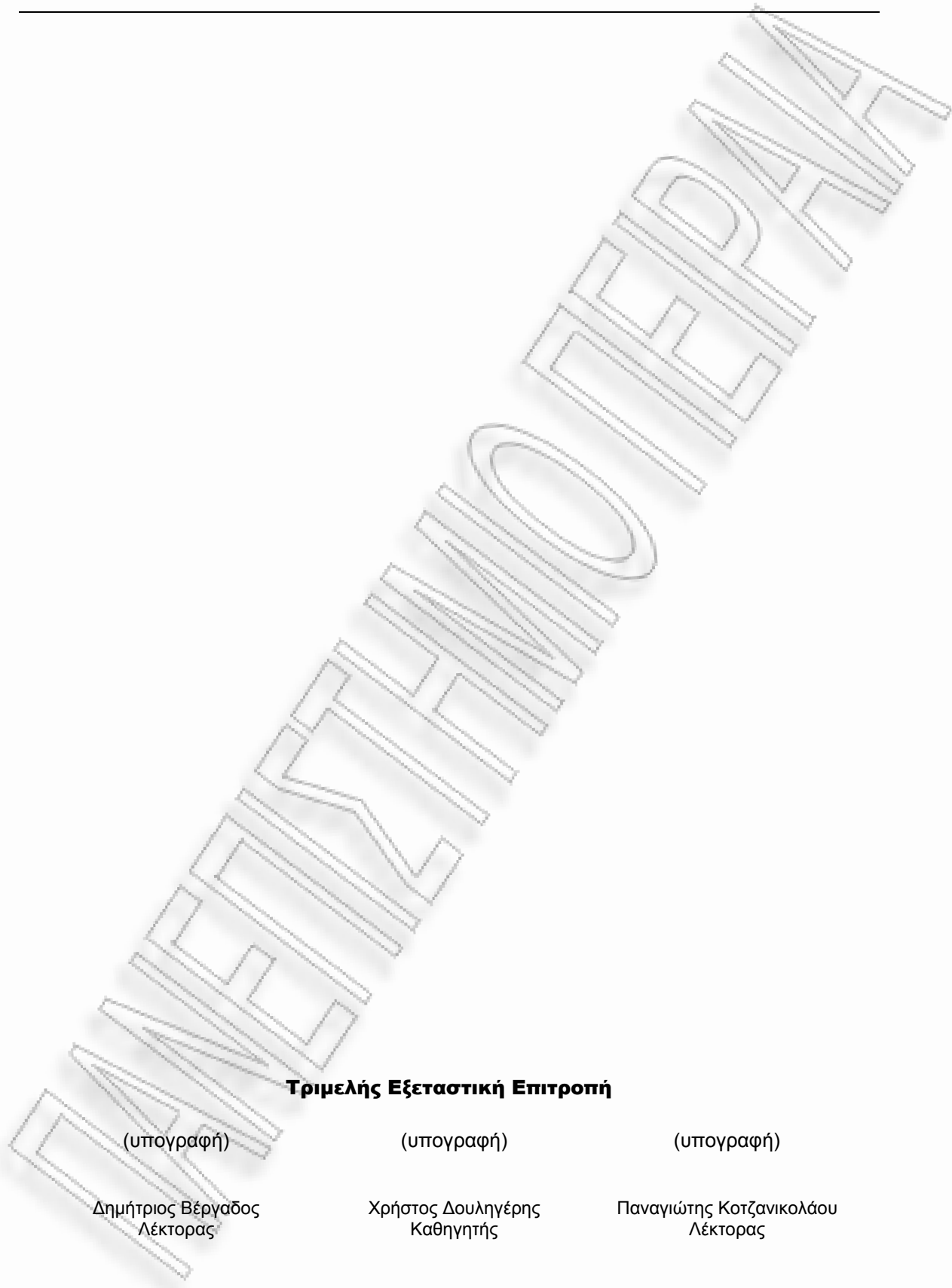




Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής  
Πρόγραμμα Μεταπτυχιακών Σπουδών  
«Προηγμένα Συστήματα Πληροφορικής»

**Μεταπτυχιακή Διατριβή**

Τίτλος Διατριβής	<b>Μελέτη και Αξιολόγηση Επίδοσης Δικτύων WiMAX για την Παροχή QoS</b>
Όνοματεπώνυμο Φοιτητή	<b>Καλυψώ Μαγκλάρα</b>
Πατρώνυμο	<b>Ιωάννης</b>
Αριθμός Μητρώου	<b>ΜΠΣΠ/ 09046</b>
Επιβλέπων	<b>Δημήτριος Δ. Βέργαδος, Λέκτορας</b>



**Τριμελής Εξεταστική Επιτροπή**

(υπογραφή)

Δημήτριος Βέργαδος  
Λέκτορας

(υπογραφή)

Χρήστος Δουληγέρης  
Καθηγητής

(υπογραφή)

Παναγιώτης Κοτζανικολάου  
Λέκτορας

### **Ευχαριστίες**

Θα ήθελα να ευχαριστήσω τον λέκτορα κ. Δημήτριο Δ. Βέργαδο για την επίβλεψη, την υποστήριξη και την συνεργασία που είχαμε για την ολοκλήρωση αυτής της μεταπτυχιακής εργασίας.

Ιδιαίτερες ευχαριστίες θα ήθελα να απευθύνω στη Δρ. Αγγελική Σγώρα για την πολύτιμη βοήθεια, την αμέριστη συμπαράσταση και συνεργασία και τα σχόλιά της που ήταν καίρια για την ολοκλήρωση της διπλωματικής. Επιπλέον, ευχαριστώ τον Δρ. Δημήτρη Ι. Βέργαδο για τις συμβουλές του και την συνεργασία του.

Τέλος, ευχαριστώ την οικογένειά μου για την διαρκή συμπαράσταση και αγάπη τους.

**ΠΕΡΙΕΧΟΜΕΝΑ**

ΠΕΡΙΛΗΨΗ.....	5
ABSTRACT .....	5
1 ΕΙΣΑΓΩΓΗ .....	6
2 ΑΣΥΡΜΑΤΑ ΕΥΡΥΖΩΝΙΚΑ ΔΙΚΤΥΑ - WiMAX.....	7
2.1 ΦΥΣΙΚΟ ΕΠΙΠΕΔΟ.....	9
2.2 MAC ΕΠΙΠΕΔΟ .....	10
2.2.1 ΥΠΟ-ΕΠΙΠΕΔΟ ΣΥΓΚΛΙΣΗΣ.....	11
2.2.2 ΚΟΙΝΟ MAC ΥΠΟ-ΕΠΙΠΕΔΟ.....	11
2.2.3 ΥΠΟ-ΕΠΙΠΕΔΟ ΑΣΦΑΛΕΙΑΣ.....	11
2.2.4 ΠΑΡΟΧΗ ΠΟΙΟΤΗΤΑΣ ΥΠΗΡΕΣΙΩΝ .....	12
2.2.5 ΤΕΧΝΙΚΕΣ ΑΜΦΙΔΡΟΜΗΣΗΣ .....	13
2.2.6 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΔΙΚΤΥΟΥ.....	14
2.2.7 ΚΙΝΗΤΙΚΟΤΗΤΑ ΚΑΙ ΕΞΟΙΚΟΝΟΜΗΣΗ ΕΝΕΡΓΕΙΑΣ.....	14
2.3 ΧΡΗΣΕΙΣ .....	15
2.4 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΑ.....	17
3 ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ.....	19
3.1 ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΤΟ WiMAX.....	19
3.2 ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ .....	22
4 QUALITY OF EXPERIENCE.....	27
4.1 ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ .....	27
4.2 ΠΡΟΤΕΙΝΟΜΕΝΟ ΜΟΝΤΕΛΟ.....	30
4.2.1 ΠΟΛΙΤΙΚΗ 1.....	30
4.2.2 ΠΟΛΙΤΙΚΗ 2.....	31
5 ΠΕΡΙΒΑΛΛΟΝ ΠΡΟΣΟΜΟΙΩΣΗΣ.....	32
5.1 NIST MODULE .....	34
5.1.1 ΥΠΟ-ΕΠΙΠΕΔΟ ΣΥΓΚΛΙΣΗΣ.....	35
5.1.2 MAC ΥΠΟ-ΕΠΙΠΕΔΟ .....	35
5.1.3 ΡΗΥ OFDM.....	39
5.1.4 ΑΡΧΕΙΟ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ (TRACING) .....	41
5.2 WiMAX MODULE ΒΑΣΙΣΜΕΝΟ ΣΤΟ NIST ΜΕ QoS.....	41
6 ΑΞΙΟΛΟΓΗΣΗ ΕΠΙΔΟΣΗΣ.....	43
6.1 ΜΟΝΤΕΛΟ ΠΡΟΣΟΜΟΙΩΣΗΣ.....	43
6.2 ΑΠΟΤΕΛΕΣΜΑΤΑ .....	44
7 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ.....	48
ΒΙΒΛΙΟΓΡΑΦΙΑ .....	49
ΠΑΡΑΡΤΗΜΑ.....	53

## ΠΕΡΙΛΗΨΗ

Τα WiMAX δίκτυα παρέχουν ασύρματη ευρυζωνική πρόσβαση στο διαδίκτυο, διαλειτουργικότητα μεταξύ των προμηθευτών της τεχνολογίας, ενώ παράλληλα χαμηλώνουν το φράγμα εισόδου στις κινητές τηλεπικοινωνίες, προσφέροντας υπηρεσίες συγκρίσιμες με αυτές της ανερχόμενης τεχνολογίας 4G. Ωστόσο το πρότυπο 802.16, στο οποίο βασίζονται τα WiMAX δίκτυα, δεν έχει ορίσει κάποιο συγκεκριμένο αλγόριθμο χρονοπρογραμματισμού, αφήνοντας έτσι ελεύθερο κάθε φορέα που θα χρησιμοποιήσει το πρότυπο για την παροχή υπηρεσιών ευρυζωνικής ασύρματης πρόσβασης στο διαδίκτυο να επιλέξει αλγόριθμο. Γι' αυτό το λόγο έχουν γίνει πολλές μελέτες και έχουν προταθεί από πολλές ομάδες διαφορετικοί αλγόριθμοι χρονοπρογραμματισμού.

Η παρούσα μεταπτυχιακή διατριβή έχει ως σκοπό την μελέτη και ανάπτυξη αλγορίθμων χρονοπρογραμματισμού για την παροχή QoE σε WiMAX δίκτυα. Ειδικότερα στο πλαίσιο εκπόνησης της παρούσας μεταπτυχιακής διατριβής πραγματοποιείται εκτενής βιβλιογραφική μελέτη σε αλγορίθμους χρονοπρογραμματισμού και μηχανισμούς παροχής QoE σε ασύρματα δίκτυα. Από την βιβλιογραφική μελέτη παρατηρήθηκε ότι οι υπάρχοντες αλγόριθμοι χρονοπρογραμματισμού λαμβάνουν υπόψη τους το QoS, τη δικαιοσύνη και άλλες παραμέτρους, αλλά δεν παρέχουν QoE. Για το λόγο αυτό προτείνονται στην παρούσα εργασία δύο διαφορετικές QoE προσεγγίσεις για τον rtPS τύπο υπηρεσίας του WiMAX. Ειδικότερα, στην πρώτη πολιτική ανάλογα με το αν υπάρχει απώλεια των πακέτων στο σύστημα μειώνει το ρυθμό μετάδοσης των συνδέσεων έως ότου φτάσουν στο ελάχιστο επιτρεπτό, δηλαδή την υποκειμενική απαίτηση του χρήστη, ενώ στην δεύτερη πολιτική μπορεί να μειωθεί και πάλι ο ρυθμός μετάδοσης αν η απώλεια της σύνδεσης είναι μεγαλύτερη από ένα κατώφλι ενώ στην αντίθετη περίπτωση μπορεί να αυξηθεί. Τα αποτελέσματα της προσομοίωσης δείχνουν ότι η χρήση διαφορετικών επιπέδων MOS βελτιώνει το QoE που παρέχεται στους χρήστες των WiMAX δικτύων. Ωστόσο, επειδή δεν υπάρχει αλγόριθμος χρονοπρογραμματισμού που να θεωρείται πανάκεια, ανάλογα με τις απαιτήσεις της εφαρμογής και του χρήστη θα πρέπει να εφαρμόζεται και η αντίστοιχη πολιτική.

## ABSTRACT

WiMAX networks provide wireless broadband internet access, interoperability, while decrease the entrance barrier in mobile communications sector, and offer services comparable to those of the emerging 4G technology. However the standard 802.16, upon which WiMAX networks are based, has not designated any particular scheduling algorithm, allowing each provider to develop its own. For this reason several different scheduling algorithms may be found in the literature.

This thesis aims to study and develop scheduling algorithms to provide QoE to WiMAX networks. In particular an extensive literature study is carried out on scheduling algorithms and mechanisms to provide QoE to wireless networks. It was observed that the existing scheduling algorithms take into account the QoS, fairness and other parameters, but do not provide QoE. For this reason, in this thesis two different approaches are proposed in order to provide QoE, especially for the rtPS WiMAX service. In the first policy depending on whether there is packet loss the system reduces the transmission rate of each connection, until their minimum allowable transmission rate is achieved. In the second policy the transmission rate of the connection is reduced if its packet loss is greater than a threshold, while in the opposite case the transmission rate is increased. Simulation results show that by applying different levels of MOS the QoE provided to the WiMAX users is improved. However, since there is no scheduling algorithm that can be considered as a panacea, according to the requirements of each user application a different policy should be applied.

## 1 ΕΙΣΑΓΩΓΗ

Η σύγχρονη πραγματικότητα χαρακτηρίζεται έντονα από την εξέλιξη των διαφορετικών μορφών τεχνολογιών. Οι επιδράσεις στην καθημερινή ζωή, οι οποίες απορρέουν από τις πολλαπλές τεχνολογικές εφαρμογές, συνθέτουν ένα πολυδιάστατο και διαρκώς μεταβαλλόμενο περιβάλλον αναφοράς, εντός του οποίου εμφανίζονται συνεχώς νέες προκλήσεις. Το σημερινό περιβάλλον υψηλής τεχνολογίας αποτελείται από προηγμένες δικτυακές υποδομές, οι οποίες προσφέρουν υψηλούς ρυθμούς μετάδοσης και αξιόπιστη λειτουργία στους χρήστες καθώς και εύκολη δυνατότητα πρόσβασης.

Η διείσδυση των εφαρμογών γίνεται εντονότερη και οι χρήστες τείνουν να συνηθίζουν όσα τους προσφέρονται και να απαιτούν περισσότερα. Η ανάγκη για υψηλότερη παραγωγικότητα και αποδοτικότητα οδήγησαν στην αλματώδη ανάπτυξη των ευρυζωνικών δικτύων. Τα δίκτυα αυτά μπορούν να καλύψουν από τεχνολογικής σκοπιάς απαιτήσεις όπως αδιάλειπτη σύνδεση, υψηλή ταχύτητα και χαμηλό κόστος, παρέχοντας πολύ γρήγορες συνδέσεις στο διαδίκτυο σε όσο το δυνατόν περισσότερους χρήστες υπό τη μορφή καταναλωτικού αγαθού.

Οι δημοφιλέστερες υπάρχουσες τεχνολογίες είναι οι εξής: DSL, 3G, WiFi. Υπάρχουν βέβαια κάποια βασικά μειονεκτήματα όπως το κόστος εγκατάστασης των ενσύρματων δικτύων, οπότε η επέκταση σε νέες περιοχές καθιστά την πρώτη ασύμφορη οικονομικά για τους παρόχους δικτυακής πρόσβασης. Παρόμοια για τη δεύτερη η κοστολόγηση των παρεχόμενων υπηρεσιών είναι ακόμα και σήμερα πολύ υψηλή. Ενώ λόγω της μικρής εμβέλειας κάλυψης που υποστηρίζει η τελευταία, έχει υιοθετηθεί για κατ' οίκον ή εταιρική χρήση μόνο. Για να ξεπεραστούν και αυτά τα εμπόδια, οι έρευνες έχουν στραφεί προς τα δίκτυα 4<sup>η</sup> γενιάς όπως το WiMAX που μελετάται σε αυτή την εργασία και το LTE.

Το WiMAX είναι μια μεταγενέστερη τεχνολογία του 3G που παρέχει ασύρματη μετάδοση πληροφοριών με ποικίλους τρόπους από Point-to-Point έως και με χρήση κυψελωτών δικτύων. Σχεδιάστηκε για να λειτουργεί σε μια ευρεία ζώνη συχνοτήτων από 2 έως 66 GHz, οπότε μπορεί να χρησιμοποιηθεί τόσο για δίκτυα οπτικής επαφής όσο και δίκτυα μη οπτικής επαφής. Η ταχύτητα μπορεί να φτάσει έως και τα 70 Mbps και η κάλυψη έως και τα 50 km [1]. Παρέχει διαλειτουργικότητα, δηλαδή με μία συσκευή μπορεί ο χρήστης να συνδεθεί σε οποιοδήποτε δίκτυο. Επίσης με μικρότερο κόστος εγκατάστασης νέων υποδομών θα είναι περισσότερο ανταγωνιστική η κοστολόγηση των υπηρεσιών από τους παρόχους, παράλληλα θα έχει καλύτερη φασματική απόδοση και με υποστήριξη του IP οι πολυμεσικές εφαρμογές θα είναι κατά πολύ φθηνότερες.

Υπάρχουν υλοποιημένοι πολλοί χρονοπρογραμματιστές με QoS σε WiMAX αλλά δεν παρέχουν QoE (Quality of Experience). Στην παρούσα εργασία γίνεται μελέτη και προσομοίωση ενός WiMAX συστήματος όπου στο policy του χρονοπρογραμματιστή λαμβάνεται υπόψη και το QoE. Το QoE είναι ουσιαστικά η μέτρηση της αποδοτικότητας όχι μόνο σύμφωνα με τεχνικές παραμέτρους αλλά και σύμφωνα με υποκειμενικές παραμέτρους, όπως είναι η αντίληψη των χρηστών για την παρεχόμενη υπηρεσία. Χρησιμοποιώντας χρονοπρογραμματισμό βασισμένο στο QoE, πετυχαίνουμε μείωση της απώλειας πακέτων και της καθυστέρησης μετάδοσής τους από άκρο σε άκρο.

Συγκεκριμένα η δομή που ακολουθείται είναι η ακόλουθη:

- Στο Κεφάλαιο 2, θα παρουσιαστεί αναλυτικά η τεχνολογία WiMAX και οι τεχνολογίες που υποστηρίζει, δίνοντας περισσότερο βάρος στο MAC επίπεδο,
- Στο Κεφάλαιο 3, αφού γίνει αναφορά στα γενικά χαρακτηριστικά του χρονοπρογραμματισμού, θα μελετηθούν διάφορες εργασίες που ασχολούνται με το χρονοπρογραμματισμό στο WiMAX.
- Στο Κεφάλαιο 4, γίνεται καταγραφή του ορισμού του QoE, αλλά και των λίγων εργασιών που έχουν ασχοληθεί με το χρονοπρογραμματισμό που λαμβάνει υπόψη του αυτό, ενώ στο δεύτερο μέρος παρουσιάζεται το προτεινόμενο μοντέλο.
- Στο Κεφάλαιο 5, γίνεται αναφορά στο περιβάλλον προσομοίωσης που χρησιμοποιήθηκε και στα διάφορα modules που υποστηρίζουν το πρότυπο.
- Στο Κεφάλαιο 6, περιγράφονται οι παράμετροι των προσομοιώσεων και τα αποτελέσματα των προσομοιώσεων.
- Τέλος στο Κεφάλαιο 7, καταγράφονται τα συμπεράσματα της εργασίας και προτείνονται μελλοντικά θέματα εργασίας.

## 2 ΑΣΥΡΜΑΤΑ ΕΥΡΥΖΩΝΙΚΑ ΔΙΚΤΥΑ - WIMAX

Το 1998 η IEEE (Institute of Electrical and Electronics Engineers) δημιούργησε μια ομάδα την 802.16 προκειμένου να αναπτύξει ένα πρότυπο για ασύρματη ευρυζωνική πρόσβαση. Τρία χρόνια αργότερα, το 2001, επικυρώθηκε το 802.16, το οποίο προδιαγράφει το φυσικό και το MAC (Media Access Control) επίπεδο. Στο φυσικό επίπεδο προσδιορίζει την χρήση τεχνικών Διαμόρφωσης Μονής Φέρουσας (Single-Carrier Modulation - SCM) ενώ στο MAC επίπεδο χρήση Πολυπλεξίας με Διαίρεση Χρόνου (Time Division Multiplexing - TDM) που υποστηρίζει τόσο Αμφιδρόμηση με Διαίρεση Χρόνου (Time Division Duplexing - TDD) όσο και Αμφιδρόμηση με Διαίρεση Συχνότητας (Frequency Division Duplexing - FDD) και λειτουργία στα 10- 66GHz με δυνατότητα συνδέσεων μόνο με οπτική επαφή (Line-of-sight - LOS) [2].

