2.12 Διαχείριση πακέτων.....	72
4.2.13 OFDM PHY.....	76
4.3 ΔΙΑΔΙΚΑΣΙΑ ΕΓΚΑΤΑΣΤΑΣΗΣ.....	77
4.4 ΠΡΟΣΟΜΟΙΩΣΗ.....	79
4.4.1 Σενάριο προσομοίωσης.....	79
4.4.2 Αποτελέσματα προσομοίωσης.....	80
4.4.3 Περαιτέρω διερεύνηση.....	82
<b>5 WIMAX FORUM MODULE.....</b>	<b>87</b>
5.1 ΕΙΣΑΓΩΓΗ.....	87
5.2 DOCUMENTATION [10].....	88
5.2.1 Σταθμός Βάσης.....	88

## Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

5.2.1.1 Δομή του σταθμού βάσης.....	88
5.2.1.1.1 Ταξινομητής ροής.....	88
5.2.1.1.2 Χρονοπρογραμματιστής (DL ARQ/HARQ).....	89
5.2.1.1.3 Uplink ARQ.....	89
5.2.1.1.4 Συμβολομεταφραστής πακέτων downlink .....	89
5.2.1.1.5 Αναλυτής πακέτων.....	89
5.2.1.1.6 Tx/Rx PHY.....	90
5.2.2 Κινητός σταθμός συνδρομητής.....	90
5.2.2.1 Δομή του κινητού σταθμού συνδρομητή.....	90
5.2.2.1.1 Χρονοπρογραμματιστής uplink (μονάδα ARQ/HARQ).....	90
5.2.2.1.2 Αναλυτής uplink.....	91
5.2.3 Ροές πακέτων στον προσομοιωτή.....	91
5.2.3.1 Κανονική ροή πακέτων δεδομένων downlink.....	91
5.2.3.2 Κανονική ροή πακέτων δεδομένων uplink.....	92
5.2.3.3 Ροή δεδομένων για τα μηνύματα αίτησης εύρους ζώνης.....	92
5.2.3.4 Ροές δεδομένων του HARQ.....	93
5.2.3.4.1 Ροή δεδομένων του uplink HARQ.....	93
5.2.3.4.2 Ροή δεδομένων του downlink HARQ.....	94
5.2.3.5 Ροές δεδομένων του ARQ.....	94
5.2.3.5.1 Ροή δεδομένων του uplink ARQ.....	94
5.2.3.5.2 Ροή δεδομένων του downlink ARQ.....	95
5.2.4 Υπόστρωμα σύγκλισης πακέτων.....	96
5.2.5 Κοινό τμήμα υποεπιπέδου MAC.....	96
5.2.5.1 Δομή του τμήματος MAC.....	96
5.2.6 Διευθυνσιοδότηση και σύνδεση.....	98
5.2.7 Μορφή των PDUs του MAC.....	99
5.2.8 Κατασκευή και μετάδοση των PDU του επιπέδου MAC.....	100
5.2.8.1 Κατακερματισμός.....	101
5.2.8.2 Πακετοποίηση.....	101
5.2.9 ARQ.....	101
5.2.9.1 ARQ με κατακερματισμό και πακετοποίηση.....	101
5.2.9.2 Λογική του πομπού και παραλήπτη ARQ.....	103
5.2.9.2.1 Λογική πομπού.....	103
5.2.9.2.2 Λογική παραλήπτη.....	106
5.2.9.2.3 Ανατροφοδότηση ARQ.....	109
5.2.9.2.4 Χρονισμός ARQ.....	109
5.2.9.3 Παράμετροι ARQ.....	109
5.2.9.4 Διάγραμμα κλάσεων του ARQ.....	109
5.2.10 Χρονοπρογραμματισμός υπηρεσιών.....	111
5.2.10.1 Χρονοπρογραμματιστές.....	112

5.2.10.2 Προεπιλεγμένος χρονοπρογραμματιστής.....	113
5.2.10.2.1 Χρονοπρογραμματιστής downlink στον σταθμό βάσης.....	114
5.2.10.2.2 Χρονοπρογραμματιστής uplink στον σταθμό βάσης.....	115
5.2.10.3 Οριζόντιο stripping.....	116
5.2.10.4 Κάθετο stripping.....	116
5.2.10.5 Χρονοπρογραμματιστής uplink στον σταθμό συνδρομητή.....	117
5.2.10.6 Αιτήσεις εύρους ζώνης και υπέρθεση (riggybacking).....	117
<b>5.2.11 Εκχώρηση εύρους ζώνης και μηχανισμοί αιτημάτων.....</b>	<b>117</b>
5.2.11.1 Τυχαία προσπέλαση βασισμένη στο CDMA.....	118
5.2.11.1.1 Δομή του frame του CDMA.....	118
5.2.11.1.2 Παράμετροι τυχαίας προσπέλασης βασισμένης στο CDMA.....	118
5.2.11.1.3 Πακέτο CDMA.....	119
5.2.11.1.4 Στάδια του CDMA.....	119
5.2.11.1.5 Χρονόμετρα CDMA.....	120
5.2.11.1.6 Αφαίρεση από την ουρά του CDMA.....	121
5.2.11.1.7 Επίλυση συγκρούσεων.....	121
5.2.12 Υποστήριξη του PHY στο επίπεδο MAC.....	122
5.2.13 Είσοδος σε δίκτυο και αρχικοποίηση.....	123
5.2.14 Τηλεμετρία.....	124
5.2.15 Πολιτική QoS.....	125
5.2.16 Διαδικασίες παράδοσης του επιπέδου MAC.....	126
5.2.16.1 Σάρωση.....	126
5.2.17 Η δομή του frame.....	128
5.2.18 Διαχείριση πακέτων.....	130
5.2.19 OFDMA PHY.....	134
5.2.19.1 Διάγραμμα κλάσεων του OFDMA του φυσικού επιπέδου.....	134
5.2.19.2 OFDMA PHY.....	136
5.2.19.2.1 Cost231.....	137
5.2.19.3 Fast/Frequency Selective Fading-Υπολογισμοί πριν την προσομοίωση.....	137
5.2.19.4 Μοντελοποίηση παρεμβολών (SIR).....	141
5.3 ΔΙΑΔΙΚΑΣΙΑ ΕΓΚΑΤΑΣΤΑΣΗΣ.....	142
5.4 ΠΡΟΣΟΜΟΙΩΣΗ.....	146
5.4.1 Σενάριο προσομοίωσης.....	146
5.4.2 Αποτελέσματα προσομοίωσης.....	146
5.4.3 Περαιτέρω διερεύνηση.....	148
<b>6 LRC MODULE.....</b>	<b>153</b>
6.1 ΕΙΣΑΓΩΓΗ.....	153
6.2 DOCUMENTATION.....	153



## Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

6.3 ΔΙΑΔΙΚΑΣΙΑ ΕΓΚΑΤΑΣΤΑΣΗΣ.....	154
6.4 ΠΡΟΣΟΜΟΙΩΣΗ.....	154
6.4.1 Σενάριο προσομοίωσης.....	154
6.4.2 Αποτελέσματα προσομοίωσης.....	155
6.4.3 Περαιτέρω διερεύνηση.....	156
<b>7 NS2MESH80216 MODULE.....</b>	<b>161</b>
7.1 ΕΙΣΑΓΩΓΗ.....	161
7.2 DOCUMENTATION.....	161
7.2.1 Δομή της υλοποίησης.....	161
7.2.2 Υποστηριζόμενες τοπολογίες [9].....	161
7.2.2.1 Τοπολογία αλυσίδας (chain).....	162
7.2.2.2 Τοπολογία δακτυλίου (ring).....	162
7.2.2.3 Τοπολογία πολλαπλού δακτυλίου (multiring).....	163
7.2.2.4 Τοπολογία κλίκας (clique).....	164
7.2.2.5 Τοπολογία πλέγματος (grid).....	165
7.2.2.6 Τοπολογία πλήρους πλέγματος (grid-full).....	166
7.2.2.7 Τοπολογία δυαδικού δέντρου (bintree).....	167
7.2.2.8 Τοπολογία αστέρα (star).....	168
7.2.2.9 Τοπολογία εξαγώνου (hexagon).....	169
7.2.2.10 Τριγωνική τοπολογία (triangular).....	170
7.2.3 Αντήληση πληροφοριών για την προσομοίωση.....	171
7.3 ΔΙΑΔΙΚΑΣΙΑ ΕΓΚΑΤΑΣΤΑΣΗΣ.....	176
7.3.1 Προβλήματα εγκατάστασης.....	179
7.4 ΠΡΟΣΟΜΟΙΩΣΗ.....	180
7.4.1 Σενάριο προσομοίωσης.....	180
7.4.2 Αποτελέσματα προσομοίωσης.....	180
<b>8 ΣΥΜΠΕΡΑΣΜΑΤΑ.....</b>	<b>182</b>
8.1 ΕΙΣΑΓΩΓΗ.....	182
8.2 ΣΥΓΚΡΙΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ.....	182
8.2.1 Περαιτέρω διερεύνηση.....	186
8.3 ΔΥΝΑΤΟΤΗΤΕΣ ΚΑΙ ΕΛΛΕΙΨΕΙΣ ΚΑΘΕ MODULE.....	188
8.4 ΣΥΜΠΕΡΑΣΜΑΤΑ.....	189
<b>9 ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>192</b>
<b>10 ΠΑΡΑΡΤΗΜΑ Α.....</b>	<b>194</b>
<b>11 ΠΑΡΑΡΤΗΜΑ Β.....</b>	<b>203</b>

<b>12 ΠΑΡΑΡΤΗΜΑ Γ</b>	<b>213</b>
-----------------------	------------

<b>13 ΠΑΡΑΡΤΗΜΑ Δ</b>	<b>223</b>
-----------------------	------------

## **5 ΠΕΡΙΕΧΟΜΕΝΑ ΕΙΚΟΝΩΝ**

Εικόνα 1: Εισαγωγή κυκλικού προθέματος.....	25
Εικόνα 2: Είδη υπο-μεταφορέων OFDMA.....	26
Εικόνα 3: Δομή tile για ζώνη συμβόλων UL-PUSC.....	27
Εικόνα 4: Τεχνική Time Division Duplex (TDD).....	33
Εικόνα 5: Δομή πλαισίου OFDM TDD.....	35
Εικόνα 6: Ροές υπηρεσιών WiMAX.....	36
Εικόνα 7: Dynamic service addition (DSA) - Ροή μηνυμάτων.....	38
Εικόνα 8: Προγραμματισμός καναλιών UL και DL σταθμού βάσης.....	44
Εικόνα 9: Διάγραμμα κλάσεων του MAC επιπέδου (NIST module).....	60
Εικόνα 10: Διάγραμμα κλάσεων της κλάσης "hdr_802_16" (NIST module).....	62
Εικόνα 11: Διάγραμμα κλάσεων χρονοδρομολόγησης (NIST module).....	65
Εικόνα 12: Χρονική σχέση του DL_MAP και του UL_MAP (NIST module).....	66
Εικόνα 13: Είσοδος σε δίκτυο (NIST module).....	67
Εικόνα 14: Διαδικασία σάρωσης (NIST module).....	69
Εικόνα 15: Διάγραμμα κλάσεων του frame (NIST module).....	71
Εικόνα 16: Ανασκόπηση διαχείρισης πακέτων (NIST module).....	72
Εικόνα 17: Διαχείριση εξερχομένων πακέτων (NIST module).....	73
Εικόνα 18: Διαχείριση νέων εισερχομένων πακέτων (NIST module).....	74
Εικόνα 19: Διαχείριση ληφθέντων πακέτων (NIST module).....	75
Εικόνα 20: Διάγραμμα κλάσεων του OFDM PHY (NIST module).....	76
Εικόνα 21: Throughput κατά την διάρκεια της προσομοίωσης (NIST).....	81
Εικόνα 22: Delay κατά την διάρκεια της προσομοίωσης (NIST module).....	82

Εικόνα 23: Μέσο delay για τις διαφορετικές προσομοιώσεις (NIST module).....	83
Εικόνα 24: Αναλυτικό delay για τις διαφορετικές προσομοιώσεις (NIST module) .	84
Εικόνα 25: Μέσο throughput για τις διαφορετικές προσομοιώσεις (NIST module)	85
Εικόνα 26: Αναλυτικό throughput για τις διαφορετικές προσομοιώσεις (NIST module).....	86
Εικόνα 27: Δομή του σταθμού βάσης (WiMAX Forum module).....	88
Εικόνα 28: Δομή του κινητού σταθμού συνδρομητή (WiMAX Forum module).....	90
Εικόνα 29: Κανονική ροή πακέτων δεδομένων downlink (WiMAX Forum module)	91
Εικόνα 30: Κανονική ροή δεδομένων πακέτων uplink (WiMAX Forum module)....	92
Εικόνα 31: Ροή πακέτων για CQICH (WiMAX Forum module).....	92
Εικόνα 32: Ροή δεδομένων του HARQ uplink (WiMAX Forum module).....	93
Εικόνα 33: Ροή δεδομένων του HARQ downlink (WiMAX Forum module).....	94
Εικόνα 34: Ροή δεδομένων του ARQ uplink (WiMAX Forum module).....	95
Εικόνα 35: Ροή δεδομένων του ARQ downlink (WiMAX Forum module).....	95
Εικόνα 36: Διάγραμμα κλάσεων του MAC επιπέδου (WiMAX Forum module).....	97
Εικόνα 37: Διάγραμμα κλάσεων της κλάσης "hdr_mac802_16" (WiMAX Forum module).....	99
Εικόνα 38: Παραδείγματα χρήσης ARQ PDUs με ή χωρίς αναδιάταξη (WiMAX Forum module).....	103
Εικόνα 39: Λογική πομπού ARQ (WiMAX Forum module).....	105
Εικόνα 40: Κατάσταση μηχανής πομπού (WiMAX Forum module).....	106
Εικόνα 41: Λογική παραλήπτη ARQ (WiMAX Forum module).....	107
Εικόνα 42: Διάγραμμα κλάσεων ARQ (WiMAX Forum module).....	110
Εικόνα 43: Διάγραμμα κλάσεων χρονοπρογραμματισμού (WiMAX Forum module).....	112

Εικόνα 44: Χρονοπρογραμματιστές στον σταθμό βάσης και στον σταθμό συνδρομητή (WiMAX Forum module).....	113
Εικόνα 45: Οριζόντιο stripping (WiMAX Forum module).....	116
Εικόνα 46: Κάθετο stripping (WiMAX Forum module).....	116
Εικόνα 47: Δομή του CDMA frame (WiMAX Forum module).....	118
Εικόνα 48: Βήματα αίτησης CDMA (WiMAX Forum module).....	120
Εικόνα 49: Υποδοχές σύγκρουσης και αιτήματα σύγκρουσης (WiMAX Forum module).....	122
Εικόνα 50: Χρονική σχέση του DL_MAP και του UL_MAP (WiMAX Forum module).....	123
Εικόνα 51: Είσοδος σε δίκτυο (WiMAX Forum module).....	124
Εικόνα 52: Ροή υπηρεσιών (WiMAX Forum module).....	125
Εικόνα 53: Διαδικασία σάρωσης (WiMAX Forum module).....	127
Εικόνα 54: Διάγραμμα κλάσεων του frame (WiMAX Forum module).....	129
Εικόνα 55: Δομή του TDD frame (WiMAX Forum module).....	129
Εικόνα 56: Επισκόπηση της διαχείρισης πακέτων (WiMAX Forum module).....	131
Εικόνα 57: Διαχείριση εξερχομένου πακέτου (WiMAX Forum module).....	132
Εικόνα 58: Διαχείριση νέου εισερχομένου πακέτου (WiMAX Forum module).....	133
Εικόνα 59: Διαχείριση ληφθέντως πακέτου (WiMAX Forum module).....	134
Εικόνα 60: Διάγραμμα κλάσεων του OFDMA PHY (WiMAX Forum module).....	135
Εικόνα 61: Propagation loss formula (WiMAX Forum module).....	137
Εικόνα 62: Μετατροπή από time domain σε frequency domain κανάλι (WiMAX Forum module).....	139
Εικόνα 63: Παράδειγμα αρχείου εκποίησης καναλιού του μοντέλου ITU (WiMAX Forum module).....	140

Εικόνα 64: Time domain correlation (WiMAX Forum module).....	141
Εικόνα 65: Απεικόνιση της παρεμβολής (WiMAX Forum module).....	142
Εικόνα 66: Throughput κατά την διάρκεια της προσομοίωσης (WiMAX Forum) ..	147
Εικόνα 67: Delay κατά την διάρκεια της προσομοίωσης (WiMAX Forum module) .....	148
Εικόνα 68: Μέσο delay για τις διαφορετικές προσομοιώσεις (WiMAX Forum module).....	149
Εικόνα 69: Αναλυτικό delay για τις διαφορετικές προσομοιώσεις (WiMAX Forum module).....	150
Εικόνα 70: Μέσο throughput για τις διαφορετικές προσομοιώσεις (WiMAX Forum module).....	151
Εικόνα 71: Αναλυτικό throughput για τις διαφορετικές προσομοιώσεις (WiMAX Forum module).....	152
Εικόνα 72: Throughput κατά την διάρκεια της προσομοίωσης (LRC).....	155
Εικόνα 73: Delay κατά την διάρκεια της προσομοίωσης (LRC module).....	156
Εικόνα 74: Μέσο delay για τις διαφορετικές προσομοιώσεις (LRC module).....	157
Εικόνα 75: Αναλυτικό delay για τις διαφορετικές προσομοιώσεις (LRC module).	158
Εικόνα 76: Μέσο throughput για τις διαφορετικές προσομοιώσεις (LRC module) .....	159
Εικόνα 77: Αναλυτικό throughput για τις διαφορετικές προσομοιώσεις (LRC module).....	160
Εικόνα 78: Τοπολογία αλυσίδας (ns2mesh80216 module).....	162
Εικόνα 79: Τοπολογία δακτυλίου (ns2mesh80216 module).....	163
Εικόνα 80: Τοπολογία πολλαπλού δακτυλίου (ns2mesh80216 module).....	164
Εικόνα 81: Τοπολογία κλίκας (ns2mesh80216 module).....	165

Εικόνα 82: Τοπολογία πλέγματος (ns2mesh80216 module).....	166
Εικόνα 83: Τοπολογία πλήρους πλέγματος (ns2mesh80216 module).....	167
Εικόνα 84: Τοπολογία δυαδικού δέντρου (ns2mesh80216 module).....	168
Εικόνα 85: Τοπολογία αστέρα (ns2mesh80216 module).....	169
Εικόνα 86: Τοπολογία εξαγώνου (ns2mesh80216 module).....	170
Εικόνα 87: Τριγωνική τοπολογία (ns2mesh80216 module).....	170
Εικόνα 88: Συγκριτικό delay καθ' όλη την προσομοίωση όλων των modules.....	183
Εικόνα 89: Συγκριτικό μέσο delay όλων των modules.....	184
Εικόνα 90: Συγκριτικό throughput κατά την διάρκεια της προσομοίωσης όλων των modules.....	185
Εικόνα 91: Συγκριτικό μέσο throughput όλων των modules.....	186
Εικόνα 92: Μέσο delay όλων των modules για προσομοιώσεις με διαφορετικό πλήθος σταθμών.....	187
Εικόνα 93: Μέσο throughput όλων των modules για προσομοιώσεις με διαφορετικό πλήθος σταθμών.....	188

## **6 ΠΕΡΙΕΧΟΜΕΝΑ ΠΙΝΑΚΩΝ**

Πίνακας 1: Δομή υπο-επιπέδων.....	30
Πίνακας 2: Παράμετροι S-OFDMA.....	48
Πίνακας 3: Διαθέσιμες διαμορφώσεις και κωδικοποιήσεις.....	49
Πίνακας 4: Επιπλέον στοιχεία διαμορφώσεων και κωδικοποιήσεων.....	49
Πίνακας 5: Προδιαγραφές υπηρεσιών QoS και χαρακτηριστικά αυτών.....	53
Πίνακας 6: Δυνατότητες και ελλείψεις του NIST module.....	58
Πίνακας 7: Υποστηριζόμενοι τύποι πακέτων (NIST module).....	63
Πίνακας 8: Δυνατότητες και ελλείψεις του WiMAX Forum module.....	87
Πίνακας 9: Υποστηριζόμενοι τύποι πακέτων (WiMAX Forum module).....	100
Πίνακας 10: Παράδειγμα του μοντέλου ITU (WiMAX Forum module).....	138
Πίνακας 11: Συγκριτικός πίνακας δυνατοτήτων των modules.....	188



# **1 Εισαγωγή**

## **1.1 Γενικά**

Ο άνθρωπος έχει έμφυτη την ανάγκη για επικοινωνία, ανταλλαγή σκέψεων και διαφόρων εμπειριών. Τα τελευταία χρόνια την ανάγκη αυτή υπάρχει η τάση να την ικανοποιούν τα δίκτυα υπολογιστών και συγκεκριμένα τα ασύρματα δίκτυα υπολογιστών, λόγω της ευελιξίας που προσφέρουν. Με την πρόοδο της τεχνολογίας πλέον υπάρχει η δυνατότητα για παροχή ακόμη και της πιο απαιτητικής εφαρμογής στον χρήστη μέσω ασυρμάτων δικτύων με παράλληλη επίτευξη απόλυτα ικανοποιητικής ποιότητας. Ο μεγαλύτερος περιορισμός των ασυρμάτων δικτύων στο παρελθόν, που ήταν οι περιορισμένοι πόροι, υπερσκελίζεται με σταθερά βήματα με εκδόσεις καινούριων προτύπων και τεχνολογιών. Το πρότυπο 802.16 ή WiMAX όπως είναι ευρέως γνωστό μπορεί να ικανοποιήσει τις αυξημένες ανάγκες των χρηστών και να παρέχει κάλυψη μεγάλων αποστάσεων με ικανοποιητικές ταχύτητες υπηρεσιών. Επίσης διαθέτει τους απαραίτητους μηχανισμούς ώστε να εξασφαλίζεται η ποιότητα της επικοινωνίας σε όλες της τις εκφάνσεις κάτι που είναι ιδιαίτερα σημαντικό για εφαρμογές φωνής και βίντεο που είναι πλέον τόσο διαδεδομένες στην εποχή μας.

## **1.2 Αντικείμενο και Σκοπός της Παρούσης Εργασίας**

Η παρούσα πτυχιακή εργασία πραγματοποιήθηκε ως αναπόσπαστο κομμάτι των σπουδών στα πλαίσια του προπτυχιακού προγράμματος μαθημάτων του τμήματος Πληροφορικής του ΑΤΕΙ Θεσσαλονίκης κατά το ακαδημαϊκό έτος 2010-2011. Η εργασία έχει τίτλο “Προσομοιωτική μελέτη ασύρματων ευρυζωνικών δικτύων (IEEE 802.16 – WiMAX) μέσω του εργαλείου NS-2” και αποτελεί μια προσπάθεια σύγκρισης των διάφορων υλοποιήσεων που έχουν γίνει για την υποστήριξη του WiMAX στον προσομοιωτή Network Simulator 2 (NS-2).

Ο βασικός σκοπός της παρούσης εργασίας είναι η υλοποίηση σεναρίων που θα χρησιμοποιηθούν για την εξαγωγή αποτελεσμάτων που θα βοηθήσουν στην

σύγκριση των διαφόρων υλοποιήσεων που έχουν γίνει για την υποστήριξη του WiMAX στον προσομοιωτή NS-2.

### **1.3 Μεθοδολογία και Δομή Εργασίας**

Για το πειραματικό κομμάτι της εργασίας χρησιμοποιήθηκε ένας από τους πλέον διαδεδομένους στην επιστημονική κοινότητα προσομοιωτές δικτύων ο Network Simulator 2 (NS2) σε διάφορες εκδόσεις του αναλόγως με τις ανάγκες του εκάστοτε module που χρησιμοποιήθηκε. Επειδή η εγκατάσταση των modules γίνεται ταυτόχρονα με την εγκατάσταση του προγράμματος επηρεάζοντας τον πηγαίο κώδικα του, ήταν αναγκαία η δημιουργία πολλαπλών εγκαταστάσεων του προγράμματος. Κάθε εγκατάσταση έγινε σε ξεχωριστό εικονικό μηχάνημα για διευκόλυνση καθώς και για αποφυγή διενέξεων μεταξύ των πολλαπλών εγκαταστάσεων του ίδιου προγράμματος σε ένα μηχάνημα. Τα εικονικά μηχανήματα που δημιουργήθηκαν είχαν όλα τα ίδια χαρακτηριστικά και είχαν σαν λειτουργικό σύστημα την διανομή Fedora Linux 13.

Τα σενάρια που χρησιμοποιήθηκαν για την σύγκριση των modules γράφτηκαν σε TCL και έγινε προσπάθεια να είναι όσο το δυνατόν πιο ισότιμα γίνεται μεταξύ τους για να μπορέσει να γίνει μια σωστή σύγκριση. Αυτό βέβαια δεν ήταν απολύτως εφικτό λόγω της διαφορετικότητας που παρουσίαζαν τα modules μεταξύ τους στην δομή καθώς και στην υποστήριξη διαφόρων λειτουργιών του WiMAX. Έτσι η προσέγγιση που έγινε ήταν σχετικά απλή για να μπορέσει να λειτουργήσει ένα όσο το δυνατόν πιο ίδιο σενάριο σε όλα τα modules.

Στην συνέχεια με κατάλληλα script γραμμένα σε awk έγινε εξόρυξη πληροφοριών από τα αρχεία αποτελεσμάτων που έβγαζε σαν έξοδο το τρέξιμο του σεναρίου ώστε να έρθουν οι πληροφορίες σε μια μορφή κατάλληλη ώστε να μπορέσει να αναπαρασταθεί γραφικά με διαγράμματα. Η δημιουργία των διαγραμμάτων έγινε με το πρόγραμμα gnuplot 4.4 παίρνοντας σαν είσοδο τα αρχεία ανάλυσης που έβγαιναν από τα awk scripts.

Στο γραπτό μέρος της εργασίας περιγράφονται τα διάφορα modules και οι δυνατότητές τους και παρουσιάζονται τα αποτελέσματα των προσομοιώσεων καθώς και τα πορίσματα που βγήκαν από αυτές.

Η εργασία δομείται στα παρακάτω κεφάλαια:

- ◆ 1<sup>ο</sup> κεφάλαιο: Εισαγωγή,  
στο κεφάλαιο αυτό παρουσιάζεται ο στόχος, η δομή και η μεθοδολογία που ακολουθήθηκε στην εργασία αυτή.
- ◆ 2<sup>ο</sup> κεφάλαιο: WiMAX,  
στο κεφάλαιο αυτό γίνεται μια σύντομη θεωρητική αναφορά στο WiMAX, τις δυνατότητές καθώς και την δομή του.
- ◆ 3<sup>ο</sup> κεφάλαιο: Ο προσομοιωτής NS-2,  
στο κεφάλαιο αυτό γίνεται μια σύντομη θεωρητική αναφορά στον NS-2 και στον τρόπο λειτουργίας του.
- ◆ 4<sup>ο</sup> κεφάλαιο: NIST module,  
στο κεφάλαιο αυτό γίνεται μια παρουσίαση του module που έχει αναπτυχθεί από την ομάδα του NIST καθώς και μια πρώτη αξιολόγηση αυτού.
- ◆ 5<sup>ο</sup> κεφάλαιο: Wimax Forum module,  
στο κεφάλαιο αυτό γίνεται μια παρουσίαση του module που έχει αναπτυχθεί από την ομάδα του WiMAX Forum καθώς και μια πρώτη αξιολόγηση αυτού.
- ◆ 6<sup>ο</sup> κεφάλαιο: LRC module,  
στο κεφάλαιο αυτό γίνεται μια παρουσίαση του module που έχει αναπτυχθεί από την ομάδα του LRC καθώς και μια πρώτη αξιολόγηση αυτού.
- ◆ 7<sup>ο</sup> κεφάλαιο: ns2mesh80216 module,  
στο κεφάλαιο αυτό γίνεται μια παρουσίαση του ns2mesh80216 module καθώς και μια πρώτη αξιολόγηση αυτού.
- ◆ 8<sup>ο</sup> κεφάλαιο: Συμπεράσματα,  
στο κεφάλαιο αυτό παρουσιάζεται αναλυτικά η σύγκριση των

προσομοιώσεων των διαφόρων σεναρίων και αναγράφονται τα πορίσματα που βγαίνουν από αυτά.

◆ Βιβλιογραφία,

όπου γίνεται παράθεση όλων των πηγών (άρθρα, βιβλία, διαδικτυακοί τόποι, κ.ά.) που χρησιμοποιήθηκαν στη παρούσα πτυχιακή τόσο για την θεωρητική προσέγγιση του θέματος, όσο και για την υλοποίηση της πειραματικής διαδικασίας.

## **2 WiMAX**

### **2.1 Εισαγωγή**

Σε αυτό το κεφάλαιο παρουσιάζεται μια περιγραφή και ανάλυση του προτύπου IEEE 802.16, το οποίο αποτελεί και το θεωρητικό υπόβαθρο της εργασίας αυτής. Συγκεκριμένα γίνεται αναφορά και ανάλυση των απαραίτητων στοιχείων για την υλοποίηση και κατανόηση του πειραματικού μέρους της εργασίας στα επόμενα κεφάλαια.

### **2.2 Το πρότυπο IEEE 802.16-2004**

Το πρώτο πρότυπο του 802.16 προσδιόρισε την διάδοση δεδομένων τύπου LOS (Line-Of-Sight) στη ζώνη συχνοτήτων 10-66GHz. Το 802.16a επέκτεινε τη λειτουργία του προτύπου ώστε να περιλάβει τη NLOS (Non-Line-Of-Sight) διάδοση στη ζώνη χαμηλής συχνότητας των 2-11 Ghz. Έτσι το πρότυπο IEEE 802.16-2004 (επιτροπή LAN/MAN, 2004) υποστηρίζει την επικοινωνία στην ζώνη συχνοτήτων 2-66GHz. Παρόλα αυτά η LOS και NLOS διάδοση-μετάδοση είναι αρκετά διαφορετικές.

Κατά συνέπεια, για να σχεδιαστεί ένα πρότυπο που να υποστηρίζει και τις δύο ζώνες συχνοτήτων, το φυσικό (PHY) και το MAC (Medium Access Control) επίπεδο πρέπει να υποστηρίζουν αυτές τις διαφορές.

Για παράδειγμα, παρά το γεγονός ότι η διάδοση σημάτων σε υψηλής ζώνης συχνότητα επηρεάζεται ιδιαίτερα από εμπόδια στον χώρο, χρησιμοποιείται η μετάδοση LOS, καθώς μειώνει την επίδραση του φαινομένου multipath interference. Multipath interference ονομάζεται το φαινόμενο κατά το οποίο η λήψη του σήματος από τον δέκτη γίνεται μέσω περισσότερων της μιας καθορισμένης πορείας, εξ αιτίας της αντανάκλασης και της διάθλασης στα εμπόδια που συναντά το σήμα κατά την πορεία του από τον πομπό στον δέκτη. Επίσης αυτό συμβαίνει καθώς η χαμηλότερη ζώνη συχνοτήτων περιλαμβάνει φάσμα εξουσιοδοτημένο και μη, το οποίο φυσικά υπακούει σε κάποιους κανονισμούς, οι οποίοι απαιτούν διαχείριση της ισχύος αποστολής σημάτων του πομπού, τεχνικές για να

αποφευχθεί η παρεμβολή συχνότητας κλπ. Αυτά και άλλα ζητήματα, έχουν επιπτώσεις όχι μόνο στην σχεδίαση του φυσικού επιπέδου αλλά και του επιπέδου MAC.

Κατά συνέπεια, το πεδίο εφαρμογής του προτύπου 802.16-2004 καλύπτει τις προδιαγραφές των δύο αυτών επιπέδων του προτύπου του OSI.

### 2.2.1 Φυσικό Επίπεδο

Ένα κύμα συχνότητας 10-66GHz είναι μια εστιασμένη ακτίνα, η οποία μπορεί θεωρητικά να καλύψει πολλά χιλιόμετρα μέσω της διάδοσης LOS. Οι σχεδιαστές έκριναν ότι η διαμόρφωση ενός ενιαίου φορέα (Single-Carrier) ήταν μια ικανοποιητική επιλογή και η έκδοση του προτύπου του φυσικού επιπέδου αυτής της ζώνης καλείται wirelessMAN-SC (Single Carrier). [1]

Το wirelessman-SC μπορεί να υποστηρίξει τεχνικές μετάδοσης σήματος FDD (Frequency Division Duplex) και TDD (Time Division Duplex). Όμως, η λειτουργία στη ζώνη των 2-11GHz απαιτήσε αλλαγές στις προδιαγραφές του φυσικού επιπέδου ώστε να υποστηριχθεί η διάδοση NLOS. Κυρίως, τρεις νέες προδιαγραφές/τεχνικές μετάδοσης του φυσικού επιπέδου(PHY) εισήχθησαν ώστε να ικανοποιηθούν αυτές τις απαιτήσεις: [3]

<sup>35</sup><sub>17</sub> single-carrier PHY,

<sup>35</sup><sub>17</sub> 256-point FFT OFDM(Orthogonal frequency division Multiplexing) PHY,

<sup>35</sup><sub>17</sub> 2048-point FFT OFDMA(Orthogonal frequency division Multiple Access) PHY

Η προδιαγραφή single-carrier PHY ορίζεται ως WirelessMAN-SCA και βασίζεται βεβαίως στο WirelessMAN-SC. Εντούτοις, υπάρχουν κάποιες διαφορές στην διαμόρφωση στοιχείων που επιτρέπουν την βελτίωση της εξισορρόπησης και επίδοσης του καναλιού για μετάδοση NLOS, στις ρυθμίσεις παραμέτρων και διευκόλυνση στις υλοποιήσεις προαιρετικών προσαρμοστικών συστημάτων κεραιών (AAS-Adaptive Antenna Systems).

Η δεύτερη και τρίτη προδιαγραφή σχετίζονται με την Ορθογωνική Πολύπλεξη με Διάρθρωση Συχνότητας (OFDM), η οποία είναι μία πολυκαναλική τεχνική μετάδοσης κατάλληλη για υψηλής ταχύτητας NLOS μετάδοση.

Η Ορθογωνική Πολύπλεξη με Διάρθρωση Συχνότητας (OFDM), χρησιμοποιεί 256 “υπο-μεταφορείς” ραδιοσυχνοτήτων (256-point FFT OFDM PHY) οι οποίοι μεταδίδουν διαφορετικά σήματα ταυτόχρονα.

Οι γειτονικοί “υπο-μεταφορείς” επιτρέπεται να επικαλύπτονται ωστόσο πρέπει να είναι ορθογωνικοί μεταξύ τους ώστε να μην παρεμβάλλεται το περιεχόμενο του ενός στον άλλον (ICI-Inter Carrier Interference).

Η βασική διαφορά μεταξύ των τεχνικών μετάδοσης WirelessMAN SCx και OFDM έγκειται στο ότι η τεχνική OFDM είναι περισσότερο ανθεκτική στο φαινόμενο του multipath interference.

Επίσης η OFDM επιτρέπει μεγαλύτερη αποδοτικότητα σε εύρος ζώνης (Bandwidth), δεδομένου ότι υποστηρίζει την επικάλυψη γειτονικών υπο-μεταφορέων. Έτσι, η τεχνική OFDM επιτρέπει ταχύτητα μετάδοσης δεδομένων της τάξης των 72 Mbps πάνω σε κανάλι εύρος ζώνης των 20 MHz, το οποίο σημαίνει φασματική απόδοση της τάξης των 3,6 bps / Hz. [1],[3]

Η Ορθογωνική Πολύπλεξη με Διάρθρωση Συχνότητας πολλαπλής πρόσβασης (OFDMA) με την σειρά της χρησιμοποιεί 2048 “υπο-μεταφορείς”. Η βασική διαφορά μεταξύ της OFDM και της OFDMA είναι ότι η OFDMA οργανώνει τους διαθέσιμους πόρους χρόνου (δηλαδή τα σύμβολα) και συχνότητας (δηλαδή τους “υπο-μεταφορείς”) σε υπο-κανάλια για κατανομή σε επιμέρους δέκτες. Αυτό ακριβώς το γεγονός επιτρέπει την πολλαπλή πρόσβαση. Η Ορθογωνική Πολύπλεξη με Διάρθρωση Συχνότητας πολλαπλής πρόσβασης (OFDMA) επιτρέπει δημιουργία υπο-καναλιών και στο κανάλι DownLink και στο κανάλι UpLink. Αντίθετα η Ορθογωνική Πολύπλεξη με Διάρθρωση Συχνότητας (OFDM) επιτρέπει μια περιορισμένη μορφή υπο-καναλιών μόνο στο κανάλι UpLink.

Η προδιαγραφή OFDM PHY είναι κοινή μεταξύ του 802.16 και του ETSI HiperMAN (το WiMax εναρμονισμένο στην Ευρωπαϊκά δεδομένα), διότι η OFDM απαιτεί πιο

αδύναμη συχνότητα συγχρονισμού και ταχύτερους υπολογισμούς στους μετασχηματισμούς Fourier (Fast Fourier Transform-FFT). [1]

Υπάρχει βέβαια και η μορφή σηματοδότησης CDMA (code division multiple access) , όμως αυτή χρειάζεται bandwidth που είναι πολύ μεγαλύτερο από την ταχύτητα μεταγωγής δεδομένων (data throughput) για τη διατήρηση επεξεργαστικής ισχύος ικανής να ξεπεράσει τις όποιες παρεμβολές. Επιπλέον, οι προδιαγραφές OFDM και OFDMA υποστηρίζουν μετάδοση NLOS επιτυγχάνοντας μέγιστη χρήση του φάσματος που διατίθενται.

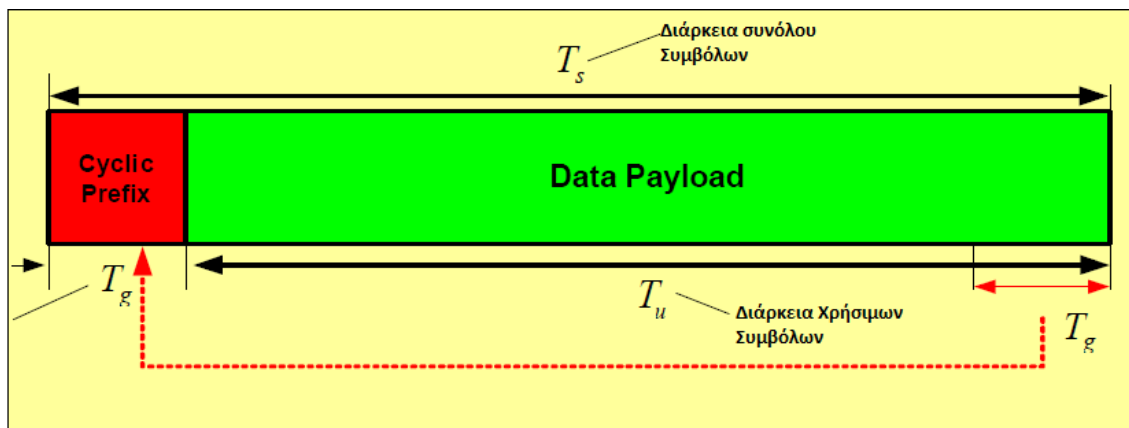
### **2.2.1.1 Επιπλέον Στοιχεία OFDM-OFDMA [3]**

Πέρα από την γενική περιγραφή των προδιαγραφών OFDM και OFDMA παραπάνω και την σύγκριση των βασικών τους σημείων, κρίνεται σκόπιμο να αναφερθεί με μεγαλύτερη λεπτομέρεια η λειτουργία τους καθώς αφορούν βασικά συστατικά του προτύπου IEEE 802.16.

Η Ορθογωνική Πολύπλεξη με Διάρθρωση Συχνότητας όπως αναφέρθηκε και παραπάνω είναι μία πολυκαναλική τεχνική μετάδοσης, η οποία υποδιαιρεί το διαθέσιμο bandwidth σε πολλαπλούς υπο-μεταφορείς συχνότητας. Σε ένα σύστημα OFDM η αρχική ροή σήματος χωρίζεται σε αρκετές παράλληλες υπο-ροές μικρότερου ρυθμού μετάδοσης δεδομένων(κάτι που αυτόματα σημαίνει αύξηση στην διάρκεια ενός συμβόλου) και η κάθε τέτοια ροή διαμορφώνεται κατάλληλα και μεταδίδεται από έναν ξεχωριστό ορθογωνικό υπο-μεταφορέα. Η αυξημένη διάρκεια κάθε συμβόλου βοηθάει και στον περιορισμό της απόκλισης των διαφορετικών καθυστερήσεων που μπορεί να υπάρχουν.

Επιπλέον η παρουσία του περιοδικού προθέματος (cyclic prefix (CP)) μπορεί να εξαλείψει ολοκληρωτικά τις δια-συμβολικές παρεμβολές (Inter-Symbol Interference (ISI)), αρκεί η διάρκεια του περιοδικού προθέματος (CP) να είναι μεγαλύτερη από την απόκλιση των καθυστερήσεων του καναλιού. Το περιοδικό πρόθεμα είναι στην ουσία μια επανάληψη των τελευταίων στοιχείων του τμήματος δεδομένων ενός block πληροφορίας, τα οποία επισυνάπτονται στο ξεκίνημα του όγκου των δεδομένων όπως φαίνεται στην εικόνα 1.





Εικόνα 1: Εισαγωγή κυκλικού προθέματος

Επίσης το κυκλικό πρόθεμα εμποδίζει την παρεμβολή ανάμεσα στα block πληροφορίας.

Ένα αρνητικό στοιχείο του είναι ότι εισάγει επιπλέον πληροφορία (overhead), η οποία επηρεάζει την επάρκεια σε bandwidth. Λόγω όμως του γεγονότος ότι η OFDM χρησιμοποιεί ένα ιδιαίτερα διαπεραστικό φάσμα, ένα μεγάλο κομμάτι του bandwidth του καναλιού μπορεί να χρησιμοποιηθεί για μετάδοση δεδομένων, κάτι το οποίο αντισταθμίζει το αρνητικό κομμάτι της χρήσης του κυκλικού προθέματος.

Η προδιαγραφή OFDM αξιοποιεί την ποικιλομορφία των συχνοτήτων του καναλιού και διαμοιράζει την πληροφορία στους υπο-μεταφορείς πριν την μετάδοση. Η διαμόρφωση στην OFDM μπορεί να υλοποιηθεί χρησιμοποιώντας μετασχηματισμούς Fourier και συγκεκριμένα Inverse Fast Fourier Transform (IFFT), οι οποίοι επιτρέπουν έναν μεγάλο αριθμό υπο-μεταφορέων (μέχρι 2048) χαμηλής πολυπλοκότητας. Σε ένα OFDM σύστημα, οι πόροι είναι διαθέσιμοι, στην διάσταση του χρόνου με την μορφή OFDM συμβόλων και στην διάσταση της συχνότητας με την μορφή υπο-μεταφορέων. Οι πόροι των δύο διαστάσεων (χρόνου και συχνότητας) μπορούν να οργανωθούν σε υπο-κανάλια τα οποία διαμοιράζονται σε ξεχωριστούς χρήστες του συστήματος.

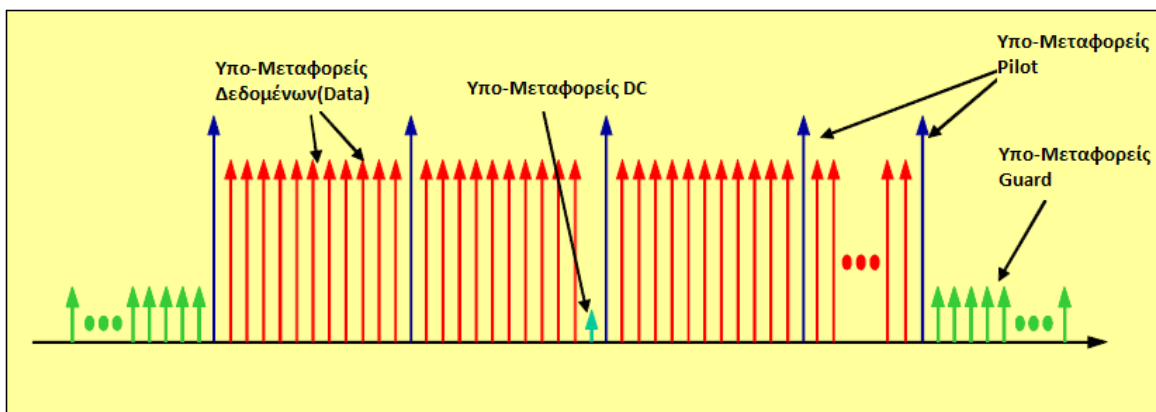
Η Ορθογωνική Πολύπλεξη με Διάρθρωση Συχνότητας πολλαπλής πρόσβασης (OFDMA) είναι όπως φαίνεται και από το όνομά της μια προδιαγραφή που επιτρέπει πολλαπλή προώθηση πολλαπλών ροών δεδομένων από πολλούς

ξεχωριστούς χρήστες στα υπο-κανάλια του DownLink καναλιού και πολλαπλή πρόσβαση σε αυτά μέσω των υπο-καναλιών του καναλιού UpLink.

### 2.2.1.2 Δομή Συμβόλων και υπο-Κανάλια Προδιαγραφής OFDMA [3],[4]

Η δομή των συμβόλων OFDMA αποτελείται από τέσσερις τύπους υπο-μεταφορέων:

- <sup>35</sup><sub>17</sub> Υπο-μεταφορείς δεδομένων (data), οι οποίοι μεταφέρουν τα δεδομένα της πληροφορίας.
- <sup>35</sup><sub>17</sub> Υπο-μεταφορείς DC, οι οποίοι υποδεικνύουν την κεντρική συχνότητα.
- <sup>35</sup><sub>17</sub> Υπο-μεταφορείς pilot, οι οποίοι χρησιμεύουν για διάφορες εκτιμήσεις και υπολογισμούς.
- <sup>35</sup><sub>17</sub> Υπο-μεταφορείς guard, οι οποίοι χρησιμεύουν στο να ξεχωρίζουν και να διατηρούν την απόσταση ανάμεσα στα OFDM και OFDMA σήματα.



Εικόνα 2: Είδη υπο-μεταφορέων OFDMA

Οι θεωρούμενοι ενεργοί υπο-μεταφορείς, υπο-μεταφορείς δεδομένων και pilot, ομαδοποιούνται σε υπο-ομάδες υπο-μεταφορέων που ονομάζονται υπο-κανάλια (sub-channels) .

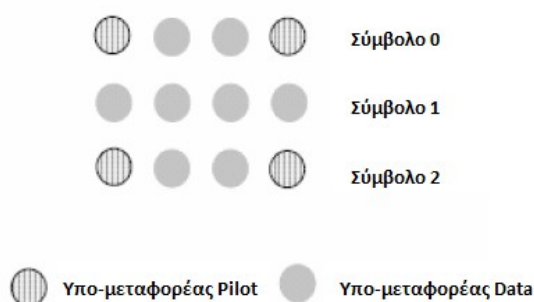
Το φυσικό επίπεδο του WiMAX OFDMA υποστηρίζει δημιουργία υπο-καναλιών και στο κανάλι DownLink και στο κανάλι UpLink. Η μικρότερη μονάδα συχνότητας-χρόνου που επιτρέπεται σε ένα υπο-κανάλι είναι 48 data tones που είναι ίσοι με 48 υπο-μεταφορείς και με ένα slot. Υπάρχουν δύο παραλλαγές δημιουργίας υπο-

καναλιών με βάση τους υπο-μεταφορείς: Η πρώτη ονομάζεται ανομοιόμορφη (diversity) και η δεύτερη συνεχόμενη (contiguous).

Στην πρώτη παραλλαγή επιλέγονται υπο-μεταφορείς με μια ψευδο-τυχαία διαδικασία ώστε να σχηματίσουν ένα υπο-κανάλι. Σαν αποτέλεσμα προσφέρει ποικιλομορφία συχνοτήτων. Η παραλλαγή αυτή περιλαμβάνει και κάποιες ζώνες, οι οποίες στην ουσία είναι μια σειρά συνεχόμενων συμβόλων που σχηματίζουν έναν συγκεκριμένο τύπο υπο-καναλιού η καθεμία. Αυτές οι ζώνες συμβόλων είναι για το κανάλι DownLink η DL FUSC (Fully Used Sub-Carrier) και η DL PUSC (Partially Used Sub-Carrier) και για το κανάλι UpLink η UL PUSC. Υπάρχουν και άλλες οι οποίες όμως είναι προαιρετικές. Η ζώνη συμβόλων DL PUSC προβλέπει ότι για κάθε ζευγάρι συμβόλων τύπου OFDM, οι διαθέσιμοι ή οι χρησιμοποιήσιμοι υπο-μεταφορείς ομαδοποιούνται ώστε να σχηματίσουν ένα cluster το οποίο περιέχει δέκα τέσσερις συνεχόμενους υπο-μεταφορείς ανά περίοδο συμβόλου.

Στην συνέχεια δημιουργούνται ομάδες τέτοιων clusters. Ένα υπο-κανάλι περιέχει δύο τέτοια clusters και περιέχει σαράντα οχτώ υπο-μεταφορείς δεδομένων (data sub-carriers) και οχτώ υπο-μεταφορείς pilot. Οι υπο-μεταφορείς δεδομένων της κάθε ομάδας στην συνέχεια συνδυάζονται καταλλήλως ώστε να δημιουργήσουν υπο- κανάλια μέσα σε αυτή. Οι υπο-μεταφορείς δεδομένων μέσα στο cluster διανέμονται σε πολλαπλά υπο-κανάλια.

Για την ζώνη συμβόλων UL-PUSC αντί για ένα cluster σχηματίζεται μια δομή που ονομάζεται tile. Η μορφή της φαίνεται στην εικόνα 3.



Εικόνα 3: Δομή tile για ζώνη συμβόλων UL-PUSC

Οι διαθέσιμοι υπο-μεταφορείς διαμοιράζονται στα διάφορα tiles. Έξι tiles ομαδοποιούνται και σχηματίζουν ένα slot. Ένα slot εμπεριέχει σαράντα οχτώ υπο-μεταφορείς δεδομένων και είκοσι τέσσερις υπο-μεταφορείς pilot ανά τρία σύμβολα OFDM.

Η δεύτερη παραλλαγή δημιουργίας υπο-καναλιών (συνεχόμενη (contiguous)) ομαδοποιεί έναν αριθμό συνεχόμενων υπο-μεταφορέων ώστε να δημιουργήσουν ένα υπο-κανάλι.

Περιλαμβάνει τις ζώνες συμβόλων DL AMC και UL AMC οι οποίες έχουν την ίδια δομή. Εδώ στην θέση του tile υπάρχει η οντότητα bin η οποία αποτελείται από εννιά συνεχόμενους υπο-μεταφορείς ανά σύμβολο, από τους οποίους οι οχτώ είναι υπο-μεταφορείς δεδομένων και το τελευταίος είναι υπο-μεταφορέας pilot. Ένα slot τύπου AMC είναι μια συλλογή bins για τα οποία πρέπει να ισχύει η σχέση ( $N \times M = 6$ ), όπου  $N$  είναι ο αριθμός των συνεχόμενων bins και  $M$  είναι ο αριθμός των συνεχόμενων συμβόλων. Συνεπώς οι επιτρεπόμενοι συνδυασμοί είναι:

$\frac{35}{17}$  6 bins, 1 σύμβολο

$\frac{35}{17}$  1 bin, 6 σύμβολα

$\frac{35}{17}$  3 bins, 2 σύμβολα

$\frac{35}{17}$  2 bins, 3 σύμβολα

Η ζώνη AMC καταφέρνει την πολλαπλή εξυπηρέτηση χρηστών έχοντας την δυνατότητα να επιλέγει το καλύτερο υπο-κανάλι με την καλύτερη απόδοση συχνότητας και να το προσφέρει για χρήση.

Γενικά, η ανομοιόμορφη (diversity) παραλλαγή δημιουργίας υπο-καναλιών αποδίδει καλά σε περιβάλλοντα έντονης και μόνιμης κινητικότητας σταθμών, ενώ η συνεχόμενη (contiguous) είναι καταλληλότερη για περιβάλλοντα με ακίνητους ή κινούμενους σταθμούς με ελάχιστη ταχύτητα και μετατόπιση.

### 2.2.1.3 Άλλα Χαρακτηριστικά [1],[3]

Το φυσικό επίπεδο έχει επίσης άλλα χαρακτηριστικά, ορισμένα από αυτά είναι υποχρεωτικά και άλλα είναι προαιρετικά. Αυτά τα χαρακτηριστικά δίνουν την

δυνατότητα σταθερών και ισχυρών επιδόσεων σε ένα ευρύ φάσμα συχνοτήτων, και σε πολλές διαφορετικές συνθήκες καναλιών μετάδοσης.

**Adaptive Antenna System (AAS):** χρησιμοποιεί πολλαπλές κεραιές τόσο στην μεριά του δέκτη όσο και στην μεριά του πομπού (σύστημα MIMO) με σκοπό την αύξηση της χωρητικότητας του καναλιού στρέφοντας τις δέσμες της κεραιάς προς πολλαπλούς χρήστες για την επίτευξη επαναχρησιμοποίησης συχνοτήτων. Το σύστημα MIMO ( Multiple-input multiple-output) είναι επίσης ευεργετικό για την βελτίωση της σχέσης σήματος-παρεμβολών (signal-to-interference ratio) ανάμεσα σε γειτονικά συνδυαζόμενα σήματα.

**Adaptive modulation (AMC):** Το πρότυπο 802.16-2004 επιτρέπει διαφορετικούς τύπους διαμόρφωσης (modulation) κατά την επικοινωνία UL (UpLink) και DL (DownLink). Οι επιτρεπόμενοι τύποι διαμόρφωσης είναι BPSK, QPSK, 16QAM, 64QAM και 256QAM. Το πρότυπο 802.16 ορίζει διαφορετικούς συνδυασμούς των προαναφερόμενων τύπων διαμόρφωσης και κωδικοποίησης, παρέχοντας ένα ευρύ φάσμα επιλογών ανάμεσα στην επιδίωξη για υψηλές ταχύτητες μεταφοράς δεδομένων και αξιοπιστίας, ανάλογα πάντα με τις συνθήκες του καναλιού. Αν και τα πρότυπα 802.11a / g χρησιμοποιούν παρόμοιους τύπους modulation, υπάρχει μία διαφορά. Το πρότυπο 802.16 χρησιμοποιεί Reed–Solomon μπλοκ κώδικα που υποστηρίζει Turbo coding. Η τελευταία δυνατότητα όμως είναι προαιρετική.

## 2.2.2 Υπο - Επίπεδο MAC

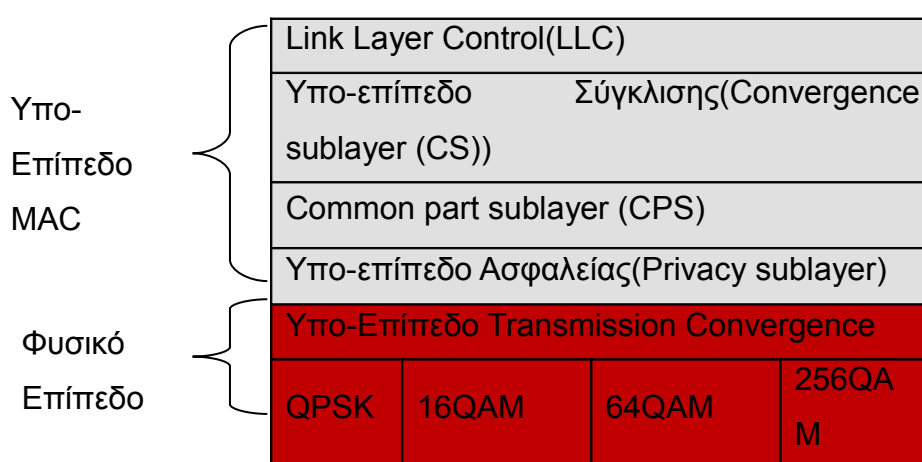
### 2.2.2.1 Γενικά [1]

Το επίπεδο MAC υποστηρίζει τις διαφορετικές προδιαγραφές του φυσικού επιπέδου χρησιμοποιώντας time division multiplexing, όπου υποδοχές χρόνου (time slots) αντιστοιχίζονται στους χρήστες για να αποκτήσουν πρόσβαση στο κανάλι. Η UpLink επικοινωνία βασίζεται στην Time Division Multiple Access (TDMA). Η TDMA επιτρέπει διαφορετικά επίπεδα QoS και συγκεκριμένη καθυστέρηση (delay) επικοινωνίας μέσω μιας προκαθορισμένης υπηρεσίας. Αυτό μπορεί να επιτευχθεί με την κατανομή Bandwidth με βάση έναν μηχανισμό

αίτησης/χορήγησης. Το πρότυπο 802.16-2004 υποστηρίζει TDD και FDD, Full και Half duplex.

Το πρότυπο 802.16-2004 είναι σχεδιασμένο να μεταφέρει οποιοδήποτε πρωτόκολλο υψηλότερου επιπέδου και υπηρεσίες αυτών, όπως IP εκδόσεις 4 και 6, VoIP, Ethernet, ATM και εικονικό τοπικό δίκτυο (VLAN). Το 802.16 το επιτυγχάνει αυτό, με το διαχωρισμό του επιπέδου MAC σε ξεχωριστά υπο-επίπεδα τα οποία χειρίζονται διαφορετικές υπηρεσίες:

Πίνακας 1: Δομή υπο-επιπέδων



### 2.2.2.1.1 Υπο-επίπεδο Σύγκλισης (Convergence sublayer (CS))

Το υπο-επίπεδο σύγκλισης (MESA, 2005) έχει σχεδιαστεί ώστε να αντιστοιχίζονται υπηρεσίες προς και από το επίπεδο MAC του 802.16.

Το 802.16 επίπεδο MAC παρέχει δύο υπηρεσίες, το υπο-επίπεδο σύγκλισης ATM και το υπο-επίπεδο σύγκλισης πακέτων. Το υπο-επίπεδο σύγκλισης πακέτων παρέχει υποστήριξη για IPv4, IPv6, Ethernet και VLAN.

Το κύριο καθήκον του υπο-επιπέδου σύγκλισης είναι να αντιστοιχίζει δεδομένα υψηλότερων πρωτοκόλλων στις σωστές υπηρεσίες. Επιπλέον, είναι υπεύθυνο για εκχώρηση εύρους ζώνης (Bandwidth) και QoS, καθώς και για την συμπίεση - ανασυγκρότηση των κεφαλίδων (headers) των πακέτων και για την ενίσχυση της αποτελεσματικότητας της ασύρματης σύνδεσης.

#### **2.2.2.1.2 Common part sublayer (CPS)**

Το 802.16 είναι σχεδιασμένο να υποστηρίζει αρχιτεκτονική δικτύου PMP (point-to-multipoint). Ωστόσο, και η λειτουργία τύπου mesh, γνωστή επίσης ως point-to-point αρχιτεκτονική, παρέχεται ως προαιρετική δυνατότητα.

#### **2.2.2.1.3 Υπο-επίπεδο Ασφαλείας(Privacy sublayer)**

Το Privacy sublayer είναι ένα ξεχωριστό υπο-επίπεδο ασφάλειας που παρέχει ανταλλαγή κλειδιών και κρυπτογράφηση. Το Privacy sublayer έχει δύο βασικά πρωτόκολλα:

- Πρωτόκολλο ενθυλάκωσης για την κρυπτογράφηση δεδομένων πακέτων σε ένα 802.16 δίκτυο.
- Ένα πρωτόκολλο διαχείρισης κλειδιών (Privacy Key Management - PKM) για τη διευκόλυνση της ασφαλούς διανομής των κλειδιών από τον σταθμό βάσης στον ασύρματο σταθμό. Το PKM ενισχύεται με την προσθήκη ελέγχου ψηφιακής πιστοποίησης της ταυτότητας ενός ασύρματου σταθμού που διευκολύνει την λειτουργία του 802.16.

Το πρωτόκολλο PKM χρησιμοποιείται για συσχετισμούς ασφαλείας. Είναι ένα σύνολο μεθόδων κρυπτογράφησης και κλειδιών. Το πρότυπο 802.16 καθορίζει τρία είδη συσχετισμών ασφαλείας: πρωταρχικός, στατικός και δυναμικός. Ο πρωταρχικός (primary) συσχετισμός ασφαλείας εδραιώνεται κατά τη διάρκεια της εκκίνησης λειτουργίας ενός ασύρματου σταθμού (SS-Simple Station). Ο στατικός συσχετισμός παρέχεται αποκλειστικά από έναν σταθμό βάσης (BS), ενώ ο δυναμικός ξεκινάει και περατώνεται ανάλογα με το αν δημιουργούνται ή τερματίζονται ροές δεδομένων.

#### **2.2.2.2 Βασικές Λειτουργίες Υπο-Επιπέδου MAC [3],[4]**

Το πρότυπο 802.16 υποστηρίζει δύο αρχιτεκτονικές δικτύου. Την αρχιτεκτονική PMP (Point –to- MultiPoint) και την αρχιτεκτονική Mesh.

Τα δίκτυα με αρχιτεκτονική PMP λειτουργούν με την λογική του ενός Σταθμού Βάσης και πολλαπλών ασύρματων τερματικών σταθμών οι οποίοι συνδέονται

πάνω του. Ο Σταθμός Βάσης είναι το κεντρικό σημείο αναφοράς και διαχειρίζεται όλη την κίνηση του δικτύου. Το πρότυπο 802.16 απαιτεί μια δημιουργία σύνδεσης και την δέσμευση του μονοπατιού από τον αποστολέα προς στον χρήστη μέχρι να τελειώσει η επικοινωνία. Κοινώς βασίζεται σε Connection Oriented επικοινωνία η οποία επιβάλλει σε κάθε έναν από τους ασυρμάτους σταθμούς την δημιουργία μιας σύνδεσης με τον σταθμό βάσης.

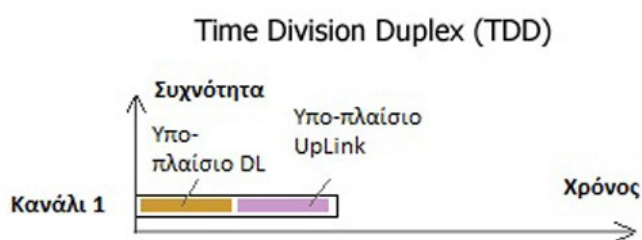
Ένας ασύρματος σταθμός αρχικά στέλνει ένα μήνυμα προς τον σταθμό βάσης με το οποίο του ζητάει να συσχετιστούν. Η συσχέτιση αυτή αφορά και το “κανάλι” σύνδεσης από τον ασύρματο σταθμό προς τον σταθμό βάσης (“κανάλι” UpLink) και το κανάλι από τον σταθμό βάσης προς τον ασύρματο σταθμό (“κανάλι” DownLink).

Τα δίκτυα με αρχιτεκτονική Mesh λειτουργούν με την λογική ότι υπάρχει ένας Σταθμός Βάσης και ένας ή περισσότεροι τερματικοί σταθμοί οι οποίοι λειτουργούν ταυτόχρονα και σαν πομποί για τους υπόλοιπους τερματικούς σταθμούς. Έτσι ένας τερματικός σταθμός δεν είναι ανάγκη να έχει άμεση πρόσβαση στον Σταθμό Βάσης γιατί μπορεί να επικοινωνήσει μαζί του έμμεσα μέσω άλλων τερματικών σταθμών που έχουν πρόσβαση σε αυτόν. Τα δίκτυα αυτά μπορούν να καλύψουν μεγαλύτερο εύρος ζώνης αλλά συνήθως δεν μπορούν να φτάσουν τις ταχύτητες των PMP δικτύων λόγω του ότι χρησιμοποιούνται ενδιάμεσα hops για να φτάσει ένας τερματικός σταθμός να επικοινωνήσει με τον Σταθμό Βάσης όταν δεν έχει άμεση πρόσβαση.

Το πρότυπο του WiMAX περιλαμβάνει δύο τεχνικές μετάδοσης σήματος. Την TDD (Time Division Duplex) και την FDD (Frequency Division Duplex) όπως αναφέρθηκε και παραπάνω.

Για την μελέτη και υλοποίηση δικτύων με προσανατολισμό την ποιότητα υπηρεσίας η κατάλληλη τεχνική είναι η TDD (Time Division Duplex) καθώς έτσι υπάρχει η δυνατότητα μετάδοσης υπο-πλαισίων (DownLink – UpLink SubFrames) μεταξύ του σταθμού βάσης και των τερματικών σταθμών με την χρήση ενός μόνο καναλιού.





Εικόνα 4: Τεχνική Time Division Duplex (TDD)

Δηλαδή, κάθε πλαίσιο χωρίζεται σε μικρότερα πλαίσια (υπο-πλαίσια) τα οποία ο σταθμός βάσης αντιμετωπίζει δυναμικά σαν time slots. Τα time slots είναι διαφορετικά για την επικοινωνία του UpLink και του DownLink καναλιού.

Για παράδειγμα, στην περίπτωση μιας απλής διαδικτυακής εφαρμογής, όπου η κίνηση κατευθύνεται κυρίως από το ίδιο το διαδίκτυο προς έναν συνδρομητή, ο σταθμός βάσης αναθέτει πιο πολλά time slots στο κανάλι του DL και λιγότερα σε αυτό του UL. Σε μια εφαρμογή VoIP από την άλλη, ο σταθμός βάσης αναθέτει τον ίδιο αριθμό time slots και στα δύο κανάλια.

Ειδικότερα, οι διαφορές των δύο τεχνικών μετάδοσης του σήματος είναι οι εξής:

Η TDD επιτρέπει ασύμμετρο ρυθμό μετάδοσης κίνησης ανάμεσα στα κανάλια DownLink και UpLink, ενώ η FDD δεν προβλέπει κάτι τέτοιο και ο ρυθμός μετάδοσης στα δύο κανάλια είναι πάντα σταθερός και ίσος.

Η TDD εγγυάται καλύτερη υποστήριξη σε τεχνολογίες κεραιών MIMO και γενικότερα σε προηγμένες τεχνολογίες κεραιών.

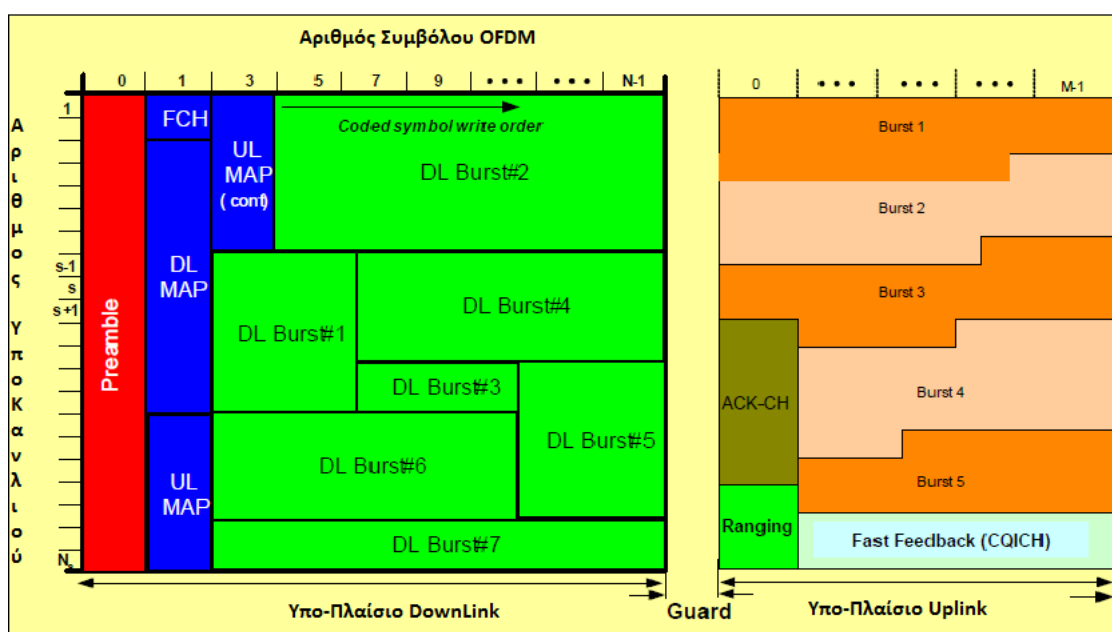
Αντίθετα με την FDD, η οποία απαιτεί ένα ζεύγος καναλιών, η TDD απαιτεί μόνο ένα κανάλι για την μετάδοση υπο-πλαισίων και στο κανάλι DownLink και στο κανάλι UpLink. Σαν αποτέλεσμα προσφέρει μεγαλύτερη ευελιξία και προσαρμοστικότητα στις διάφορες φασματικές κατανομές ανά τον κόσμο.

Οι δέκτες και πομποί (transceivers) που σχεδιάζονται για την TDD είναι πολύ πιο απλοί από άποψη κατασκευής και μοιραία πιο οικονομικοί.

Η εικόνα 5 δείχνει την δομή και τα περιεχόμενα ενός πλαισίου OFDM για τεχνική μετάδοσης TDD. Κάθε πλαίσιο χωρίζεται σε δύο υπο-πλαίσια που αντιστοιχούν

στα κανάλια DownLink και UpLink τα οποία χωρίζονται από κενά τύπου 'εκπομπής\λήψης' και 'λήψης\εκπομπής', ώστε να αποφευχθούν συγκρούσεις μεταξύ τους. Σε ένα τέτοιο πλαίσιο τα παρακάτω στοιχεία χρησιμοποιούνται ώστε να διασφαλιστεί η βέλτιστη απόδοσης του συστήματος:

- <sup>35</sup><sub>17</sub> Preamble: Το Preamble χρησιμοποιείται για λόγους συγχρονισμού και είναι το πρώτο σύμβολο ενός πλαισίου OFDM.
- <sup>35</sup><sub>17</sub> Frame Control Header(FCH) : Το σύμβολο FCH ακολουθεί το Preamble. Παρέχει γενικά στοιχεία των ρυθμίσεων του πλαισίου όπως το μέγεθος των μηνυμάτων MAP και την μορφή κωδικοποίησής τους και τα διαθέσιμα για χρήση υπο-κανάλια.
- <sup>35</sup><sub>17</sub> DL-MAP και UL-MAP : Τα στοιχεία DL-MAP και UL-MAP προσφέρουν πληροφορίες σχετικά με τον καταμερισμό των υπο-καναλιών και άλλα στοιχεία ελέγχου για τα υπο-πλαίσια UL και DL.
- <sup>35</sup><sub>17</sub> UL Ranging : Το στοιχείο UL Ranging χρησιμεύει σε ασύρματους σταθμούς του δικτύου ώστε να επαναπροσδιορίζουν στοιχεία όπως η συχνότητα , το απαιτούμενο bandwidth και κάποια στοιχεία που αφορούν την διαχείριση ενέργειας ενός ασύρματου σταθμού.
- <sup>35</sup><sub>17</sub> UL CQICH : Το στοιχείο UL CQICH επιτρέπει σε έναν ασύρματο σταθμό να στέλνει πληροφορίες - ενημερώσεις(feedback) σχετικές με το κανάλι επικοινωνίας.
- <sup>35</sup><sub>17</sub> UL ACK : Το στοιχείο UL ACK επιτρέπει σε έναν ασύρματο σταθμό να στέλνει επιβεβαιώσεις για το DownLink HARQ .



Εικόνα 5: Δομή πλαισίου OFDM TDD.

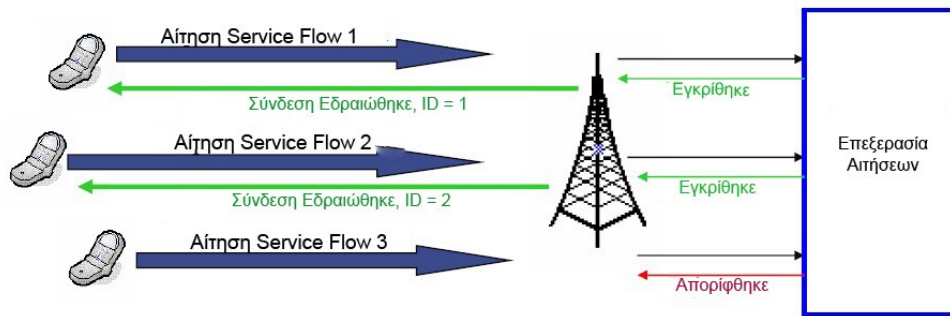
### 2.2.2.3 Ροή Υπηρεσιών (Service Flow) [1]

Γενικά μια Ροή Υπηρεσιών είναι μια ιδεατή οντότητα που φέρει την κυκλοφορία WiMAX σε ένα πλαίσιο WiMAX. Είναι δηλαδή στην ουσία μια ροή δεδομένων. Μια ροή υπηρεσιών αποτελεί μέρος μιας Κλάσης Υπηρεσιών (Service Class). Μια κλάση υπηρεσιών ορίζει την σύμβαση της κυκλοφορίας δεδομένων μεταξύ τερματικού σταθμού και σταθμού βάσης.

Μια Ροή Υπηρεσιών περιλαμβάνει σημαντικά χαρακτηριστικά για την εξασφάλιση της ποιότητας της επικοινωνίας (QoS) όπως την διαμόρφωση και κωδικοποίηση (modulation & coding), το σύστημα αναμετάδοσης, το μέγεθος των buffers, το μέγεθος του SDU πακέτου, κ.λπ.

Πιο συγκεκριμένα όταν ο σταθμός βάσης λάβει μια αίτηση σύνδεσης από έναν ασύρματο σταθμό, αρχικά ελέγχει εάν ο χρήστης του σταθμού αυτού ικανοποιεί τις προϋποθέσεις ώστε να λάβει εξουσιοδότηση χρήσης της υπηρεσίας. Ο σταθμός βάσης στην συνέχεια ελέγχει εάν υπάρχουν διαθέσιμοι δικτυακοί πόροι (διαθέσιμο bandwidth για παράδειγμα), ώστε να εξυπηρετήσει την αίτηση αυτή. Εάν υπάρχουν, ο σταθμός βάσης συσχετίζει την αίτηση σύνδεσης με μια Ροή Υπηρεσιών και τα δεδομένα ποιότητας επικοινωνίας που καθορίζονται μέσω

αυτής. Κάθε τέτοια σύνδεση χαρακτηρίζεται από ένα αναγνωριστικό μεγέθους 16bit (CID). Να σημειωθεί ότι το πρότυπο IEEE 802.16 δεν χρησιμοποιεί φυσικές διευθύνσεις (Mac Addresses) πομπού και δέκτη στο πλαίσιο MAC.



Εικόνα 6: Ροές υπηρεσιών WIMAX

Μια Ροή Υπηρεσιών έχει τα παρακάτω γνωρίσματα:

- <sup>35</sup><sub>17</sub> Αναγνωριστικό Ροής Υπηρεσιών (Service Flow ID – SFID): Κάθε ροή υπηρεσιών έχει ένα μοναδικό αναγνωριστικό το οποίο καθορίζει την κατεύθυνση της μετάδοσης (DL ή UL).
- <sup>35</sup><sub>17</sub> Αναγνωριστικό Σύνδεσης (CID – Connection ID): Ένα αναγνωριστικό σύνδεσης προσδιορίζει ένα SFID μετά την ολοκλήρωση της σύνδεσης πομπού δέκτη.
- <sup>35</sup><sub>17</sub> Προκαθορισμένες Παράμετροι ποιότητας επικοινωνίας: Η ποιότητα επικοινωνίας (QoS) προκαθορίζεται από ένα κατάλληλο δικτυακό σύστημα διαχείρισης.
- <sup>35</sup><sub>17</sub> Δεδομένες Παράμετροι ποιότητας επικοινωνίας: Παράμετροι της ποιότητας επικοινωνίας για τις οποίες ο σταθμός βάσης δεσμεύει εξ' αρχής πόρους. Ο σημαντικότερος από αυτούς τους πόρους είναι το απαραίτητο bandwidth.
- <sup>35</sup><sub>17</sub> Ενεργές παράμετροι της ποιότητας επικοινωνίας: Οι παράμετροι που στην πραγματικότητα παρέχονται για μια ροή υπηρεσιών. Μόνο ενεργές ροές δεδομένων μπορούν να στείλουν πακέτα στην ασύρματη διασύνδεση.

Μια ροή υπηρεσιών μπορεί να προκαθοριστεί στατικά από το δικτυακό σύστημα διαχείρισης ή να δημιουργηθεί δυναμικά από τα παρακάτω μηνύματα ελέγχου που υποστηρίζονται από το 802.16:

<sup>35</sup><sub>17</sub> Δυναμική πρόσθεση υπηρεσίας (Dynamic Service Addition – DSA):  
Χρησιμοποιείται για την δημιουργία νέας ροής υπηρεσιών.