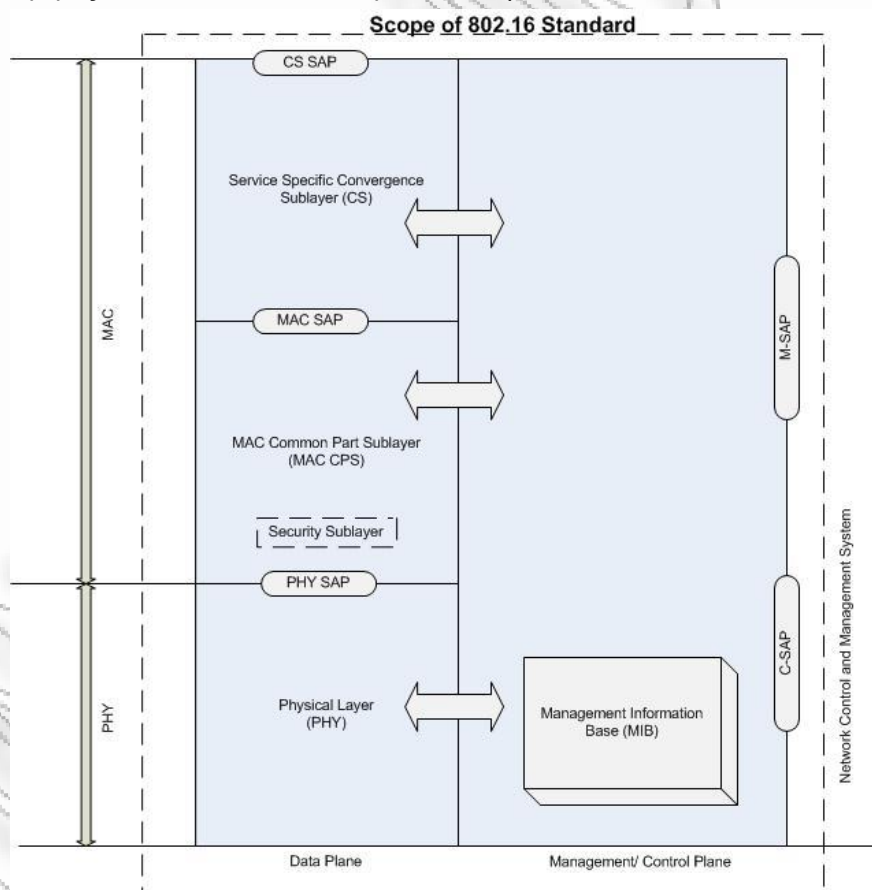
Στη συνέχεια ακολούθησαν προσθήκες και τροποποιήσεις στο πρότυπο ξεκινώντας με το 802.16a-2003 το οποίο καθορίζει τη χρήση Ορθογωνικής Πολυπλεξίας με Διαίρεση Συχνότητας (Orthogonal Frequency Division Modulation - OFDM) και υποστήριξη Πολλαπλής Πρόσβασης με Ορθογωνική Διαίρεση Συχνότητας (Orthogonal Frequency Division Multiple Access - OFDMA) ενώ προβλέπει λειτουργία στις συχνότητες 2- 11 GHz συνδέσεων χωρίς οπτική επαφή (Non-line-of-sight - NLOS) [2], [3]. Το 802.16b συγχωνεύτηκε τελικά με το a και καθόριζε τα ασύρματα δίκτυα από 10 GHz και πάνω [4]. Το 802.16c-2002 όρισε λεπτομερή προφίλ συστήματος σε συχνότητες 10- 66 GHz [5]. Το 802.16d ή 802.16-2004 αποτελεί μια αναθεώρηση και ενσωμάτωση όλων των προαναφερθέντων προτύπων και είναι γνωστό ως Fixed WiMAX [6]. Ακολούθησε το 802.16e-2005, το οποίο επιτρέπει τη χρήση του δικτύου σε χρήστες που κινούνται και είναι γνωστό ως Mobile WiMAX. Το συγκεκριμένο πρότυπο καλύπτει το φυσικό και MAC επίπεδο για συνδυασμένη λειτουργία τόσο κινητή όσο και σταθερή σε συχνότητες που απαιτούν άδεια χρήσης [7]. Το αμέσως επόμενο πρότυπο είναι το 802.16f-2005 το οποίο επιτρέπει mesh δίκτυα και περιέχει την Management Information Base (MIB) [8]. Το 802.16g-2007 αποτελεί τροποποίηση σχετικά με τη διαχείριση του δικτύου (Management Plane Procedures and Services), ενώ το 802.16k-2007 σχετικά με το Bridging [9], [10]. Το 802.16j-2009 αφορά Multihop relay το οποίο αναφέρεται στην έννοια της μεταφοράς δεδομένων του χρήστη και πληροφορίες ελέγχου μεταξύ σταθμών βάσης μέσω ενός ή περισσότερων σταθμών αναμετάδοσης, όπου χρησιμοποιείται αδειοδοτούμενο φάσμα, με στόχο την μεγαλύτερη κάλυψη, το εύρος, την ταχύτητα και ενδεχομένως την συμμετοχή συσκευών χαμηλής ισχύος και το 802.16i περιέχει την Mobile Management Information Base, το περιεχόμενο του οποίου έχει συγχωνευτεί στο P802.16Rev2 [11], [12]. Τέλος, τέσσερις ακόμα τροποποιήσεις είναι οι εξής: 802.16h-2010 το οποίο αφορά βελτιωμένους μηχανισμούς συνύπαρξης για λειτουργίες που απαλλάσσονται από την άδεια χρήσης, το 802.16m βρίσκεται σε εξέλιξη και η ομάδα ασχολείται με προηγμένη διεπαφή για λειτουργία 100 Mbit/s σε κινητό περιβάλλον και 1 Gbit/s σε σταθερό, το 802.16n βρίσκεται επίσης σε εξέλιξη και αφορά μεγαλύτερη αξιοπιστία σε διαταραχθέντα μητροπολιτικά δίκτυα και το 802.16r που αφορά επέκταση για υποστήριξη εφαρμογών μηχανήμα-σε-μηχανήμα, δηλαδή εφαρμογές όπου οι πολλαπλές συσκευές ή συστήματα παρακολουθούνται/ελέγχονται απομακρυσμένα χρησιμοποιώντας ασύρματη ή και ενσύρματη επικοινωνία [13], [14], [15], [16]. Τα πρότυπα που βρίσκονται σε ισχύ σήμερα είναι το 802.16-2009, όπως έχει τροποποιηθεί από τα 802.16j-2009 και 802.16h-2010, το 802.16.2-2004, το 802.16/Conformance04-2006 και το 802.16k-2007 [17], [18], [19]. Ακολουθεί συνοπτικός πίνακας με το 802.16 και τις τροποποιήσεις του, Πίνακας 1.

Πρότυπο/ Τροποποίηση	Περιγραφή
802.16-2001	Αρχικό πρότυπο, σταθερή ευρυζωνική πρόσβαση, 10-66GHz, LOS
802.16a-2003	Χρήση OFDM και OFDMA, συχνότητες 2-11GHz, NLOS
802.16b	Καθορισμός ασύρματων δικτύων πάνω από 10GHz
802.16c-2002	Προφίλ συστήματος σε συχνότητες 10- 66 GHz
802.16d/ 802.16-2004/ Fixed WiMAX	Αναθεώρηση και ενσωμάτωση όλων των προαναφερθέντων
802.16e-2005/ Mobile	Επιτρέπει κινητικότητα στους χρήστες

Πρότυπο/ Τροποποίηση	Περιγραφή
WiMAX	
802.16f-2005	Επιτρέπει mesh δίκτυα και περιέχει την Management Information Base
802.16g-2007	Management Plane Procedures and Services
802.16k-2007	Bridging στο 802.16
802.16j-2009	Multihop relay
802.16i	Mobile Management Information Base
802.16h-2010	Βελτιωμένοι μηχανισμοί συνύπαρξης για λειτουργίες που απαλλάσσονται από την άδεια χρήσης
802.16m	Λειτουργία 100 Mbit/s σε κινητό περιβάλλον και 1 Gbit/s σε σταθερό
802.16n	Μεγαλύτερη αξιοπιστία σε μητροπολιτικά δίκτυα
802.16p	Υποστήριξη εφαρμογών μηχανήμα-σε-μηχάνημα

**Πίνακας 1 – Πρότυπο 802.16 και οι τροποποιήσεις του**

Μια οντότητα IEEE 802.16 είναι μια λογική οντότητα στο σταθμό βάσης ή στο κινητό σταθμό του χρήστη, η οποία περιλαμβάνει το φυσικό και MAC επίπεδο των διαστάσεων Μεταφοράς Δεδομένων (Data Plane), Ελέγχου (Control Plane) και Διαχείρισης (Management Plane) [19]. Το μοντέλο αναφοράς του επιπέδου δικτύου φαίνεται στην Εικόνα 2.1.



**Εικόνα 2.1 – Μοντέλο Αναφοράς του 802.16 [19]**

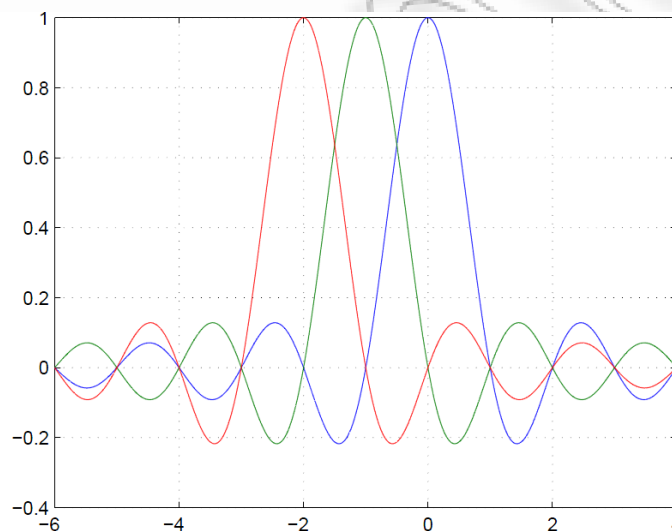
Στη συνέχεια θα αναλυθούν με περισσότερη λεπτομέρεια τα δύο επίπεδα που ορίζονται από το WiMAX.



## 2.1 ΦΥΣΙΚΟ ΕΠΙΠΕΔΟ

Το πρότυπο υποστηρίζει τέσσερις διαφορετικές υλοποιήσεις για το φυσικό στρώμα. Στις υλοποιήσεις WirelessMAN-OFDM και WirelessMAN-OFDMA χρησιμοποιούνται αντίστοιχα οι διαμορφώσεις OFDM και OFDMA ενώ μπορεί να χρησιμοποιηθεί και η διαμόρφωση μονής φέρουσας με υλοποιήσεις τις WirelessMAN SC και WirelessMAN SCa [2], [20], [21].

Το φυσικό επίπεδο βασίζεται στην OFDM, η οποία είναι μια αξιόπιστη τεχνική μετάδοσης ακόμα και σε διαύλους με έντονη την παρουσία της πολυδιαδρομικής διάδοσης. Παρόλο που είναι μια τεχνική πολυπλεξίας χρησιμοποιείται ως διαμόρφωση καθώς όλες οι διαθέσιμες φέρουσες κατανέμονται σε ένα μόνο χρήστη και για αυτό το λόγο η OFDM είναι επίσης γνωστή ως διαμόρφωση πολλών φερουσών (Multiple- Carrier Modulation – MCM). Στην τεχνική αυτή γίνεται διαίρεση του σήματος σε μεγάλο αριθμό παράλληλων σημάτων, τα οποία χωρίζονται σε τακτά διαστήματα, έτσι ώστε οι φέρουσες να είναι ορθογωνικές μεταξύ τους. Το αποτέλεσμα είναι να μην παρεμβάλλει το περιεχόμενο της μιας στην άλλη, παρόλη την επικάλυψη των συχνοτήτων της μια φέρουσας με την άλλη [2], [21]. Σε μια OFDM μετάδοση τα σήματα των σταθμών που περιέχουν τα δεδομένα συνδυάζονται και έτσι προκύπτει μια πολυπλεγμένη ακολουθία δεδομένων, τα οποία στη συνέχεια μεταδίδονται όλα μαζί διαμορφώνοντας ένα σύνολο από φέρουσες, όπως φαίνεται και στην Εικόνα 2.2.



Εικόνα 2.2 – Το σύμβολο OFDM στο πεδίο της συχνότητας για τρεις φέρουσες [22].

Με τον συγχρονισμό των φερουσών τόσο χρονικά όσο και ως προς την συχνότητα δεν μειώνεται μόνο η παρεμβολή αλλά και τα διαστήματα ασφαλείας, γεγονός που βελτιώνει την φασματική απόδοση. Κάθε διάυλος αντιστοιχεί σε μια φέρουσα. Οι φέρουσες που αποτελούν την OFDM μετάδοση αναφέρονται και ως υποφέρουσες, ενώ ο όρος φέρουσα είναι πλέον η φέρουσα συχνότητα με την οποία γίνεται ο συνδυασμός του σήματος βασικής ζώνης. Η OFDM βασίζεται σε σύμβολα, η συχνότητα κάθε υποφέρουσας στη βασική ζώνη επιλέγεται να είναι ακέραιο πολλαπλάσιο του αντιστρόφου της διάρκειας συμβόλου, έχοντας ως αποτέλεσμα όλες οι υποφέρουσες να έχουν ένα ακέραιο αριθμό κύκλων ανά σύμβολο. Αρχικά γίνεται μετατροπή των σειριακών δεδομένων σε παράλληλα, ακολουθεί η διαμόρφωση (16- QAM, 64- QAM, QPSK) όπου κάθε υποφέρουσα μπορεί να διαμορφωθεί διαφορετικά. Στη συνέχεια χρησιμοποιείται ο Αντίστροφος Γρήγορος Μετασχηματισμός Fourier (IFFT) για να μετατρέψει το σήμα από το πεδίο συχνότητας στο πεδίο του χρόνου και να αποτρέψει την μετάδοσή του. Τέλος, εισάγονται τα διαστήματα ασφαλείας και μετατρέπονται τα παράλληλα δεδομένα σε σειριακά για να μεταδοθούν. Στον παραλήπτη ακολουθεί η αντίστροφη διαδικασία [2], [20], [21]. Για την επιλογή των παραμέτρων αρχικά ορίζεται η διάρκεια του διαστήματος ασφαλείας που πρέπει να είναι τουλάχιστον ίση με τη μέγιστη χρονική διασπορά του διαύλου και έπειτα η διάρκεια του OFDM συμβόλου τουλάχιστον 4 φορές μεγαλύτερη από το διάστημα ασφαλείας [21].

Στην OFDMA η διαφοροποίηση των χρηστών μπορεί να γίνει είτε στο χρόνο είτε στη συχνότητα είτε και στα δύο. Επιπροσθέτως, το Scalable- OFDMA, που είναι μια μορφή OFDMA, επιτρέπει την εκχώρηση διαφορετικού εύρους καναλιών από 1.25MHz έως 20MHz, μεταβάλλοντας δυναμικά το μέγεθος του IFFT ανάλογα με το εύρος ζώνης που διατίθεται [21], [23].

Επίσης, στο WiMAX χρησιμοποιείται υποκαναλοποίηση δηλαδή οι φέροντες κατανομούνται σε μια ή περισσότερες ζεύξεις. Ένα υποκανάλι είναι ένα σύνολο φερουσών, των οποίων το πλήθος και η διάταξη καθορίζονται από τον τρόπο αντιμετάθεσης των υποφερουσών (sub-carrier permutation mode). Στο mobile WiMAX μπορεί να γίνει και στις δύο κατευθύνσεις ενώ στο fixed μόνο στην ανερχόμενη ζεύξη [21].

Για την αντιμετώπιση της ανάγκης για αύξηση της χωρητικότητας του διαύλου, την ασφάλεια των επικοινωνιών και την βελτιστοποίηση της ποιότητας των παρεχόμενων υπηρεσιών το πρότυπο WiMAX υποστηρίζει την τεχνολογία Εξελιγμένων Συστημάτων Κεραίων (Advanced Antenna Systems - AAS) που συνδυάζεται με την τεχνολογία MIMO (Multiple-Input-Multiple-Output). Η MIMO αυξάνει την φασματική απόδοση Point-to-Point χρησιμοποιώντας πολλαπλές κεραίες και αλυσίδες συχνοτήτων τόσο στο σταθμό όσο και στο χρήστη [23]. Τα MIMO συστήματα εκμεταλλεύονται το φαινόμενο της πολυδιαδρομικής διάδοσης για να μεταδώσουν ταυτόχρονα πολλές ροές δεδομένων από διαφορετικές κεραίες χωρίς να απαιτούν επιπλέον εύρος ζώνης [24].

Το WiMAX υποστηρίζει μια ποικιλία σχημάτων διαμόρφωσης και κωδικοποίησης όπως προαναφέρθηκε και επιτρέπει την εναλλαγή των σχημάτων αυτών ανά ριπή πακέτων σε κάθε σύνδεση. Η Προσαρμοστική Διαμόρφωση και Κωδικοποίηση (Adaptive Coding and Modulation - AMC) είναι μια τεχνική που συνδυάζει τη διαμόρφωση υψηλότερης τάξης με την προσαρμογή καναλιού για την βελτιστοποίηση της στιγμιαίας χρήσης του εξασθενίζοντας καναλιού. Έτσι επιλέγεται το κατάλληλο σχήμα διαμόρφωσης και κωδικοποίησης, σύμφωνα με τις συνθήκες που επικρατούν στο κανάλι, δηλαδή υπο ευνοϊκές συνθήκες επιλέγεται η χρήση της φασματικά αποδοτικότερης υψηλότερης τάξης διαμόρφωση 16 QAM (Quaternary Amplitude Modulation με 16 καταστάσεις), ενώ για λιγότερο ευνοϊκές εφαρμόζεται η συμβατική και πιο δυνατή διαμόρφωση QPSK (Quaternary Phase Shift Keying) [21], [23], [25].

Για την αξιόπιστη μετάδοση των ψηφιακών δεδομένων χρησιμοποιούνται μέθοδοι ελέγχου και διόρθωσης σφαλμάτων. Η τεχνική αυτόματων αιτήσεων επαναμετάδοσης (Automatic Repeat reQuest - ARQ) είναι μια από της μεθόδους αυτές που χρησιμοποιεί την προσθήκη πλεονάζουσας πληροφορίας στο σήμα. Ο πομπός εισάγει επιπλέον πληροφορία στα αρχικά μήνυμα, δημιουργώντας έτσι μια κωδικοποιημένη λέξη και αντίστροφα ο δέκτης αφαιρεί την πλεονάζουσα πληροφορία και στη συνέχεια τη συγκρίνει με το περιεχόμενο του μηνύματος. Τα σφάλματα ενός πλαισίου ξεετάζονται στο παραλήπτη από έναν κώδικα ανίχνευσης λάθους, συνήθως τον κυκλικό έλεγχο πλεονασμού (CRC). Εάν ο έλεγχος επιβεβαιώσει το πλαίσιο τότε στέλνει μια βεβαίωση (acknowledgement - ACK) της επιτυχούς μετάδοσης προς το δέκτη. Εάν ο παραλήπτης ανιχνεύσει ότι το πλαίσιο έχει αποκωδικοποιηθεί λάθος, στέλνει μια αρνητική βεβαίωση και αίτημα για αναμετάδοση του πλαισίου μέχρι την επιτυχία του αποκωδικοποίηση ή την συμπλήρωση ενός προκαθορισμένου πλήθους αποτυχημένων μεταδόσεων [21]. Τα βασικά συστήματα ARQ είναι η stop-and-wait, go-back-N και selective repeat protocols. Για μεγαλύτερη αξιοπιστία η ARQ εφαρμόζεται σε συνδυασμό με την εμπροσθόδοτη διόρθωση σφάλματος (forward error correction - FEC) [26]. Μια παραλλαγή της ARQ είναι η Hybrid-ARQ, η διαφορά τους είναι ότι η δεύτερη χρησιμοποιεί όλα τα αντίγραφα που έχει λάβει κατά την αποκωδικοποίηση ενός πακέτου [23].