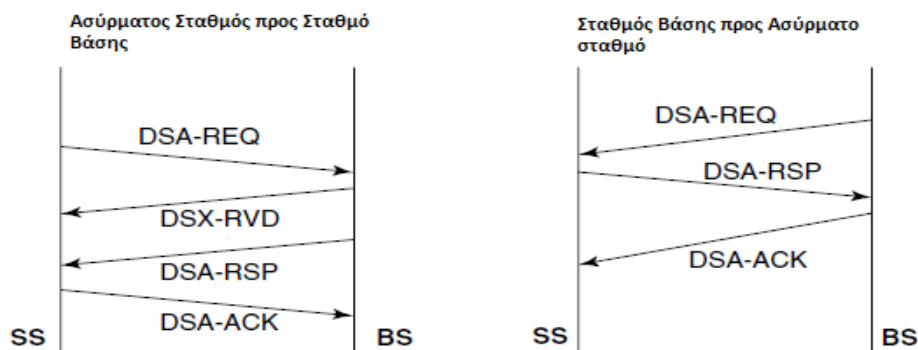
<sup>35</sup><sub>17</sub> Δυναμική τροποποίηση υπηρεσίας (Dynamic Service Change - DSC):  
Χρησιμοποιείται για την τυχόν τροποποίηση μιας υπάρχουσας ροής υπηρεσιών.

<sup>35</sup><sub>17</sub> Δυναμική διαγραφή υπηρεσίας (Dynamic Service Deletion - DSD):  
Χρησιμοποιείται για την διαγραφή μιας ροής υπηρεσιών.

Αυτά τα μηνύματα ελέγχου επιπέδου MAC επιτρέπουν στους εκάστοτε παρόχους υπηρεσιών δικτύου, να προσθέσουν εύκολα νέους συνδρομητές, να τροποποιήσουν την ποιότητα υπηρεσιών για τους υπάρχοντες συνδρομητές, να επανακαταμερίσουν το διαθέσιμο bandwidth στις υπάρχουσες διασυνδέσεις και να ανακτήσουν μη χρησιμοποιημένους πόρους. Όλες αυτές οι ενέργειες μπορούν να πραγματοποιηθούν σε πραγματικό χρόνο χωρίς να υπάρχει παρεμβολή στις ενεργές υπηρεσίες των εκάστοτε συνδρομητών.

Μια δυναμική αίτηση για κάποια υπηρεσία μπορεί να δημιουργηθεί είτε από έναν σταθμό βάσης είτε από έναν ασύρματο σταθμό. Στην περίπτωση που ξεκινήσει από τον ασύρματο σταθμό, ένα μήνυμα αίτησης DSA – REQ (Request) στέλνεται το οποίο έχει μια αναφορά στην ροή υπηρεσιών του και καθορισμένες τις παραμέτρους ποιότητας επικοινωνίας που χρειάζεται. Μόλις ο σταθμός βάσης λάβει την αίτηση DSA – REQ, στέλνει στον ασύρματο σταθμό ένα μήνυμα απάντησης DSX – RVD (Received) για να τον πληροφορήσει ότι έλαβε επιτυχημένα. Στην συνέχεια ο σταθμός βάσης στέλνει ένα μήνυμα τύπου DSA – RSP (Response) στον ασύρματο σταθμό για να τον πληροφορήσει εάν μπορεί να τον εξυπηρετήσει ή όχι τελικά. Ο ασύρματος σταθμός τέλος στέλνει με την σειρά του ένα μήνυμα DSA – ACK (Acknowledgment) για να επιβεβαιώσει ότι έλαβε την ενημέρωση. Η διαδικασία αυτή εφόσον ξεκινήσει από τον σταθμό βάσης είναι η

ίδια με την διαφορά ότι δεν περιλαμβάνει το μήνυμα τύπου DSX – RVD . Το σχετικό διάγραμμα φαίνεται παρακάτω.



Εικόνα 7: Dynamic service addition (DSA) - Ροή μηνυμάτων

Η μετάδοση δεδομένων στο κανάλι του DownLink ελέγχεται από τον σταθμό βάσης και πλαίσια MAC αποστέλλονται μαζικά (Broadcasted) σε όλους τους ασύρματους σταθμούς. Όταν κάποιος ασύρματος σταθμός λάβει ένα τέτοιο πλαίσιο ελέγχει το CID του ώστε να διαπιστώσει εάν προορίζεται για αυτόν. Για να βελτιωθεί η απόδοση της μετάδοσης δεδομένων, πολλαπλά πλαίσια συνδυάζονται ώστε να αποτελέσουν μια 'ριπή κίνησης'(burst), η οποία προορίζεται για μαζική αποστολή μέσω του αέρα. Όταν ένας ασύρματος σταθμός αντιληφθεί αυτή την 'ριπή κίνησης' θα παραλάβει μόνο τα πλαίσια τα οποία περιέχουν το σωστό για αυτόν CID και θα απορρίψει τα υπόλοιπα.

Κάθε τέτοια "ριπή κίνησης" (burst) περιλαμβάνει μηνύματα τύπου MAP , ένα για κάθε κανάλι επικοινωνίας (DL-MAP και UL-MAP), τα οποία καθορίζουν την δομή της ριπής και τον τρόπο με τον οποίο θα πρέπει οι ασύρματοι σταθμοί να ανακτήσουν - ξεχωρίσουν τα πλαίσια MAC. Η μετάδοση δεδομένων για το κανάλι UpLink είναι πιο πολύπλοκη από το κανάλι του DownLink, καθώς όλοι οι ασύρματοι σταθμοί πρέπει αρχικά να συγχρονιστούν με τον σταθμό βάσης για να ξεκινήσει αυτή. Η πολυπλοκότητα αυτή οφείλεται στο κομμάτι της ποιότητας της επικοινωνίας το οποίο αναφέρεται παρακάτω.

#### 2.2.2.4 Υποστήριξη QoS (Quality of Service) στο 802.16

Το πρότυπο 802.16 καθορίζει τέσσερις προδιαγραφές κλάσεων υπηρεσιών (Service Classes):

**Unsolicited grant service (UGS):** Αυτή η προδιαγραφή υπηρεσιών χρησιμοποιείται κυρίως για την μετάδοση μη-συμπιεσμένου τύπου φωνής. Η προδιαγραφή προϋποθέτει σταθερό όγκο δεδομένων ανά ένα σταθερό χρονικό διάστημα και εγγυάται υψηλή ρυθμαπόδοση και χαμηλές καθυστερήσεις.

**Real-time polling service (rtPS):** Η προδιαγραφή αυτή είναι κατάλληλη για συμπιεσμένη κίνηση πολυμέσων, όπως είναι για παράδειγμα το Video Streaming αλλά και άλλες εφαρμογές πραγματικού χρόνου όπου το ποσό του απαραίτητου bandwidth μπορεί να διαφοροποιείται ανά πάσα στιγμή. Η προδιαγραφή απαιτεί από τον εκάστοτε σταθμό βάσης να εφαρμόσει μια υπηρεσία σταθμοσκόπησης (polling) ανά καθορισμένο σταθερό χρονικό διάστημα. Σε κάθε τέτοια επικοινωνία σταθμού βάσης και ασυρμάτου σταθμού, ο σταθμός βάσης ζητάει από τον ασύρματο σταθμό να του επανακαθορίσει τις ανάγκες του σε bandwidth.

**Nonreal-time polling service (nrtPS):** Αυτή η προδιαγραφή υπηρεσιών είναι κατάλληλη για εφαρμογές μη πραγματικού χρόνου οι οποίες χρειάζονται εγγυημένη απόδοση. Επιβάλλει και αυτή υπηρεσία σταθμοσκόπησης (polling) στον σταθμό βάσης σε καθορισμένο χρόνο, με την διαφορά όμως ότι ο ασύρματος σταθμός υπάρχει η δυνατότητα να μην απαντήσει πάντα σε αυτόν. Αν ένας ασύρματος σταθμός δεν απαντήσει στην αίτηση επικοινωνίας του σταθμού βάσης μετά από “n” φορές στην σειρά, ο σταθμός βάσης τοποθετεί τον ασύρματο σταθμό σε μια ομάδα αναμονής. Όταν ο σταθμός βάσης επικοινωνήσει εκ νέου με την ομάδα αναμονής, οι ασύρματοι σταθμοί που έχουν τοποθετηθεί εκεί, συναγωνίζονται για την εκ νέου πρόσβασή τους στην παραπάνω λειτουργία. Ο μηχανισμός αυτός στην ουσία δεν επιτρέπει για ένα απαραίτητο χρονικό διάστημα το οποίο χρειάζεται ο σταθμός βάσης να διαχειριστεί την κίνηση του δικτύου αποτελεσματικά, σε σταθμούς οι οποίοι δεν παράγουν μεγάλο όγκο κίνησης να καταναλώνουν πόρους του μηχανισμού επικοινωνίας.

**Best effort (BE) service:** Η προδιαγραφή αυτή δεν απαιτεί μηχανισμό επικοινωνίας. Ένας ασύρματος σταθμός θα πρέπει να συναγωνιστεί με τους υπολοίπους για το απαιτούμενο bandwidth και την πρόσβαση του στο δίκτυο. Οι αιτήσεις για bandwidth περιέχονται σε time slots στα μηνύματα διαχείρισης του καναλιού Uplink (UL-MAP) για τα οποία ο κάθε σταθμός πρέπει να ανταγωνιστεί κάποιον άλλο. Αν μια αίτηση ενός ασύρματου σταθμού είναι επιτυχημένη, αυτό θα του γίνει γνωστό μέσω του αντίστοιχου μηνύματος διαχείρισης για το κανάλι DownLink (DL-MAP) και τότε ο σταθμός μπορεί να μεταδώσει τα δεδομένα του. Αν η αίτηση δεν είναι επιτυχημένη ο ασύρματος σταθμός θα πρέπει να προσπαθήσει ξανά μετά την πάροδο κάποιου χρόνου. Φυσικά υπάρχει η περίπτωση και η εκ νέου αίτηση να 'συγκρουστεί' με την αίτηση κάποιου άλλου σταθμού. Σε αυτήν την περίπτωση εφαρμόζεται ο ίδιος αλγόριθμος που εφαρμόζεται και για το Ethernet.

Σε αυτό στο σημείο είναι αναγκαίο να παρουσιαστούν τα στοιχεία μέσω των οποίων μπορούν οι παραπάνω προδιαγραφές, ορισμένες στα πλαίσια κλάσεων υπηρεσιών, να υλοποιηθούν την ποιότητα στην επικοινωνία.

**Μέγιστος επιτρεπτός ρυθμός απόδοσης(Maximum sustained traffic rate (MSTR)):**

Ο μέγιστος επιτρεπτός ρυθμός απόδοσης μετρημένος σε bit ανά δευτερόλεπτο (bps) μιας ροής υπηρεσιών. Μέσω μιας κλάσης υπηρεσιών η ασύρματη διασύνδεση συμμορφώνεται με αυτό. Δεν επιτρέπεται σε καμία περίπτωση κίνηση μεγαλύτερη από αυτό το όριο για μια ροή υπηρεσιών.

**Ελάχιστος δεσμευμένος ρυθμός απόδοσης(Minimum reserved traffic rate (MRTR)):**

Ο ελάχιστος δεσμευμένος ρυθμός απόδοσης μετρημένος σε bit ανά δευτερόλεπτο (bps) μιας ροής υπηρεσιών. Αυτός ο ρυθμός είναι εγγυημένος σε κάθε περίπτωση. Αν υπάρχει επιπλέον διαθέσιμο bandwidth στο δίκτυο, τότε μια ροή υπηρεσιών μπορεί να το χρησιμοποιήσει μέχρι βέβαια να φτάσει στο όριο του μέγιστου επιτρεπτού ρυθμού απόδοσης.



**Μέγιστη καθυστέρηση ροής (Maximum latency):** Η μέγιστη επιτρεπτή καθυστέρηση μετρημένη σε milliseconds ανάμεσα στην αποστολή και την παραλαβή ενός πακέτου από τον σταθμό βάσης προς τον ασύρματο σταθμό και αντίστροφα.

**Μέγιστος μέγεθος “ριπής κίνησης” (Maximum traffic burst):** Το μέγιστο μέγεθος burst κίνησης για μια ροή υπηρεσιών. Το χαρακτηριστικό αυτό δεν υποστηρίζεται από τον OPNET MODELLER v16.0 και δεν θα απασχολήσει την εργασία αυτή.

### 2.2.2.5 Αιτήσεις Σύνδεσης και Bandwidth

Μια αίτηση σύνδεσης μπορεί να προέρχεται είτε από την σταθμό βάσης είτε από κάποιον ασύρματο σταθμό. Το πρότυπο 802.16 διαθέτει ειδικό μηχανισμό ο οποίος προβλέπει τις διαδικασίες την αίτησης σύνδεσης αλλά και το πώς και αν αυτή θα εγκριθεί.

Όταν ένας ασύρματος σταθμός δημιουργεί και στέλνει μια αίτηση στον σταθμό βάσης με την οποία του ζητάει ορισμένο bandwidth και ορισμένες παραμέτρους ποιότητας επικοινωνίας της σύνδεσης, θα πρέπει πρώτα απ’ όλα να υπάρξει αυθεντικοποίηση του σταθμού αυτού από τον σταθμό βάσης. Μετά το πέρας της διαδικασίας αυθεντικοποίησης ο σταθμός βάσης θα πρέπει να αποφανθεί για το κατά πόσο υπάρχουν διαθέσιμοι πόροι να προσφέρει στον ασύρματο σταθμό. Ο γενικός τύπος όσον αφορά το Bandwidth είναι ο εξής: [1],[2]

$$\Sigma (\text{Συνολικό bandwidth που έχει δεσμευτεί}) + \text{νέα αίτηση για bandwidth} \leq \text{Συνολικό Bandwidth}$$

Στην περίπτωση της UGS, το bandwidth που πρέπει να δεσμευτεί για κάθε σταθμό είναι ο Μέγιστος επιτρεπτός ρυθμός απόδοσης (Maximum sustained traffic rate (MSTR)) που αναφέρθηκε παραπάνω. Για τις περιπτώσεις των rtPS και nrtPS το bandwidth που πρέπει να δεσμευτεί για κάθε σταθμό είναι ο Ελάχιστος δεσμευμένος ρυθμός απόδοσης (Minimum reserved traffic rate (MRTR)). Για την Best Effort προδιαγραφή δεν υπάρχει δεσμευμένο bandwidth.

Ένα ακόμη σημαντικό στοιχείο του προτύπου 802.16 είναι ότι το bandwidth των καναλιών UpLink και DownLink μπορεί να διατίθεται δυναμικά και ανάλογα με τις ανάγκες των χρηστών. Δηλαδή εάν υπάρχουν περισσότερες ανάγκες για το κανάλι DownLink και την ίδια στιγμή υπάρχουν διαθέσιμοι πόροι στο κανάλι UpLink, ο σταθμός βάσης έχει την δυνατότητα να αυξήσει τα time slots που διαθέτει για το κανάλι DownLink και να μειώσει αντίστοιχα αυτά του UpLink.

Μια αίτηση για Bandwidth από έναν ασύρματο σταθμό αποστέλλεται συνήθως την ώρα που γίνεται ο προγραμματισμός για ολόκληρο το UpLink κανάλι επικοινωνίας. Το πρότυπο 802.16 προσφέρει όμως και έναν προαιρετικό μηχανισμό που χρησιμεύει για τον ίδιο σκοπό και ονομάζεται “αίτηση piggyback”. Πρέπει να σημειωθεί ότι σε μια τέτοια αίτηση δεν προσφέρεται ασφάλεια μετάδοσης. Κοινώς υπάρχει πιθανότητα να χαθεί στην πορεία λόγω κάποιας σύγκρουσης. Σαν αποτέλεσμα ο σταθμός βάσης θα πρέπει να αποστείλει ένα μήνυμα τύπου DSX-RVD για να επιβεβαιώσει την παραλαβή του μηνύματος τύπου DSA-REQ. Ο σταθμός βάσης χορηγεί το bandwidth που του ζητήθηκε ενσωματώνοντας το μέσα σε ένα μήνυμα διαχείρισης του UpLink καναλιού δηλαδή σε ένα UL-MAP το οποίο αποστέλλεται μαζικά σε όλους τους σταθμούς και οι σταθμοί με βάση το CID ξεχωρίζουν αυτό που προορίζεται για αυτούς.

#### **2.2.2.6 Προγραμματισμός Ποιότητας Επικοινωνίας (QoS) [1],[2]**

Όταν μια αίτηση σύνδεσης λάβει απάντηση και χορηγηθεί, μια ροή υπηρεσιών η οποία περιέχει και τα χαρακτηριστικά για την ποιότητα επικοινωνίας δημιουργείται για την σύνδεση αυτή. Οι υπηρεσίες προγραμματισμού των προδιαγραφών υπηρεσιών που αναφέρθηκαν παραπάνω, είναι αυτές που διαχειρίζονται όλα τα δεδομένα ώστε να υποστηρίξουν τελικά την γενικότερη υπηρεσία προγραμματισμού επιπέδου MAC για την μετάδοση δεδομένων μέσω μιας σύνδεσης.

Ο σταθμός βάσης ελέγχει τον προγραμματισμό και των δύο καναλιών μετάδοσης (UL και DL). Ο σταθμός βάσης λοιπόν μέσω της υπηρεσίας προγραμματισμού, υπολογίζει τις απαιτήσεις για ρυθμαπόδοση και καθυστέρηση της κίνησης για τα

κανάλια UL και DL και παρέχει τις αιτήσεις και τις απαντήσεις αυτών ανά κατάλληλα χρονικά διαστήματα.

Για το κανάλι DL τα πράγματα είναι πιο εύκολα για την υπηρεσία προγραμματισμού καθώς οι πληροφορίες, βασισμένες στα καθορισμένα στοιχεία του QoS των πλαισίων, μεταδίδονται μαζικά μέσω μιας “ριπής κίνησης” (traffic burst).

Για το κανάλι UL η υπηρεσία προγραμματισμού χρησιμοποιεί έναν πιο σύνθετο μηχανισμό αίτησης-απάντησης αφού απαιτείται και συγχρονισμός ανάμεσα στον σταθμό βάσης και τους διάφορους ασύρματους σταθμούς.

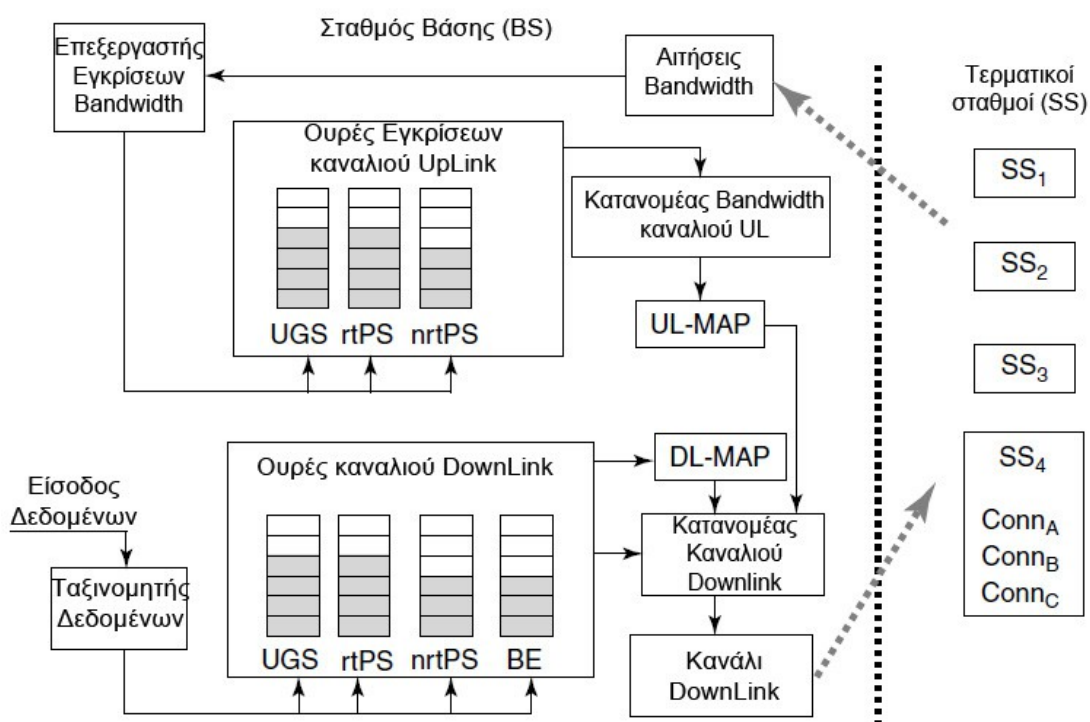
### **Προγραμματισμός Προδιαγραφής Υπηρεσιών UGS**

Η προδιαγραφή UGS σχεδιάστηκε ώστε να υποστηρίζει υπηρεσίες πραγματικού χρόνου με σταθερό μέγεθος πλαισίων ανά σταθερά χρονικά διαστήματα. Η προδιαγραφή παρέχει απαντήσεις σταθερού μεγέθους ανά τακτά χρονικά διαστήματα, με στόχο να εξαλείψει την καθυστέρηση που παρατηρείται κατά την αποστολή των αιτήσεων από την μεριά των ασυρμάτων σταθμών και να εξασφαλίσει έτσι ότι οι απαιτήσεις μιας εφαρμογής πραγματικού χρόνου καλύπτονται.

Ο σταθμός βάσης παρέχει τις αποκρίσεις του προς τους ασύρματους σταθμούς ανά καθορισμένα χρονικά διαστήματα βασιζόμενος πάντα στον Μέγιστο επιτρεπτό ρυθμό απόδοσης (Maximum sustained traffic rate (MSTR)). Το μέγεθος των αποκρίσεων αυτών είναι αρκετά μεγάλο ώστε να περιέχουν τα καθορισμένου μεγέθους δεδομένα αλλά και τις επιβαρύνσεις του επιπέδου MAC (overhead). Ο ασύρματος σταθμός λαμβάνει μαζικές αποστολές πλαισίων (πλαίσια broadcast) από τον σταθμό βάσης ανά τακτά χρονικά διαστήματα. Το μήνυμα διαχείρισης (UL-MAP) που περιέχεται στο broadcast πλαίσιο, περιλαμβάνει πληροφορίες σχετικά με το κανάλι Uplink ώστε ο ασύρματος σταθμός να μπορέσει να μεταδώσει τα δεδομένα του.

Μόλις ο ασύρματος σταθμός είναι έτοιμος να μεταδώσει δεδομένα, τα τοποθετεί στο κανάλι που του έχει εκχωρηθεί(στην ουσία είναι ένα time slot) και στέλνει τα

δεδομένα αυτά άμεσα μέσω του αέρα. Δεν υπάρχει διαμοιρασμός bandwidth πολλαπλών συνδέσεων για την προδιαγραφή UGS και έτσι σε κάθε σύνδεση (service flow) ανατίθεται ένα αποκλειστικό κανάλι (time slot) για την μετάδοση προς τον σταθμό βάσης.



Εικόνα 8: Προγραμματισμός καναλιών UL και DL σταθμού βάσης

### Προγραμματισμός Προδιαγραφής Υπηρεσιών rtPS

Η προδιαγραφή υπηρεσιών rtPS έχει σχεδιαστεί ώστε να υποστηρίζει εφαρμογές πραγματικού χρόνου με μεταβλητό μέγεθος πλαισίων ανά καθορισμένα χρονικά διαστήματα. Τέτοιες εφαρμογές είναι το Streaming ήχου και εικόνας. Η προδιαγραφή επιτρέπει σε έναν ασύρματο σταθμό να συγκεκριμενοποιεί το μέγεθος του bandwidth που χρειάζεται και για αυτόν ακριβώς τον λόγο οι αιτήσεις προδιαγραφής rtPS περιέχουν περισσότερο Overhead από την προδιαγραφή UGS.

Ο σταθμός βάσης προσφέρει ανοιχτά ευκαιρίες στους σταθμούς ώστε να χρησιμοποιήσουν το κανάλι UL για τις αιτήσεις τους. Πολλαπλές συνδέσεις της προδιαγραφής ttPS μοιράζονται το ίδιο Bandwidth για να πραγματοποιήσουν την αποστολή δεδομένων τους από το κανάλι UL. Ένας σταθμός μπορεί να στείλει τα δεδομένα του την ώρα που πραγματοποιείται η καθιερωμένη επιβεβαίωση σύνδεσης ανάμεσα στον σταθμό βάσης και τον ασύρματο σταθμό (rolling). Η υλοποίηση της διαδικασίας σταθμοσκόπησης δεν είναι καθορισμένη από το πρότυπο 802.16 οπότε ο κάθε κατασκευαστής μπορεί να χρησιμοποιήσει μια αντίστοιχη υπηρεσία δικής του κατασκευής.

Ακολουθεί ένα παράδειγμα του υπηρεσίας σταθμοσκόπησης:

Ένας ασύρματος σταθμός έχει συνδεθεί με τον σταθμό βάσης και χρησιμοποιείται η προδιαγραφή ttPS. Στον ασύρματο σταθμό έχει εγγραφεί bandwidth της τάξης των 400Kb το δευτερόλεπτο και καθυστέρηση της τάξης των 50ms.

Ο σταθμός βάσης επικοινωνεί με τον ασύρματο σταθμό κάθε 50ms.

Όποτε ο ασύρματος σταθμός έχει δεδομένα να αποστείλει περιμένει την σταθμοσκόπηση από τον σταθμό βάσης, η οποία πραγματοποιείται όταν ο σταθμός βάσης στείλει ακριβώς ένα τέτοιου τύπου μήνυμα μαζικά στο κανάλι DL. Στην συνέχεια ο σταθμός εντοπίζει το δικό του Polling μήνυμα και το δικό του CID τα οποία περιέχεται στο μήνυμα UL-MAP.

Όταν ο ασύρματος σταθμός λάβει το μήνυμα και πλέον ξέρει το χρονικό διάστημα το οποίο πρέπει κάθε φορά να παρέλθει ώστε να πραγματοποιείται η διαδικασία της σταθμοσκόπησης , τότε λαμβάνει και τις πληροφορίες σχετικά με το bandwidth που του αναλογεί εκείνη την στιγμή πάλι μέσω του UL-MAP και χρησιμοποιεί αυτό το Bandwidth για να στείλει τα απαραίτητα δεδομένα μέσω του καθορισμένου UL καναλιού.

Αν χρειάζεται να στείλει και άλλα δεδομένα και δεν επαρκούσε το αρχικό bandwidth που του χορηγήθηκε, ο σταθμός κάνει χρήση του δικαιώματός του να χρησιμοποιήσει τα 400Kb ανά δευτερόλεπτο που του είχαν εγγραφεί αρχικά.

Αφού στείλει όλα του τα δεδομένα περιμένει και πάλι την διαδικασία της σταθμοσκόπησης.

### **Προγραμματισμός Προδιαγραφής Υπηρεσιών nrtPS**

Η προδιαγραφή υπηρεσιών nrtPS προβλέπει την χρήση της υπηρεσίας σταθμοσκόπησης ανά τακτά χρονικά διαστήματα τα οποία όμως δεν είναι σαφώς καθορισμένα και εξασφαλίζει στη ροή υπηρεσίας ευκαιρίες για αιτήσεις ακόμη και σε καταστάσεις συμφόρησης δικτύου. Συνήθως η σταθμοσκόπηση της προδιαγραφής nrtPS από τον σταθμό βάσης πραγματοποιείται με μια συχνότητα περίπου ενός δευτερολέπτου. Ο σταθμός βάσης θα πρέπει να προσφέρει τις κατάλληλες ευκαιρίες για αιτήσεις στους ασύρματους σταθμούς όπως αυτές έχουν καθοριστεί στις παραμέτρους της ποιότητας επικοινωνίας.

### **Προγραμματισμός Προδιαγραφής Υπηρεσιών BE**

Η προδιαγραφή BE χρησιμοποιείται ώστε να κάνει την κίνηση τύπου best effort όσο το δυνατόν αποδοτική. Οι ασύρματοι σταθμοί επιτρέπεται να χρησιμοποιούν αιτήσεις σε συνθήκες συμφόρησης δικτύου. Σύγκρουση μπορεί να υπάρξει όταν πολλαπλοί σταθμοί μεταδώσουν δεδομένα την ίδια χρονική στιγμή. Όταν κάτι τέτοιο συμβεί, ο κάθε ασύρματος σταθμός χρησιμοποιεί έναν αλγόριθμο ο οποίος τους επιτρέπει να μεταδώσουν τα δεδομένα τους όσο το δυνατόν γρηγορότερα. Οι εγκρίσεις στις αιτήσεις των ασύρματων σταθμών στέλνονται μέσω του καναλιού DL και χρησιμοποιείται και εδώ το μήνυμα διαχείρισης UL-MAP για να καθοριστεί το UL κανάλι το οποίο θα χρησιμοποιηθεί για την αποστολή των δεδομένων προς τον σταθμό βάσης.

## **2.3 IEEE 802.16e-2005**

### **2.3.1 Γενικά**

Το IEEE 802.16e-2005 αποτελεί στην ουσία μια επέκταση και ουσιώδη τροποποίηση του IEEE 802.16-2004. Παρακάτω περιγράφονται οι αλλαγές που εισήγαγε στο φυσικό επίπεδο του προτύπου.

### **2.3.2 Φυσικό Επίπεδο**

Η λειτουργία του 802.16e περιορίζει την λειτουργία του σε επιτρεπόμενες ζώνες συχνότητας, κατάλληλες να υποστηρίξουν την κινητικότητα κόμβων, κάτω των 6 GHz. Αυτό μπορούν να προκαλέσει πρόβλημα συμβατότητας μεταξύ των 802.16-2004 και 802.16e, δεδομένου ότι το διαθέσιμο με άδεια χρήσης ραδιοφάσμα μπορεί να χρειαστεί να χωριστεί για να υποστηρίξει και τις δύο τεχνολογίες.

Το 802.16e ορίζει ένα νέο air Interface φυσικού επιπέδου με δυνατότητα κλιμάκωσης που ονομάζεται S-OFDMA (Scalable-OFDMA), εκτός από εκείνα που ορίζονται από το 802.16-2004. Το S-OFDMA χρησιμοποιεί Fast Fourier Transform (FFT) όπως και το OFDM, όμως “υπο-μεταφορείς” μεγέθους 128, 512, 1024 ή 2048. Το S-OFDMA χρησιμοποιεί αυτό τον αριθμό των subcarriers ώστε να παρέχει τη δυνατότητα αύξησης εύρους ζώνης (Bandwidth), ενώ την ίδια στιγμή η διάρκεια διαχωρισμού των subcarriers και των συμβόλων παραμένουν σταθερά κατά την οποιαδήποτε αλλαγή του bandwidth. [1]

Το AAS (Advanced Antenna System), το space time coding και το MIMO κλειστού βρόχου ενισχύουν το 802.16e με στόχο την βελτίωση της περιοχής κάλυψης και της ταχύτητας μετάδοσης δεδομένων. Επιπλέον, η υποστήριξη για Space-Division Multiple Access (SDMA) εισάγατε.

Το 802.16e περιλαμβάνει μια πρόσθετη, προηγμένη, χαμηλής πολυπλοκότητας μέθοδο κωδικοποίησης, την low-density parity check (LDPC) για μια πιο ευέλικτη κωδικοποίηση. Η LDPC κωδικοποιεί 6 bit για κάθε 5 bit δεδομένων με ρυθμό 5/6. Αυτό έχει σαν αποτέλεσμα την επίτευξη υψηλότερων επιδόσεων κωδικοποίησης από τις μεθόδους που περιλαμβάνονται στο 802.16-2004 και παρέχουν ρυθμό κωδικοποίησης 3/4.

#### **2.3.2.1 Λεπτομέρειες Μεταβλητού OFDMA(Scalable-OFDMA) [3]**

Το IEEE 802.16e-2005 OFDMA είναι βασισμένο στις ιδέες που διέπουν το S-OFDMA. Το S-OFDMA υποστηρίζει ένα ευρύ φάσμα διαφορετικών τιμών bandwidth ώστε να μπορέσει να αντιμετωπίσει το πρόβλημα των διαφορετικών φασματικών κατανομών και των διαφορετικών αναγκών ανάλογα με τις επιλογές

των εκάστοτε δικτυακών μοντέλων. Η έννοια της μεταβλητότητας εφαρμόζεται στο γεγονός ότι προσφέρεται η δυνατότητα ρύθμισης του μεγέθους του FFT ενώ η τιμή του Διαστήματος Συχνότητας ανά υπο-μεταφορέα είναι πάντα σταθερή και ίση με 10.94 kHz. Λαμβάνοντας ακόμη υπόψη ότι η διάρκεια συμβόλων είναι και αυτή σταθερή, προκύπτει ότι οι επιπτώσεις στα υψηλότερα επίπεδα του δικτύου από την εναλλαγή του bandwidth είναι μηδαμινές. Οι παράμετροι του S-OFDMA και οι τιμές τους κατά περίπτωση φαίνονται στον πίνακα 2:

ΠΙΝΑΚΑΣ 2: ΠΑΡΑΜΕΤΡΟΙ S-OFDMA

ΠΑΡΑΜΕΤΡΟΙ	ΤΙΜΕΣ			
Εύρος Ζώνης Καναλιού Συστήματος (MHZ)	1,25	5	10	20
Μέγεθος FFT(Fp σε MHZ)	128	512	1024	2048
Αριθμός Υπο-Καναλιών	2	8	16	32
Διάστημα Συχνότητας ανά Υπό-Μεταφορέα	10,94 KHz			
Διάρκεια Χρήσιμων Συμβόλων ( $T_b=1/f$ )	91,4 microseconds			

### 2.3.2.2 Άλλοι Μηχανισμοί Φυσικού Επιπέδου

Οι μηχανισμοί της Προσαρμοσίμης Διαμόρφωσης και Κωδικοποίησης (Adaptive modulation and coding ή AMC), ο μηχανισμός Hybrid Automatic Repeat Request (HARQ) επανεκπομπής πακέτων και ο επονομαζόμενος Fast Channel Feedback (CQICH) ενσωματώθηκαν στο WiMAX ώστε να βελτιώσουν στοιχεία όπως απόσταση κάλυψης αλλά και γενικότερα να λύσουν αποτελεσματικότερα ορισμένα προβλήματα στην λειτουργία του WiMAX τα οποία υπήρχαν πριν την χρησιμοποίησή τους.

Η υποστήριξη για διαμορφώσεις τύπου QPSK, 16QAM και 64QAM είναι υποχρεωτική για το κανάλι DownLink. Στο κανάλι UpLink η 64QAM είναι προαιρετική. Για το κομμάτι της κωδικοποίησης υποστηρίζονται οι τεχνικές Convolutional Code (CC) και Convolutional Turbo Code (CTC) που προσφέρουν μεταβλητό ρυθμό κωδικοποίησης και δυνατότητα επαναληπτικότητας. Οι τεχνικές



Block Turbo Code και Low Density Parity Check Code (LDPC), υποστηρίζονται αλλά είναι προαιρετικές.

Ο παρακάτω πίνακας περιλαμβάνει λεπτομέρειες σχετικά με τις διαμορφώσεις και τους διαθέσιμους ρυθμούς κωδικοποίησης:

Πίνακας 3: Διαθέσιμες διαμορφώσεις και κωδικοποιήσεις

		Κανάλι DOWNLINK	Κανάλι UPLINK
Τύπος Διαμόρφωσης		QPSK, 16QAM, 64QAM	QPSK, 16QAM, 64QAM
Ρυθμός Κωδικοποίησης	CC	1/2, 2/3, 3/4, 5/6	1/2, 2/3, 5/6
	CTC	1/2, 2/3, 3/4, 5/6	1/2, 2/3, 5/6
	Επαναληπτικότητα	x2, x4, x6	x2, x4, x6

Επιπλέον σημαντικές πληροφορίες σε σχέση με τις διαθέσιμες διαμορφώσεις, όπως είναι το ελάχιστο απαραίτητο SNR(Signal to Noise Ratio,σχέση θορύβου-σήματος) για τα κανάλια του DownLink και του UpLink, αλλά και τα bits πληροφορίας ανά σύμβολο φαίνονται στον πίνακα 4.

Πίνακας 4: Επιπλέον στοιχεία διαμορφώσεων και κωδικοποιήσεων

Διαμόρφωση	Κωδικοποίηση	Bits Πληροφορίας/Σύμβολο/HZ	Απαραίτητο SNR(dB)
QPSK	1/2	1	9,4
	3/4	1,5	11,2
16-QAM	1/2	2	16,4
	3/4	3	18,2
64-QAM	2/3	4	22,7
	3/4	4,5	24,4

Ο μηχανισμός διαχείρισης και προγραμματισμού (base station scheduler) που υπάρχει στον σταθμό βάσης αποφασίζει για τον κατάλληλο ρυθμό αποστολής

δεδομένων και το προφίλ 'ριπής κίνησης'(burst profile), βασιζόμενος στο μέγεθος της προσωρινής μνήμης buffer του δέκτη, στις συνθήκες μετάδοσης του καναλιού και σε άλλα αντίστοιχα στοιχεία.

Υλοποιείται ένα ξεχωριστό κανάλι (Channel Quality Indicator (CQI) channel) το οποίο χρησιμοποιείται ώστε να μεταβιβάζονται στοιχεία των συνθηκών που επικρατούν ανά πάσα στιγμή στο κανάλι από τους τερματικούς σταθμούς προς τον μηχανισμό διαχείρισης του σταθμού βάσης. Επίσης στοιχεία σχετικά με την κατάσταση του καναλιού προσφέρει και ο μηχανισμός CQICH. Τέτοια είναι το στοιχείο CINR το οποίο δείχνει τον βαθμό παρεμβολών ανάμεσα στους υπό-μεταφορείς και τον θόρυβο του καναλιού και το είδος MIMO που θα χρησιμοποιηθεί.

Επίσης εφόσον η τεχνική μετάδοσης του σήματος είναι η TDD, μπορεί να προσφέρει και αυτή πληροφορίες σχετικές με την κατάσταση του καναλιού, καθώς εκμεταλλεύεται την δυνατότητα της διαρκής προσαρμοστικότητας των διασυνδέσεων και καναλιών που σημαίνει διαρκή ανταλλαγή σχετικών πληροφοριών.

Ο μηχανισμός HARQ του WiMAX ενεργοποιείται ώστε να προσφέρει άμεση ανταπόκριση σε περιπτώσεις χαμένων ή προβληματικών πακέτων και να βελτιώσει την μέγιστη απόσταση αποτελεσματικής εκπομπής σήματος μέσα σε ένα δικτυακό κελί. Ο μηχανισμός "Chase Combining" και ο προαιρετικός μηχανισμός "Incremental Redundancy" υποστηρίζονται ώστε να βελτιωθεί ακόμη περισσότερο η αξιοπιστία των επανεκπομπών πακέτων.

Επίσης ένα κανάλι σηματοδότησης- επιβεβαιώσεων ACK είναι δεσμευμένο στο κανάλι UpLink και προσφέρεται ακριβώς για μηνύματα τύπου HARQ ACK/NACK. Ο μηχανισμός HARQ υποστηρίζεται και σε πολύ-καναλικές μεταδόσεις. Το πρωτόκολλο "stop-and-wait ARQ" υποστηρίζει πολύ-καναλικές μεταδόσεις και ελαχιστοποιεί την μνήμη η οποία χρειάζεται για την λειτουργία του HARQ.

Το WiMAX υποστηρίζει τις ασύγχρονες μεταδόσεις. Η ασύγχρονη μετάδοση επιτρέπει ποικιλία καθυστέρησης ανάμεσα στις επανεκπομπές πακέτων κάτι το

οποίο προσφέρει μεγαλύτερη ευελιξία όμως δημιουργεί και περισσότερο overhead σε κάθε περίπτωση. Ο μηχανισμός HARQ συνδυασμένος με τα στοιχεία του προτύπου, CQICH και AMC προσφέρει προσαρμοστικότητα στις διαρκώς μεταβαλλόμενες συνθήκες των διασυνδέσεων που προκύπτουν στα ασύρματα περιβάλλοντα και σε ταχύτητες κίνησης σταθμών που ξεπερνάν τα 120 Km/h.

### **2.3.3 Υπο - Επίπεδο MAC**

Οι προδιαγραφές του επιπέδου MAC παρουσιάζουν σημαντικές αποκλίσεις από το πρότυπο 802.16-2004, ώστε να παρέχεται υποστήριξη για κινητικότητα κόμβων. Προσθέτετε υποστήριξη για handoff (δυνατότητα να συνδέονται οι εκάστοτε χρήστες από τον έναν σταθμό βάσης σε κάποιον άλλον γειτονικό την ώρα που κινούνται) και διαχείρισης της ενέργειας.

#### **2.3.3.1 Υποστήριξη QoS (Quality of Service)**

Το πρότυπο 802.16e καθορίζει νέους μηχανισμούς προγραμματισμού:

Την προδιαγραφή υπηρεσιών εκτεταμένης σταθμοσκόπησης σε πραγματικό χρόνο (ErtPS - Extended real-time Polling Service), η οποία βασίζεται σε δύο άλλες υπηρεσίες που ορίζονται στο 802.16-2004, στην υπηρεσία UGS (Unsolicited Grant Service) και την υπηρεσία σταθμοσκόπησης πραγματικού χρόνου rtPS (real time Polling Service). Η ErtPS είναι παρόμοια με την UGS στην παροχή unicast χορηγήσεων, οπότε μπορεί και αποθηκεύει την καθυστέρηση της διαδικασίας αυτής για να την χρησιμοποιήσει για την αίτηση Bandwidth που θα χρειαστεί κάποιος σταθμός. Ωστόσο, οι χορηγήσεις τύπου ErtPS και rtPS είναι δυναμικές σε αντίθεση με αυτές της UGS που συμβαίνουν σε καθορισμένες χρονικές στιγμές.

Η υπηρεσία ErtPS εισήχθηκε για την υποστήριξη ροών πραγματικού χρόνου (Real Time service flows) οι οποίες παράγουν περιοδικά πακέτα μεταβλητού μεγέθους δεδομένων. Έτσι, η ErtPS είναι ιδιαίτερα σημαντική για την υποστήριξη VoIP, δεδομένου ότι επιτρέπει καλύτερη διαχείριση κίνησης του δικτύου και βελτιώνει την καθυστέρηση (latency) και το jitter.

Ακολουθεί ένας ολοκληρωμένος πίνακας των προδιαγραφών υπηρεσιών και των χαρακτηριστικών τους:

Πίνακας 5: Προδιαγραφές υπηρεσιών QoS και χαρακτηριστικά αυτών

Κατηγορίες QoS	Κλάσεων	Κατάλληλες Εφαρμογές	Παράμετροι QoS
UGS (Unsolicited Service)	Grant	Εφαρμογές VoIP	<ul style="list-style-type: none"> <li>▪ Μέγιστος Ρυθμός Απόδοσης</li> <li>▪ Μέγιστη Επιτρεπτή Καθυστέρηση</li> <li>▪ Επιτρεπτό Jitter</li> </ul>
ErtPS (Extended Real-Time Polling Service)		Εφαρμογές VoIP με δυνατότητα ανίχνευσης Φωνής	<ul style="list-style-type: none"> <li>▪ Ελάχιστος Δεσμευμένος Ρυθμός Απόδοσης</li> <li>▪ Μέγιστος Ρυθμός Απόδοσης</li> <li>▪ Μέγιστη Επιτρεπτή Καθυστέρηση</li> <li>▪ Επιτρεπτό Jitter</li> <li>▪ Προτεραιότητα Κίνησης</li> </ul>
rtPS (Real-Time Polling Service)		Εφαρμογές Streaming Φωνής και Βίντεο	<ul style="list-style-type: none"> <li>▪ Ελάχιστος Δεσμευμένος Ρυθμός Απόδοσης</li> <li>▪ Μέγιστος Ρυθμός Απόδοσης</li> <li>▪ Μέγιστη Επιτρεπτή Καθυστέρηση</li> <li>▪ Προτεραιότητα Κίνησης</li> </ul>
nrtPS (non-Real-Time Polling Service)		Εφαρμογές FTP	<ul style="list-style-type: none"> <li>▪ Ελάχιστος Δεσμευμένος Ρυθμός Απόδοσης</li> <li>▪ Μέγιστος Ρυθμός Απόδοσης</li> <li>▪ Προτεραιότητα Κίνησης</li> </ul>
BE (Best Effort)		Περιήγηση Διαδικτύου, Μεταφορά Δεδομένων	<ul style="list-style-type: none"> <li>▪ Μέγιστος Ρυθμός Απόδοσης</li> <li>▪ Προτεραιότητα Κίνησης</li> </ul>

### 2.3.3.2 Υποστήριξη Handover/Handoff [1]

Το 802.16e περιλαμβάνει νέους μηχανισμούς αίτησης/χορήγησης επιπέδου MAC, ώστε να επιτύχει παρόμοια υποστήριξη κάλυψης φορητών κόμβων με αυτή που παρέχεται σε χρήστες κινητής τηλεφωνίας. Το 802.16e περιλαμβάνει γρήγορη εναλλαγή σταθμού βάσης και έναν Handoff μηχανισμό για την διαδικασία εναλλαγής ασύρματων κυψελών και τομέων. Στο 802.16e, η διαδικασία handoff μπορεί να ενεργοποιείται για δύο λόγους:

- ✓ Πρώτον λόγω της εξασθένησης του σήματος, το επίπεδο παρεμβολών, κ.λ.π. μέσα στην κυψέλη ή τον τομέα που βρίσκεται κάποιος.
- ✓ Δεύτερον εάν μια άλλη κυψέλη μπορεί να προσφέρει ένα υψηλότερο επίπεδο QoS στον φορητό σταθμό (MS).

Επιπλέον το 802.16e υποστηρίζει τις λειτουργίες macrodiversity handovers και intertechnology roaming (περιαγωγή). Η λειτουργία Macrodiversity handovers υποστηρίζει handoffs μεταξύ διαφορετικών μεγεθών κελιών, ενώ η intertechnology roaming διευθετεί handoffs των κινητών σταθμών από έναν σταθμό βάσης στο backbone του δικτύου ή σε κάποιο ενσύρματο δίκτυο, παρέχοντας μηχανισμούς ελέγχου ταυτότητας περιαγωγής.

## **3 Ο προσομοιωτής NS-2**

### **3.1 Εισαγωγή**

Ο NS-2 είναι ένας αντικειμενοστραφής προσομοιωτής, γραμμένος σε C++ με έναν μεταφραστή OTcl ως διερμηνευτή. Αποτελεί ένα πρόγραμμα που αναπτύχθηκε στα πλαίσια του έργου VINT ως μια αμοιβαία προσπάθεια μεταξύ των ιδρυμάτων UC Berkeley, USC/ISI, LBL και Xerox PARC και διανέμεται σαν open source project υπό την άδεια GNU. Ο προσομοιωτής υποστηρίζει μια ιεραρχία κλάσεων στην C++ (στον compiled κώδικα δηλαδή) καθώς και μια παρόμοια ιεραρχία στον διερμηνευτή του OTcl. Οι δύο αυτές ιεραρχίες σχετίζονται στενά η μια με την άλλη. Από πλευράς χρήστη υπάρχει πλήρης αντιστοίχιση μεταξύ μιας κλάσης στον διερμηνευτή και μιας κλάσης στον compiled κώδικα. Η ρίζα αυτής της ιεραρχίας είναι η κλάση TclObject. Οι χρήστες δημιουργούν αντικείμενα προσομοίωσης μέσω του διερμηνευτή, τα οποία αρχικοποιούνται μέσα σε αυτόν και παρακολουθούνται στενά από ένα αντίστοιχο αντικείμενο στην compiled ιεραρχία. Η ιεραρχία κλάσεων του διερμηνευτή δημιουργείται αυτόματα μέσω μεθόδων που είναι ορισμένες στην κλάση TclClass. Αντικείμενο που έχουν οριστεί από τον χρήστη αντανακλούνται μέσω μεθόδων που είναι ορισμένες στην κλάση TclObject. Υπάρχουν και άλλες ιεραρχίες τόσο στον compiled κώδικα όσο και στα OTcl σενάρια, οι οποίες όμως δεν έχουν κάποια ανταπόκριση στην κλάση TclObject.

### **3.2 Η χρησιμότητα δύο διαφορετικών γλωσσών [5],[7]**

Ο NS-2 χρησιμοποιεί δύο γλώσσες επειδή υπάρχουν δύο διαφορετικά πράγματα που πρέπει να κάνει ο προσομοιωτής. Από την μια οι αναλυτικές προσομοιώσεις πρωτοκόλλων απαιτούν μια προγραμματιστική γλώσσα ικανή να διαχειρίζεται bytes, κεφαλίδες πακέτων και να υλοποιεί αλγορίθμους που τρέχουν πάνω σε μεγάλους όγκους δεδομένων. Γι' αυτές τις διαδικασίες, η ταχύτητα εκτέλεσης είναι σημαντική και ο χρόνος διεκπεραίωσης (εκτέλεση προσομοίωσης, εύρεση bug, διόρθωση bug, recompile, επανεκτέλεση) είναι λιγότερο σημαντικός.

Από την άλλη, ένα μεγάλο κομμάτι της έρευνας δικτύων περιέχει ποικίλες παραμέτρους ή διαμορφώσεις ή την γρήγορη εξερεύνηση ενός αριθμού σεναρίων. Σε αυτές τις περιπτώσεις ο χρόνος επανάληψης (αλλαγή του μοντέλου και επανάληψη εκτέλεσης του σεναρίου) είναι πιο σημαντικός. Εφόσον η διαμόρφωση τρέχει μόνο μια φορά (στην αρχή της προσομοίωσης, ο χρόνος εκτέλεσης αυτού του τμήματος δεν είναι και τόσο σημαντικός.

Ο NS-2 ανταποκρίνεται και στις δύο αυτές ανάγκες χρησιμοποιώντας και την C++ αλλά και την OTcl. Η C++ είναι γρήγορη στην εκτέλεση αλλά αργή σε αλλαγές, κάτι που την κάνει ιδεατή για την αναλυτική υλοποίηση ενός πρωτοκόλλου. Η OTcl τρέχει πολύ πιο αργά αλλά μπορεί να αλλάξει πολύ εύκολα (και διαδραστικά), κάτι που την καθιστά ιδεατή για διαμορφώσεις προσομοίωσης. Ο NS-2 παρέχει έναν σύνδεσμο μεταξύ των δύο γλωσσών, την Tclcl, που επιτρέπει την δημιουργία αντικειμένων και μεταβλητών που θα είναι προσβάσιμες και από τις δύο γλώσσες.

### **3.3 Επισκόπηση του προσομοιωτή [5],[7]**

Ο προσομοιωτής αρχικοποιείται μέσω του διερμηνευτή OTcl με την δημιουργία ενός αντικειμένου τύπου προσομοιωτή σε συνδυασμό με εντολές που ορίζουν την τοπολογία και τις ρυθμίσεις ενός δικτύου. Κατά την δημιουργία ενός αντικειμένου προσομοίωσης, γίνεται μια αρχικοποίηση η οποία εκτελεί τις παρακάτω διαδικασίες:

- Αρχικοποίηση της δομής των πακέτων
- Δημιουργία ενός αντικειμένου τύπου χρονοπρογραμματιστή
- Δημιουργία ενός κενού πράκτορα (null agent)

Τα πακέτα διακινούνται μέσω χειριστών γεγονότων, παραδίδονται σε δεδομένες χρονικές στιγμές και κάθε ενέργεια που εκτελείται σε ένα πακέτο θεωρείται ένα γεγονός. Κατά την αρχικοποίηση της προσομοίωσης ορίζονται διάφορες ρυθμίσεις για τα πακέτα, όπως για παράδειγμα το μέγεθός τους, ο χρόνος αναμετάδοσης κ.α.

Ο μηχανισμός χρονοπρογραμματισμού είναι ένα υπόστρωμα διαχείρισης το οποίο εκκινείται από γεγονότα και εκτελεί κάθε φορά το προγραμματισμένο γεγονός, συνεχίζοντας με το επόμενο μετά την ολοκλήρωση του τρέχοντος. Ένα γεγονός συνδέεται με την χρονική στιγμή που επιλέγεται να εκτελεστεί καθώς και με την λειτουργία χειρισμού του. Όταν δύο ή περισσότερα γεγονότα συμβούν ταυτόχρονα, η εκτέλεσή τους γίνεται βάση της σειράς χρονοπρογραμματισμού τους.

Ο κενός πράκτορας που δημιουργείται λειτουργεί σαν ένας υποδοχέας πακέτων που είτε παραδόθηκαν επιτυχώς είτε απορρίφθηκαν. Όλα τα πακέτα που απορρίπτονται ή πακέτα που δεν λαμβάνονται υπ' όψη ή και πακέτα που παραδόθηκαν επιτυχώς στέλνονται σε αυτόν τον πράκτορα.

Ο προσομοιωτής μπορεί να λειτουργήσει σε δύο καταστάσεις. Είτε διαδραστικά με τον χρήστη γράφοντας εντολές στο παράθυρο της κονσόλας που θα ορίζουν ρυθμίσεις για το σενάριο είτε δίνοντας σαν είσοδο ένα αρχείο το οποίο περιέχει όλες τις εντολές που περιγράφουν ένα σενάριο. Συνήθως ένα αρχείο με εντολές που καθορίζει ένα σενάριο περιέχει και ένα σύνολο εντολών που έχουν ως αποτέλεσμα την δημιουργία κάποιων αρχείων μετά την εκτέλεση του σεναρίου. Συνήθως τα αρχεία αυτά είναι αρχεία \*.tr (Trace Files) και \*.nam (NAM Traces). Τα μεν αρχεία \*.tr αποτελούν αρχεία τα οποία καταγράφουν όλες τις ενέργειες που έλαβαν χώρα κατά την εκτέλεση της προσομοίωσης. Περιέχει δηλαδή πληροφορίες όπως η μετάδοση και λήψη πακέτων, το είδος των πακέτων επικοινωνίας, το γεγονός απόρριψης πακέτων και την αιτία του, πληροφορίες που αφορούν κατανάλωση ενέργειας κ.α.. Τα \*.nam αρχεία περιέχουν πληροφορίες που αφορούν την τοπολογία του δικτύου, όπως οι τοποθεσίες των κόμβων, οι συνδέσεις, τα πακέτα που ανταλλάχτηκαν και τότε και χρησιμοποιείται για την οπτική αναπαράσταση του δικτύου μέσω του εργαλείου NAM.



## 4 NIST Module

### 4.1 Εισαγωγή

Η υλοποίηση του συγκεκριμένου module βασίζεται στο πρότυπο IEEE 802.16 (802.16-2004) καθώς και στην επέκταση του 802.16<sup>e</sup>-2005. Παρακάτω παραθέτονται οι δυνατότητες καθώς και οι ελλείψεις του module.

Πίνακας 6: Δυνατότητες και ελλείψεις του NIST module

<b>Δυνατότητες</b>	<b>Ελλείψεις</b>
- Υλοποίηση της προδιαγραφής OFDM για το φυσικό επίπεδο	- Υλοποίηση της προδιαγραφής OFDMA για το φυσικό επίπεδο
- Time Division duplexing (TDD)	- Frequency Division duplexing (FDD)
- Μηνύματα διαχείρισης για την είσοδο σε δίκτυο (χωρίς αυθεντικοποίηση)	- ARQ (Automatic Repeat Request- Αυτόματη επαναλαμβανόμενη αίτηση)
- Εξ' ορισμού χρονοπρογραμματιστής που κατανέμει πόρους στους εγγεγραμμένους κινητούς σταθμούς χρησιμοποιώντας round robin σύμφωνα με το εύρος ζώνης που αιτούνται.	- Ροή υπηρεσιών και χρονοπρογραμματισμός QoS
- Πρόσθετα του 802.16e για την υποστήριξη σαρώσεων και μεταβιβάσεων	- Περιοδική κύμανση και προσαρμογές ενέργειας
- Κατακερματισμός και επανασύνθεση των πλαισίων (frames)	- Πακετοποίηση
	- Διόρθωση Σφαλμάτων

Αξίζει να σημειωθεί ότι πολλά συστατικά δεν υπάρχουν στο πρότυπο. Γι' αυτό το μοντέλο υλοποιεί μιας λύση η οποία μπορεί να μην ταιριάζει απαραίτητα στις ανάγκες κάθε χρήστη. Γι' αυτό ο σχεδιασμός του module είναι τέτοιος ώστε να είναι όσο το δυνατόν πιο επεκτάσιμο.

### 4.2 Documentation [11]

#### 4.2.1 Υπόστρωμα σύγκλισης πακέτων

Το υπόστρωμα σύγκλισης πακέτων κάθετα πάνω από το επίπεδο MAC και εκτελεί τις παρακάτω λειτουργίες:

- Λαμβάνει PDUs υψηλότερου επιπέδου
- Ταξινομεί
- Μεταφέρει τα PDUs του CS στο SAP του MAC
- Λαμβάνει τα PDUs από τους συνδέσμους

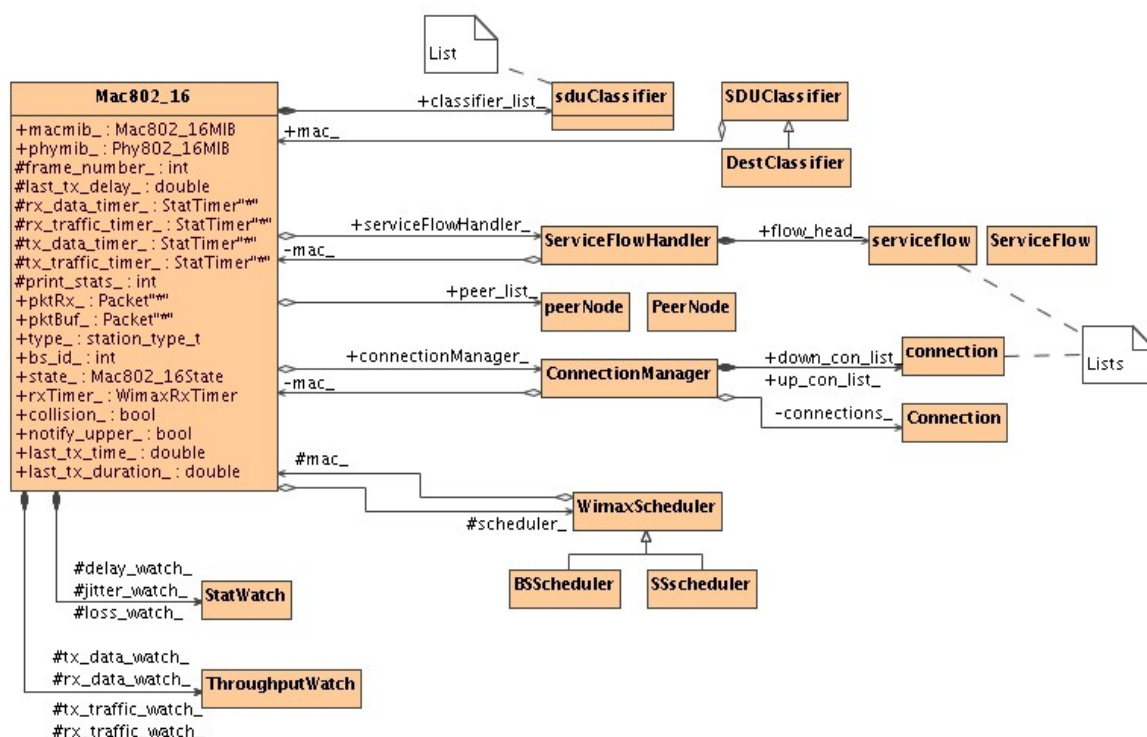
Στην τρέχουσα υλοποίηση το επίπεδο πακέτου CS μόνο ταξινομεί. Η μέθοδος που χρησιμοποιείται για την ταξινόμηση πακέτων εξαρτάται από την υλοποίηση. Ίσως βέβαια να είναι βέλτιστο να υλοποιηθούν πολλές λύσεις ώστε να βρεθεί η βέλτιστη δυνατή. Το μοντέλο είναι έτσι κατασκευασμένο ώστε να υποστηρίζει ταξινομητές φτιαγμένους από τον χρήστη.

#### **4.2.2 Υποστρώμα MAC**

Η ενότητα αυτή περιλαμβάνει όλες τις πληροφορίες για το υλοποιημένο υπόστρωμα MAC το οποίο υποστηρίζει PMP.

##### **4.2.2.1 Δομή του προσθέτου MAC**

Η Mac802\_16 είναι μια αφηρημένη υποκλάση της Mac, η οποία περιέχει τα κύρια στοιχεία των BS και MS. Αποτελεί την διασύνδεση με τα άλλα στρώματα για την αποστολή και λήψη πακέτων. Στην εικόνα 9 φαίνεται η κλάση και οι σχέσεις της με άλλες ενότητες.



Εικόνα 9: Διάγραμμα κλάσεων του MAC επιπέδου (NIST module)

Μια κλάση MAC έχει μια λίστα ταξινομητών πακέτων (SDUClassifier) η οποία καταγράφει κάθε εξερχόμενο πακέτο με τον κατάλληλο αναγνωριστή σύνδεσης. Με την χρήση της TCL, ο χρήστης καθορίζει την λίστα των ταξινομητών που θα χρησιμοποιηθούν. Η τρέχουσα υλοποίηση χρησιμοποιεί την IP του προορισμού σαν στοιχεία ταξινόμησης.

Η κλάση ServiceFlowHandler είναι υπεύθυνη για τον έλεγχο ροής των αιτημάτων και απαντήσεων. Επιπλέον αποθηκεύει την λίστα των ροών ενός κόμβου.

Ένας σταθμός συνδρομητής συνδέεται πάνω στον σταθμό βάσης ενώ ο σταθμός βάσης μπορεί να συνδεθεί σε πολλούς σταθμούς συνδρομητές. Η κλάση PeerNode περιέχει δεδομένα αναφορικά με τον σταθμό, όπως οι συνδέσεις του και η κατάσταση του. Η συνδέσεις (class Connections) είναι προσβάσιμες επίσης μέσω της κλάσης ConnectionManager, η οποία περιέχει την λίστα με τις εισερχόμενες και εξερχόμενες συνδέσεις.

Η αφηρημένη κλάση WimaxScheduler χρησιμοποιείται για την δημιουργία μιας διεπαφής με το MAC. Κατά κύριο λόγο υπάρχει ένας χρονοπρογραμματιστής για

τον σταθμό βάσης και ένας για τον κόμβο συνδρομητή και εφόσον ο χρονοπρογραμματιστής καθορίζεται με την χρήση TCL, είναι εύκολο να υλοποιηθεί η αφηρημένη κλάση και να αλλαχθεί.

Τέλος η κλάση MAC υπολογίζει στατιστικά μέσω των αντικειμένων StatWatch και ThroughputWatch για πληροφορίες πακέτων και ροής δεδομένων, τα οποία χρησιμοποιούνται για να πυροδοτήσουν διαδικασίες και επιπλέον μπορούν να τυπωθούν κατά την διάρκεια της προσομοίωσης.

Μιας και ο σταθμός βάσης και ο σταθμός συνδρομητής έχουν διαφορετικές καταστάσεις μηχανής έχουν υλοποιηθεί δύο διαφορετικές κλάσεις για τον σταθμό βάσης και τον σταθμό συνδρομητή με ονόματα Mac802\_16BS και Mac802\_16SS αντίστοιχα.

#### **4.2.3 Διευθυνσιοδότηση και σύνδεση**

Κάθε MAC έχει μια ξεχωριστή διεύθυνση η οποία καταχωρείται σαν ακέραιος αριθμός. Το μοντέλο επίσης καθορίζει αναγνωριστές σύνδεσης σαν ακέραιους των 16 bit.

Οι παρακάτω συνδέσεις δημιουργούνται κατά την αρχικοποίηση ενός κόμβου σταθμού:

- Αρχική ρύθμιση (εισερχόμενη και εξερχόμενη)
- Padding (εισερχόμενη και εξερχόμενη)
- Broadcast (εξερχόμενη)
- Adaptive Antenna System (AAS) (εξερχόμενη, αλλά δεν χρησιμοποιείται)

Οι παρακάτω συνδέσεις δημιουργούνται κατά την αρχικοποίηση ενός σταθμού συνδρομητή:

- Αρχική ρύθμιση (εισερχόμενη και εξερχόμενη)
- Padding (εισερχόμενη και εξερχόμενη)
- Broadcast (εισερχόμενη)

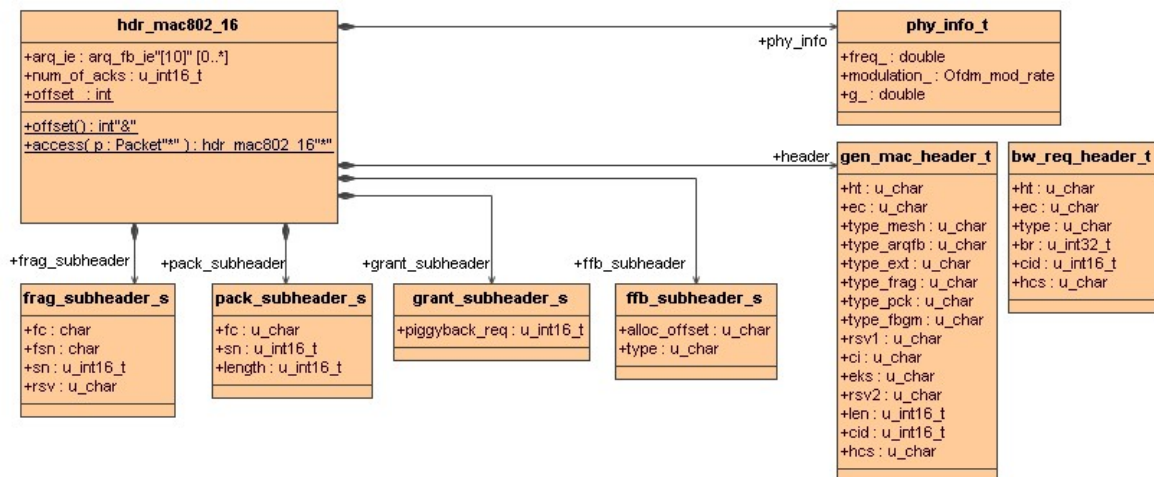
Επιπλέον κατά την καταχώρηση στο δίκτυο καθορίζονται οι παρακάτω συνδέσεις και τα CIDs:

- Basic CID (εισερχόμενη και εξερχόμενη)
- Primary CID (εισερχόμενη και εξερχόμενη)
- Secondary CID (εισερχόμενη και εξερχόμενη)
- Data CIDs. Προς το παρόν το μοντέλο υποστηρίζει μόνο μια σύνδεση δεδομένων

#### 4.2.4 Μορφή των PDUs της MAC

Οι σχεδιαστές του μοντέλου έκριναν απαραίτητη την δημιουργία ενός νέου header για να κρατάει τα πακέτα για το IEEE 802\_16.

Στην εικόνα 10 βλέπουμε το διάγραμμα κλάσεων της κλάσης `hdr_802_16` που είναι η νέα κλάση που αναφέραμε.



Εικόνα 10: Διάγραμμα κλάσεων της κλάσης "hdr\_802\_16" (NIST module)

Το header αποτελείται από τα παρακάτω τρία κύρια στοιχεία:

- Ένα εικονικό header φυσικού επιπέδου τύπου `phy_info_t`, το οποίο χρησιμοποιείται για την μεταφορά πληροφοριών όπως η συχνότητα, η διαμόρφωση (modulation) και το κυκλικό πρόθεμα (cyclic prefix).

- Ένα γενικό MAC header τύπου `gen_mac_header_t` που περιέχει τις γενικές πληροφορίες σχετικά με το MAC επίπεδο. Η συγκεκριμένη δομή μπορεί να μετατραπεί σε `bw_req_header_t` όταν το πακέτο αποτελεί αίτηση για εύρος ζώνης.
- Δομές που κρατάνε δεδομένα για διάφορα υπο-header, οι οποίες είναι παρούσες σε όλα τα πακέτα ενώ η μεταβλητή `type` του κυρίως header καθορίζει αν η εγγραφή είναι σωστή ή όχι.

Επιπλέον όταν είναι ενεργοποιημένο το ARQ, το header περιέχει και πληροφορίες που χρησιμοποιούνται για ανατροφοδότηση.

Στον πίνακα 7 μπορούμε να δούμε τα πακέτα που έχουν υλοποιηθεί στο μοντέλο μέχρι στιγμής.

Πίνακας 7: Υποστηριζόμενοι τύποι πακέτων (NIST module)

Κατηγορία	Οριζόμενα μηνύματα
Συγχρονισμός	DL-MAP / DCD UL-MAP / UCD RNG-REQ/RSP REG-REQ/RSP
Ροές υπηρεσιών	DSA-REQ/PSR/ACK MOB_NBR_ADV MOB_SCN-REQ/RSP
Κινητικότητα	MOB_BSHO-REQ/RSP MON_SSHO-REQ MOB-HO-IND MOB_SCN-REP MOB_ASC-REP

#### 4.2.5 Κατασκευή και μετάδοση των PDU του επιπέδου MAC

Η κατασκευή και μετάδοση των πακέτων χωρίζεται στα τρία παρακάτω βήματα:

- Λήψη ενός εξερχομένου πακέτου από το ανώτερο επίπεδο: Το επίπεδο MAC αναζητά το σωστό CID στους ταξινομητές. Αν αυτό βρεθεί τότε προσαρτά στο πακέτο ένα προεπιλεγμένο header και το τοποθετεί στην ουρά της σύνδεσης.

- Χρονοπρογραμματισμός: Για κάθε frame οι χρονοπρογραμματιστές ψάχνουν στην λίστα των συνδέσεων για να βρουν το πακέτο που θα εκπέμψουν.
- Μετάδοση: Δύο timers εκπέμπουν τα πακέτα που είναι κρατημένα στις ουρές.

#### **4.2.5.1 Κατακερματισμός**

Ο κατακερματισμός μπορεί να ενεργοποιηθεί ή απενεργοποιηθεί σε κάθε σύνδεση, ενώ η κατ' εξοχήν επιλογή είναι να είναι ενεργοποιημένος.

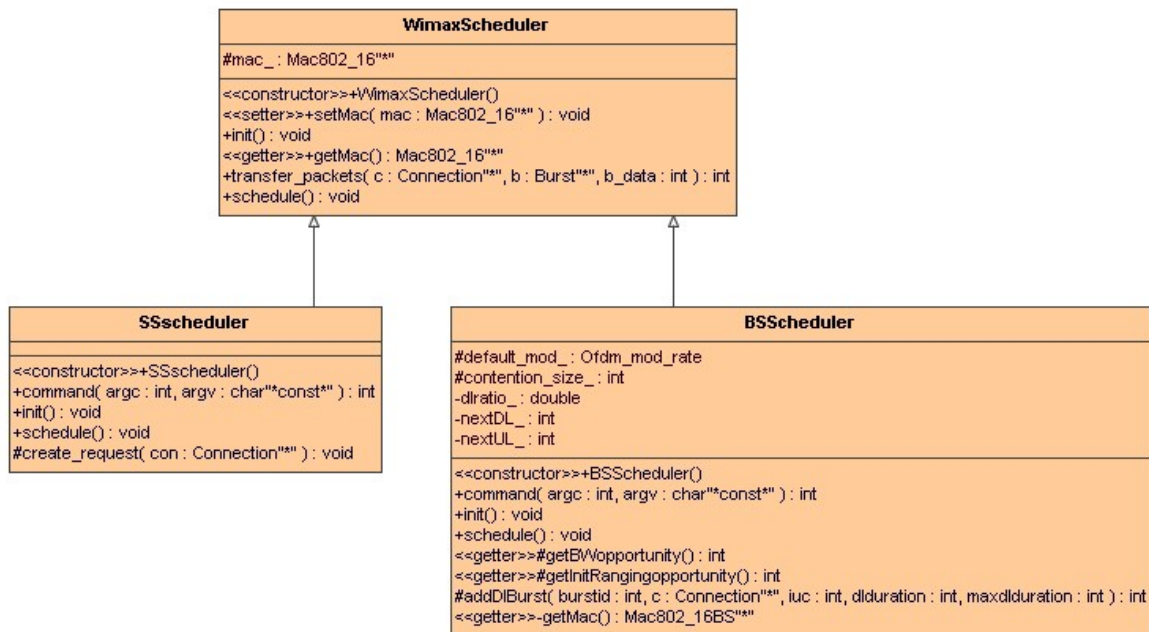
Κατά τον χρονοπρογραμματισμό πακέτων για μετάδοση, ο χρονοπρογραμματιστής ελέγχει εάν ο κατακερματισμός είναι ενεργός για την συγκεκριμένη σύνδεση και αν είναι σπάει τα πακέτα. Το πλαίσιο κατακερματισμού αποθηκεύεται στην σύνδεση (class Connection).

#### **4.2.6 Υπηρεσίες Χρονοπρογραμματισμού**

Η δομή των κλάσεων επιτρέπει την χρήση διάφορων υπηρεσιών δεδομένων όπως οι UGS, rtGS, nrTPS και Best Effort. Ο χρονοπρογραμματισμός των πακέτων εκτελείται από έναν χρονοπρογραμματιστή ο οποίος αλληλεπιδρά με το επίπεδο MAC μέσω ενός API το οποίο δέχεται προσαρμοσμένες υλοποιήσεις.

##### **4.2.6.1 Χρονοπρογραμματιστές**

Όπως είναι λογικό κάθε είδος κόμβου χρειάζεται και διαφορετικό χρονοπρογραμματιστή πακέτων. Στο IEEE 802\_16, ο σταθμός βάσης ελέγχει τις εκχωρήσεις εύρους ζώνης και υπάρχουν πάρα πολλές υλοποιήσεις. Το μοντέλο περιέχει μια αφηρημένη κλάση, την WimaxScheduler, η οποία δημιουργήθηκε για να είναι εύκολη η χρήση διαφορετικών χρονοπρογραμματιστών πακέτων. Όπως φαίνεται στην εικόνα 11 υπάρχουν ήδη δύο υλοποιήσεις μια για τους κόμβους σταθμούς και μια για τους κόμβους πελάτες, οι οποίοι μπορούν εύκολα να αντικατασταθούν με την χρήση της TCL.



Εικόνα 11: Διάγραμμα κλάσεων χρονοδρομολόγησης (NIST module)

Κατά την δημιουργία ενός νέου χρονοπρογραμματιστή, οι παρακάτω μέθοδοι πρέπει να υλοποιηθούν:

- `init()` : αρχικοποιεί τον χρονοπρογραμματιστή
- `process (Packet *)` : Η μέθοδος αυτή χρησιμοποιείται για την επεξεργασία των ληφθέντων πακέτων από τον χρονοπρογραμματιστή.
- `start_ulsubframe()` : Περιέχει τον κώδικα που θα εκτελεστεί στην αρχή ενός νέου υπο-πλασίου σύνδεσης εκπομπής.
- `start_dlsubframe()` : Περιέχει τον κώδικα που θα εκτελεστεί στην αρχή ενός νέου υπο-πλασίου σύνδεσης λήψης.

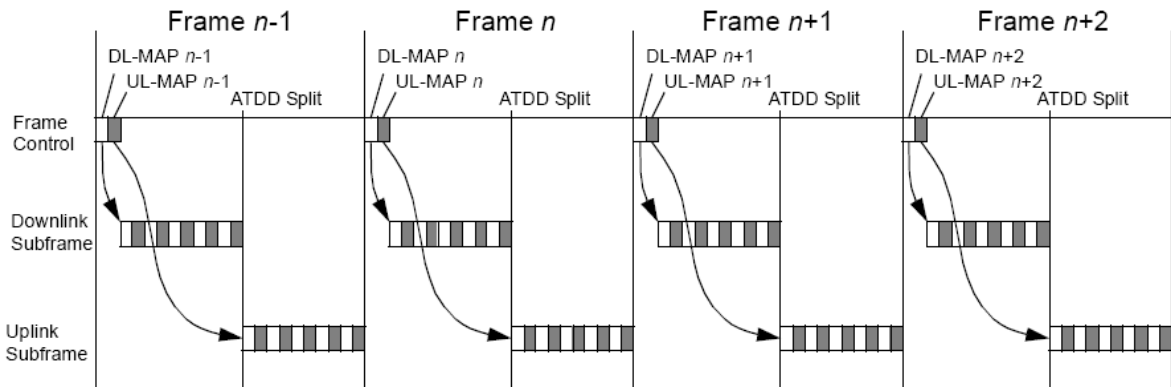
#### 4.2.7 Υποστήριξη του PHY στο επίπεδο MAC

Το μοντέλο προς το παρόν υποστηρίζει το TDD. Σε αυτή τη λειτουργία, οι μεταδόσεις της σύνδεσης εκπομπής γίνονται μετά την σύνδεση λήψης για κάθε πλαίσιο.

Τα μηνύματα DL\_MAP και UL\_MAP που στέλνει κάθε πλαίσιο καθορίζουν την κατανομή καταιγισμού και τις δυνατότητες μετάδοσης για κάθε εξυπηρετητή. Οι



πληροφορίες που περιέχονται στο UL\_MAP ανήκουν στο ίδιο πλαίσιο όπως φαίνεται στην εικόνα 12.

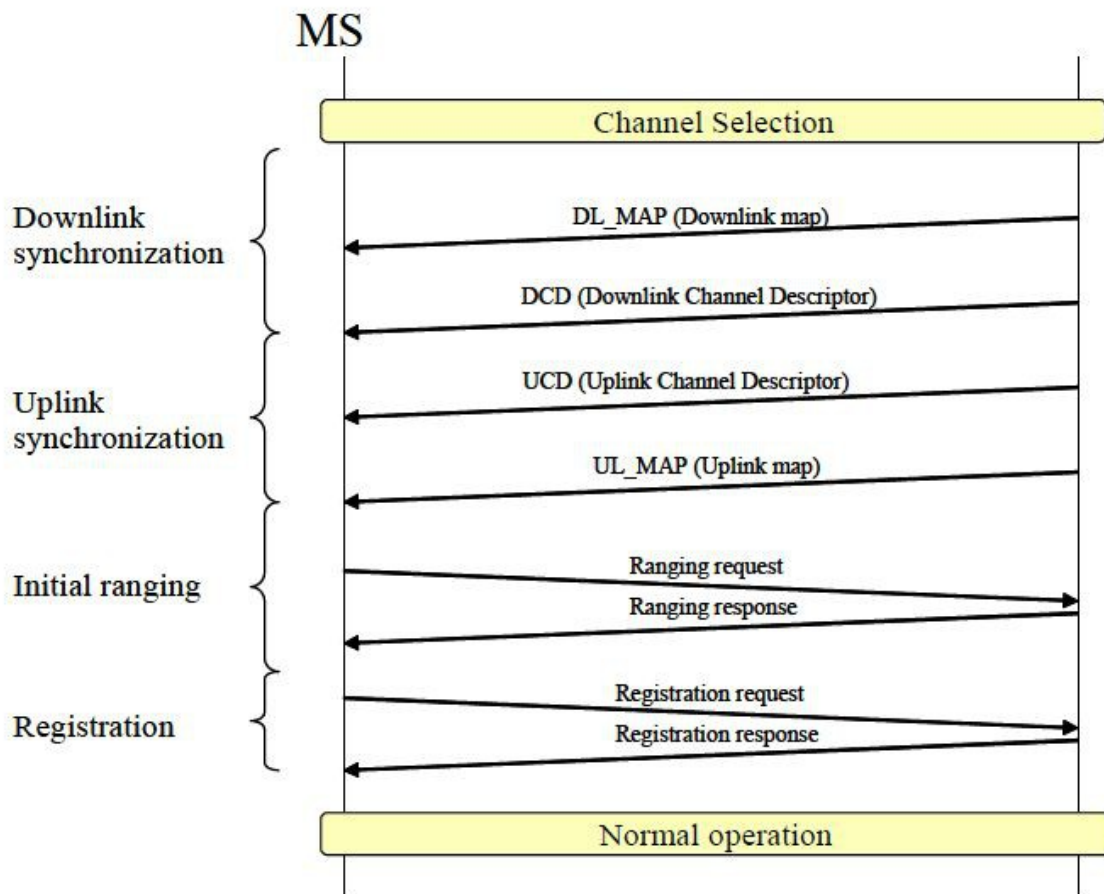


Εικόνα 12: Χρονική σχέση του DL\_MAP και του UL\_MAP (NIST module)

#### 4.2.8 Είσοδος σε δίκτυο και αρχικοποίηση

Όταν ένας σταθμός συνδρομητής θέλει να συνδεθεί σε ένα δίκτυο πρέπει να εκτελέσει μια διαδικασία εισόδου σε δίκτυο. Όπως φαίνεται στην εικόνα 13 το μοντέλο υλοποιεί τα παρακάτω συστατικά σε κάθε είσοδο σε δίκτυο:

- Σάρωση του καναλιού σύνδεσης εκπομπής
- Απόκτηση παραμέτρων μετάδοσης
- Αρχική τηλεμετρία
- Εγγραφή



Εικόνα 13: Είσοδος σε δίκτυο (NIST module)

Οι παρακάτω παράμετροι μπορούν να οριστούν:

- Χρονόμετρα που θα εκτελούν σάρωση του καναλιού
- Η συχνότητα των μηνυμάτων DCD/UCD
- Παράμετροι για την αρχική τηλεμετρία
- Εκχώρηση καναλιού

#### 4.2.9 Τηλεμετρία

Η τηλεμετρία είναι ένας μηχανισμός που επιτρέπει στην σταθμό συνδρομητή να διατηρήσει μια καλή ποιότητα σύνδεσης ρυθμίζοντας την ισχύ εκπομπής του και την διαμόρφωσή του.

Κατά την αρχική τηλεμετρία, οι σταθμοί συνδρομητές χρησιμοποιούν το κατ' εξοχήν προφίλ DIUC για να εκπέμπουν, κάτι που επιτρέπει την προσομοίωση του να εκπέμπουν οι κόμβοι σε διαφορετικούς ρυθμούς.

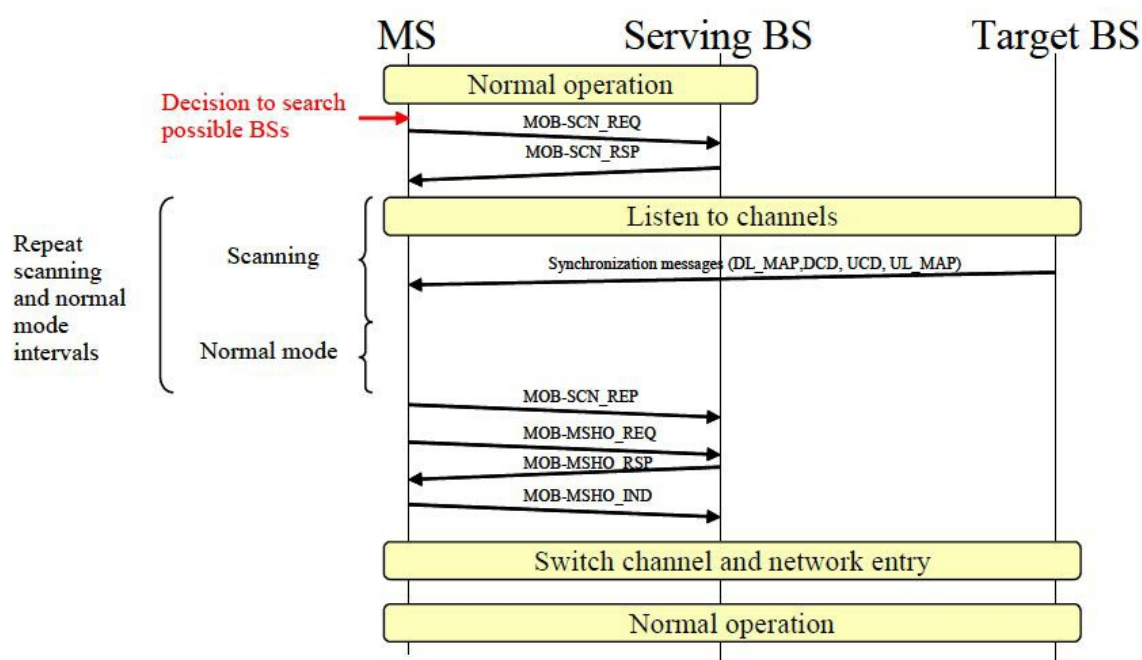
Προς το παρόν δεν υπάρχει κάποιος αλγόριθμος υλοποιημένος που να κάνει χρήση των δυνατοτήτων της τηλεμετρίας και χρησιμοποιείται μόνο για να προσθέσει επιπλέον καθυστέρηση σε μια καταχώρηση δικτύου. Επιπλέον δεν είναι υλοποιημένη η περιοδική τηλεμετρία και τα αιτήματα CDMA.

#### **4.2.10 Διαδικασίες παράδοσης του επιπέδου MAC**

Το μοντέλο υποστηρίζει κινητικότητα 2<sup>ου</sup> επιπέδου. Ανάλογα με την διαμόρφωση, ένας κινούμενος σταθμός συνδρομητής μπορεί να εκτελέσει σάρωση και παράδοση με τον σταθμό βάσης. Σε αυτή τη ενότητα παρουσιάζεται η παράμετροι της διαμόρφωσης που επηρεάζουν την ικανότητα παράδοσης.

##### **4.2.10.1 Σάρωση**

Όταν η ποιότητα την σύνδεσης επιδεινώνεται ο κινούμενος σταθμός συνδρομητής μπορεί να στείλει ένα αίτημα MOB-SCN\_REQ στον σταθμό βάσης υπηρεσιών για να ζητήσει μια σάρωση του δικτύου για να βρει τους υπόλοιπους κόμβους εξυπηρετητές στην εμβέλειά του. Στην εικόνα 14 φαίνεται η αλληλουχία μηνυμάτων που ανταλλάσσονται κατά την διαδικασία σάρωσης.



Εικόνα 14: Διαδικασία σάρωσης (NIST module)

Για να πυροδοτήσει την αποστολή ενός MOB-SCN\_REQ, ο κινητός σταθμός συνδρομητής παρακολουθεί το επίπεδο του σήματος στα εισερχόμενα πακέτα και όταν αυτό φτάσει ένα κατώφλι, το μήνυμα αποστέλλεται. Σαν προεπιλογή η σάρωση δεν είναι ενεργή. Για να ενεργοποιηθεί, πρέπει να αλλάξουμε την τιμή της μεταβλητής *lgd\_factor\_* της MIB σε μια τιμή μεγαλύτερη του 1. Όσο μεγαλύτερη η τιμή τόσο πιο σύντομα θα πυροδοτηθεί η σάρωση.

Κατά τη διάρκεια της σάρωσης, ο κινητός σταθμός συνδρομητής συλλέγει τιμές RSSI από τα εισερχόμενα πακέτα, οι οποίες χρησιμοποιούνται από τον σταθμό βάσης υπηρεσιών για να επιλεγεί ο καλύτερος στόχος σταθμός βάσης.

Αφού ο κινητός σταθμός συνδρομητής λάβει κάποια ένδειξη από τον επιλεγμένο σταθμό βάσης, περιμένει μερικά frames για να δείξει την πρόθεσή του να εκτελέσει παράδοση. Η καθυστέρηση αυτή έχει σκοπό να επιτρέψει την ανταλλαγή της κίνησης, που έχει αποθηκευτεί στους buffers κατά την διαδικασία σάρωσης, πριν γίνει η αλλαγή του κόμβου εξυπηρετητή.

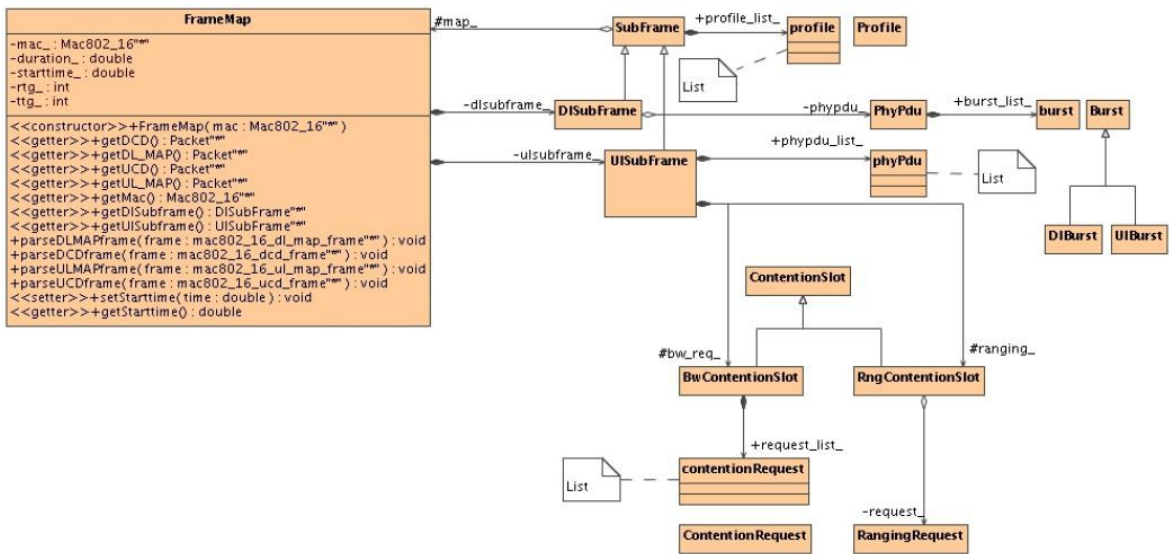
Στο μοντέλο έχουν υλοποιηθεί διάφοροι τρόποι σάρωσης:

- Στην σάρωση χωρίς συσχέτιση, ο κινητός σταθμός συνδρομητή προσπαθεί να αναγνωρίσει και να συγχρονιστεί με έναν ή περισσότερους κόμβους εξυπηρετητές. Επιπλέον εκτιμά την ποιότητα του σήματος.
- Στο επίπεδο συσχέτισης 0, ο στόχος σταθμός βάσης δεν έχει πληροφορίες για τον κινητό κόμβο πελάτη που σαρώνει και παρέχει μόνο εκχωρήσεις παράδοσης με βάση ισχυρισμούς. Μόλις ένας κινητός σταθμός συνδρομητής αποστείλει μια αίτηση παράδοσης, περιμένει απάντηση από τον σταθμό βάσης με ένα εξ' ορισμού όριο χρόνου των 50ms για την ακύρωση της αίτησης.
- Στο επίπεδο συσχέτισης 1, ο σταθμός βάσης που παρέχει υπηρεσίες, διαπραγματεύεται με τον στόχο σταθμό βάσης μόλις ο κινητός σταθμός συνδρομητής βρει μια αφιερωμένη περιοχή παράδοσης. Μόλις ένας κινητός σταθμός συνδρομητής αποστείλει μια αίτηση παράδοσης, περιμένει απάντηση από τον σταθμό βάσης με ένα εξ' ορισμού όριο χρόνου των 50ms για την ακύρωση της αίτησης.

Το επίπεδο συσχέτισης 2, δεν υποστηρίζεται προς το παρόν.

Για να επιτραπούν αυτοί οι διαφορετικοί τρόπου σάρωσης και για να γίνουν γρήγορες παραδώσεις, η κλάση WiMAXCtrlAgent απαιτείται, η οποία είναι μια κλάση τύπου Agent που εκτελεί τρεις διαδικασίες. Η πρώτη είναι να ανταλλάσει πληροφορίες DCD/UCD μεταξύ των γειτονικών κόμβων εξυπηρετητών. Η δεύτερη είναι να πυροδοτεί την αποστολή των NBR-ADV μηνυμάτων στους κινητούς κόμβους πελάτες. Η τρίτη είναι να συγχρονίζει τον σταθμό βάσης υπηρεσιών με τον στόχο σταθμό βάσης όταν εκτελείται σάρωση επιπέδου 1 ή 2. Τα μηνύματα αυτά ανταλλάσσονται μέσω καλωδίων με την χρήση κοινών IP πακέτων.

#### **4.2.11 Η δομή του frame**



Εικόνα 15: Διάγραμμα κλάσεων του frame (NIST module)

Ο σχεδιασμός που χρησιμοποιήθηκε για να αναπαρασταθεί ένα frame μοιάζει πολύ με την δομή που έχει οριστεί στο IEEE 802.16 για το TDD. Ένα frame αποτελείται από ένα υπο-πλαίσιο uplink και ένα downlink, τα οποία με την σειρά τους διαχωρίζονται σε PHY PDU διαστήματα. Σε κάθε ένα από αυτά τα διαστήματα, το εύρος ζώνης εκχωρείται σε ριπές για τους διάφορους σταθμούς. Κάθε μια από αυτές τις ριπές μπορεί να έχει διαφορετική διαμόρφωση και συχνότητα τα οποία ονοματίζονται προφίλ.

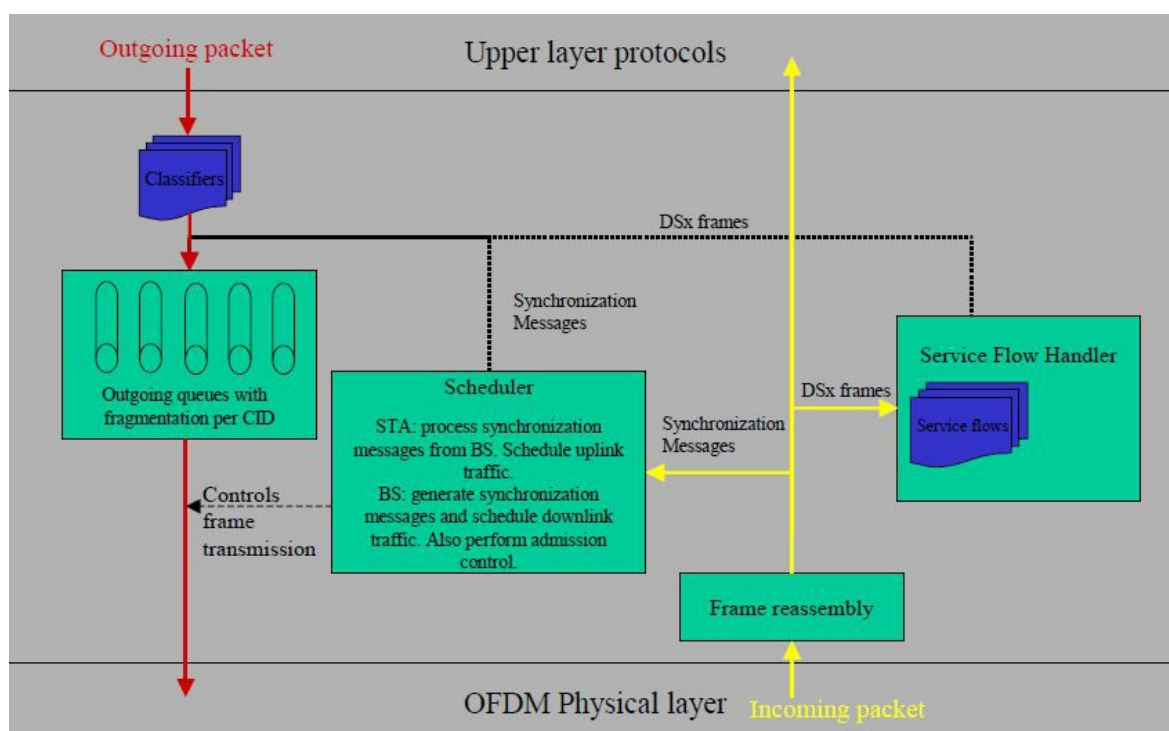
Κανονικά ο σταθμός βάσης αναθέτει εύρος ζώνης για έναν σταθμό ώστε να εκπέμψει τα δεδομένα του. Σε μερικές περιπτώσεις, κυρίως κατά την αρχική ρύθμιση και στις αιτήσεις εύρους ζώνης, οι συμπληρωματικές υπηρεσίες πρέπει να ανταγωνιστούν η μια την άλλη για να αποκτήσουν πρόσβαση στο μέσο. Αυτά τα διαστήματα παρουσιάζονται μόνο στο uplink καθώς ο σταθμός βάσης έχει πλήρη έλεγχο στην κίνηση του downlink.

Η κλάση FrameMap επιπλέον περιέχει μεθόδους για την εξαγωγή και ανάγνωση των μηνυμάτων ελέγχου. Στο σταθμό βάσης ο χρονοπρογραμματιστής δημιουργεί την δομή του χάρτη σύμφωνα με έναν αλγόριθμο ανάθεσης, και στην συνέχεια καλεί τις μεθόδους getDL\_MAP, getUL\_MAP, getDCD και getUCD, για να λάβει τα πακέτα που περιέχουν τις απαραίτητες πληροφορίες που πρέπει να σταλούν στις

συμπληρωματικές υπηρεσίες. Στις συμπληρωματικές υπηρεσίες, ο χρονοπρογραμματιστής καλεί τις αντίστροφες συναρτήσεις, parsed\_MAP, parseUL\_MAP, parseDCD και parseUCD για να αναδομήσει την απαραίτητη δομή για να μπορέσει να πράξει σωστή λήψη και εκπομπή πακέτων.

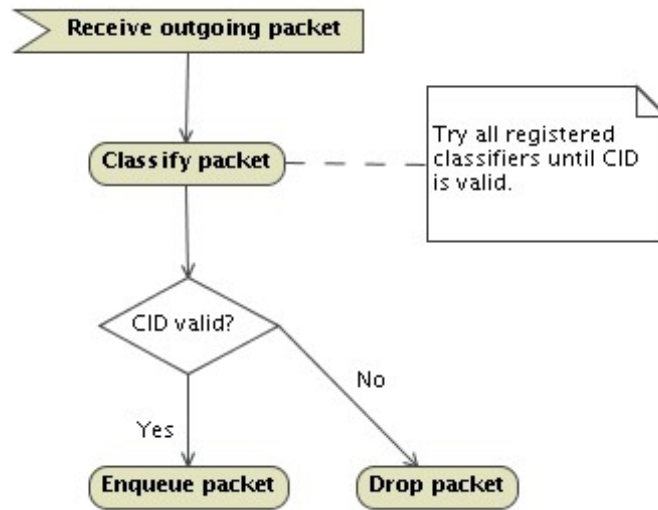
#### 4.2.12 Διαχείριση πακέτων

Η εικόνα 16 απεικονίζει τις ροές των πακέτων για τα εισερχόμενα και τα εξερχόμενα πακέτα.



Εικόνα 16: Ανασκόπηση διαχείρισης πακέτων (NIST module)

Οι εικόνες 17,18 και 19 παρέχουν μερικές αναλυτικές πληροφορίες για το πώς τα πακέτα περνάνε από το επίπεδο MAC.

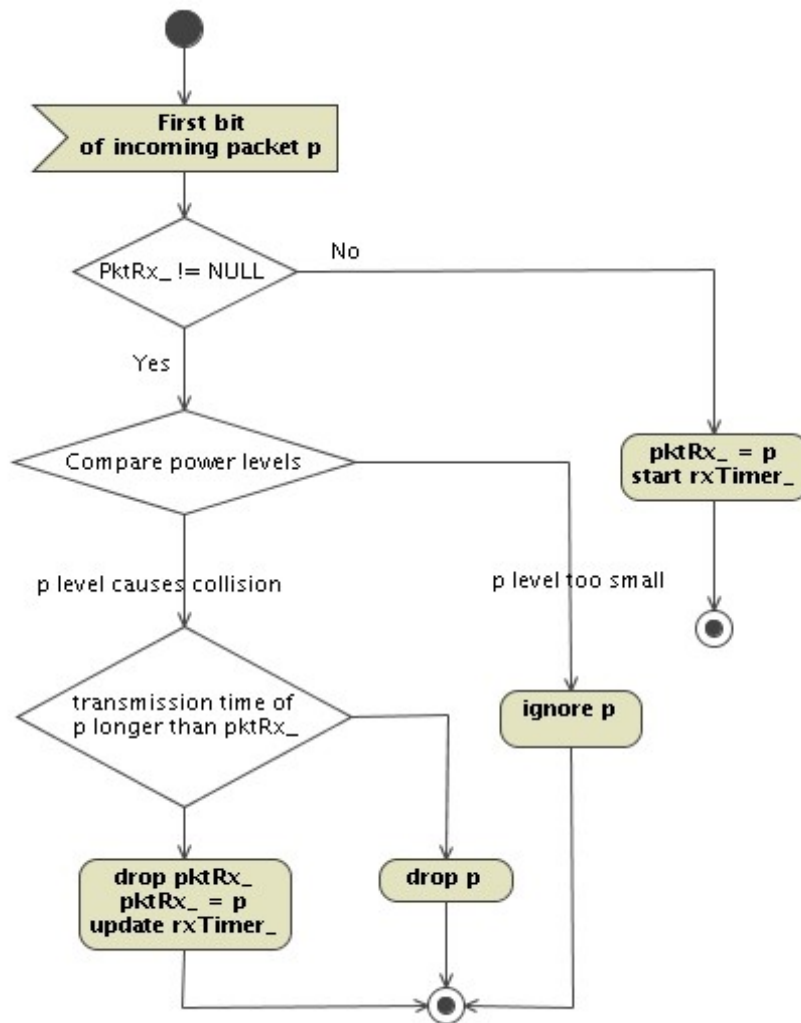


Εικόνα 17: Διαχείριση εξερχομένων πακέτων (NIST module)

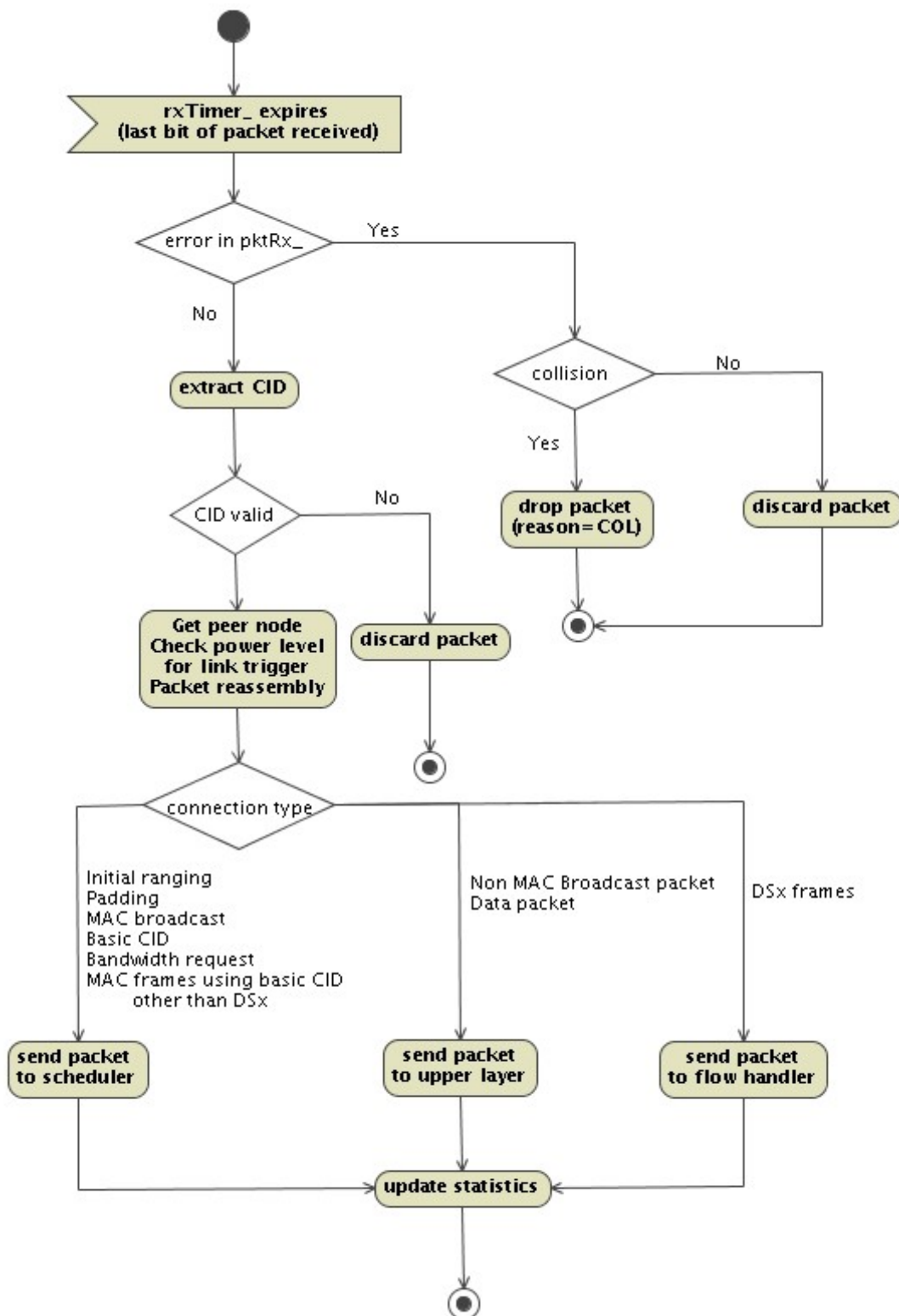
Ένα πακέτο που παραλαμβάνεται από ένα ανώτερο επίπεδο, ταξινομείται από τους καταχωρημένους ταξινομητές, και μιας και μπορεί να υπάρχουν πολλαπλοί ταξινομητές, το επίπεδο MAC τους διαπερνά όλους με την σειρά μέχρι να βρεθεί ένα έγκυρο CID ή μέχρι να έχουν δοκιμαστεί όλοι. Εάν το CID είναι έγκυρο, το πακέτο μπαίνει στην ουρά, αλλιώς ακυρώνεται.

Όταν ένα πακέτο λαμβάνεται, συγκεκριμένα το πρώτο του bit, ακολουθείται η διαδικασία της εικόνας 18. Στο τέλος της διαδικασίας το πακέτο διαχειρίζεται όπως φαίνεται στην εικόνα 19.





Εικόνα 18: Διαχείριση νέων εισερχομένων πακέτων (NIST module)



Εικόνα 19: Διαχείριση ληφθέντων πακέτων (NIST module)

### 4.2.13 OFDM PHY

Το φυσικό επίπεδο OFDM χρησιμοποιείται για την εκπομπή πακέτων στο υλοποιημένο μοντέλο, του οποίου η ρύθμιση για την συχνότητα του εύρους ζώνης και του κυκλικού προθέματος έχει γίνει μέσω συνδέσμων TCL. Αφού κληρονομεί από την κλάση WirelessPhy, παράμετροι όπως για παράδειγμα η συχνότητα ή η δύναμη εκπομπής μπορούν επίσης να ορισθούν μέσω TCL.

Όπως φαίνεται στην εικόνα 20, το φυσικό επίπεδο μπορεί να είναι σε διάφορες καταστάσεις. Όταν βρίσκεται σε κατάσταση αποστολής, όλα τα εισερχόμενα πακέτα απορρίπτονται, ενώ στην κατάσταση λήψης, δεν μπορούν να σταλούν πακέτα. Επιπλέον το header των πακέτων περιέχει πληροφορίες όπως, η συχνότητα, η διαμόρφωση και το κυκλικό πρόθεμα, τα οποία μπορούν να χρησιμοποιηθούν σαν φίλτρα στα εισερχόμενα πακέτα.



Εικόνα 20: Διάγραμμα κλάσεων του OFDM PHY (NIST module)

Το μοντέλο υποστηρίζει διάφορες διαμορφώσεις. Το επίπεδο MAC αναθέτει ριπές οι οποίες μπορούν να χρησιμοποιήσουν διαφορετικές διαμορφώσεις σύμφωνα με την απόσταση ή την παρεμβολή, κάτι που επιδρά στον ρυθμό δεδομένων και στον χρόνο εκπομπής. Το φυσικό επίπεδο περιέχει βοηθητικές συναρτήσεις που καλούνται από το επίπεδο MAC όταν εκπέμπει δεδομένα:

- Η `getTxTime` επιστρέφει τον χρόνο που απαιτείται για να σταλεί ένα πακέτο σύμφωνα με το μέγεθός του και την διαμόρφωση.
- Η `getMaxPktSize` είναι η αντίστροφη από την προηγούμενη συνάρτηση και επιστρέφει το μέγιστο μέγεθος πακέτου δεδομένου του αριθμού των συμβόλων OFDM που είναι διαθέσιμα και στην διαμόρφωση που χρησιμοποιείται.

Οι συναρτήσεις `node_on` και `node_off` ενεργοποιούν ή απενεργοποιούν το μπλοκάρισμα όλων των μεταδόσεων και λήψεων πακέτων, αλλά προς το παρόν δεν συνδέονται με κάποιους μηχανισμούς κατανάλωσης ενέργειας.

### 4.3 Διαδικασία εγκατάστασης

Η εγκατάσταση του `module` έγινε σε ένα εικονικό μηχάνημα του λογισμικού `virtualbox` της `sun`, το οποίο είχε σαν λειτουργικό το `Fedora 13`.

Τα βήματα της εγκατάστασης είναι τα παρακάτω:

- 1) Κατεβάζουμε το `ns-allinone-2.31` από την παρακάτω διεύθυνση :  
<http://sourceforge.net/projects/nsnam/files/allinone/>
- 2) Αποσυμπιέζουμε το αρχείο που κατεβάσαμε στην επιθυμητή τοποθεσία.
- 3) Κατεβάζουμε το `patch` από την σελίδα του project :  
[http://www.nist.gov/itl/antd/emntg/ssm\\_tools.cfm](http://www.nist.gov/itl/antd/emntg/ssm_tools.cfm)
- 4) Αποσυμπιέζουμε το `patch` ώστε να έχουμε τον φάκελο `src-prerelease-041707` και τα αρχεία `INSTALL.TXT` και `patch-ns-2.31-041707`.
- 5) Από τον φάκελο του `ns-allinone-2.31` τρέχουμε την ακόλουθη εντολή :  
`patch -p0 <patch-ns-2.31-041707` (χωρίς τα εισαγωγικά).

- 6) Στο αρχείο install στον φάκελο του ns-allinone-2.31 προσθέτουμε την εντολή “make clean” (χωρίς τα εισαγωγικά) στην γραμμή πριν την εντολή “make”.
- 7) Από τον φάκελο του ns-allinone-2.31 τρέχουμε την εντολή “./install”
- 8) Μετά την ολοκλήρωση της εγκατάστασης πρέπει να προσθέσουμε τα παρακάτω στο αρχείο ~/.bashrc :

```
#LD_LIBRARY_PATH
OTCL_LIB=/home/gkaran/ns-allinone-2.31/otcl-1.13
NS2_LIB=/home/gkaran/ns-allinone-2.31/lib
X11_LIB=/usr/X11R6/lib
USR_LOCAL_LIB=/usr/local/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$X11_LIB:
$USR_LOCAL_LIB

#TCL_LIBRARY
TCL_LIB=/home/gkaran/ns-allinone-2.31/tcl8.4.14/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB

#PATH
XGRAPH=/home/gkaran/ns-allinone-2.31/bin:/home/gkaran/ns-allinone-
2.31/tcl8.4.14/unix:/home/gkaran/ns-allinone-2.31/tk8.4.14/unix
NS=/home/gkaran/ns-allinone-2.31/ns-2.31
NAM=/home/gkaran/ns-allinone-2.31/nam-1.13
PATH=$PATH:$XGRAPH:$NS:$NAM
```

Όπου /home/gkaran/ns-allinone-2.31/ βάζετε τον φάκελο που έχετε το ns-allinone-2.31.

Προαιρετικά πριν το βήμα 6 αν θέλουμε να λειτουργήσει σωστά το nam ακολουθούμε τα παρακάτω βήματα:

- 1) Μέσα στον φάκελο του tk8.4.14 που βρίσκεται μέσα στον φάκελο του ns-allinone-2.31 δημιουργούμε ένα αρχείο με όνομα “tk-lastevent.patch” ή όπως αλλιώς θέλουμε και γράφουμε σε αυτό τις παρακάτω εντολές.

```
--- generic/tk.h.orig 2008-02-06 16:31:40.000000000 +0100
+++ generic/tk.h      2008-07-24 08:21:46.000000000 +0200
@@ -635,17 +635,15 @@
 *
```

```
*-----  
-----  
*/  
-#define VirtualEvent      (LASTEvent)  
-#define ActivateNotify   (LASTEvent + 1)  
-#define DeactivateNotify (LASTEvent + 2)  
-#define MouseWheelEvent  (LASTEvent + 3)  
-#define TK_LASTEVENT    (LASTEvent + 4)  
+#define VirtualEvent      (MappingNotify + 1)  
+#define ActivateNotify   (MappingNotify + 2)  
+#define DeactivateNotify (MappingNotify + 3)  
+#define MouseWheelEvent  (MappingNotify + 4)  
+#define TK_LASTEVENT    (MappingNotify + 5)  
#define MouseWheelMask   (1L << 28)  
-  
#define ActivateMask      (1L << 29)  
#define VirtualEventMask  (1L << 30)  
-#define TK_LASTEVENT    (LASTEvent + 4)  
/*
```

- 2) Στον ίδιο φάκελο που είμαστε τρέχουμε την παρακάτω εντολή :  
“patch -p0 < tk-lastevent.patch” (χωρίς τα εισαγωγικά).
- 3) Αν η εντολή μας επιστρέψει σφάλμα μπορούμε να το κάνουμε χειροκίνητα.  
Ανοίγουμε το αρχείο tk.h στον φάκελο generic και περίπου στην 635 γραμμή πάμε και σβήνουμε όσες σειρές είναι στον πιο πάνω πλαίσιο με ένα “-“ μπροστά από το “#”. Στην συνέχεια προσθέτουμε τις σειρές που έχουν το “+” μπροστά από το “#”. Φροντίζουμε μόνο η σειρά των γραμμών να είναι όπως στο πλαίσιο πιο πάνω.

## 4.4 Προσομοίωση

### 4.4.1 Σενάριο προσομοίωσης

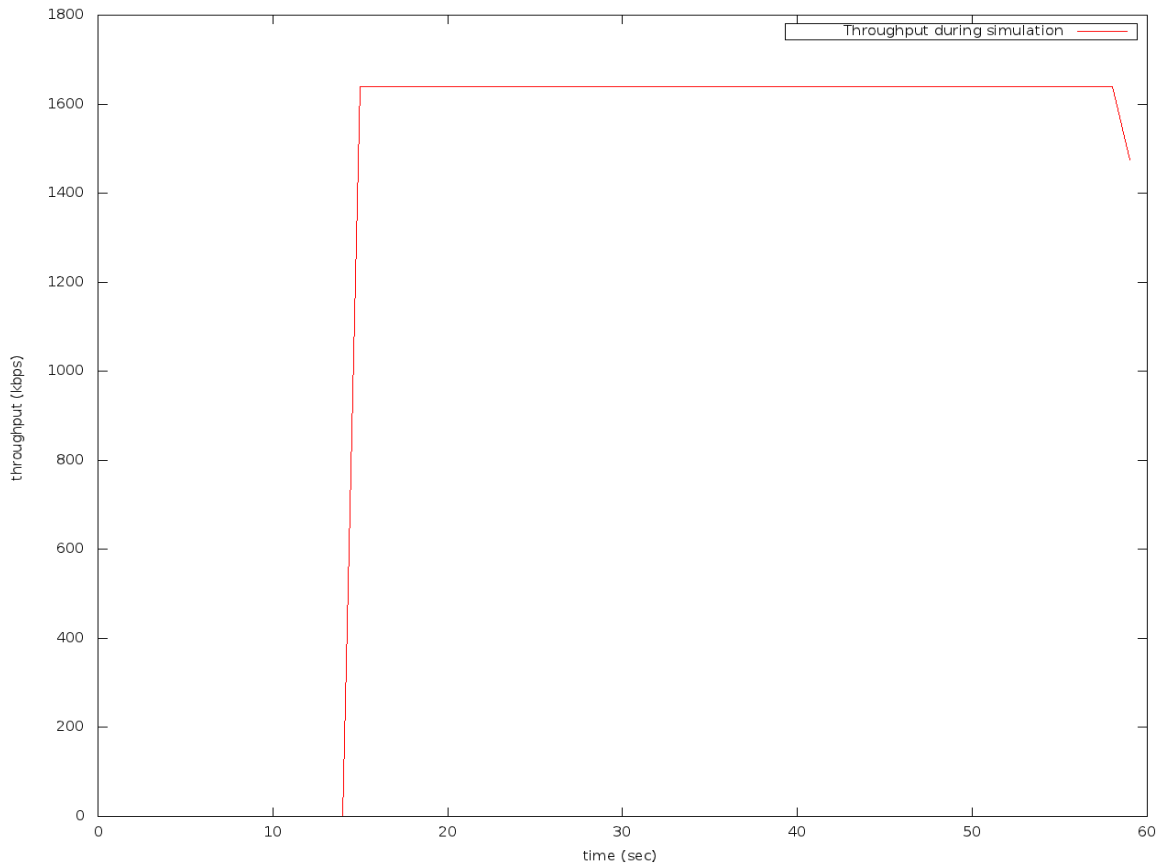
Το σενάριο το οποίο γράφτηκε για τις ανάγκες της εργασίας αυτής ακολουθεί την παρακάτω δομή. Έχουμε ένα σταθμό βάσης και έναν σταθμό συνδρομητή όπου και οι δύο είναι στάσιμοι. Ορίζουμε μεταξύ τους μια βασική επικοινωνία τύπου CBR με μέγεθος πακέτου 1500 bytes και διάστημα μεταξύ αποστολής δύο πακέτων 0.005 δευτερόλεπτα. Δεν ορίζεται κάποιος μηχανισμός QoS και ο χρόνος προσομοίωσης ορίζεται στα 60 δευτερόλεπτα, όπου στα 15 πρώτα δευτερόλεπτα

δεν υπάρχει κάποια CBR επικοινωνία μεταξύ των δύο σταθμών. Στο 15ο δευτερόλεπτο η επικοινωνία ξεκινάει και σταματάει 1 δέκατο του δευτερολέπτου πριν το τέλος του χρόνου προσομοίωσης. Σαν έξοδο όταν τρέξουμε το σενάριο έχουμε δύο αρχεία. Ένα LRC.tr και ένα LRC.nam. Το πρώτο χρησιμοποιείται για την εξόρυξη των στατιστικών της προσομοίωσης και το δεύτερο για την γραφική απεικόνιση των γεγονότων κατά την διάρκεια της προσομοίωσης. Ο κώδικας TCL που εκφράζει το σενάριο της προσομοίωσης είναι διαθέσιμος στο παράρτημα Α.

#### **4.4.2 Αποτελέσματα προσομοίωσης**

Στην τρέχουσα ενότητα θα δούμε και θα σχολιάσουμε διάφορα γραφήματα που αφορούν την προσομοίωση.

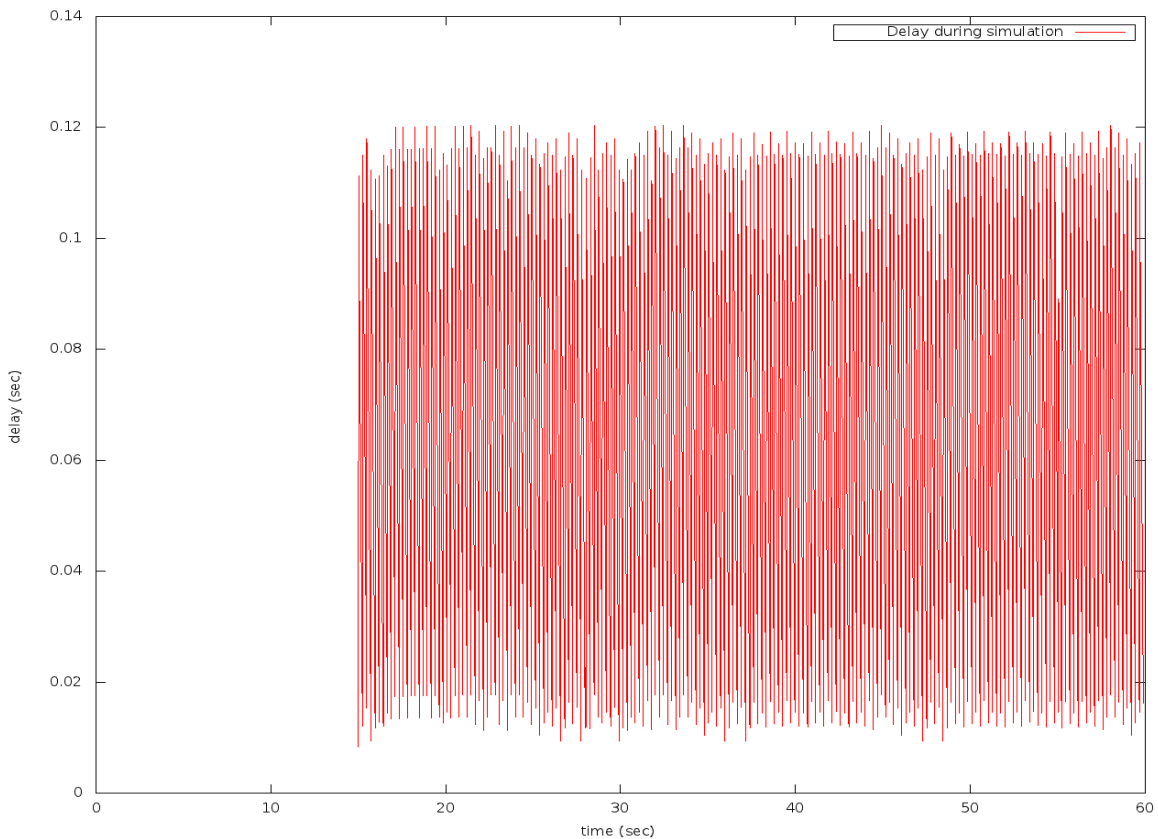
Στην εικόνα 21 βλέπουμε το throughput κατά την διάρκεια της προσομοίωσης. Παρατηρούμε ότι είναι σταθερό από το 15<sup>ο</sup> δευτερόλεπτο (όπου στην ουσία αρχίζει η επικοινωνία) και καθ' όλη την διάρκεια της προσομοίωσης και μειώνεται κατακόρυφα στα τελευταία δευτερόλεπτα όπου και κλείνει η επικοινωνία.



Εικόνα 21: Throughput κατά την διάρκεια της προσομοίωσης (NIST)

Στην εικόνα 22 όπου εμφανίζεται το delay κατά την διάρκεια της προσομοίωσης βλέπουμε ότι έχουμε μια διακύμανση μεταξύ 0.01 και 0.12 δευτερολέπτων κατά την χρονική περίοδο που έχουμε κίνηση στο δίκτυο μας, κάτι το οποίο δεν είναι και τόσο καλό από την στιγμή που υπάρχει κίνηση μόνο μεταξύ των δύο αυτών σταθμών που έχουμε χωρίς να παρεμβάλλει κάποιος τρίτος ώστε να κινείται μεγάλος αριθμός πακέτων στο δίκτυο.





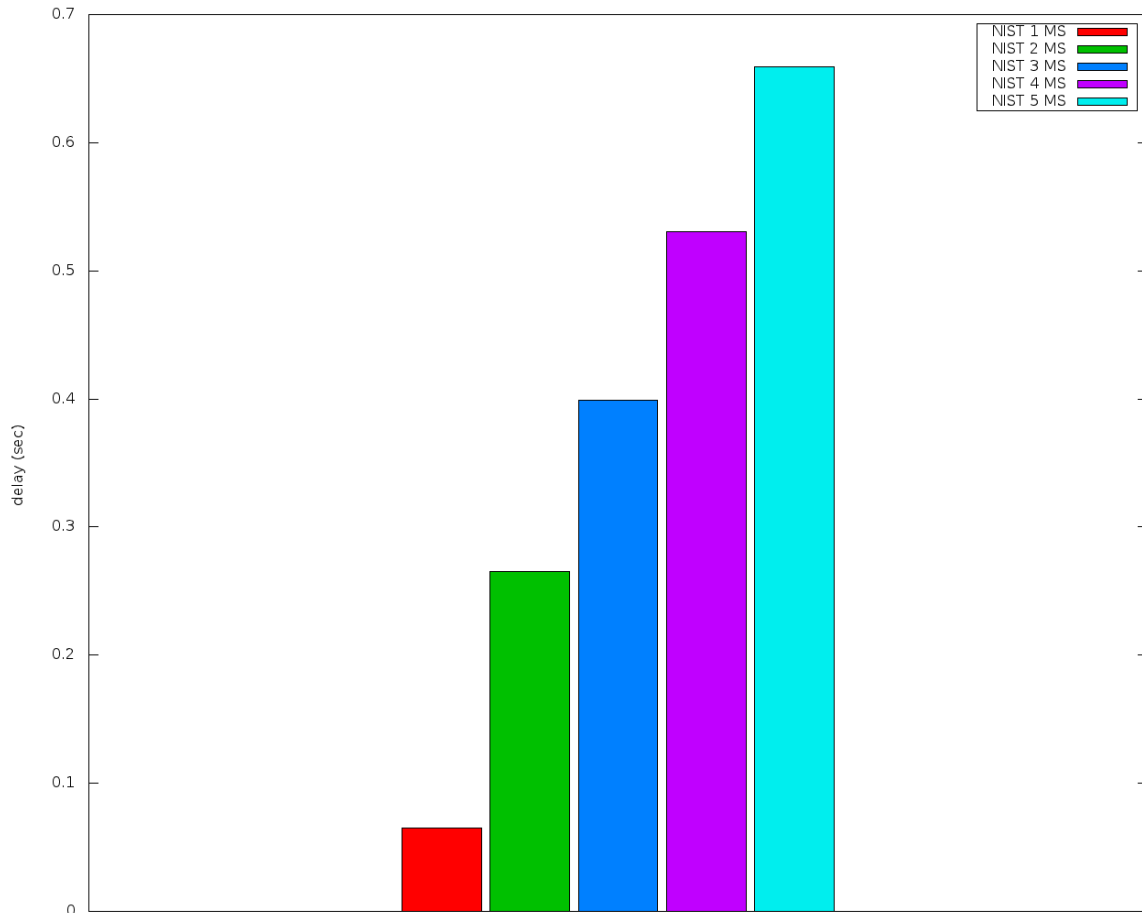
Εικόνα 22: Delay κατά την διάρκεια της προσομοίωσης (NIST module)

#### 4.4.3 Περαιτέρω διερεύνηση

Για να έχουμε μια καλύτερη αντίληψη για την συμπεριφορά του module το σενάριο επαναλήφθηκε με ίδιες ρυθμίσεις αλλά με διαφορετικό αριθμό σταθμών κάθε φορά, ορίζοντας ένα χρονικό κενό μεταξύ της έναρξης εκπομπής δύο σταθμών της τάξης του μισού δευτερολέπτου. Συγκεκριμένα έγιναν δοκιμές για 1,2,3,4 και 5 σταθμούς συνδρομητές. Τα αποτελέσματα των προσομοιώσεων φαίνονται στις παρακάτω εικόνες.

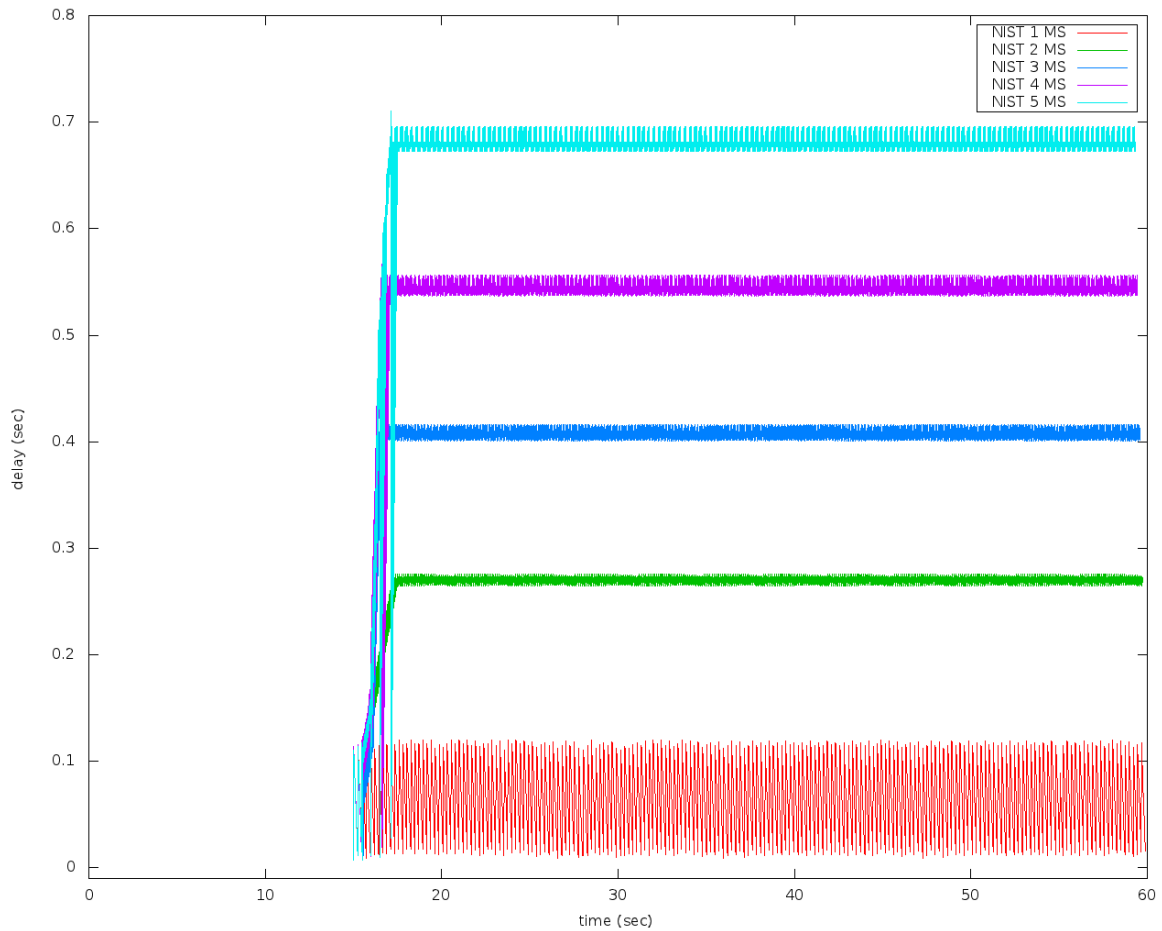
Στην εικόνα 23 παρατηρούμε ότι αυξάνοντας τους σταθμούς συνδρομητές, αυξάνεται αντίστοιχα και η μέση καθυστέρηση στο δίκτυο. Η μετάβαση βέβαια από τον 1 στους δύο σταθμούς έχει μεγαλύτερη διαφορά από ότι οι επόμενες

μεταβάσεις πράγμα που δείχνει ότι πέρα του ενός σταθμού η καθυστέρηση αυξάνεται σημαντικά στο δίκτυο.



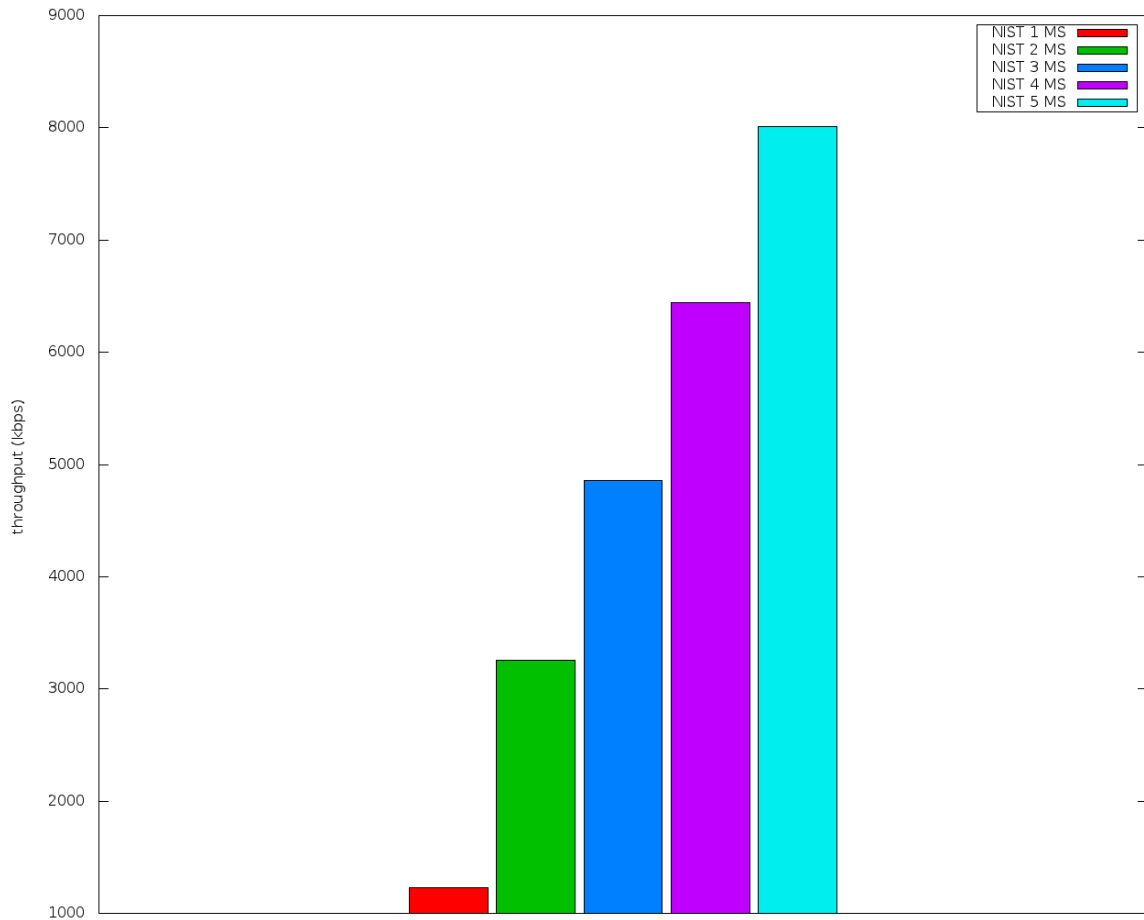
Εικόνα 23: Μέσο delay για τις διαφορετικές προσομοιώσεις (NIST module)

Στην εικόνα 24 βλέπουμε ξανά την καθυστέρηση στο δίκτυο, αυτή τη φορά όμως πιο αναλυτικά κατά την διάρκεια της προσομοίωσης. Βλέπουμε ότι μόλις αρχίσει η εκπομπή από όλους τους σταθμούς η καθυστέρηση αυξάνεται εκθετικά.



Εικόνα 24: Αναλυτικό delay για τις διαφορετικές προσομοιώσεις (NIST module)

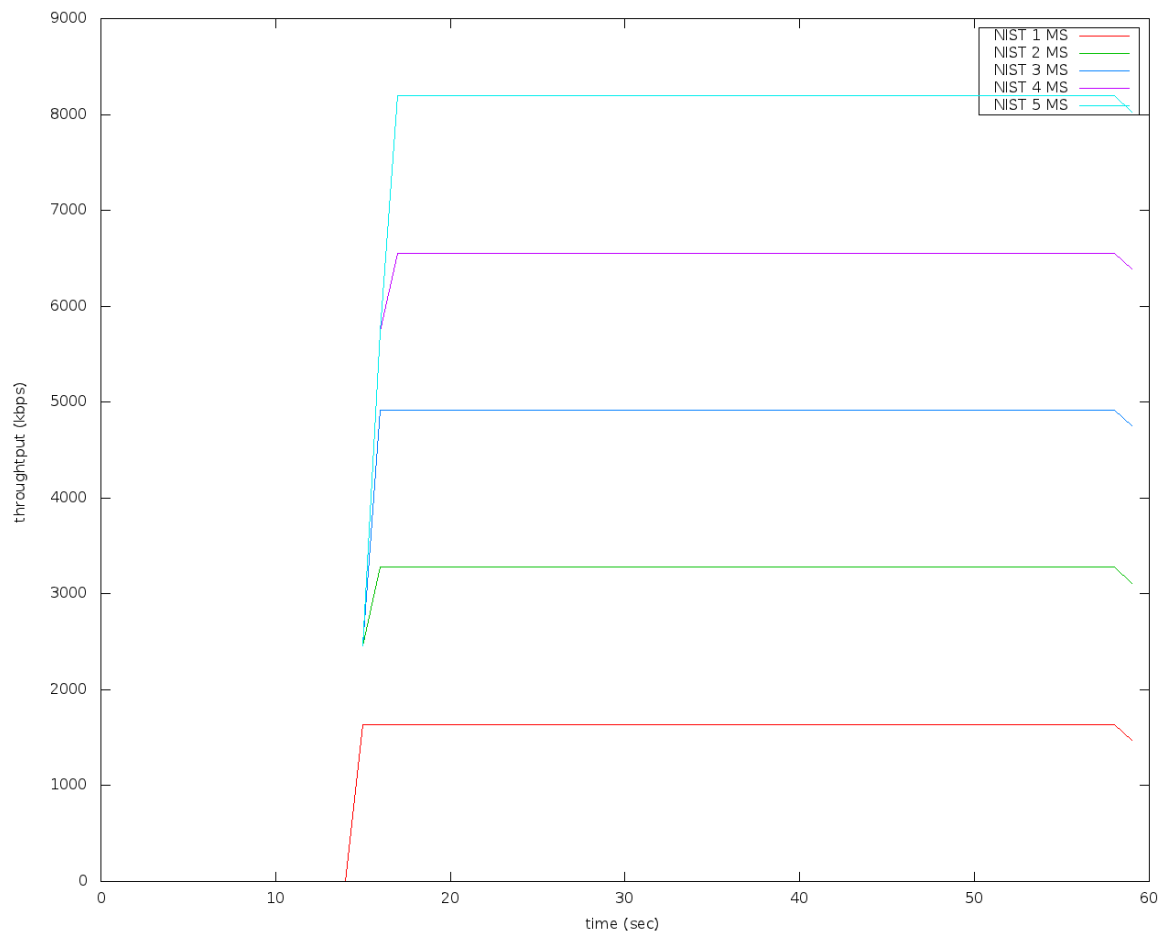
Στην εικόνα 25 όπου παρουσιάζεται το μέσο throughput για τις διαφορετικές προσομοιώσεις παρατηρούμε ότι με την χρήση δύο σταθμών έχουμε πολύ μεγάλη αύξηση στο throughput του δικτύου σε σχέση με τον έναν σταθμό. Από εκεί και πέρα στην χρήση περαιτέρω σταθμών υπάρχει πάλι μια αύξηση αλλά πιο ήπια.



Εικόνα 25: Μέσο throughput για τις διαφορετικές προσομοιώσεις (NIST module)

Στην εικόνα 26 τέλος όπου εμφανίζεται το throughput καθ' όλη την διάρκεια της προσομοίωσης παρατηρούμε μια κατακόρυφη αύξησή του κατά της αρχή εκπομπής των σταθμών, και μια διατήρηση των επιπέδων μέχρι το τέλος της προσομοίωσης.

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου



Εικόνα 26: Αναλυτικό throughput για τις διαφορετικές προσομοιώσεις (NIST module)

## 5 Wimax Forum module

### 5.1 Εισαγωγή

Η υλοποίηση του συγκεκριμένου module βασίζεται στο πρότυπο IEEE 802.16 (802.16-2004) καθώς και στην επέκταση του 802.16e-2005. Συγκεκριμένα το module αυτό αποτελεί μια επέκταση του NIST module. Παρακάτω παραθέτονται οι δυνατότητες καθώς και οι ελλείψεις του module.

Πίνακας 8: Δυνατότητες και ελλείψεις του WiMAX Forum module

#### Δυνατότητες

- Υποστήριξη του φυσικού επιπέδου WirelessMAN-OFDMA με παραμετροποιήσιμη διαμόρφωση.
- Time Division Duplexing (TDD)
- Μηνύματα διαχείρισης για την είσοδο σε δίκτυο (χωρίς αυθεντικοποίηση)
- Εξ' ορισμού χρονοπρογραμματιστής που κατανέμει πόρους στους εγγεγραμμένους κινητούς σταθμούς χρησιμοποιώντας round robin σύμφωνα με το εύρος ζώνης που αιτούνται.
- Πρόσθετα του 802.16e για την υποστήριξη σαρώσεων και μεταβιβάσεων
- Κατακερματισμός και επανασύνθεση των πλαισίων (frames)
- Υποστήριξη του φυσικού επιπέδου OFDMA
- Δυνατότητα χρήσης EESM
- Δυνατότητα επιλογής μοντέλων ταχείας διάλειασης : ITU PED A, PED B και VEHID A
- Ροή υπηρεσιών και χρονοπρογραμματισμός QoS
- AQQ
- Επίλυση ανταγωνισμού με την χρήση CDMA

#### Ελλείψεις

- Περιοδική κύμανση και προσαρμογές ενέργειας
- Διόρθωση Σφαλμάτων
- NrtPS και ertPS
- Προσαρμόσιμη διαμόρφωση και συγγραφή κώδικα
- MIMO
- HARQ
- Έλεγχος εισόδου
- Αυθεντικοποίηση

- Συνένωση (πολλαπλά PDUs του επιπέδου MAC ενώνονται σε μια ριπή του φυσικού επιπέδου)

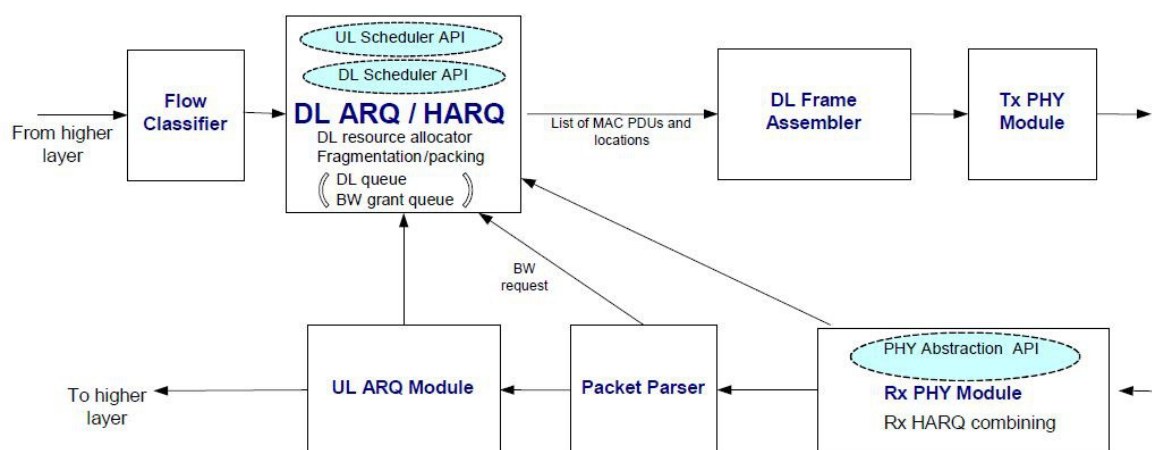
Αξίζει να σημειωθεί ότι πολλά συστατικά δεν υπάρχουν στο πρότυπο. Γι' αυτό το μοντέλο υλοποιεί μιας λύση η οποία μπορεί να μην ταιριάζει απαραίτητα στις ανάγκες κάθε χρήστη. Γι' αυτό ο σχεδιασμός του module είναι τέτοιος ώστε να είναι όσο το δυνατόν πιο επεκτάσιμο.

## 5.2 Documentation [10]

### 5.2.1 Σταθμός Βάσης

#### 5.2.1.1 Δομή του σταθμού βάσης

Η δομή του σταθμού βάσης φαίνεται στην εικόνα 27.



Εικόνα 27: Δομή του σταθμού βάσης (WiMAX Forum module)

Τα κύρια χαρακτηριστικά του είναι τα παρακάτω :

#### 5.2.1.1.1 Ταξινομητής ροής

Το κομμάτι αυτό εκτελεί συναρτήσεις του 802.16 MAC CS καταγραφής των εισερχομένων μονάδων δεδομένων υπηρεσιών δικτύου (PDUs) στον σωστό αναγνωριστή ροής δεδομένων (SFID) και στον αναγνωριστή σύνδεσης (CID). Επιπλέον εκτελεί απόκρυψη των header των πακέτων (PHS) εάν ο αντίστοιχος

κανόνας έχει οριστεί στην ροή δεδομένων. Όλα τα εισερχόμενα πακέτα από τα υψηλότερα στρώματα περνάνε από αυτό το κομμάτι πριν κατευθυνθούν στην ουρά που αντιστοιχεί στο CID.

#### **5.2.1.1.2 Χρονοπρογραμματιστής (DL ARQ/HARQ)**

Ο χρονοπρογραμματιστής είναι ένα περίπλοκο κομμάτι καθώς πρέπει να κρατά πολλές πληροφορίες ώστε να μπορεί να χρονοπρογραμματίσει σωστά και αποδοτικά τα πακέτα. Ο χρονοπρογραμματιστής πρέπει να κρατάει τις παρακάτω πληροφορίες:

- Πληροφορίες QoS για κάθε ροή
- Την κατάσταση της ουράς του downlink για κάθε ροή
- Την εκχώρηση εύρους ζώνης του uplink για κάθε ροή ή για κάθε κινητό
- Πληροφορίες για την κατάσταση του καναλιού για κάθε κινητό

#### **5.2.1.1.3 Uplink ARQ**

Αυτό το κομμάτι διαχειρίζεται τα χαλασμένα ή ανολοκλήρωτα πακέτα που έχουν ληφθεί. Η ανατροφοδότηση του ARQ στέλνεται πίσω στο πομπό μέσω των πληροφοριών κατάστασης που μεταφέρονται μεταξύ αυτού του κομματιού και το DL ARQ/HARQ.

#### **5.2.1.1.4 Συμβολομεταφραστής πακέτων downlink**

Ο συμβολομεταφραστής πακέτων του downlink ενώνει όλα τα πακέτα που παράγονται από τον χρονοπρογραμματιστή για να φτιάξουν ένα frame και προσθέτει κάποιες επιπλέον πληροφορίες στο frame όπως οι χάρτες των downlink και uplink. Στην συνέχεια ο συμβολομεταφραστής προωθεί το downlink frame στο κομμάτι στο Tx του φυσικού επιπέδου.

#### **5.2.1.1.5 Αναλυτής πακέτων**

Ο σταθμός βάσης αναλύει πακέτα, δηλαδή ταξινομεί τα εισερχόμενα πακέτα σύμφωνα με τον τύπο των header των πακέτων : πακέτα δεδομένων και πακέτα ελέγχου.



### 5.2.1.1.6 Tx/Rx PHY

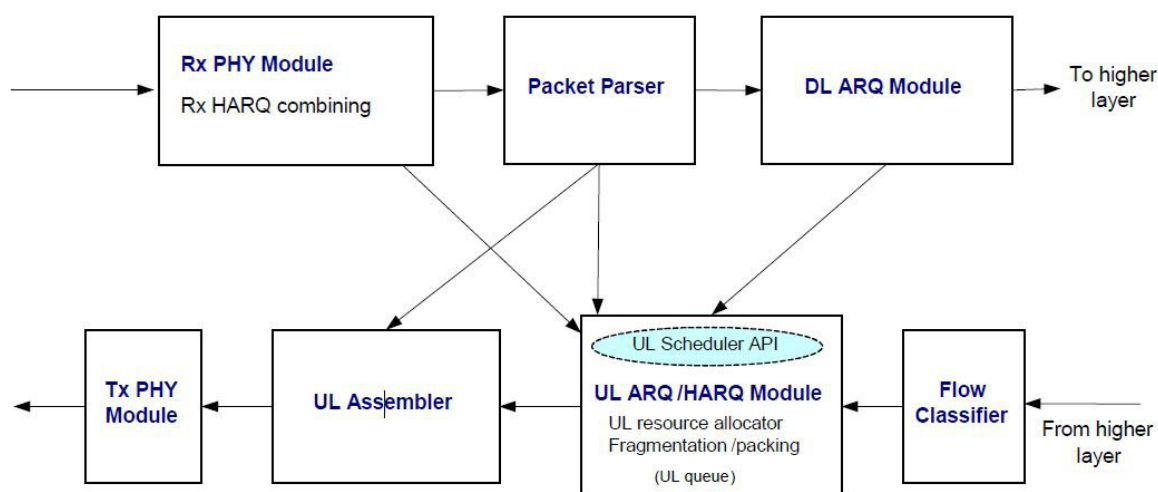
Το downlink κομμάτι του φυσικού επιπέδου λειτουργεί απλά σαν μεταφορέα πακέτων στο ασύρματο κανάλι. Επιλεκτικά, μπορεί να γράψει κάποιες πληροφορίες στο πακέτο, όπως η ώρα εκπομπής, η ισχύς και η συχνότητα.

Το uplink κομμάτι του φυσικού επιπέδου υπολογίζει τις SINR πληροφορίες για όλα τα εισερχόμενα πακέτα και υλοποιεί την διεπαφή στους πίνακες του φυσικού επιπέδου, οι οποίοι παρέχουν πληροφορίες σφαλμάτων blocks όταν ερωτηθούν με το μέγεθος του block και την τιμή του SINR.

## 5.2.2 Κινητός σταθμός συνδρομητής

### 5.2.2.1 Δομή του κινητού σταθμού συνδρομητή

Η δομή του κινητού σταθμού συνδρομητή απεικονίζεται στην εικόνα 28.



Εικόνα 28: Δομή του κινητού σταθμού συνδρομητή (WiMAX Forum module)

Μια από τις κύριες λειτουργίες του κινητού σταθμού συνδρομητή είναι να αναλύει του εισερχόμενους uplink χάρτες και να αντλεί πληροφορίες ( χρόνος εκκίνησης και τερματισμού ριπής στην περίπτωση του SC-PHY) και να φτιάχνει ριπές χρησιμοποιώντας δεδομένα από τις ουρές δεδομένων που αφορούν τον κινητό σταθμό συνδρομητή.

#### 5.2.2.1.1 Χρονοπρογραμματιστής uplink (μονάδα ARQ/HARQ)

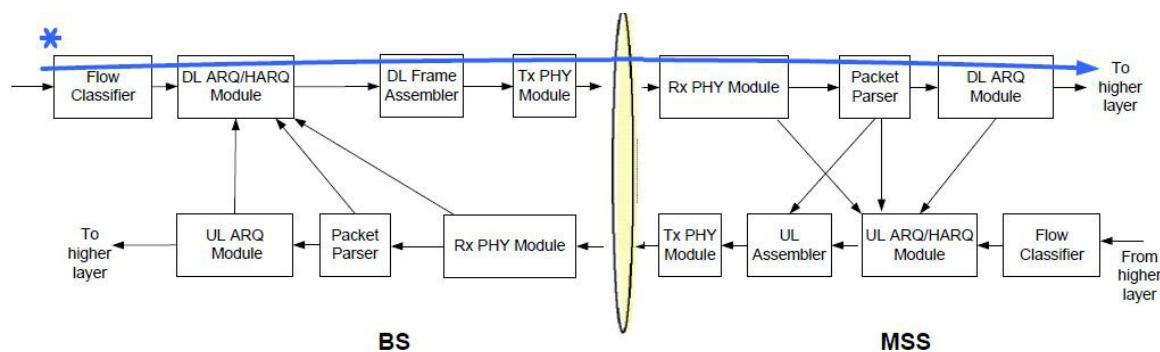
Ο χρονοπρογραμματιστής του uplink παίρνει την εκχώρηση του εύρους ζώνης στο κινητό για κάθε frame από τον αναλυτή πακέτων και μετά χρονοπρογραμματίζει τις κατάλληλες ποσότητες δεδομένων στις εκχωρημένες υποδοχές του uplink.

### 5.2.2.1.2 Αναλυτής uplink

Αυτό το κομμάτι ανιχνεύει τις ουρές στον κινητό σταθμό συνδρομητή και δημιουργεί μια ριπή δεδομένων η οποία μπαίνει στις εκχωρημένες υποδοχές. Οι λειτουργίες των υπόλοιπων κομματιών είναι οι ίδιες με τον σταθμό βάσης.

## 5.2.3 Ροές πακέτων στον προσομοιωτή

### 5.2.3.1 Κανονική ροή πακέτων δεδομένων downlink



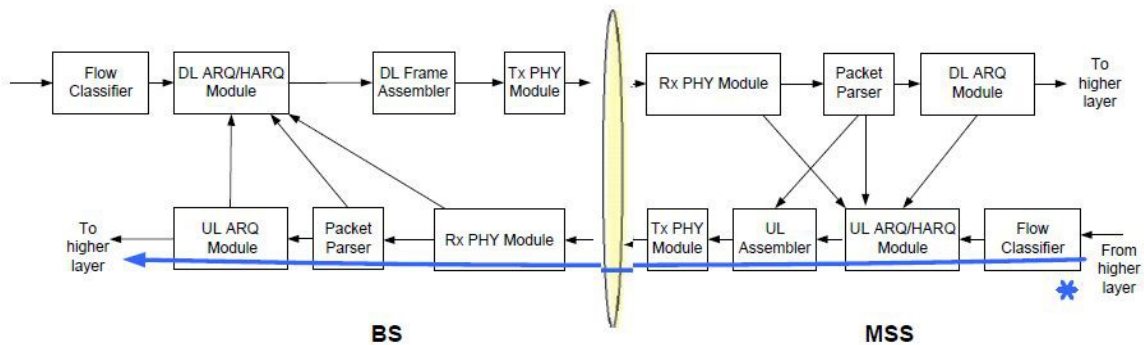
\* indicates the beginning of the flow.