Επίσης το επίπεδο PHY είναι υπεύθυνο για τις διαδικασίες συγχρονισμού (ranging), ελέγχου ισχύος και κατά επέκταση εξοικονόμησης ενέργειας [2], [21].

## 2.2 MAC ΕΠΙΠΕΔΟ

Ο κύριος ρόλος του MAC επιπέδου είναι να παρέχει μια διεπαφή μεταξύ του φυσικού επιπέδου και των ανώτερων επιπέδων μεταφοράς. Το επίπεδο αυτό επιλέγει το κατάλληλο επίπεδο ισχύος και προφίλ εκπομπής για την μετάδοση των πακέτων, κάνει αναμετάδοση όταν είναι απαραίτητο, είναι υπεύθυνο για την ποιότητα των παρεχόμενων υπηρεσιών στις συνδέσεις, δηλαδή τον χρονοπρογραμματισμό και την τήρηση της προτεραιότητας, ενώ τέλος υποστηρίζει διαδικασίες κινητικότητας, ασφάλειας και εξοικονόμησης ενέργειας. Στο MAC επίπεδο υπάρχουν

τρία υπο-επίπεδα: το υπο-επίπεδο σύγκλισης (Convergence Sublayer – CS), το κοινό MAC υπο-επίπεδο (Common Part MAC Sublayer – CPS) και το υπο-επίπεδο ασφάλειας (Security Sublayer) [19].

### 2.2.1 ΥΠΟ-ΕΠΙΠΕΔΟ ΣΥΓΚΛΙΣΗΣ

Το υπο-επίπεδο σύγκλισης είναι το υψηλότερο από τα τρία και είναι υπεύθυνο για την λήψη δεδομένων από υψηλότερα επίπεδα. Το MAC λαμβάνει πακέτα από τα σημεία πρόσβασης υπηρεσίας (Service Access Point – SAP), τα οποία ονομάζονται Service Data Units (SDU), τα μετατρέπει σε Protocol Data Units (PDU), τα οποία αποτελούνται από μια επικεφαλίδα (header), ένα χρήσιμο τμήμα (payload) και προαιρετικά ένα τμήμα ελέγχου σφαλμάτων (Cyclic Redundancy Check - CRC). Στη συνέχεια τα ταξινομεί και τα συσχετίζει με τις κατάλληλες συνδέσεις του MAC σύμφωνα με τα αναγνωριστικά των συνδέσεων (connection identifier – CID) τους, μπορεί ακόμα και να τα επεξεργαστεί εφαρμόζοντας τεχνικές συμπίεσης της επικεφαλίδας (Payload Suppression Header – PHS), προτού μεταδοθούν από το φυσικό επίπεδο. Το ίδιο πραγματοποιεί και την αντίστροφη διαδικασία στον αποδέκτη [21], [23]. Η δομή και το μέγεθος των PDU μπορεί να αλλάζει, ενώ μπορούν πολλά PDU να συνενώνονται κάτω από μια κοινή επικεφαλίδα, ανάλογα με το μέγεθος ενός SDU. Αν χρειάζεται διάσπαση και υποστηρίζεται η ARQ τότε κάθε τμήμα λαμβάνει έναν αριθμό και στο τέλος επιβεβαιώνεται η λήψη όλων, διαφορετικά εκπέμπονται με τη σειρά που βρίσκονται στο αρχικό SDU.

Υπάρχουν δύο είδη PDU με διαφορετική δομή επικεφαλίδας:

- Γενικής μορφής, το οποίο μεταφέρει δεδομένα και πληροφορίες σηματοδότησης του επιπέδου, όπως εάν έχουν κρυπτογραφηθεί δεδομένα, το κλειδί, αν υπάρχει τμήμα CRC, το μήκος του πακέτου, το CID και τμήμα για την επαλήθευση των περιεχομένων της.
- Αίτηση εύρους ζώνης, το οποίο δεν έχει ούτε χρήσιμο τμήμα ούτε και CRC.

Στα γενικής μορφής PDU υποστηρίζεται η ύπαρξη υπο-επικεφαλίδων που αφορά η μία δίκτυα Mesh, η δεύτερη ακολουθεί την επικεφαλίδα γενικής μορφής και δείχνει τεμαχισμό των SDU σε πολλά PDU, η τρίτη δείχνει αν το πακέτο περιλαμβάνει πολλές SDU και η τέταρτη χρησιμοποιείται όταν εφαρμόζεται η τεχνολογία MIMO.

### 2.2.2 ΚΟΙΝΟ MAC ΥΠΟ-ΕΠΙΠΕΔΟ

Το κοινό MAC υπο-επίπεδο είναι υπεύθυνο για την πρόσβαση στο δίαυλο, τις αιτήσεις επαναμετάδοσης, την διαχείριση των συνδέσεων και την εφαρμογή της Παροχής Ποιότητας Υπηρεσιών (Quality of Service – QoS). Επίσης δημιουργεί τις MAC επικεφαλίδες (Protocol Control Information) και παραδίδει τα PDU στα ομότιμα υπο-επίπεδα σύμφωνα με το QoS της συγκεκριμένης Ροής Υπηρεσίας (Service Flow) [23]. Το MAC είναι connection-oriented, κάθε συσκευή χρήστη έχει μια αναγνωριστική διεύθυνση MAC και κάθε σύνδεση ένα CID. Όλες οι συνδέσεις για μεταφορά ή έλεγχο είναι μονής κατεύθυνσης, δηλαδή ο Σταθμός Βάσης προγραμματίζει την κατερχόμενη και ανερχόμενη ζεύξη ώστε να ανταποκρίνονται στις απαιτήσεις Παροχής Ποιότητας Υπηρεσιών, τα οποία αναλύονται στη συνέχεια.

Ανάμεσα στις λειτουργίες αυτού του υπο-επιπέδου είναι και ο έλεγχος του φυσικού επιπέδου.

### 2.2.3 ΥΠΟ-ΕΠΙΠΕΔΟ ΑΣΦΑΛΕΙΑΣ

Τέλος, υπάρχει και το υπο-επίπεδο ασφάλειας, το οποίο προσφέρει έλεγχο πρόσβασης και εμπιστευτικότητα, δηλαδή αυθεντικοποίηση χρηστών, ασφαλή ανταλλαγή κλειδιών και κρυπτογραφία. Όταν ένας χρήστης συνδέεται με ένα σταθμό η χειραψία μεταξύ των δύο οντοτήτων βασίζεται στους συσχετισμούς ασφάλειας (Security Associations – SA), οι οποίοι αναπαριστούν την συμφωνία μεταξύ των οντοτήτων, τα κρυπτογραφικά κλειδιά. Το WiMAX βασίζεται στο Privacy Key Management (PKMv2) για την αυθεντικοποίηση τόσο του χρήστη όσο και του σταθμού. Για την αυθεντικοποίηση το 802.16 χρησιμοποιεί το Επεκτάσιμο Πρωτόκολλο Ελέγχου Ταυτότητας (Extensible Authentication Protocol – EAP), το οποίο με τη σειρά του χρησιμοποιεί μεθόδους όπως το Transport Layer Security (TLS). Στη συνέχεια τα δεδομένα που ανταλλάσσονται κρυπτογραφούνται με τη βοήθεια του Προηγμένου Πρότυπου Κρυπτογράφησης (Advanced Encryption Standard – AES) [27], [28].

### 2.2.4 ΠΑΡΟΧΗ ΠΟΙΟΤΗΤΑΣ ΥΠΗΡΕΣΙΩΝ

Το πρότυπο περιγράφει τέσσερις διαφορετικούς τύπους υπηρεσιών, ενώ στην τροποποίηση για υποστήριξη της κινητικότητας προστίθεται ένας ακόμα τύπος ως εξής:

- Σταθερού Ρυθμού Μετάδοσης (Unsolicited Grant Services – UGS), σε αυτόν τον τύπο υπηρεσιών κατατάσσονται οι εφαρμογές πραγματικού χρόνου που χρησιμοποιούν σε περιοδικά διαστήματα πακέτα σταθερού μήκους, όπως το VoIP. Όταν ο σταθμός κάνει κατανομή του εύρους ζώνης για τη δημιουργία μιας σύνδεσης που ανήκει σε αυτή την κατηγορία τότε ο χρήστης διατηρεί αυτό το εύρος ζώνης μέχρι τον τερματισμό της σύνδεσης. Βέβαια ο χρήστης μπορεί να χρησιμοποιήσει αυτή την σύνδεση για να κάνει αίτηση για άλλες υπηρεσίες οπότε μπορεί να αλλάξει η κλάση και κατά συνέπεια το εύρος ζώνης που έχει κατανεμηθεί.
- Μεταβαλλόμενου Ρυθμού Υψηλής Διαθεσιμότητας (Real-Time Polling Services – rt-PS), σε αυτή τη κλάση ανήκουν οι εφαρμογές πραγματικού ρυθμού που σε τακτικά και πάλι διαστήματα στέλνουν πακέτα μεταβλητού μήκους, ένα παράδειγμα εδώ είναι τα MPEG video. Σε αυτή την περίπτωση ο χρήστης κάνει αίτηση για εύρος ζώνης είτε με μηχανισμούς μονοεκπομπής είτε με riggypacking και στη συνέχεια κατανέμεται εύρος ζώνης το οποίο χρησιμοποιεί.
- Μεταβαλλόμενου Ρυθμού Χαμηλής Διαθεσιμότητας (Non-Real-Time Polling Services – nr-t-PS), αυτός ο τύπος είναι για εφαρμογές μη-πραγματικού χρόνου με πακέτα μεταβλητού μήκους. Η αίτηση γίνεται όπως και πριν με μόνη διαφορά ότι εκτός από μηχανισμούς μονοεκπομπής χρησιμοποιούνται και ευρυεκπομπής, όπου και σε αυτή την περίπτωση υπάρχει αυξημένη πιθανότητα συγκρούσεων οπότε εφαρμόζονται διαδικασίες εκθετικής οπισθοχώρησης (Exponential Backoff Process).
- Διαθέσιμου Εύρους (Best Effort), σε αυτή την τελευταία κατηγορία δεν υπάρχουν περιορισμοί QoS. Χρησιμοποιούνται όλοι οι παραπάνω μηχανισμοί όμως οι σταθμοί βάσεις όχι απλά δεν δίνουν προτεραιότητα αλλά ακόμα και παραμελούν αυτές τις συνδέσεις [17], [29].
- Μεταβαλλόμενου Ρυθμού Εκτεταμένης Υψηλής Διαθεσιμότητας (Extended Real-Time Polling Services – ert-PS), αποτελεί ενδιάμεσο του UGS και του rt-PS. Ένα παράδειγμα υπηρεσίας είναι το VoIP με καταστολή σιωπής (Silence Suppression). Χρησιμοποιεί μηχανισμό παρόμοιο με του UGS, με μόνη εξαίρεση ότι περιοδικά οι εκχωρήσεις (grants) μπορούν να χρησιμοποιηθούν για να σταλεί αίτηση για εύρος ζώνης έτσι ώστε να ενημερώσουν για το απαιτούμενο μέγεθος εκχωρήσεων και άρα το εύρος ζώνης μπορεί δυναμικά να αλλάζει όπως στο rt-PS [30].

Η σύνδεση περιγράφεται από τις παραμέτρους QoS, δηλαδή ανήκει σε μία ροή υπηρεσίας. Η ροή υπηρεσίας είναι μια ροή πακέτων, η οποία καθορίζεται από ένα σύνολο QoS παραμέτρων, οι οποίες ποσοτικοποιούν την συμπεριφορά της. Η διαχείριση αυτών των παραμέτρων μπορεί να γίνει χρησιμοποιώντας μηνύματα DSA και DSC [19]. Οι παράμετροι QoS είναι οι ακόλουθοι: η ανεκτή τιμή jitter (Tolerated Jitter), η μέγιστη καθυστέρηση (Maximum Latency), ο μέγιστος σταθερός ρυθμός μετάδοσης (Maximum Sustained Traffic Rate), ο ελάχιστος δεσμευμένος ρυθμός μετάδοσης (Minimum Reserved Traffic Rate), η προτεραιότητα κίνησης (Traffic Priority) και η πολιτική αίτησης/ μετάδοσης (Request/ Transmission Policy). Με τον συγκεκριμένο καθορισμό των τιμών των παραμέτρων μπορούν να υλοποιηθούν διαφορετικές υπηρεσίες χρονοπρογραμματισμού. Στη συνέχεια περιγράφονται αναλυτικότερα οι παραπάνω παράμετροι [29].

- Μέγιστος σταθερός ρυθμός μετάδοσης: Αντιπροσωπεύει τις αιχμές του ρυθμού μετάδοσης των πληροφοριών κάποιας υπηρεσίας. Ο ρυθμός μετράται σε bits ανά δευτερόλεπτο και αφορά τα SDUs στην είσοδο του συστήματος. Πιο συγκεκριμένα, η τιμή της παραμέτρου δεν περιλαμβάνει τις μεταφορές, το πρωτόκολλο ή το φόρτο του δικτύου, όπως οι MAC επικεφαλίδες, οι Κυκλικό Έλεγχος Πλεονασμού (Cyclic Redundancy Check - CRC) ή το μη ωφέλιμο φορτίο συντήρησης σύνοδου, όπως η διαχείριση του Πρωτοκόλλου Εκκίνησης Συνόδου (Session Initiation Protocol - SIP). Η υπηρεσία θα πρέπει να επιβλέπεται ώστε να τηρείται η παράμετρος κατά μέσο όρο στην πάροδο του χρόνου στην ανερχόμενη ζεύξη, ενώ στην κατερχόμενη ζεύξη τηρείται κατά την είσοδο στο δίκτυο. Εάν στην παράμετρο δοθεί η τιμή μηδέν τότε δεν ισχύει κάποιος μέγιστος ρυθμός. Επίσης αναφέρεται ότι η παράμετρος ορίζει αποκλειστικά και μόνο ένα άνω φράγμα και όχι εγγύηση για την παροχή αυτού του ρυθμού.

- Ελάχιστος δεσμευμένος ρυθμός μετάδοσης: Αντιπροσωπεύει τον ελάχιστο ρυθμό μετάδοσης και μετράται σε bits ανά δευτερόλεπτο που θα πρέπει να παρέχεται στην ροή υπηρεσίας. Εάν κάποιος σταθμός συνδρομητή (Subscriber Station – SS ή Mobile Station – MS) κάνει αιτήσεις για μικρότερο ρυθμό από την τιμή της παραμέτρου, ο σταθμό βάσης (Base Station – BS) μπορεί να το χρησιμοποιήσει για κάτι άλλο. Ωστόσο, ο BS πρέπει να διαθέσει τον ελάχιστο ρυθμό στη σύνδεση. Η τιμή καθορίζεται μετά τον αποκλεισμό του MAC φόρτου (overhead). Εάν η τιμή της παραμέτρου είναι μηδέν, δεν απαιτείται ελάχιστος ρυθμός.
- Μέγιστη καθυστέρηση: Δείχνει τη μέγιστη τιμή του χρόνου μεταξύ της παραλαβής ενός πακέτου στο υπο-επίπεδο σύγκλισης του BS ή του SS και της άφιξής του στη συσκευή του παραλήπτη. Η μηδενική τιμή αυτής της παραμέτρου σημαίνει ότι δεν υπάρχει καμία δέσμευση.
- Ανεκτή τιμή jitter: Δείχνει τη μέγιστη διακύμανση των στιγμιαίων αποκλίσεων καθοριστικών τμημάτων ενός ψηφιακού σήματος σε σχέση με τις ιδανικές θέσεις του στον χρόνο για τη σύνδεση.
- Πολιτική αίτησης/ μετάδοσης: Είναι η ικανότητα ορισμού κάποιων χαρακτηριστικών για τη ροή υπηρεσίας. Παραδείγματα τέτοιων χαρακτηριστικών για τις ροές υπηρεσιών ανερχόμενης ζεύξης είναι οι περιορισμοί στους τύπους των επιλογών αίτησης εύρους ζώνης που μπορούν να χρησιμοποιηθούν. Ένα άλλο παράδειγμα είναι οι επιλογές για το σχηματισμό PDU. Ένα χαρακτηριστικό είναι ενεργοποιημένο όταν η σημαία του bit είναι ένα, ενώ αν είναι μηδέν χρησιμοποιούνται προεπιλεγμένες ενέργειες.
- Προτεραιότητα κίνησης: Η τιμή της παραμέτρου αντιπροσωπεύει την προτεραιότητα που δίνεται σε μια ροή υπηρεσίας. Όταν δύο ροές έχουν τις ίδιες τιμές σε όλες τις παραμέτρους QoS, εκτός από την προτεραιότητα, η ροή υπηρεσιών με υψηλότερη προτεραιότητα θα πρέπει να εξυπηρετηθεί νωρίτερα. Σε οποιαδήποτε άλλη περίπτωση οι υπόλοιποι παράμετροι QoS έχουν μεγαλύτερη σημασία. Οι τιμές προτεραιότητας κυμαίνονται από 0 έως 7 και η προκαθορισμένη τιμή είναι 0 [19], [29].

## 2.2.5 ΤΕΧΝΙΚΕΣ ΑΜΦΙΔΡΟΜΗΣΗΣ

Δύο τεχνικές αμφιδρόμησης υποστηρίζονται από το πρωτόκολλο MAC. Η επιλογή τεχνικής αμφιδρόμησης μπορεί επηρεάσει ορισμένες παραμέτρους του φυσικού επιπέδου καθώς επίσης και να έχει επιπτώσεις στα χαρακτηριστικά που μπορούν να υποστηριχθούν [17]. Η επικρατέστερη τεχνική είναι η Αμφιδρόμηση με Διάρθρωση Χρόνου (TDD), αφού επιτρέπει πιο ευέλικτο διαμοιρασμό του εύρους ζώνης μεταξύ ανερχόμενης (uplink - UL) και κατερχόμενης (downlink - DL) ζεύξης, ωστόσο υπάρχουν περιπτώσεις που καλύτερη αποδεικνύεται η Αμφιδρόμηση με Διάρθρωση Συχνότητας (FDD) [2]. Σε ένα σύστημα TDD οι μεταδόσεις UL και DL λαμβάνουν χώρα σε διαφορετικές χρονικές στιγμές και συνήθως μοιράζονται την ίδια συχνότητα. Όλα τα πλαίσια του επιπέδου ξεκινούν με μία επικεφαλίδα που περιέχει το CID, το μέγεθος του και πληροφορίες σχετικά με την κρυπτογράφηση των δεδομένων. Ένα TDD πλαίσιο έχει συγκεκριμένη διάρκεια και περιέχει ένα UL και ένα DL υπο-πλαίσιο. Κάθε πλαίσιο χωρίζεται σε ένα ακέραιο αριθμό Φυσικών Σχισμών (Physical Slots – PS), όπου σχισμή είναι ο ελάχιστος χρόνος του βασικού εύρους συχνότητας που δίνεται σε μια σύνδεση. Το γεγονός αυτό κάνει το εύρος ζώνης προσαρμόσιμο ανάμεσα σε UL και DL, ο διαμοιρασμός είναι παράμετρος του συστήματος και ελέγχεται από ανώτερα επίπεδά του, μπορούν να έχουν ίση διάρκεια ή το DL να είναι έως και τρεις φορές μεγαλύτερο. Σε ένα σύστημα FDD τα κανάλια UL και DL βρίσκονται σε διαφορετικές συχνότητες και τα δεδομένα του DL μπορούν να μεταδοθούν σε ριπές. Και πάλι η διάρκεια των πλαισίων είναι συγκεκριμένη. Σε αυτό το σύστημα είναι δυνατή η χρήση διαφορετικών διαμορφώσεων και υποστηρίζεται τόσο η χρήση full-duplex όσο και half-duplex, δηλαδή η αποστολή και η παραλαβή να γίνονται ταυτόχρονα ή ξεχωριστά αντιστοίχως [2], [19], [21].

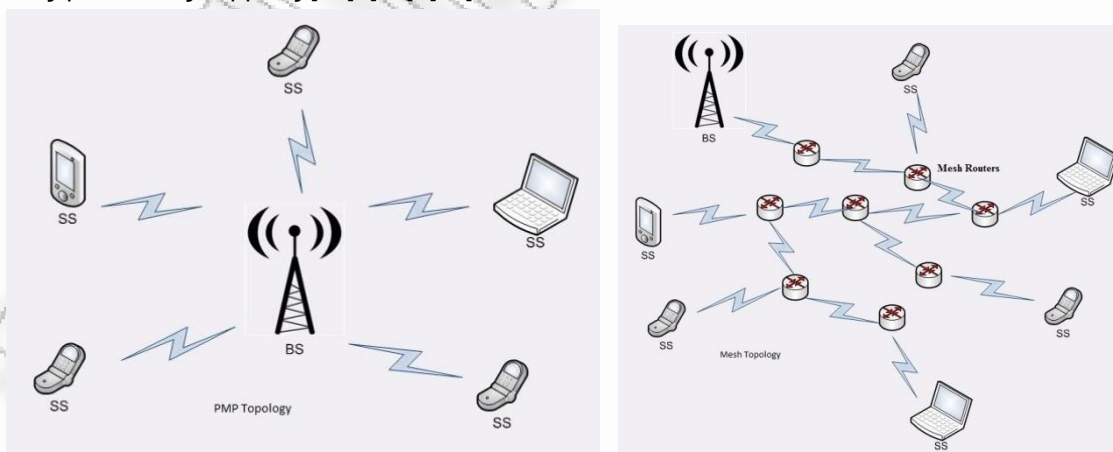
Το DL υποπλαίσιο αποτελείται από μια επικεφαλίδα (preamble), που περιέχει μεταξύ άλλων πληροφορίες συγχρονισμού και την αρχική εκτίμηση του διαύλου, μια επικεφαλίδα ελέγχου του πλαισίου (frame control header), που περιέχει πληροφορίες για τη δομή και τη διάρθρωση του πλαισίου καθώς και ριπές διαφορετικού μεγέθους, διαφορετικών χρηστών και διαφορετικών πληροφοριών.

Το UL υποπλαίσιο αποτελείται από ριπές, οι οποίες αντιστοιχούν στους χρήστες, ένα τμήμα για ganging, που ρυθμίζει τον χρονισμό της συχνότητας και της ισχύος, ένα τμήμα για την

ποιότητα του διαύλου όπως έχουν δοθεί από τον SS και τέλος ένα τμήμα επιβεβαίωσης για τη λήψη μηνυμάτων από την κατερχόμενη ζεύξη.

## 2.2.6 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΔΙΚΤΥΟΥ

Ως προς την τοπολογία του δικτύου, το MAC επίπεδο του WiMAX υποστηρίζει δύο διαφορετικές καταστάσεις: την Point-to-Multipoint (PMP) και την Multipoint-to-Multipoint ή γνωστή και ως Mesh, όπως φαίνεται στην Εικόνα 2.3, χωρίς ωστόσο να αποκλείεται και η χρήση του για σημείο-προς-σημείο συνδέσεις (Point-to-Point). Η κατάσταση PMP μοιάζει με την κυψελωτή δομή και συνήθως εφαρμόζεται για πρόσβαση των τελικών χρηστών δηλαδή για κάλυψη των τελευταίων μέτρων [20]. Οι μεταδόσεις από το χρήστη προς το σταθμό, δηλαδή της ανερχόμενης ζεύξης, λαμβάνουν χώρα σε διαφορετικά χρονικά πλαίσια, ενώ στα πλαίσια κατερχόμενης ζεύξης ο σταθμός μπορεί να μεταδώσει ριπές PDU. Εφόσον η κατερχόμενη μετάδοση είναι ευρυεκπομπή (broadcast) ο χρήστης μπορεί να επεξεργαστεί μόνο εκείνα τα PDU που απευθύνονται στο ίδιο ή εκείνα που απευθύνονται σε όλους τους κινητούς σταθμούς. Οι χρήστες μοιράζονται την ανερχόμενη ζεύξη με βάση τη ζήτηση. Σε κατάσταση Mesh οι κόμβοι οργανώνονται όπως και στα ad hoc δίκτυα και δεν υπάρχει διαχωρισμός ανερχόμενης και κατερχόμενης ζεύξης στα πλαίσια. Κάθε κόμβος μπορεί να δημιουργήσει απευθείας συνδέσεις με άλλους κόμβους. Το 802.16 προσδιορίζει δύο μηχανισμούς για τον προγραμματισμό της μετάδοσης των δεδομένων σε κατάσταση mesh, τον κεντρικό και τον κατανεμημένο προγραμματισμό. Στο κεντρικό προγραμματισμό ο σταθμός βάσης είναι στο κέντρο και αποφασίζει πως οι κινητοί σταθμοί θα μοιραστούν τα κανάλια σε διαφορετικές χρονικές στιγμές. Ο BS μαζεύει αιτήσεις μέσω μηνυμάτων MSH-CSH (Mesh Centralized Scheduling) από όλα τα SS που απέχουν συγκεκριμένη απόσταση (μέγιστος αριθμός βημάτων – max hop) και επικοινωνούν τις πληροφορίες σε όλους τους SS. Με αυτό τον τρόπο οι SS καθορίζουν τις δικές τους ευκαιρίες μετάδοσης σε στυλ κατανομής χρησιμοποιώντας ένα κοινό προκαθορισμένο αλγόριθμο με ίδιες πληροφορίες εισόδου και έτσι η έξοδος είναι η ίδια για όλους. Οι SS ενημερώνουν τον BS για αλλαγές στις απαιτήσεις τους για πόρους με MSH-CSH μηνύματα, έπειτα ο BS επαναλαμβάνει με ευρυεκπομπή τις προσαρμοσμένες απαιτήσεις και τέλος οι SS μπορούν να ξανά υπολογίσουν τις ευκαιρίες τους για μετάδοση [31]. Ο κεντρικός χρονοπρογραμματισμός τυπικά χρησιμοποιείται με περισσότερο βελτιστοποιημένο τρόπο από τον κατανεμημένο για ροές κίνησης, το οποίο επιμένει σε διάρκεια μεγαλύτερη από το χρόνο ενός κύκλου για να αναμεταδώσει αιτήσεις νέων πόρων και κατανομή ανανεωμένου προγράμματος [18], [31]. Στον κατανεμημένο προγραμματισμό κάθε κόμβος συγχρονίζεται με τους υπόλοιπους για την πρόσβασή του στο κανάλι ανάλογα με τους διαθέσιμους πόρους, τις αιτήσεις και τα δικαιώματα. Συνήθως η επιλογή του κόμβου γίνεται με χρήση κάποιου ψευδοτυχαίου αλγόριθμου που βασίζεται σε πληροφορίες των γειτόνων του που βρίσκονται έως και δύο βήματα μακριά και αφού συγχρονιστεί, εκπέμπει τις σχετικές πληροφορίες σε όλους τους γειτονικούς κόμβους [21], [24], [33].



Εικόνα 2.3 – PMP και Mesh τοπολογίες

## 2.2.7 ΚΙΝΗΤΙΚΟΤΗΤΑ ΚΑΙ ΕΞΟΙΚΟΝΟΜΗΣΗ ΕΝΕΡΓΕΙΑΣ

Με τις τροποποιήσεις στο mobile WiMAX υποστηρίζεται τόσο η κινητικότητα εντός μίας κυψέλης όσο και μεταξύ διαφορετικών κυψελών. Για την δεύτερη υποστηρίζονται τριών ειδών

διαπομπές. Υποχρεωτική είναι η hard handover, δηλαδή όλες οι συνδέσεις στο παλιό σταθμό βάσης τερματίζονται πριν δημιουργηθούν νέες. Μια διαπομπή αποτελείται από τέσσερα στάδια:

- Επιλογή νέας κυψέλης: ο SS αναζητά τους καλύτερους γειτονικούς BS
- Απόφαση και έναρξη διαπομπής: είτε ο SS είτε ο BS που τον εξυπηρετεί αποφασίζει ότι χρειάζεται διαπομπή και ξεκινά την ανταλλαγή μηνυμάτων
- Συγχρονισμό με το νέο BS: ο SS συγχρονίζεται με το νέο BS, όπως και κατά την είσοδο στο δίκτυο ο SS βρίσκει τις τιμές των παραμέτρων των επιπέδων PHY και MAC και ανιχνεύει τον κατάλληλο δίαυλο, τέλος
- Τερματισμός σύνδεσης: ο SS ενημερώνει τον παλιό BS για την διαπομπή και αφού περάσει ένα χρονικό διάστημα ελευθερώνονται οι πόροι του συστήματος.

Εάν διακοπή η παραπάνω διαδικασία τότε ο SS πρέπει να ξανά κάνει είσοδο στο δίκτυο στον νέο BS.

Οι άλλες δύο διαπομπές είναι προαιρετικές και είναι με διαφορισμό BS (Macro-Diversity Handover) και γρήγορης μεταγωγής BS (Fast BS Switching). Στη πρώτη ο SS έχει ταυτόχρονα πρόσβαση σε πολλούς BS, οι οποίοι σαν σύνολο ονομάζονται ομάδα διαφορισμού και ένας είναι ο βασικός. Στην κατερχόμενη ζεύξη ο SS λαμβάνει ταυτόχρονα τα σήματα από όλους τους BS και τα συνδυάζει, ενώ στην ανερχόμενη όλοι οι BS λαμβάνουν από τον SS και τα στέλνουν στο βασικό για να επιλέξει το πακέτο με τα λιγότερα σφάλματα. Στη δεύτερη υπάρχει και πάλι ομάδα διαφορισμού, αλλά εδώ ο SS επικοινωνεί μόνο με τον βασικό BS και έχει την δυνατότητα να προσθαφαιρέσει BS από την ομάδα [21], [32].

Ως προς την δυνατότητα εξοικονόμησης ενέργειας ο κάθε SS έχει την δυνατότητα να κλείσει διάφορα τμήματά του. Υπάρχουν δύο καταστάσεις στις οποίες πετυχαίνει η εξοικονόμηση, η κατάσταση νάρκης (sleep mode) και η αδρανής κατάσταση (idle mode).

Τα διαστήματα μεταξύ ενεργής κατάστασης και νάρκης εναλλάσσονται συνεχώς. Όταν έστω και μια σύνδεση του SS είναι ενεργή τότε θεωρείται διαθέσιμος (available), διαφορετικά θεωρείται μη διαθέσιμος (unavailable). Ο τρόπος εναλλαγής εξαρτάται από τις καταστάσεις εξοικονόμησης ενέργειας και είναι οι τρεις ακόλουθες:

- Πρώτη κατάσταση, σε αυτή ο SS αποφασίζει για την μετάβαση του σε αυτή, ενώ ταυτόχρονα χρειάζεται και έγκριση από το δίκτυο. Τα διαστήματα ενεργής κατάστασης είναι σταθερά, ενώ τα διαστήματα νάρκης αυξάνονται μέχρι να φτάσουν μια μέγιστη προκαθορισμένη τιμή.
- Δεύτερη κατάσταση, τα δύο διαστήματα έχουν ίση χρονική διάρκεια και εναλλάσσονται σύμφωνα με απόφαση του BS.
- Τρίτη κατάσταση, σε αυτή δεν υπάρχουν εναλλαγές ο SS βρίσκεται σε κατάσταση νάρκης για μεγάλα διαστήματα, αφού ο BS γνωρίζει πότε πρέπει να βρίσκεται σε ενεργή οπότε σε όλα τα ενδιάμεσα είναι σε νάρκη.

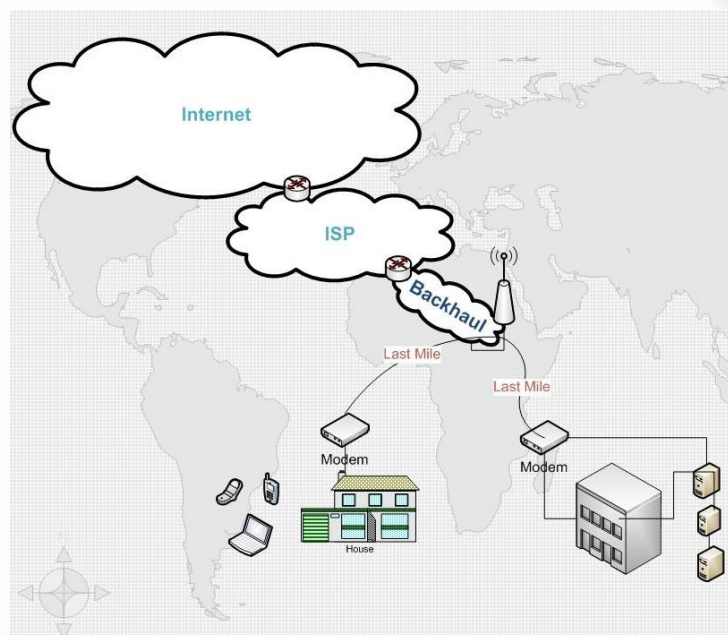
Η αδρανής κατάσταση είναι για να λαμβάνουν οι SS μηνύματα ευρυεκπομπής από τους BS στους οποίους δεν έχουν καταχωρηθεί και έτσι να αποφεύγονται οι άσκοπες διαπομπές σε SS με ανενεργές συνδέσεις. Με αυτό τον τρόπο εξοικονομούνται πόροι του συστήματος αφού μειώνεται ο όγκος της σηματοδότησης. Ο SS όταν είναι σε αδρανή κατάσταση σε κάποια χρονικά διαστήματα είναι προσβάσιμος και σε άλλα όχι [21].

## 2.3 ΧΡΗΣΕΙΣ

Σε αυτό το υποκεφάλαιο αναφέρονται περιληπτικά οι κυριότερες χρήσεις του 802.16.

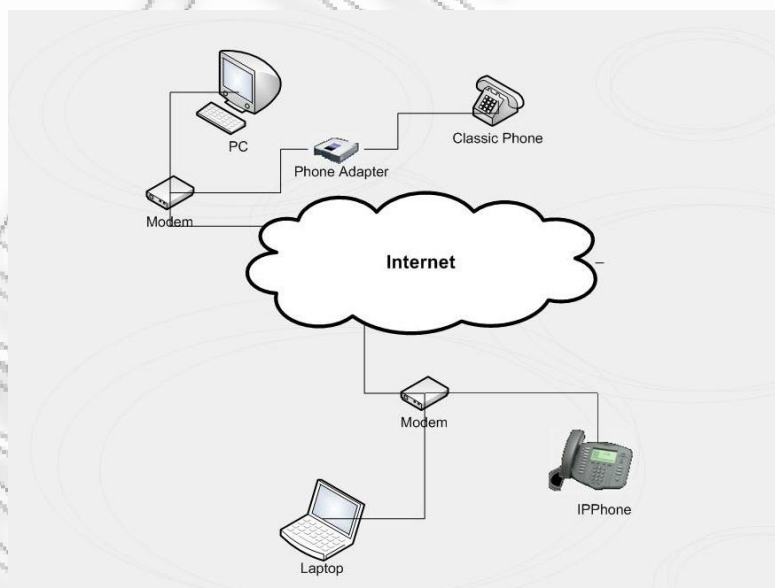
Η βασικότερη και ουσιαστικότερη χρήση του προτύπου από είναι για πρόσβαση στο Internet. Δύο είναι τα τμήματα στα οποία μπορεί να χρησιμοποιηθεί: στο Backhaul κομμάτι και στο κομμάτι που καλύπτει τα τελευταία μέτρα γνωστό ως Last Mile. Το 802.16 ενδείκνυται για χρήση ως backhaul δίκτυο, δηλαδή για το τμήμα που συνδέει τα απομακρυσμένα σημεία συγκέντρωσης με το κεντρικό δίκτυο του παρόχου, όπως για παράδειγμα η σύνδεση των κεραιών των κινητών τηλεπικοινωνιών ή των πολυπλεκτών / αποπολυπλεκτών των ψηφιακών συνδρομητικών γραμμών DSL (Digital Subscriber Line Access Multiplexer - DSLAM) με το κεντρικό δίκτυο. Η δεύτερη κατηγορία είναι η δημιουργία μοναδικών συνδέσεων μεταξύ κάθε χρήστη και του σταθμού βάσης που έχει πρόσβαση στο διαδίκτυο, γεγονός που θα βοηθούσε όχι μόνο στο πρόβλημα του εύρους ζώνης αλλά και στην ασφάλεια και αξιοπιστία της σύνδεσης. Δηλαδή χρήση της τεχνολογίας WiMAX ως δίκτυο για Last Mile Broadband Access, το οποίο

αφορά τη σύνδεση μεταξύ τελικού χρήστη και του πρώτου σημείου συγκέντρωσης του παρόχου του δικτύου [20], [27]. Στην Εικόνα 2.4 διακρίνονται οι δύο αυτές κατηγορίες.



**Εικόνα 2.4 - Τμήματα δικτύου: Backhaul και Last Mile [34]**

Μια δεύτερη εφαρμογή είναι η real-time επικοινωνία, δηλαδή η τηλεφωνία μέσω διαδικτύου (Voice-over-IP – VoIP). Η VoIP τηλεφωνία έχει πλέον μεγάλη ζήτηση τόσο από μεγάλες εταιρείες όσο και από ιδιώτες, αφού προσφέρει φωνητική συνομιλία σε πραγματικό χρόνο με πολύ καλή ποιότητα και το σημαντικότερο χωρίς κόστος [20], [27]. Αναπαράσταση κλασικών συνδέσεων VoIP φαίνονται στην Εικόνα 2.5.

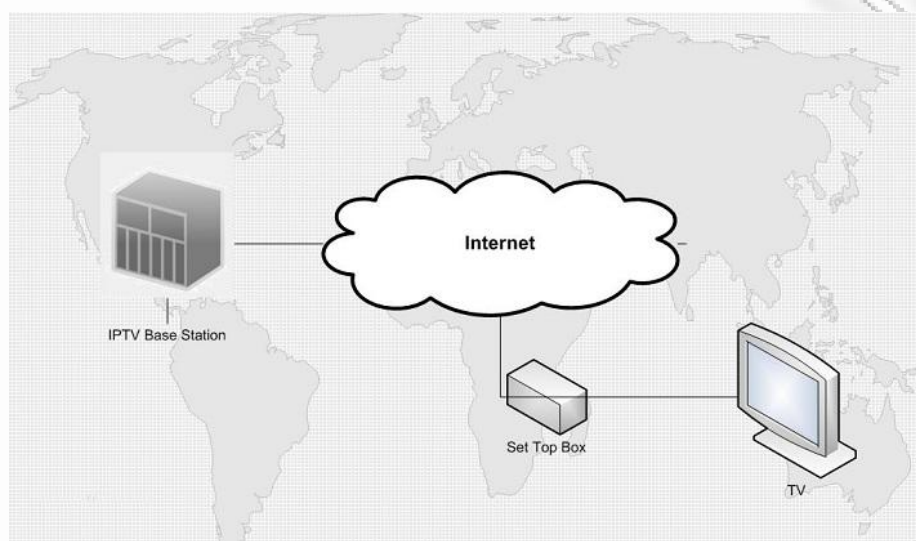


**Εικόνα 2.5 – VoIP συνδέσεις**

Άλλη μια χρήση του WiMAX μπορεί να γίνει στη δικτυακή τηλεόραση (IPTV), όπου σε αυτό το σύστημα εκπέμπεται το ψηφιακό σήμα στους συνδρομητές με ευρυζωνική σύνδεση. Επίσης, σημαντικό ρόλο μπορεί να παίξει το πρότυπο στην παροχή καταναμημένων υπηρεσιών σε μητροπολιτικά δίκτυα, όπως είναι ο ολοένα αυξανόμενος αριθμός υπηρεσιών προστιθέμενης



αξίας που προσφέρουν διάφορες εταιρείες [20], [27]. Στην Εικόνα 2.6 φαίνεται ο τρόπος σύνδεσης της δικτυακής τηλεόρασης.