Εικόνα 29: Κανονική ροή πακέτων δεδομένων downlink (WiMAX Forum module)

Η ροή δεδομένων του downlink στέλνεται από τον σταθμό βάσης στον κινητό σταθμό συνδρομητή. Τα πακέτα δεδομένων που έλαβε ο σταθμός βάσης από υψηλότερα στρώματα προωθούνται στον Ταξινομητή Ροής ο οποίος εκτελεί συναρτήσεις του υποστρώματος σύγκλισης του IEEE 802.16 αναθέτοντας πακέτα στην ροή δεδομένων και στην σύνδεση MAC. Τα ταξινομημένα πακέτα προωθούνται στο κομμάτι ARQ/HARQ του downlink και αποθηκεύονται σε ουρές. Ο ταξινομητής downlink αναλύει τις ουρές και χρονοπρογραμματίζει πακέτα για εκπομπή στον κινητό σταθμό συνδρομητή. Κάθε ξεχωριστό PDU του MAC επιπέδου δημιουργείται από την συνάρτηση κομματιασμού/πακετοποίησης του κομματιού ARQ/HARQ του downlink. Το κομμάτι ARQ/HARQ του downlink επιστρέφει μια λίστα με PDUs του επιπέδου MAC και την θέση τους στον αναλυτή frames του downlink, και μετά ο αναλυτής των frames τα συναρμολογεί στο

υποπλαίσιο του downlink και προσθέτει επιπλέον χάρτες του downlink και uplink και άλλες δομές δεδομένων όπου χρειάζονται για την προσομοίωση για να συμπληρώσει το frame του downlink. Ο αναλυτής frames του downlink προωθεί το ολοκληρωμένο frame του downlink στο Tx κομμάτι του φυσικού επιπέδου. Το Tx κομμάτι διατηρεί τον φυσικό σύνδεσμο και εκπέμπει το frame του downlink στον κινητό σταθμό συνδρομητή.

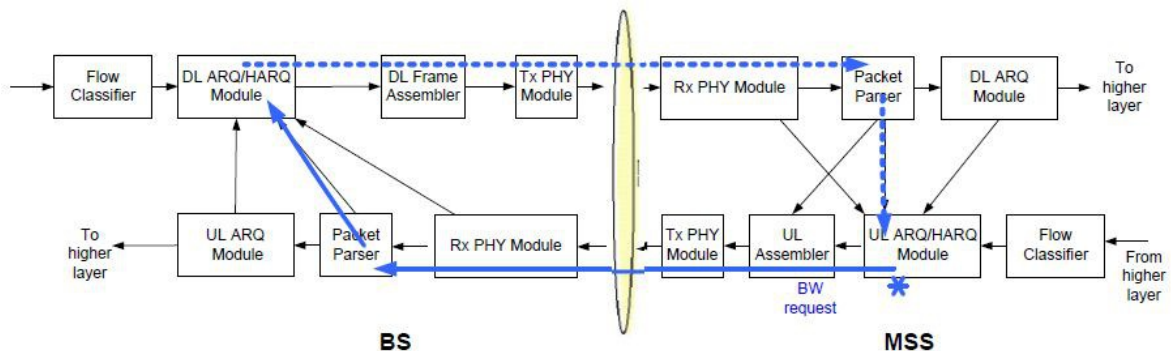
### 5.2.3.2 Κανονική ροή πακέτων δεδομένων uplink



Εικόνα 30: Κανονική ροή δεδομένων πακέτων uplink (WiMAX Forum module)

Η ροή δεδομένων του uplink στέλνεται από τον κινητό σταθμό συνδρομητή στον σταθμό βάσης. Η ροή δεδομένων πακέτων του uplink είναι παρόμοια με του downlink μόνο που η μετάδοση γίνεται προς την αντίθετη κατεύθυνση.

### 5.2.3.3 Ροή δεδομένων για τα μηνύματα αίτησης εύρους ζώνης



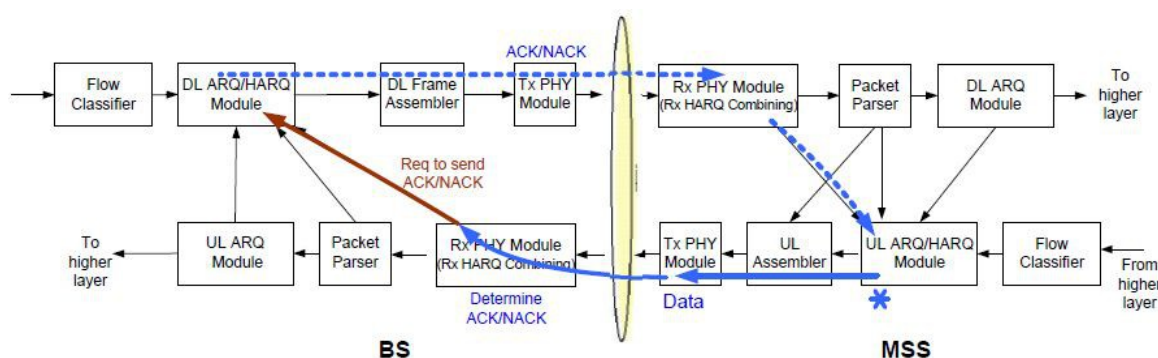
Εικόνα 31: Ροή πακέτων για CQICH (WiMAX Forum module)

Το CQICH είναι σχεδιασμένο για κινητούς σταθμούς συνδρομητές που υποστηρίζουν HARQ. Το CQI κυρίως αναφέρεται από το κομμάτι Tx του φυσικού επιπέδου του κινητού σταθμού συνδρομητή. Το Tx κομμάτι του φυσικού επιπέδου

του κινητού σταθμού συνδρομητή χρησιμοποιεί το κανάλι γρήγορης ανάδρασης του CQICH για να στέλνει πληροφορίες που αφορούν την ποιότητα του καναλιού. Το Rx κομμάτι του φυσικού επιπέδου του σταθμού βάσης αντλεί το CQI από το uplink frame, και το μεταφράζει σε κατάσταση καναλιού και στην συνέχεια προωθεί τις πληροφορίες στο κομμάτι ARQ/HARQ του downlink για χρονοπρογραμματισμό ενημέρωσης της ποιότητας του καναλιού. Το κομμάτι ARQ/HARQ του downlink μπορεί προαιρετικά να προωθήσει το CQI στο Tx του φυσικού επιπέδου του σταθμού βάσης, εάν υπάρχει κάποια εργασία προσομοίωσης στο φυσικό επίπεδο που χρειάζεται πληροφορίες για το CQI.

### 5.2.3.4 Ροές δεδομένων του HARQ

#### 5.2.3.4.1 Ροή δεδομένων του uplink HARQ



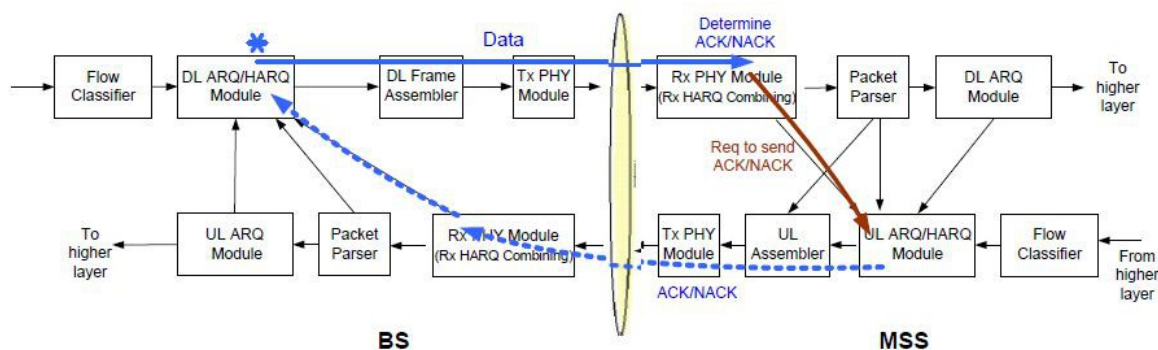
Εικόνα 32: Ροή δεδομένων του HARQ uplink (WiMAX Forum module)

Στην ροή δεδομένων HARQ του uplink, ο κινητός σταθμός συνδρομητής είναι ο πομπός και ο σταθμός βάσης είναι ο παραλήπτης. Η ριπές HARQ εκπέμπονται στο υποπλαίσιο του uplink και εκπέμπονται ACK/NACKs σαν κομμάτι του χάρτη του downlink. Το Rx HARQ του σταθμού βάσης καθορίζει εάν θα στείλει ACK/NACK στον κινητό σταθμό συνδρομητή, ενώ το HARQ του uplink απλά μεταφέρει τα επιτυχή πακέτα στον αναλυτή πακέτων. Η καθυστέρηση του ACK αποτελεί κομμάτι της ρύθμισης του καναλιού στο UCD και το μήνυμα ACK/NACK δεν χρειάζεται να χρονοπρογραμματιστεί. Το Rx HARQ του κινητού σταθμού συνδρομητή προωθεί τα ACK/NACK στο κομμάτι ARQ/HARQ του uplink για την ενημέρωση της δομής των ουρών του uplink. Το ARQ/HARQ του uplink του

κινητού σταθμού συνδρομητή, μόλις παραλάβει ένα ACK/NACK κάνει τα παρακάτω:

- ACK : Αφαιρεί όλα τα σχετικά με το ACK πακέτα από την ουρά του uplink
- NACK : Προετοιμασία για την αποστολή του επόμενου πακέτου.

#### 5.2.3.4.2 Ροή δεδομένων του downlink HARQ



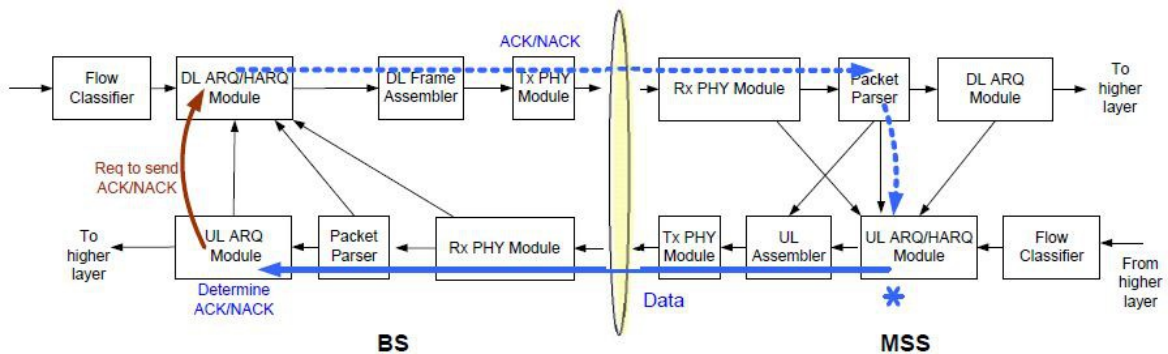
Εικόνα 33: Ροή δεδομένων του HARQ downlink (WiMAX Forum module)

Στη ροή δεδομένων HARQ του downlink, ο σταθμός βάσης είναι ο πομπός και ο κινητός σταθμός συνδρομητής είναι ο παραλήπτης. Οι ριπές HARQ εκπέμπονται στο υποπλαίσιο του downlink και εκπέμπονται ACK/NACKs στις περιοχές του HARQ ACK στο uplink υποπλαίσιο. Το downlink HARQ του κινητού σταθμού συνδρομητή καθορίζει εάν θα σταλούν ACK/NACK στον σταθμό βάσης ενώ το Rx HARQ μεταφέρει τα επιτυχή πακέτα στον αναλυτή πακέτων. Η καθυστέρηση του ACK αποτελεί κομμάτι της ρύθμισης του καναλιού στο DCD και το μήνυμα ACK/NACK δεν χρειάζεται να χρονοπρογραμματιστεί. Το ARQ/HARQ του downlink του σταθμού βάσης, μόλις παραλάβει ένα ACK/NACK κάνει τα παρακάτω:

- ACK : Αφαιρεί όλα τα σχετικά με το ACK πακέτα από την ουρά του uplink
- NACK : Προετοιμασία για την αποστολή του επόμενου πακέτου.

#### 5.2.3.5 Ροές δεδομένων του ARQ

##### 5.2.3.5.1 Ροή δεδομένων του uplink ARQ

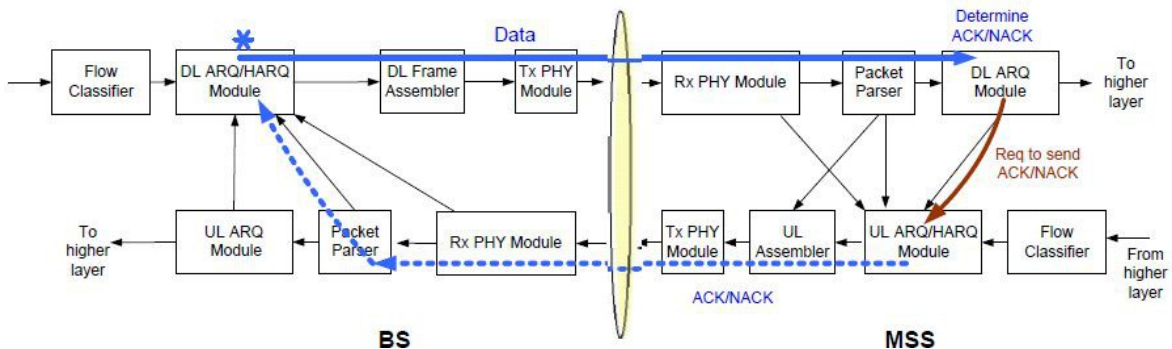


Εικόνα 34: Ροή δεδομένων του ARQ uplink (WiMAX Forum module)

Στην ροή δεδομένων του uplink ARQ, ο κινητός σταθμός συνδρομητής είναι ο πομπός και ο σταθμός βάσης είναι ο δέκτης. Το ARQ κομμάτι του uplink του σταθμού βάσης καθορίζει εάν θα στείλει ACK/NACK στον κινητό σταθμό συνδρομητή. Το μήνυμα ACK/NACK είναι ένα πακέτο και πρέπει να χρονοπρογραμματιστεί για μετάδοση. Μόλις παραλάβει ένα ACK/NACK, το uplink ARQ/HARQ του κινητού σταθμού συνδρομητή πρέπει να κάνει τα παρακάτω:

- ACK : να διαγράψει το πακέτο από την ουρά του uplink
- NACK : προετοιμασία για την επαναποστολή του πακέτου

### 5.2.3.5.2 Ροή δεδομένων του downlink ARQ



Εικόνα 35: Ροή δεδομένων του ARQ downlink (WiMAX Forum module)

Στην ροή δεδομένων του downlink ARQ, ο σταθμός βάσης είναι ο πομπός και ο κινητός σταθμός συνδρομητής είναι ο δέκτης. Το ARQ κομμάτι του downlink του κινητού σταθμού συνδρομητή καθορίζει εάν θα στείλει ACK/NACK στον σταθμό βάσης. Το μήνυμα ACK/NACK είναι ένα πακέτο και πρέπει να

χρονοπρογραμματιστεί για μετάδοση. Μόλις παραλάβει ένα ACK/NACK, το downlink ARQ/HARQ του σταθμού βάσης πρέπει να κάνει τα παρακάτω:

- ACK : να διαγράψει το πακέτο από την ουρά του downlink
- NACK : προετοιμασία για την επαναποστολή του πακέτου

#### **5.2.4 Υπόστρωμα σύγκλισης πακέτων**

Το υπόστρωμα σύγκλισης πακέτων κάθεται πάνω από το επίπεδο MAC\_CPS και εκτελεί τις παρακάτω ενέργειες:

- Λαμβάνει PDUs από υψηλότερο επίπεδα
- Ταξινομεί
- Παραδίδει τα PDUs του στο MAC SAP
- Λαμβάνει PDUs του υποστρώματος σύγκλισης από τον σύνδεσμο

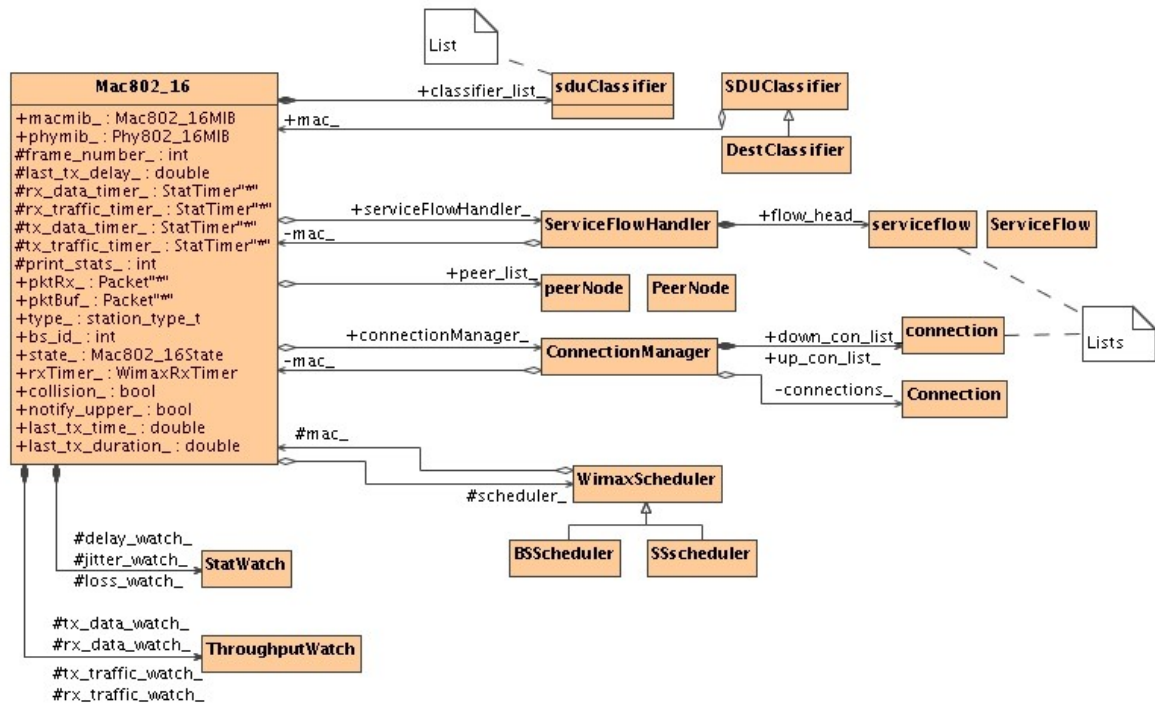
Στην τρέχουσα υλοποίηση, το υπόστρωμα σύγκλισης πακέτων μόνο ταξινομεί. Η μέθοδος που χρησιμοποιείται για ταξινόμηση είναι ανεξάρτητη της υλοποίησής της. Ίσως να είναι και χρήσιμο να υλοποιηθούν πολλαπλές λύσεις ώστε να βρεθεί η καλύτερη δυνατή, αφού το μοντέλο υποστηρίζει ταξινομητές φτιαγμένους από τον χρήστη.

#### **5.2.5 Κοινό τμήμα υποεπιπέδου MAC**

Σε αυτή την ενότητα παρουσιάζεται το υποεπίπεδο MAC το οποίο υποστηρίζει RMP.

##### **5.2.5.1 Δομή του τμήματος MAC**

Η Mac802\_16 είναι μια αφηρημένη υποκλάση της Mac, η οποία περιέχει τα κύρια στοιχεία των BS και MS. Αποτελεί την διασύνδεση με τα άλλα στρώματα για την αποστολή και λήψη πακέτων. Στην εικόνα 36 φαίνεται η κλάση και οι σχέσεις της με άλλες ενότητες.



Εικόνα 36: Διάγραμμα κλάσεων του MAC επιπέδου (WiMAX Forum module)

Μια κλάση MAC έχει μια λίστα ταξινομητών πακέτων (SDUClassifier) η οποία καταγράφει κάθε εξερχόμενο πακέτο με τον κατάλληλο αναγνωριστή σύνδεσης. Με την χρήση της TCL, ο χρήστης καθορίζει την λίστα των ταξινομητών που θα χρησιμοποιηθούν. Η τρέχουσα υλοποίηση χρησιμοποιεί την IP του προορισμού σαν στοιχεία ταξινόμησης.

Η κλάση ServiceFlowHandler είναι υπεύθυνη για τον έλεγχο ροής των αιτημάτων και απαντήσεων. Επιπλέον αποθηκεύει την λίστα των ροών ενός κόμβου.

Ένας κινητός σταθμός συνδρομητής συνδέεται πάνω στον σταθμό βάσης ενώ ο σταθμός βάσης μπορεί να συνδεθεί σε πολλούς κινητούς σταθμούς συνδρομητές. Η κλάση PeerNode περιέχει δεδομένα αναφορικά με τον σταθμό, όπως οι συνδέσεις του και η κατάσταση του. Η συνδέσεις (class Connections) είναι προσβάσιμες επίσης μέσω της κλάσης ConnectionManager, η οποία περιέχει την λίστα με τις εισερχόμενες και εξερχόμενες συνδέσεις.

Η αφηρημένη κλάση WimaxScheduler χρησιμοποιείται για την δημιουργία μιας διεπαφής με το MAC. Κατά κύριο λόγο υπάρχει ένας χρονοπρογραμματιστής για



τον σταθμό εξυπηρετητή και ένας για τον κινητό σταθμό συνδρομητή και εφόσον ο χρονοπρογραμματιστής καθορίζεται με την χρήση TCL, είναι εύκολο να υλοποιηθεί η αφηρημένη κλάση και να αλλαχθεί.

Τέλος η κλάση MAC υπολογίζει στατιστικά μέσω των αντικειμένων StatWatch και ThroughputWatch για πληροφορίες πακέτων και ροής δεδομένων, τα οποία χρησιμοποιούνται για να πυροδοτήσουν διαδικασίες και επιπλέον μπορούν να τυπωθούν κατά την διάρκεια της προσομοίωσης.

Μιας και ο σταθμός βάσης και ο κινητός σταθμός συνδρομητής έχουν διαφορετικές καταστάσεις μηχανής έχουν υλοποιηθεί δύο διαφορετικές κλάσεις για τον σταθμό βάσης και τον κινητό σταθμό συνδρομητή με ονόματα Mac802\_16BS και Mac802\_16SS αντίστοιχα.

#### **5.2.6 Διευθυνσιοδότηση και σύνδεση**

Κάθε MAC έχει μια ξεχωριστή διεύθυνση η οποία καταχωρείται σαν ακέραιος αριθμός. Το μοντέλο επίσης καθορίζει αναγνωριστές σύνδεσης σαν ακέραιους των 16 bit.

Οι παρακάτω συνδέσεις δημιουργούνται κατά την αρχικοποίηση ενός κόμβου σταθμού:

- Αρχική ρύθμιση (εισερχόμενη και εξερχόμενη)
- Padding (εισερχόμενη και εξερχόμενη)
- Broadcast (εξερχόμενη)
- Adaptive Antenna System (AAS) (εξερχόμενη, αλλά δεν χρησιμοποιείται)

Οι παρακάτω συνδέσεις δημιουργούνται κατά την αρχικοποίηση ενός κινητού σταθμού συνδρομητή:

- Αρχική ρύθμιση (εισερχόμενη και εξερχόμενη)
- Padding (εισερχόμενη και εξερχόμενη)
- Broadcast (εισερχόμενη)

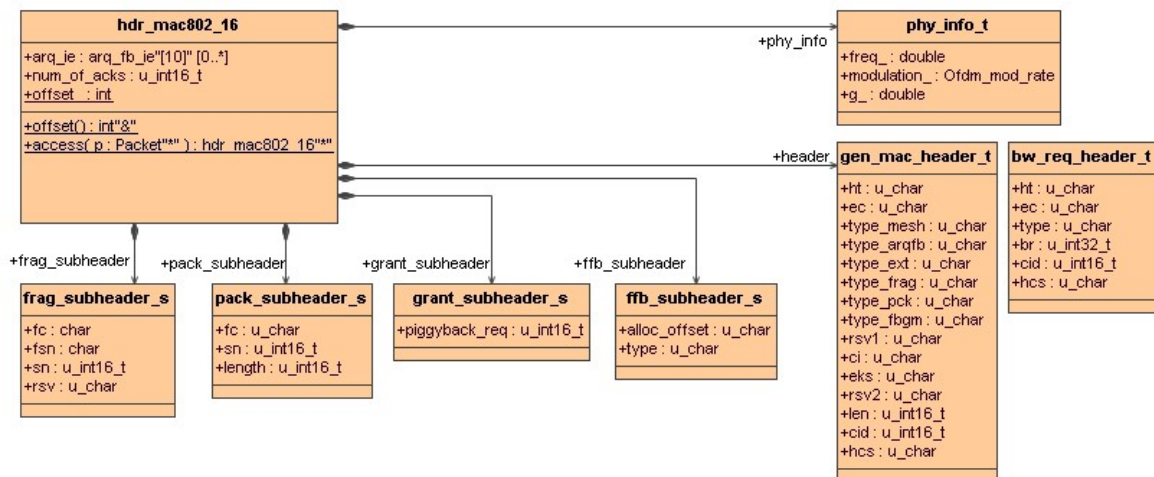
Επιπλέον κατά την καταχώρηση στο δίκτυο καθορίζονται οι παρακάτω συνδέσεις και τα CIDs:

- Basic CID (εισερχόμενη και εξερχόμενη)
- Primary CID (εισερχόμενη και εξερχόμενη)
- Secondary CID (εισερχόμενη και εξερχόμενη)
- Data CIDs. Προς το παρόν το μοντέλο υποστηρίζει μόνο μια σύνδεση δεδομένων

### 5.2.7 Μορφή των PDUs του MAC

Οι σχεδιαστές του μοντέλου έκριναν απαραίτητη την δημιουργία ενός νέου header για να κρατάει τα πακέτα για το IEEE 802\_16.

Στην εικόνα 37 βλέπουμε το διάγραμμα κλάσεων της κλάσης `hdr_mac802_16` που είναι η νέα κλάση που αναφέραμε.



Εικόνα 37: Διάγραμμα κλάσεων της κλάσης "hdr\_mac802\_16" (WiMAX Forum module)

Το header αποτελείται από τα παρακάτω τρία κύρια στοιχεία:

- Ένα εικονικό header φυσικού επιπέδου τύπου `phy_info_t`, το οποίο χρησιμοποιείται για την μεταφορά πληροφοριών όπως η συχνότητα, η αλλοίωση (modulation) και το κυκλικό πρόθεμα (cyclic prefix).

- Ένα γενικό MAC header τύπου `gen_mac_header_t` που περιέχει τις γενικές πληροφορίες σχετικά με το MAC επίπεδο. Η συγκεκριμένη δομή μπορεί να μετατραπεί σε `bw_req_header_t` όταν το πακέτο αποτελεί αίτηση για εύρος ζώνης.
- Δομές που κρατάνε δεδομένα για διάφορα υπο-header, οι οποίες είναι παρούσες σε όλα τα πακέτα ενώ η μεταβλητή *type* του κυρίως header καθορίζει αν η εγγραφή είναι σωστή ή όχι.

Επιπλέον όταν είναι ενεργοποιημένο το ARQ, το header περιέχει και πληροφορίες που χρησιμοποιούνται για ανατροφοδότηση.

Στον πίνακα 9 μπορούμε να δούμε τα πακέτα που έχουν υλοποιηθεί στο μοντέλο μέχρι στιγμής.

Πίνακας 9: Υποστηριζόμενοι τύποι πακέτων (WiMAX Forum module)

Κατηγορία	Οριζόμενα μηνύματα
Συγχρονισμός	DL-MAP / DCD
	UL-MAP / UCD
	RNG-REQ/RSP
Ροές υπηρεσιών	REG-REQ/RSP
	DSA-REQ/PSR/ACK
	MOB_NBR_ADV
	MOB_SCN-REQ/RSP
Κινητικότητα	MOB_BSHO-REQ/RSP
	MON_SSHO-REQ
	MOB-HO-IND
	MOB_SCN-REP
	MOB_ASC-REP

### 5.2.8 Κατασκευή και μετάδοση των PDU του επιπέδου MAC

Η κατασκευή και μετάδοση των πακέτων χωρίζεται στα τρία παρακάτω βήματα:

- Λήψη ενός εξερχομένου πακέτου από το ανώτερο επίπεδο: Το επίπεδο MAC αναζητά το σωστό CID στους ταξινομητές. Αν αυτό βρεθεί τότε προσαρτά στο πακέτο ένα προεπιλεγμένο header και το τοποθετεί στην ουρά της σύνδεσης.
- Χρονοπρογραμματισμός: Για κάθε frame οι χρονοπρογραμματιστές ψάχνουν στην λίστα των συνδέσεων για να βρουν το πακέτο που θα εκπέμψουν.
- Μετάδοση: Δύο timers εκπέμπουν τα πακέτα που είναι κρατημένα στις ουρές.

#### **5.2.8.1 Κατακερματισμός**

Ο κατακερματισμός μπορεί να ενεργοποιηθεί ή απενεργοποιηθεί σε κάθε σύνδεση, ενώ η κατ' εξοχήν επιλογή είναι να είναι ενεργοποιημένος.

Κατά τον χρονοπρογραμματισμό πακέτων για μετάδοση, ο χρονοπρογραμματιστής ελέγχει εάν ο κατακερματισμός είναι ενεργός για την συγκεκριμένη σύνδεση και αν είναι σπάει τα πακέτα. Το πλαίσιο κατακερματισμού αποθηκεύεται στην σύνδεση (class Connection).

#### **5.2.8.2 Πακετοποίηση**

Η πακετοποίηση λειτουργεί μόνο όταν είναι ενεργοποιημένο το ARQ.

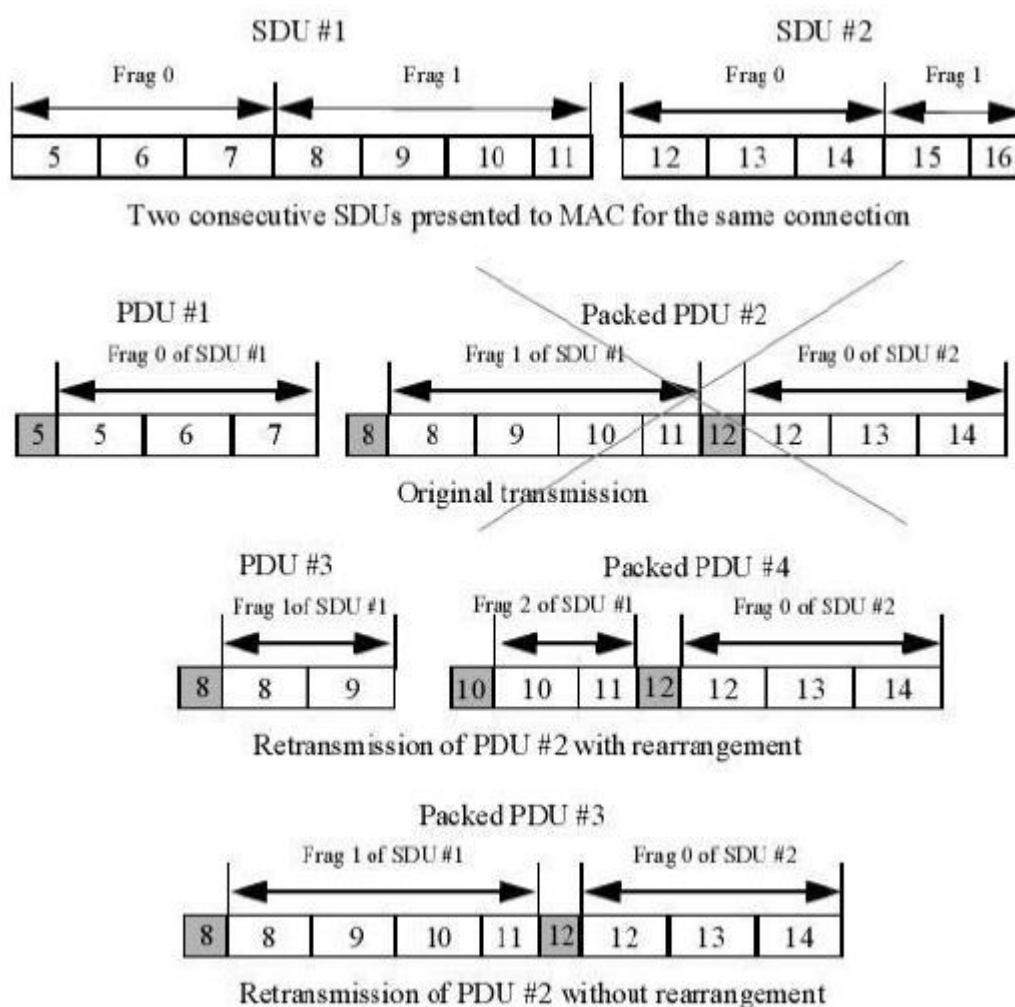
#### **5.2.9 ARQ**

Ο μηχανισμός ARQ είναι ένα κομμάτι του επιπέδου MAC και μπορεί να ενεργοποιηθεί ανά σύνδεση, όπου θα πρέπει να ορισθεί και να αρχικοποιηθεί κατά την δημιουργία της σύνδεσης.

##### **5.2.9.1 ARQ με κατακερματισμό και πακετοποίηση**

Στα όρια του ARQ συνέχεια παρουσιάζεται κατακερματισμός. Με άλλα λόγια το κομμάτι του κατακερματισμού αποτελείται πάντα από πολλά κομμάτια ARQ. Εάν είναι ενεργοποιημένη η πακετοποίηση, τα πακετοποιημένα PDUs μπορεί να

αποτελούνται από κομμάτια ARQ από πολλαπλά SDUs. Το ακόλουθο διάγραμμα παρουσιάζει τον κατακερματισμό και την πακετοποίηση των κομματιών ARQ με ή χωρίς αναδιάταξη. Στην τρέχουσα υλοποίηση υποστηρίζεται η αναδιάταξη. Η μικρότερη μονάδα στις ουρά εκπομπής και επανεκπομπής στην πλευρά του πομπού είναι το κομμάτι ARQ. Μόλις παραλάβει μια ανάθεση για ριπή δεδομένων από τον σταθμό βάσης, ο αποστολέας θα προσπαθήσει να γεμίσει την ανάθεση αυτή όσο το δυνατόν περισσότερο. Το όριο του αρχικού PDU θα ληφθεί υπ' όψιν μόνο όταν το τρέχον ARQ ανήκει σε ένα διαφορετικό MAC SDU. Σε αυτή την περίπτωση, θα χρειαστούν μερικά FSH για να σχηματίσουν νέα MAC PDUs.



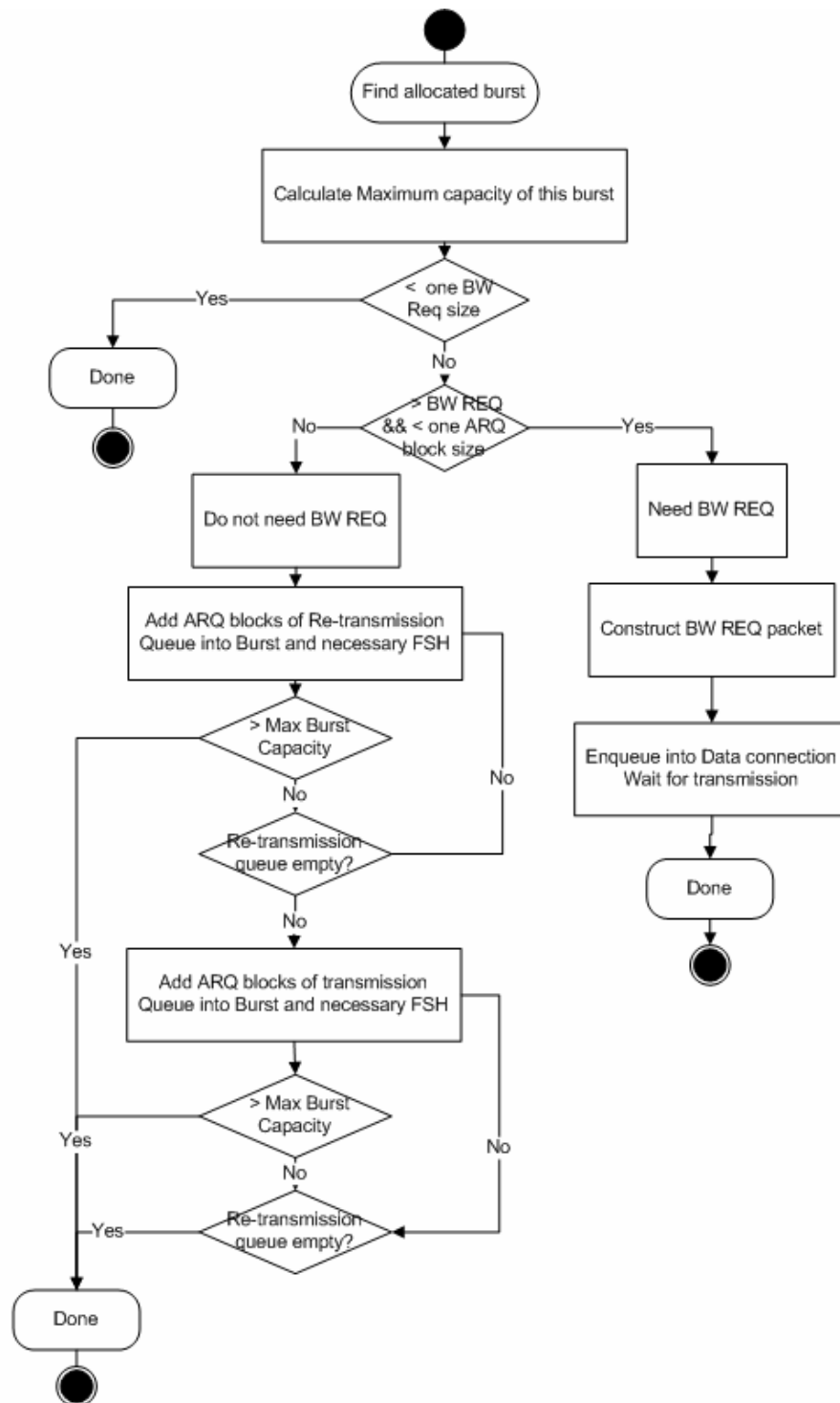
Εικόνα 38: Παραδείγματα χρήσης ARQ PDUs με ή χωρίς αναδιάταξη (WiMAX Forum module)

## 5.2.9.2 Λογική του πομπού και παραλήπτη ARQ

### 5.2.9.2.1 Λογική πομπού

Όταν τα SDUs από ένα υψηλότερο επίπεδο έρχονται στο επίπεδο MAC, θα χωριστούν σε κομμάτια ARQ σύμφωνα με το μέγεθος του κομματιού ARQ (ARQ\_BLOCK\_SIZE). Το μέγεθος του τελευταίου ARQ είναι μεταβλητό και μπορεί να είναι μικρότερο του ARQ\_BLOCK\_SIZE. Μόλις διασπαστεί το SDU, θα ενημερωθούν τα FRAG\_STATUS και SEQ\_NUM στις υποκεφαλίδες πακετοποίησης και κατακερματισμού και στην συνέχεια ο πομπός προσθέτει στην ουρά μετάδοσης του ARQ ένα αντίγραφο του πακέτου. Όταν φθάσει η ώρα της

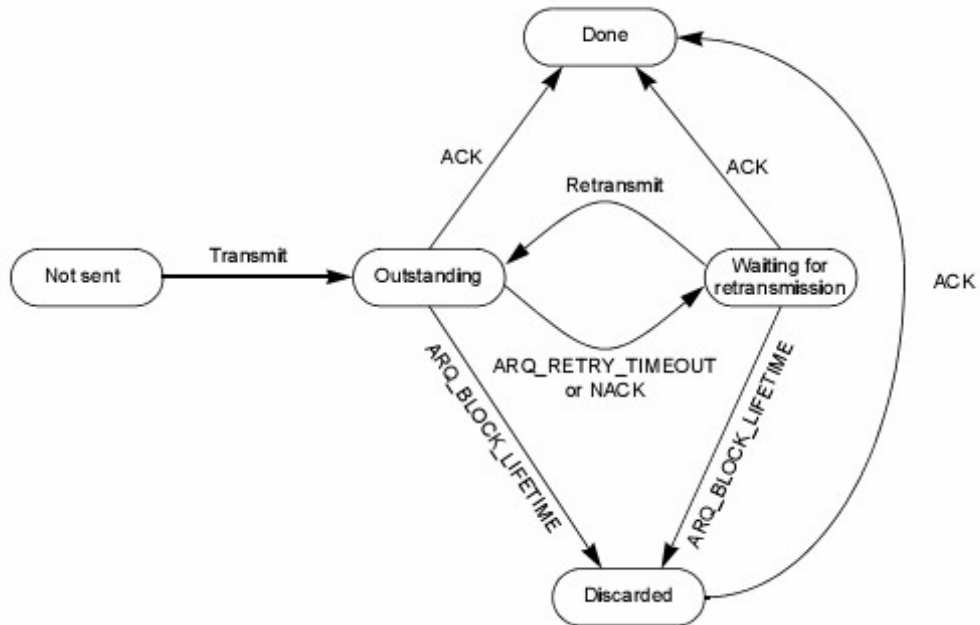
μετάδοσης, ο χρονοπρογραμματιστής θα προσπαθήσει να μεταδώσει τα κομμάτια ARQ πρώτα από την ουρά αναμετάδοσης και μετά από την ουρά μετάδοσης. Ο πομπός θα υπολογίσει πρώτα το μέγιστο μέγεθος δεδομένων που μπορεί να χωρέσει στην εκχωρημένη ριπή δεδομένων. Εάν η εκχωρημένη ριπή μπορεί να χωρέσει μόνο ένα πακέτο αιτήματος εύρους ζώνης, ο κινητός σταθμός συνδρομητής θα στείλει μια αίτηση για εύρος ζώνης ώστε να μπορέσει να στείλει τα υπόλοιπα πακέτα δεδομένων. Εάν η εκχωρημένη ριπή δεδομένων μπορεί να χωρέσει περισσότερα από ένα κομμάτια ARQ και ένα αίτημα εύρους ζώνης, το επίπεδο MAC θα στείλει τα κομμάτια ARQ καθώς και ένα αίτημα εύρους ζώνης για τα υπόλοιπα κομμάτια ARQ. Όταν το αιτούμενο εύρος ζώνης είναι έτοιμο, ο πομπός θα στείλει τα υπόλοιπα κομμάτια ARQ.



Εικόνα 39: Λογική πομπού ARQ (WiMAX Forum module)

Παρακάτω βλέπουμε την πλήρη όψη της κατάστασης μηχανής του πομπού.

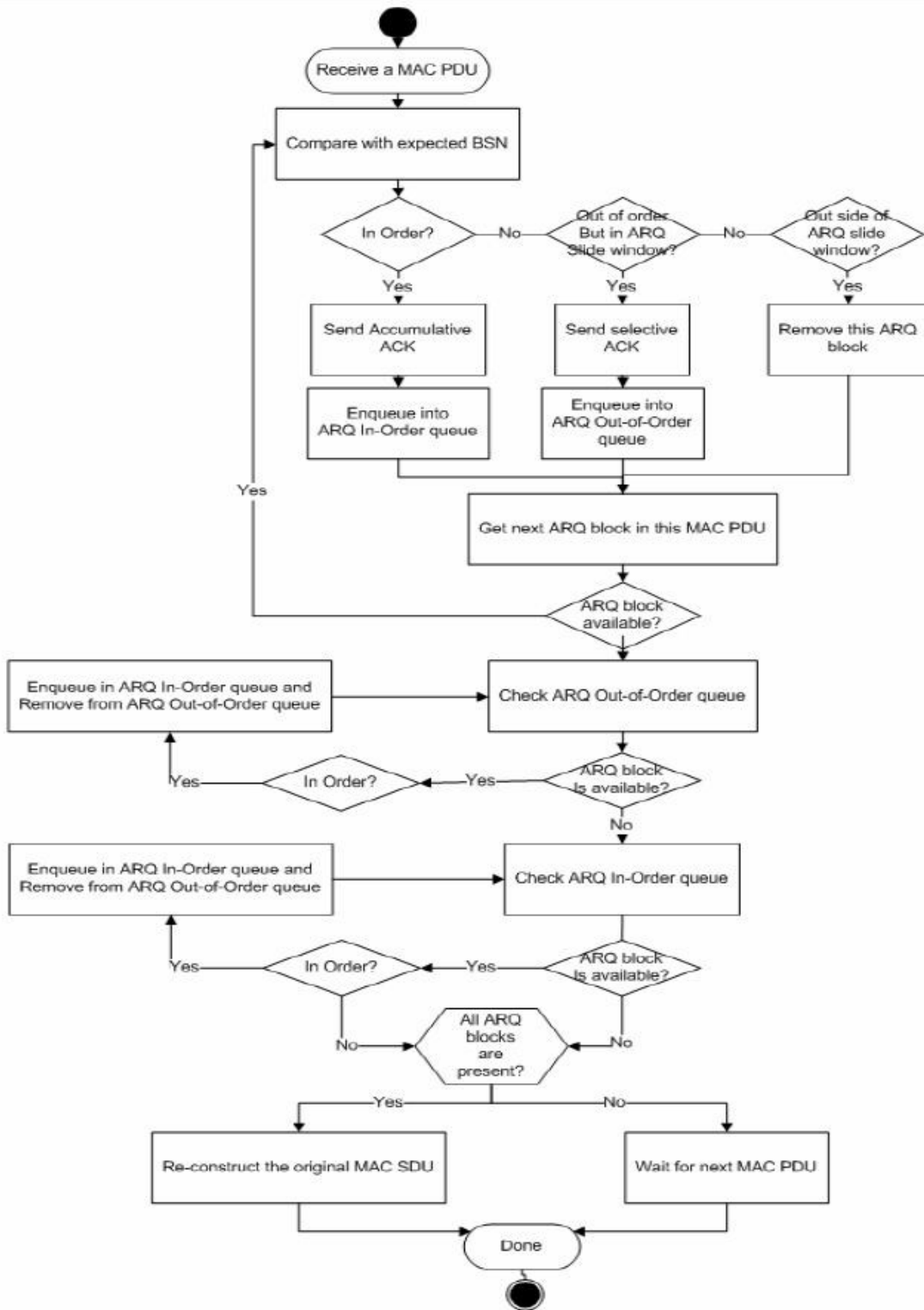




Εικόνα 40: Κατάσταση μηχανής πομπού (WiMAX Forum module)

Στην τρέχουσα υλοποίηση δεν υποστηρίζεται η κατάσταση απόρριψης, το οποίο σημαίνει ότι εάν τα κομμάτια ARQ είναι κατεστραμμένα και πρέπει να αναμεταδοθούν, ο πομπός θα αναμεταδώσει αυτά τα κομμάτια ARQ μέχρι να παραληφθούν επιτυχώς καθώς και τα ανάλογα ARQ ACK.

#### 5.2.9.2.2 Λογική παραλήπτη



Εικόνα 41: Λογική παραλήπτη ARQ (WiMAX Forum module)

Όταν το κομμάτι ARQ φτάνουν στον παραλήπτη, αυτός προσπαθεί να επαναφέρει τα MAC PDUs στο αρχικό τους μέγεθος και επιπλέον αφαιρεί την γενική κεφαλίδα και το κομμάτι CRC. Στην συνέχεια όλα τα MAC PDUs στις in-order ή out-order ουρές του ARQ σύμφωνα με το BSN που είναι αποθηκευμένο στο FSH ή στο PSH. Στην συνέχεια ο παραλήπτης θα ελέγξει εάν τα PDUs από την out-order ουρά μπορούν να χωρέσουν στην in-order ουρά αφού ένα χαμένο πακέτο μπορεί να αναμεταδοθεί στον μηχανισμό ARQ. Ο αλγόριθμος με τον οποίο διαχειρίζεται ο παραλήπτης τις in-order και out-order ουρές είναι ο ακόλουθος.

- 1) Επαναφορά του επαναληπτή και αρχή λήψης των πακέτων από την out-order ουρά.
- 2) Εάν είναι in order, συγκεντρωτική ενημέρωση ACK, μεταφέρει τα πακέτα στην in-order ουρά. Ενημέρωση του αναμενόμενου BSN από το επόμενο νέο πακέτο.
- 3) Εάν είναι out of order, φέρνει το επόμενο πακέτο να δει αν μπορεί να χωρέσει.
- 4) Η επανάληψη τα τερματιστεί όταν θα έχουν ελεγχθεί όλα τα πακέτα από τις ουρές in-order και out-order.

Το πλεονέκτημα του διαχωρισμού πακέτων σε διαφορετικές ουρές είναι ότι μειώνεται η πιθανότητα αναμετάδοσης και ταυτόχρονα είναι πιο εύκολη η αναδόμηση του αρχικού MAC SDU στην in-order ουρά.

Ο αλγόριθμος με τον οποίο το ARQ φτιάχνει τα MAC SDU είναι ο ακόλουθος. Αρχικά ο παραλήπτης διαλέγει πακέτα από την in-order ουρά αρχίζοντας από τα FLAG\_FIRST προς τα FLAG\_LAST. Εάν όλα τα κομμάτια είναι παρόντα, ο παραλήπτης δημιουργεί ένα και μόνο MAC SDU, ενημερώνει το μέγεθος και το στέλνει στο ανώτερο επίπεδο. Αφού στείλει το MAC SDU στο ανώτερο επίπεδο σβήνει όλα τα αντίστοιχα κομμάτια από την in-order ουρά. Αλλιώς ο παραλήπτης θα κρατήσει τα παραληφθέντα κομμάτια ARQ και θα περιμένει μέχρι τα αναμενόμενα κομμάτια ARQ να μπορούν να ληφθούν επιτυχώς.

### 5.2.9.2.3 Ανατροφοδότηση ARQ

Τα ΙΕ μηνύματα ανατροφοδότησης του ARQ μπορούν να μεταδοθούν στο βασικό CID ή στο CID δεδομένων. Η πρώτη ανατροφοδότηση που δίνει πάντα το ΙΕ είναι ένα αθροιστικό ACK και τα ACK που θα ακολουθούν θα είναι επιλεκτικά. Πριν την αποστολή του ACK ανατροφοδότησης, το πεδίο Num\_of\_ACKs πρέπει να ενημερωθεί. Μόλις παραληφθούν τα πακέτα ACK, ο παραλήπτης θα αλλάξει το μέγεθος του πακέτου δεδομένων και θα σβήσει τα αντίστοιχα πακέτα από την ουρά εκπομπής. Μερικές φορές όταν ένα ARQ ACK παραληφθεί, ο παραλήπτης ίσως να χρειαστεί να ελέγξει εάν το πακέτο αυτό υπάρχει και στην ουρά αναμετάδοσης και αν υπάρχει να το σβήσει και από εκεί.

### 5.2.9.2.4 Χρονισμός ARQ

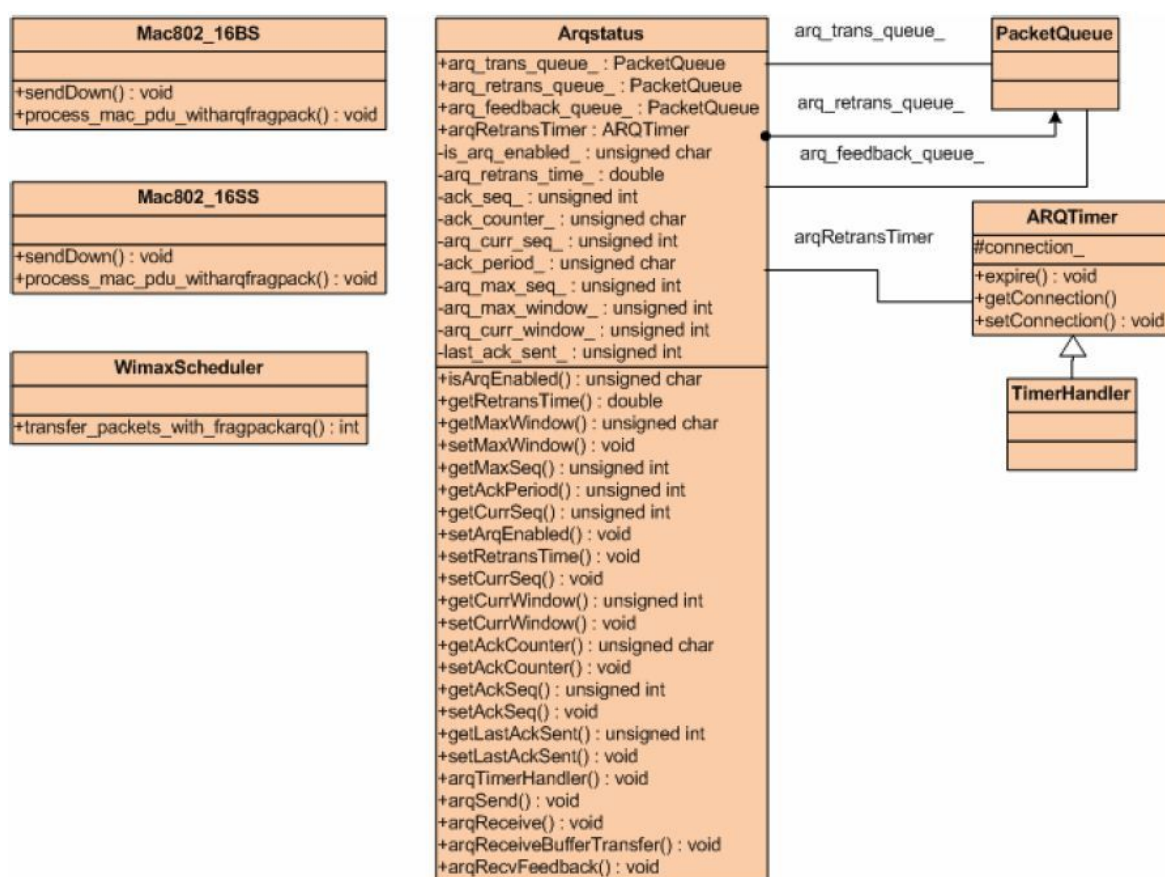
Όταν ένα αντικείμενο ARQ δημιουργείται, ένα χρονόμετρο θα ξεκινήσει με περίοδο λήξης στα 20ms. Σε κάθε λήξη, ο πομπός αντιγράφει όλα τα πακέτα από την ουρά μετάδοσης ARQ στην ουρά αναμετάδοσης και ξεκινά ξανά το χρονόμετρο. Όσον αφορά τον χρονισμό μετάδοσης των κομματιών ARQ, εάν υπάρχει αντίστοιχη εκχώρηση ριπής δεδομένων από τον σταθμό βάσης, ο πομπός θα μεταδώσει τα κομμάτια ARQ από τις ουρές αναμετάδοσης και μετάδοσης, χρησιμοποιώντας την.

### 5.2.9.3 Παράμετροι ARQ

Υπάρχουν 6 βασικές παράμετροι για τον μηχανισμό ARQ και είναι οι ακόλουθες:

1. ARQ\_BSN\_MODULUS = 2048 (δεν αλλάζει)
2. ARQ\_BLOCK\_LIFETIME = 20ms (δεν αλλάζει)
3. ARQ\_WINDOW\_SIZE = 1-1024 (αλλάζει)
4. ARQ\_BLOCK\_SIZE = 16,32,64,128,256,512,1024 (αλλάζει)
5. ARQ\_ACK\_PERIOD (αλλάζει)
6. ARQ\_ENABLE (αλλάζει)

### 5.2.9.4 Διάγραμμα κλάσεων του ARQ



Εικόνα 42: Διάγραμμα κλάσεων ARQ (WiMAX Forum module)

Η κλάση Arqstatus, οι μεταβλητές και οι συναρτήσεις της, χρησιμοποιούνται για την κατανόηση των λειτουργιών του ARQ, όπως ο μηχανισμός παραθύρου, ο μηχανισμός αναγνώρισης, ο μηχανισμός αναμετάδοσης κτλ.

<sup>35</sup><sub>17</sub> ArqSend() : Καλείται στον πομπό κατά την μετάδοση πακέτου εάν το ARQ είναι ενεργό. Ο αριθμός ακολουθίας καταγράφεται στην κεφαλίδα του 802.16 και ένα αντίγραφο του απεσταλμένου πακέτου κρατείται στο arq\_send\_queue.

<sup>35</sup><sub>17</sub> ArqReceive() : Αυτή η συνάρτηση καλείται στην μεριά του παραλήπτη όπου εάν το πακέτο που παραλήφθηκε δεν έχει σφάλμα, μια αναγνώριση μπαίνει στην ουρά ack\_queue, αλλιώς το πακέτο απορρίπτεται.

<sup>35</sup><sub>17</sub> ArqReceiveFeedback() : Αυτή η συνάρτηση καλείται στον πομπό όταν παραληφθεί η ανατροφοδότηση για το πακέτο που στάλθηκε. Η συνάρτηση

θα αυξήσει τον αριθμό ακολουθίας για το επόμενο πακέτο που περιμένει να παραλάβει και επιπλέον θα μηδενίσει το χρονόμετρο για το οποίο παραλήφθηκε η ανατροφοδότηση. Τέλος διαγράφει το πακέτο από την ουρά `arg_send_queue`.

<sup>35</sup><sub>17</sub> `ArgTimerHandler()` : Αυτή η συνάρτηση καλείται όταν το λήξει ο χρόνος μέσα στον οποίο έπρεπε να έχει παραληφθεί ανατροφοδότηση για ένα πακέτο που μεταδόθηκε σύμφωνα με τον τρέχον αριθμό ακολουθίας. Όταν συμβεί αυτό το πακέτο μεταφέρεται από την ουρά `arg_send_queue` στην ουρά `arg_retrans_queue`. Όταν αποσταλεί μια αίτηση για εύρος ζώνης, το μέγεθος του εύρους ζώνης που αιτείται είναι το άθροισμα της ουράς `arg_retrans_queue` και της ουράς όπου αποθηκεύονται τα MAC PDUs για μετάδοση. Τέλος τα πακέτα που βρίσκονται στην ουρά `arg_retrans_queue` έχουν μεγαλύτερη προτεραιότητα από τα πακέτα που βρίσκονται στην κανονική ουρά μετάδοσης όταν πρόκειται να χρονοπρογραμματιστούν.

`Mac802_16BS:: process_mac_pdu_withargfrackpack()` : Χρησιμοποιείται στον σταθμό βάσης για τον έλεγχο λήψης ARQ του uplink και για να καλεί τις διάφορες σχετικές με το ARQ λειτουργίες.

`Mac802_16SS:: process_mac_pdu_withargfrackpack()` : Χρησιμοποιείται στον σταθμό συνδρομητή για τον έλεγχο λήψης ARQ του downlink και για να καλεί τις διάφορες σχετικές με το ARQ λειτουργίες.

`WimaxScheduler:: transfer_packets_with_frackpackarg()` : Χρησιμοποιείται και στον σταθμό βάσης και στον σταθμό συνδρομητή. Κυρίως ελέγχει την μετάδοση των κομματιών ARQ καθώς και τις ουρές μετάδοσης και αναμετάδοσης του ARQ και ταυτόχρονα αντιμετωπίζει τις αιτήσεις για εύρος ζώνης που γίνονται από τον κινητό σταθμό συνδρομητή στον σταθμό βάσης.

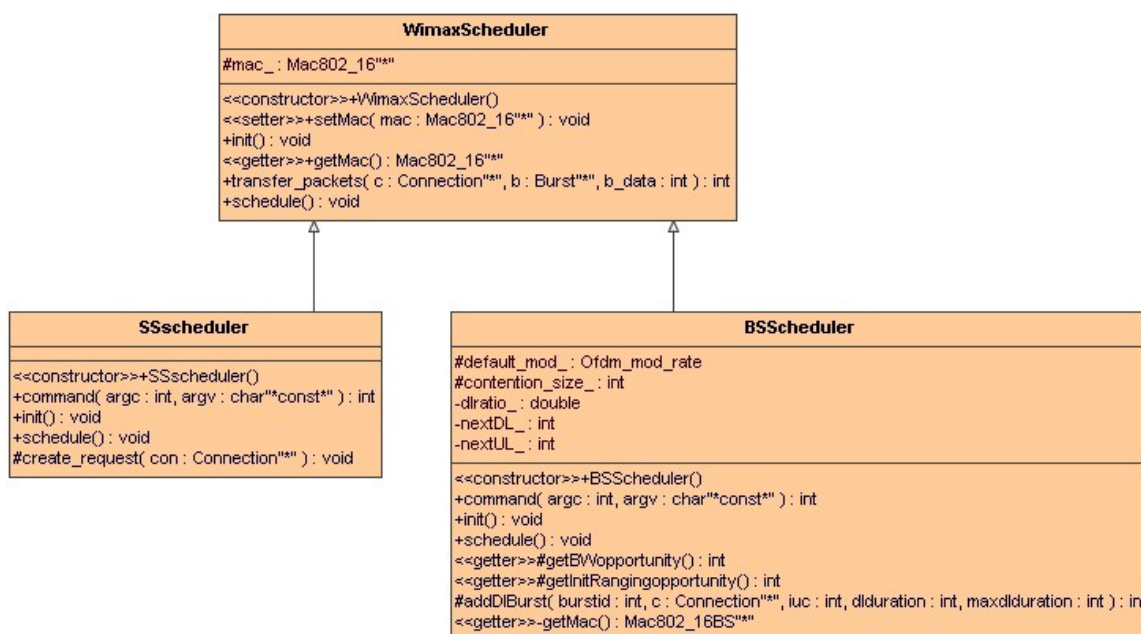
#### **5.2.10 Χρονοπρογραμματισμός υπηρεσιών**

Η δομή των κλάσεων επιτρέπει την χρήση διάφορων υπηρεσιών δεδομένων όπως οι UGS, rtGS, nrTPS και Best Effort. Ο χρονοπρογραμματισμός των πακέτων

εκτελείται από έναν χρονοπρογραμματιστή ο οποίος αλληλεπιδρά με το επίπεδο MAC μέσω ενός API το οποίο δέχεται προσαρμοσμένες υλοποιήσεις.

### 5.2.10.1 Χρονοπρογραμματιστές

Όπως είναι λογικό κάθε είδος κόμβου χρειάζεται και διαφορετικό χρονοπρογραμματιστή πακέτων. Στο IEEE 802\_16, ο σταθμός βάσης ελέγχει τις εκχωρήσεις εύρους ζώνης και υπάρχουν πάρα πολλές υλοποιήσεις. Το μοντέλο περιέχει μια αφηρημένη κλάση, την WimaxScheduler, η οποία δημιουργήθηκε για να είναι εύκολη η χρήση διαφορετικών χρονοπρογραμματιστών πακέτων. Όπως φαίνεται στην εικόνα 43 υπάρχουν ήδη δύο υλοποιήσεις μια για τους κόμβους σταθμούς και μια για τους κόμβους πελάτες, οι οποίοι μπορούν εύκολα να αντικατασταθούν με την χρήση της TCL.



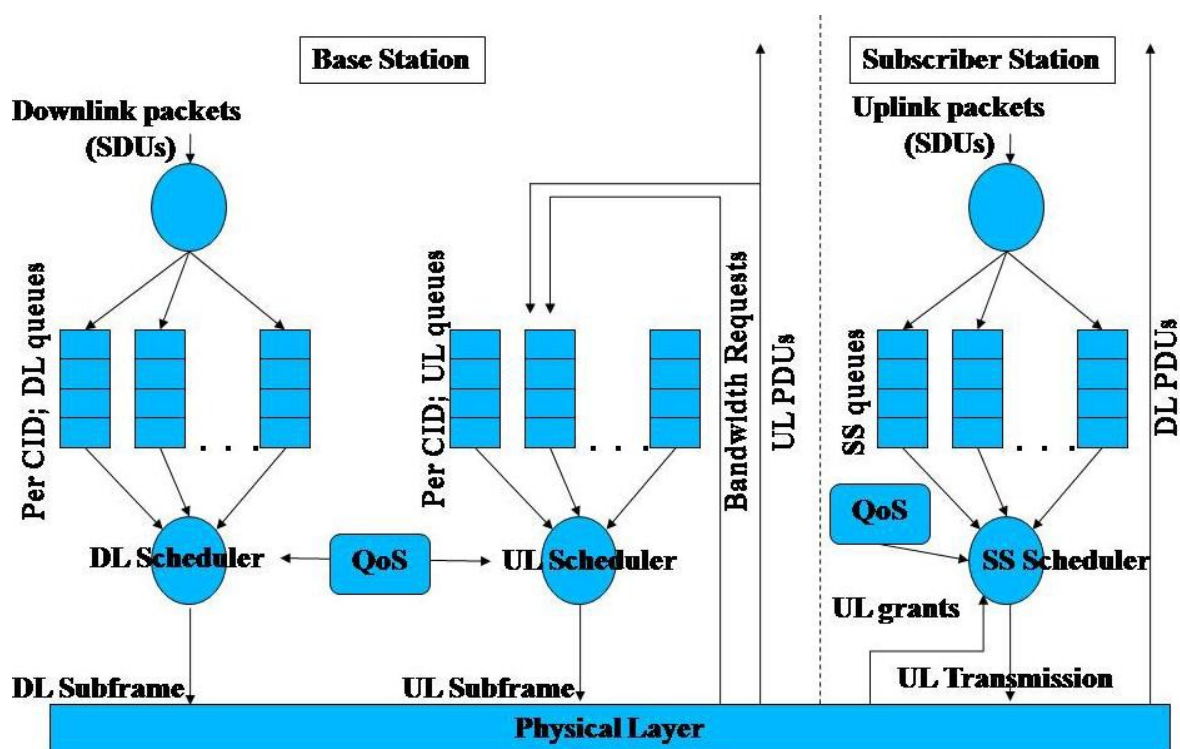
Εικόνα 43: Διάγραμμα κλάσεων χρονοπρογραμματισμού (WiMAX Forum module)

Κατά την δημιουργία ενός νέου χρονοπρογραμματιστή, οι παρακάτω μέθοδοι πρέπει να υλοποιηθούν:

- `init()` : αρχικοποιεί τον χρονοπρογραμματιστή
- `process (Packet *)` : Η μέθοδος αυτή χρησιμοποιείται για την επεξεργασία των ληφθέντων πακέτων από τον χρονοπρογραμματιστή.

- start\_ulsubframe() : Περιέχει τον κώδικα που θα εκτελεστεί στην αρχή ενός νέου υπο-πλασιού σύνδεσης εκπομπής.
- start\_dlsubframe() : Περιέχει τον κώδικα που θα εκτελεστεί στην αρχή ενός νέου υπο-πλασιού σύνδεσης λήψης.

### 5.2.10.2 Προεπιλεγμένος χρονοπρογραμματιστής



Εικόνα 44: Χρονοπρογραμματιστές στον σταθμό βάσης και στον σταθμό συνδρομητή (WiMAX Forum module)

Ο χρονοπρογραμματιστής χωρίζεται σε δύο κομμάτια, ένα για την σταθμό βάσης και ένα για τον σταθμό συνδρομητή. Το κομμάτι στον σταθμό βάσης είναι υπεύθυνο για το γέμισμα του downlink υποπλασιού και για την παραγωγή του uplink χάρτη. Το κομμάτι στον σταθμό συνδρομητή είναι υπεύθυνο για τον διαχωρισμό του εύρους ζώνης στις διάφορες συνδέσεις του.

Ανάμεσα στον χρονοπρογραμματιστή και στον υπόλοιπο κώδικα υπάρχει μια διασύνδεση, η οποία διαθέτει ένα σύνολο από παραμέτρους εισόδου και επιστρέφει την δομή του χάρτη σαν έξοδο. Οι παράμετροι εισόδου είναι οι ακόλουθοι:



<sup>35</sup><sub>17</sub> Λίστα των συνδέσεων downlink/uplink

<sup>35</sup><sub>17</sub> Αριθμός των υπο-καναλιών

<sup>35</sup><sub>17</sub> Αριθμός των συμβόλων

<sup>35</sup><sub>17</sub> Την μετατόπιση των συμβόλων

<sup>35</sup><sub>17</sub> Μια μέθοδο εκχώρησης σε υποδοχές

<sup>35</sup><sub>17</sub> Οριζόντιο ή κάθετο stripping

Η κάθε διασύνδεση επιστρέφει τον δικό της αντίστοιχο χάρτη. Στον χρονοπρογραμματιστή του downlink μια λίστα από downlink συνδέσεις περνιούνται σαν παράμετροι και το αντίστοιχο γίνεται για το uplink. Ο αριθμός των συμβόλων και των υπο-καναλιών αναπαριστά το πλήθος τους που πρέπει να εκχωρηθεί από τον χρονοπρογραμματιστή. Η μετατόπιση των συμβόλων είναι μια μετατόπιση από την οποία ξεκινά η εκχώρηση σε όλα το υπο-πλαίσιο του downlink.

#### **5.2.10.2.1 Χρονοπρογραμματιστής downlink στον σταθμό βάσης**

Ο χρονοπρογραμματιστής του downlink του σταθμού βάσης λειτουργεί με round robin, εφόσον έχει δεσμεύσει εύρος ζώνης για την βασική, την κύρια και την δευτερεύουσα σύνδεση. Στην συνέχεια αναθέτει εύρος ζώνης για τις συνδέσεις δεδομένων με την ακόλουθη σειρά: UGS, rtPS και nrtPS. Στην συνέχεια εάν υπάρχουν εναπομείναντες πόροι ο αλγόριθμος τους κατανέμει δίκαια ανάμεσα στα rtPS, nrtPS και BE. Ο αλγόριθμος αναθέτει εύρος ζώνης για μια σύνδεση μόνο όταν έχει δεδομένα να μεταφέρει.

Όσον αφορά το UGS, εάν υπάρχουν πακέτα στην ουρά υπάρχουν δύο επιλογές : πρώτον η ανάθεση γίνεται για κάθε frame και δεύτερον ο χρονοπρογραμματιστής αναθέτει το πλήρες μέγεθος του SDU για κάθε frame\_period.

Όσον αφορά το rtPS και το nrtPS εάν υπάρχουν πακέτα στην ουρά, η ανάθεση γίνεται σε κάθε frame.

Για το BE και τις επιπλέον αιτήσεις για εύρος ζώνης των rtPS και nrtPS, ο αλγόριθμος μοιράζει τους εναπομείναντες πόρους δίκαια.

#### **5.2.10.2.2 Χρονοπρογραμματιστής uplink στον σταθμό βάσης**

Ο χρονοπρογραμματισμός του uplink του σταθμού βάσης είναι παρόμοιος με τον χρονοπρογραμματισμό του downlink μόνο που αντί πακέτων στην ουρά έχουμε πληροφορίες για αιτήσεις εύρους ζώνης. Για παράδειγμα, όσον αφορά το UGS η ανάθεση γίνεται είτε σε κάθε frame είτε κάθε UGS\_period, όμως σε αντίθεση με τον χρονοπρογραμματισμό του downlink δεν υπάρχει αίτηση για εύρος ζώνης οπότε η ανάθεση θα γίνει ανεξάρτητα με το αν υπάρχουν πακέτα στην ουρά μετάδοσης του κινητού σταθμού συνδρομητή. Το εύρος ζώνης στην κατεύθυνση του uplink δεσμεύεται ανά κινητό σταθμό συνδρομητή. Αυτό πρακτικά σημαίνει ότι αν ένας κινητός σταθμός συνδρομητής έχει πολλαπλές συνδέσεις το εύρος ζώνης που του παραχωρείται είναι το άθροισμα του εύρους ζώνης για όλες τις συνδέσεις του.

Για το rtPS εάν υπάρχει αίτηση για εύρος ζώνης η ανάθεση γίνεται σε κάθε frame. Εάν δεν υπάρχει, εφόσον το rtPS δεν υποστηρίζει μηχανισμό επίλυσης διενέξεων, κατά την τρέχουσα περίοδο δεσμεύεται τουλάχιστον μια ευκαιρία για αίτηση εύρους ζώνης.

Ο χρονοπρογραμματισμός του nrtPS είναι παρόμοιος με αυτόν του rtPS. Ωστόσο, επιτρέπει τον μηχανισμό επίλυσης διενέξεων.

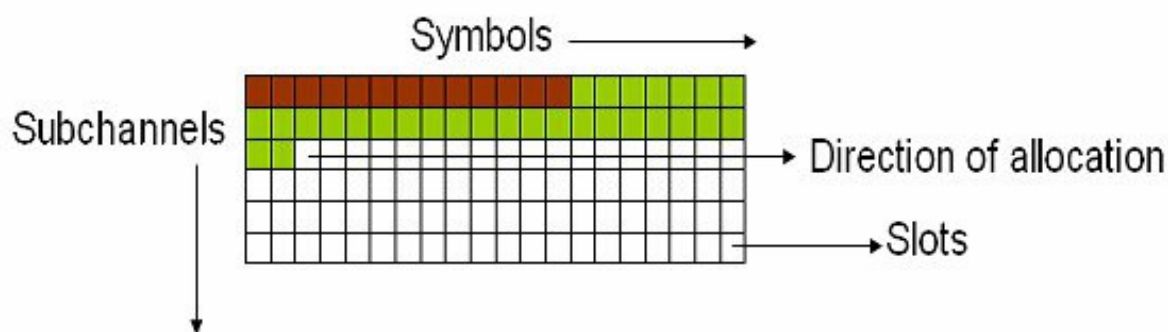
Για το BE και τις επιπλέον αιτήσεις για εύρος ζώνης των rtPS και nrtPS, ο αλγόριθμος μοιράζει τους εναπομείναντες πόρους δίκαια.

Δεδομένου του αριθμού των υποδοχών που απαιτείται/εγγρύνεται για κάθε κλάση κίνησης, οι υποδοχές είτε εκχωρούνται οριζόντια είτε κάθετα, το οποίο περνάει σαν παράμετρος στον χρονοπρογραμματιστή. Στην πραγματικότητα ο προεπιλεγμένος χρονοπρογραμματισμός για το downlink του σταθμού βάσης γίνεται κάθετα, ενώ για το uplink οριζόντια. Στην συγκεκριμένη υλοποίηση όμως και για τους δύο χρησιμοποιείται η κάθετη μέθοδος.

Μόλις το uplink του σταθμού βάσης, εγκρίνει την ανάθεση, ο χρονοπρογραμματιστής ενημερώνει την παράμετρο αιτήματος εύρους ζώνης.

### 5.2.10.3 Οριζόντιο stripping

Στο οριζόντιο stripping, οι υποδοχές εκχωρούνται χρονικά πρώτα και όταν το τελευταίο σύμβολο έχει γεμίσει η εκχώρηση προχωρά στο επόμενο υπο-κανάλι.

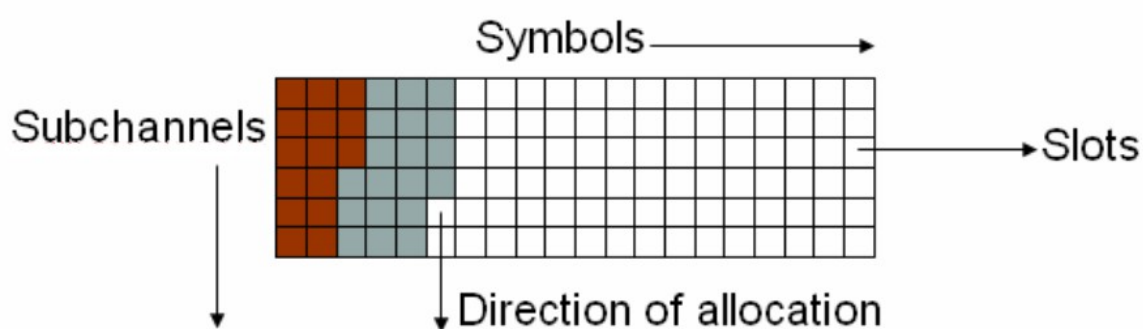


Allocation of slots in a subframe

Εικόνα 45: Οριζόντιο stripping (WiMAX Forum module)

### 5.2.10.4 Κάθετο stripping

Στο κάθετο stripping η εκχώρηση γίνεται πρώτα στα υπο-κανάλια, και όταν και το τελευταίο υπο-κανάλι γεμίσει, η εκχώρηση προχωρά στο επόμενο σύμβολο.



Εικόνα 46: Κάθετο stripping (WiMAX Forum module)

Στον κινητό σταθμό συνδρομητή υπάρχει μόνο μια μονάδα χρονοπρογραμματισμού η οποία αφορά το uplink. Οι κύριες λειτουργίες είναι η δημιουργία αιτήσεων για εύρος ζώνης και αφού δοθεί η έγκριση, εφόσον η έγκριση

δίνεται ανά κινητό σταθμό συνδρομητή, το uplink του κινητού σταθμού συνδρομητή πρέπει να διανέμει την ευκαιρία μετάδοσης στις συνδέσεις.

#### **5.2.10.5 Χρονοπρογραμματιστής uplink στον σταθμό συνδρομητή**

Το uplink του κινητού σταθμού συνδρομητή αρχικά μεταδίδει δεδομένα από την βασική, την πρωταρχική και την δευτερεύουσα σύνδεση μέχρι να αδειάσουν. Στην συνέχεια εξυπηρετεί τις συνδέσεις δεδομένων. Στην τρέχουσα υλοποίηση οι κινητοί σταθμοί συνδρομητή μπορούν να έχουν μόνο μια σύνδεση στην κατεύθυνση uplink και downlink. Ο χρονοπρογραμματιστής ελέγχει ποια σύνδεση εκπέμπει δεδομένα μέσω αυτής της σύνδεσης. Για όλες τις συνδέσεις εκτός από τις UGS και rtPS, ένα πακέτο BWR δημιουργείται το οποίο μπαίνει στην ουρά για εκπομπή. Στην συνέχεια πολλαπλά PDUs τοποθετούνται στις ριπές του uplink για εκπομπή.

#### **5.2.10.6 Αιτήσεις εύρους ζώνης και υπέρθεση (piggybacking)**

Υπάρχουν δύο μηχανισμού για να σταλούν αιτήσεις για εύρος ζώνης: ένα πακέτο αίτησης εύρους ζώνης και η υπέρθεση.

Χωρίς την υπέρθεση, μόλις δοθεί ευκαιρία για εκπομπή, αρχικά το uplink του κινητού σταθμού συνδρομητή υπολογίζει εάν όλα τα πακέτα που βρίσκονται στην ουρά μπορούν να μεταδοθούν και αν ναι, δεν εκπέμπεται πακέτο αίτησης εύρους ζώνης. Αλλιώς εκπέμπεται πρώτα ένα πακέτο αίτησης εύρους ζώνης και ακολουθούν πακέτα που είναι στην ουρά εάν αυτό είναι δυνατό.

Με την υπέρθεση, μόλις δοθεί ευκαιρία για εκπομπή, αρχικά το uplink του κινητού σταθμού συνδρομητή ελέγχει εάν η ευκαιρία είναι μεγαλύτερη από το εύρος ζώνης που αιτήθηκε. Εάν όχι, στέλνεται μόνο ένα πακέτο αίτησης εύρους ζώνης. Εάν ναι υπολογίζεται εάν όλα τα πακέτα που είναι στην ουρά μπορούν να μεταδοθούν και αν ναι, τότε δεν απαιτείται υπέρθεση. Εάν όχι η αίτηση εύρος ζώνης υπερτίθεται στο πρώτο MPDU. Εφόσον η υπέρθεση συμβαίνει στο πρώτο MPDU, η μεταβλητή αίτησης εύρους ζώνης υπολογίζεται στο περίπου από το σύνολο του μεγέθους των πακέτων στην ουρά και το μέγεθος της ευκαιρίας εκπομπής.

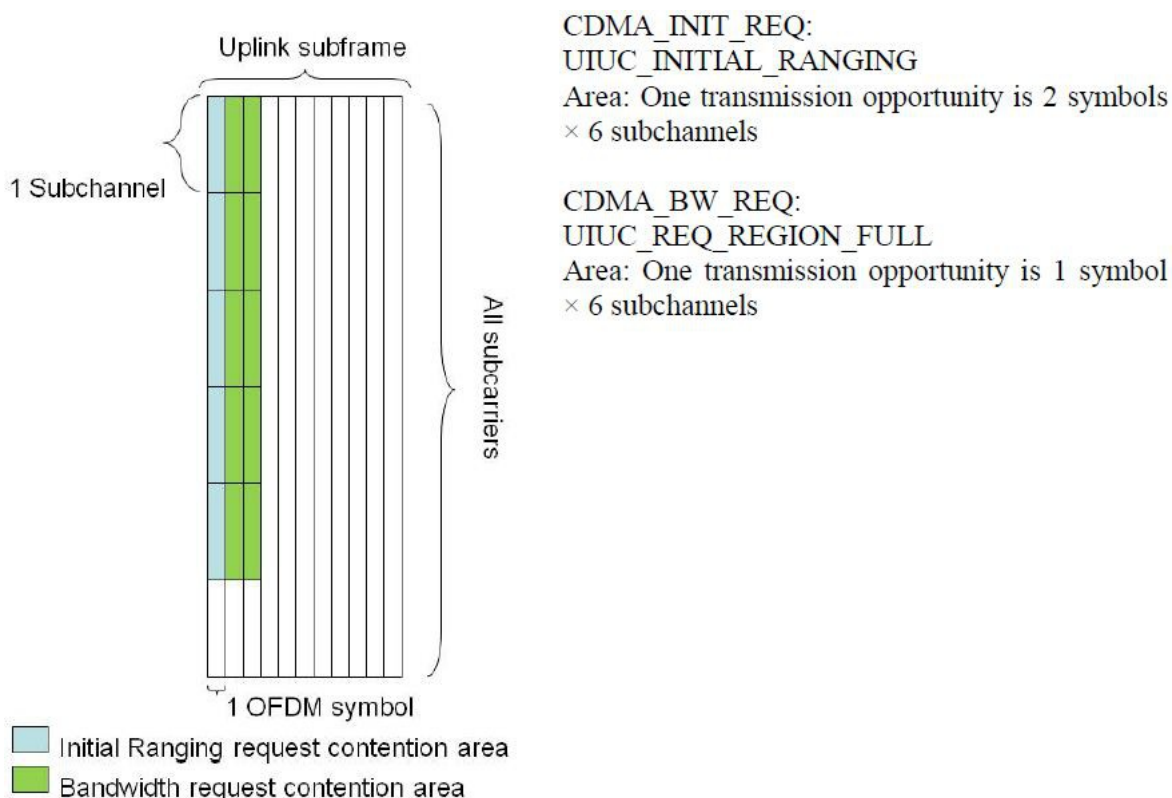
#### **5.2.11 Εκχώρηση εύρους ζώνης και μηχανισμοί αιτημάτων**

Αυτή η ενότητα περιγράφει την υλοποίηση των διάφορων μηχανισμών με τους οποίους μπορεί ένας κινητός σταθμός συνδρομητής να αιτηθεί εύρος ζώνης.

### 5.2.11.1 Τυχαία προσπέλαση βασισμένη στο CDMA

#### 5.2.11.1.1 Δομή του frame του CDMA

Η δομή της τυχαίας προσπέλασης CDMA φαίνεται στην εικόνα 47.



Εικόνα 47: Δομή του CDMA frame (WiMAX Forum module)

#### 5.2.11.1.2 Παράμετροι τυχαίας προσπέλασης βασισμένης στο CDMA

Υπάρχουν δύο κύριοι παράμετροι: ο αριθμός των CDMA\_INIT\_REQ ευκαιριών ανά frame και ο αριθμός των CDMA\_BW\_REQ ανά frame. Αυτές οι δύο παράμετροι έχουν αρχική τιμή 5. Σημειωτέον πως 5 ευκαιρίες έχουν σαν αποτέλεσμα 3 σύμβολα OFDMA για την περιοχή του CDMA. Σαν αποτέλεσμα 5 υπο-κανάλια και 2 σύμβολα μένουν αχρησιμοποίητα.

Επιπλέον υπάρχει η μεταβλητή που καθορίζει το όριο λήξης χρόνου για αναμετάδοση στο CDMA με αρχική τιμή 1 frame. Εφόσον ο κινητός σταθμός

συνδρομητής αποφασίζει εάν χρειάζεται να στείλει μια αίτηση CDMA ή όχι στην αρχή του υπο-πλασίου του uplink, η τιμή του CDMA\_TIMEOUT καθορίζεται στο 2 και μειώνεται κατά 1 για κάθε WiMAX frame. Τέλος ο μέγιστος κωδικός CDMA έχει τεθεί στο 256 για κάθε ευκαιρία εκπομπής.

#### **5.2.11.1.3 Πακέτο CDMA**

Τα σήματα CDMA αντιμετωπίζονται σαν ειδικά πακέτα (μεγέθους 6 bytes). Ωστόσο η ευκαιρία μετάδοσης είναι 1 υπο-κανάλι και 2 σύμβολα OFDM για το CDMA\_INIT\_REQ και 1 υπο-κανάλι και 1 σύμβολο OFDM για το CDMA\_BW\_REQ.

#### **5.2.11.1.4 Στάδια του CDMA**

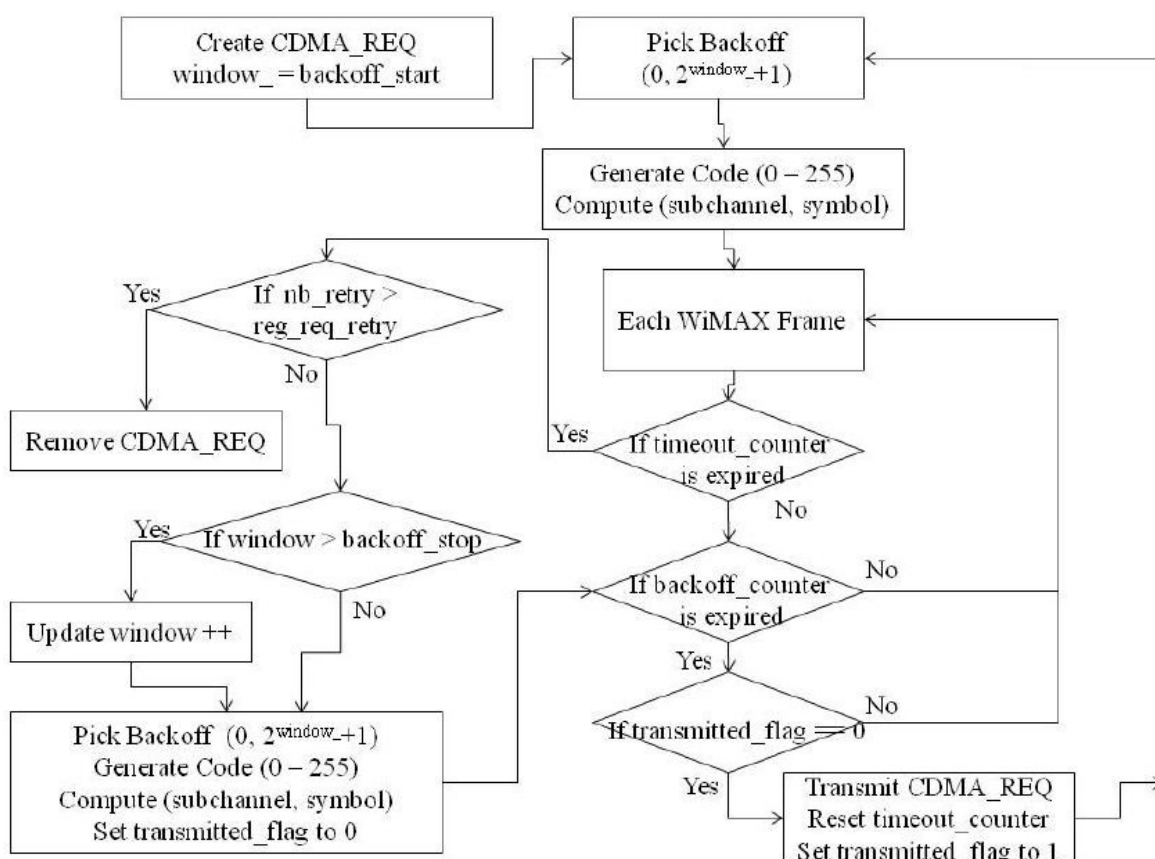
Σε αντίθεση με τα άλλα MAC πακέτα, τα πακέτα CDMA αντιμετωπίζονται σαν ειδικά πακέτα. Σαν αποτέλεσμα ο υπολογισμός ενέργειας δεν εκτελείται στις κλάσεις wimax/mac802\_16BS.cc και mobile/prop\_OFDMA.cc κάτι που γίνεται για λόγους ανίχνευσης συγκρούσεων. Ωστόσο η σύγκρουση θα ανιχνευθεί στο στάδιο χρονοπρογραμματισμού εάν ο κωδικός και η ευκαιρία μετάδοσης είναι ίδια μεταξύ δύο οι περισσοτέρων πακέτων CDMA.

Στο στάδιο χρονοπρογραμματισμού αρχικά ο σταθμός βάσης κάνει έλεγχο για συγκρούσεις αιτημάτων CDMA και αν υπάρχει σύγκρουση τότε ο απορρίπτει την αίτηση CDMA. Αλλιώς αναθέτει μια ευκαιρία αιτήματος ρύθμισης (12 bytes) για το CDMA\_INIT\_REQ ή τουλάχιστον μια ευκαιρία αιτήματος εύρους ζώνης (6 bytes) για το CDMA\_BW\_REQ.

Αυτή η έκδοση του module δεν υποστηρίζει προς το παρόν το CDMA\_MAP\_IE. Ωστόσο το CID και το ID του συνδρομητή υπάρχουν στο πακέτο CDMA. Συγκεκριμένα για το CDMA\_BW\_REQ ο σταθμός βάσης χρησιμοποιεί αυτές τις παραμέτρους για να δεσμεύσει ευκαιρίες αιτημάτων εύρους ζώνης για κάθε συνδρομητή που αιτήθηκε.

Μόλις ο σταθμός βάσης δεσμεύσει την ευκαιρία μετάδοσης ή την ευκαιρία αίτησης εύρους ζώνης στον κινητό σταθμό συνδρομητή, αρχικά ελέγχει εάν ο κωδικός είναι ίδιος με αυτόν που αιτήθηκε και αν είναι ίδιος ο κινητός σταθμός συνδρομητής

μεταδίδει ένα μήνυμα ρύθμισης. Όσον αφορά το CDMA\_BW\_REQ, εφόσον η ανάθεση αντιμετωπίζεται σαν μια συνηθισμένη εκχώρηση, ο κινητός σταθμός συνδρομητής ελέγχει πρώτα εάν μπορεί να μεταδώσει όλα τα πακέτα που βρίσκονται στην ουρά. Επιπλέον ο κινητός σταθμός συνδρομητής μεταδίδει και άλλα πακέτα μετά το αίτημα για εύρος ζώνης εφόσον αυτό είναι εφικτό.



Εικόνα 48: Βήματα αίτησης CDMA (WiMAX Forum module)

### 5.2.11.1.5 Χρονόμετρα CDMA

Υπάρχουν κυρίως δύο χρονόμετρα, τα  $t_6$  και  $t_{16}$ . Το  $t_6$  είναι ένα χρονικό όριο για αιτήματα ρύθμισης. Σε αυτή την έκδοση εάν ο κινητός σταθμός συνδρομητής στείλει αιτήματα ρύθμισης, θα ορίσει την τιμή του `timeout_timer` στην τιμή του  $t_6$  και έτσι εάν η διαδικασία ρύθμισης δεν είναι επιτυχής μέσα στα χρονικά περιθώρια του  $t_6$ , το CDMA\_INIT\_REQ θα μεταδοθεί ξανά.

Το  $t_{16}$  είναι ένα χρονικό όριο για αιτήματα εύρους ζώνης. Παρόμοια με το  $t_6$  σε αυτή την έκδοση εάν ο κινητός σταθμός συνδρομητής στείλει ένα αίτημα εύρους

ζώνης, θα ορίσει την τιμή του `timeout_timer` στην τιμή του `t16` και έτσι εάν ο κινητός σταθμός συνδρομητής δεν καταφέρει να στείλει κανένα πακέτο μέσα στο χρονικό αυτό περιθώριο θα γίνει αναμετάδοση του `CDMA_BW_REQ`.

#### **5.2.11.1.6 Αφαίρεση από την ουρά του CDMA**

Για να αφαιρέσουμε ένα πακέτο CDMA από την ουρά αιτημάτων CDMA, όσον αφορά το `CDMA_INIT_REQ` εάν η διαδικασία ρύθμισης ήταν επιτυχής, το `CDMA_INIT_REQ` θα αφαιρεθεί. Όσον αφορά το `CDMA_BW_REQ` εάν ο κινητός σταθμός συνδρομητής μπορεί να στείλει μερικά πακέτα προς τα έξω, το `CDMA_BW_REQ` αφαιρείται.

#### **5.2.11.1.7 Επίλυση συγκρούσεων**

Ο σταθμός βάσης αναθέτει υποδοχές που υπόκεινται σε συγκρούσεις στην κατεύθυνση του `urllink`. Αυτές οι υποδοχές χρησιμοποιούνται σε δύο περιπτώσεις:

- Αίτημα αρχικής ρύθμισης
- Αίτημα εύρους ζώνης

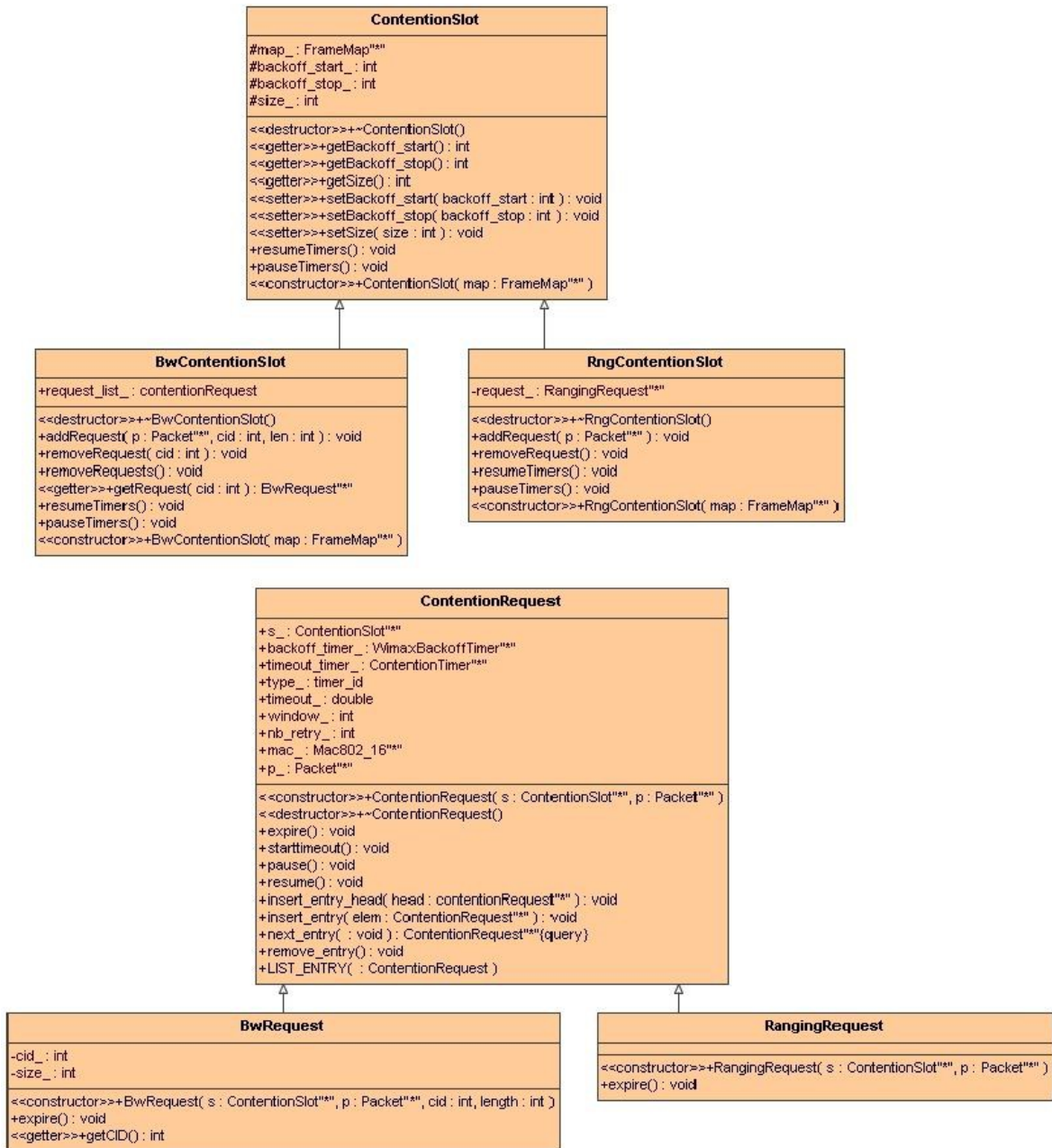
Το μοντέλο υποστηρίζει μια κατατετημημένη δυαδική εκθετική υποχώρηση (`truncated binary exponential backoff`) για επίλυση συγκρούσεων. Τα μηνύματα UCD που εκπέμπονται από τον σταθμό βάσης περιέχουν τα μεγέθη των παραθύρων. Ο σταθμός βάσης επιπλέον αποφασίζει τον αριθμό των υποδοχών που θα χρησιμοποιούνται για κάθε `frame`.

Η εικόνα 49 παρουσιάζει το διάγραμμα κλάσεων που χρησιμοποιείται στην επίλυση συγκρούσεων. Ένα υποπλαίσιο του `urllink` περιέχει μια κλάση `BwContentionslot` και μια `RngContentionSlot`. Και οι δύο είναι υποκλάσεις της `ContentionSlot` η οποία παρέχει τους βασικούς μηχανισμούς που σχετίζονται με τις συγκρούσεις.

Κατά την είσοδο στο δίκτυο, ο κινητός σταθμός συνδρομητής πραγματοποιεί μια ρύθμιση για να προσαρμόσει την ισχύ μετάδοσής του. Στην διάρκεια αυτής της διαδικασίας παράγει ένα αντικείμενο τύπου `RangingRequest`. Ο κινητός σταθμός συνδρομητή παίρνει ένα τυχαίο `backoff` από τα παράθυρα που παρέχονται από



τον σταθμό βάσης και το αποθηκεύει. Στην συνέχεια μειώνει τον μετρητή κάθε φορά που μια υποδοχή σύγκρουσης βρίσκεται μέσα στο frame. Όταν ο μετρητής φτάσει το 0, το πακέτο μεταδίδεται.

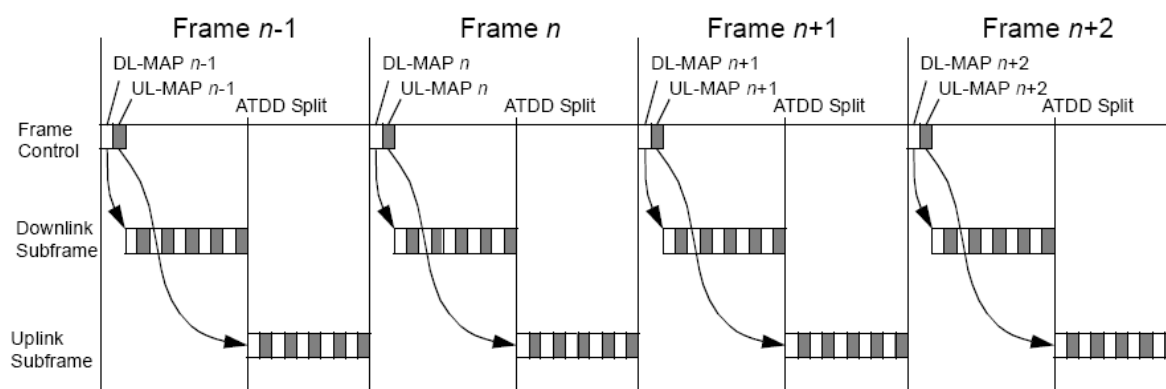


Εικόνα 49: Υποδοχές σύγκρουσης και αιτήματα σύγκρουσης (WiMAX Forum module)

### 5.2.12 Υποστήριξη του PHY στο επίπεδο MAC

Το μοντέλο προς το παρόν υποστηρίζει το TDD. Σε αυτή τη λειτουργία, οι μεταδόσεις της σύνδεσης εκπομπής γίνονται μετά την σύνδεση λήψης για κάθε πλαίσιο.

Τα μηνύματα DL\_MAP και UL\_MAP που στέλνει κάθε πλαίσιο καθορίζουν την κατανομή καταιγισμού και τις δυνατότητες μετάδοσης για κάθε εξυπηρετητή. Οι πληροφορίες που περιέχονται στο UL\_MAP ανήκουν στο ίδιο πλαίσιο όπως φαίνεται στην εικόνα 50.

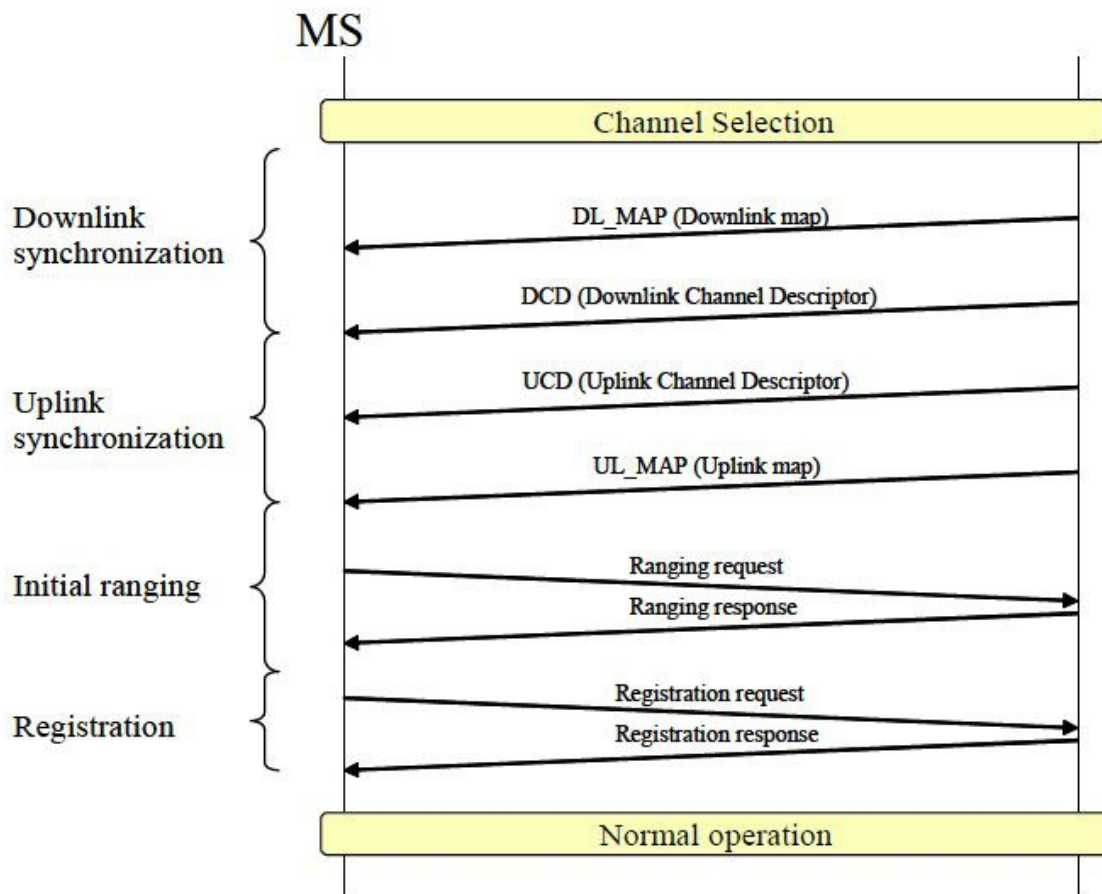


Εικόνα 50: Χρονική σχέση του DL\_MAP και του UL\_MAP (WiMAX Forum module)

### 5.2.13 Είσοδος σε δίκτυο και αρχικοποίηση

Όταν ένας σταθμός συνδρομητής θέλει να συνδεθεί σε ένα δίκτυο πρέπει να εκτελέσει μια διαδικασία εισόδου σε δίκτυο. Όπως φαίνεται στην εικόνα 51 το μοντέλο υλοποιεί τα παρακάτω συστατικά σε κάθε είσοδο σε δίκτυο:

- Σάρωση του καναλιού σύνδεσης εκπομπής
- Απόκτηση παραμέτρων μετάδοσης
- Αρχική τηλεμετρία
- Εγγραφή



Εικόνα 51: Είσοδος σε δίκτυο (WiMAX Forum module)

Οι παρακάτω παράμετροι μπορούν να οριστούν:

- Χρονόμετρα που θα εκτελούν σάρωση του καναλιού
- Η συχνότητα των μηνυμάτων DCD/UCD
- Παράμετροι για την αρχική τηλεμετρία
- Εκχώρηση καναλιού

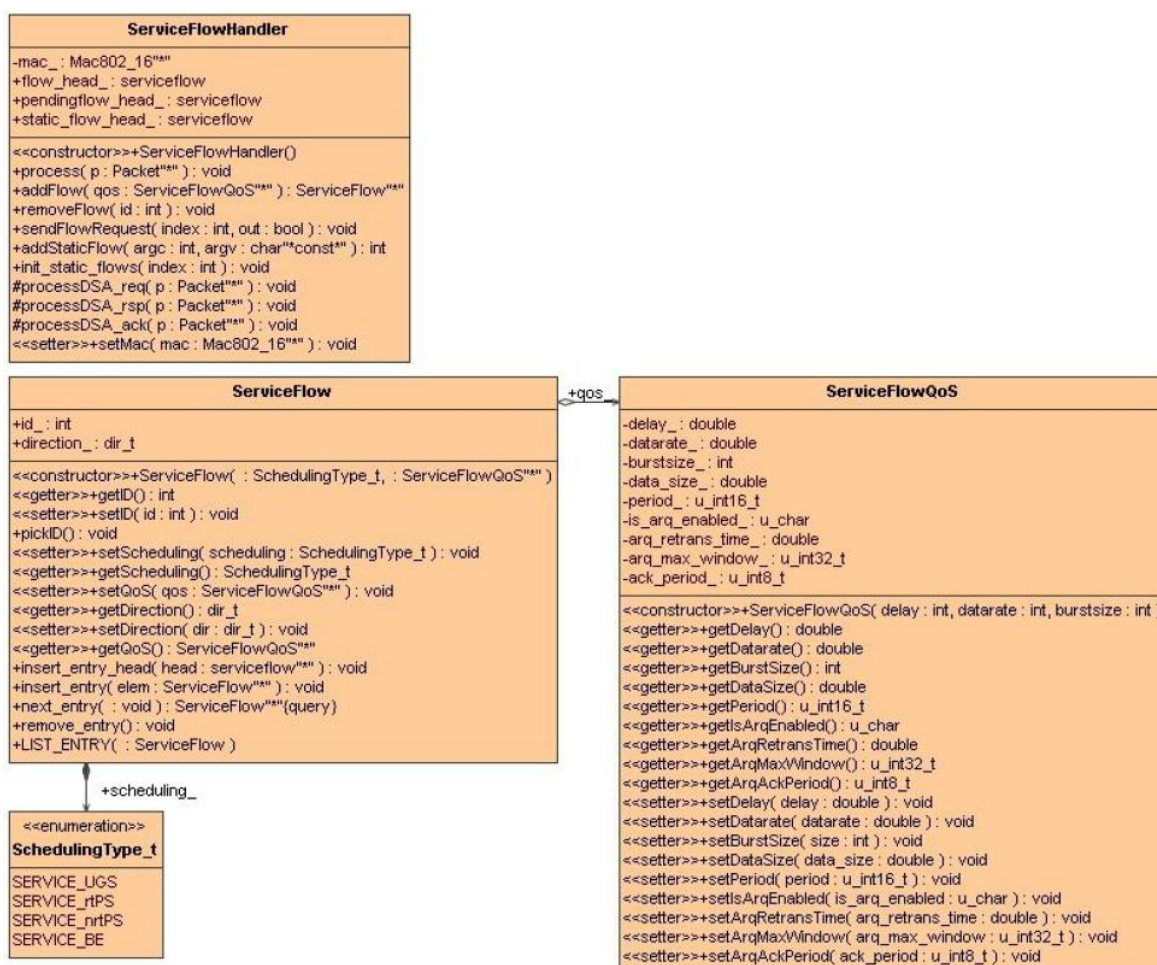
#### 5.2.14 Τηλεμετρία

Η τηλεμετρία είναι ένας μηχανισμός που επιτρέπει στην σταθμό συνδρομητή να διατηρήσει μια καλή ποιότητα σύνδεσης ρυθμίζοντας την ισχύ εκπομπής του και την διαμόρφωσή του.

Κατά την αρχική τηλεμετρία, οι σταθμοί συνδρομητές χρησιμοποιούν το κατ' εξοχήν προφίλ DIUC για να εκπέμπουν, κάτι που επιτρέπει την προσομοίωση του να εκπέμπουν οι κόμβοι σε διαφορετικούς ρυθμούς.

### 5.2.15 Πολιτική QoS

Κάθε σύνδεση (class Connection) μπορεί να συνδεθεί με μια ροή υπηρεσιών (class ServiceFlow) και τις αντίστοιχες παραμέτρους του QoS όπως φαίνεται στην εικόνα 52.



Εικόνα 52: Ροή υπηρεσιών (WiMAX Forum module)

Ο χρήστης ρυθμίζει την λίστα των ροών του κάθε κινητού σταθμό συνδρομητή μέσω της TCL. Αυτές οι διαταγμένες ροές αποθηκεύονται στην κλάση ServiceFlowHandler ως στατικές συνδέσεις και καθορίζονται κάθε φορά που ο κινητός σταθμός συνδρομητή συνδέεται σε ένα νέο σταθμό βάσης.

Ενώ η δομή υποστηρίζει τον ορισμό ροών QoS, ο χρονοπρογραμματιστής κάνει χρήση αυτών των πληροφοριών.

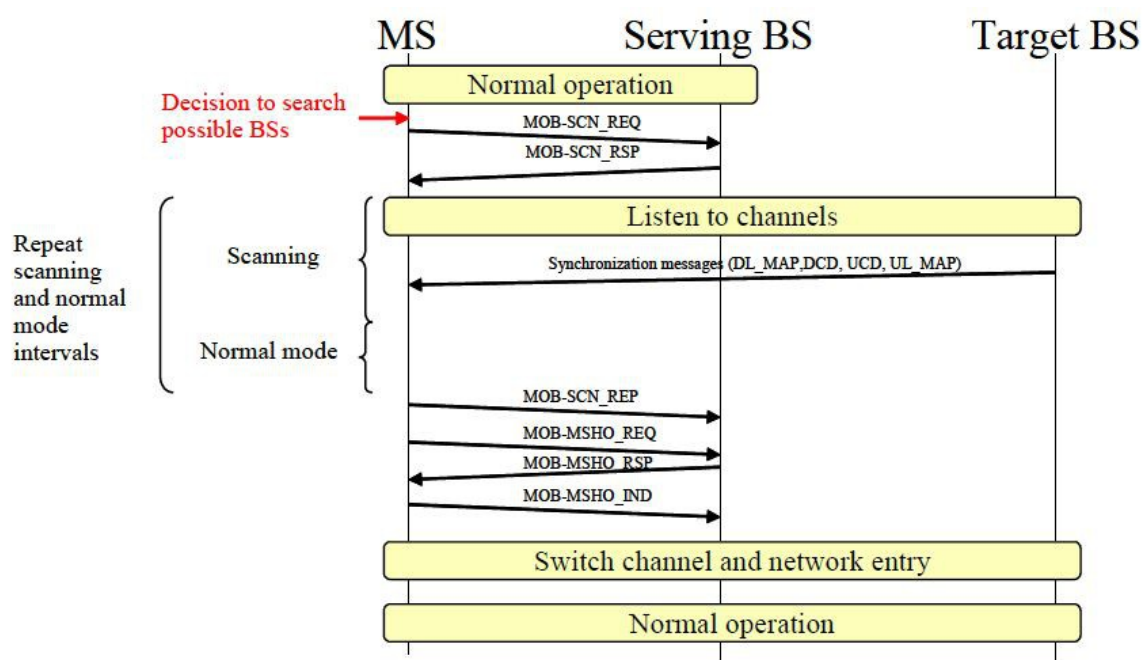
Η κλάση ServiceFlowHandler είναι υπεύθυνη για την δημιουργία των ροών. Η προεπιλεγμένη υλοποίηση δεν παρέχει κάποιον μηχανισμό ελέγχου εισόδου. Δέχεται όλα τα αιτήματα ροών από τους σταθμούς συνδρομητές.

#### **5.2.16 Διαδικασίες παράδοσης του επιπέδου MAC**

Το μοντέλο υποστηρίζει κινητικότητα 2<sup>ου</sup> επιπέδου. Ανάλογα με την διαμόρφωση, ένας κινούμενος σταθμός συνδρομητής μπορεί να εκτελέσει σάρωση και παράδοση με τον σταθμό βάσης. Σε αυτή τη ενότητα παρουσιάζεται η παράμετροι της διαμόρφωσης που επηρεάζουν την ικανότητα παράδοσης.

##### **5.2.16.1 Σάρωση**

Όταν η ποιότητα την σύνδεσης επιδεινώνεται ο κινούμενος σταθμός συνδρομητής μπορεί να στείλει ένα αίτημα MOB-SCN\_REQ στον σταθμό βάσης υπηρεσιών για να ζητήσει μια σάρωση του δικτύου για να βρει τους υπόλοιπους κόμβους εξυπηρετητές στην εμβέλειά του. Στην εικόνα 53 φαίνεται η αλληλουχία μηνυμάτων που ανταλλάσσονται κατά την διαδικασία σάρωσης.



Εικόνα 53: Διαδικασία σάρωσης (WiMAX Forum module)

Για να πυροδοτήσει την αποστολή ενός MOB-SCN\_REQ, ο κινητός σταθμός συνδρομητής παρακολουθεί το επίπεδο του σήματος στα εισερχόμενα πακέτα και όταν αυτό φτάσει ένα κατώφλι, το μήνυμα αποστέλλεται. Σαν προεπιλογή η σάρωση δεν είναι ενεργή. Για να ενεργοποιηθεί, πρέπει να αλλάξουμε την τιμή της μεταβλητής *Igd\_factor* της MIB σε μια τιμή μεγαλύτερη του 1. Όσο μεγαλύτερη η τιμή τόσο πιο σύντομα θα πυροδοτηθεί η σάρωση.

Κατά τη διάρκεια της σάρωσης, ο κινητός σταθμός συνδρομητής συλλέγει τιμές RSSI από τα εισερχόμενα πακέτα, οι οποίες χρησιμοποιούνται από τον σταθμό βάσης υπηρεσιών για να επιλεγεί ο καλύτερος στόχος σταθμός βάσης.

Αφού ο κινητός σταθμός συνδρομητής λάβει κάποια ένδειξη από τον επιλεγμένο σταθμό βάσης, περιμένει μερικά frames για να δείξει την πρόθεσή του να εκτελέσει παράδοση. Η καθυστέρηση αυτή έχει σκοπό να επιτρέψει την ανταλλαγή της κίνησης, που έχει αποθηκευτεί στους buffers κατά την διαδικασία σάρωσης, πριν γίνει η αλλαγή του κόμβου εξυπηρετητή.

Στο μοντέλο έχουν υλοποιηθεί διάφοροι τρόποι σάρωσης:

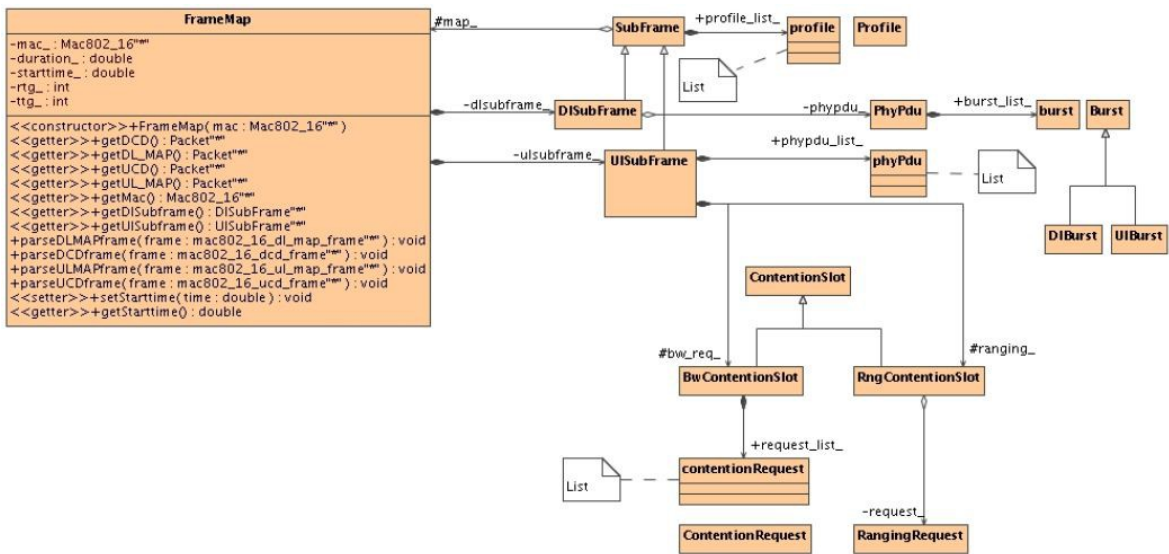
- Στην σάρωση χωρίς συσχέτιση, κινητός σταθμός συνδρομητής προσπαθεί να αναγνωρίσει και να συγχρονιστεί με έναν ή περισσότερους κόμβους εξυπηρετητές. Επιπλέον εκτιμά την ποιότητα του σήματος.
- Στο επίπεδο συσχέτισης 0, ο στόχος σταθμός βάσης δεν έχει πληροφορίες για τον κινητό κόμβο πελάτη που σαρώνει και παρέχει μόνο εκχωρήσεις παράδοσης με βάση ισχυρισμούς. Μόλις ένας κινητός σταθμός συνδρομητής αποστείλει μια αίτηση παράδοσης, περιμένει απάντηση από τον σταθμό βάσης με ένα εξ' ορισμού όριο χρόνου των 50ms για την ακύρωση της αίτησης.
- Στο επίπεδο συσχέτισης 1, ο σταθμός βάσης που παρέχει υπηρεσίες, διαπραγματεύεται με τον στόχο σταθμό βάσης μόλις ο κινητός σταθμός συνδρομητής βρει μια αφιερωμένη περιοχή παράδοσης. Μόλις ένας κινητός σταθμός συνδρομητής αποστείλει μια αίτηση παράδοσης, περιμένει απάντηση από τον σταθμό βάσης με ένα εξ' ορισμού όριο χρόνου των 50ms για την ακύρωση της αίτησης.

Το επίπεδο συσχέτισης 2, δεν υποστηρίζεται προς το παρόν.

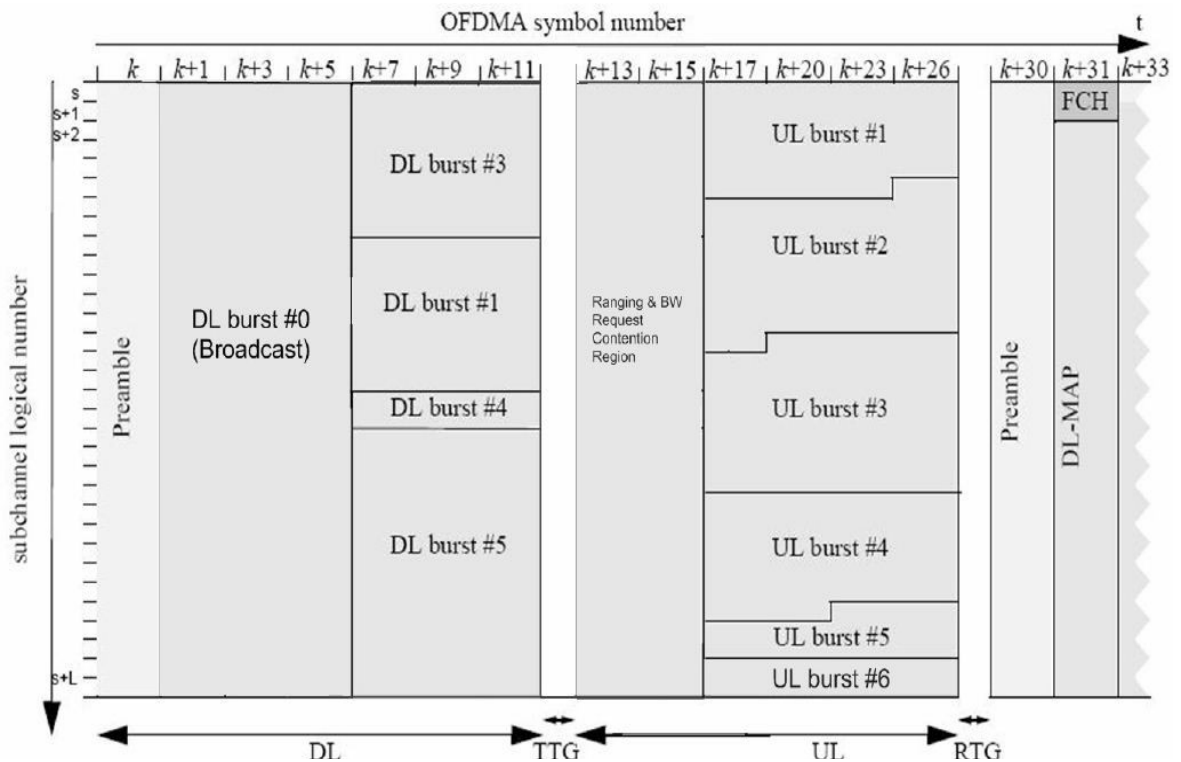
Για να επιτραπούν αυτοί οι διαφορετικοί τρόπου σάρωσης και για να γίνουν γρήγορες παραδώσεις, η κλάση WiMAXCtrlAgent απαιτείται, η οποία είναι μια κλάση τύπου Agent που εκτελεί τρεις διαδικασίες. Η πρώτη είναι να ανταλλάσσει πληροφορίες DCD/UCD μεταξύ των γειτονικών κόμβων εξυπηρετητών. Η δεύτερη είναι να πυροδοτεί την αποστολή των NBR-ADV μηνυμάτων στους κινητούς κόμβους πελάτες. Η τρίτη είναι να συγχρονίζει τον σταθμό βάσης υπηρεσιών με τον στόχο σταθμό βάσης όταν εκτελείται σάρωση επιπέδου 1 ή 2. Τα μηνύματα αυτά ανταλλάσσονται μέσω καλωδίων με την χρήση κοινών IP πακέτων.

### **5.2.17 Η δομή του frame**

Πτυχιακή του φοιτητή Καρανίκας Γεωργίου



Εικόνα 54: Διάγραμμα κλάσεων του frame (WiMAX Forum module)



Εικόνα 55: Δομή του TDD frame (WiMAX Forum module)

Ο σχεδιασμός που χρησιμοποιήθηκε για να αναπαρασταθεί ένα frame μοιάζει πολύ με την δομή που έχει οριστεί στο IEEE 802.16 για το TDD. Ένα frame αποτελείται από ένα υπο-πλαίσιο uplink και ένα downlink, τα οποία με την σειρά τους διαχωρίζονται σε PHY PDU διαστήματα. Σε κάθε ένα από αυτά τα



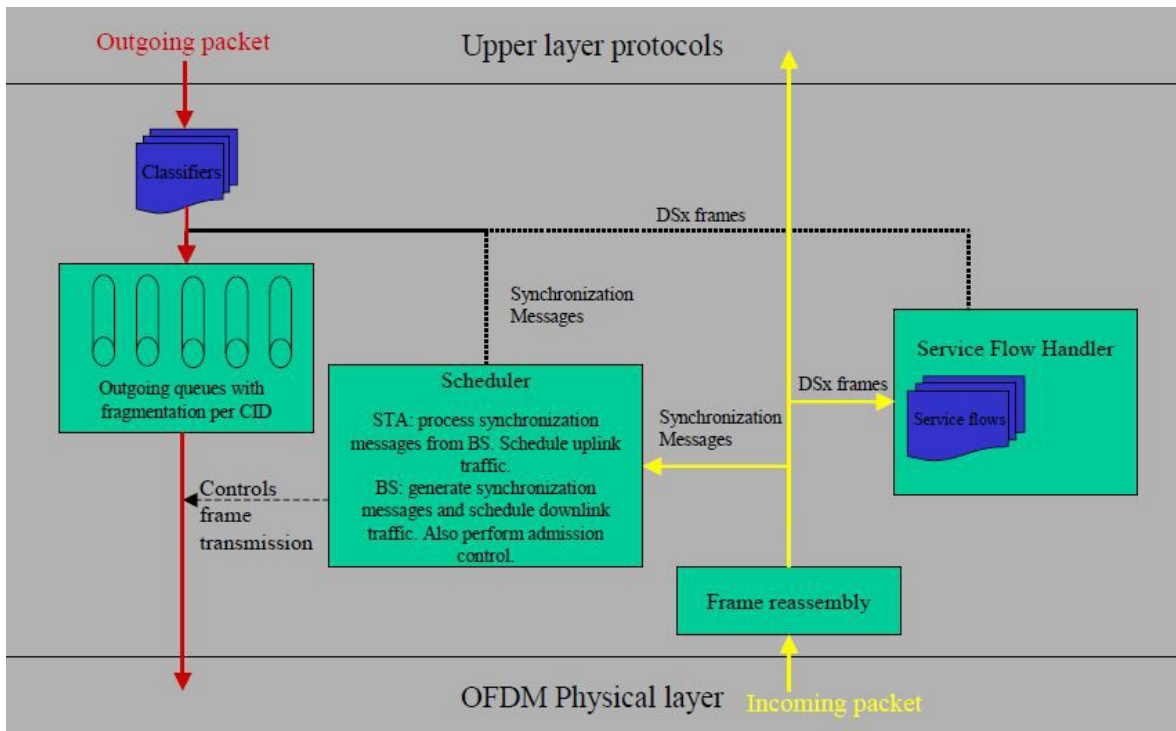
διαστήματα, το εύρος ζώνης εκχωρείται σε ριπές για τους διάφορους σταθμούς. Κάθε μια από αυτές τις ριπές μπορεί να έχει διαφορετική διαμόρφωση και συχνότητα τα οποία ονοματίζονται προφίλ.

Κανονικά ο σταθμός βάσης αναθέτει εύρος ζώνης για έναν σταθμό ώστε να εκπέμψει τα δεδομένα του. Σε μερικές περιπτώσεις, κυρίως κατά την αρχική ρύθμιση και στις αιτήσεις εύρους ζώνης, οι συμπληρωματικές υπηρεσίες πρέπει να ανταγωνιστούν η μια την άλλη για να αποκτήσουν πρόσβαση στο μέσο. Αυτά τα διαστήματα παρουσιάζονται μόνο στο uplink καθώς ο σταθμός βάσης έχει πλήρη έλεγχο στην κίνηση του downlink.

Η κλάση FrameMap επιπλέον περιέχει μεθόδους για την εξαγωγή και ανάγνωση των μηνυμάτων ελέγχου. Στο σταθμό βάσης ο χρονοπρογραμματιστής δημιουργεί την δομή του χάρτη σύμφωνα με έναν αλγόριθμο ανάθεσης, και στην συνέχεια καλεί τις μεθόδους getDL\_MAP, getUL\_MAP, getDCD και getUCD, για να λάβει τα πακέτα που περιέχουν τις απαραίτητες πληροφορίες που πρέπει να σταλούν στις συμπληρωματικές υπηρεσίες. Στις συμπληρωματικές υπηρεσίες, ο χρονοπρογραμματιστής καλεί τις αντίστροφες συναρτήσεις, parsed\_MAP, parseUL\_MAP, parseDCD και parseUCD για να αναδομήσει την απαραίτητη δομή για να μπορέσει να πράξει σωστή λήψη και εκπομπή πακέτων.

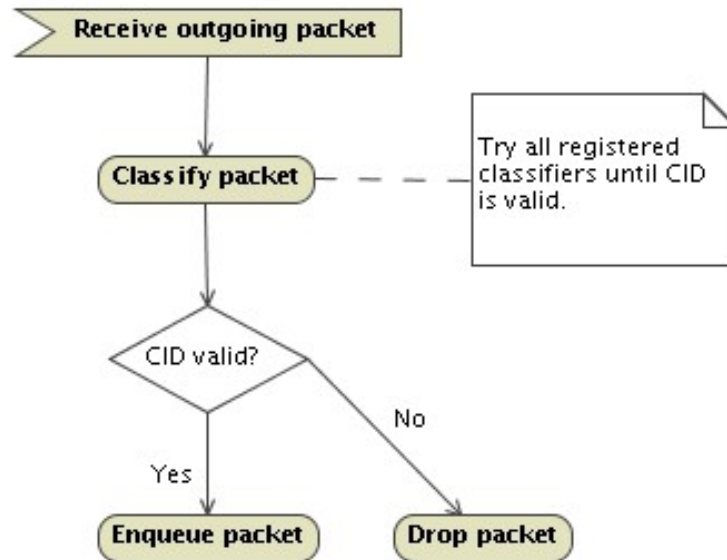
### **5.2.18 Διαχείριση πακέτων**

Η εικόνα 56 απεικονίζει τις ροές των πακέτων για τα εισερχόμενα και τα εξερχόμενα πακέτα.



Εικόνα 56: Επισκόπηση της διαχείρισης πακέτων (WiMAX Forum module)

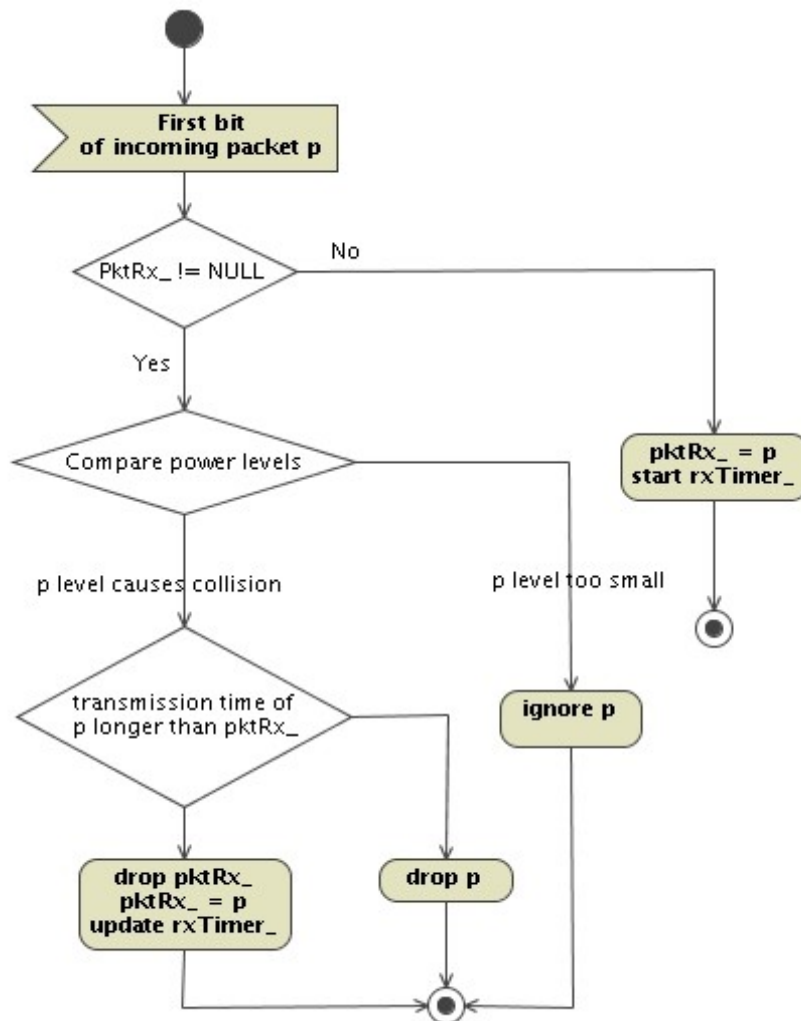
Οι εικόνες 53,54 και 55 παρέχουν μερικές αναλυτικές πληροφορίες για το πώς τα πακέτα περνάνε από το επίπεδο MAC.



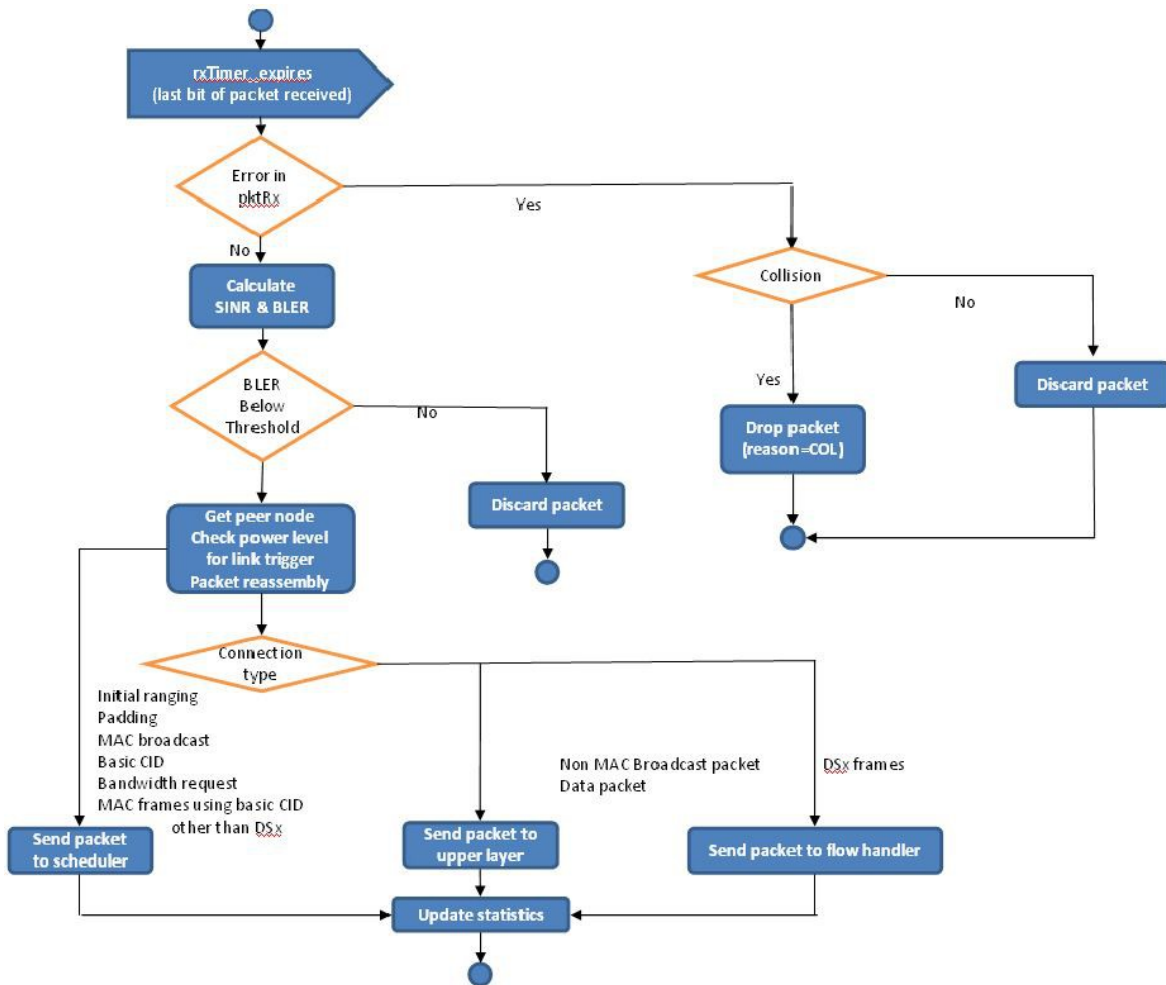
Εικόνα 57: Διαχείριση εξερχομένου πακέτου (WiMAX Forum module)

Ένα πακέτο που παραλαμβάνεται από ένα ανώτερο επίπεδο, ταξινομείται από τους καταχωρημένους ταξινομητές, και μιας και μπορεί να υπάρχουν πολλαπλοί ταξινομητές, το επίπεδο MAC τους διαπερνά όλους με την σειρά μέχρι να βρεθεί ένα έγκυρο CID ή μέχρι να έχουν δοκιμαστεί όλοι. Εάν το CID είναι έγκυρο, το πακέτο μπαίνει στην ουρά, αλλιώς ακυρώνεται.

Όταν ένα πακέτο λαμβάνεται, συγκεκριμένα το πρώτο του bit, ακολουθείται η διαδικασία της εικόνας 58. Στο τέλος της διαδικασίας το πακέτο διαχειρίζεται όπως φαίνεται στην εικόνα 59.



Εικόνα 58: Διαχείριση νέου εισερχομένου πακέτου (WiMAX Forum module)



Εικόνα 59: Διαχείριση ληφθέντος πακέτου (WiMAX Forum module)

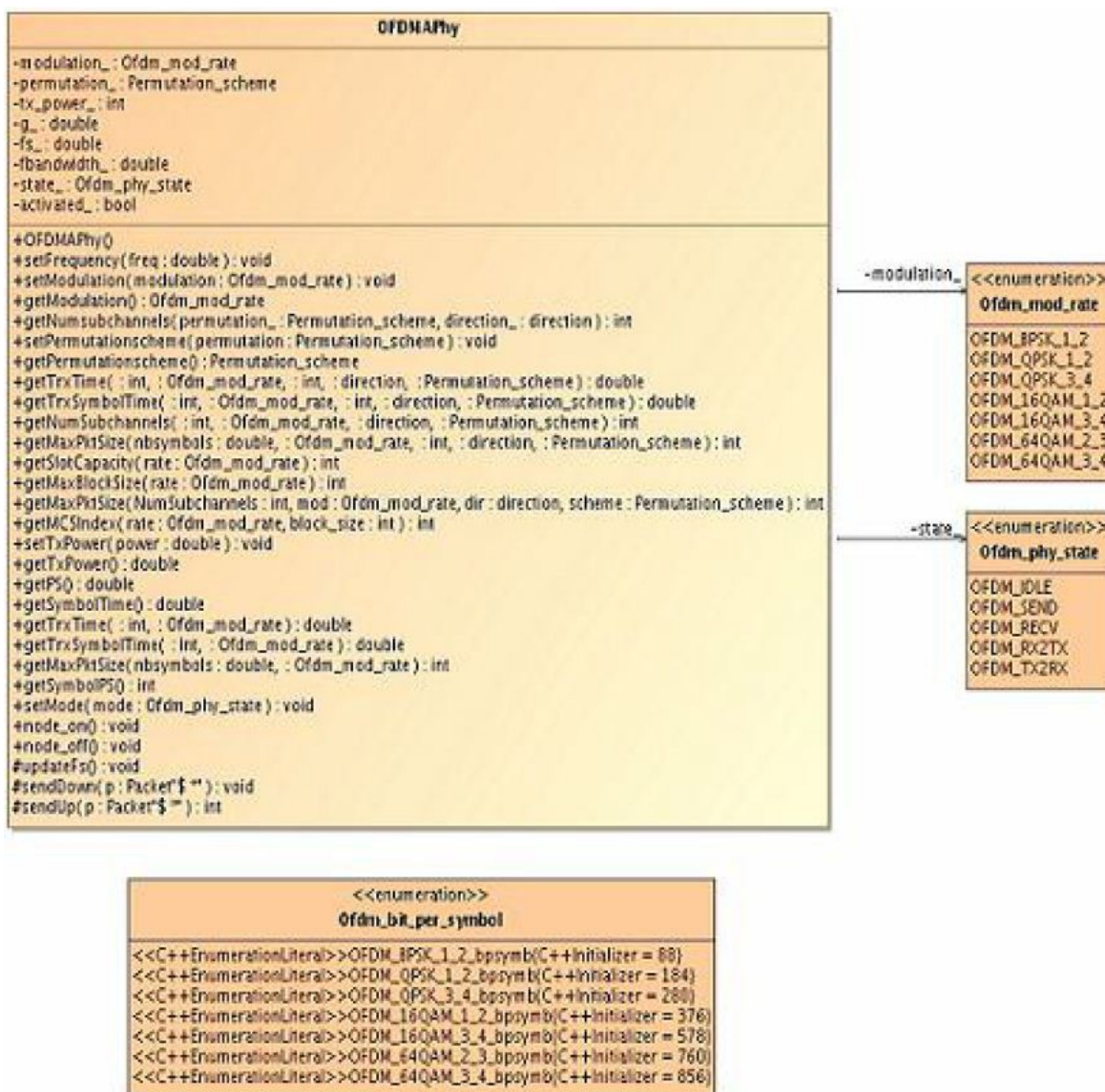
## 5.2.19 OFDMA PHY

### 5.2.19.1 Διάγραμμα κλάσεων του OFDMA του φυσικού επιπέδου

Το φυσικό επίπεδο OFDMA χρησιμοποιείται για την εκπομπή πακέτων στο υλοποιημένο μοντέλο, του οποίου η ρύθμιση για την συχνότητα του εύρους ζώνης και του κυκλικού προθέματος έχει γίνει μέσω συνδέσμων TCL. Αφού κληρονομεί από την κλάση WirelessPhy, παράμετροι όπως για παράδειγμα η συχνότητα ή η δύναμη εκπομπής μπορούν επίσης να ορισθούν μέσω TCL.

Όπως φαίνεται στην εικόνα 60, το φυσικό επίπεδο μπορεί να είναι σε διάφορες καταστάσεις. Όταν βρίσκεται σε κατάσταση αποστολής, όλα τα εισερχόμενα

πακέτα απορρίπτονται, ενώ στην κατάσταση λήψης, δεν μπορούν να σταλούν πακέτα. Επιπλέον το header των πακέτων περιέχει πληροφορίες όπως, η συχνότητα, η διαμόρφωση και το κυκλικό πρόθεμα, τα οποία μπορούν να χρησιμοποιηθούν σαν φίλτρα στα εισερχόμενα πακέτα.



Εικόνα 60: Διάγραμμα κλάσεων του OFDMA PHY (WiMAX Forum module)

Το μοντέλο υποστηρίζει διάφορες διαμορφώσεις και αντικατάσταση PUSC. Το επίπεδο MAC αναθέτει ριπές οι οποίες μπορούν να χρησιμοποιήσουν διαφορετικές διαμορφώσεις σύμφωνα με την απόσταση ή την παρεμβολή, κάτι που επιδρά στον ρυθμό δεδομένων και στον χρόνο εκπομπής. Το φυσικό επίπεδο

περιέχει βοηθητικές συναρτήσεις που καλούνται από το επίπεδο MAC όταν εκπέμπει δεδομένα:

- Η `getTxTime` επιστρέφει τον χρόνο που απαιτείται για να σταλεί ένα πακέτο σύμφωνα με το μέγεθός του και την διαμόρφωση, τον αριθμό των υπο-καναλιών και το σχήμα αντικατάστασης.
- Η `getMaxPktSize` είναι η αντίστροφη από την προηγούμενη συνάρτηση και επιστρέφει το μέγιστο μέγεθος πακέτου δεδομένου του αριθμού των συμβόλων OFDM που είναι διαθέσιμα, τον αριθμό των υπο-καναλιών, το σχήμα αντικατάστασης και την διαμόρφωση που χρησιμοποιείται.
- `getNumSubchannels`, `getSlotCapacity`, `getMCSindex`, `getMaxVlocksize`, κ.α.

Οι συναρτήσεις `node_on` και `node_off` ενεργοποιούν ή απενεργοποιούν το μπλοκάρισμα όλων των μεταδόσεων και λήψεων πακέτων, αλλά προς το παρόν δεν συνδέονται με κάποιους μηχανισμούς κατανάλωσης ενέργειας.

#### **5.2.19.2 OFDMA PHY**

Το μοντέλο καναλιού που χρησιμοποιείται στο OFDMA είναι ένας συνδυασμός ενός μαζικού `Cost231 path loss` και μιας υλοποίησης του μοντέλου `Rayleigh Fading`. Το φαινόμενο του `Doppler` περιλαμβάνεται για τον εγκλωβισμό φαινομένων κινητικότητας. Επιπλέον περιλαμβάνεται και το `fast fading` συνθέτοντας το κανάλι σαν ένα `Rayleigh fading` κανάλι με πολλαπλά χτυπήματα (`taps`).

Από την φύση του OFDMA frame, τα σύμβολα δημιουργούνται στο domain συχνοτήτων, κάτι που επιτρέπει την παράβλεψη ενός μεγάλου βήματος της πραγματική διαδικασία μετάδοσης σε ένα σύστημα WiMAX. Στην πράξη, μόλις τα σύμβολα δημιουργηθούν στο domain συχνοτήτων, πρέπει να μετατραπούν σε ένα `time domain signal` μέσω του IFFT. Αυτό το `time domain signal` στην συνέχεια συνελίσσεται αποτελεσματικά με το πραγματικό φυσικό κανάλι, αποδίδοντας το παραληφθέν σήμα,  $y(t)$ . Ωστόσο το σήμα που πρέπει να δει ο παραλήπτης είναι το  $Y(f)$  οπότε πρέπει να ληφθεί ένα FFT για να μετατραπεί πάλι σε αναπαράσταση συχνότητας domain.

Το συστατικό του bulk path loss του καναλιού υπολογίζεται κατά την προσομοίωση επειδή είναι απαραίτητα σαν παράμετροι, η απόσταση μεταξύ των σταθμών και η τρέχουσα ισχύς μετάδοσης. Το fast fading συστατικό του καναλιού όμως μπορεί να υπολογιστεί πριν αρχίσει η προσομοίωση. Σε αυτό το patch περιλαμβάνονται 1000 προ-υπολογισμένα κανάλια για κάθε μοντέλο ITU.

#### 5.2.19.2.1 Cost231

Το cost231 είναι το μοντέλο bulk path loss που χρησιμοποιείται, λεπτομέρειες για οποίο έχουν παρθεί από το βιβλίο “Channel Models: A Tutorial, Feb 21, 2007, Raj Jain”.

Το propagation loss υπολογίζεται από τον παρακάτω τύπο:

$$P_{L,urban}(d)dB = 46.3 + 33.9\log_{10}(f_c) - 13.82\log_{10}(h_t) - a(h_r) + (44.9 - 6.55\log_{10}(h_t))\log_{10}(d) + C_M$$

Εικόνα 61: Propagation loss formula (WiMAX Forum module)

Όπου:

P\_L = Το propagation loss

f\_c = Η συχνότητα του παρόχου (2GHz σε αυτή την περίπτωση)

h\_t = Το ύψος της κεραίας εκπομπής (30-300m)

h\_r = Το ύψος της κεραίας λήψης (1-10m)

d = Η απόσταση μεταξύ του πομπού και του δέκτη (1-20km)

C\_M = Είναι 0dB για τα προάστια και 3dB για τις μητροπολιτικές περιοχές

#### 5.2.19.3 Fast/Frequency Selective Fading-Υπολογισμοί πριν την προσομοίωση

Το κομμάτι αυτό του μοντέλου είναι για την συμπερίληψη του φαινομένου του Doppler. Ένα διάνυσμα από αριθμούς Rayleigh δημιουργείται και πολλαπλασιάζεται με ένα φάσμα Doppler ίδιου μήκους. Το IFFT από αυτόν τον πίνακα δίνει μια σχετική συχνότητα time domain του κατάλληλου μεγέθους. Αυτή η διαδικασία επαναλαμβάνεται η φορές, όπου η είναι ο αριθμός των taps στο μοντέλο ITU-PDP. Τα η στοιχεία του πίνακα μετά τοποθετούνται στα κατάλληλα σημεία δείγματος και κλιμακώνονται από τις δυνάμεις των taps που δίνονται από

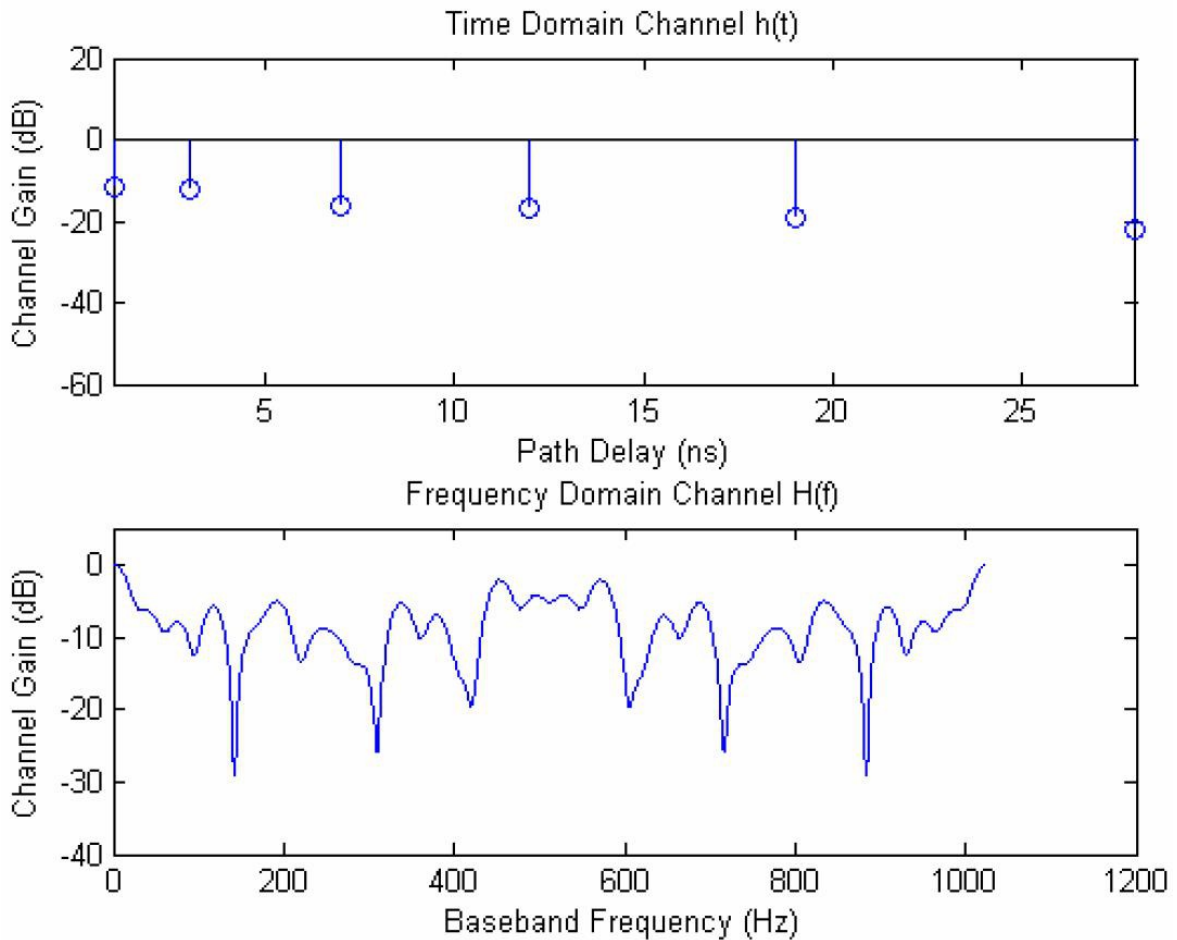


το μοντέλο ITU. Στην συνέχεια ένα FFT μεγέθους 1024pt βγαίνει από αυτά τα νούμερα το οποίο δίνει τους 1024 συντελεστές του καναλιού. Ένα παράδειγμα του μοντέλου ITU δίνεται στον πίνακα 10.

Πίνακας 10: Παράδειγμα του μοντέλου ITU (WiMAX Forum module)

Tap	Channel A	
	Relative delay (ns)	Average power (dB)
1	0	0,0
2	310	-1,0
3	710	-9,0
4	1090	-10,0
5	1730	-15,0
6	2510	-20,0

Το FFT του διανύσματος time domain channel που βγαίνει σαν αποτέλεσμα αποθηκεύεται σαν μια γραμμή σε ένα αρχείο (του οποίου το όνομα ταιριάζει στο μοντέλο ITU που χρησιμοποιείται). Το ίδιο κανάλι φαίνεται και στο time και στο frequency domain στην εικόνα 62.