Εικόνα 2.6 – IPTV

Ένα ακόμα πεδίο στο οποίο μπορεί το πρότυπο να βρει πρόσφορο έδαφος για την εφαρμογή του είναι η ιατρική τηλεμετρία. Ο κλάδος αυτός έχει σκοπό την συλλογή ιατρικών δεδομένων με μεγάλη ακρίβεια από ένα πλήθος ιατρικών συσκευών. Οι συσκευές στέλνουν αυτόματα τα δεδομένα στον υπολογιστή ασύρματα για να δημιουργηθούν, να αποθηκευτούν και να διαχειριστούν οι εκάστοτε εφαρμογές τα ιατρικά αρχεία. Η ιατρική τηλεμετρία χρησιμοποιείται σε ασθενείς εν κινήσει και στόχο έχει την βελτίωση της περίθαλψης και της ποιότητας της ζωής των ασθενών [35].

Τέλος, αναφέρονται οι στρατιωτικές εφαρμογές αφού θα μπορούν να μοιράζονται με ασφάλεια τις υψηλές συχνότητες του WiMAX με τις εμπορικές εφαρμογές αλλά και ένας ακόμα τομέας που βρίσκεται σε εξέλιξη σήμερα είναι εφαρμογές location based για διαφημιστικούς σκοπούς [20], [27], [35].

## 2.4 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΑ

Τα πλεονεκτήματα και στη συνέχεια τα μειονεκτήματα της τεχνολογίας αυτής έναντι των ασύρματων και ενσύρματων τεχνολογιών της σημερινής εποχής περιγράφονται παρακάτω:

Ένα πολύ σημαντικό πλεονέκτημα του προτύπου WiMAX είναι ότι παρέχει διαλειτουργικότητα μεταξύ των προμηθευτών της τεχνολογίας, κάτι που επιτρέπει την μείωση του κόστους, την αύξηση της ελευθερίας και της ευελιξίας σε όσους το υιοθετήσουν. Στον τηλεπικοινωνιακό τομέα οι επιχειρήσεις μπορούν να κατηγοριοποιηθούν ως πάροχοι δικτύων, οι οποίοι έχουν στην κατοχή τους και εκμεταλλεύονται ένα δίκτυο επικοινωνιών και ως πάροχοι υπηρεσιών οι οποίοι μεταδίδουν μηνύματα μέσω ενός δικτύου επικοινωνιών [35]. Ο διαχωρισμός του δικτύου, δηλαδή να ανήκει σε άλλη εταιρεία το δίκτυο πρόσβασης και σε άλλη το κεντρικό δίκτυο, χαμηλώνει το φράγμα εισόδου στις κινητές τηλεπικοινωνίες. Μπορεί να προσφέρει υπηρεσίες συγκρίσιμες με αυτές της ανερχόμενης τεχνολογίας 4G, ενώ παρέχει συμπληρωματικές υπηρεσίες ως προς την DSL και την καλωδιακή. Η ανάπτυξη ενός συστήματος WiMAX για να υποστηρίξει νέους χρήστες είναι πολύ γρηγορότερη σε αντίθεση με τις ενσύρματες τεχνολογίες [36]. Επιπροσθέτως, η διαπερατότητα και η συνολική χωρητικότητα, δηλαδή η φασματική απόδοση, ενός WiMAX συστήματος κατά μέσο όρο είναι πολύ καλύτερη, ενώ όπως προαναφέρθηκε η τεχνολογία MIMO έξυπνων κεραιών αυξάνει την ποιότητα. Χάρη στη IP αρχιτεκτονική του είναι χαμηλότερο το κόστος των προηγμένων εφαρμογών όπως η φωνή, το βίντεο και τα πολυμέσα [2], [24].

Ωστόσο υπάρχουν και πολλές προκλήσεις. Μια αδυναμία είναι το roaming, αφού δεν έχει αποδειχτεί η ικανότητα του συστήματος σε αυτόν τον τομέα, αλλά και γενικότερα τα συστήματα WiMAX δεν έχουν εφαρμοστεί ευρέως και δεν υπάρχει ακόμα εμπειρία. Το WiMAX δύσκολα θα

ανταγωνιστεί τις υπάρχουσες ευρυζωνικές συνδέσεις όπως η DSL και η καλωδιακή αφού συνεχίζουν να εξελίσσονται και να ικανοποιούν της απαιτήσεις του κοινού, οπότε στο κομμάτι της αγοράς που διαθέτει ήδη μια τέτοια σύνδεση το μόνο που μπορεί να προβληθεί είναι η κινητικότητα που προσφέρει το 802.16. Σε κάθε μελέτη θα πρέπει να εξετάζεται το θέμα των συχνοτήτων αφού σε διαφορετικά μέρη του κόσμου έχουν διαφορετικές χρήσεις για το κάθε κανάλι. Υπάρχουν βέβαια και θέματα ασφάλειας που δεν έχουν λυθεί, όπως ο μηχανισμός αυθεντικοποίησης ή η ποιότητα των κλειδιών του προτύπου [36], [37]. Τέλος, αναφέρεται ακόμα ένα οικονομικό μειονέκτημα, η ανάγκη για κατασκευή κατάλληλων συσκευών και η διάθεσή τους στην αγορά [2], [24].

Σε σχέση με το 3G υποστηρίζει πολύ υψηλότερες ταχύτητες και μεγαλύτερο εύρος ζώνης για την επιλογή των καναλιών που θα χρησιμοποιηθούν. Οι συχνότητες είναι μεγαλύτερες και θα κοστίζουν φθηνότερα στις επιχειρήσεις σε αντίθεση με το UMTS, όμως ταυτόχρονα οι μεγαλύτερες συχνότητες σημαίνουν και μικρότερη περιοχή κάλυψης ανά σταθμό [38]. Γενικά υποστηρίζει μηχανισμούς για εγγυημένο QoS σε IP εφαρμογές, ενώ βασίζεται στο OFDM του οποίου η αποδοτικότητα έχει αποδειχτεί. Η υποστήριξη τόσο συνδέσεων χωρίς οπτική επαφή (NLOS) όσο και συνδέσεων με οπτική επαφή (LOS) το κάνει κατάλληλο όχι μόνο για συνεργασία αλλά και αντικατάσταση άλλων ασύρματων τεχνολογιών [37]. Για το 3G βέβαια υπάρχει μεγάλη εμπειρία και έχουν ήδη γίνει πολλές επενδύσεις από εταιρείες, ανάμεσα στους υποστηρικτές είναι η NOKIA, ενώ την τεχνολογία WiMAX υποστηρίζουν οι Intel, KT και Samsung [38].

Σε σύγκριση με το WiFi είναι αρκετά διαφορετικές τεχνολογίες. Το WiFi καλύπτει μικρές αποστάσεις κυρίως εσωτερικού χώρου, ενώ το WiMAX καλύπτει μεγάλες αποστάσεις και υποστηρίζει κινητικότητα μεταξύ κυψελών, έτσι το WiMAX μπορεί να σταθεί ως backhauling δίκτυο πίσω από το WiFi. Ένα σημαντικό πλεονέκτημα του WiMAX έναντι του WiFi είναι οι μηχανισμοί για QoS που διαθέτει και έτσι έχει καλύτερη απόδοση, ενώ το κόστος του 802.11 είναι χαμηλότερο [38].

### 3 ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Ο χρονοπρογραμματισμός (scheduling) είναι το πρόβλημα εύρεσης μιας βέλτιστης σειράς για την εκτέλεση ενός συνόλου λειτουργιών, ενώ ταυτόχρονα δεν παραβιάζονται συγκεκριμένοι κανόνες. Ο χρονοπρογραμματιστής (scheduler) είναι υπεύθυνος για την εκτέλεση των λειτουργιών με τη σειρά που όρισε ο αλγόριθμος χρονοπρογραμματισμού (scheduling algorithm), ο οποίος πρέπει να εξασφαλίζει μεταξύ άλλων δικαιοσύνη (fairness), αποδοτικότητα (efficiency), ελάχιστο χρόνο απόκρισης και διεκπεραίωσης (response/ turnaround time), διαπερατότητα (throughput), απλή υλοποίηση και επεκτασιμότητα (scalability) καθώς και διατήρηση της Ποιότητας Υπηρεσιών.

Υπάρχει ο χωρίς διακοπές χρονοπρογραμματισμός (non preemptive scheduling), όπου η μετάδοση των πακέτων δεν διακόπτεται εφόσον ξεκινήσει και χρησιμοποιείται συνήθως στα δίκτυα υπολογιστών, αφού τέτοιες διακοπές θα αποτελούσαν σπατάλη πόρων. Επίσης υπάρχει ο χρονοπρογραμματισμός με διακοπές (preemptive scheduling) που είναι συνηθισμένος στο χρονοπρογραμματισμό διεργασιών, εφόσον είναι εφικτή η προσωρινή διακοπή τους.

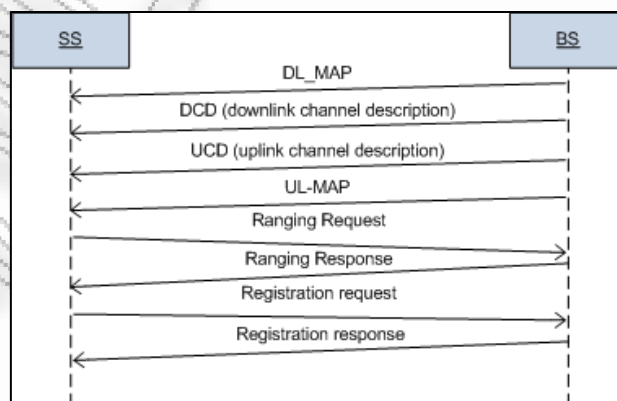
Στη συνέχεια παρατίθεται μια λίστα με τους γνωστότερους αλγόριθμους χρονοπρογραμματισμού σε ασύρματα δίκτυα [21]:

- Εξυπηρέτηση εκ περιτροπής (Round Robin - RR)
- Μέγιστος λόγος φέροντος προς παρεμβολή (Maximum Carrier to Interference - Max C/I)
- Αναλογική αμεροληψία (Proportional Fair - PF)
- Περιορισμοί στο κατά αναλογία ρυθμό (Proportional Rate Constraints - PRC)

Παρακάτω αναλύεται ο χρονοπρογραμματισμός των υπηρεσιών και οι προδιαγραφές του σύμφωνα με το πρότυπο του WiMAX.

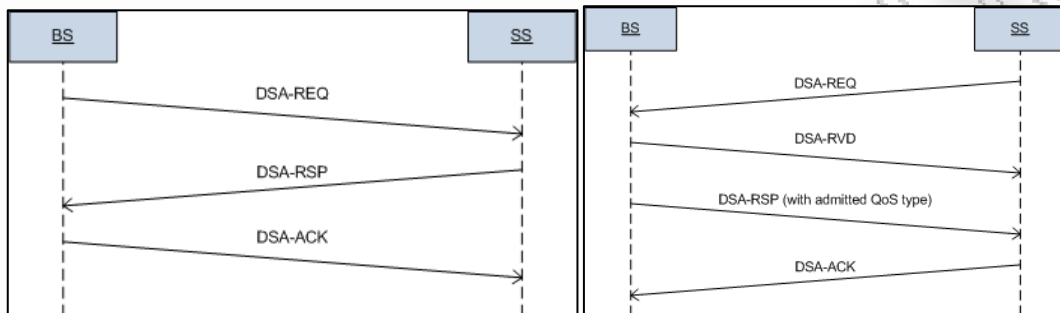
#### 3.1 ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΤΟ WiMAX

Υπάρχουν δύο διαδικασίες που πρέπει να ακολουθηθούν προκειμένου να εξυπηρετηθεί μια ροή υπηρεσίας, ο έλεγχος πρόσβασης (Admission Control) και ο έλεγχος εκχώρησης (Granting Control). Και οι δύο λαμβάνουν υπόψη τους τις QoS απαιτήσεις και τη χρησιμοποίηση του εύρους ζώνης του δικτύου. Μόλις μια νέα υπηρεσία φτάσει στον SS, εκείνο προσπαθεί να δημιουργήσει μια σύνδεση με τον BS. Μόλις ο BS λάβει την αίτηση, αποφασίζει εάν θα δεχτεί την σύνδεση ή όχι και πόσο εύρος ζώνης πρέπει να κατανεμηθεί στη σύνδεση για όλη τη διάρκεια της μετάδοσης. Αυτή η διαδικασία είναι ο έλεγχος πρόσβασης. Μετά τη δημιουργία της σύνδεσης ο χρήστης αρχίζει την μετάδοση των δεδομένων. Αφού τα δεδομένα παράγονται σε ριπές, όταν η σύνδεση έχει δεδομένα προς μετάδοση, ο SS χρησιμοποιεί τις εκχωρήσεις ή τις ευκαιρίες για αίτηση εύρους ζώνης που του δίνονται από τον BS, ανάλογα με τον τύπο της ροής υπηρεσίας, και ο BS αποφασίζει αν θα εκχωρήσει εύρος ζώνης και πόσο. Η διαδικασία αυτή ονομάζεται έλεγχος εκχώρησης [39]. Μια διαφορά των διαδικασιών είναι ότι στη πρώτη σημαντικό ρόλο παίζει η μακροπρόθεσμη απόδοση και χρήση του εύρους ζώνης του συστήματος, ενώ στη δεύτερη σημασία έχει το στιγμιαίο αποτέλεσμα. Η παρούσα εργασία ασχολείται με την δεύτερη διαδικασία. Στην επόμενη εικόνα παρουσιάζεται η διαδικασία εισαγωγής ενός SS στο δίκτυο.



Εικόνα 3.1 – Εισαγωγή SS στο δίκτυο

Ακολουθεί δύο εικόνες που δείχνουν την έναρξη μιας ροής υπηρεσίας από τον BS και από τον SS αντίστοιχα.



Εικόνα 3.2 – Έναρξη ροής υπηρεσιών από τον BS και τον SS

Οι υπηρεσίες ροών είναι κανάλια επικοινωνίας, δηλαδή ροές πακέτων, που έχουν συγκεκριμένα χαρακτηριστικά υπηρεσιών συσχετισμένα με τη μεταφορά των δεδομένων [39]. Οι αλγόριθμοι χρονοπρογραμματισμού πρέπει να υιοθετούν μια ευέλικτη στρατηγική στην κατανομή των χρονοσχισμών στους SS σύμφωνα με τις ανάγκες τους και πρέπει να παραμένουν στον SS για όλη τη διάρκεια της επικοινωνίας. Το σχέδιο χρονοπρογραμματισμού πρέπει να βασίζεται στις προτεραιότητες ώστε να κατανεμηθούν οι πόροι σωστά στους διαφορετικούς τύπους υπηρεσιών ανάλογα με τις QoS απαιτήσεις. Η διαχείριση των πόρων τόσο της ανερχόμενης όσο και της κατερχόμενης ζεύξης πραγματοποιείται από τον BS. Βέβαια ένας SS μπορεί να διατηρεί πολλές συνδέσεις με τον BS, σε αυτή την περίπτωση ο ίδιος μοιράζει το εύρος ζώνης που του κατανεμήθηκε μεταξύ των συνδέσεών του [21]. Εκτός από τις κλασσικές απαιτήσεις που πρέπει να εξασφαλίζει ένας αλγόριθμος, όπως προαναφέρθηκαν στην αρχή του κεφαλαίου, στο WiMAX πρέπει να είναι και ενήμερος ως προς το κανάλι (channel-awareness) και να λαμβάνει ευκαιριακές αποφάσεις εξυπηρετώντας τον SS που έχει καλό σήμα χωρίς να παραβιάζει τις QoS απαιτήσεις των SS με κακή ποιότητα σήματος [40], [41].

Ο χρονοπρογραμματισμός εξερχόμενων μεταδόσεων επιλέγει τα δεδομένα σε ένα συγκεκριμένο κατανεμημένο πλαίσιο και εκτελείται από τον BS στην περίπτωση της κατερχόμενης ζεύξης και από τον SS στη περίπτωση της ανερχόμενης. Λαμβάνονται πάντα υπόψη η υπηρεσία χρονοπρογραμματισμού που έχει προσδιοριστεί για τη ροή, οι τιμές που έχουν δοθεί στις QoS παραμέτρους, η διαθεσιμότητα δεδομένων προς μετάδοση και η χωρητικότητα του εκχωρημένου εύρους ζώνης [19].

Ο χρονοπρογραμματισμός των αιτήσεων/ εκχωρήσεων στην ανερχόμενη ζεύξη εκτελείται από τον BS με σκοπό την παροχή εύρους ζώνης σε κάθε SS για τις ανερχόμενες μεταδόσεις ή για τις ευκαιρίες για αίτηση εύρους ζώνης. Με τον ορισμό του τύπου χρονοπρογραμματισμού και των QoS παραμέτρων του, ο BS χρονοπρογραμματιστής μπορεί να ξέρει τις ανάγκες σε διαπερατότητα και καθυστέρηση της ανερχόμενης κίνησης και να παρέχει διαδοχικές διερευνήσεις ή και εκχωρήσεις στους κατάλληλους χρόνους [19].

Οι απαραίτητες πληροφορίες για να αποφασίσει ο BS για την εκχώρηση εύρους ζώνης μεταφέρονται σε αυτόν με τα dl-map και μέσω αυτών ενημερώνει τους SS για την κατανομή των πόρων [2], [21]. Ενώ τα ul-map μηνύματα του DL υποπλάσιου ενημερώνουν τους SS για εκχώρηση ευκαιριών μονοεκπομπής διαδοχικής διερεύνησης (unicast polling) [2].

Ο Πίνακας 2 συνοψίζει τους τύπους χρονοπρογραμματισμού και τις επιλογές διαδοχικής διερεύνησης/ εκχώρησης που διατίθεται για το καθένα.

Τύπος	Piggybacking Request	Bandwidth stealing	Polling
UGS	-	-	-
rt-PS	✓	✓	Μονοεκπομπές
nrt-PS	✓	✓	Μερικές φορές μονοεκπομπές, διαφορετικά όλες οι μορφές (εξαρτάται από την πολιτική)
BE	✓	✓	Όλες οι μορφές
ert-PS	✓	-	Μονοεκπομπές ή ανταγωνιστικές ευκαιρίες αίτησης

**Πίνακας 2 - Τύποι χρονοπρογραμματισμού και επιλογές διαδοχικής διερεύνησης/ εκχώρησης [19]**

Δεν χρειάζεται κάθε υπηρεσία όλες τις παραμέτρους QoS, οπότε παρακάτω καθορίζονται ποιες παράμετροι χρειάζονται σε ποίο τύπο υπηρεσίας [19].

- Σταθερού Ρυθμού Μετάδοσης: Οι απαιτούμενες παράμετροι QoS είναι η ανεκτή τιμή jitter, ο ελάχιστος δεσμευμένος ρυθμός μετάδοσης, η μέγιστη καθυστέρηση και η πολιτική αίτησης/ μετάδοσης.
- Μεταβαλλόμενου Ρυθμού Υψηλής Διαθεσιμότητας: Οι απαιτούμενες παράμετροι QoS είναι ο ελάχιστος δεσμευμένος ρυθμός μετάδοσης, ο μέγιστος σταθερός ρυθμός μετάδοσης, η μέγιστη καθυστέρηση, η προτεραιότητα κίνησης και η πολιτική αίτησης/ μετάδοσης.
- Μεταβαλλόμενου Ρυθμού Χαμηλής Διαθεσιμότητας: Οι απαιτούμενες παράμετροι QoS είναι ο ελάχιστος δεσμευμένος ρυθμός μετάδοσης, ο μέγιστος σταθερός ρυθμός μετάδοσης, η προτεραιότητα κίνησης και η πολιτική αίτησης/ μετάδοσης.
- Διαθέσιμου Εύρους: Οι απαιτούμενες παράμετροι QoS είναι ο μέγιστος σταθερός ρυθμός μετάδοσης και πολιτική αίτησης/ μετάδοσης.
- Μεταβαλλόμενου Ρυθμού Εκτεταμένης Υψηλής Διαθεσιμότητας: Οι απαιτούμενες παράμετροι QoS είναι η ανεκτή τιμή jitter, ο ελάχιστος δεσμευμένος ρυθμός μετάδοσης, ο μέγιστος σταθερός ρυθμός μετάδοσης, η μέγιστη καθυστέρηση, η προτεραιότητα κίνησης και η πολιτική αίτησης/ μετάδοσης.
- Στις UGS και ert-PS συνδέσεις λαμβάνουν εκχωρήσεις (grants) σε διαστήματα που καθορίζονται από την παράμετρο διάστημα αυτόκλητης εκχώρησης (unsolicited grant interval), στις rt-PS συνδέσεις οι μονοεκπομπές διαδοχικής διερεύνησης (unicast polling) λαμβάνουν χώρα σε διαστήματα σύμφωνα με την παράμετρο διάστημα αυτόκλητης μονοεκπομπής (unsolicited polling interval), ενώ για τις nrt-PS το διάστημα καθορίζεται από τον BS [42].