Εικόνα 62: Μετατροπή από time domain σε frequency domain κανάλι (WiMAX Forum module)

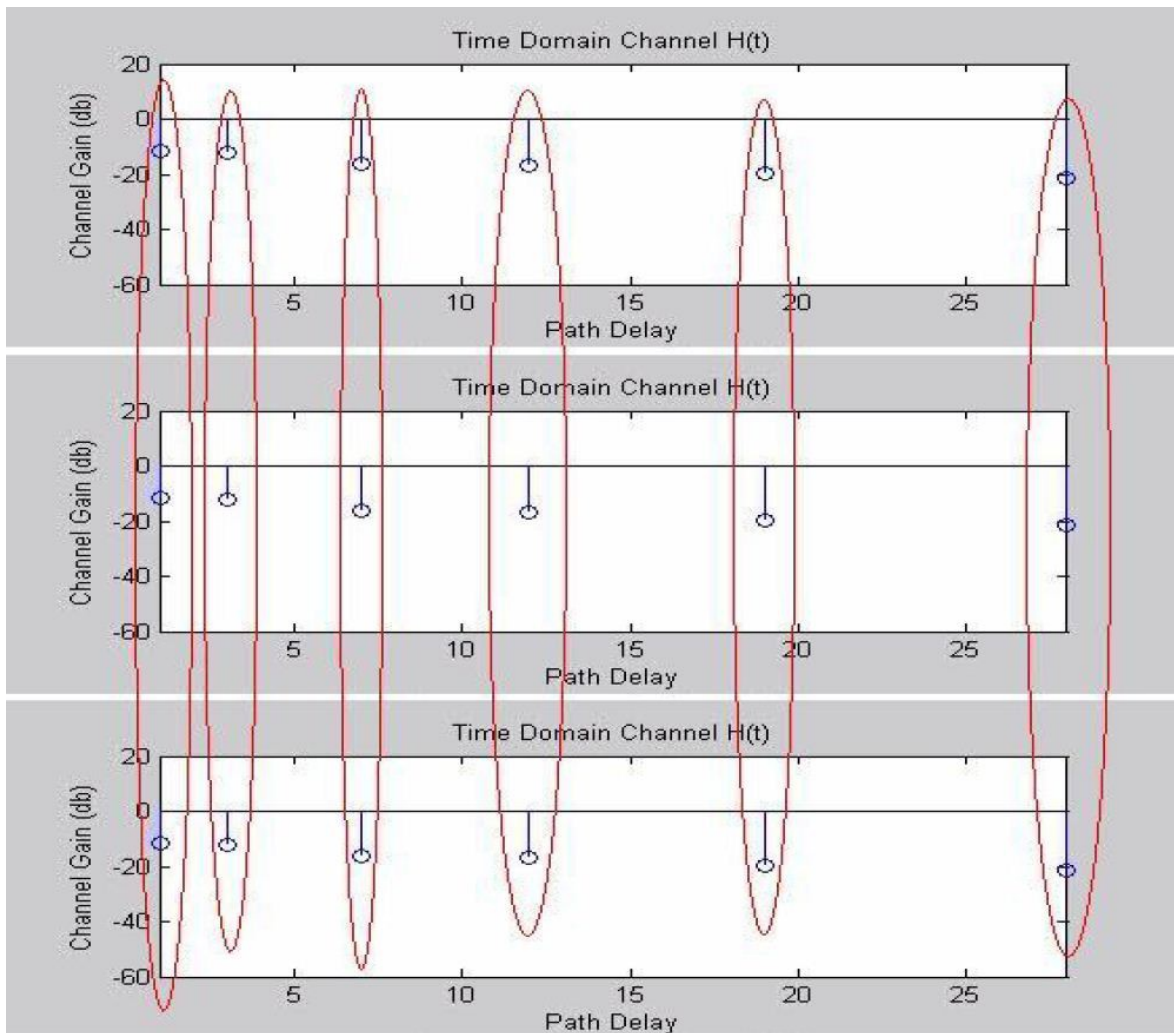
Τα αποτελέσματα αυτής της υλοποίησης είναι ότι οι υπολογισμοί των μοντέλων αυτών των καναλιών γίνονται τελείως εκτός προσομοίωσης, κάτι που γλυτώνει πολύ όγκο εργασίας κατά την διάρκεια της προσομοίωσης. Ένα παράδειγμα του αρχείου εκποίησης των συχνοτήτων φαίνεται στην εικόνα 63.

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

	Subcarrier 1	Subcarrier 2	Subcarrier 3	Subcarrier 4	Subcarrier 5	Subcarrier 6			
Channel 1	0.999952	0.999574	0.998603	0.997050	0.994932	0.992275	*	*	*
Channel 2	0.999749	0.998962	0.996599	0.992683	0.987248	0.980347	*	*	*
Channel 3	0.999638	0.998834	0.996341	0.992180	0.986389	0.979020	*	*	*
Channel 4	0.997511	0.996608	0.993877	0.989342	0.983044	0.975040	*	*	*
Channel 5	0.998988	0.998225	0.995546	0.990970	0.984536	0.976298	*	*	*
Channel 6	0.998702	0.998099	0.995523	0.990995	0.984548	0.976235	*	*	*
Channel 7	0.820728	0.820330	0.818093	0.814030	0.808172	0.800560	*	*	*
Channel 8	0.935429	0.934921	0.932630	0.928573	0.922780	0.915297	*	*	*
	*	*	*	*	*	*	*		
	*	*	*	*	*	*	*	*	
	*	*	*	*	*	*	*	*	*

Εικόνα 63: Παράδειγμα αρχείου εκποίησης καναλιού του μοντέλου ITU (WiMAX Forum module)

Το αποτέλεσμα της παραπάνω διαδικασίας είναι η μοντελοποίηση της χρονικής σχετικότητας ανά tap σε σχέση με ένα κανάλι πολλαπλών taps. Το τελευταίο βήμα της διαδικασίας είναι η μετατροπή των time domain channel gains σε ένα frequency domain view του καναλιού, κάτι που είναι χρήσιμο γιατί στον NS2 μας ενδιαφέρει η δύναμη λήψης σε κάθε υπο-πάροχο. Ωστόσο η σχετικότητα μπορεί να φανεί καλύτερα σε μια απεικόνιση του time domain. Η εικόνα 64 παρουσιάζει τρεις συνεχείς απεικονίσεις του καναλιού. Η σχετικότητα είναι ανάμεσα στα taps που είναι κυκλωμένα με κόκκινο.

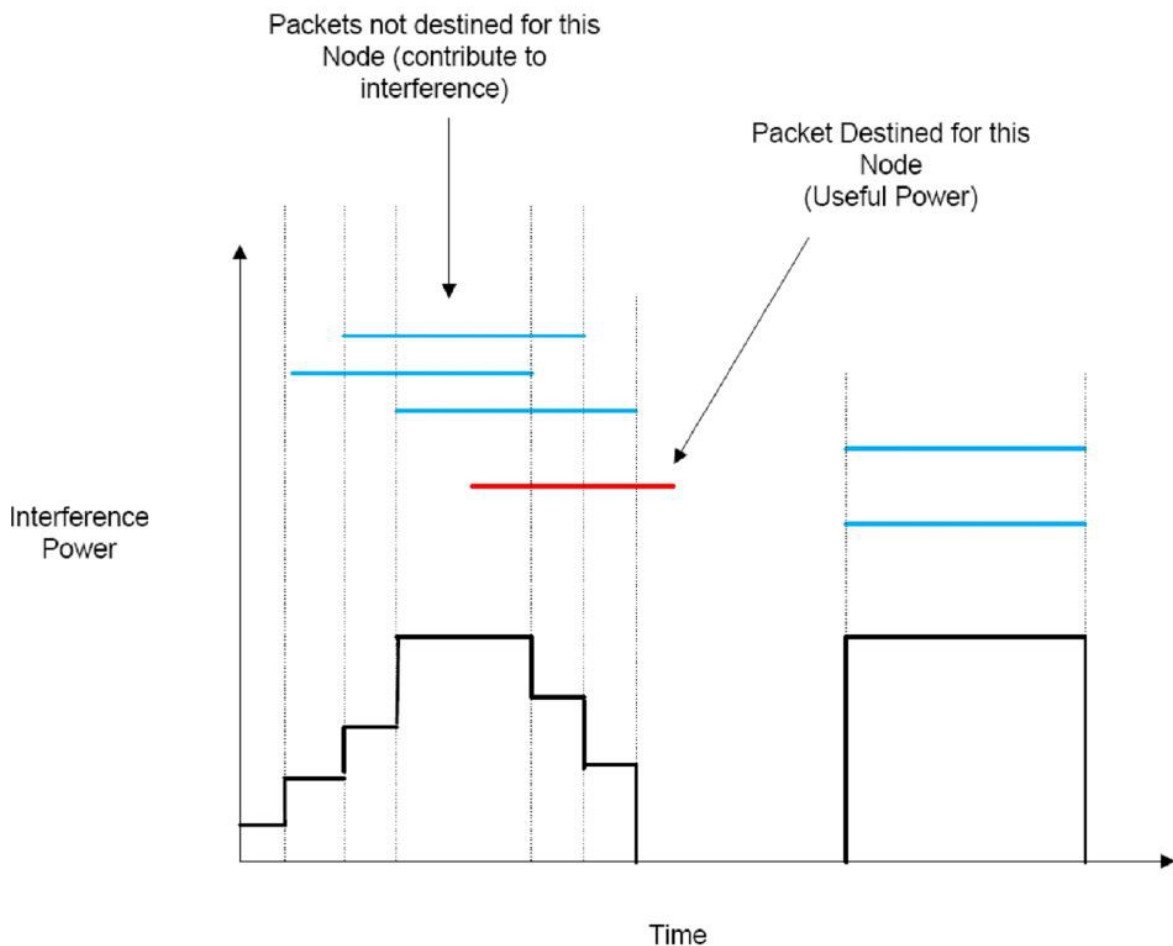


Εικόνα 64: Time domain correlation (WiMAX Forum module)

#### 5.2.19.4 Μοντελοποίηση παρεμβολών (SIR)

Σε κάθε δέκτη, όταν λαμβάνεται ένα πακέτο, η ισχύς λήψης των υπο-παρόχων που παραλαμβάνεται υπολογίζεται. Στην συνέχεια το πακέτο καταγράφεται και ένα μετρητής ξεκινά και η δύναμή του προστίθεται σε έναν πίνακα I1 σαν παρέμβαση σε κάθε υπο-πάροχο. Όλη αυτή η διαδικασία λαμβάνει χώρα στο φυσικό επίπεδο. Όταν παραληφθεί το πρώτο bit ενός MAC πακέτου, αποφασίζεται αν πρόκειται για ένα S-πακέτο (πακέτο που αναφέρεται σε αυτόν τον παραλήπτη το οποίο συνεισφέρει στην ισχύ του σήματος) ή για I-πακέτο (πακέτο που δεν αναφέρεται σε αυτόν τον παραλήπτη και συνεισφέρει στην ισχύ παρεμβολής). Εάν πρόκειται για S-πακέτο αφαιρούμε από τον πίνακα I1 την ισχύ του πακέτου (εφόσον την είχαμε

ήδη προσθέσει πιο πριν) και ένας μετρητής πακέτου ξεκινά για να παραλάβουμε το πακέτο ολοκληρωτικά. Ο πίνακας I1 είναι η παρεμβολή που προκλήθηκε από τις υποκείμενες μεταδόσεις των πακέτων όταν παραλάβαμε το πρώτο bit. Τώρα για μετρήσουμε την παρεμβολή που θα προκληθεί όσο λαμβάνουμε το πακέτο, ορίζουμε ένα I2 πίνακα και προσθέτουμε την ισχύ των πακέτων που παραλαμβάνουμε. Όταν παραλάβουμε το S-πακέτο μας ολοκληρωτικά, ελέγχουμε αυτόν τον πίνακα, τον προσθέτουμε στο επίπεδο υπο-παρόχου του I1 πίνακα και εκτελούμε ένα EESM για να πάρουμε την τιμή του SIR. Στην συνέχεια παρατίθεται ο πίνακας SIR για να πάρουμε την πιθανότητα σφάλματος. Εάν υπάρχει σφάλμα στο πακέτο, απορρίπτεται, αλλιώς επεξεργάζεται.



Εικόνα 65: Απεικόνιση της παρεμβολής (WiMAX Forum module)

### 5.3 Διαδικασία εγκατάστασης

Η εγκατάσταση του module έγινε σε ένα εικονικό μηχάνημα του λογισμικού virtualbox της sun, το οποίο είχε σαν λειτουργικό το Fedora 13.

Τα βήματα της εγκατάστασης είναι τα παρακάτω:

- 9) Κατεβάζουμε το ns-allinone-2.31 από την παρακάτω διεύθυνση :  
<http://sourceforge.net/projects/nsnam/files/allinone/>
- 10) Αποσυμπιέζουμε το αρχείο που κατεβάσαμε στην επιθυμητή τοποθεσία.
- 11) Κατεβάζουμε το patch από την σελίδα του project :  
<http://code.google.com/p/ns2-wimax-awg/downloads/list>
- 12) Αποσυμπιέζουμε το patch και το αντιγράφουμε στον φάκελο “ns-allinone-2.31/ns-2.31”.
- 13) Από τον φάκελο “ns-allinone-2.31/ns-2.31” τρέχουμε την ακόλουθη εντολή :  
“ patch -p1 < ns-2.31-WiMAX\_AATG\_R2.6.patch” (χωρίς τα εισαγωγικά).
- 14) Στο αρχείο install στον φάκελο του ns-allinone-2.31 προσθέτουμε την εντολή “make clean” (χωρίς τα εισαγωγικά) στην γραμμή πριν την εντολή “make”.
- 15) Από τον φάκελο του ns-allinone-2.31 τρέχουμε την εντολή “./install”
- 16) Μετά την ολοκλήρωση της εγκατάστασης πρέπει να προσθέσουμε τα παρακάτω στο αρχείο ~/.bashrc :

## Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

```
#LD_LIBRARY_PATH

OTCL_LIB=/home/gkaran/Desktop/ns/ns-allinone-2.31/otcl-1.13

NS2_LIB=/home/gkaran/Desktop/ns/ns-allinone-2.31/lib

X11_LIB=/usr/X11R6/lib

USR_LOCAL_LIB=/usr/local/lib

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$X11_LIB:
$USR_LOCAL_LIB

#TCL_LIBRARY

TCL_LIB=/home/gkaran/Desktop/ns/ns-allinone-2.31/tcl8.4.14/library

USR_LIB=/usr/lib

export TCL_LIBRARY=$TCL_LIB:$USR_LIB

#PATH

XGRAPH=/home/gkaran/Desktop/ns/ns-allinone-
2.31/bin:/home/gkaran/Desktop/ns/ns-allinone-
2.31/tcl8.4.14/unix:/home/gkaran/Desktop/ns/ns-allinone-
2.31/tk8.4.14/unix

NS=/home/gkaran/Desktop/ns/ns-allinone-2.31/ns-2.31/

NAM=/home/gkaran/Desktop/ns/ns-allinone-2.31/nam-1.13

PATH=$PATH:XGRAPH:$NS:$NAM
```

Όπου “/home/gkaran/Desktop/ns/ns-allinone-2.31/” βάζετε τον φάκελο που έχετε το ns-allinone-2.31

Προαιρετικά πριν το βήμα 6 αν θέλουμε να λειτουργήσει σωστά το nam ακολουθούμε τα παρακάτω βήματα:

- 1) Μέσα στον φάκελο του tk8.4.14 που βρίσκεται μέσα στον φάκελο του ns-allinone-2.31 δημιουργούμε ένα αρχείο με όνομα “tk-lastevent.patch” ή όπως αλλιώς θέλουμε και γράφουμε σε αυτό τις παρακάτω εντολές.

```
--- generic/tk.h.orig 2008-02-06 16:31:40.000000000 +0100
+++ generic/tk.h 2008-07-24 08:21:46.000000000 +0200
@@ -635,17 +635,15 @@
 *
 *-----
-----
 */
-#define VirtualEvent (LASTEvent)
-#define ActivateNotify (LASTEvent + 1)
-#define DeactivateNotify (LASTEvent + 2)
-#define MouseWheelEvent (LASTEvent + 3)
-#define TK_LASTEVENT (LASTEvent + 4)
+#define VirtualEvent (MappingNotify + 1)
+#define ActivateNotify (MappingNotify + 2)
+#define DeactivateNotify (MappingNotify + 3)
+#define MouseWheelEvent (MappingNotify + 4)
+#define TK_LASTEVENT (MappingNotify + 5)
#define MouseWheelMask (1L << 28)
-
#define ActivateMask (1L << 29)
#define VirtualEventMask (1L << 30)
-#define TK_LASTEVENT (LASTEvent + 4)
/*
```

- 2) Στον ίδιο φάκελο που είμαστε τρέχουμε την παρακάτω εντολή :  
“patch -p0 < tk-lastevent.patch” (χωρίς τα εισαγωγικά).
- 3) Αν η εντολή μας επιστρέψει σφάλμα μπορούμε να το κάνουμε χειροκίνητα.  
Ανοίγουμε το αρχείο tk.h στον φάκελο generic και περίπου στην 635 γραμμή πάμε και σβήνουμε όσες σειρές είναι στον πιο πάνω πλαίσιο με ένα “-“ μπροστά από το “#”. Στην συνέχεια προσθέτουμε τις σειρές που έχουν το “+” μπροστά από το “#”. Φροντίζουμε μόνο η σειρά των γραμμών να είναι όπως στο πλαίσιο πιο πάνω.



## **5.4 Προσομοίωση**

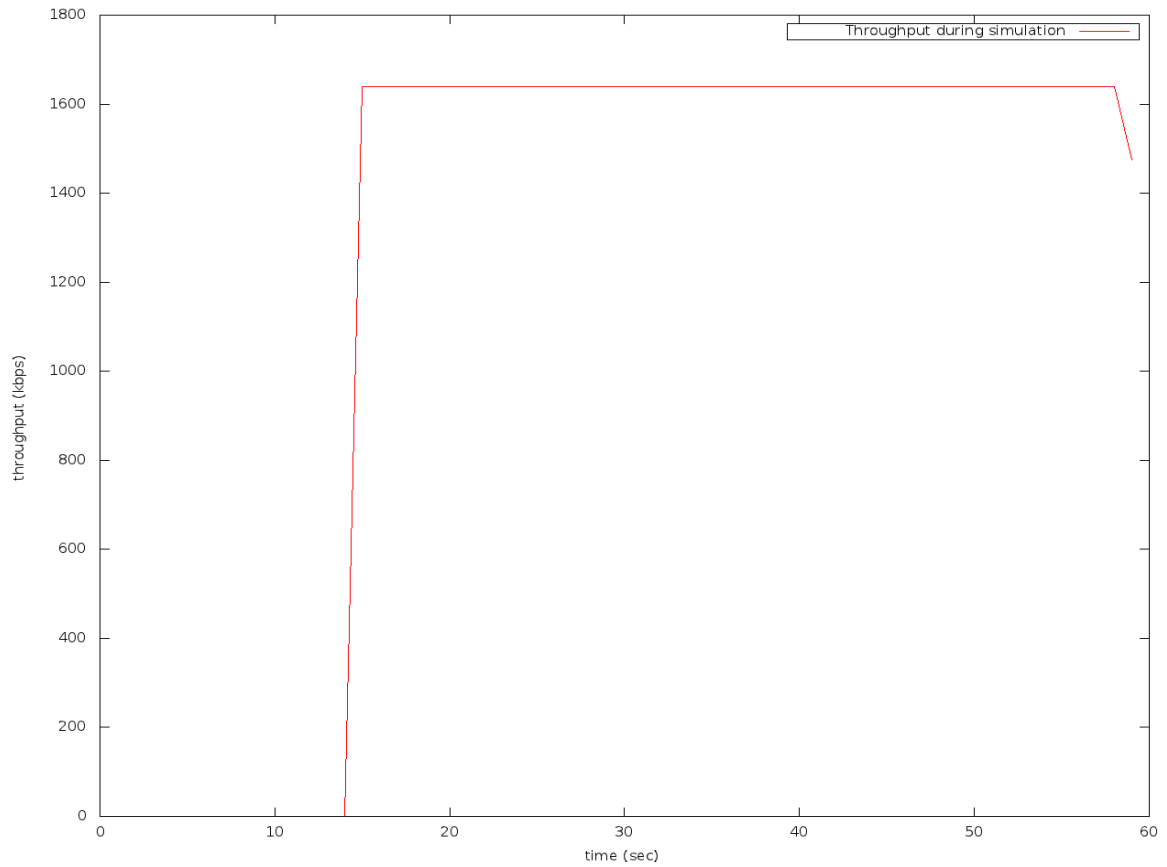
### **5.4.1 Σενάριο προσομοίωσης**

Το σενάριο το οποίο γράφτηκε για τις ανάγκες της εργασίας αυτής ακολουθεί την παρακάτω δομή. Έχουμε ένα σταθμό βάσης και έναν σταθμό συνδρομητή όπου και οι δύο είναι στάσιμοι. Ορίζουμε μεταξύ τους μια βασική επικοινωνία τύπου CBR με μέγεθος πακέτου 1500 bytes και διάστημα μεταξύ αποστολής δύο πακέτων 0.005 δευτερόλεπτα. Δεν ορίζεται κάποιος μηχανισμός QoS και ο χρόνος προσομοίωσης ορίζεται στα 60 δευτερόλεπτα, όπου στα 15 πρώτα δευτερόλεπτα δεν υπάρχει κάποια CBR επικοινωνία μεταξύ των δύο σταθμών. Στο 15ο δευτερόλεπτο η επικοινωνία ξεκινάει και σταματάει 1 δέκατο του δευτερολέπτου πριν το τέλος του χρόνου προσομοίωσης. Σαν έξοδο όταν τρέξουμε το σενάριο έχουμε δύο αρχεία. Ένα WIMAX\_FORUM.tr και ένα WIMAX\_FORUM.nam. Το πρώτο χρησιμοποιείται για την εξόρυξη των στατιστικών της προσομοίωσης και το δεύτερο για την γραφική απεικόνιση των γεγονότων κατά την διάρκεια της προσομοίωσης. Ο κώδικας TCL που εκφράζει το σενάριο της προσομοίωσης είναι διαθέσιμος στο παράρτημα Β.

### **5.4.2 Αποτελέσματα προσομοίωσης**

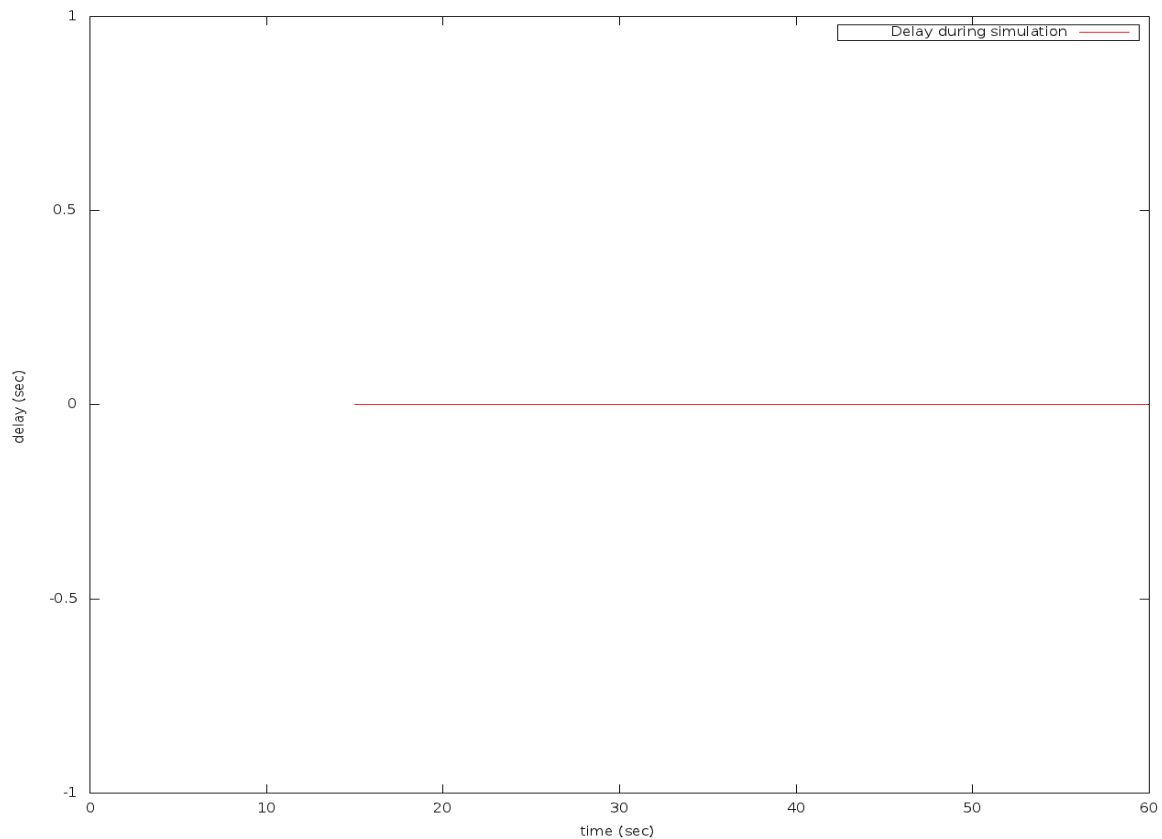
Στην τρέχουσα ενότητα θα δούμε και θα σχολιάσουμε διάφορα γραφήματα που αφορούν την προσομοίωση.

Στην εικόνα 66 βλέπουμε το throughput κατά την διάρκεια της προσομοίωσης. Παρατηρούμε ότι είναι σταθερό από το 15<sup>ο</sup> δευτερόλεπτο (όπου στην ουσία αρχίζει η επικοινωνία) και καθ' όλη την διάρκεια της προσομοίωσης και μειώνεται κατακόρυφα στα τελευταία δευτερόλεπτα όπου και κλείνει η επικοινωνία.



Εικόνα 66: Throughput κατά την διάρκεια της προσομοίωσης (WiMAX Forum)

Στην εικόνα 67 βλέπουμε το delay κατά την διάρκεια της προσομοίωσης για το WiMAX Forum module. Παρατηρούμε ότι από την στιγμή που αρχίζει η κινητικότητα στο δίκτυο το delay παραμένει στο μηδέν μέχρι να τελειώσει η κινητικότητα στο δίκτυο.

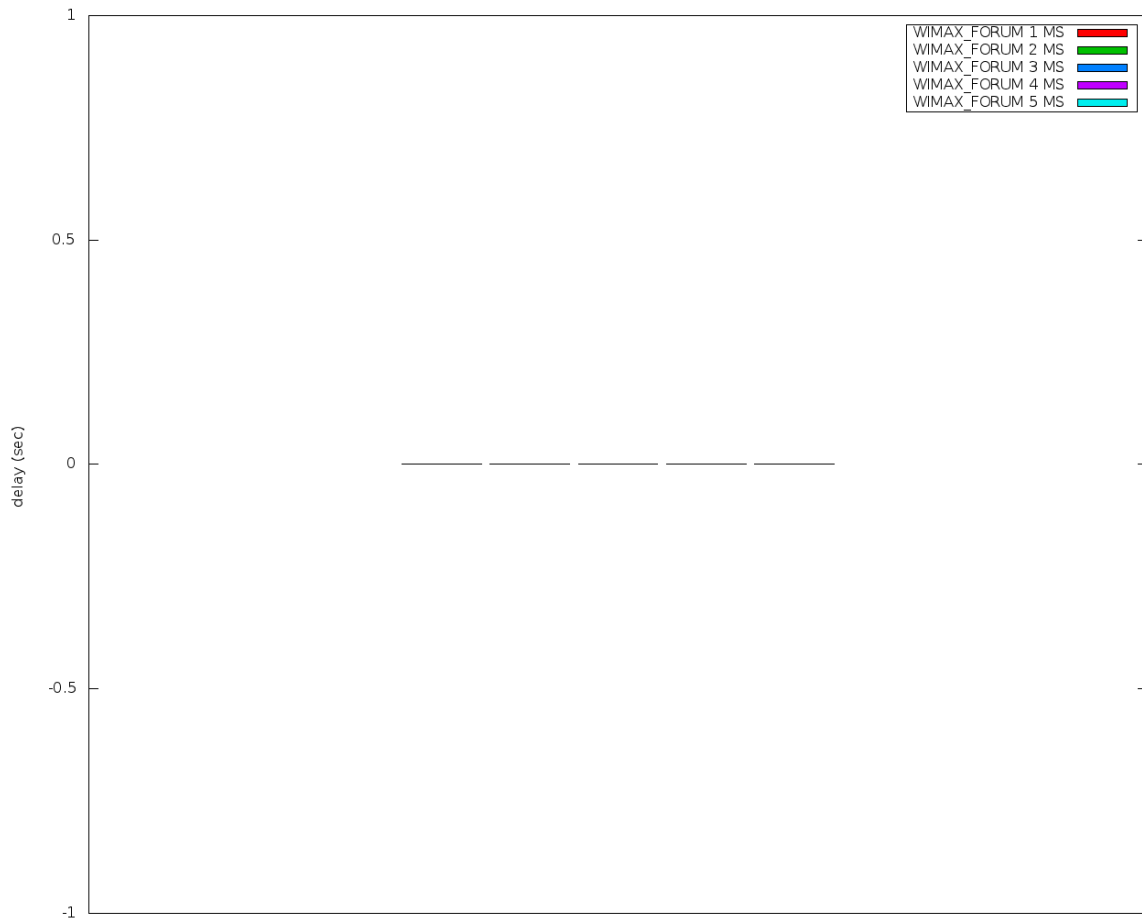


Εικόνα 67: Delay κατά την διάρκεια της προσομοίωσης (WiMAX Forum module)

### 5.4.3 Περαιτέρω διερεύνηση

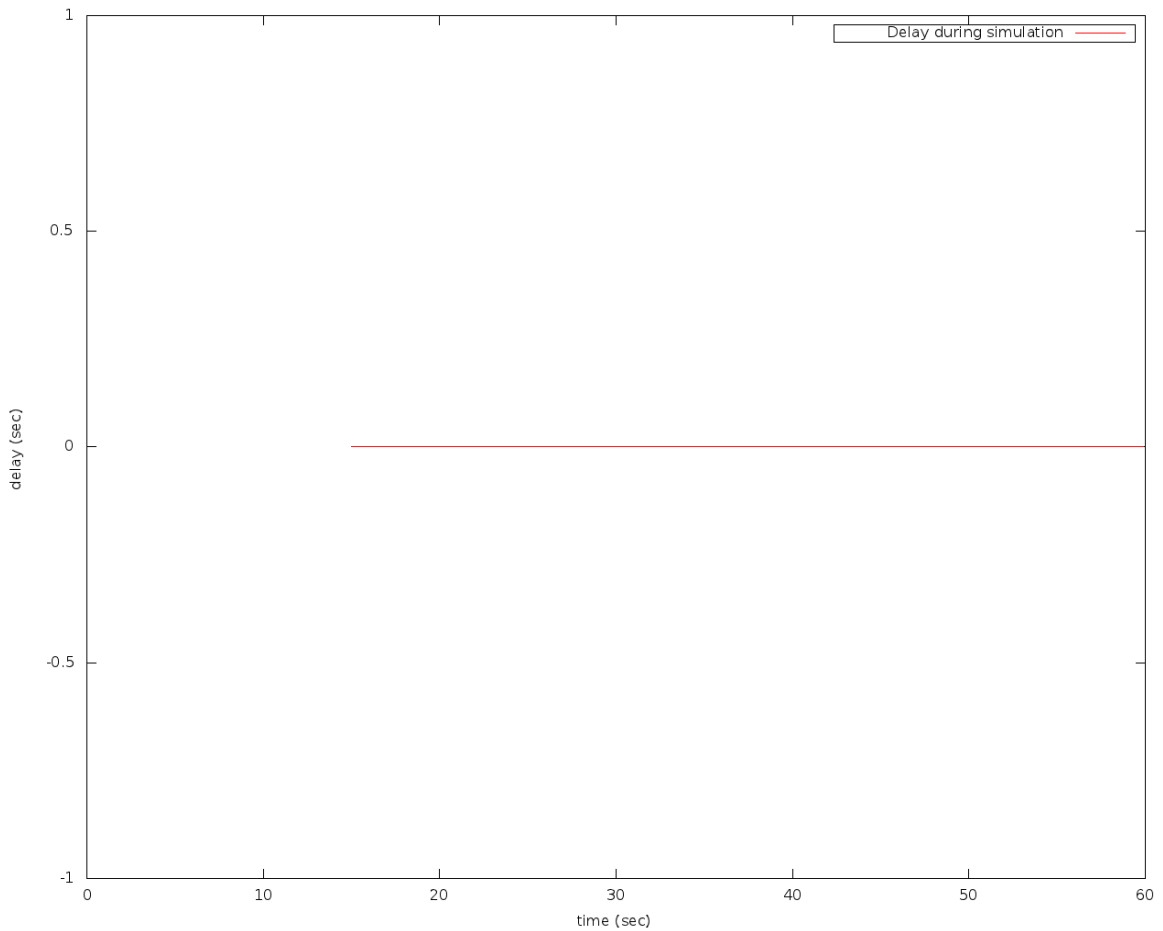
Για να έχουμε μια καλύτερη αντίληψη για την συμπεριφορά του module το σενάριο επαναλήφθηκε με ίδιες ρυθμίσεις αλλά με διαφορετικό αριθμό σταθμών κάθε φορά, ορίζοντας ένα χρονικό κενό μεταξύ της έναρξης εκπομπής δύο σταθμών της τάξης του μισού δευτερολέπτου. Συγκεκριμένα έγιναν δοκιμές για 1,2,3,4 και 5 σταθμούς συνδρομητές. Τα αποτελέσματα των προσομοιώσεων φαίνονται στις παρακάτω εικόνες.

Στην εικόνα 68 παρατηρούμε ότι παρόλο που αυξάνουμε τους σταθμούς συνδρομητές, δεν υπάρχει κάποια αλλαγή στην καθυστέρηση του δικτύου η οποία παραμένει μηδενική.



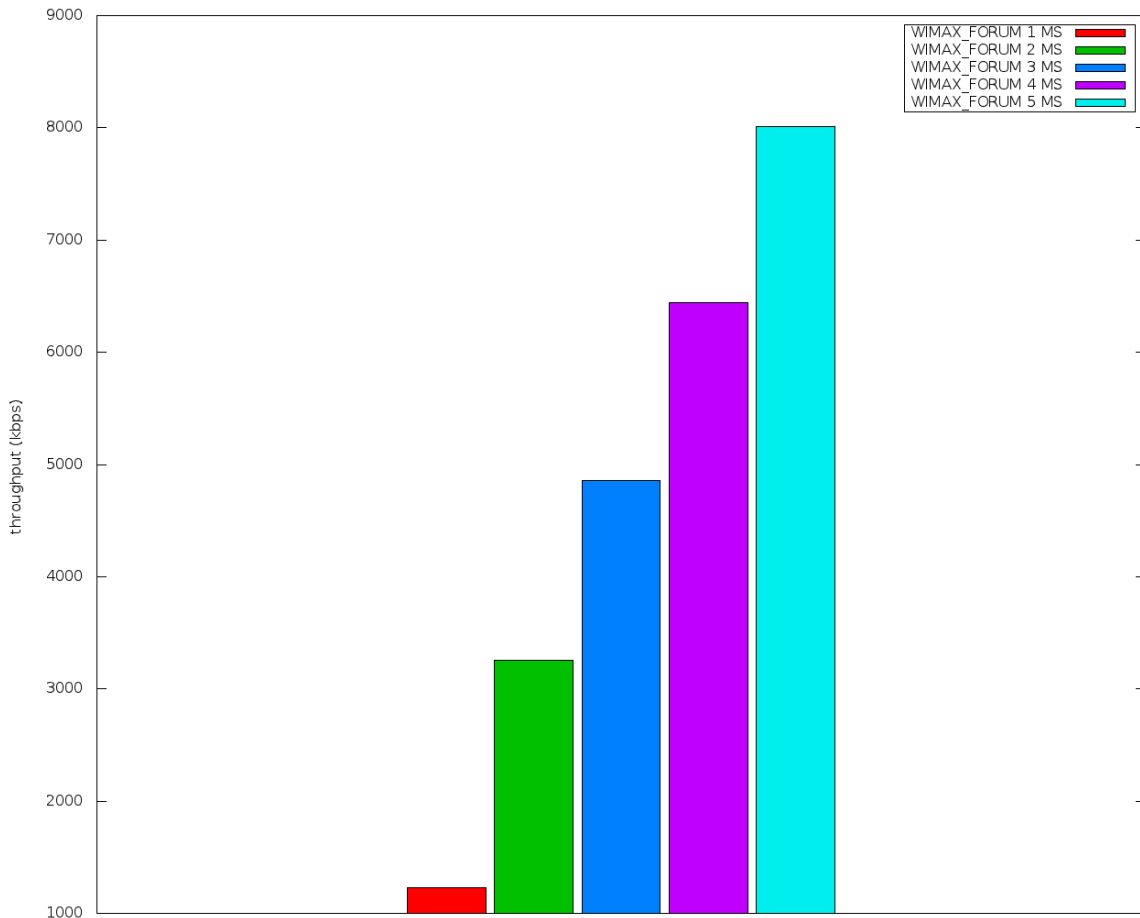
Εικόνα 68: Μέσο delay για τις διαφορετικές προσομοιώσεις (WiMAX Forum module)

Στην εικόνα 69 βλέπουμε ξανά την καθυστέρηση στο δίκτυο, αυτή τη φορά όμως πιο αναλυτικά κατά την διάρκεια της προσομοίωσης. Παρατηρούμε και εδώ ότι δεν υπάρχει καμία διακύμανση στην καθυστέρηση του δικτύου σε καμία χρονική στιγμή της προσομοίωσης.



Εικόνα 69: Αναλυτικό delay για τις διαφορετικές προσομοιώσεις (WiMAX Forum module)

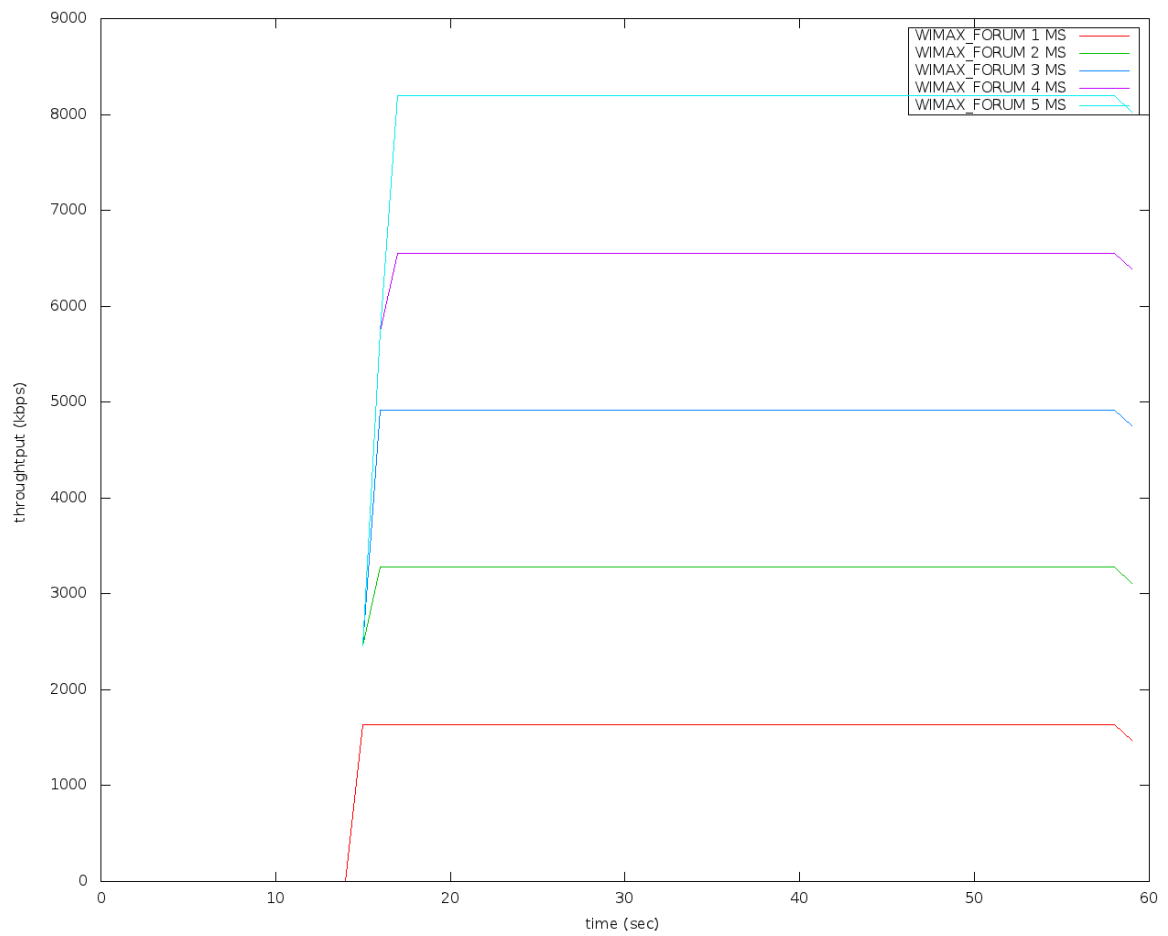
Στην εικόνα 70 όπου παρουσιάζεται το μέσο throughput για τις διαφορετικές προσομοιώσεις παρατηρούμε ότι με την χρήση δύο σταθμών έχουμε πολύ μεγάλη αύξηση στο throughput του δικτύου σε σχέση με τον έναν σταθμό. Από εκεί και πέρα στην χρήση περαιτέρω σταθμών υπάρχει πάλι μια αύξηση αλλά πιο ήπια.



Εικόνα 70: Μέσο throughput για τις διαφορετικές προσομοιώσεις (WiMAX Forum module)

Στην εικόνα 71 τέλος όπου εμφανίζεται το throughput καθ' όλη την διάρκεια της προσομοίωσης παρατηρούμε μια κατακόρυφη αύξησή του κατά της αρχή εκπομπής των σταθμών, και μια διατήρηση των επιπέδων μέχρι το τέλος της προσομοίωσης.

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου



Εικόνα 71: Αναλυτικό throughput για τις διαφορετικές προσομοιώσεις (WiMAX Forum module)

## **6 LRC module**

### **6.1 Εισαγωγή**

Το module αυτό βασίζεται στο πρότυπο 802.16 της IEEE για την έκδοση 2.28 του εξομοιωτή ns-2. Η υλοποίηση αυτή επικεντρώνεται στο επίπεδο MAC και στους μηχανισμούς του για κατανομή bandwidth καθώς και στην υποστήριξη QoS. Το module υλοποιεί τους πέντε μηχανισμούς ροής υπηρεσιών καθώς και τους μηχανισμούς τους για απαίτηση και χορήγηση. Επιπλέον επιτρέπει στους χρήστες να ρυθμίσουν τις απαιτήσεις QoS των εφαρμογών. Οι ροές των υπηρεσιών έχουν μοντελοποιηθεί από μηχανές πεπερασμένων καταστάσεων οι οποίες «αιχμαλωτίζουν» τις αντιδράσεις κάθε είδους υπηρεσίας σε διαφορετικά γεγονότα. Σε αυτό το module υποστηρίζεται το TDD και η τοπολογία PMP και χρησιμοποιείται το ασύρματο κανάλι που είναι διαθέσιμο στον εξομοιωτή ns-2. Ο σχεδιασμός του module βασίστηκε σε ένα άλλο module το οποίο σχεδιάστηκε για να εξομοιώσει το πρότυπο DOCSIS. Παρόλο που ήταν πιθανή η επαναχρησιμοποίηση κώδικα, έγιναν πολλές μετατροπές στον πυρήνα του module για το DOCSIS για να είναι συμβατός με το πρότυπο IEEE 802.16.

### **6.2 Documentation**

Κάποιο έγγραφο documentation για το συγκεκριμένο module δεν υπάρχει που να καθοδηγεί τον χρήστη στο πως να γράψει ένα σενάριο και ποιες μεταβλητές είναι διαθέσιμες. Το μόνο που δίνεται είναι ένα έγγραφο που αναφέρει όλες τις κλάσεις που γράφηκαν για την δημιουργία του module καθώς και τις δηλώσεις των μεταβλητών και συναρτήσεων τους καθώς και ένα σενάριο σαν παράδειγμα. Μέσω των δηλώσεων των κλάσεων δεν μπορεί κανείς να καταλάβει πολλά πράγματα. Η πιο εύκολη προσέγγιση είναι η παραμετροποίηση του υπάρχοντος σεναρίου ώστε να το φέρει ο καθένας στα μέτρα του.

Όπως αναφέρθηκε και πιο πριν το module επικεντρώνεται κυρίως γύρω από το MAC επίπεδο και υποστηρίζει τόσο μηχανισμούς κατανομής bandwidth όσο και



QoS. Επιπλέον υποστηρίζεται το TDD και μόνο η τοπολογία PMP. Τέλος έχουμε υποστήριξη για όλους τους μηχανισμούς συνδέσεων (ugs,rtps,ertps,nrpts,be). [8]

### 6.3 Διαδικασία εγκατάστασης

Για να εγκαταστήσετε το patch, ακολουθήστε τα παρακάτω βήματα:

1. Κατεβάστε το πακέτο ns-allinone-2.28.
2. Κατεβάστε την τελευταία έκδοση το module.
3. Αποσυμπιέστε τον πηγαίο κώδικα του module.
4. Αντιγράψτε τα αρχεία από τον φάκελο mac, στον φάκελο ns-allinone-2.28\ns-2.28\mac.
5. Αντιγράψτε τα αρχεία από τον φάκελο common, στον φάκελο ns-allinone-2.28\ns-2.28\common.
6. Αντιγράψτε τα αρχεία από τον φάκελο tcl, στον φάκελο ns-allinone-2.28\ns-2.28\tcl.
7. Κάντε τις παρακάτω διορθώσεις στο makefile.in
  - a. Μετά την πρόταση mac/mac-802\_3.o προσθέστε τα:  
mac/mac-802\_16-base.o mac/mac-802\_16-ss.o mac/mac 802\_16-  
bs.o \  
mac/mac-802\_16-FSM.o mac/mac-802\_16-timers.o \
8. Πηγαίνετε στον φάκελο ns-allinone-2.28/ns-2.28 και εκτελέστε της παρακάτω εντολές:
  - a. ./configure
  - b. make clean
  - c. make

### 6.4 Προσομοίωση

#### 6.4.1 Σενάριο προσομοίωσης

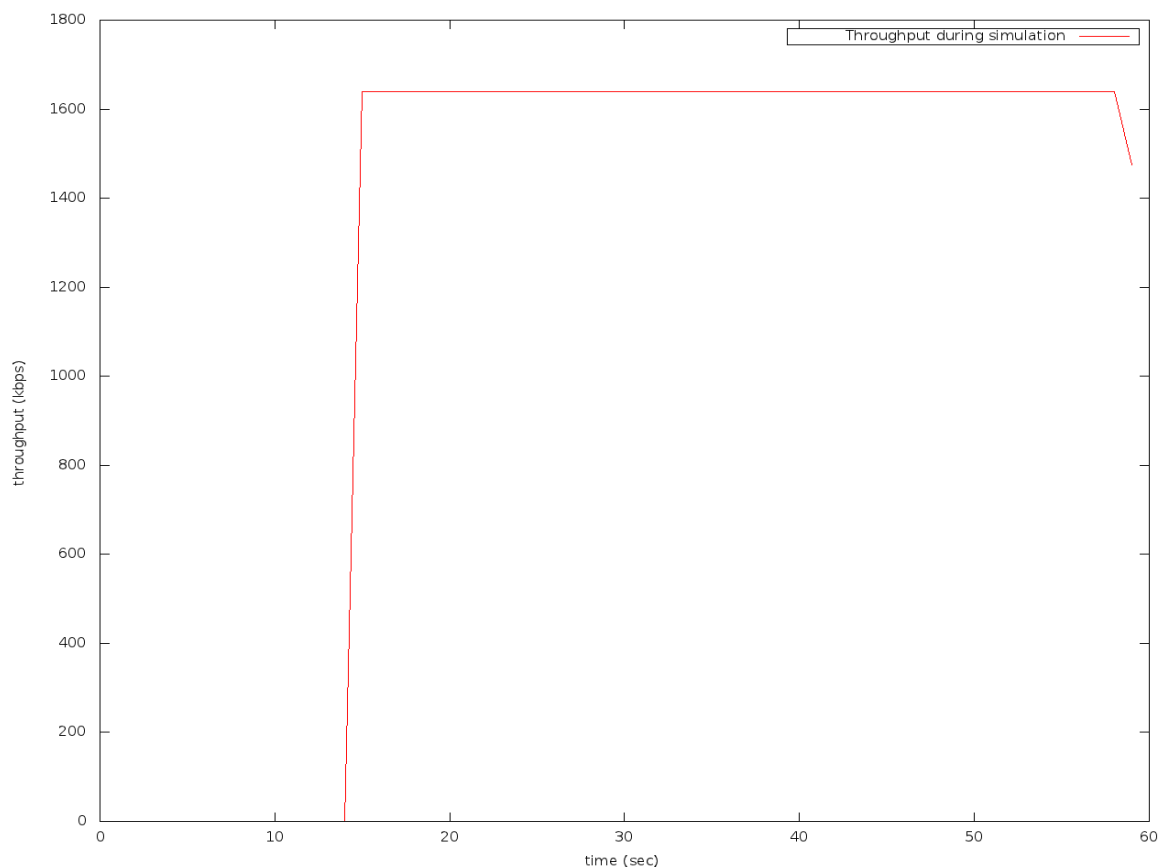
Το σενάριο το οποίο γράφτηκε για τις ανάγκες της εργασίας αυτής ακολουθεί την παρακάτω δομή. Έχουμε ένα σταθμό βάσης και έναν σταθμό συνδρομητή όπου και οι δύο είναι στάσιμοι. Ορίζουμε μεταξύ τους μια βασική επικοινωνία τύπου CBR με μέγεθος πακέτου 1500 bytes και διάστημα μεταξύ αποστολής δύο πακέτων

0.005 δευτερόλεπτα. Δεν ορίζεται κάποιος μηχανισμός QoS και ο χρόνος προσομοίωσης ορίζεται στα 60 δευτερόλεπτα, όπου στα 15 πρώτα δευτερόλεπτα δεν υπάρχει κάποια CBR επικοινωνία μεταξύ των δύο σταθμών. Στο 15ο δευτερόλεπτο η επικοινωνία ξεκινάει και σταματάει 1 δέκατο του δευτερολέπτου πριν το τέλος του χρόνου προσομοίωσης. Σαν έξοδο όταν τρέξουμε το σενάριο έχουμε δύο αρχεία. Ένα LRC.tr και ένα LRC.nam. Το πρώτο χρησιμοποιείται για την εξόρυξη των στατιστικών της προσομοίωσης και το δεύτερο για την γραφική απεικόνιση των γεγονότων κατά την διάρκεια της προσομοίωσης. Ο κώδικας TCL που εκφράζει το σενάριο της προσομοίωσης είναι διαθέσιμος στο παράρτημα Γ.

#### 6.4.2 Αποτελέσματα προσομοίωσης

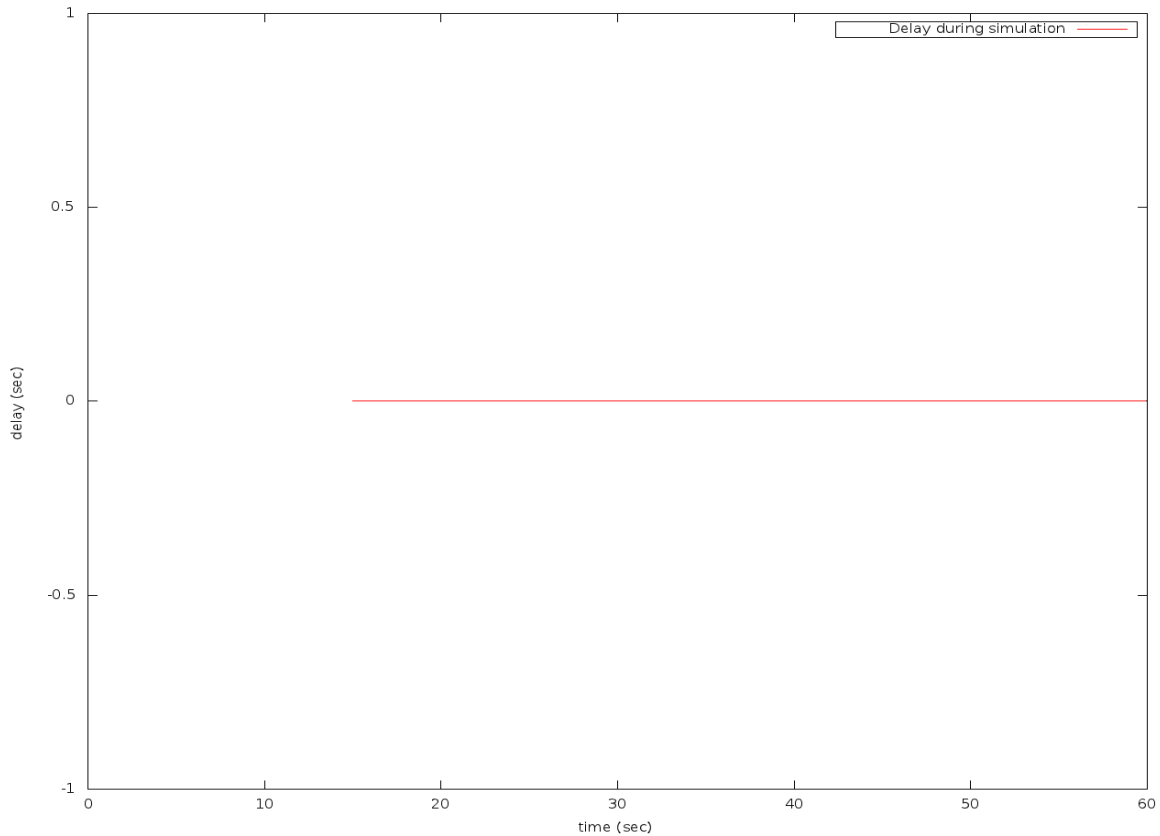
Στην τρέχουσα ενότητα θα δούμε και θα σχολιάσουμε διάφορα γραφήματα που αφορούν την προσομοίωση.

Στην εικόνα 72 βλέπουμε το throughput κατά την διάρκεια της προσομοίωσης. Παρατηρούμε ότι είναι σταθερό από το 15<sup>ο</sup> δευτερόλεπτο (όπου στην ουσία αρχίζει η επικοινωνία) και καθ' όλη την διάρκεια της προσομοίωσης και μειώνεται κατακόρυφα στα τελευταία δευτερόλεπτα όπου και κλείνει η επικοινωνία.



Εικόνα 72: Throughput κατά την διάρκεια της προσομοίωσης (LRC)

Στην εικόνα 73 βλέπουμε το delay κατά την διάρκεια της προσομοίωσης για το LRC module. Παρατηρούμε ότι από την στιγμή που αρχίζει η κινητικότητα στο δίκτυο το delay παραμένει στο μηδέν μέχρι να τελειώσει η κινητικότητα στο δίκτυο.

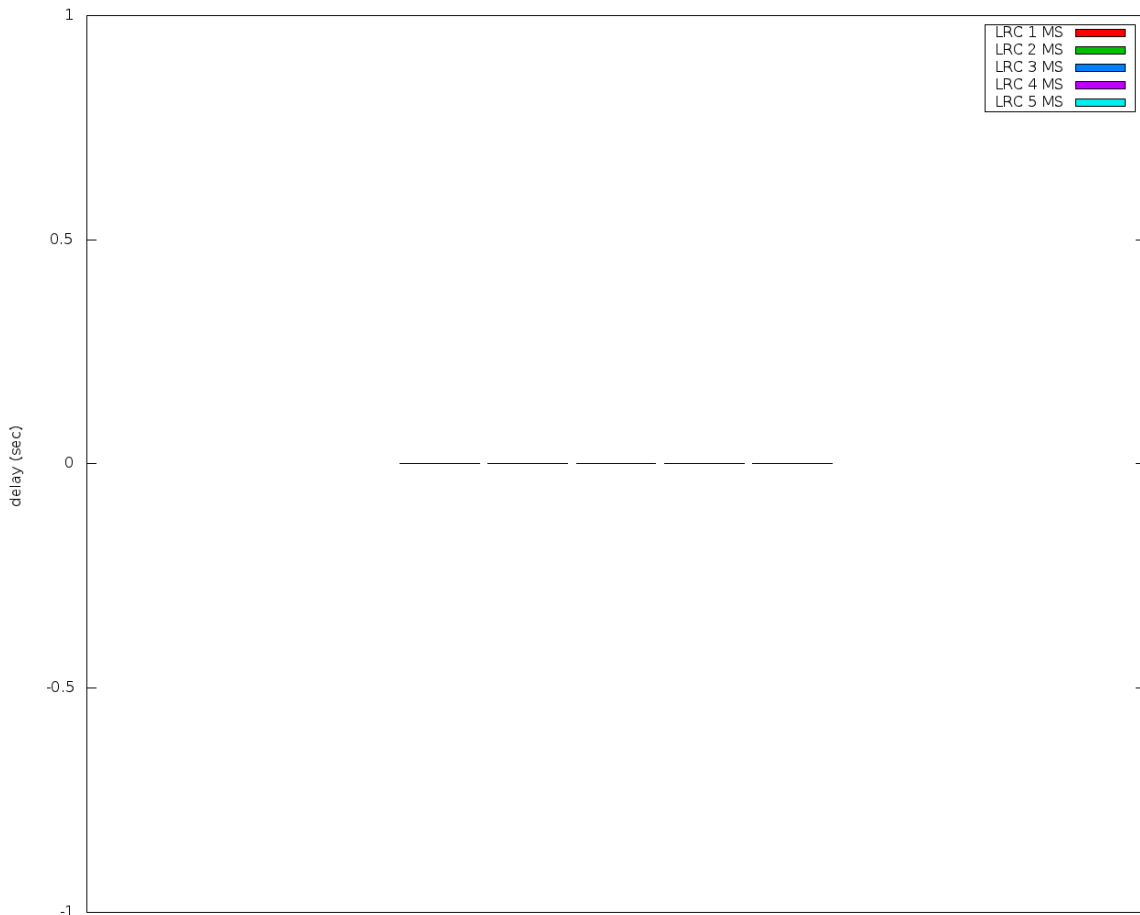


Εικόνα 73: Delay κατά την διάρκεια της προσομοίωσης (LRC module)

### 6.4.3 Περαιτέρω διερεύνηση

Για να έχουμε μια καλύτερη αντίληψη για την συμπεριφορά του module το σενάριο επαναλήφθηκε με ίδιες ρυθμίσεις αλλά με διαφορετικό αριθμό σταθμών κάθε φορά, ορίζοντας ένα χρονικό κενό μεταξύ της έναρξης εκπομπής δύο σταθμών της τάξης του μισού δευτερολέπτου. Συγκεκριμένα έγιναν δοκιμές για 1,2,3,4 και 5 σταθμούς συνδρομητές. Τα αποτελέσματα των προσομοιώσεων φαίνονται στις παρακάτω εικόνες.

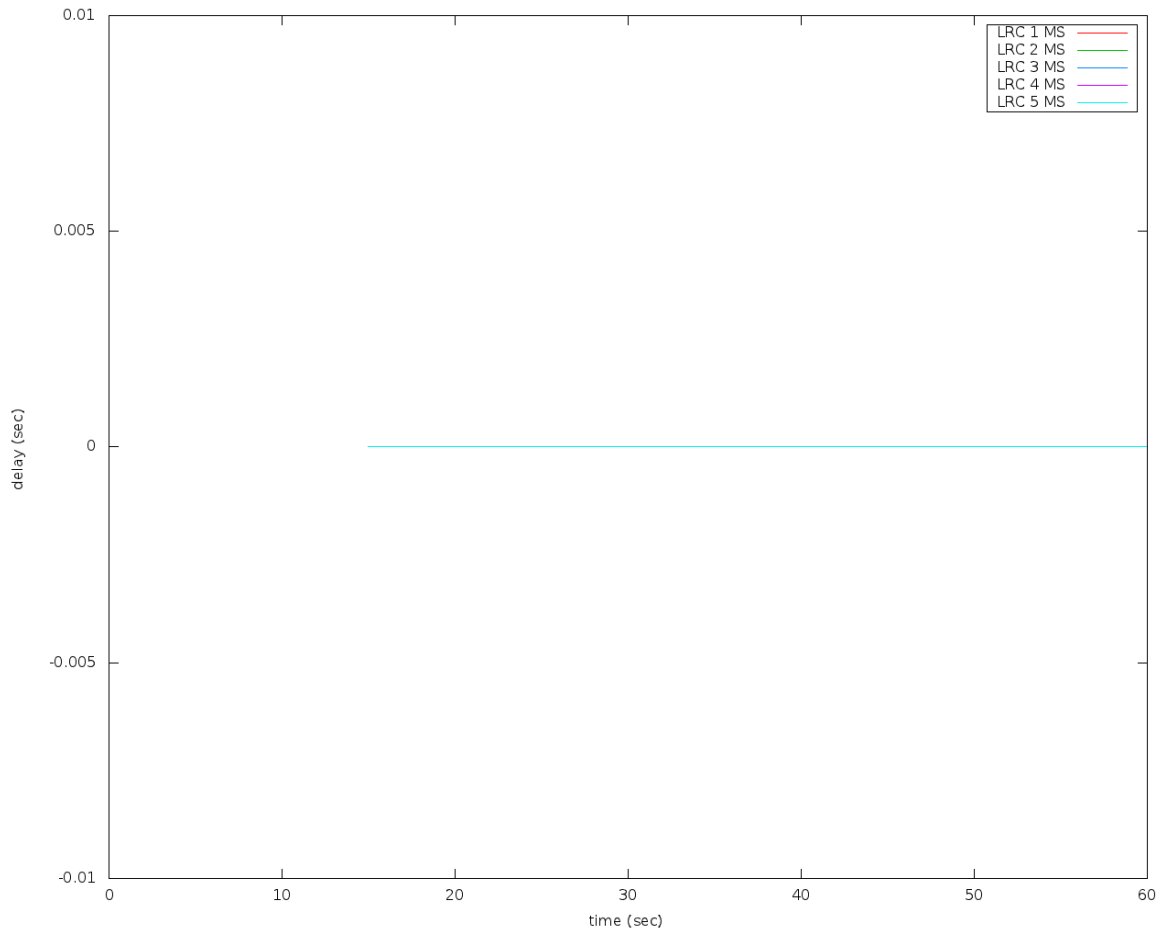
Στην εικόνα 74 παρατηρούμε ότι παρόλο που αυξάνουμε τους σταθμούς συνδρομητές, δεν υπάρχει κάποια αλλαγή στην καθυστέρηση του δικτύου η οποία παραμένει μηδενική.



Εικόνα 74: Μέσο delay για τις διαφορετικές προσομοιώσεις (LRC module)

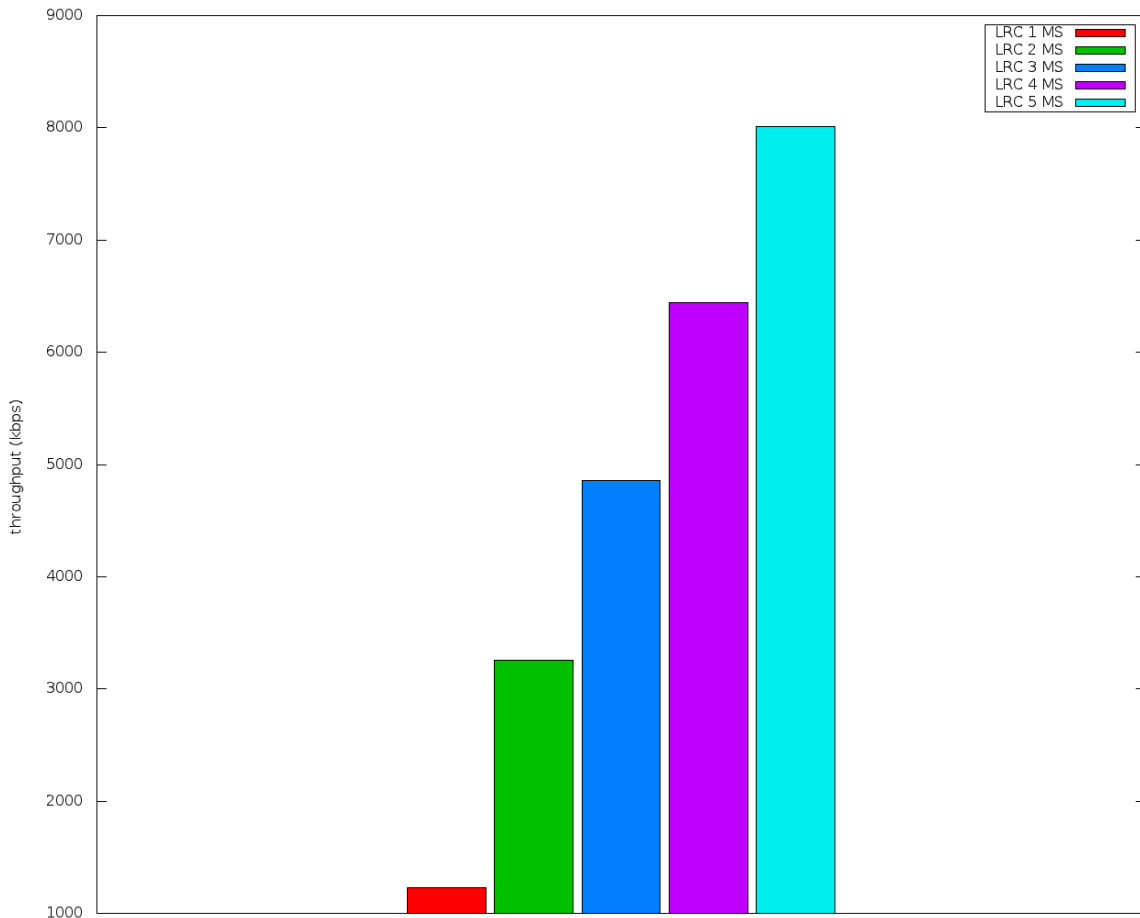
Στην εικόνα 75 βλέπουμε ξανά την καθυστέρηση στο δίκτυο, αυτή τη φορά όμως πιο αναλυτικά κατά την διάρκεια της προσομοίωσης. Παρατηρούμε και εδώ ότι δεν υπάρχει καμία διακύμανση στην καθυστέρηση του δικτύου σε καμία χρονική στιγμή της προσομοίωσης.

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου



Εικόνα 75: Αναλυτικό delay για τις διαφορετικές προσομοιώσεις (LRC module)

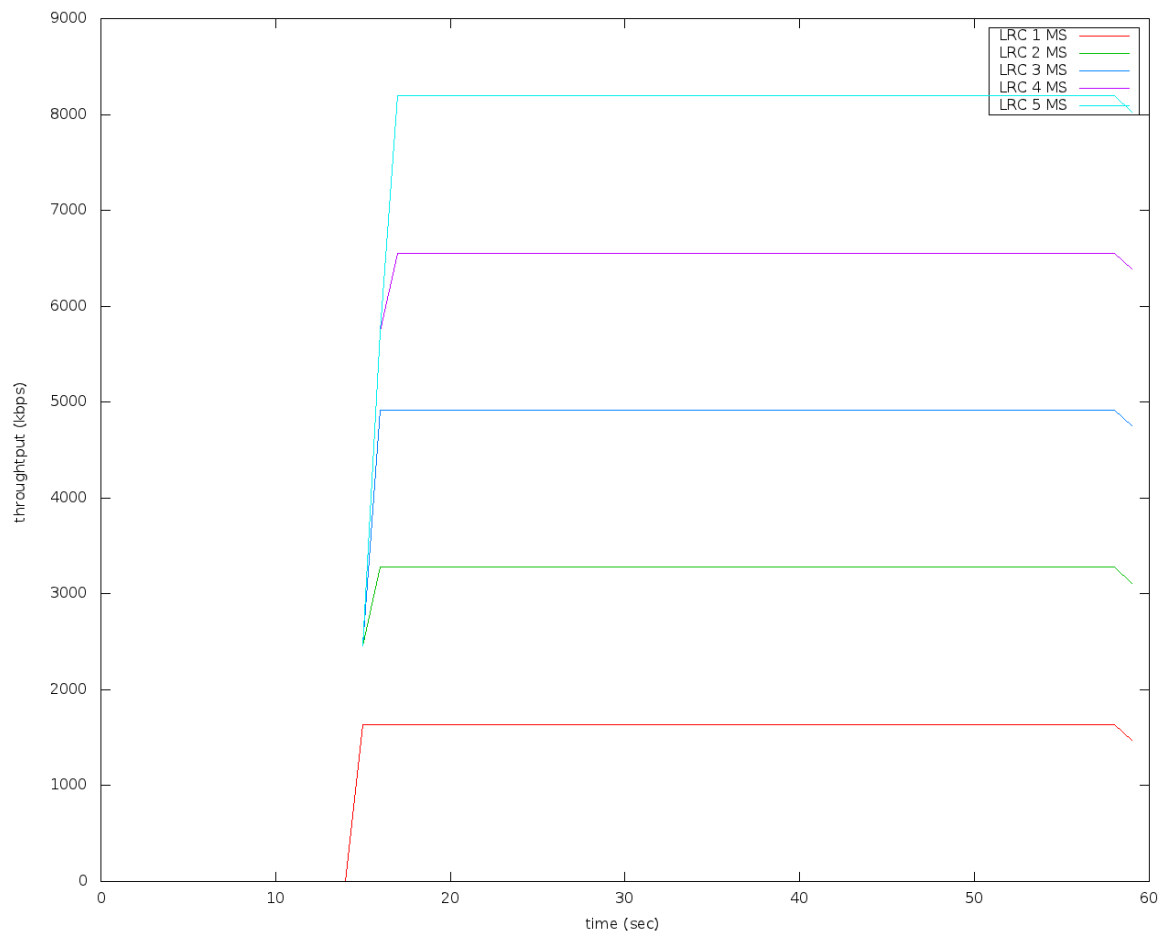
Στην εικόνα 76 όπου παρουσιάζεται το μέσο throughput για τις διαφορετικές προσομοιώσεις παρατηρούμε ότι με την χρήση δύο σταθμών έχουμε πολύ μεγάλη αύξηση στο throughput του δικτύου σε σχέση με τον έναν σταθμό. Από εκεί και πέρα στην χρήση περαιτέρω σταθμών υπάρχει πάλι μια αύξηση αλλά πιο ήπια.



Εικόνα 76: Μέσο throughput για τις διαφορετικές προσομοιώσεις (LRC module)

Στην εικόνα 77 τέλος όπου εμφανίζεται το throughput καθ' όλη την διάρκεια της προσομοίωσης παρατηρούμε μια κατακόρυφη αύξησή του κατά της αρχή εκπομπής των σταθμών, και μια διατήρηση των επιπέδων μέχρι το τέλος της προσομοίωσης.

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου



Εικόνα 77: Αναλυτικό throughput για τις διαφορετικές προσομοιώσεις (LRC module)

## **7 Ns2mesh80216 module**

### **7.1 Εισαγωγή**

Η υλοποίηση αυτή δημιουργήθηκε από την συνεργασία του Συλλόγου Δικτύωσης Υπολογιστών του πανεπιστημίου της Πίζας και του εργαστηρίου ασύρματης δικτύωσης του τεχνολογικού ιδρύματος της Γεωργίας.

Το πρόσθετο αυτό επιτρέπει την προσομοίωση ασυρμάτων δικτύων πλέγματος υπό το πρότυπο IEEE 802.16e στον προσομοιωτή NS2. Δεν υπάρχει υλοποίηση προς το παρόν για PMP δίκτυα. Οι συναρτήσεις για την ενεργοποίηση μετάδοσης δεδομένων στο επίπεδο MAC είναι πλήρως υλοποιημένες. Η πρόσβαση στο υπο-πλαίσιο δεδομένων γίνεται μέσω της χειραψίας τριών σημείων όπως έχει καθοριστεί από το πρότυπο, ενώ ο χρονοπρογραμματισμός έχει υλοποιηθεί σύμφωνα με τον αλγόριθμο Fair End-to-end Bandwidth Access (FEBA). Η πρόσβαση στο υπο-πλαίσιο ελέγχου έχει υλοποιηθεί σύμφωνα με το πρότυπο διαμοιρασμένης διαδικασίας εκλογών (distributed election procedure).

### **7.2 Documentation**

#### **7.2.1 Δομή της υλοποίησης**

Η υλοποίηση αυτή συνοδεύεται με δύο ακόμα πρόσθετα. Το ns2measure για την εξαγωγή μετρικών για το σενάριο και το ns2voip για υποστήριξη VoIP επικοινωνιών.

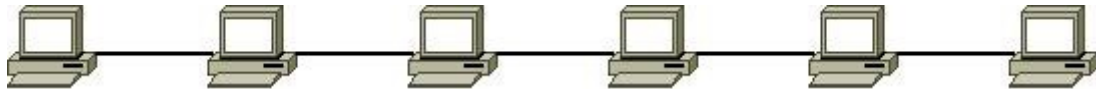
Τα νέα στοιχεία τα οποία ορίζονται σε αυτό το module σχετικά με το WiMAX είναι καταρχάς η υλοποίηση του MAC επιπέδου ώστε να μπορεί να λειτουργήσει για mesh δίκτυα. Επιπλέον έχει υλοποιηθεί στο φυσικό επίπεδο το OFDM.

#### **7.2.2 Υποστηριζόμενες τοπολογίες [9]**



Το πρόσθετο έχει σχεδιαστεί ώστε να μπορεί να υποστηρίξει τις παρακάτω τοπολογίες ασύρματων δικτύων πλέγματος. Για διευκόλυνση θα αναφέρουμε τον αριθμό των σταθμών ως “V” και τον αριθμό των συνδέσεων με “E”.

#### 7.2.2.1 Τοπολογία αλυσίδας (chain)

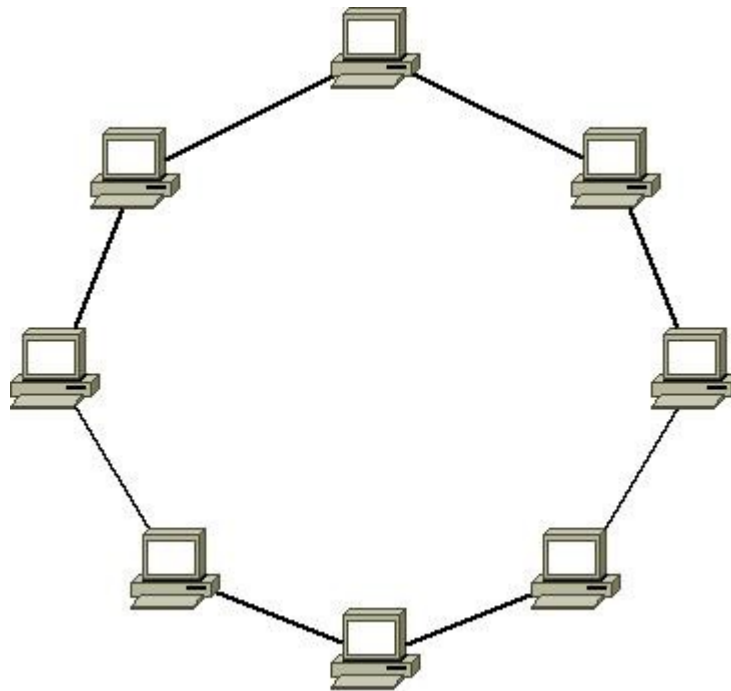


Εικόνα 78: Τοπολογία αλυσίδας (ns2mesh80216 module)

Στην τοπολογία αλυσίδας οι σταθμοί είναι συνδεδεμένοι συριακά με τους εκατέρωθεν σταθμούς εκτός από τους ακριανούς σταθμούς που έχουν σύνδεση μόνο προς την εσωτερική πλευρά της τοπολογίας. Για να επικοινωνήσουν μεταξύ τους για παράδειγμα οι δύο σταθμοί που βρίσκονται στα άκρα θα πρέπει να περάσουν την επικοινωνία τους μέσω των καναλιών επικοινωνίας όλων των υπόλοιπων υπολογιστών.

Για την τρέχουσα τοπολογία έχουμε ότι  $V = \text{opp}(n)$  και  $E = V - 1$ .