Στον ακόλουθο πίνακα συγκεντρώνεται η προηγούμενη πληροφορία.

QoS παράμετροι	Ανεκτή τιμή jitter	Μέγιστη καθυστέρηση	Μέγιστος σταθερός ρυθμός μετάδοσης	Ελάχιστος δεσμευμένος ρυθμός μετάδοσης	Προτεραιότητα κίνησης	Πολιτική αίτησης/ μετάδοσης
Υπηρεσίες						
UGS	✓	✓		✓		✓
rt-PS		✓	✓	✓	✓	✓
ert-PS		✓	✓	✓		✓
nrt-PS			✓	✓	✓	✓
BE			✓			✓

**Πίνακας 3 – QoS παράμετροι των διαφορετικών τύπων υπηρεσιών [19]**

Η υπολογιζόμενη κίνηση στο πρότυπο είναι κυρίως από και προς τον BS οπότε μελετάται ο χρονοπρογραμματισμός ανά ζεύξη [31]. Η παρούσα εργασία επικεντρώνεται στην ανερχόμενη κατεύθυνση.

### 3.2 ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ

Στο [30] προτείνεται ένα σχήμα χρονοπρογραμματισμού για την κίνηση στην ανερχόμενη ζεύξη και μπορεί να εφαρμοστεί στον BS. Ο χρονοπρογραμματιστής κατανέμει εκχωρήσεις (grants) με μέγεθος μικρότερο ή ίσο του max traffic burst με τέτοιο τρόπο ώστε κάθε σύνδεση να μεταδίδει με ρυθμό που ποικίλει από το μέγιστο σταθερό ρυθμό μετάδοσης μέχρι τον ελάχιστο δεσμευμένο ρυθμό μετάδοσης. Επίσης, τηρείται η απαίτηση ως προς την μέγιστη καθυστέρηση στις συνδέσεις πραγματικού χρόνου για την κίνηση που δεν ξεπερνά τον ελάχιστο δεσμευμένο ρυθμό μετάδοσης. Στις UGS και ert-PS συνδέσεις ισχύει το διάστημα αυτόκλητης εκχώρησης, στις rt-PS συνδέσεις το διάστημα αυτόκλητης μονοεκπομπής, ενώ για τις nrt-PS το διάστημα καθορίζεται από τον BS. Ο BS διαθέτει τρεις ουρές τις low για τις BE συνδέσεις, intermediate για τις rt-PS, nrt-PS και high για όσες rt-PS και nrt-PS αιτήσεις πρέπει να μεταδοθούν οπωσδήποτε διότι τελειώνει ο χρόνος τους, καθώς επίσης και τις περιοδικές εκχωρήσεις και μονοεκπομπές ευκαιριών αιτήσεων που πρέπει να προγραμματιστούν στο επόμενο πλαίσιο (frame). Ο BS εκτελεί τον χρονοπρογραμματιστή σε κάθε πλαίσιο και κάνει ευρυεκπομπή του προγράμματος στους SS με ul-mar μηνύματα. Σε κάθε πλαίσιο ο χρονοπρογραμματιστής δημιουργεί περιοδικές εκχωρήσεις και τα εισάγει στη ουρά high, με αυτόν τον τρόπο εγγυάται την εξυπηρέτηση των UGS και ert-PS συνδέσεων. Για να εγγυηθεί τη μέγιστη καθυστέρηση αναθέτει μια προθεσμία (deadline) σε κάθε rt-PS σύνδεση κατά την εισαγωγή της αίτησης στην ουρά intermediate, οι αιτήσεις που λήγουν σε δύο πλαίσια από αυτό της εκτέλεσης του χρονοπρογραμματιστή και δεν έχουν λάβει τον ελάχιστο δεσμευμένο ρυθμό μετάδοσης μεταφέρονται στην ουρά high. Ο ελάχιστος δεσμευμένος ρυθμός μετάδοσης εγγυάται για τις rt-PS, nrt-PS συνδέσεις σε μια διάρκεια T. Σε κάθε εκτέλεση υπολογίζεται η προτεραιότητα κάθε αίτησης στην ουρά intermediate σύμφωνα με τον ελάχιστο δεσμευμένο ρυθμό μετάδοσης, τις backlogged αιτήσεις και το παρόν ρυθμό μετάδοσης. Όσο χαμηλότερο από τον ελάχιστο δεσμευμένο ρυθμό μετάδοσης τόσο μεγαλύτερη η προτεραιότητα. Οι προσομοιώσεις απέδειξαν την αποτελεσματικότητα του σχήματος, αφού όταν υπάρχουν διαφορετικοί τύποι υπηρεσιών ο χρονοπρογραμματιστής κατανέμει ένα ικανοποιητικό αριθμό σχισμών σε κάθε σύνδεση έτσι ώστε να εγγυάται τις QoS απαιτήσεις.

Στο [43] ο αλγόριθμος λαμβάνει υπόψη του μόνο τρεις τύπους υπηρεσιών τον UGS, rt-PS και BE. Ένας χρήστης UGS εισάγεται μόνο εάν υπάρχουν αρκετοί πόροι για τον ρυθμό δεδομένων του και ένας χρήστης rt-PS μόνο εάν υπάρχουν αρκετοί πόροι για τον ελάχιστο δεσμευμένο ρυθμό μετάδοσης. Οι BE υπηρεσίες δεν ελέγχονται σε αυτό το στάδιο, αφού υπάρχει έλεγχος μέσω κοστολόγησης, δηλαδή τα BE δεν απορρίπτονται ποτέ, αλλά σε δεδομένες στιγμές το κόστος μετάδοσης των δεδομένων μπορεί να είναι πολύ υψηλό για ένα χρήστη και τελικά να επιλέξει να μην μεταδώσει. Ο BS εφαρμόζει χρονοπρογραμματιστή Round Robin για να μοιράζει πόρους για τις ανερχόμενες ζεύξεις. Προστίθενται στην υλοποίηση του NS- 2 τρία πεδία στο μήνυμα Dynamic Service Addition response (DSA-RSP), το basic\_price που αντιπροσωπεύει την χρέωση ανά μονάδα ρυθμού μετάδοσης των UGS και rt-PS με ελάχιστο δεσμευμένο ρυθμό μετάδοσης, το extra\_price που αντιπροσωπεύει το κόστος για τον επιπλέον ρυθμό των rt-PS και το extra\_rate\_max που αντιπροσωπεύει τον μέγιστο ρυθμό για τον rt-PS. Ένας UGS χρήστης στέλνει αίτηση εμπεριέχοντας τον σταθερό ρυθμό που χρειάζεται και ο BS υπολογίζει αν μπορεί να τον εξυπηρετήσει ή όχι. Εάν μπορεί τότε υπολογίζει το κόστος basic\_price και ο SS απαντά εάν δέχεται την σύνδεση ή όχι. Ένας rt-PS χρήστης στέλνει αίτηση εμπεριέχοντας τον μέγιστο και ελάχιστο ρυθμό που χρειάζεται και ο BS υπολογίζει αν μπορεί να τον εξυπηρετήσει ή όχι. Εάν μπορεί τότε υπολογίζει το κόστος extra\_price και extra\_rate\_price και ο SS απαντά εάν δέχεται την σύνδεση ή όχι. Ενώ ένας BE χρήστης συνδέεται αυτόματα διότι δεν έχει QoS απαιτήσεις και η κατανομή γίνεται δυναμικά στην αρχή κάθε πλαισίου.

Στο [40] προτείνεται ένας χρονοπρογραμματιστής για τον BS για την κίνηση στην κατερχόμενη ζεύξη σε σταθερούς χρήστες και τοπολογία PMP. Χρησιμοποιεί Deficit Round Robin (DRR) ως βάση και ως στόχο έχει το καλύτερο σχήμα χρονοπρογραμματισμού ανάμεσα στο ευκαιριακό και στον βασισμένο στην αποζημίωση όταν η AMC υλοποιείται στο φυσικό επίπεδο. Ο χρονοπρογραμματιστής είναι κεντρικός, χρησιμοποιείται TDD και λαμβάνει υπόψη του και τους πέντε τύπους υπηρεσιών. Ο BS αποθηκεύει πακέτα ροών σε διαφορετικές ουρές

ανάλογα με τον τύπο και εκεί περιμένουν για μετάδοση σε κάθε SS. Κάθε πακέτο λαμβάνει μια χρονοσφραγίδα κατά την άφιξή του. Η διαχείριση των ουρών γίνεται σε κάθε πλαίσιο εκτός από τις UGS και rt-PS που δεν χρειάζονται. Ο χρονοπρογραμματιστής χρησιμοποιεί την κατάσταση των καναλιών (channel-awareness), όπου κάθε SS εκτελεί Radio Signal Strength Intensity και Carrier to Interference and Noise Ratio και στέλνει τα αποτελέσματα στον BS χωρίς λάθη. Ο DRR συντηρεί ένα μετρητή και το ποσοστό της υπηρεσίας για κάθε ουρά. Κάθε φορά που μια ουρά επιλέγεται ο μετρητής αυξάνεται κατά την τιμή του ποσοστού και τα πακέτα στέλνονται εάν το μέγεθος τους είναι μικρότερο από το μετρητή. Κάθε φορά που στέλνεται κάποιο πακέτο ο μετρητής μειώνεται κατά το μέγεθος του πακέτου. Η βελτίωση της απόδοσης γίνεται με διαφορετικές τεχνικές, οι οποίες χρησιμοποιούν τις πληροφορίες σχετικά με την κατάσταση του καναλιού ανά SS, όπου ο BS συγκρίνει τις μετρήσεις του καθενός και αποφασίζει εάν θα εξυπηρετηθεί ή όχι και τις πληροφορίες των μη εξυπηρετημένων υπηρεσιών ανά ροή, όπου ο BS μαζεύει πληροφορίες για τις χαμένες δυνατότητες μετάδοσης κάθε ροής. Έχουν αναπτυχθεί τρεις διαφορετικοί αλγόριθμοι, ο ευκαιριακός (opportunistic) στον οποίο μόλις επιλεγεί μία ουρά επιλέγονται τα πακέτα που θα μεταδοθούν στο καλύτερο κανάλι, ο βασισμένος στην αποζημίωση στον οποίο μόλις επιλεγεί η ουρά επιλέγονται τα πακέτα που ανήκουν σε ροές με καλά κανάλια και αναβάλλει την μετάδοση άλλων πακέτων μέχρι να επανέλθει η ποιότητα των καναλιών τους και τέλος ένας υβριδικός αλγόριθμος που διαλέγει τα πακέτα με το καλύτερο κανάλι ανάμεσα στις ροές που έχουν χάσει δυνατότητες μετάδοσης. Τα αποτελέσματα της προσομοίωσης απέδειξαν πόσο απαραίτητη είναι η γνώση της κατάστασης του καναλιού καθώς επίσης και την περισσότερο δίκαιη συμπεριφορά του αλγορίθμου με βάση την αποζημίωση.

Το [44] μελετά έναν αλγόριθμο για τοπολογία PMP, με χρήση TDD και OFDM για ροές rt-PS και nrt-PS. Η εφαρμογή δημιουργεί μια σύνδεση μεταξύ του BS και των σχετικών ροών. Ο BS δίνει ένα μοναδικό CID σε κάθε σύνδεση, το οποίο χρησιμοποιείται στις αιτήσεις για εύρος ζώνης στο MAC επίπεδο και στις επικεφαλίδες των MAC PDU, ενώ στο πρότυπο η καταχώρηση εύρους ζώνης γίνεται ανά SS. Αφού γίνει η σύνδεση καταχωρούνται πόροι στις ροές της σύνδεσης. Ο χρονοπρογραμματιστής μπορεί να εφαρμοστεί τόσο σε SS με μια ροή όσο και σε SS με πολλές ροές και χρησιμοποιεί δύο εργαλεία την κατανομή πόρων (uplink resource allocation) στην ανερχόμενη ζεύξη και την προσαρμογή του διαστήματος διαδοχικής διερεύνησης (polling interval adjustment). Στο πρώτο υπολογίζονται οι προτεραιότητες του κάθε SS σύμφωνα με το εύρος ζώνης που ζητά και το διάστημα αναμονής από την τελευταία φορά που εξυπηρετήθηκε. Οι SS ταξινομούνται και εξυπηρετούνται με όσες σχισμές ζητήσουν μέχρι που δεν μπορεί να ικανοποιηθεί ολόκληρη κάποια αίτηση οπότε δίνονται όσες σχισμές απομένουν ή μέχρι να φτάσει σε προτεραιότητα ίση με το μηδέν και ότι δεν εξυπηρετηθεί διαγράφεται. Ως προς το δεύτερο το πρότυπο ορίζει ένα μηχανισμό αίτησης/ εκχώρησης με διαδοχική διερεύνηση αλλά η διάρκεια του κύκλου είναι ακαθόριστη και ενώ συνήθως δίνεται μια σταθερή τιμή σε αυτή την εργασία παρουσιάζεται μια προσαρμοστική μέθοδος. Το διάστημα διαδοχικής διερεύνησης του nrt-PS είναι πάντα μεγαλύτερο από αυτό του rt-PS, ενώ το ίδιο ισχύει και για το μέγεθος των πακέτων. Επειδή αυτό μπορεί να προκαλέσει έλλειψη πόρων εισάγεται μια σημαία κατάστασης των ουρών (status-flag), εάν είναι μηδέν δεν υπάρχουν αποθέματα και εάν είναι ένα υπάρχουν. Η προσομοίωση αποδεικνύει ότι αυξάνεται η απόδοση της nrt-PS κίνησης και ειδικά ως προς την ανεπάρκεια των πόρων, χωρίς να επιβαρύνεται η rt-PS, ενώ παράλληλα σημαντικότερο ρόλο είχε η τεχνική προσαρμογής του διαστήματος διαδοχικής διερεύνησης.

Στο [39] προτείνεται μια MAC τεχνική για την διαφοροποίηση των υπηρεσιών δύο κλάσεων του WiMAX, του UGS και BE στην ανερχόμενη ζεύξη. Ο χρονοπρογραμματιστής χρησιμοποιεί ένα κουβά με σκυτάλες (token bucket) για να καθορίσει τις εκχωρήσεις μετάδοσης για την εξυπηρέτηση των UGS, για την οποία κατανέμονται πόροι σε περιοδική βάση σύμφωνα με τον αριθμό των συσσωρευμένων σκυταλών της κάθε ροής. Κάθε σύνδεση UGS έχει το δικό της κουβά σκυταλών, οι οποίες φτάνουν περιοδικά σύμφωνα με το διάστημα που ορίζεται στην συμφωνία για το επίπεδο παροχής των υπηρεσιών (Service Level Agreement – SLA) της ροής. Κάθε σκυτάλη σημαίνει δικαίωμα για μετάδοση ενός πακέτου και κάθε φορά που μια μετάδοση λαμβάνει χώρα αφαιρείται μια σκυτάλη από τον κουβά της ροής. Ένα μικρό μέρος της χωρητικότητας του συστήματος φυλάσσεται πάντα για τις ροές BE και εξαρτάται από το πόσοι πόροι έχουν κατανεμηθεί στις UGS, αλλά σίγουρα υπάρχει μια ελάχιστη ποσότητα πόρων. Εάν δεν μπορεί να μεταφερθεί ένα ολόκληρο πακέτο εφαρμόζεται κατακερματισμός. Αφού

εξυπηρετηθούν οι UGS συνδέσεις με σειρά προτεραιότητας, συνεχίζεται η εξυπηρέτηση των BE συνδέσεων με Round Robin. Μετά τις προσομοιώσεις συμπεραίνεται ότι η εκδοχή μπορεί να υποστηρίξει UGS κίνηση μαζί με τις άλλες.

Το [45] παρουσιάζει ένα αλγόριθμο χρονοπρογραμματισμού που βασίζεται στην θεωρία παιγνίων, ο οποίος εκμεταλλεύεται τις Time- Utility Functions (TUF - χρησιμεύουν στους υπολογισμούς πραγματικού χρόνου όταν λήγει μια προθεσμία). Ο αλγόριθμος ακολουθεί μια διαστρωματική προσέγγιση λαμβάνοντας υπόψη και το φυσικό επίπεδο. Η απόφαση του χρονοπρογραμματισμού χωρίζεται σε δύο βήματα, στο πρώτο οι απαιτήσεις των εφαρμογών του συστήματος ικανοποιούνται από το TUF και στο δεύτερο αποφασίζεται η ποσότητα της κίνησης ανά χρήστη που θα εξυπηρετηθεί. Το πρόβλημα του χρονοπρογραμματισμού μοντελοποιείται ως ένα συνεργατικό παιχνίδι μη-μεταφέρσιμης χρησιμότητας (cooperative game with non-transferable utility) όπου κάθε χρήστης αναπαριστά ένα παίκτη, όπου οι παίκτες μπορούν να έχουν ένα κοινό σκοπό δημιουργώντας συμμαχίες και αν κερδίσουν να λάβουν προκαθορισμένο κέρδος. Κάθε χρήστης έχει πολλούς τύπους κίνησης και καθένας αντιπροσωπεύεται από μία ουρά First in First out (FIFO). Για να βρεθούν οι απαιτήσεις της εκάστοτε εφαρμογής ανατίθεται μια TUF ανά τύπο κίνησης. Τα πακέτα ενός συγκεκριμένου τύπου υπηρεσίας συνδέονται με την σχετική TUF, η οποία θα καθορίσει την ποσότητα της χρησιμότητας που φέρει η εξυπηρέτηση του πακέτου και στη συνέχεια συνδέεται η χρησιμότητα του πακέτου με την προθεσμία του. Ο αλγόριθμος χωρίζεται σε δυο τμήματα τον διαταξικό χρονοπρογραμματισμό (inter-class) και τον ενδοταξικό χρονοπρογραμματισμό (intra-class). Στο πρώτο γίνεται κατανομή των πόρων ανάμεσα στους διαφορετικούς τύπους υπηρεσιών και κάθε χρήστης βάζει τα πακέτα του σε λίστα με φθίνουσα σειρά ως προς την χρησιμότητα. Στο δεύτερο κάθε χρήστης έχει δική του στρατηγική και ο αλγόριθμος πρέπει να επιλέξει ποιον συνδυασμό θα εξυπηρετήσει, ανάλογα με την ποιότητα του σήματος και κατά συνέπεια τον αριθμό των σχισμών που θα χρειαστεί. Κάθε τέτοιος συνδυασμός αποτελεί μια συμμαχία. Η Utilitarian λύση δεν είναι δίκαιη διότι αγνοεί τους χρήστες με χαμηλή ποιότητα σήματος, η Nash είναι περισσότερο δίκαιη αφού δεν επιλέγονται οι συμμαχίες με χρήστες που δεν έχουν πακέτα προς αποστολή, επίσης υπάρχουν οι λύσεις Kalai- Smorodinsky και Egalitarian. Τα αποτελέσματα των προσομοιώσεων έδειξαν ότι η Utilitarian λύση εκμεταλλεύεται καλύτερα τους πόρους του δικτύου, η Egalitarian είναι η δικαιότερη από όλες αλλά με πολύ χαμηλή χρήση των πόρων. Καλύτερη αποδείχτηκε η Nash γιατί συνδυάζει και τα δύο. Αναφέρεται ότι η χρήση των TUF είναι χρήσιμη για υπηρεσίες φωνής και βίντεο.