#### 7.2.2.2 Τοπολογία δακτυλίου (ring)

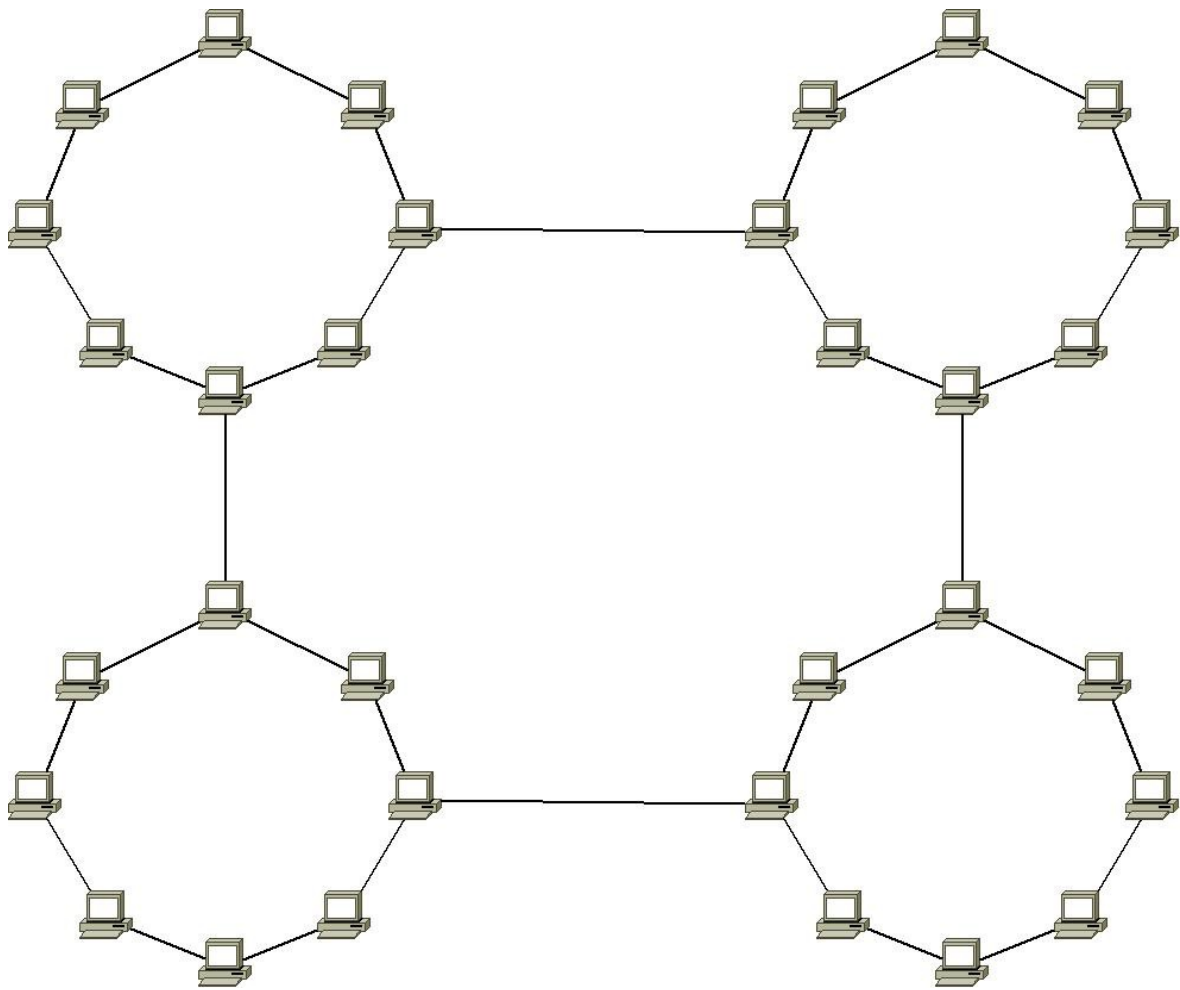


Εικόνα 79: Τοπολογία δακτυλίου (ns2mesh80216 module)

Η τοπολογία δακτυλίου μοιάζει με την τοπολογία αλυσίδας. Η μόνη διαφορά που έχουν είναι ότι εδώ δεν υπάρχουν ακριανοί σταθμοί μιας και όλοι οι υπολογιστές συνδέονται συριακά με τους εκατέρωθεν τους.

Για την τρέχουσα τοπολογία έχουμε ότι  $V = \text{opp}(n)$  και  $E = V$ .

### 7.2.2.3 Τοπολογία πολλαπλού δακτυλίου (multiring)



Εικόνα 80: Τοπολογία πολλαπλού δακτυλίου (ns2mesh80216 module)

Η τοπολογία πολλαπλού δακτυλίου αποτελείται από πολλαπλές τοπολογίες δακτυλίου συνδεδεμένες μεταξύ τους ξανά σε τοπολογία δακτυλίου. Σε κάθε υπο-δακτυλίδι ένας σταθμός παίζει τον διαμεσολαβητή για την σύνδεση του υπο-δακτυλιδιού αυτού με τα υπόλοιπα υπο-δακτυλίδια.

Για την τρέχουσα τοπολογία έχουμε ότι:

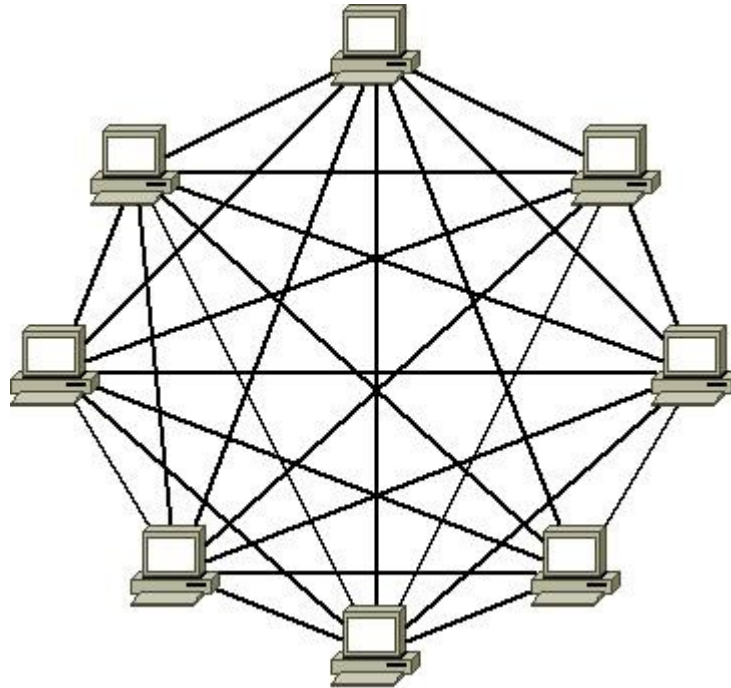
$$V = \text{opt}(n)$$

$B = \text{opt}(\text{branches}) \rightarrow$  πρέπει να είναι ζυγός αριθμός

$$E = V * B / 2 \quad \text{αν } B < N$$

$$E = V * (V - 1) / 2 \quad \text{αν } B = N$$

#### 7.2.2.4 Τοπολογία κλίκας (clique)

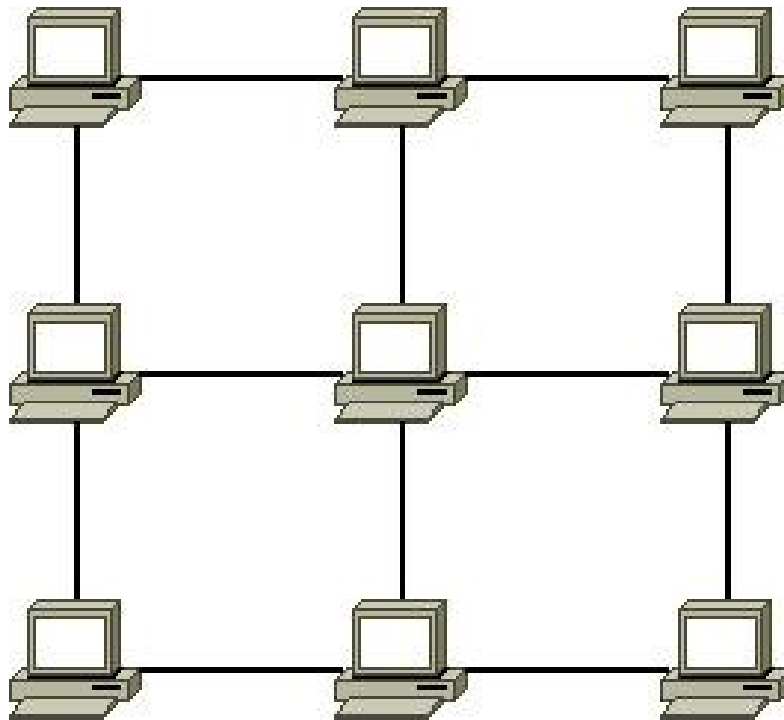


Εικόνα 81: Τοπολογία κλίκας (ns2mesh80216 module)

Στην τοπολογία κλίκας κάθε σταθμός είναι συνδεδεμένος με κάθε έναν από όλους τους υπόλοιπους. Κάθε σταθμός δηλαδή έχει αφιερωμένο κανάλι επικοινωνίας με κάθε άλλο σταθμό.

Γι' αυτήν την τοπολογία ισχύει ότι  $V = \text{ort}(n)$  και  $E = V * (V-1) / 2$ .

#### 7.2.2.5 Τοπολογία πλέγματος (grid)

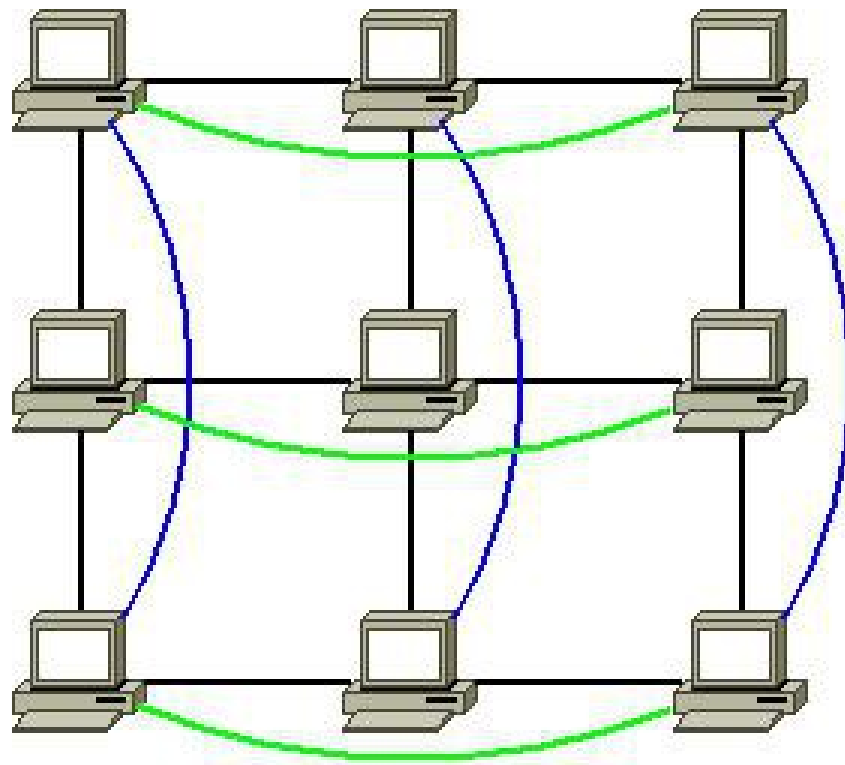


Εικόνα 82: Τοπολογία πλέγματος (ns2mesh80216 module)

Στην τοπολογία πλέγματος μπορούμε να την φανταστούμε σαν ένα τετράγωνο το οποίο χωρίζεται σε περισσότερα μικρότερα τετράγωνα στο εσωτερικό του. Κάθε ακμή των μικρών αυτών τετραγώνων αποτελεί έναν σταθμό και οι πλευρές αποτελούν τους συνδέσμους μεταξύ των σταθμών.

Για την τοπολογία αυτή ισχύει ότι  $V = \text{opt}(n)^2$  και  $E = 2 * \text{opt}(n) * (\text{opt}(n) - 1)$

#### 7.2.2.6 Τοπολογία πλήρους πλέγματος (grid-full)

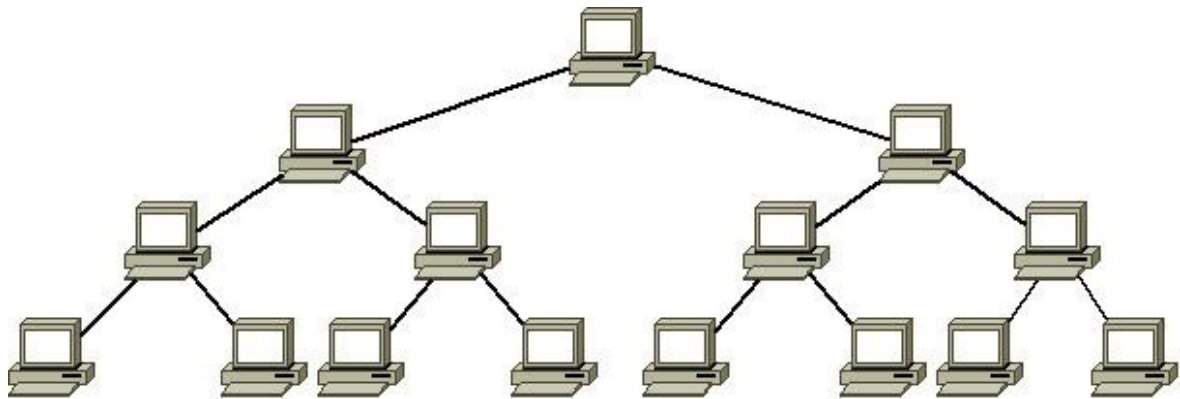


Εικόνα 83: Τοπολογία πλήρους πλέγματος (ns2mesh80216 module)

Η τοπολογία πλήρους πλέγματος είναι στην ουσία μια τοπολογία πλέγματος μόνο που οι εξωτερικοί σταθμοί έχουν σύνδεση με τους εξωτερικούς σταθμούς της απέναντι πλευράς.

Για την τοπολογία αυτή ισχύει ότι  $V = \text{opt}(n)^2$  και  $E = 2 * \text{opt}(n) * (\text{opt}(n) - 1)$ .

### 7.2.2.7 Τοπολογία δυαδικού δέντρου (bintree)

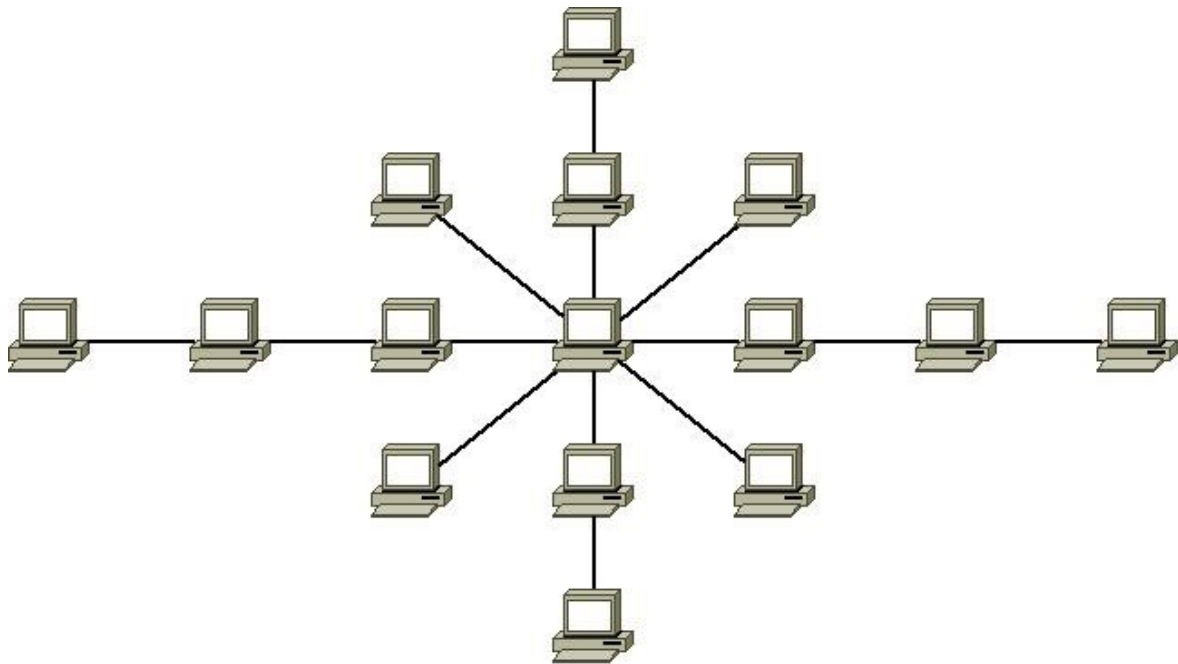


Εικόνα 84: Τοπολογία δυαδικού δέντρου (ns2mesh80216 module)

Στην τοπολογία δυαδικού δέντρου υπάρχει ένας σταθμός που αποτελεί την ρίζα της τοπολογίας. Κάτω από αυτόν υπάρχουν ένας ή δύο σταθμοί που συνδέονται άμεσα με τον σταθμό ρίζας οι οποίοι με την σειρά τους έχουν έναν ή δύο σταθμούς που συνδέονται με αυτούς κ.ο.κ.

Γι' αυτή την τοπολογία ισχύει ότι  $V = 2^{\text{opt}(n)} - 1$  και  $E = 2^{\text{opt}(n)} - 2$ .

#### 7.2.2.8 Τοπολογία αστέρα (star)



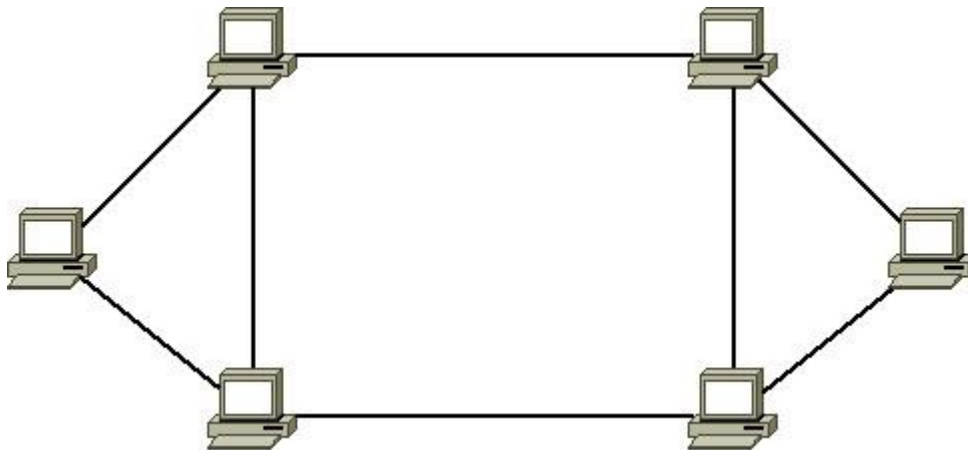
Εικόνα 85: Τοπολογία αστέρα (ns2mesh80216 module)

Στην τοπολογία αστέρα οι σταθμοί είναι έτσι συνδεδεμένοι ώστε να υπάρχει ένας κεντρικός σταθμός πάνω στον οποίο συνδέονται τοπολογίες αλυσίδας. Αποτελεί δηλαδή έναν συνδετικό κρίκο για πολλαπλές τοπολογίες αλυσίδας.

Γι' αυτήν την τοπολογία ισχύει ότι  $V = \text{opt}(\text{braches}) * (\text{opt}(n)-1) + 1$  και  $E = \text{opt}(\text{braches}) * (\text{opt}(n)-1)$ .

#### 7.2.2.9 Τοπολογία εξαγώνου (hexagon)



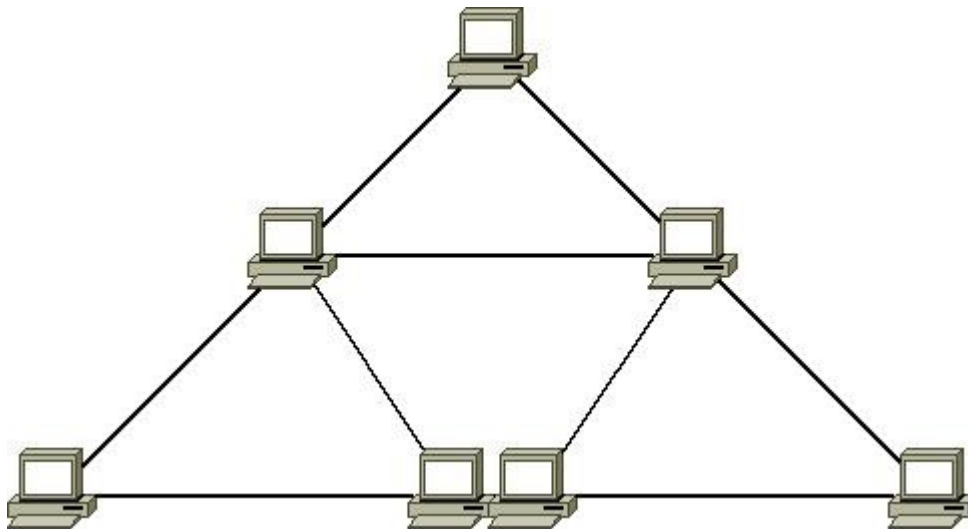


Εικόνα 86: Τοπολογία εξαγώνου (ns2mesh80216 module)

Στην τοπολογία εξαγώνου οι σταθμοί είναι τοποθετημένοι ώστε να θυμίζουν εξαγώνο. Εκτός από τις εξωτερικές συνδέσεις των σταθμών όμως μεταξύ τους υπάρχουν και δύο ακόμα συνδέσεις μεταξύ των τεσσάρων ακμών από τις έξι ώστε η τοπολογία να θυμίζει ένα τετράγωνο μέσα σε ένα εξαγώνο.

Γι' αυτήν την τοπολογία ισχύει ότι  $V=6$  και  $E=8$ .

#### 7.2.2.10 Τριγωνική τοπολογία (triangular)



Εικόνα 87: Τριγωνική τοπολογία (ns2mesh80216 module)

Η τριγωνική τοπολογία αποτελείται από ένα μεγάλο τρίγωνο το οποίο περιέχει εγγεγραμμένα τρίγωνα. Προς το παρόν η τοπολογία αυτή δεν έχει υλοποιηθεί.

Γι' αυτήν την τοπολογία ισχύει ότι  $V = (opt(n)+1)*(opt(n)+2)/2$  και  $E = 2*opt(n)*(opt(n)+1)/2$ .

### 7.2.3 Άντληση πληροφοριών για την προσομοίωση

Η συγκεκριμένη υλοποίηση δεν υποστηρίζει τον κλασικό τρόπο με τον οποίο το ns βγάζει τα αποτελέσματά του, δηλαδή ένα trace αρχείο από το οποίο ο χρήστης μπορεί να αντλήσει πληροφορίες με scripts και να τα εμφανίσει με την μορφή που θέλει. Στην τρέχουσα υλοποίηση έχει ενσωματωθεί ένα πρόγραμμα ονόματι ns2measure το οποίο κατά την διάρκεια της προσομοίωσης κρατάει σε δικές τους μεταβλητές σύμφωνα με το πως είναι ορισμένο διάφορα στατιστικά τα οποία στο τέλος αποθηκεύονται σε ένα binary αρχείο. Το binary αρχείο αυτό μπορεί να διαβαστεί με την εντολή recover που υπάρχει στο πακέτο stat-utilities, το οποίο μπορεί κανείς να κατεβάσει από την σελίδα του project. Ο χρήστης από πλευράς του έχει μόνο την υποχρέωση να δηλώσει πριν ξεκινήσει το σενάριο ποιές μετρικές θέλει να καταγράψει. Το σύνολο των μετρικών που υποστηρίζονται από το module είναι οι παρακάτω: [9]

#### End-to-end traffic flows

- ◆ e2e\_owd\_a (average, per flow) : δίνει το delay από άκρη σε άκρη σε δευτερόλεπτα.
- ◆ e2e\_owd\_d (distribution, per flow) : δίνει το delay από άκρη σε άκρη, μετρούμενο σε δευτερόλεπτα όπου ο χρόνος έναρξης της μέτρησης, το τέλος μέτρησης και το βήμα κατά το οποίο θα γίνεται η μέτρηση ορίζεται από τον χρήστη.
- ◆ e2e\_tpt (average, per flow) : δίνει το throughput από άκρη σε άκρη σε bytes/s.
- ◆ e2e\_owpl (average, per flow) : δίνει το packet loss από άκρη σε άκρη σε packets/s.

- ◆ e2e\_ipdv\_d (distribution, per flow) : δίνει την διακύμανση του inter-packet delay όπου ο χρόνος έναρξης της μέτρησης, το τέλος μέτρησης και το βήμα κατά το οποίο θα γίνεται η μέτρηση ορίζεται από τον χρήστη.

## VoIP

- ◆ voip\_mos\_talkspurt (average, per flow) : δίνει το μέσο Mean Opinion Score ανά talkspurt σε μια VoIP ροή δεδομένων.
- ◆ voip\_%\_of\_bad\_talkspurt (average, per flow) : δίνει την αναλογία του VoIP talkspurt το οποίο υπέστη “χαμηλή ποιότητα” (“bad quality”) όπως αυτή έχει οριστεί στο αρχείο wimax/tcl/traffic.tcl.

## TCP

- ◆ tcp\_cwnd\_a (average, per flow) : δίνει το congestion window στους κινητούς σταθμούς συνδρομητές.
- ◆ tcp\_cwnd\_a (distribution, per flow) : δίνει το congestion window για κάθε κινητό σταθμό συνδρομητή.
- ◆ tcp\_ssthresh (average, per flow) : δίνει το slot-start threshold των κινητών σταθμών συνδρομητών.
- ◆ tcp\_rtt (average, per flow) : δίνει τον round-trip χρόνο σε δευτερόλεπτα.
- ◆ tcp\_stt (average, per flow) : δίνει τον round-trip χρόνο εξομαλυμένο σε δευτερόλεπτα.
- ◆ tcp\_rexmitpack (average, per flow) : δίνει τον ρυθμό αναμετάδοσης σε packets/s.
- ◆ tcp\_rexmitbytes (average, per flow) : δίνει τον ρυθμό αναμετάδοσης σε bytes/s.

## Distribution election

- ◆ `wimsh_dsch_election_slots` (average, per node) : δίνει τον αριθμό των υποδοχών ελέγχου μεταξύ δύο συνεχόμενων εισόδων στο υπο-πλαίσιο ελέγχου για αποστολή MSH-DSCH μηνυμάτων.
- ◆ `wimsh_dsch_election_slots_d` (distribution, per node) : δίνει τον αριθμό των υποδοχών ελέγχου μεταξύ δύο συνεχόμενων εισόδων στο υπο-πλαίσιο ελέγχου για αποστολή MSH-DSCH μηνυμάτων ορίζοντας ποιες υποδοχές θα ληφθούν υπ' όψιν και με ποιο βήμα.
- ◆ `wimsh_dsch_inter_frame_rcv` (average, per node) : δίνει τον χρόνο μεταξύ δύο συνεχόμενων MSH-DSCH μηνυμάτων που λήφθηκαν από έναν σταθμό σε frames.
- ◆ `wimsh_dsch_inter_time_rcv` (average, per node) : δίνει τον χρόνο μεταξύ δύο συνεχόμενων MSH-DSCH μηνυμάτων που λήφθηκαν από έναν σταθμό σε δευτερόλεπτα.
- ◆ `wimsh_dsch_inter_time_rcv_d` (distribution, per node): δίνει τον χρόνο μεταξύ δύο συνεχόμενων MSH-DSCH μηνυμάτων που λήφθηκαν από έναν σταθμό σε δευτερόλεπτα ορίζοντας τον χρόνο εκκίνησης και τέλους την μέτρησης καθώς και το βήμα.
- ◆ `wimsh_ncfg_election_slots` (average, per node) : δίνει τον αριθμό των υποδοχών ελέγχου μεταξύ δύο συνεχόμενων εισόδων στο υπο-πλαίσιο ελέγχου για αποστολή MSH-NCFG μηνυμάτων.
- ◆ `wimsh_ncfg_inter_frame` (average, per node) : δίνει τον χρόνο μεταξύ δύο συνεχόμενων MSH-NCFG μηνυμάτων που στάλθηκαν από έναν κόμβο σε frames.
- ◆ `wimsh_ncfg_inter_time` (average, per node) : δίνει τον χρόνο μεταξύ δύο συνεχόμενων MSH-NCFG μηνυμάτων που στάλθηκαν από έναν κόμβο σε δευτερόλεπτα.

- ◆ `wimsh_nent_inter_frame` (average, per node) : δίνει τον χρόνο μεταξύ δύο συνεχόμενων MSH-NENT μηνυμάτων που στάλθηκαν από έναν κόμβο σε frames.
- ◆ `wimsh_nent_inter_time` (average, per node) : δίνει τον χρόνο μεταξύ δύο συνεχόμενων MSH-NENT μηνυμάτων που στάλθηκαν από έναν κόμβο σε δευτερόλεπτα.
- ◆ `wimsh_dsch_inter_frame_snd` (average, per node) : δίνει τον χρόνο μεταξύ δύο συνεχόμενων MSH-DSCH μηνυμάτων που στάλθηκαν από έναν κόμβο σε frames.
- ◆ `wimsh_dsch_inter_time_snd` (average, per node) : δίνει τον χρόνο μεταξύ δύο συνεχόμενων MSH-DSCH μηνυμάτων που στάλθηκαν από έναν κόμβο σε δευτερόλεπτα.
- ◆ `wimsh_dsch_inter_time_snd_d` (distribution, per node) : δίνει τον χρόνο μεταξύ δύο συνεχόμενων MSH-DSCH μηνυμάτων που στάλθηκαν από έναν σταθμό ορίζοντας τον χρόνο εκκίνησης και τέλους την μέτρησης καθώς και το βήμα.
- ◆ `wimsh_dsch_size_a` (average, per node) : δίνει το μέγεθος του MSH-DSCH.
- ◆ `wimsh_dsch_size_d` (distribution, per node) : δίνει το μέγεθος των MSH-DSCH μηνυμάτων που στάλθηκαν από έναν κόμβο ορίζοντας έναρξη, λήξη και βήμα για τα bytes.
- ◆ `wimsh_dsch_error` (average, per node) : δίνει την αναλογία των κατεστραμμένων MSH-DSCH μηνυμάτων επί του συνόλου που λήφθηκε από έναν κόμβο
- ◆ `wimsh_dsch_competing` (average, per node) : δίνει τον αριθμό των ανταγωνιστών ανά υποδοχή κατά το virtual contention.

### **Network entry procedure (disabled by default)**

- ◆ `wimsh_sponsor_latency` (average, per node) : δίνει την καθυστέρηση για το άνοιγμα του sponsor καναλιού σε δευτερόλεπτα.
- ◆ `wimsh_linkest_latency` (average, per node) : δίνει την καθυστέρηση για την δημιουργία των γειτονικών συνδέσμων σε δευτερόλεπτα.
- ◆ `wimsh_scan_latency` (average, per node) : δίνει την καθυστέρηση για την εκτέλεση ενός αρχικού σαρώματος σε δευτερόλεπτα.
- ◆ `udp_tunnel_delay` (average, per node) : δίνει την καθυστέρηση για την δημιουργία του UPD tunnel για την ρύθμιση των κόμβων σε δευτερόλεπτα.

### **Three-way handshake mechanism**

- ◆ `wimsh_gnt_in` (average, per node) : δίνει τα bytes που εκχωρήθηκαν στον κόμβο.
- ◆ `wimsh_gnt_size` (average, per node) : δίνει το μέγεθος της εκχώρησης σε bytes.
- ◆ `wimsh_gnt_out` (average, per node) : δίνει τα bytes που εκχωρήθηκαν από τον κόμβο.
- ◆ `wimsh_cnf_in` (average, per node) : δίνει τα bytes που επιβεβαιώθηκαν στον κόμβο.
- ◆ `wimsh_cnf_size` (average, per node) : δίνει το μέγεθος την επιβεβαίωσης σε bytes.
- ◆ `wimsh_cnf_out` (average, per node) : δίνει τα bytes που επιβεβαιώθηκαν από τον κόμβο.
- ◆ `wimsh_req_in` (average, per node) : δίνει τα bytes που ζήτησε ο κόμβος.
- ◆ `wimsh_req_out` (average, per node) : δίνει τα bytes που ζητήθηκαν από τον κόμβο.

- ◆ `wimsh_regnt_in` (average, per node) : δίνει τα bytes που επανα-εκχωρήθηκαν στον κόμβο.

## MAC

- ◆ `wimsh_mac_tpt` (average, per node) : δίνει το throughput του κόμβου σε bytes/s.
- ◆ `wimsh_chn_data_tpt` (average, per channel) : δίνει το data rate σε bytes/s.
- ◆ `wimsh_chn_ctrl_tpt` (average, per channel) : δίνει το control rate σε bytes/s.
- ◆ `wimsh_drop_overflow` (average, per node) : δίνει το buffer overflow rate σε packets/s.
- ◆ `wimsh_bufsize_mac_a` (average, per node) : δίνει το buffer occupancy σε bytes.
- ◆ `wimsh_active_flows` (average, per node) : δίνει τον αριθμό των ενεργών ροών.
- ◆ `wimsh_dd_timeout` (average, per node) : δίνει το deadlock-detection timeout expiration rate.

Για να χρησιμοποιήσει κάποιος λοιπόν τις παραπάνω μετρικές θα πρέπει να τις ορίσει στο σενάριο του και μετά να τις διαβάσει με τον προαναφερθέν τρόπο.

## 7.3 Διαδικασία εγκατάστασης

Η εγκατάσταση του module έγινε σε ένα εικονικό μηχάνημα του λογισμικού virtualbox της sun, το οποίο είχε σαν λειτουργικό το Fedora 13.

Τα βήματα της εγκατάστασης είναι τα παρακάτω:

- 1) Κατεβάζουμε το `ns-allinone-2.33` από την παρακάτω διεύθυνση :

<http://sourceforge.net/projects/nsnam/files/allinone/>

- 2) Αποσυμπιέζουμε το αρχείο που κατεβάσαμε στην επιθυμητή τοποθεσία.
- 3) Κατεβάζουμε το patch από την σελίδα του project :  
<http://cngl.iet.unipi.it/archive/ns2mesh80216/ns2mesh80216-2.33-081113.patch.gz>
- 4) Από τον φάκελο του ns-allinone2.33/ns-2.33 τρέχουμε την παρακάτω εντολή (όπου /tmp/ υποθέτουμε ο φάκελος που βρίσκεται το patch):

```
gzip -dc /tmp/ns2mesh80216-2.33-081113.patch.gz | patch -Np1
```

- 5) Από τον φάκελο του ns-allinone-2.33 τρέχουμε την εντολή “./install”
- 6) Μετά την ολοκλήρωση της εγκατάστασης πρέπει να προσθέσουμε τα παρακάτω στο αρχείο ~/.bashrc :



## Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

```
#LD_LIBRARY_PATH

OTCL_LIB=/home/gkaran/ns-allinone-2.33/otcl-1.13

NS2_LIB=/home/gkaran/ns-allinone-2.33/lib

X11_LIB=/usr/X11R6/lib

USR_LOCAL_LIB=/usr/local/lib

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$X11_LIB:
$USR_LOCAL_LIB

#TCL_LIBRARY

TCL_LIB=/home/gkaran/ns-allinone-2.33/tcl8.4.18/library

USR_LIB=/usr/lib

export TCL_LIBRARY=$TCL_LIB:$USR_LIB

#PATH

XGRAPH=/home/gkaran/ns-allinone-2.33/bin:/home/gkaran/ns-allinone-
2.33/tcl8.4.18/unix:/home/gkaran/ns-allinone-2.33/tk8.4.18/unix

NS=/home/gkaran/ns-allinone-2.33/ns-2.33

NAM=/home/gkaran/ns-allinone-2.33/nam-1.13

PATH=$PATH:$XGRAPH:$NS:$NAM
```

Όπου /home/gkaran/ns-allinone-2.33/ βάζετε τον φάκελο που έχετε το ns-allinone-2.33

Προαιρετικά πριν το βήμα 6 αν θέλουμε να λειτουργήσει σωστά το nam ακολουθούμε τα παρακάτω βήματα:

- 1) Μέσα στον φάκελο του tk8.4.18 που βρίσκεται μέσα στον φάκελο του ns-allinone-2.33 δημιουργούμε ένα αρχείο με όνομα “tk-lastevent.patch” ή όπως αλλιώς θέλουμε και γράφουμε σε αυτό τις παρακάτω εντολές.

## Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

```
--- generic/tk.h.orig 2008-02-06 16:31:40.000000000 +0100
+++ generic/tk.h 2008-07-24 08:21:46.000000000 +0200
@@ -635,17 +635,15 @@
     *

*-----
-----
    */
-#define VirtualEvent      (LASTEvent)
-#define ActivateNotify    (LASTEvent + 1)
-#define DeactivateNotify  (LASTEvent + 2)
-#define MouseWheelEvent   (LASTEvent + 3)
-#define TK_LASTEVENT      (LASTEvent + 4)
+#define VirtualEvent      (MappingNotify + 1)
+#define ActivateNotify    (MappingNotify + 2)
+#define DeactivateNotify  (MappingNotify + 3)
+#define MouseWheelEvent   (MappingNotify + 4)
+#define TK_LASTEVENT      (MappingNotify + 5)
    #define MouseWheelMask  (1L << 28)
-
    #define ActivateMask    (1L << 29)
    #define VirtualEventMask (1L << 30)
-#define TK_LASTEVENT      (LASTEvent + 4)
    /*
```

- 2) Στον ίδιο φάκελο που είμαστε τρέχουμε την παρακάτω εντολή :  
“patch -p0 < tk-lastevent.patch” (χωρίς τα εισαγωγικά).
- 3) Αν η εντολή μας επιστρέψει σφάλμα μπορούμε να το κάνουμε χειροκίνητα.  
Ανοίγουμε το αρχείο tk.h στον φάκελο generic και περίπου στην 635 γραμμή πάμε και σβήνουμε όσες σειρές είναι στον πιο πάνω πλαίσιο με ένα “-“ μπροστά από το “#”. Στην συνέχεια προσθέτουμε τις σειρές που έχουν το “+” μπροστά από το “#”. Φροντίζουμε μόνο η σειρά των γραμμών να είναι όπως στο πλαίσιο πιο πάνω.

### 7.3.1 Προβλήματα εγκατάστασης

Κατά την διαδικασία της εγκατάστασης παρουσιάστηκε το ακόλουθο πρόβλημα. Στα παρακάτω αρχεία εμφανιζόταν μήνυμα λάθους ότι δεν υπάρχει μεταβλητή με το όνομα UINT\_MAX:

```
35  
17 Ns-allinone-2.33/ns-2.33/wimax/mesh/wimsh_topology.cc
```

```
35  
17 Ns-allinone-2.33/ns-2.33/wimax/mesh/wimsh_packet.h
```

Το παραπάνω πρόβλημα είχε ως αποτέλεσμα την διακοπή της διαδικασίας εγκατάστασης αφού δεν μπορούσε να γίνει σωστά compile ο κώδικας.

Για να αντιμετωπίσουμε αυτό το πρόβλημα έπρεπε να προσθέσουμε στα αρχεία αυτά στο ύψος των δηλώσεων την παρακάτω γραμμή κώδικα.

```
#define UINT_MAX 4294967295U
```

## 7.4 Προσομοίωση

### 7.4.1 Σενάριο προσομοίωσης

Μιας και το συγκεκριμένο module δεν υποστηρίζει PMP τοπολογίες παρά μόνο Mesh, το σενάριο στην προκειμένη περίπτωση έχει αλλαχθεί μερικώς ώστε να μπορέσει να μοιάζει όσο το δυνατόν περισσότερο με τα υπόλοιπα. Έτσι αυτό το σενάριο αναπαριστά μια CBR επικοινωνία μεταξύ δύο σταθμών συνδεδεμένων σε τοπολογία αλυσίδας σε ένα Mesh δίκτυο. Το μέγεθος των CBR πακέτων που ανταλλάσσονται είναι 1500 bytes, δεν ορίζουμε κάποιον μηχανισμό QoS και ο χρόνος προσομοίωσης ορίζεται στα 60 δευτερόλεπτα με την CBR επικοινωνία να ξεκινά στο 15ο δευτερόλεπτο και να τελειώνει στο τέλος της προσομοίωσης. Η έξοδος της προσομοίωσης όπως αναφέρθηκε και πιο πριν είναι ένα binary αρχείο από το οποίο αντλούμε μέσω συγκεκριμένου εργαλείου τα αποτελέσματα. Ο κώδικας TCL που εκφράζει το σενάριο της προσομοίωσης είναι διαθέσιμος στο παράρτημα Δ.

### 7.4.2 Αποτελέσματα προσομοίωσης

Στο συγκεκριμένο module τα στοιχεία τα οποία αντλούνται είναι στατικά και δεν κρίθηκε αναγκαία η εμφάνισή τους σε γράφημα. Συγκεκριμένα είχαμε σαν αποτελέσματα τα ακόλουθα:

- average delay : 4.87485 δευτερόλεπτα
- average throughput : 1523.4375 kbps

Όπως φαίνεται και από τα παραπάνω δυστυχώς το συγκεκριμένο module δεν μας δίνει πολλές δυνατότητες ως προς την εξαγωγή δεδομένων ώστε να τα αξιοποιήσουμε όπως θέλουμε.

## **8 Συμπεράσματα**

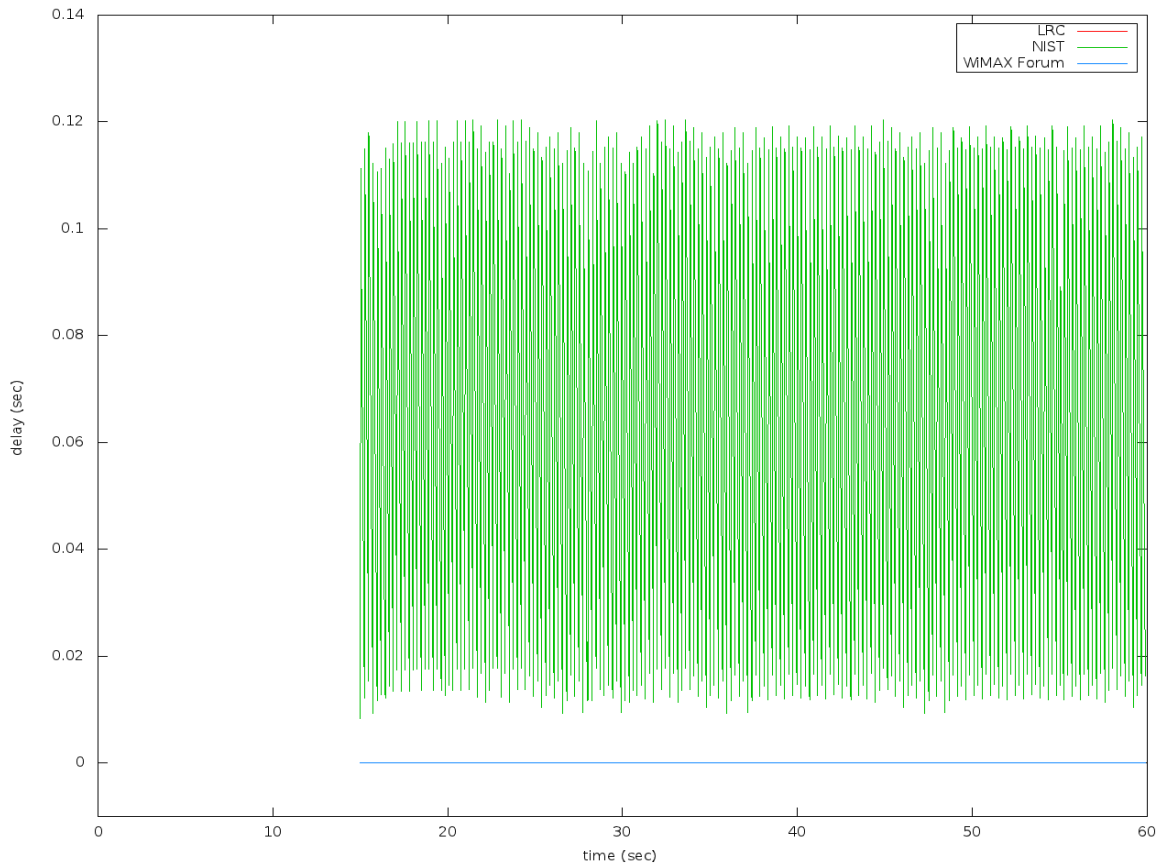
### **8.1 Εισαγωγή**

Σε αυτό το κεφάλαιο θα παρουσιαστούν συμπεράσματα που προκύπτουν από τα αποτελέσματα των προσομοιώσεων καθώς και προτάσεις που αφορούν την βελτίωση αυτών ή παρατηρήσεις για τυχόν ελλείψεις που παρουσιάζονται. Στο τέλος θα παρουσιαστεί ένας συγκριτικός πίνακας με το τι υποστηρίζει το κάθε module από αυτά που εξετάστηκαν ώστε να δοθεί στον αναγνώστη μια σχηματική εικόνα για τις ελλείψεις του κάθε module.

### **8.2 Συγκριτική παρουσίαση**

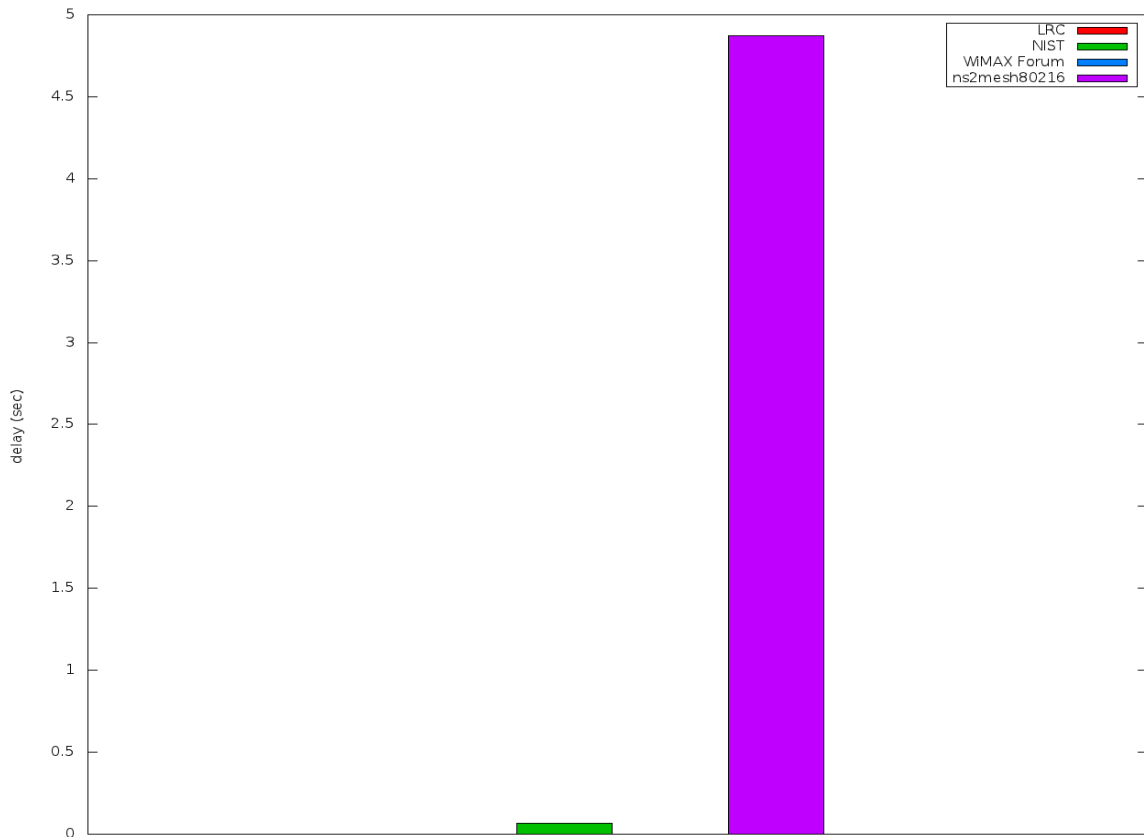
Μέχρι στιγμής έχουμε αναλύσει το κάθε module ξεχωριστά όπως και τα διαγράμματα των αποτελεσμάτων τους. Σε αυτό το κομμάτι της εργασίας λοιπόν θα επανεξετάσουμε λοιπόν συνοπτικά τα αποτελέσματα από όλα τα modules ταυτόχρονα με μια ανοχή στα αποτελέσματα μιας και όπως έχει προαναφερθεί κανένα από τα modules δεν υποστηρίζει ότι τα άλλα και κάθε σενάριο είναι γραμμένο έτσι ώστε να μπορεί να παίξει στο καθένα module ξεχωριστά.

Στην εικόνα 88 βλέπουμε ότι τα modules από το WiMAX Forum και το LRC συμπίπτουν με 0 delay καθ' όλη την διάρκεια της προσομοίωσης, ενώ το module από το NIST ταλαντεύεται μεταξύ 0.1 και 0.12 δευτερολέπτων. Σαφώς λοιπόν σε αυτό το κομμάτι υπερτερούν τα modules των LRC και WiMAX Forum. Δυστυχώς δεν μπορούμε να έχουμε την απαιτούμενη πληροφορία για να παρουσιάσουμε και το ns2mesh80216 module.



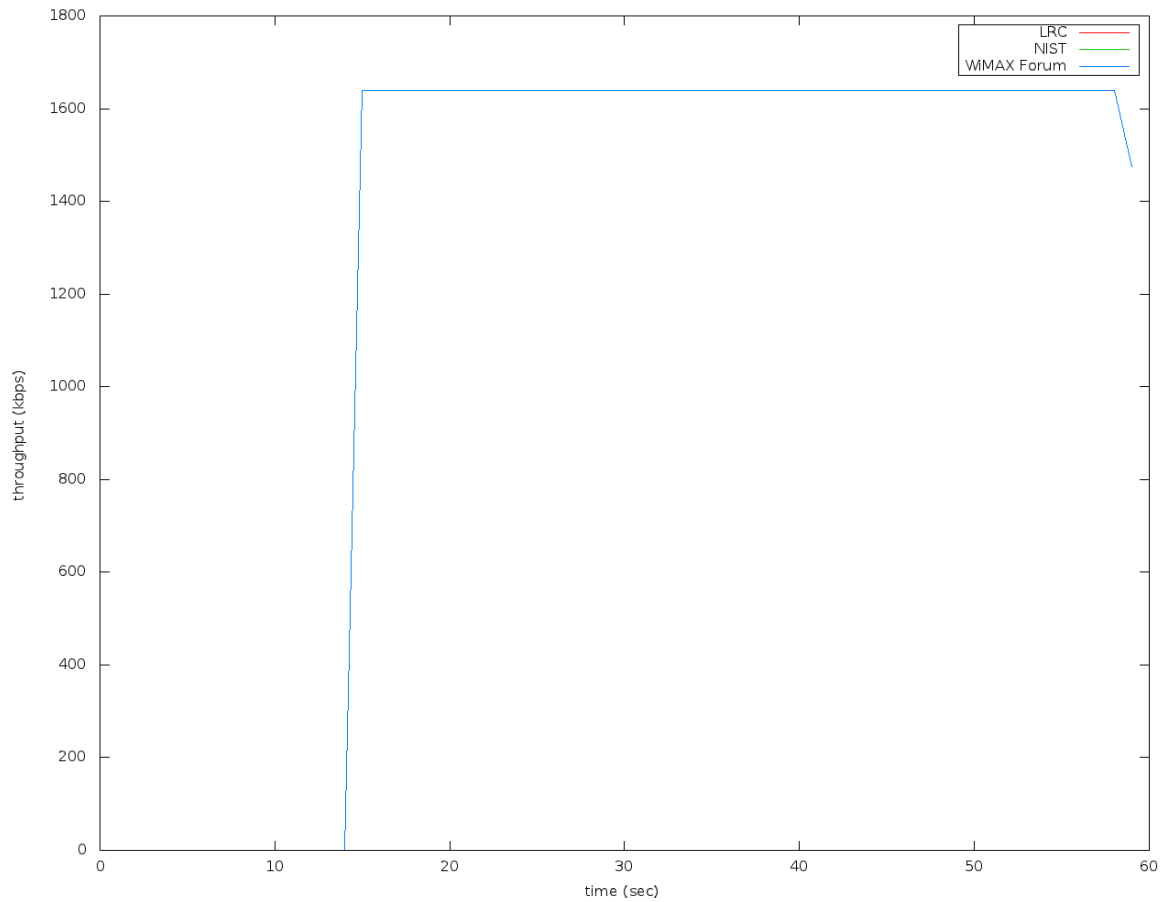
Εικόνα 88: Συγκριτικό delay καθ' όλη την προσομοίωση όλων των modules

Επιπλέον στην εικόνα 89 παρατηρούμε το μέσο όρο του delay κατά την διάρκεια της προσομοίωσης. Βλέπουμε λοιπόν ότι και εδώ ότι υπερτερούν κατά πολύ τα modules από το LRC και το WiMAX Forum καθώς έχουν μηδέν delay. Στην συνέχεια παρατηρούμε και εδώ ότι ο μέσος όρος του delay για το NIST module είναι σχετικά πιο υψηλός από τα άλλα δύο ενώ από το ns2mesh80216 module είναι υπερβολικά πιο υψηλό από τα άλλα τρία module.



Εικόνα 89: Συγκριτικό μέσο delay όλων των modules

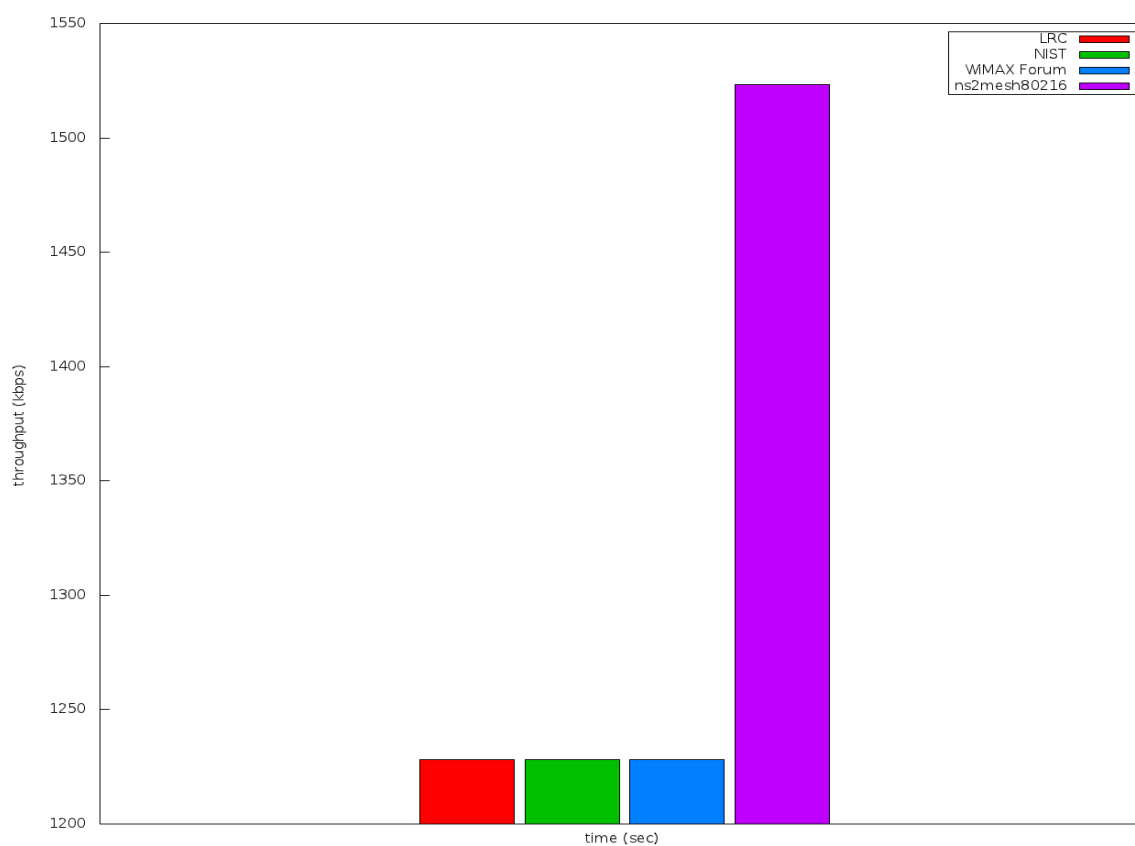
Στην εικόνα 90 εμφανίζεται το throughput καθ' όλη την διάρκεια της προσομοίωσης. Δυστυχώς και εδώ δεν μπορούμε να έχουμε την απαιτούμενη πληροφορία για το ns2mesh80216 module. Για τα υπόλοιπα module όμως όπως βλέπουμε το throughput είναι ίδιο και στα τρία καθ' όλη την διάρκεια της προσομοίωσης, κάτι που τα καθιστά ισάξια σε αυτόν τον τομέα.



Εικόνα 90: Συγκριτικό throughput κατά την διάρκεια της προσομοίωσης όλων των modules

Στην εικόνα 91 έχουμε το μέσο throughput για όλα τα modules. Εδώ παρατηρούμε ότι τα υπόλοιπα τρία module πέραν του ns2measure module κυμαίνονται στα ίδια επίπεδα. Όπως βλέπουμε το ns2mesh80216 module έχει αρκετά υψηλότερο μέσο throughput από τα υπόλοιπα τρία module.





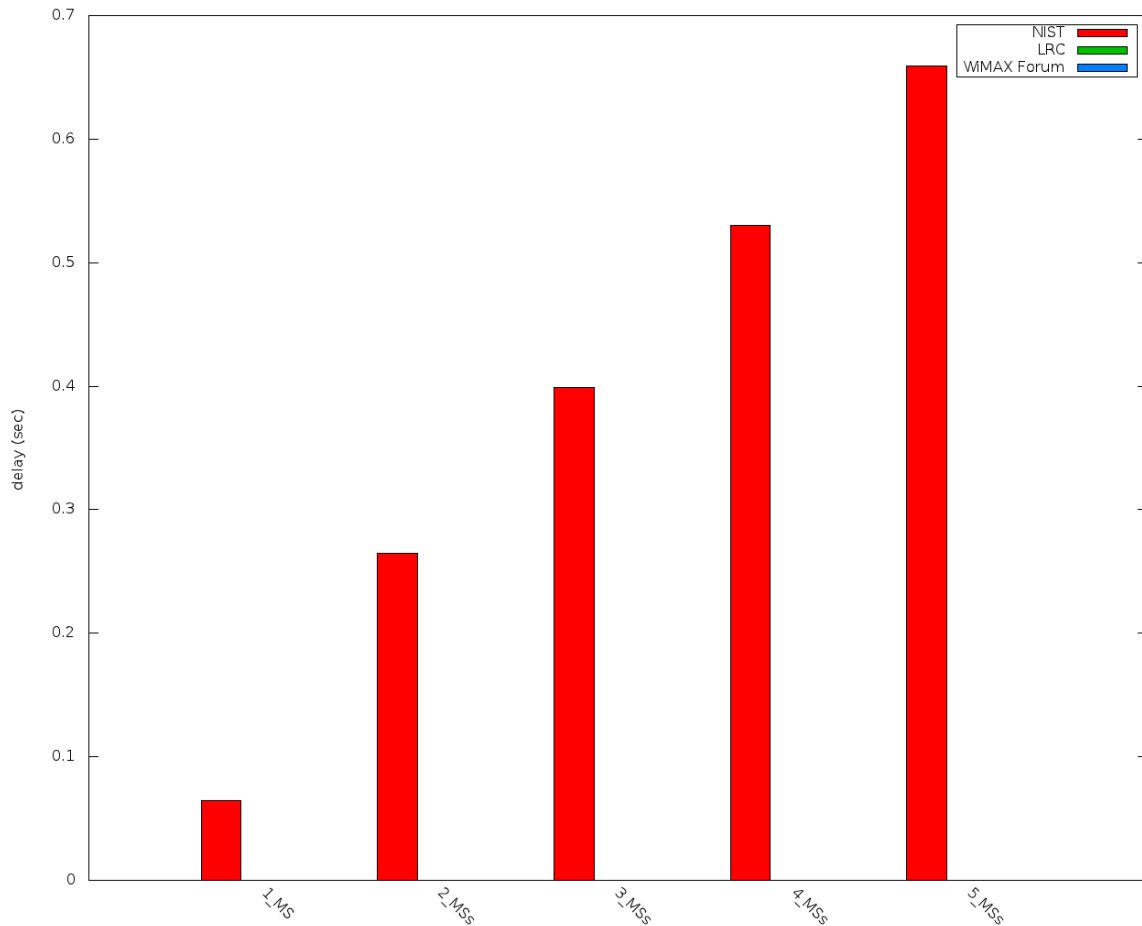
Εικόνα 91: Συγκριτικό μέσο throughput όλων των modules

Εν τέλη βλέπουμε ότι μεταξύ των αποτελεσμάτων των LRC, NIST και WiMAX Forum modules δεν υπάρχουν μεγάλες διαφορές, ενώ όταν αναφερόμαστε στα αποτελέσματα του ns2mesh80216 module πέραν των τεχνικών δυσκολιών που παρουσιάζει όσον αφορά την εξόρυξη πληροφοριών για την προσομοίωση βλέπουμε ότι υπάρχει και αρκετή διαφορά στα αποτελέσματα. Το γεγονός αυτό μπορεί να οφείλεται βέβαια και σε κάποιον διαφορετικό τρόπο υπολογισμού των μεταβλητών αυτών που αναλύονται από το ns2mesh80216 module σε σχέση με αυτόν που χρησιμοποιήθηκε για τα υπόλοιπα module.

### 8.2.1 Περαιτέρω διερεύνηση

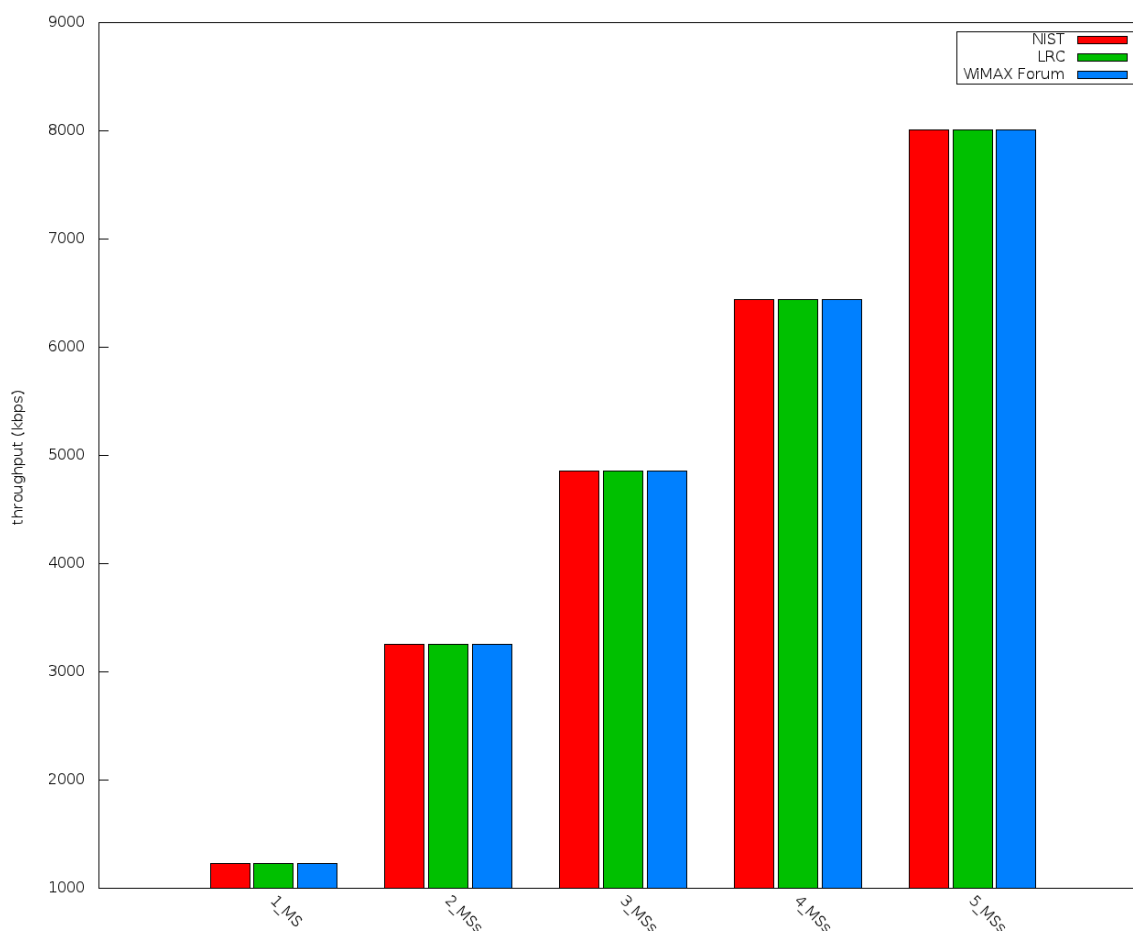
Εδώ θα δούμε κάποια γραφήματα για τα modules για τα οποία μπορούσαμε να προσομοιώσουμε ένα PMP δίκτυο, με πολλαπλούς σταθμούς. Πιο συγκεκριμένα στην εικόνα 92 βλέπουμε την μέση καθυστέρηση για τα σενάρια από 1 έως 5

σταθμούς. Όπως είχαμε δει αναλυτικά για κάθε module, έτσι και εδώ παρατηρούμε ότι το NIST module εμφανίζει μεγάλες καθυστερήσεις σε σχέση με τα άλλα δύο τα οποία εμφανίζουν μηδενική καθυστέρηση ακόμα και με 5 σταθμούς.



Εικόνα 92: Μέσο delay όλων των modules για προσομοιώσεις με διαφορετικό πλήθος σταθμών

Στην εικόνα 93 παρατηρούμε ότι το throughput είναι το ίδιο για όλα τα modules σε όλες τις περιπτώσεις του διαφορετικού πλήθους σταθμών.



Εικόνα 93: Μέσο throughput όλων των modules για προσομοιώσεις με διαφορετικό πλήθος σταθμών

### 8.3 Δυνατότητες και ελλείψεις κάθε module

Στον πίνακα 11 παρουσιάζονται αναλυτικά οι δυνατότητες και οι ελλείψεις του κάθε module ώστε να είναι εύκολο να αναγνωρίσει κάποιος ποιο από τα modules μπορεί να υποστηρίξει αυτό που θέλει να κάνει.

Πίνακας 11: Συγκριτικός πίνακας δυνατοτήτων των modules

Δυνατότητα	Modules			
	<u>NIST</u>	<u>WiMAX Forum</u>	<u>LRC</u>	<u>ns2mesh80216</u>
Mesh τοπολογία	O	O	O	N
PMP τοπολογία	N	N	N	O
Σταθμός BS	N	N	N	O

Σταθμός MS	N	N	N	O
Σταθμός MMS	O	N	O	O
Σταθμο Mesh	O	O	O	N
TDD	N	N	N	Δ
OFDM	N	O	O	N
OFDMA	O	N	O	O
ARQ	O	N	Δ	Δ
QoS	O	N	N	N
Fragmentation	N	N	Δ	Δ
Packing	O	N	Δ	Δ

Όπου N=ΝΑΙ, O=ΟΧΙ και Δ=ΔΕΝ ΥΠΑΡΧΕΙ ΠΛΗΡΟΦΟΡΙΑ

## 8.4 Συμπεράσματα

Μετά από την σειρά των σεναρίων που αναλύθηκε καθώς και των δυνατοτήτων του κάθε module καταλήγουμε στο συμπέρασμα ότι κάθε module έχει τις δυνατότητες και τις ελλείψεις του οι οποίες καθιστούν το καθένα κατάλληλο για διαφορετικές περιπτώσεις.

Σε περίπτωση που κάποιος θέλει να εξετάσει τοπολογίες που αφορούν mesh δίκτυα τότε το ns2mesh80216 module αποτελεί μονόδρομο μιας και όπως είδαμε και πιο πριν είναι το μόνο που έχει την δυνατότητα να υποστηρίξει αυτές τις περιπτώσεις. Θα πρέπει βέβαια να συμβιβαστεί κάποιος με του περιορισμούς του module καθώς και με την μεγάλη έλλειψη σε documentation που υπάρχει.

Σε περιπτώσεις που θέλουμε να εξετάσουμε PMP τοπολογίες, τότε έχουμε να διαλέξουμε μεταξύ τριών module.

Αρχικά ας πάρουμε τον παράγοντα του QoS παρατηρούμε ότι έχουμε να διαλέξουμε μεταξύ των modules των LRC και WiMAX Forum καθώς μόνο αυτά τα δύο υποστηρίζουν μηχανισμούς για QoS. Στο LRC module έναν από τους πέντε τρόπους χρονοπρογραμματισμού (ugs,BE,rtPS,nrtPS,ertPS) και να τους χρησιμοποιήσουμε πάνω σε κάθε κόμβο ενώ στο WiMAX Forum module δεν υπάρχει υποστήριξη για nrtPS και ertPS. Ένα υπέρ του module του WiMAX Forum σε αυτό το κομμάτι είναι η τρόπος με τον οποίο ορίζεται το QoS πάνω στους

κόμβους χρησιμοποιώντας μια εντολή TCL με τα κατάλληλα ορίσματα, κάτι που στο LRC module απαιτεί παραπάνω από μια εντολές.

Σαν δεύτερο κριτήριο θα εξετάσουμε την υποστήριξη των πρωτοκόλλων OFDM και OFDMA. Εδώ το LRC module υστερεί μιας και δεν έχει ασχοληθεί καθόλου με το επίπεδο PHY και χρησιμοποιεί το ασύρματο κανάλι που υπάρχει ήδη υλοποιημένο στον NS-2. Το NIST module είχε υλοποιήσει το OFDM ενώ το WiMAX Forum module εξέλιξε τον κώδικα του NIST και υλοποίησε το OFDMA χωρίς όμως να διατηρήσει την υποστήριξη του OFDM. Οπότε συμπεραίνουμε ότι αν θέλουμε να δοκιμάσουμε το OFDM αναγκαστικά θα χρησιμοποιήσουμε το NIST module, ενώ αν θέλουμε να χρησιμοποιήσουμε το OFDMA το WiMAX Forum module.

Ένα επιπλέον πλεονέκτημα του WiMAX Forum module είναι η υποστήριξη της πακετοποίησης καθώς και η οριζόντια καταχώρηση στα uplink και downlink κανάλια, πράγματα που δεν έχουν υλοποιηθεί σε κανένα άλλο module.

Από άποψης documentation, μπορούμε να πούμε ότι τα LRC και ns2mesh80216 modules είναι πολύ ελλιπή καθώς δεν υπάρχει κάποιο έγγραφο που να καθοδηγεί τον χρήστη στην υλοποίηση ενός σεναρίου. Το μόνο που υπάρχει είναι ένα παράδειγμα σεναρίου μέσα στο οποίο υπάρχει ένα τυπικό documentation σε ελάχιστα σημεία. Πράγμα που καθιστά τα δύο αυτά module δύσκολα στην εκμάθηση και χρήση. Από την άλλη πλευρά τα NIST και WiMAX Forum modules έχουν επαρκές documentation σε ηλεκτρονική μορφή το οποίο είναι αναλυτικό εξηγώντας τις δυνατότητές τους καθώς και την σύνταξη των εντολών που απαιτούνται για ένα TCL σενάριο.

Εν κατακλείδι, φτάνουμε στο συμπέρασμα ότι η τελική επιλογή ανάλογα με τις ανάγκες των σεναρίων μπορεί να γίνει μεταξύ δύο κυρίως module. Αν χρειαζόμαστε προσομοίωση mesh δικτύων αναγκαστικά θα χρησιμοποιήσουμε το ns2mesh80216 module, ενώ αν θέλουμε PMP δίκτυα τότε μέσα από τις διαθέσιμες επιλογές είναι προτιμότερο να χρησιμοποιήσουμε το WiMAX Forum module για τους εξής λόγους:

- υποστηρίζει ότι και τα άλλα δύο modules με μικρές ελλείψεις στο QoS

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

- έχει περισσότερες δυνατότητες από τα άλλα δύο modules
- η ανάπτυξη του συνεχίζεται ακόμα και σήμερα ενώ των άλλων δύο modules έχει σταματήσει εδώ και καιρό
- έχει επαρκές documentation και παραδείγματα τα οποία καθοδηγούν εύκολα κάποιον στην δημιουργία του δικού του σεναρίου προσομοίωσης

## 9 Βιβλιογραφία

1. Ahson S. & Ilyas M (2007). WiMAX: Technologies, Performance Analysis, and QoS. CRC Press Taylor & Francis Group.
2. Ahson S. and Ilyas M. (2008). WiMAX : Standards and Security. CRC Press Taylor & Francis Group.
3. Mobile WiMAX – Part I: A Technical Overview and Performance Evaluation, WiMAX Forum™, August 2006.
4. Mobile WiMAX – Part II: A Comparative Analysis, WiMAX Forum™, May 2006.
5. Teerawat I. & Ekram H. (2008). Introduction to Network Simulator NS2. Springer.
6. Ohrtman F. (2005). WiMAX Handbook Building 802.16 Wireless Networks. McGraw-Hill Communications
7. Drakos N. & Moore R. (1992-1999). The Network Simulator ns-2: Documentation. Retrieved from <http://www.isi.edu/nsnam/ns/ns-documentation.html>
8. Computer Networks Laboratory (2008). WiMAX module for the ns-2 simulator. Retrieved from [http://www.lrc.ic.unicamp.br/wimax\\_ns2/](http://www.lrc.ic.unicamp.br/wimax_ns2/)
9. Computer Networking Group (2008). Ns2mesh80216 module. Retrieved from <http://cng1.iet.unipi.it/wiki/index.php/Ns2mesh80216>
10. WiMAX Forum (2010). ns2-wimax-awg. Retrieved from <http://code.google.com/p/ns2-wimax-awg/>
11. Information Technology Laboratory (2007). Simulation Models for NS-2. Retrieved from: [http://www.nist.gov/itl/antd/emntg/ssm\\_tools.cfm](http://www.nist.gov/itl/antd/emntg/ssm_tools.cfm)
12. Wimax Made Simple. Retrieved from <http://wimax-made-simple.blogspot.com/>

13. NS-2 Trace Formats. Retrieved from Wiki :

[http://nslam.isi.edu/nslam/index.php/NS-2\\_Trace\\_Formats](http://nslam.isi.edu/nslam/index.php/NS-2_Trace_Formats)

14. Mohid P. Tahiliani. Trace analysis awk scripts. Retrieved from

<http://mohittahiliani.blogspot.com/>

15. Udin Harun. Visualize Trace File with GnuPlot. Retrieved from

<http://alkautsarpens.wordpress.com/2008/05/15/visualize-trace-file-data-with-gnuplot/>



## 10 Παράρτημα Α

Σε αυτό το παράρτημα υπάρχει το TCL σενάριο που χρησιμοποιήθηκε για το NIST module.

```
# Author: Giorgos Karanikas

# This is the script for the NIST WiMax module

#

# |----|
# | MNO | 1.0.1
# |----|

#

# (^)
# |

# |-----|
# | Base Station | 1.0.0
# |-----|

# |
# |

# |-----|
# | Sink Node | 0.0.0
# |-----|

#

# Check input arguments

if {$argc != 0} {
    puts ""
    puts "Wrong number of arguments. No arguments in this topology"
```

```
puts ""
exit (1)
}

# =====
#   Declare global variables
# =====

set max_mobile_nodes 1 ; # max number of mobile nodes
set packet_size 1500 ; # packet size in bytes at CBR application
set output_dir . ; # the output directory of the topology
set gap_size 1 ; # compute gap size between packets
set traffic_start 100 ; # define when the traffic will start
set traffic_stop 500 ; # define when the traffic will stop
set simulation_stop 510 ; # define when the simulation will stop

# =====
#   Declare debug variables
# =====

Mac/802_16 set debug_ 1 ; # print out debug data
Mac/802_16 set rtg_ 20 ; # number of PS to switch from receiving to transmitting
state [default = 10]
Mac/802_16 set ttg_ 20 ; # number of PS to switch from transmitting to receiving
state [default = 10]
Mac/802_16 set frame_duration_ 0.004 ; # frame duration (s)
[default = 0.004]
Mac/802_16 set client_timeout_ 110 ; # value for detecting out of range client
[default = 0.5]
```

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

```
Phy/WirelessPhy/OFDM set g_ 0.25 ; # cyclic prefix [Valid values are 0.25 (1/4), 0.125 (1/8),  
0.0625 (1/16), 0.03125 (1/32)]
```

```
# =====
```

```
# Declare topology variables
```

```
# =====
```

```
# Define coverage area for base station: 20m coverage
```

```
Phy/WirelessPhy set Pt_ 0.025 ; # transmitted signal power (W)
```

```
Phy/WirelessPhy set RXThresh_ 2.025e-12 ; # receive power threshold  
(W) [500m radius]
```

```
Phy/WirelessPhy set CStresh_ [expr 0.9*[Phy/WirelessPhy set RXThresh_]] ; # carrier  
sense threshold (W)
```

```
# Define Parameters for wireless nodes
```

```
set opt(chan) Channel/WirelessChannel ;# channel type
```

```
set opt(prop) Propagation/TwoRayGround ;# radio-propagation model
```

```
set opt(netif) Phy/WirelessPhy/OFDM ;# network interface type
```

```
set opt(mac) Mac/802_16/BS ;# MAC type
```

```
set opt(ifq) Queue/DropTail/PriQueue ;# interface queue type
```

```
set opt(ll) LL ;# link layer type
```

```
set opt(ant) Antenna/OmniAntenna ;# antenna model
```

```
set opt(ifqlen) 50 ;# max packet in ifq
```

```
set opt(adhocRouting) DSDV ;# routing protocol
```

```
set opt(x) 1100 ;# X dimension of the topography
```

```
set opt(y) 1100 ;# Y dimension of the topography
```

```
Mac/802_11 set basicRate_ 11Mb
```

```
Mac/802_11 set dataRate_ 11Mb
```

```
Mac/802_11 set bandwidth_ 11Mb
```

```
# =====
```

```
# Declare simulator variables
```

```
# =====
```

```
# Create the simulator object
```

```
set ns [new Simulator]
```

```
$ns use-newtrace
```

```
# Create the topography
```

```
set topo [new Topography]
```

```
$topo load_flatgrid $opt(x) $opt(y)
```

```
# Now we open a file for writing that is going to be used for the nam trace data.
```

```
set nf [open $output_dir/out.res w]
```

```
$ns trace-all $nf
```

```
# 'finish' procedure that closes the trace file and starts nam.
```

```
proc finish {} {
```

```
    global ns nf output_dir max_mobile_nodes
```

```
    $ns flush-trace
```

```
    close $nf
```

```
    #exec nam out.nam &
```

```
    exit 0
}

# =====

#      Create the scenario

# =====

# set up for hierarchical routing (needed for routing over a basestation)

$ns node-config -addressType hierarchical           ; # declare the use of hierarchical
routing

AddrParams set domain_num_ 2                       ; # domain number

lappend cluster_num 1 1                            ; # cluster number for each domain

AddrParams set cluster_num_ $cluster_num           ; # set the cluster number

lappend eilastlevel 1 [expr ($max_mobile_nodes+1)] ; # number of nodes for each
cluster (1 for sink and one for mobile nodes + base station)

AddrParams set nodes_num_ $eilastlevel             ; # set the number of nodes for
each cluster

# Create God

create-god [expr ($max_mobile_nodes + 2)]          ; # maximum mobile nodes + 2
(base station and sink node)

# Create the Sink Node in first addressing space.

set sinkNode [$ns node 0.0.0]                      ; # create the sink node with address
0.0.0

# provide coordinations for sink node

$sinkNode set X_ 50.0
$sinkNode set Y_ 50.0
$sinkNode set Z_ 0.0
```

*#creates the Access Point (Base station)*

```
$ns node-config -adhocRouting $opt(adhocRouting) \
```

```
  -llType $opt(ll) \
```

```
  -macType Mac/802_16/BS \
```

```
  -ifqType $opt(ifq) \
```

```
  -ifqLen $opt(ifqlen) \
```

```
  -antType $opt(ant) \
```

```
  -propType $opt(prop) \
```

```
  -phyType $opt(netif) \
```

```
  -channel [new $opt(chan)] \
```

```
  -topoInstance $topo \
```

```
  -wiredRouting ON \
```

```
  -agentTrace ON \
```

```
  -routerTrace ON \
```

```
  -macTrace ON \
```

```
  -movementTrace OFF
```

```
set bstation [$ns node 1.0.0] ; # create the base node with address 1.0.0
```

```
$bstation random-motion 0 ; # disable random motion
```

*# provide coordinations for base node*

```
$bstation set X_ 550.0
```

```
$bstation set Y_ 550.0
```

```
$bstation set Z_ 0.0
```

```
[$bstation set mac_(0)] set-channel 0
```

*# creation of the mobile nodes*

```
$ns node-config -macType Mac/802_16/SS \  
    -wiredRouting OFF \  
    -macTrace ON  
  
# Mobile nodes cannot do routing  
  
for {set i 0} {$i < $max_mobile_nodes} {incr i} {  
    set wl_node_($i) [$ns node 1.0.[expr $i + 1]]           ; # create the node with given  
    @.  
    $wl_node_($i) random-motion 0                           ; # disable random motion  
    $wl_node_($i) base-station [AddrParams addr2id [$bstation node-addr]] ; # attach mn to  
    basestation  
  
    #compute position of the node  
    $wl_node_($i) set X_ [expr 340.0]  
    $wl_node_($i) set Y_ 550.0  
    $wl_node_($i) set Z_ 0.0  
  
    $ns at 0 "$wl_node_($i) setdest 1060.0 550.0 1.0"      ; # set the destination of the  
    node  
  
    [$wl_node_($i) set mac_(0)] set-channel 0  
    [$wl_node_($i) set mac_(0)] set-diuc 7                 ; # Change the node profile here  
    [7=64QAM_3_4]  
  
    #create source traffic  
    #Create a UDP agent and attach it to node n0  
    set udp_($i) [new Agent/UDP]                            ; # create the UDP agent  
    $udp_($i) set packetSize_ 1500                         ; # set the packetSize to 1500  
    $ns attach-agent $wl_node_($i) $udp_($i)               ; # attach the UDP agent to the
```

## Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

*mobile node*

```
# Create a CBR traffic source and attach it to udp0

set cbr_($i) [new Application/Traffic/CBR] ; # create the CBR traffic source

$cbr_($i) set packetSize_ $packet_size ; # set the packetsize to 1500
[the value of the global variable]

$cbr_($i) set interval_ $gap_size ; # set the gap size to 1 [the
value of the global variable]

$cbr_($i) attach-agent $udp_($i) ; # attach te CBR traffic source to
the UDP agent

#create an sink into the sink node

# Create the Null agent to sink traffic

set null_($i) [new Agent/Null] ; # create the NULL agent

$ns attach-agent $sinkNode $null_($i) ; # attach the NULL agent to the
sink node

# Attach the 2 agents

$ns connect $udp_($i) $null_($i) ; # connect the UDP and the NULL
agent
}

# create the link between the SINK NODE and BASE STATION

$ns duplex-link $sinkNode $bstation 100Mb 1ms DropTail ; # create a duplex link with
100Mb bandwidth, 1ms delay and DropTail queue

# Traffic scenario: if alla the nodes start taliking at the same tim, we may see packe tloss due to
bandwidth request collision
```



Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

```
set diff 0.0

for {set i 0} {$i < $max_mobile_nodes} {incr i} {

    $ns at [expr $traffic_start + $i * $diff] "$cbr_($i) start" ; # start the traffic an mobile node i
    after i * diff seconds from traffic start

    $ns at [expr $traffic_stop + $i * $diff] "$cbr_($i) stop" ; # stop the traffic an mobile node i
    after i * diff seconds from traffic stop

}

# =====

#      Run the simulation

# =====

# Execute the 'finish' procedure after 5.0 seconds of simulation time.

$ns at $simulation_stop "finish" ; # when simulation stops run finish

# Run the simulation

$ns run
```

## 11 Παράρτημα Β

Σε αυτό το παράρτημα υπάρχει το TCL σενάριο που χρησιμοποιήθηκε για το WiMAX Forum module.

```
# Author: Giorgos Karanikas

# This is the script for the WiMAX Forum WiMax module

#

# |----|
# | MNO |   1.0.1
# |----|

#

# (^)
# |

# |-----|
# | Base Station | 1.0.0
# |-----|

# |
# |

# |-----|
# | Sink Node |   0.0.0
# |-----|

#

# Check input arguments

if {$argc != 0} {
    puts ""
    puts "Wrong number of arguments. No arguments in this topology"
```

```
puts ""
exit (1)
}

#
=====
=====

#   Declare global variables
#
=====
=====

set max_mobile_nodes    1      ; # max number of mobile nodes
set packet_size        1500    ; # packet size in bytes at CBR application
set output_dir          .       ; # the output directory of the topology
set gap_size            0.005   ; # compute gap size between packets
set traffic_start       0.1     ; # define when the traffic will start
set traffic_stop        59.9    ; # define when the traffic will stop
set simulation_stop     60.0    ; # define when the simulation will stop

#
=====
=====

#   Declare debug variables
#
=====
=====

Mac/802_16 set debug_    1      ; # print out debug data
Mac/802_16 set rtg_     20     ; # number of PS to switch from receiving to transmitting state
```

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

```
[default = 10]
Mac/802_16 set ttg_ 20 ; # number of PS to switch from transmitting to receiving state
[default = 10]
Mac/802_16 set frame_duration_ 0.004 ; # frame duration (s) [default =
0.004]
Mac/802_16 set client_timeout_ 110 ; # value for detecting out of range client
[default = 0.5]
Mac/802_16 set ITU_PDP_ 2
Phy/WirelessPhy/OFDMA set g_ 0.25 ; # cyclic prefix [Valid values are 0.25 (1/4), 0.125
(1/8), 0.0625 (1/16), 0.03125 (1/32)]

# =====
# Declare topology variables
# =====

# Define coverage area for base station: 20m coverage
Phy/WirelessPhy set Pt_ 0.025 ; # transmitted signal power (W)
Phy/WirelessPhy set RXThresh_ 2.025e-12 ; # receive power threshold (W) [500m
radius]
Phy/WirelessPhy set CSThresh_ [expr 0.9*[Phy/WirelessPhy set RXThresh_]] ; # carrier
sense threshold (W)
Phy/WirelessPhy set OFDMA_ 1

# Define Parameters for wireless nodes
set opt(chan) Channel/WirelessChannel ;# channel type
set opt(prop) Propagation/OFDMA ;# radio-propagation model
set opt(netif) Phy/WirelessPhy/OFDMA ;# network interface type
set opt(mac) Mac/802_16/BS ;# MAC type
set opt(ifq) Queue/DropTail/PriQueue ;# interface queue type
```

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

```
set opt(ll)          LL                ;# link layer type
set opt(ant)         Antenna/OmniAntenna ;# antenna model
set opt(ifqlen)     100                ;# max packet in ifq
set opt(adhocRouting) DSDV             ;# routing protocol
```

```
set opt(x)          1100               ;# X dimension of the topography
set opt(y)          1100               ;# Y dimension of the topography
```

```
Mac/802_11 set basicRate_ 11Mb
Mac/802_11 set dataRate_  11Mb
Mac/802_11 set bandwidth_ 11Mb
```

```
#
```

```
=====
=====
```

```
#      Declare simulator variables
```

```
#
```

```
=====
=====
```

```
# Create the simulator object
```

```
set ns [new Simulator]
```

```
#$ns use-newtrace
```

```
# Create the topography
```

```
set topo [new Topography]
```

```
$topo load_flatgrid $opt(x) $opt(y)
```

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

*# Now we open a file for writing that is going to be used for the nam trace data.*

```
set nf [open $output_dir/WIMAX_FORUM.res w]
```

```
$ns trace-all $nf
```

```
set namtr [open $output_dir/WIMAX_FORUM.nam w]
```

```
$ns namtrace-all $namtr
```

*# 'finish' procedure that closes the trace file and starts nam.*

```
proc finish {} {
```

```
    global ns nf output_dir max_mobile_nodes namtr
```

```
    $ns flush-trace
```

```
    close $nf
```

```
    close $namtr
```

```
    exec nam WIMAX_FORUM.nam &
```

```
    exit 0
```

```
}
```

```
#
```

```
=====
```

```
#      Create the scenario
```

```
#
```

```
=====
```

*# set up for hierarchical routing (needed for routing over a basestation)*

```
$ns node-config -addressType hierarchical ; # declare the use of hierarchical routing
```

```
AddrParams set domain_num_2 ; # domain number
```

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

```
lappend cluster_num 1 1 ; # cluster number for each domain

AddrParams set cluster_num_ $cluster_num ; # set the cluster number

lappend eilastlevel 1 [expr ($max_mobile_nodes+1)] ; # number of nodes for each cluster (1 for
sink and one for mobile nodes + base station

AddrParams set nodes_num_ $eilastlevel ; # set the number of nodes for each cluster

# Create God

create-god [expr ($max_mobile_nodes + 2)] ; # maximum mobile nodes + 2 (base station and
sink node)

# Create the Sink Node in first addressing space.

set sinkNode [$ns node 0.0.0] ; # create the sink node with address 0.0.0

# provide coordinations for sink node

$sinkNode set X_ 50.0
$sinkNode set Y_ 50.0
$sinkNode set Z_ 0.0

#creates the Access Point (Base station)

$ns node-config -adhocRouting $opt(adhocRouting) \
    -llType $opt(ll) \
        -macType Mac/802_16/BS \
        -ifqType $opt(ifq) \
        -ifqLen $opt(ifqlen) \
        -antType $opt(ant) \
        -propType $opt(prop) \
        -phyType $opt(netif) \
        -channel [new $opt(chan)] \
        -topoInstance $topo \
```

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

```
-wiredRouting ON \  
-agentTrace ON \  
-routerTrace ON \  
-macTrace ON \  
-movementTrace OFF
```

```
#setup channel model
```

```
set prop_inst [$ns set propInstance_]
```

```
$prop_inst ITU_PDP PED_A
```

```
#puts "after set pPDP"
```

```
set bstation [$ns node 1.0.0] ; # create the base node with address 1.0.0
```

```
bstation random-motion 0 ; # disable random motion
```

```
# provide coordinations for base node
```

```
bstation set X_ 550.0
```

```
bstation set Y_ 550.0
```

```
bstation set Z_ 0.0
```

```
[$bstation set mac_(0)] set-channel 0
```

```
# creation of the mobile nodes
```

```
$ns node-config -macType Mac/802_16/SS \  
    -wiredRouting OFF \  
    -macTrace ON
```

```
# Mobile nodes cannot do routing
```

```
for {set i 0} {$i < $max_mobile_nodes} {incr i} {
```

```
    set wl_node_($i) [$ns node 1.0.[expr $i + 1]] ; # create the node with given @.
```



Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

```
$wl_node_($i) random-motion 0 ; # disable random motion

$wl_node_($i) base-station [AddrParams addr2id [$bstation node-addr]] ; # attach mn to
basestation

#compute position of the node

$wl_node_($i) set X_ [expr 240.0+50*$i] ; #340
$wl_node_($i) set Y_ 500.0 ; #550
$wl_node_($i) set Z_ 0.0

$ns at 0 "$wl_node_($i) setdest 1060 500.0 30.0" ; # set the destination of the node
$ns at 40 "$wl_node_($i) setdest 200 500.0 50.0" ; # set the destination of the node

[$wl_node_($i) set mac_(0)] set-channel 0
[$wl_node_($i) set mac_(0)] set-diuc 7 ; # Change the node profile here
[7=64QAM_3_4]

# ===== CREATE SOURCE TRAFFIC =====

#Create a UDP agent and attach it to node n0

set udp_($i) [new Agent/UDP] ; # create the UDP agent
$udp_($i) set packetSize_ 1500 ; # set the packetSize to 1500
$ns attach-agent $wl_node_($i) $udp_($i) ; # attach the UDP agent to the mobile
node

# Create a CBR traffic source and attach it to udp0

set cbr_($i) [new Application/Traffic/CBR] ; # create the CBR traffic source
$cbr_($i) set packetSize_ $packet_size ; # set the packetSize to 1500 [the value of
the global variable]
$cbr_($i) set interval_ $gap_size ; # set the gap size to 1 [the value of the
```

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

*global variable]*

```
$cbr_($i) attach-agent $udp_($i) ; # attach te CBR traffic source to the UDP agent

# ----- CREATE A SINK INTO THE SINK NODE -----

# Create the Null agent to sink traffic

set null_($i) [new Agent/Null] ; # create the NULL agent

$ns attach-agent $sinkNode $null_($i) ; # attach the NULL agent to the sink node

# Attach the 2 agents

$ns connect $udp_($i) $null_($i) ; # connect the UDP and the NULL agent
}

# create the link between the SINK NODE and BASE STATION

$ns duplex-link $sinkNode $bstation 100Mb 1ms DropTail ; # create a duplex link with
100Mb bandwidth, 1ms delay and DropTail queue

# Traffic scenario: if alla the nodes start taliking at the same tim, we may see packe tloss due to
bandwidth request collision

set diff 0.0

for {set i 0} {$i < $max_mobile_nodes} {incr i} {

    $ns at [expr $traffic_start + $i * $diff] "$cbr_($i) start" ; # start the traffic an mobile node i
after i * diff seconds from traffic start

    $ns at [expr $traffic_stop + $i * $diff] "$cbr_($i) stop" ; # stop the traffic an mobile node i
after i * diff seconds from traffic stop
}

#
=====
```

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

```
=====
#      Run the simulation
#
=====
=====

# Execute the 'finish' procedure after 5.0 seconds of simulation time.
$ns at $simulation_stop "finish"          ; # when simulation stops run finish

# Run the simulation
$ns run
```

## 12 Παράρτημα Γ

Σε αυτό το παράρτημα υπάρχει το TCL σενάριο που χρησιμοποιήθηκε για το LRC module.

```
# Author: Giorgos Karanikas

# This is the script for the LRC WiMax module

#

# |----|

# | MNO |   1.0.1

# |----|

#

# (^)

# |

# |-----|

# | Base Station | 1.0.0

# |-----|

# |

# |

# |-----|

# | Sink Node |   0.0.0

# |-----|

#

# Check input arguments

if {$argc != 0} {

    puts ""

    puts "Wrong number of arguments. No arguments in this topology"
```

```
puts ""
exit (1)
}

#
=====
=====

#   Declare global variables
#
=====
=====

set max_mobile_nodes 1      ; # max number of mobile nodes
set packet_size      1500   ; # packet size in bytes at CBR application
set output_dir       .      ; # the output directory of the topology
set gap_size         0.005   ; # compute gap size between packets
set traffic_start    0.1     ; # define when the traffic will start
set traffic_stop     59.9    ; # define when the traffic will stop
set simulation_stop  60      ; # define when the simulation will stop

#
=====
=====

#   Declare debug variables
#
=====
=====

#Mac/802_16 set debug_ 1      ; # print out debug data
#Mac/802_16 set rtg_ 20      ; # number of PS to switch from receiving to transmitting
```

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

```
state [default = 10]

#Mac/802_16 set ttg_ 20 ; # number of PS to switch from transmitting to receiving
state [default = 10]

#Mac/802_16 set frame_duration_ 0.004 ; # frame duration (s) [default =
0.004]

#Mac/802_16 set client_timeout_ 110 ; # value for detecting out of range client
[default = 0.5]

#Phy/WirelessPhy/OFDM set g_ 0.25 ; # cyclic prefix [Valid values are 0.25 (1/4), 0.125
(1/8), 0.0625 (1/16), 0.03125 (1/32)]

# =====
# Declare topology variables
# =====

# Define coverage area for base station: 20m coverage

Phy/WirelessPhy set Pt_ 0.025 ; # transmitted signal power (W)

Phy/WirelessPhy set RXThresh_ 2.025e-12 ; # receive power threshold (W) [500m
radius]

Phy/WirelessPhy set CStresh_ [expr 0.9*[Phy/WirelessPhy set RXThresh_]] ; # carrier
sense threshold (W)

# Define Parameters for wireless nodes

set opt(chan) Channel/WirelessChannel ;# channel type
set opt(prop) Propagation/TwoRayGround ;# radio-propagation model
set opt(netif) Phy/WirelessPhy ;# network interface type
set opt(mac) Mac/802_16BS ;# MAC type
set opt(ifq) Queue/DropTail/PriQueue ;# interface queue type
set opt(ll) LL ;# link layer type
set opt(ant) Antenna/OmniAntenna ;# antenna model
```

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

```
set opt(ifqlen)      100           ;# max packet in ifq
set opt(adhocRouting) DSDV         ;# routing protocol

set opt(x)          1100          ;# X dimension of the topography
set opt(y)          1100          ;# Y dimension of the topography

Mac/802_11 set basicRate_ 11Mb
Mac/802_11 set dataRate_  11Mb
Mac/802_11 set bandwidth_ 11Mb
Mac/802_16BS set window_  2

set BW              5e+6

#
=====
=====

#      Declare simulator variables
#
=====
=====

# Create the simulator object
set ns [new Simulator]
#$ns use-newtrace

# Create the topography
set topo [new Topography]
$topo load_flatgrid $opt(x) $opt(y)
```

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

*# Now we open a file for writing that is going to be used for the nam trace data.*

```
set nf [open $output_dir/LRC.res w]
```

```
$ns trace-all $nf
```

```
set namtr [open $output_dir/LRC.nam w]
```

```
$ns namtrace-all $namtr
```

*# 'finish' procedure that closes the trace file and starts nam.*

```
proc finish {} {
```

```
    global ns nf output_dir max_mobile_nodes namtr
```

```
    $ns flush-trace
```

```
    close $nf
```

```
    close $namtr
```

```
    exec nam LRC.nam &
```

```
    exit 0
```

```
}
```

```
#
```

```
=====
```

```
#      Create the scenario
```

```
#
```

```
=====
```

*# set up for hierarchical routing (needed for routing over a basestation)*

```
$ns node-config -addressType hierarchical ; # declare the use of hierarchical routing
```

```
AddrParams set domain_num_2 ; # domain number
```



## Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

```
lappend cluster_num 1 1 ; # cluster number for each domain

AddrParams set cluster_num_ $cluster_num ; # set the cluster number

lappend eilastlevel 1 [expr ($max_mobile_nodes+1)] ; # number of nodes for each cluster (1 for
sink and one for mobile nodes + base station

AddrParams set nodes_num_ $eilastlevel ; # set the number of nodes for each cluster

# Create God

create-god [expr ($max_mobile_nodes + 2)] ; # maximum mobile nodes + 2 (base station and
sink node)

# Create the Sink Node in first addressing space.

set sinkNode [$ns node 0.0.0] ; # create the sink node with address 0.0.0

# provide coordinations for sink node

$sinkNode set X_ 50.0
$sinkNode set Y_ 50.0
$sinkNode set Z_ 0.0

#creates the Access Point (Base station)

$ns node-config -adhocRouting $opt(adhocRouting) \
    -llType $opt(ll) \
        -macType Mac/802_16BS \
        -ifqType $opt(ifq) \
        -ifqLen $opt(ifqlen) \
        -antType $opt(ant) \
        -propType $opt(prop) \
        -phyType $opt(netif) \
        -channel [new $opt(chan)] \
        -topoInstance $topo \
```

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

```
-wiredRouting ON \  
-agentTrace ON \  
-routerTrace ON \  
-macTrace ON \  
-movementTrace OFF
```

```
set bstation [$ns node 1.0.0] ; # create the base node with address 1.0.0  
$bstation random-motion 0 ; # disable random motion  
  
# provide coordinations for base node  
$bstation set X_ 550.0  
$bstation set Y_ 550.0  
$bstation set Z_ 0.0  
#[$bstation set mac_(0)] set-channel 0  
  
# creation of the mobile nodes  
$ns node-config -macType Mac/802_16SS \  
-wiredRouting OFF \  
-macTrace ON  
  
# Mobile nodes cannot do routing  
  
for {set i 0} {$i < $max_mobile_nodes} {incr i} {  
    set wl_node_($i) [$ns node 1.0.[expr $i + 1]] ; # create the node with given @.  
    $wl_node_($i) random-motion 0 ; # disable random motion  
    $wl_node_($i) base-station [AddrParams addr2id [$bstation node-addr]] ; # attach mn to  
    basestation  
  
    #compute position of the node
```

Πτυχιακή του φοιτητή Καρανίκας Γεωργίου

```
$wl_node_($i) set X_ [expr 240.0+50*$i] ; #340
$wl_node_($i) set Y_ 500.0 ;#550
$wl_node_($i) set Z_ 0.0
$ns at 0 "$wl_node_($i) setdest 1060 500.0 30.0" ; # set the destination of the node
$ns at 40 "$wl_node_($i) setdest 200 500.0 50.0" ; # set the destination of the node

#[$wl_node_($i) set mac_(0)] set-channel 0
#[$wl_node_($i) set mac_(0)] set-diuc 7 ; # Change the node profile here
[7=64QAM_3_4]

# ----- CREATE SOURCE TRAFFIC -----

#Create a UDP agent and attach it to node n0
set udp_($i) [new Agent/UDP] ; # create the UDP agent
$udp_($i) set packetSize_ 1500 ; # set the packetSize to 1500
$ns attach-agent $wl_node_($i) $udp_($i) ; # attach the UDP agent to the mobile
node

# Create a CBR traffic source and attach it to udp0
set cbr_($i) [new Application/Traffic/CBR] ; # create the CBR traffic source
$cbr_($i) set packetSize_ $packet_size ; # set the packetSize to 1500 [the value of
the global variable]
$cbr_($i) set interval_ $gap_size ; # set the gap size to 1 [the value of the
global variable]
$cbr_($i) attach-agent $udp_($i) ; # attach te CBR traffic source to the UDP agent

# ----- CREATE A SINK INTO THE SINK NODE -----
```

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

```
# Create the Null agent to sink traffic

set null_($i) [new Agent/Null] ; # create the NULL agent

$ns attach-agent $sinkNode $null_($i) ; # attach the NULL agent to the sink node

# Attach the 2 agents

$ns connect $udp_($i) $null_($i) ; # connect the UDP and the NULL agent
}

# create the link between the SINK NODE and BASE STATION

$ns duplex-link $sinkNode $bstation 100Mb 1ms DropTail ; # create a duplex link with
100Mb bandwidth, 1ms delay and DropTail queue

# Traffic scenario: if alla the nodes start taliking at the same tim, we may see packe tloss due to
bandwidth request collision

set diff 2.0

for {set i 0} {$i < $max_mobile_nodes} {incr i} {
    $ns at [expr $traffic_start + $i * $diff] "$cbr_($i) start" ; # start the traffic an mobile node i
after i * diff seconds from traffic start

    $ns at [expr $traffic_stop + $i * $diff] "$cbr_($i) stop" ; # stop the traffic an mobile node i
after i * diff seconds from traffic stop
}

#
=====
=====
# Run the simulation
#
=====
=====
```

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

*# Execute the 'finish' procedure after 5.0 seconds of simulation time.*

```
$ns at $simulation_stop "finish" ; # when simulation stops run finish
```

*# Run the simulation*

```
$ns run
```

## **13 Παράρτημα Δ**

Σε αυτό το παράρτημα υπάρχει το TCL σενάριο που χρησιμοποιήθηκε για το ns2mesh80216 module.

Αρχείο mesh.tcl:

```
#  
  
# Copyright (C) 2007 Dip. Ing. dell'Informazione, University of Pisa, Italy  
  
# http://info.iet.unipi.it/~cng/ns2mesh80216/  
  
#  
  
# This program is free software; you can redistribute it and/or modify  
# it under the terms of the GNU General Public License as published by  
# the Free Software Foundation; either version 2 of the License, or  
# (at your option) any later version.  
  
#  
  
# This program is distributed in the hope that it will be useful,  
# but WITHOUT ANY WARRANTY; without even the implied warranty of  
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
# GNU General Public License for more details.  
  
#  
  
# You should have received a copy of the GNU General Public License  
# along with this program; if not, write to the Free Software  
# Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA, USA  
  
#  
  
# test 802.16 mesh Tcl scenario  
  
# author: C.Cicconetti <claudio.cicconetti@iet.unipi.it>  
  
# website: http://info.iet.unipi.it/~cng/ns2mesh80216/
```

```
#  
#  
# A simple string topology with four nodes is created:  
#  
# 0 -- 1 -- 2 -- 3  
#  
# Three traffic flows are established:  
# - a VoIP flow with one aggregated source from 0 to 3, with priority 1  
# - a VoIP flow with two aggregated sources from 1 to 3, with priority 1  
# - a VoD flow with one aggregated source from 2 to 3, with priority 0  
#  
#  
# load configuration file with default options  
#  
source header.msh  
source traffic.msh  
source metrics.msh  
  
#####  
###  
# SCENARIO CONFIGURATION  
#####  
###  
  
#  
# simulation environment
```

```

#
set opt(run) 1 ;# replic ID
set opt(duration) 60.0 ;# run duration, in seconds
set opt(warm) 2.0 ;# run duration, in seconds
set opt(out) "out" ;# statistics output file
set opt(debug) "" ;# debug configuration file, "" = no debug
set opt(startdebug) 0.0 ;# start time of debug output

#
# topology
#
set opt(topology) "chain" ;# see TOPOLOGIES
set opt(n) 2 ;# meaningful with: chain, ring, grid,
                ;# multiring, star, clique, bintree
                ;#
trinagular (see TOPOLOGIES)
set opt(branches) 1 ;# meaningful with: star, multiring
                ;# (see TOPOLOGIES)
set opt(random-nodeid) "on" ;# mix up NodeIDs randomly

#
# PHY
#
set opt(channel) 1 ;# number of channels
set opt(radio) 1 ;# number of radios
set opt(chan-data-per) 0 ;# channel data PDU error rate
set opt(chan-ctrl-per) 0 ;# channel control PDU error rate

```



```
set opt(propagation) 4    ;# physical propagation, in us
set opt(sym-duration) 20  ;# OFDM symbol duration, in us
set opt(sym-perframe) 500 ;# number of OFDM symbols per frame
set opt(bandwidth) "10"  ;# in MHz, if set overrides the OFDM symbol
                        ;# duration and the number of OFDM symbols
                        ;# per frame (ie. the frame duration)
set opt(cyclic-prefix) "1/4" ;# cyclic prefix, only used if bandwidth is set
set opt(control) 4       ;# number of MSH-DSCH opportunities per frame
set opt(cfg-interval) 16  ;# number of MSH-DSCH frames between two
                        ;# consecutive MSH-NCFG frames
set opt(msh-dsch-avg-bad) -1 ;# consecutive bad MSH-DSCH messages
set opt(msh-dsch-avg-good) -1 ;# consecutive good MSH-DSCH messages

#
# MAC
#
set opt(allocation) "contiguous" ;# MSH-DSCH allocation type
set opt(hest-curr) .1             ;# weight for H's estimations (current value)
set opt(hest-past) .9            ;# weight for H's estimations (past values)

#
# bandwidth manager
#
set opt(bwmanager) "fair-rr" ;# bandwidth manager type
set opt(availabilities) "on" ;# RR, FairRR bwmanagers
set opt(regrant) "on" ;# RR, FairRR bwmanagers
set opt(regrant-offset) 1 ;# RR, FairRR bwmanagers, in frames
```

```

set opt(regrant-duration) 1      ;# RR, FairRR bwmanagers, in handshakes
set opt(grant-fairness) "on"    ;# FairRR bwmanager = {on, off}
set opt(regrant-same-horizon) "off" ;# FairRR bwmanager = {on, off}
set opt(regrant-fairness) "on"  ;# FairRR bwmanager = {on, off}
set opt(request-fairness) "on"  ;# FairRR bwmanager = {on, off}
set opt(bwm-round-duration) 21312 ;# FairRR bwmanager, in bytes
set opt(weight-timeout) 120     ;# FairRR bwmanager, in sec
set opt(max-deficit) 0         ;# FairRR bwmanager, in bytes
set opt(max-backlog) 0         ;# FairRR bwmanager, in bytes
set opt(weight-flow) "on"     ;# FairRR bwmanager = {on, off}
set opt(grant-rnd-channel) "on" ;# FairRR bwmanager = {on, off}
set opt(dd-timeout) "50"      ;# FairRR bwmanager, in MSH-DSCH opps
set opt(min-grant) "1"        ;# FairRR bwmanager, in OFDM symbols

set opt(prio-weight) "1 2 4" ;# priority weights, used by both the
                               ;# FairRR bwmanager and the scheduler

#
# distributed election coordinator
#
set opt(coordinator) "standard" ;# coordinator type
set opt(coordmode) "xmtunaware" ;# coordinator mode
set opt(holdoff-dsch-def) 0      ;# holdoff exponent for MSH-DSCH msgs
set opt(holdoff-ncfg-def) 0      ;# holdoff exponent for MSH-NCFG msgs
set opt(holdoff-dsch) {}        ;# used to set different holdoffs
set opt(holdoff-ncfg) {}        ;# used to set different holdoffs
set opt(control-period) 4       ;# dummy coordinator, number of frames

```

```
set opt(max-advertised) -1      ;# maximum number of advertised neighbors

#

# forwarding

#

set opt(forwarding)      "spf"  ;# forwarding type

set opt(maxnumhops)      20     ;# maximum path length

set opt(profileupdateperiod) 4    ;# number of frames to update profiles

set opt(samplingperiod)  4      ;# sampling nodes'load period (frames)

set opt(chi)              0.5    ;# weight of grants for links' cost

set opt(beta)             0.5    ;# weight of new samples for nodes' load

#

# packet scheduler

#

set opt(scheduler)        "fair-rr" ;# packet scheduler type

set opt(buffer)           1000000 ;# buffer size, in bytes

set opt(sch-round-duration) 21312 ;# FairRR scheduler, in bytes

set opt(buffer-sharing)   "per-flow" ;# FairRR scheduler only

#

# link profiles

#

set opt(prfall) 0        ;# default burst profile index, overridden by prfndx's

set opt(prfsrc) {}      ;# QPSK_1_2 = 0, QPSK_3_4 = 1,

set opt(prfdst) {}     ;# QAM16_1_2 = 2, QAM16_3_4 = 3,

set opt(prfndx) {}     ;# QAM64_2_3 = 4, QAM64_3_4 = 5
```

```
#####  
###  
# TRAFFIC CONFIGURATION  
#####  
###  
  
#  
# traffic  
#  
set opt(trfsrc) { 0 } ;#{ 0 1 2 }  
set opt(trfdst) { 1 } ;#{ 3 3 3 }  
set opt(trftype-def) "cbr" ;# default traffic type  
set opt(trfstart-def) 15.0 ;# default application start time  
set opt(trfstop-def) "never" ;# default application start time  
set opt(trfnsrc-def) 1 ;# default number of sources per flow  
set opt(trfprio-def) 1 ;# default priority level  
#set opt(trftype) { voip voip vod }  
#set opt(trfnsrc) { 1 2 1 }  
#set opt(trfstart) { }  
#set opt(trfstop) { }  
#set opt(trfprio) { 1 1 0 }  
  
#  
# VoIP traffic  
#  
#set opt(voip-model) "one-to-one" ;# VoIP VAD model
```

Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

```
#set opt(voip-codec) "GSM.AMR" ;# VoIP codec
#set opt(voip-aggr) 2 ;# number of VoIP frames per packet

#
# BWA traffic
#
#set opt(bwa-rate) 100000 ;# rate of BWA traffic
#set opt(bwa-pkt) 192 ;# packet size of BWA traffic flows, in bytes

#
# CBR traffic
#
set opt(cbr-pkt) 1500 ;# packet size, in bytes
set opt(cbr-rate) 2000000 ;# rate, in b/s
set opt(cbr-rnd) 1 ;# set to 0 to have perfect CBR generation

#
# VoD traffic
#
#set opt(vod-trace) "traces/streaming.ns2"

#
# FTP traffic
#
#set opt(ftp-wnd) 64 ;# TCP maximum congestion window size
#set opt(ftp-pkt) 1024 ;# TCP Maximum Segment Size
```

```
#####  
###  
# NETWORK ENTRY CONFIGURATION  
#####  
###  
  
# can be one of  
  
# - scan  
  
# - sponsor  
  
# - tunnel  
  
# - link-est  
  
set opt(net-entry) ""  
set opt(net-entry-node) 12  
set opt(net-entry-sponsor) 11  
set opt(net-entry-start) 1  
set opt(net-entry-bs) 0  
  
#  
# workload definition procedure  
#  
proc create_connections {} {  
    global ns opt macmib node topo  
  
    # create traffic sources  
    for { set i 0 } { $i < [length $opt(trfsrc)] } { incr i } {  
  
        # if there is a default application, use it
```

```
set traffic $opt(trftype-def)
if { [length $opt(trftype)] > 0 } {
    set traffic [index $opt(trftype) $i]
}

# if there is a default number of sources per flow, use it
set nsources $opt(trfnsrc-def)
if { [length $opt(trfnsrc)] > 0 } {
    set nsources [index $opt(trfnsrc) $i ]
}

# if there is a default application start time, use it
set start $opt(trfstart-def)
if { [length $opt(trfstart)] > 0 } {
    set start [index $opt(trfstart) $i ]
}

# if there is a default application stop time, use it
set stop $opt(trfstop-def)
if { [length $opt(trfstop)] > 0 } {
    set stop [index $opt(trfstop) $i ]
}

# if there is a default priority level, use it
set prio $opt(trfprio-def)
if { [length $opt(trfprio)] > 0 } {
    set prio [index $opt(trfprio) $i ]
}
```

```
    }

# get the source/destination nodes

    set src [lindex $opt(trfsrc) $i]
    set dst [lindex $opt(trfdst) $i]

# print some debug info about this flow
if { $opt(debug) != "" } {
    puts "*** creating flow $i ($src -> $dst) with priority $prio"
    puts "  traffic = $traffic, number of sources = $nsources"
}

for { set j 0 } { $j < $nsources } { incr j } {

# create the application

if { $traffic == "cbr" } {
    create_cbr $src $dst $prio $i $start "never"

} elseif { $traffic == "bwa" } {
    create_bwa $src $dst $prio $i $start "never"

} elseif { $traffic == "voip" } {
    create_voip $src $dst $prio $i $start "never"

} elseif { $traffic == "ftp" } {
    create_ftp $src $dst $prio $i $start "never"

}
```



```
    } elseif { $traffic == "vod" } {  
        create_vod $src $dst $prio $i $start "never"  
  
        } elseif { $traffic == "udptunnel" } {  
            create_udptunnel $src $dst $prio $i $start "never"  
  
        } else {  
            puts "traffic '$traffic' not available"  
            exit 0  
        }  
    }  
}  
}  
}  
}  
  
#####  
###  
# MAIN BODY  
#####  
###  
  
getopt $argc $argv  
  
init  
create_topology  
create_nodes  
  
if { $opt(net-entry) == "scan" } {
```

```
    $ns at $opt(net-entry-start) ["$node($opt(net-entry-node)) getMac 0] scan"
} elseif { $opt(net-entry) == "sponsor" } {
    $ns at $opt(net-entry-start) \
        ["$node($opt(net-entry-node)) getMac 0] open-sponsor $opt(net-entry-sponsor)"
} elseif { $opt(net-entry) == "tunnel" } {
    set src $opt(net-entry-node)
    set dst $opt(net-entry-bs)
    set start $opt(net-entry-start)

    lappend opt(trfsrc) "$src"
    lappend opt(trfdst) "$dst"
    lappend opt(trftype) "udptunnel"
    lappend opt(trfnsrc) "1"
    lappend opt(trfstart) "$start"
    lappend opt(trfprio) "0"

    set opt(trftype-def) ""
    set opt(trfstart-def) ""
    set opt(trfnsrc-def) ""
    set opt(trfprio-def) ""
} elseif { $opt(net-entry) == "link-est" } {
    $ns at $opt(net-entry-start) \
        ["$node($opt(net-entry-node)) getMac 0] link-establishment"
}

create_connections
create_profiles
```

```
create_metrics
if { $opt(debug) != "" } {
    printopt
}
alive

$ns run
```

Αρχείο header.msh:

```
# vim: syntax=tcl
# Copyright (C) 2007 Dip. Ing. dell'Informazione, University of Pisa, Italy
# http://info.iet.unipi.it/~cng/ns2mesh80216/
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA, USA
```

```
#

#
# 802.16 mesh Tcl scenario configuration header file
# author: C.Cicconetti <claudio.cicconetti@iet.unipi.it>
#

#####
###
#           CONFIGURATION OF PARAMETERS           #
#####
###

#
# simulation environment
#
set opt(run)      1      ;# replic ID
set opt(duration) 10.0   ;# run duration, in seconds
set opt(warm)     2.0    ;# run duration, in seconds
set opt(out)      "out"  ;# statistics output file
set opt(debug)    ""     ;# debug configuration file, "" = no debug
set opt(startdebug) 0.0  ;# start time of debug output

#
# topology
#
set opt(topology) "chain" ;# see TOPOLOGIES
```

```

set opt(n)      4      ;# meaningful with: chain, ring, grid,
                    ;# multiring, star, clique, bintree
                                                    ;#

trinagular (see TOPOLOGIES)

set opt(branches)  3      ;# meaningful with: star, multiring
                    ;# (see TOPOLOGIES)

set opt(random-nodeid) "on" ;# mix up NodeIDs randomly

#
# PHY
#

set opt(channel)   1      ;# number of channels

set opt(radio)     1      ;# number of radios

set opt(chan-data-per) 0      ;# channel data PDU error rate

set opt(chan-ctrl-per) 0      ;# channel control PDU error rate

set opt(propagation) 4      ;# physical propagation, in us

set opt(sym-duration) 20     ;# OFDM symbol duration, in us

set opt(sym-perframe) 500    ;# number of OFDM symbols per frame

set opt(bandwidth)  "10"    ;# in MHz, if set overrides the OFDM symbol
                    ;# duration and the number of OFDM symbols
                    ;# per frame (ie. the frame duration)

set opt(cyclic-prefix) "1/16" ;# cyclic prefix, only used if bandwidth is set

set opt(control)   4      ;# number of MSH-DSCH opportunities per frame

set opt(cfg-interval) 16    ;# number of MSH-DSCH frames between two
                    ;# consecutive MSH-NCFG frames

set opt(msh-dsch-avg-bad) -1 ;# consecutive bad MSH-DSCH messages

set opt(msh-dsch-avg-good) -1 ;# consecutive good MSH-DSCH messages

```

```

#
# MAC
#
set opt(allocation) "contiguous" ;# MSH-DSCH allocation type
set opt(hest-curr) .1 ;# weight for H's estimations (current value)
set opt(hest-past) .9 ;# weight for H's estimations (past values)

#
# bandwidth manager
#
set opt(bwmanager) "fair-rr" ;# bandwidth manager type
set opt(availabilities) "on" ;# RR, FairRR bwmanagers
set opt(regrant) "on" ;# RR, FairRR bwmanagers
set opt(regrant-offset) 1 ;# RR, FairRR bwmanagers, in frames
set opt(regrant-duration) 1 ;# RR, FairRR bwmanagers, in handshakes
set opt(grant-fairness) "on" ;# FairRR bwmanager = {on, off}
set opt(regrant-same-horizon) "off" ;# FairRR bwmanager = {on, off}
set opt(regrant-fairness) "on" ;# FairRR bwmanager = {on, off}
set opt(request-fairness) "on" ;# FairRR bwmanager = {on, off}
set opt(bwm-round-duration) 21312 ;# FairRR bwmanager, in bytes
set opt(weight-timeout) 120 ;# FairRR bwmanager, in sec
set opt(max-deficit) 0 ;# FairRR bwmanager, in bytes
set opt(max-backlog) 0 ;# FairRR bwmanager, in bytes
set opt(weight-flow) "on" ;# FairRR bwmanager = {on, off}
set opt(grant-rnd-channel) "on" ;# FairRR bwmanager = {on, off}
set opt(dd-timeout) "50" ;# FairRR bwmanager, in MSH-DSCH opps

```

```

set opt(min-grant)      "1"      ;# FairRR bwmanager, in OFDM symbols

set opt(prio-weight)   "1 2 4"   ;# priority weights, used by both the
                               ;# FairRR bwmanager and the scheduler

#
# distributed election coordinator
#
set opt(coordinator)   "standard" ;# coordinator type
set opt(coordmode)     "xmtunaware" ;# coordinator mode
set opt(holdoff-dsch-def) 0       ;# holdoff exponent for MSH-DSCH msgs
set opt(holdoff-ncfg-def) 0       ;# holdoff exponent for MSH-NCFG msgs
set opt(holdoff-dsch)   {}        ;# used to set different holdoffs
set opt(holdoff-ncfg)   {}        ;# used to set different holdoffs
set opt(control-period) 4         ;# dummy coordinator, number of frames
set opt(max-advertised) -1        ;# maximum number of advertised neighbors

#
# forwarding
#
set opt(forwarding)     "spf"     ;# forwarding type
set opt(maxnumhops)     20        ;# maximum path length
set opt(profileupdateperiod) 4     ;# number of frames to update profiles
set opt(samplingperiod) 4         ;# sampling nodes'load period (frames)
set opt(chi)            0.5       ;# weight of grants for links' cost
set opt(beta)           0.5       ;# weight of new samples for nodes' load

```

```

#
# packet scheduler
#
set opt(scheduler)      "fair-rr"  ;# packet scheduler type
set opt(buffer)         1000000    ;# buffer size, in bytes
set opt(sch-round-duration) 21312  ;# FairRR scheduler, in bytes
set opt(buffer-sharing) "per-flow" ;# FairRR scheduler only

#
# link profiles
#
set opt(prfall) 0 ;# default burst profile index, overridden by prfndx's
set opt(prfsrc) {} ;# QPSK_1_2 = 0, QPSK_3_4 = 1,
set opt(prfdst) {} ;# QAM16_1_2 = 2, QAM16_3_4 = 3,
set opt(prfndx) {} ;# QAM64_2_3 = 4, QAM64_3_4 = 5

#####
###
#           DEFINITION OF PROCEDURES           #
#####
###

#
# parse command-line options and store values into the $opt(.) hash
#
proc getopt {argc argv} {
    global opt

```



```
for {set i 0} {$i < $argc} {incr i} {
    set arg [lindex $argv $i]
    if {[string range $arg 0 0] != "-"} continue

    set name [string range $arg 1 end]
    set opt($name) [lindex $argv [expr $i+1]]
}
}

#
# print out options
#
proc printopt {} {
    global opt

    foreach x [lsort [array names opt]] {
        puts "$x = $opt($x)"
    }
}

#
# alive function
#
proc alive {} {
    global ns opt
```

```
if { [$ns now] != 0 } {
  puts -nonewline \
    [format "elapsed %.0f s (remaining %.0f s) completed %.f%%" \
    [$ns now] \
    [expr $opt(duration) - [$ns now]] \
    [expr 100 * [$ns now] / $opt(duration)]]
  if { [$ns now] >= $opt(warm) } {
    puts " stat collection ON"
  } else {
    puts ""
  }
}
$ns at [expr [$ns now] + $opt(duration) / 10.0] "alive"
}

#
# collect statistics at the end of the simulation
#
proc finish {} {
  global ns simtime opt

  $ns flush-trace
  close $opt(trace)
  # print statistics to output file
  $ns stat print

  # print out the simulation time
```

```
set simtime [expr [clock seconds] - $simtime]
puts "run duration: $simtime s"

exit 0
}

#
# initialize simulation
#
proc init {} {
    global opt defaultRNG ns simtime

    # create the simulator instance
    set ns [new Simulator] ;# create a new simulator instance
    $defaultRNG seed 1

    # initialize statistics collection
    $ns run-identifier $opt(run)
    $ns stat file "$opt(out)"
    $ns at $opt(warm) "$ns stat on"
    $ns at $opt(duration) "finish"

    # add default probes
    $ns stat add e2e_owd_a avg discrete
    $ns stat add e2e_tpt avg rate
    $ns stat add e2e_owpl avg rate
    $ns stat add tcp_cwnd_a avg continuous
```

```
$ns stat add tcp_dupacks avg continuous
$ns stat add tcp_ssthresh avg continuous
$ns stat add tcp_rtt avg continuous
$ns stat add tcp_srtt avg continuous
$ns stat add fairness avg continuous

$ns stat add tcp_cwnd_d dst continuous 0 128 128
$ns stat add e2e_owd_d dst discrete 0.0 5.0 100
$ns stat add e2e_ipdv_d dst discrete 0.0 5.0 100

# open trace files
set opt(trace) [open ./MESH.res w]

# configure 802.16 debug
set debug [new WimaxDebug]
$debug disable

# read configuration from the debug file, in needed
if { $opt(debug) != "" } {
    # enable debug at startdebug time
    $ns at $opt(startdebug) "$debug enable"

    # try to open the debug file for reading debug configuration
    if { ![file exists $opt(debug)] } {
        puts "Oops, file '$opt(debug)' does not exist. Please use -debug"
        exit 0
    }
}
```

```
set debugfile [open $opt(debug) r]
while { [gets $debugfile line] >= 0 } {
    if { [string index $line 0] != "#" } { ;# skip comments
        $debug trace $line on
    }
}

# close the debug configuration file
close $debugfile
}

set simtime [clock seconds]

$ns trace-all $opt(trace)
}

#
# define the network topology
#
proc create_topology {} {
    global ns opt node node topo defaultRNG invmap map

    # set the node configuration options
    # most of them are not used with IEEE 802.16 nodes, but we still
    # set them so as to avoid annoying ns2 complaints

    set topo_unused [new Topography] ;# not used
```

```
set chan_unused [new Channel/WirelessChannel] ;# not used

$ns node-config -lType LL -macType WimshMac -adhocRouting DumbAgent

$ns node-config \
  -ifqType Queue/DropTail/PriQueue \
  -ifqLen 50 \
  -antType Antenna/OmniAntenna \
  -propType Propagation/TwoRayGround \
  -phyType Phy/WirelessPhy \
  -topoInstance $topo_unused \
  -agentTrace OFF \
  -routerTrace OFF \
  -macTrace OFF \
  -movementTrace OFF \
  -channel $chan_unused

# create and initialize the network topology

set topo [new WimshTopology/Simple]

# compute the number of nodes according to different topologies
if { $opt(topology) == "chain" \
  || $opt(topology) == "ring" \
  || $opt(topology) == "multiring" } {
  # the number of nodes is n
  set opt(nodes) $opt(n)
} elseif { $opt(topology) == "grid" || $opt(topology) == "grid-full" } {
  # the number of nodes is n^2
```

```

set opt(nodes) [expr $opt(n) * $opt(n)]
} elseif { $opt(topology) == "triangular" } {
    # the number of nodes is (n+1)(n+2)/2
    set opt(nodes) [expr ( ( $opt(n) + 1 ) * ( $opt(n) + 2 ) ) / 2]
} elseif { $opt(topology) == "clique" } {
    # the number of nodes is n
    set opt(nodes) $opt(n)
} elseif { $opt(topology) == "star" } {
    # the number of nodes is branches * (n-1) + 1
    set opt(nodes) [expr $opt(branches) * ( $opt(n) - 1 ) + 1]
} elseif { $opt(topology) == "bintree" } {
    # the number of nodes is 2^n - 1
    set opt(nodes) [expr int(pow(2, $opt(n)) - 1)]
} elseif { $opt(topology) == "hexagon" } {
    # the number of nodes is 6
    set opt(nodes) 6
} elseif { $opt(topology) == "star2" } {
    # the number of nodes is 8
    set opt(nodes) 8
} else {
    # unknown topology
    puts "unknown topology '$opt(topology)'"
    exit 1
}

#
# mix up the node indexes into the map

```

```
#
if { $opt(random-nodeid) == "on" } {
  for { set i 0 } { $i < $opt(nodes) } { incr i } {
    while { 1 } {
      set x [$defaultRNG integer $opt(nodes)]
      set found 0
      for { set j 0 } { $j < $i } { incr j } {
        if { $map($j) == $x } {
          set found 1
          break
        }
      }
    }
    if { $found == 0 } {
      set map($i) $x
      break
    }
  }
}

# invert the map created
for { set i 0 } { $i < $opt(nodes) } { incr i } {
  set invmap($map($i)) $i
}
} else {
  for { set i 0 } { $i < $opt(nodes) } { incr i } {
    set map($i) $i
    set invmap($i) $i
  }
}
```



```
}  
}  
  
# print the map created  
if { $opt(debug) != "" } {  
  puts "*** index <-> NodeID/IP address ***"  
  for { set i 0 } { $i < $opt(nodes) } { incr i } {  
    puts "index = $invmap($i), NodeID = $i"  
  }  
}  
  
#  
# CHAIN, RING  
#  
if { $opt(topology) == "chain" || $opt(topology) == "ring" } {  
  for { set i 0 } { $i < [expr $opt(n) - 1] } { incr i } {  
    $topo connect $map($i) $map([expr $i + 1])  
  }  
  if { $opt(topology) == "ring" } {  
    $topo connect $map([expr $opt(n) - 1]) $map(0)  
  }  
}  
  
#  
# MULTIRING  
#  
} elseif { $opt(topology) == "multiring" } {  
  if { [expr $opt(branches) % 2] != 0 } {
```

```

puts "In a multiring, the number of branches must be an even number"

exit 0

}

for { set i 1 } { $i <= [expr $opt(branches) / 2] } { incr i } {

  for { set j 0 } { $j < $opt(n) } { incr j } {

    $topo connect \

      $map([expr $j % $opt(n)]) \

      $map([expr ($j + $i) % $opt(n)])

    }

  }

#

# GRID

#

} elseif { $opt(topology) == "grid" } {

  for { set i 0 } { $i < $opt(n) } { incr i } {

    for { set j 0 } { $j < $opt(n) } { incr j } {

      set curnode [expr $j + $opt(n) * $i]

      if { $i > 0 } {

        $topo connect $map($curnode) $map([expr $curnode - $opt(n)])

      }

      if { $j > 0 } {

        $topo connect $map($curnode) $map([expr $curnode - 1])

      }

    }

  }

}

}

```

```

#
# FULL GRID (no edge effect)
#
} elseif { $opt(topology) == "grid-full" } {
    for { set i 0 } { $i < $opt(n) } { incr i } {
        for { set j 0 } { $j < $opt(n) } { incr j } {
            set curnode [expr $j + $opt(n) * $i]
                if { $i > 0 } {
                    $topo connect $map($curnode) $map([expr $curnode - $opt(n)])
                } else {
                    $topo connect $map($curnode) $map([expr $curnode + ( $opt(n) - 1 ) * $opt(n)])
                }
            if { $j > 0 } {
                $topo connect $map($curnode) $map([expr $curnode - 1])
            } else {
                $topo connect $map($curnode) $map([expr $curnode + $opt(n) - 1])
            }
        }
    }
}

#
# TRIANGULAR
#
} elseif { $opt(topology) == "triangular" } {
    for { set i 0 } { $i < $opt(n) } { incr i } {
        for { set j 0 } { $j <= $i } { incr j } {

```

```

set curnode [expr ( $i * ( $i + 1 ) ) / 2 + $j]

        set leftnode [expr $curnode + $i + 1]
        set rightnode [expr $leftnode + 1]
        topo connect $map($curnode) $map($leftnode)
        topo connect $map($curnode) $map($rightnode)
        topo connect $map($leftnode) $map($rightnode)

    }
}

#
# CLIQUE
#
elseif { $opt(topology) == "clique" } {
    for { set i 0 } { $i < $opt(n) } { incr i } {
        for { set j [expr $i + 1] } { $j < $opt(n) } { incr j } {
            topo connect $map($i) $map($j)
        }
    }
}

#
# BINTREE
#
elseif { $opt(topology) == "bintree" } {
    for { set i 0 } { $i < [expr $opt(n) - 1] } { incr i } {
        for { set j 0 } { $j < [expr int(pow(2, $i))] } { incr j } {
            topo connect $map([expr int(pow(2, $i)) - 1 + $j]) \
                $map([expr int(pow(2, $i+1)) - 1 + $j * 2])
        }
    }
}

```

```

$topo connect $map([expr int(pow(2, $i)) - 1 + $j]) \
    $map([expr int(pow(2, $i+1)) - 1 + $j * 2 + 1])
}
}

#
# STAR
#
} elseif { $opt(topology) == "star" } {
    for { set i 0 } { $i < [expr $opt(n) - 1] } { incr i } {
        for { set j 0 } { $j < $opt(branches) } { incr j } {
            if { $i == 0 } {
                $topo connect $map(0) $map([expr $j + 1])
            } else {
                $topo connect $map([expr $opt(branches) * ( $i - 1 ) + $j + 1]) \
                    $map([expr $opt(branches) * $i + $j + 1])
            }
        }
    }
}

#
# HEXAGON
#
} elseif { $opt(topology) == "hexagon" } {
    $topo connect $map(0) $map(1)
    $topo connect $map(0) $map(2)
    $topo connect $map(1) $map(2)
}

```

```
$topo connect $map(1) $map(3)
$topo connect $map(2) $map(4)
$topo connect $map(3) $map(4)
$topo connect $map(3) $map(5)
$topo connect $map(4) $map(5)

#
# STAR2
#
} elseif { $opt(topology) == "star2" } {
    $topo connect $map(0) $map(1)
    $topo connect $map(0) $map(2)
    $topo connect $map(0) $map(3)
    $topo connect $map(1) $map(2)
    $topo connect $map(1) $map(3)
    $topo connect $map(2) $map(3)

    $topo connect $map(0) $map(4)
    $topo connect $map(1) $map(5)
    $topo connect $map(2) $map(6)
    $topo connect $map(3) $map(7)
}

# initialize the topology
$topo initialize

# dump the topology to standard error, if needed
```

```
if { $opt(debug) != "" } {
    $topo dump
}
}

#
# creates and initializes nodes (MAC/PHY) + management information bases
#
proc create_nodes {} {
    global ns opt node macmib node topo invmap phymib map

    # compute the OFDM symbol duration and the number of OFDM symbols
    # per frame according to the standard document, par. 12.3.2
    # if the bandwidth is not set, then user-defined values are used

    if { $opt(bandwidth) != "" } {
        if { $opt(cyclic-prefix) == "1/4" } {
            set G .25
        } elseif { $opt(cyclic-prefix) == "1/8" } {
            set G .125
        } elseif { $opt(cyclic-prefix) == "1/16" } {
            set G .0625
        } elseif { $opt(cyclic-prefix) == "1/32" } {
            set G .03125
        } else {
            puts "The cyclic prefix $opt(cyclic-prefix) is not valid"
            exit 0
        }
    }
}
```

```
}

# select the frame duration, in ms, and the 'x' values
if { $opt(bandwidth) == "3.5" } {
    set f 4
    set x 4e6
} elseif { $opt(bandwidth) == "7" } {
    set f 4
    set x 8e6
} elseif { $opt(bandwidth) == "3" } {
    set f 10
    set x 3.44e6
} elseif { $opt(bandwidth) == "5.5" } {
    set f 10
    set x 6.32e6
} elseif { $opt(bandwidth) == "10" } {
    set f 4
    set x 1.152e7
} else {
    puts "The bandwidth value $opt(bandwidth) MHz is not valid"
    exit 0
}

# compute the OFDM symbol duration
set T [expr 1e6 * (1.0 + $G) * (256.0 / $x)];

# user-defined values
```



```
} else {  
    set T $opt(sym-duration)  
    set f [expr 1e-3 * $opt(sym-duration) * $opt(sym-perframe)]  
}  
  
# create and initialize the PHY MIB  
set phymib [new WimshPhyMib]  
$phymib symDuration $T  
$phymib frameDuration $f  
$phymib controlSlots $opt(control)  
$phymib cfg-interval $opt(cfg-interval)  
$phymib recompute  
  
# create and initialize the MAC MIB  
set macmib [new WimshMacMib]  
$macmib phymib $phymib  
$macmib allocation $opt(allocation)  
  
# create and initialize the 802.16 channels  
for { set j 0 } { $j < $opt(channel) } { incr j } {  
    set chan($j) [new WimshChannel]  
    $chan($j) topology $topo  
    $chan($j) propagation $opt(propagation)  
    $chan($j) id [expr 1 + $j]  
    if { $opt(chan-data-per) != 0 } {  
        $chan($j) error data $opt(chan-data-per)  
    }  
}
```

```
if { $opt(chan-ctrl-per) != 0 } {  
    $chan($j) error control $opt(chan-ctrl-per)  
}  
}  
  
# create the 802.16 nodes  
for { set i 0 } { $i < $opt(nodes) } { incr i } {  
    # create node (+MAC) and bind MAC-LL  
    set node($invmap($i)) [$ns node]  
  
    # get a reference to the node's MAC  
    set mac($i) [$node($invmap($i)) getMac 0]  
  
    # set the index for statistical purposes  
    $mac($i) index $invmap($i)  
  
    # set the weight for the estimation of the H values  
    $mac($i) estcurr $opt(hest-curr)  
    $mac($i) estpast $opt(hest-past)  
  
    # bind the MAC to the PHY and MAC MIBs  
    $mac($i) macmib $macmib  
    $mac($i) phymib $phymib  
  
    # bind the MAC to the topology  
    $mac($i) topology $topo
```

```
# define the list of channels that all the nodes can use
for { set j 0 } { $j < $opt(channel) } { incr j } {
    $mac($i) channel $chan($j)
}

# create and initialize the list of radios of this node
for { set j 0 } { $j < $opt(radio) } { incr j } {
    set phy [new WimshPhy]
    $phy mac $mac($i)
    $phy topology $topo
    $phy phymib $phymib
    $phy epsilon 5
    $mac($i) phy $phy
}

# create some MAC submodules
$mac($i) forwarding $opt(forwarding)
$mac($i) bwmanager $opt(bwmanager)
$mac($i) scheduler $opt(scheduler)
$mac($i) coordinator $opt(coordinator)

# configure the forwarding module
if { $opt(forwarding) == "opp" } {
    $mac($i) forwarding maxnumhops $opt(maxnumhops)
    $mac($i) forwarding samplingperiod $opt(samplingperiod)
    $mac($i) forwarding profileupdateperiod $opt(profileupdateperiod)
    $mac($i) forwarding chi $opt(chi)
}
```

```

$mac($i) forwarding beta $opt(beta)
}

# configure the coordinator
if { $opt(coordinator) == "dummy" } {
    if { $opt(coordmode) == "fixed" } {
        $mac($i) coordinator slot [expr $i % $opt(control)]
        $mac($i) coordinator frame $opt(control-period)
        $mac($i) coordinator first [expr $i / $opt(control)]
    } elseif { $opt(coordmode) == "moving" } {
        $mac($i) coordinator slot [expr $i * \
            ( $opt(control) * $opt(control-period) ) / $opt(nodes)]
        $mac($i) coordinator period \
            [expr $opt(control) * $opt(control-period)]
    }
    $mac($i) coordinator mode $opt(coordmode)
} elseif { $opt(coordinator) == "standard" } {
    $mac($i) coordinator nextxmttime $i
    if { [llength $opt(holdoff-dsch)] == 0 } {
        $mac($i) coordinator holdoffexp-dsch $opt(holdoff-dsch-def)
    } else {
        $mac($i) coordinator holdoffexp-dsch \
            [lindex $opt(holdoff-dsch) $invmap($i)]
    }
    if { [llength $opt(holdoff-ncfg)] == 0 } {
        $mac($i) coordinator holdoffexp-ncfg $opt(holdoff-ncfg-def)
    } else {

```

```
$mac($i) coordinator holdoffexp-ncfg \  
    [lindex $opt(holdoff-ncfg) $invmap($i)]  
  
}  
  
$mac($i) coordinator mode $opt(coordmode)  
  
$mac($i) coordinator max-advertised $opt(max-advertised)  
  
}  
  
# configure the bwmanager  
  
if { $opt(bwmanager) == "fair-rr" } {  
    $mac($i) bwmanager availabilities $opt(availabilities)  
  
    $mac($i) bwmanager fairness no  
  
    if { $opt(grant-fairness) == "on" } {  
        $mac($i) bwmanager fairness grant  
    }  
  
    if { $opt(regrant-fairness) == "on" } {  
        $mac($i) bwmanager fairness regrant  
    }  
  
    if { $opt(request-fairness) == "on" } {  
        $mac($i) bwmanager fairness request  
    }  
  
    $mac($i) bwmanager regrant $opt(regrant)  
  
    $mac($i) bwmanager regrant-offset $opt(regrant-offset)  
  
    $mac($i) bwmanager regrant-duration $opt(regrant-duration)  
  
    $mac($i) bwmanager regrant-same-horizon $opt(regrant-same-horizon)  
  
    $mac($i) bwmanager round-duration $opt(bwm-round-duration)  
  
    $mac($i) bwmanager max-deficit $opt(max-deficit)  
  
    $mac($i) bwmanager max-backlog $opt(max-backlog)
```

```
if { $opt(grant-rnd-channel) == "on" } {
    $mac($i) bwmanager grant-fit channel random
}

$mac($i) bwmanager dd-timeout $opt(dd-timeout)

    $mac($i) bwmanager min-grant $opt(min-grant)

    # configure the weight manager of the bandwidth manager

$mac($i) bwmanager wm weight-flow $opt(weight-flow)

    for { set j 0 } { $j < [llength $opt(prio-weight)] } { incr j } {
        $mac($i) bwmanager wm prio-weight $j [lindex $opt(prio-weight)
$]]
    }

if { $opt(weight-timeout) != "never" } {
    $mac($i) bwmanager wm weight-timeout $opt(weight-timeout)
}
} else {
    die "Invalid bandwidth manager '$opt(bwmanager)'"
}

# configure the scheduler

$mac($i) scheduler size $opt(buffer)

if { $opt(scheduler) == "fair-rr" } {
    $mac($i) scheduler round-duration $opt(sch-round-duration)
    $mac($i) scheduler buffer-sharing $opt(buffer-sharing)
    if { $opt(weight-timeout) != "never" } {
        $mac($i) scheduler weight-timeout $opt(weight-timeout)
    }
}
```

```
        for { set j 0 } { $j < [length $opt(prio-weight)] } { incr j } {  
            $mac($i) scheduler prio-weight $j [index $opt(prio-weight) $j]  
        }  
    }  
  
    # configure corruption of MSH-DSCH messages at the MAC layer  
    $mac($i) msh-dsch-avg-bad $opt(msh-dsch-avg-bad)  
    $mac($i) msh-dsch-avg-good $opt(msh-dsch-avg-good)  
  
    # initialize MAC data structures  
    $mac($i) initialize  
}  
}  
  
#  
# definition of link profiles. Note: links are symmetric  
#  
proc create_profiles {} {  
    global ns opt node map phymib  
  
    # QPSK_1_2 = 0 QPSK_3_4 = 1  
    # QAM16_1_2 = 2 QAM16_3_4 = 3  
    # QAM64_2_3 = 4 QAM64_3_4 = 5  
  
    # set the default burst profile on all MACs  
    for { set i 0 } { $i < $opt(nodes) } { incr i } {  
        [$node($i) getMac 0] profile $opt(prfall) all  
    }  
}
```

```
}  
  
# set specific burst profiles, overriding the default one  
for { set i 0 } { $i < [length $opt(prfsrc)] } { incr i } {  
    set macsrc [$node([index $opt(prfsrc) $i]) getMac 0]  
    set macdst [$node([index $opt(prfdst) $i]) getMac 0]  
    $macsrc profile [index $opt(prfndx) $i] $map([index $opt(prfdst) $i])  
    $macdst profile [index $opt(prfndx) $i] $map([index $opt(prfsrc) $i])  
}  
  
# dump the forwarding data structures if needed  
# dump the PHY MIB profiles of each MAC layer  
if { $opt(debug) != "" } {  
    for { set i 0 } { $i < $opt(nodes) } { incr i } {  
        set mac [$node($i) getMac 0]  
        $mac dump profile  
        $mac forwarding dump  
    }  
    $phymib dump  
}  
}
```

Αρχείο traffic.msh:

```
# vim: syntax=tcl  
# Copyright (C) 2007 Dip. Ing. dell'Informazione, University of Pisa, Italy  
# http://info.iet.unipi.it/~cng/ns2mesh80216/  
#
```



Πτυχιακή του φοιτητή Καρανίκα Γεωργίου

```
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA, USA
#
#
# 802.16 mesh Tcl scenario traffic models configuration
# author: C.Cicconetti <claudio.cicconetti@jet.unipi.it>
#
#
# load default values
#
set opt(trfsrc) {}
set opt(trfdst) {}
```

```
set opt(trftype-def) "cbr" ;# default traffic type
set opt(trfstart-def) 1.0 ;# default application start time
set opt(trfstop-def) "never" ;# default application start time
set opt(trfnsrc-def) 1 ;# default number of sources per flow
set opt(trfprio-def) 1 ;# default priority level

set opt(trftype) { }
set opt(trfnsrc) { }
set opt(trfstart) { }
set opt(trfstop) { }
set opt(trfprio) { }

set opt(voip-dur) 0 ;# average call duration (0 = no VoIP calls)
set opt(voip-start) 1 ;# start time of the first call
set opt(voip-prio) 1 ;# priority level of VoIP within the IEEE 802.16 mesh
set opt(voip-int) 60 ;# average interval between two consecutive calls
set opt(voip-model) "one-to-one" ;# VoIP VAD model
set opt(voip-codec) "GSM.AMR" ;# VoIP codec
set opt(voip-aggr) 2 ;# number of VoIP frames per packet

set opt(bwa-flows) 0 ;# number of random BWA traffic flows
set opt(bwa-rate) 100000 ;# rate of BWA traffic
set opt(bwa-pkt) 192 ;# packet size of BWA traffic flows, in bytes
set opt(bwa-start) 1 ;# start time of all BWA traffic flows
set opt(bwa-stop) "never" ;# end time of all BWA traffic flows
set opt(bwa-prio) 0 ;# priority of BWA traffic
set opt(bwa-fid) 5 ;# flow ID of BWA traffic as a whole
```

```
set opt(cbr-pkt) 1000 ;# packet size, in bytes
set opt(cbr-rate) 2000000 ;# rate, in b/s
set opt(cbr-rnd) 1 ;# set to 0 to have perfect CBR generation

set opt(vod-trace) "traces/videoconf.mpeg4.ns2"

set opt(ftp-wnd) 64 ;# TCP maximum congestion window size
set opt(ftp-pkt) 1024 ;# TCP Maximum Segment Size

# can be one of
# - scan
# - sponsor
# - tunnel
# - link-est

set opt(net-entry) ""
set opt(net-entry-node) 12
set opt(net-entry-sponsor) 11
set opt(net-entry-start) 1
set opt(net-entry-bs) 0

#
# create a video traffic flow between two nodes, agents included
#
proc create_vod { src dst prio fid start stop } {
    global macmib opt ns node
    if { $opt(debug) != "" } {
        puts "create_vod $src $dst $prio $fid $start $stop"
```

```
}

set trace [new Tracefile]

set app [new Application/Traffic/Trace]

$trace filename $opt(vod-trace)

$app attach-tracefile $trace

set agtsrc [new Agent/UDP]

set agtdst [new Agent/UDP]

$agtsrc set class_ $fid

$agtsrc set packetSize_ 65536

# set application start/stop times

$ns at $start "$app start"

if { $stop != "never" } {
    $ns at $stop "$app stop"
}

$macmib crc $fid crc

$macmib priority $fid $prio

$macmib precedence $fid 0

$ns attach-agent $node($src) $agtsrc

$ns attach-agent $node($dst) $agtdst

$ns connect $agtsrc $agtdst

$app attach-agent $agtsrc
```

```
# end-to-end modules statistics collection

set tag [new e2et]

set mon [new e2em]

$agtsrc attach-e2et $tag

$agtdst attach-e2em $mon

$mon index $fid

$mon start-log

}

#

# create an UDP tunnel traffic flow between two nodes, agents included

#

proc create_udptunnel { src dst prio fid start stop } {

    global macmib opt ns node

    if { $opt(debug) != "" } {

        puts "create_udptunnel $src $dst $prio $fid $start $stop"

    }

    set agtsrc [new Agent/WimshUdpTunnel]

    set agtdst [new Agent/WimshUdpTunnel]

    if { $opt(debug) != "" } {

        $agtsrc set debug_1

        $agtdst set debug_1

    }

    $ns at $start "$agtsrc start"

    $ns stat add udp_tunnel_delay avg discrete

}
```

```
$macmib crc $fid crc
$macmib priority $fid $prio
$macmib precedence $fid 0

$agtsrc set class_ $fid
$agtdst set class_ $fid

$ns attach-agent $node($src) $agtsrc
$ns attach-agent $node($dst) $agtdst
$ns connect $agtsrc $agtdst

# end-to-end modules statistics collection

set tag [new e2et]
set mon [new e2em]
$agtsrc attach-e2et $tag
$agtdst attach-e2em $mon
$mon index $fid
$mon start-log
}

#
# create an FTP traffic flow between two nodes, agents included
#
proc create_ftp { src dst prio fid start stop } {
    global macmib opt ns node
    if { $opt(debug) != "" } {
        puts "create_ftp $src $dst $prio $fid $start $stop"
```

```
}

set app [new Application/FTP]

set agtsrc [new Agent/TCP/Reno]
set agtdst [new Agent/TCPSink]

$agtsrc set class_ $fid
$agtdst set class_ $fid
$agtsrc set window_ $opt(ftp-wnd)
$agtsrc set maxcwnd_ $opt(ftp-wnd)
$agtsrc set packetSize_ $opt(ftp-pkt)
$agtsrc set ts_resetRTO_ true

$agtsrc tracevar cwnd_
$agtsrc tracevar dupacks_
$agtsrc tracevar ssthresh_
$agtsrc tracevar rtt_
$agtsrc tracevar srtt_

# set application start/stop times
$ns at $start "$app start"
if { $stop != "never" } {
    $ns at $stop "$app stop"
}

$macmib crc $fid crc
```

```
$macmib priority $fid $prio
$macmib precedence $fid 0

$ns attach-agent $node($src) $agtsrc
$ns attach-agent $node($dst) $agtdst
$ns connect $agtsrc $agtdst
$app attach-agent $agtsrc

# end-to-end modules statistics collection
set tag [new e2et]
set mon [new e2em]
$agtsrc attach-e2et $tag
$agtdst attach-e2em $mon
$mon index $fid
$mon start-log
# $mon fairness-estimator $fairness
}
#
# create a CBR traffic flow between two nodes, agents included
#
proc create_cbr { src dst prio fid start stop } {
    global macmib opt ns node
    if { $opt(debug) != "" } {
        puts "create_cbr $src $dst $prio $fid $start $stop"
    }
}

set app [new Application/Traffic/CBR]
```



```
$app set random_ $opt(cbr-rnd)
$app set packetSize_ $opt(cbr-pkt)
$app set rate_ $opt(cbr-rate)

set agtsrc [new Agent/UDP]
set agtdst [new Agent/UDP]
$agtsrc set class_ $fid

# set application start/stop times
$ns at $start "$app start"
if { $stop != "never" } {
    $ns at $stop "$app stop"
}

$macmib crc $fid crc
$macmib priority $fid $prio
$macmib precedence $fid 0

$ns attach-agent $node($src) $agtsrc
$ns attach-agent $node($dst) $agtdst
$ns connect $agtsrc $agtdst
$app attach-agent $agtsrc

# end-to-end modules statistics collection
set tag [new e2et]
set mon [new e2em]
$agtsrc attach-e2et $tag
```

```
$agtdst attach-e2em $mon
$mon index $fid
$mon start-log
# $mon fairness-estimator $fairness
}
#
# create a BWA traffic flow between two nodes, agents included
#
proc create_bwa { src dst prio fid start stop } {
    global macmib opt ns node
    if { $opt(debug) != "" } {
        puts "create_bwa $src $dst $prio $fid $start $stop"
    }
}

# create a RNG for this application
set rng [new RNG]

# create and configure the BWA Internet
set app [new Application/Traffic/BwaInternet]
$app pkt-size $opt(bwa-pkt)
$app rate $opt(bwa-rate)
$app use-rng $rng

set agtsrc [new Agent/UDP]
set agtdst [new Agent/UDP]
$agtsrc set class_ $fid
```

```
# set application start/stop times

$ns at $start "$app start"

if { $stop != "never" } {
    $ns at $stop "$app stop"
}

$macmib crc $fid crc

$macmib priority $fid $prio

$macmib precedence $fid 0

$ns attach-agent $node($src) $agtsrc
$ns attach-agent $node($dst) $agtdst
$ns connect $agtsrc $agtdst
$app attach-agent $agtsrc

# end-to-end modules statistics collection

set tag [new e2et]
set mon [new e2em]

$agtsrc attach-e2et $tag
$agtdst attach-e2em $mon

$mon index $fid

$mon start-log
}

#
# create a VoIP traffic flow between two nodes, agents included
#
```

```
proc create_voip { src dst prio fid start stop } {  
    global macmib opt ns node  
    if { $opt(debug) != "" } {  
        puts "create_voip $src $dst $prio $fid $start $stop"  
    }  
  
    # create a RNG for this application  
    set rng [new RNG]  
  
    # create and configure the VoIP application  
    set app [new VoipSource]  
    $app model $opt(voip-model)  
    $ns at $start "$app start"  
    if { $stop != "never" } {  
        $ns at $stop "$app stop"  
    }  
  
    set encoder [new Application/VoipEncoder]  
    $encoder codec $opt(voip-codec)  
    $app encoder $encoder  
  
    set decoder [new Application/VoipDecoderOptimal]  
    $decoder id $fid  
    $decoder cell-id 0  
    $decoder emodel $opt(voip-codec)  
  
    set aggregate [new Application/VoipAggregate]
```

```
$aggregate nframes $opt(voip-aggr)

set agtsrc [new Agent/UDP]
set agtdst [new Agent/UDP]

$agtsrc set class_ $fid

$macmib crc $fid crc
$macmib priority $fid $prio
$macmib precedence $fid 0

$ns attach-agent $node($src) $agtsrc
$ns attach-agent $node($dst) $agtdst
$ns connect $agtsrc $agtdst

$encoder attach-agent $agtsrc
$decoder attach-agent $agtdst

$aggregate attach-agent $agtsrc
$encoder aggregate $aggregate

# end-to-end modules statistics collection
set tag [new e2et]
set mon [new e2em]
$agtsrc attach-e2et $tag
$agtdst attach-e2em $mon
$mon index $fid
```

```
$mon start-log  
}
```

Αρχείο metrics.msh:

```
proc create_metrics {} {  
  global ns opt  
  $ns stat add e2e_owd_a avg rate  
  $ns stat add e2e_tpt avg rate  
  $ns stat add e2e_owpl avg rate  
  $ns stat add wimsh_mac_tpt avg rate  
}
```