Στο [46] μελετήθηκε ένας αλγόριθμος για τον έλεγχο πρόσβασης, ο οποίος υποστηρίζει τέσσερις τύπους ροών υπηρεσιών (UGS, rt-PS, nrt-PS, BE). Η απόφαση βασίζεται στις μακροπρόθεσμες απαιτήσεις των συνδέσεων και την κατάσταση του δικτύου εκείνη τη στιγμή. Με τη έννοια μακροπρόθεσμες απαιτήσεις εννοείται η εκτίμηση του πιθανού εύρους ζώνης που θα απαιτηθεί κατά όλη τη διάρκεια της μετάδοσης. Ο συχνότερος τύπος είναι ο UGS, το να απορριφθεί μια τέτοια σύνδεση είναι μεγαλύτερο πρόβλημα από το να απορριφθεί οποιαδήποτε άλλη και έτσι έχει υψηλή προτεραιότητα. Ουσιαστικά, η UGS σύνδεση γίνεται αποδεκτή χωρίς κανένα περιορισμό εάν υπάρχει εύρος ζώνης. Οι υπόλοιπες συνδέσεις γίνονται δεκτές μόνο όταν το συνολικό χρησιμοποιούμενο εύρος ζώνης δεν είναι μεγαλύτερο από μια προκαθορισμένη τιμή, εκτός από την BE που γίνεται πάντα αποδεκτή. Η εργασία υποθέτει ότι το σενάριο είναι ομογενοποιημένο και έτσι μπορεί να μελετηθεί ένα μεμονωμένο SS. Ορίζεται ένα ολικό εύρος ζώνης  $B$  και μια προκαθορισμένη τιμή η οποία ισούται με το ολικό εύρος μείον το εύρος ζώνης αποκλειστικό για τις UGS συνδέσεις  $U$ , δηλαδή  $B-U$ . Επίσης, ορίζονται ξεχωριστά οι τιμές εύρους ζώνης που χρειάζονται για τις UGS, rt-PS συνδέσεις, ενώ για την nrt-PS ένα μέγιστο και ένα ελάχιστο. Η UGS γίνεται δεκτή με τον τρόπο που προαναφέρθηκε και αφαιρείται το εύρος ζώνης. Όταν εισέρχεται μια rt-PS ελέγχεται εάν το χρησιμοποιούμενο εύρος όλων των συνδέσεων συν το εύρος της εισερχόμενης δεν υπερβαίνει την προκαθορισμένη τιμή και τότε γίνεται δεκτή, διαφορετικά μειώνεται το εύρος ζώνης όλων των nrt-PS έως ότου ισχύει ο κανόνας αυτός, εάν δεν μπορεί να μειωθεί το εύρος άλλο τότε απορρίπτεται η σύνδεση. Το ίδιο ισχύει και για τις συνδέσεις nrt-PS λαμβάνοντας υπόψη τη μέγιστη απαιτούμενη τιμή.

Το [41] είναι μια περιγραφή για το Mobile WiMAX και συγκεκριμένα το 802.16m και 802.16j. Αναφέρονται οι προδιαγραφές των επιπέδων PHY και MAC, ενώ περισσότερο βάρος δίνεται στο χρονοπρογραμματισμό. Περιγράφεται ο τρόπος αντιμετάθεσης των υποφερουσών, οι τύποι ροής υπηρεσιών και συμπεραίνεται ότι η μελλοντική έρευνα θα εστιαστεί στην κινητικότητα και την διαλειτουργικότητα του προτύπου με άλλες τεχνολογίες.



Η δημοσίευση [47] ασχολείται με τον αλγόριθμο Ομοιόμορφα Δίκαιη Ελλειμματική Εξυπηρέτηση εκ περιτροπής (Uniformly-Fair Deficit Round Robin), ο οποίος είναι μια παραλλαγή της Ασύρματης Ελλειμματικής Εξυπηρέτησης εκ περιτροπής στον οποίο αλλάζει η στρατηγική ανακατανομής. Επίσης στη συγκεκριμένη εργασία γίνεται ανάλυση της απόδοσης μέσω εξαντλητικών προσομοιώσεων με ρεαλιστικά σενάρια σε αντίθεση με άλλες εργασίες που οι προσομοιώσεις τους γίνονται σε συστήματα χωρίς λάθη. Για να είναι καλός ένας αλγόριθμος σε ασύρματο περιβάλλον πρέπει να διαθέτει τα παρακάτω χαρακτηριστικά:

- Βραχυπρόθεσμη δικαιοσύνη ανάμεσα στις συνδέσεις με καλό κανάλι
- Μακροπρόθεσμη δικαιοσύνη ανάμεσα σε όλες τις συνδέσεις
- Εγγύηση του QoS
- Δυνατότητα επιβεβαίωσης της μέγιστης χρησιμοποίησης του ασύρματου καναλιού
- Απλή υλοποίηση και
- Χαμηλή υπολογιστική πολυπλοκότητα

Η Ελλειμματική Εξυπηρέτηση εκ περιτροπής δουλεύει σε κύκλους ως εξής, κάθε κύκλος σημαίνει μια επανάληψη των αποθηκευμένων ουρών. Σε κάθε ροή κατανέμεται μια ποσότητα bits στην αρχή κάθε κύκλου και για να υπάρχει δικαιοσύνη κάθε ροή έχει και ένα αντίστροφο μετρητή, ο οποίος μετρά την ποσότητα των bits που δεν κατάφερε να σταλεί σε προηγούμενους κύκλους. Σε κάθε κύκλο ο μετρητής κάθε μη άδειας ροής αυξάνεται κατά την ποσότητα των bits που έχει κατανεμηθεί στην εκάστοτε ροή και στη συνέχεια επισκέπτεται τις ίδιες ροές και μεταδίδει τόσα bits όσα είναι στο μετρητή. Οποιοδήποτε ποσό bits μένει στο μετρητή μεταφέρετε στον επόμενο κύκλο. Η Ομοιόμορφα Δίκαιη έκδοση του αλγορίθμου λαμβάνει υπόψη την κατάσταση του καναλιού οπότε υπάρχουν clean και dirty συνδέσεις και αλλάζει ανά κύκλο. Επίσης διακρίνει τις ροές σε unbacklogged όταν η ουρά είναι άδεια και backlogged διαφορετικά. Τα κανάλια είναι χωρίς λάθη και υπάρχουν τρεις καταστάσεις για να καθορίζεται ποια ροή υστερεί και ποια ηγείται και ποιά είναι ουδέτερη, lag, lead και insync αντίστοιχα, ενώ εφαρμόζεται μοντέλο αποζημίωσης που επιβάλλει στις ροές που ηγούνται να αποζημιώνουν αυτές που υστερούν για υπηρεσίες που έλαβαν όσο τα κανάλια των δεύτερων ήταν σε κακή κατάσταση. Στην αρχή κάθε κύκλου ξετάζονται όλες οι ροές σειριακά και τους κατανέμονται ανάλογα με την κατάστασή τους οι πόροι. Εάν υπάρχει σύνδεση που ηγείται και ταυτόχρονα υπάρχουν clean backlogged συνδέσεις που υστερούν τότε οι πόροι της πρώτης σύνδεσης ανακατανέμονται στις άλλες συνδέσεις σύμφωνα με την υστέρηση τους. Οι πόροι των dirty συνδέσεων ανακατανέμονται μόνο σε clean backlogged συνδέσεις που υστερούν και εάν δεν υπάρχουν τέτοιες συνδέσεις τότε η ανακατανομή γίνεται σε συνδέσεις insync και τέλος σε αυτές που ηγούνται. Ουσιαστικά ο αλγόριθμος αποτελείται από τα παρακάτω βήματα. Κατανομή πόρων σε όλες τις clean backlogged συνδέσεις που υστερούν και τις insync προσθέτοντας στον αντίστροφο μετρητή της καθεμίας την αντίστοιχη ποσότητα. Στο δεύτερο βήμα χωρίζονται οι clean backlogged συνδέσεις που ηγούνται σε αυτές που ο μετρητής ποσότητας είναι μεγαλύτερος από την ποσότητα που κατανέμεται στη εκάστοτε ροή σε κάθε κύκλο και σε αυτές που ο μετρητής είναι μικρότερος. Στις πρώτες αφαιρείται η προκαθορισμένη ποσότητα ενώ από τις δεύτερες αφαιρείται όσο είναι η τιμή του μετρητή. Προστίθενται όλες οι παραπάνω τιμές και ανακατανέμονται στις clean backlogged συνδέσεις που υστερούν, εάν δεν υπάρχουν τότε συνεχίζει σε αυτές που ηγούνται. Στο τελευταίο βήμα προστίθενται οι ποσότητες bits όλων των dirty συνδέσεων και αυξάνεται η τιμή του μετρητή κατά την προκαθορισμένη ποσότητα της ροής και ανακατανέμεται στις lag, έπειτα στις insync και τέλος στις lead συνδέσεις εάν οι προηγούμενες είναι άδειες. Σύμφωνα με τις προσομοιώσεις συμπεραίνεται ότι ο προτεινόμενος αλγόριθμος είναι πιο αποτελεσματικός στην κατανομή του εύρους ζώνης και δικαιότερος από την Ελλειμματική Εξυπηρέτηση εκ περιτροπής ή την Ασύρματη Ελλειμματική Εξυπηρέτηση εκ περιτροπής. Επιπλέον ανανεώνοντας την κατάσταση της ουράς κάθε χρήστη ανά μερικά πλαίσια μπορεί να χρησιμοποιηθεί και σε ασύρματα ευρυζωνικά δίκτυα.

Στην εργασία [31] αξιολογείται η απόδοση ενός κεντρικού αλγορίθμου χρονοπρογραμματισμού σε δίκτυα mesh στην ανερχόμενη ζεύξη χωρίς QoS. Στην έρευνα αυτή καθορίζονται οι γείτονες του κόμβου που μελετάται, οι οποίοι απέχουν ένα βήμα από αυτόν, οι σπόνσορες του κόμβου δηλαδή ο γειτονικός κόμβος που μεταδίδει μηνύματα από και προς τον BS για λογαριασμό του. Ο κύκλος ορίζεται ως ο χρόνος που χρειάζεται για την μετάδοση όλης της κίνησης από και προς τον BS κάτω από συγκεκριμένο μοντέλο κίνησης και το μήκος της ζεύξης ως ο αριθμός των σχισμών στο κύκλο. Ένας κύκλος επαναλαμβάνεται μέχρι να ανανεωθεί ο χρονοπρογραμματισμός. Ο SS λαμβάνει μια σκυτάλη ανάλογα με τις απαιτήσεις

του. Ένας κόμβος προγραμματίζεται όταν ο αριθμός των σκυταλών που διαθέτει είναι μεγαλύτερος από το μηδέν. Κάθε φορά που δίνεται σε ένα SS μια σχισμή μειώνεται ο αριθμός των σκυταλών του κατά ένα, ενώ του παραλήπτη αυξάνεται κατά ένα για να μπορέσει να αναμεταδώσει το σήμα. Οι προσομοιώσεις έδειξαν ότι εάν δίνεται μεγαλύτερη προτεραιότητα στους κοντινότερους στον BS κόμβους τότε μειώνεται το μέγεθος της καθυστέρησης του χρονοπρογραμματισμού και της μετάδοσης και παράλληλα βελτιώνεται ο ρυθμός χρησιμοποίησης των καναλιών.

Ο Πίνακας 4 περιέχει μια συνοπτική περιγραφή των ανωτέρω εργασιών.

Εργασία	Ροές Υπηρεσιών	Κατεύθυνση Ζεύξης	Περιγραφή
J.F. Borin et al. [30]	UGS, ertPs, rtPS, nrtPS, BE	Ανερχόμενη	Τρεις ουρές: high (UGS, ertPS), intermediate (rtPS, nrtPS) και low (BE). Της δεύτερης ουράς πακέτα στέλνονται στη πρώτη για εγγύηση μέγιστης καθυστέρησης και ελάχιστου εύρους ζώνης.
A. Belghith et al. [43]	UGS, rtPS, BE	Ανερχόμενη	Round Robin αλγόριθμο με κοστολόγηση των υπηρεσιών
S. Pizzi et al. [40]	UGS, ertPs, rtPS, nrtPS, BE	Ανερχόμενη	DRR διαχείριση των ουρών εκτός των UGS και ertPS, σύμφωνα με την κατάσταση των καναλιών. Τρεις διαφορετικές εκδόσεις: ευκαιριακός, βασισμένος στην αποζημίωση και υβριδικός.
J. Liang et al. [44]	rtPS, nrtPS	Ανερχόμενη	Κατανομή πόρων βάσει: <ul style="list-style-type: none"> <li>• Εύρος ζώνης που ζητήθηκε και διάστημα αναμονής (ταξινόμηση με προτεραιότητα)</li> <li>• Προσαρμογή διαστήματος διαδοχικής διερεύνησης.</li> </ul>
A. Bestettiet al. [39]	UGS, ertPs, rtPS, nrtPS, BE	Ανερχόμενη	Αλγόριθμος κούβας με σκυτάλες, στη UGS φτάνουν περιοδικά ενώ στη BE υπάρχει τουλάχιστον ελάχιστο εύρος ζώνης διαθέσιμο.
R.G. Garroppo et al. [45]	UGS, BE	Ανερχόμενη	TUF, θεωρία παιγνίων
H. Wang et al. [46]	UGS, rtPS, nrtPS, BE	Ανερχόμενη	Έλεγχος πρόσβασης, η απόφαση βασίζεται στις μακροπρόθεσμες απαιτήσεις πόρων.
K. Fall et al. [64]	-	Ανερχόμενη	Uniformly Fair DRR, δουλεύει σε κύκλους επαναλαμβάνοντας όλες τις ουρές και κάνει χρήση αντίστροφων μετρητών bits που δεν στάληκαν τους οποίους μεταφέρει στον επόμενο κύκλο.
B. Han et al. [31]	-	Ανερχόμενη	Τοπολογία mesh με σκυτάλες.

Πίνακας 4 – Σχετικές Εργασίες για χρονοπρογραμματισμό στο WiMAX

## 4 QUALITY OF EXPERIENCE

Σήμερα όλο και περισσότεροι πάροχοι δικτύου και ερευνητές εστιάζουν την προσοχή τους στην Ποιότητα Εμπειρίας (QoE) των παρεχόμενων υπηρεσιών τους, αντί της Ποιότητας Υπηρεσιών (QoS), δεδομένου ότι το υψηλό QoS δεν ταυτίζεται υποχρεωτικά με υψηλό QoE. Σε αντίθεση με το QoS και τις αντικειμενικές παραμέτρους του, το QoE ορίζεται ως μια υποκειμενική μέτρηση της εμπειρίας του πελάτη, χρησιμοποιώντας μια υπηρεσία, όπως η περιήγηση στο web, Video on Demand, τηλεφωνική κλήση ή IPTV. Όλα τα συστήματα που βασίζονται σε QoE προσπαθούν να μετρήσουν πελατοκεντρικές παραμέτρους, που σημαίνει δεδομένα που οι συνδρομητές αντιλαμβάνονται άμεσα ως παράμετρο της ποιότητας. Αυτές οι παράμετροι μπορούν να είναι τόσο τεχνικές όσο και μη, για παράδειγμα, η κάλυψη ενός δικτύου, η επιτυχής είσοδος σε μια ιστοσελίδα χωρίς επανάληψη ή την επιτυχή λήψη και αναπαραγωγή μουσική / βίντεο. Είναι πραγματικά σημαντικό και χρήσιμο να βασιστεί η ανάπτυξη νέων συστημάτων σε QoE μηχανισμό με στόχο τη βελτίωση των υπηρεσιών και την ικανοποίηση των πελατών. Έτσι, ο μηχανισμός QoE θα πρέπει να λαμβάνεται υπόψη κατά την ανάπτυξη συστημάτων που αλληλεπιδρούν με πελάτες και προσφέρουν υπηρεσίες ή αγαθά. Λίγοι είναι οι μηχανισμοί και τα συστήματα στη βιβλιογραφία που χρησιμοποιούν QoE στην κύρια λειτουργία τους, επίσης υπάρχει μεγάλο ενδιαφέρον για τις μεθόδους και τα συστήματα που χρησιμοποιούνται για τη συλλογή έγκυρων πληροφοριών ως προς το QoE, όπως περιγράφονται συνοπτικά στις παρακάτω εργασίες.

### 4.1 ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ

Πιο συγκεκριμένα στο [48] παρουσιάζεται μια αυτόνομη αρχιτεκτονική που βελτιστοποιεί το QoE σε δίκτυα πολυμέσων από το κόμβο εκκίνησης της υπηρεσίας προς το χρήστη. Αυτός ο μηχανισμός μπορεί επίσης να ανιχνεύσει αυτόνομα τα προβλήματα του δικτύου, όπως η συμφόρηση, τα λάθη bit και να καθορίσει την κατάλληλη διορθωτική δράση, για παράδειγμα, τη μετάβαση σε ένα χαμηλότερο ρυθμό μετάδοσης βίντεο, προσθέτοντας τον κατάλληλο αριθμό FEC πακέτων. Η αρχιτεκτονική είναι τριών στρωμάτων, όπου η διάσταση Παρακολούθησης (Monitor Plane) παρακολουθεί το δίκτυο και συλλέγει πληροφορίες για την τρέχουσα κατάσταση, η διάσταση Γνώσης (Knowledge Plane) χρησιμοποιεί αυτές τις πληροφορίες για να προσδιορίσει τον καλύτερο τρόπο βελτιστοποίησης του QoE και τέλος η διάσταση Δράσης (Action Plane) εφαρμόζει τη λύση που επιλέχθηκε προηγουμένως. Σχεδιάστηκαν δύο διαφορετικοί αλγόριθμοι, ένας αναλυτικός που βασίζεται σε ένα σύνολο εξισώσεων και ένας που βασίζεται σε νευρωνικό δίκτυο. Οι προσομοιώσεις έδειξαν ότι και οι δύο αλγόριθμοι έχουν την δυνατότητα να αυξήσουν την ποιότητα του video και να μειώσουν το ποσοστό απώλειας πακέτων, ωστόσο το νευρωνικό δίκτυο φαίνεται να ανταποκρίνεται καλύτερα αφού υποστηρίζει on-line συμπεριφορά μάθησης. Στην [49] παρουσιάζεται από την ίδια ομάδα ένας παρόμοιος μηχανισμός που χρησιμοποιεί αυτά τα νέα δεδομένα, προκειμένου ενεργοποίηση συναγερμούς προς τα υψηλότερα επίπεδα ή να προτείνει αυτόνομα δράσεις για την αποκατάσταση των χαμηλών QoE επιδόσεων των υπηρεσιών.

Στο έγγραφο [50] προτείνεται ένα δίκτυο επικάλυψης για QoE από άκρο σε άκρο, το οποίο πετυχαίνει βελτιστοποίηση με αλλαγή δρομολόγησης σε περίπτωση αποτυχίας. Αποτελείται από ένα σύνολο διακομιστών, οι οποίοι παρακολουθούν το QoS από άκρο σε άκρο και διατηρούν μια τοπολογία επικάλυψης, που τους επιτρέπει να δρομολογούν πακέτα μακριά από χαλασμένες ζεύξεις και συμφορήσεις. Επίσης οι διακομιστές αυτοί διαχειρίζονται το εύρος ζώνης στο άκρο του δικτύου προς το χρήστη (Last Mile). Η πρόταση αυτή πετυχαίνει βελτιστοποίηση της κλασικής υπηρεσίας δρομολόγησης μειώνοντας το ποσοστό απώλειας πακέτων αλλά και τη χρήση του εύρους ζώνης στο άκρο του χρήστη, γεγονότα που ενισχύουν το QoE.

Από την άλλη το [51] εστιάζει σε streaming video με την παρουσίαση ένα προσαρμοστικού συστήματος, το MARC (Mobile-aware Adaptive Rate Control) ρυθμίζει την ποιότητα bit-stream και τον ρυθμό μετάδοσης ανάλογα με τη κατάσταση του καναλιού που αλλάζει με την κίνηση του τερματικού και την κατάσταση του δικτύου που αλλάζει λόγω συμφορήσεων. Αποτελείται από δύο module:

- NCAR (Network and Client-aware Rate Control), το οποίο υποστηρίζει QoE και διαθέτει δύο συναρτήσεις. Η συνάρτηση ελέγχου συμφόρησης προσαρμόζει το bit-rate βάσει πληροφοριών για το δίκτυο, ενώ ο έλεγχος ροής βάσει πληροφοριών για το χρήστη.

- CLAR (Channel and Location-aware Rate Control), το οποίο μειώνει το bit-rate για να λύσει το πρόβλημα της κατανομής των πόρων που στο κέντρο μιας κυψέλης, όπου είναι μικρότερη η πιθανότητα διαπομπής σε σχέση με τα άκρα της, διαφοροποιώντας τις περιοχές βάσει πληροφοριών για το SIR (Signal-to-Interference Ration), το RSSI (Received Signal Strength Information) και το CINR (Carrier to Interference-and-Noise Ratio).

Οι προσομοιώσεις έδειξαν ότι μπορεί το σχήμα να ελέγχει το ρυθμό μετάδοσης των πολυμέσων βάσει της κατάστασης του τερματικού του χρήστη, ενώ παρέχει real-time playback χωρίς διακοπές κατά την διάρκεια της κίνησης.

Στο [52], το σύστημα χρησιμοποιεί ένα κινητό πράκτορα QoS που είναι εγκατεστημένος στα τερματικά των τελικών χρηστών. Υπάρχουν δύο βασικές μεθοδολογίες για την παρακολούθηση του QoE:

- Μέτρηση της απόδοσης των υπηρεσιών μέσω στατιστικών δειγμάτων
- Σύστημα διαχείρισης δικτύου με χρήση παραμέτρων QoS

Στην εργασία προτείνεται ο κινητός πράκτορας σε συνδυασμό με την πρώτη μεθοδολογία. Συλλέγονται μετρήσεις για την ποιότητα των κινητών πολυμεσικών υπηρεσιών, της παραμέτρους του ασύρματου μέσου και επιστρέφονται στατιστικά για την απόδοση. Τα αποτελέσματα έδειξαν η παραπάνω πρόταση αποτελεί το κλειδί για την βέβαιη λειτουργία των ασύρματων υπηρεσιών.

Στη εργασία [53] η μέθοδος που προτείνεται μια μέθοδος που αξιολογεί το QoE ενός χρήστη σε peer-to-peer σύστημα τηλεόρασης και συγκεκριμένα στο Joost. Οι παράμετροι που παίζουν ρόλο στο QoE για ένα τέτοιο σύστημα είναι οι εξής:

- η καθυστέρηση έναρξης
- οι διακοπές στην παροχή της υπηρεσίας
- και η ποιότητα του video

Χρησιμοποιήθηκαν δύο μέθοδοι, η reverse engineering για την κατανόηση της αρχιτεκτονικής του Joost και μια υποκειμενική μέθοδος αξιολόγησης για να απομονωθούν οι παράμετροι στις οποίες είναι ευαίσθητο το δίκτυο. Η εργασία ουσιαστικά παρέχει μια εικόνα κάποιων μηχανισμών peer-to-peer streaming και μια μέθοδο για αναγνώριση των αδυναμιών αυτών των συστημάτων.

Στη [54] παρουσιάζονται οι προκλήσεις και μια πιθανή θεωρητική λύση για την παροχή QoE. Η πρόταση αποτελείται από δύο βασικά τμήματα:

- την έκθεση επιδόσεων QoE/ QoS που μετρά το QoE και το QoS όπως λαμβάνονται από τους χρήστες
- τη διαχείριση QoE που προσαρμόζει τις μεθόδους μετάδοσης και επαναπροσδιορίζει παραμέτρους του στρώματος εφαρμογής.

Στο [55] προτείνεται ένας χρονοπρογραμματιστής που λαμβάνει υπόψη του το QoE. Σε σύγκριση με τους υπάρχοντες χρονοπρογραμματιστές έχει κερδοφόρα απόδοση ως προς την αποδοχή του χρήστη, το ρυθμό μετάδοσης των δεδομένων και την δικαιοσύνη. Χρησιμοποιείται η τεχνική Pseudo- Subjective Quality Assessment (PSQA) που βασίζεται σε στατιστική μάθηση χρησιμοποιώντας ένα Τυχαίο Νευρωνικό Δίκτυο. Προτείνονται δύο αλγόριθμοι, ο QoE – CI που βασίζεται στο Carrier-to-Interference Ratio για να μεγιστοποιήσει το ρυθμό μετάδοσης των δεδομένων και ο QoE – PF που βασίζεται στο Proportionally Fair για να μεγιστοποιήσει τη δικαιοσύνη. Η βασική ιδέα είναι να δίνεται μεγαλύτερη προτεραιότητα στους χρήστες που έχουν μεγαλύτερους περιορισμούς ως προς την ποιότητα.

Η εργασία [56] παρουσιάζει ένα μηχανισμό που βασίζεται και πάλι στην τεχνική PSQA για την μέτρηση του QoE από τους χρήστες. Το σχήμα αποφασίζει σε ποίο δίκτυο θα συνδεθεί κάθε χρήστης. Οι παράμετροι που ελέγχονται είναι ο ρυθμός απώλειας πλαισίων και το μέσο μήκος μιας σειράς από συνεχόμενα χαμένα πακέτα. Το σημείο πρόσβασης παρακολουθεί τις απώλειες και επικοινωνεί με το PSQA . οι χρήστες μπορούν να επιλέξουν να μην συνδεθούν με κανένα δίκτυο αν οι συνθήκες δεν επιτρέπουν τη ικανοποίηση των ελάχιστων απαιτήσεών τους. Τα αποτελέσματα έδειξαν ότι το σχήμα αποδίδει και παρέχει υψηλό QoE στους νέους χρήστες . το μόνο μειονέκτημα είναι η πολυπλοκότητα του PSQA, το οποίο πρέπει να εφαρμόζεται προσεκτικά για κάθε νέα εφαρμογή.

Στην επόμενη δημοσίευση [57] συγκρίνεται το QoE για web streaming ως προς το WiFi και το HSDPA, χρησιμοποιώντας CUBIC TCP, το οποίο εφαρμόζεται στα Android τηλέφωνα. Συμπληρώνονται οι εκτιμήσεις QoE από τις παραμέτρους QoS, όπως ο ρυθμός μετάδοσης, η καθυστέρηση και τα στατιστικά στοιχεία του στρώματος μετάδοσης, αποκαλύπτοντας τις δυναμικές του cross layer. Τα αποτελέσματα έδειξαν ότι:

- το QoE για video παραμένει σταθερό στο WiFi ακόμα και με μεγάλη απώλεια πακέτων ενώ είναι ευαίσθητο στο HSDPA ακόμα και για μικρές απώλειες λόγω των μεγάλων διαφοροποιήσεων στο ρυθμό και την καθυστέρηση
- η μείωση του QoE και του QoS στο HSDPA οφείλεται στις αρνητικές αλληλεπιδράσεις του με τον επιθετικό τρόπο ελέγχου των συμφορήσεων που επιβάλλει το CUBIC TCP
- και η διαπομπή από WiFi σε HSDPA μειώνει το QoE.

Για την προσομοίωση χρησιμοποιήθηκε το QoE-Lab που περιγράφεται στην εργασία [58].

Στην [59] προτείνεται ένας μηχανισμός ελέγχου εισόδου βασισμένος στο QoE, το οποίο μετράται μέσω του PSQA. Αριθμητικό μέτρο ποιότητας παρέχει το μέσο αποτέλεσμα άποψης (Mean Opinion Score – MOS). Το PSQA συνδυάζει το αντικειμενικό και υποκειμενικό τρόπο εκτίμησης της ποιότητας, αφού μαθαίνει να χαρτογραφεί το QoE αποτέλεσμα στις τεχνικές παραμέτρους. Το PSQA επικοινωνεί με το σημείο πρόσβασης, το οποίο υπολογίζει το μέσο MOS και αναλόγως επιτρέπει ή απορρίπτει μια νέα σύνδεση. Σύμφωνα με την προσομοίωση είναι καλύτερο από άλλα σχήματα χωρίς έλεγχο εισόδου και ως προς το MOS αλλά και την διαπερατότητα.

Η [60] ερευνά ερευνώνται η επιρροή των αισθητήριων εφέ σε διαφορετικό bit-rate και κατά συνέπεια της ποιότητας όσον αφορά το καλύτερο λόγο signal-to-noise. Χρησιμοποιήθηκε το test-bed που δημιουργήθηκε από την ίδια ομάδα στην εργασία [61]. Τα αποτελέσματα έδειξαν ότι οι πληροφορίες αυτές αποτελούν ένα κατάλληλο εργαλείο για την αύξηση του QoE διεγείροντας άλλες αισθήσεις του χρήστη εκτός της όρασης και της ακοής, όπως η όσφρηση. Επίσης για κάθε διαφορετικό bit-rate η τιμή του MOS είναι υψηλότερη σε σχέση με το αντίστοιχο σενάριο χωρίς αισθητήρια εφέ.

Η “επαυξημένη” πραγματικότητα (Augmented Reality) ενισχύει την εμπειρία του χρήστη προσθέτοντας πραγματικές πληροφορίες σε αυτές που παράγει ο υπολογιστής και εφαρμόζεται κυρίως στην παρουσίαση 3D εικονικών αντικειμένων. Η εργασία [62] προτείνει ένα μοντέλο που συνδέει αυτές τις πληροφορίες με τις παραμέτρους του QoE. Οι μεταβλητές που λαμβάνονται υπόψη είναι το μέγεθος της εικόνας, ο ρυθμός αλλαγής των πλαισίων, η καθυστέρηση και η ταχύτητα κίνησης της κεφαλής. Το μοντέλο χρησιμοποιείται για να καθοριστούν ο βέλτιστος συνδυασμός των ανωτέρω μεταβλητών. Σύμφωνα με τα αποτελέσματα το μοντέλο πετυχαίνει το μέγιστο QoE.

Ο Πίνακας 5 περιέχει μια συνοπτική περιγραφή των ανωτέρω εργασιών.

Εργασία	Περιγραφή
S. Latréa et al. [48]	Αρχιτεκτονική τριών διαστάσεων: Παρακολούθησης, Γνώσης και Δράσης. Η δεύτερη λαμβάνει πληροφορίες από την πρώτη σχετικά με την κατάσταση του δικτύου και μέσω νευρωνικού δικτύου αποφασίζει μια λύση για να πραγματοποιηθεί από την τρίτη διάσταση.
S. Latre et al. [49]	Παρόμοια με την προηγούμενη, μόνο που με τις νέες πληροφορίες ενεργοποιεί συναγερμούς σε ανώτερα στρώματα.
B. De Vleeschauwer et al. [50]	Δίκτυο επικάλυψης το οποίο πετυχαίνει βελτιστοποίηση με αλλαγή δρομολόγησης όταν υπάρχουν συμφορήσεις ή χαλασμένες ζεύξεις.
J. Koo et al.[51]	Ρυθμίζει το bit-stream και το ρυθμό μετάδοσης ανάλογα με την κατάσταση του καναλιού.
D. Soldani et al.[52]	Προτείνει κινητό πράκτορα QoS για τα τερματικά των χρηστών που θα μετρά

	την απόδοση.
F. Agboma et al. [53]	Αξιολογεί το QoE σε σύστημα peer-to-peer τηλεόρασης.
J. Zhang et al. [54]	Προκλήσεις και πιθανή θεωρητική λύση για παροχή QoE με μέτρηση QoE και QoS και διαχείριση του QoE με προσαρμογή της μεθόδου μετάδοσης.
K. Piamrat et al. [55]	Χρήση PSQA για μέτρηση του QoE και πρόταση δύο αλγορίθμων: QoE βάση του Carrier to Intereference, QoE βάση του Proportionally Fair αλγορίθμου.
K. Piamrat et al. [56]	Χρήση PSQA για μέτρηση του QoE. Οι χρήστες επιλέγουν σύμφωνα με τις πληροφορίες για το QoE που λαμβάνουν από το σταθμό αν θα συνδεθούν ή όχι και σε ποίο δίκτυο.
M.A. Mehmood et al. [57]	QoE για web streaming στο WiFi και το HSDPA, με χρήση ενός testbed QoE-Lab που δημιούργησε η ομάδα σε παλιότερη εργασία της.
K. Piamrat et al. [59]	Μηχανισμός ελέγχου εισόδου, με χρήση PSQA για μέτρηση του QoE. Ο σταθμός αποφασίζει ανάλογα με τις μετρήσεις αν θα επιτραπεί η θα απορριφθεί μια νέα σύνδεση.
M. Walzl et al. [60]	Με ειδικό test-bed που δημιούργησε η ομάδα σε προηγούμενη εργασία της για μέτρηση του QoE, εδώ μελετά την επιρροή των αισθητήριων εφέ σε διαφορετικό bit-rate.
D. Perritaz et al. [62]	Μοντέλο που συνδέει τις πραγματικές πληροφορίες, τις οποίες προσθέτει η "επαυξημένη πραγματικότητα" σε αυτές του υπολογιστή, με τις παραμέτρους του QoE, για την μεγιστοποίησή του.

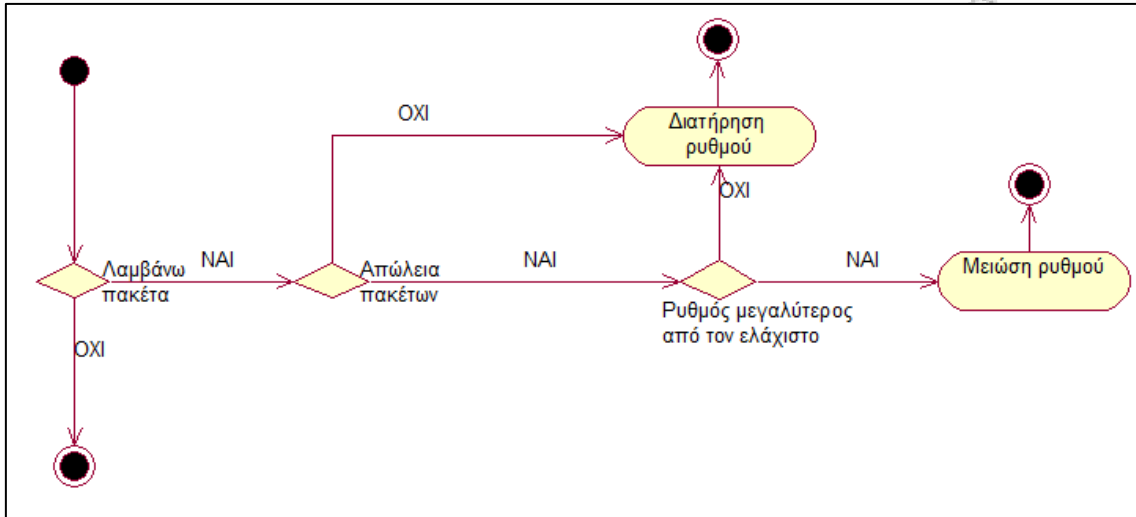
Πίνακας 5 - Σχετικές Εργασίες για χρονοπρογραμματισμό με QoE

## 4.2 ΠΡΟΤΕΙΝΟΜΕΝΟ ΜΟΝΤΕΛΟ

Η συνεισφορά της εργασίας έγκειται στην εφαρμογή QoE, όπου ουσιαστικά κάθε χρήστης έχει διαφορετική υποκειμενική απαίτηση από το σύστημα ως προς την ποιότητα της υπηρεσίας που χρησιμοποιεί. Λόγω της δημοτικότητας επικεντρώνεται στο video streaming και ουσιαστικά στον rtPS τύπο υπηρεσιών. Προτείνονται στον χρονοπρογραμματιστή λοιπόν δύο διαφορετικές QoE προσεγγίσεις.

### 4.2.1 ΠΟΛΙΤΙΚΗ 1

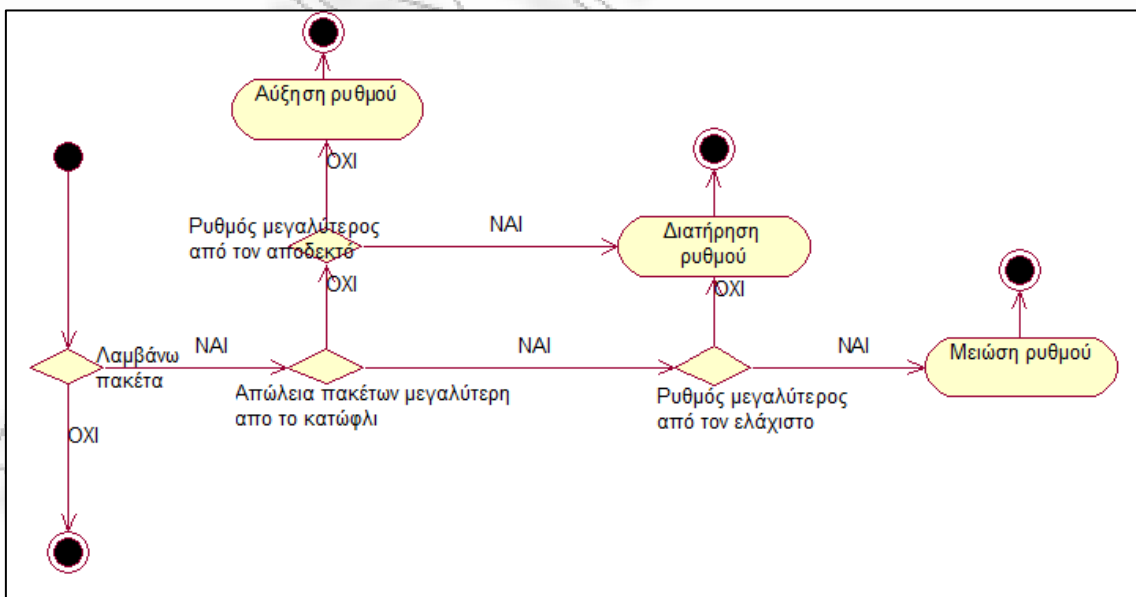
Στο πρώτο σχήμα χρησιμοποιήθηκε QoE δύο επιπέδων, όπου κάθε χρήστης έχει ένα αρχικό μέγιστο ρυθμό μετάδοσης και μία ελάχιστη υποκειμενική απαίτηση. Ο χρονοπρογραμματιστής λειτουργεί ως εξής, ξεκινά κάθε κόμβος να στέλνει κίνηση με το μέγιστο ρυθμό του. Όταν υπάρχει απώλεια πακέτων σε οποιοδήποτε χρήστη τότε ελέγχεται κάθε χρήστης και αν ο ρυθμός μετάδοσης είναι μεγαλύτερος από τον ελάχιστο, τότε μειώνεται το ρυθμό του, διαφορετικά τον παραμένει στο ίδιο επίπεδο. Ο ρυθμός επανέρχεται στην αρχική μέγιστή τιμή του κατά τη διάρκεια της προσομοίωσης. Συγκεκριμένα η επαναφορά γίνεται κάθε 18 δευτερόλεπτα ενώ παρατηρήθηκε ότι χρειάζονται 15 δευτερόλεπτα για να φτάσουν όλοι οι χρήστες στον ελάχιστο ρυθμό μετάδοσης. Στην Εικόνα 4.1 παρουσιάζεται η διαδικασία.



Εικόνα 4.1 – Διάγραμμα δραστηριοτήτων Πολιτικής 1

**4.2.2 ΠΟΛΙΤΙΚΗ 2**

Στο δεύτερο σχήμα χρησιμοποιήθηκε QoE τριών επιπέδων, όπου κάθε χρήστης έχει ένα αρχικό μέγιστο ρυθμό μετάδοσης, μια μέση υποκειμενική αποδεκτή τιμή και μία ελάχιστη υποκειμενική απαίτηση. Και πάλι οι χρήστες ξεκινάνε να στέλνουν με το μέγιστο ρυθμό. Ο χρονοπρογραμματιστής σε αυτή τη πολιτική λειτουργεί ως εξής, όταν υπάρχει απώλεια πακέτων μεγαλύτερο από ένα κατώφλι που επιλέγεται κατά την εκτέλεση του προσομοιωτή, τότε ελέγχεται κάθε χρήστης και αν ο ρυθμός μετάδοσης του είναι μεγαλύτερος από τον ελάχιστο, τότε μειώνεται ο ρυθμός, διαφορετικά παραμένει στο ίδιο επίπεδο. Αν από την άλλη το ποσοστό απώλειας για ένα κόμβο είναι μικρότερο από το όριο που έχει τεθεί τότε σε αυτούς τους χρήστες ελέγχεται αν ο ρυθμός μετάδοσης είναι μικρότερος από τον αποδεκτό και αυξάνεται ο ρυθμός, διαφορετικά παραμένει και πάλι ως είχε. Το κατώφλι που μπορεί να επιλέξει ο χρήστης είναι ένα ποσοστό επί του ρυθμού μετάδοσης των δεδομένων του εκάστοτε χρήστη, δηλαδή αν στο πρόγραμμα εισαχθεί η τιμή 20 τότε το κατώφλι για την απώλεια πακέτων είναι το 20% του ρυθμού μετάδοσης κατά την διάρκεια του ελέγχου. Στην Εικόνα 4.2 φαίνεται το διάγραμμα δραστηριοτήτων.



Εικόνα 4.2 Διάγραμμα δραστηριοτήτων Πολιτικής 2

## 5 ΠΕΡΙΒΑΛΛΟΝ ΠΡΟΣΟΜΙΩΣΗΣ

Ο Network Simulator - 2 (NS- 2) είναι ένας αντικειμενοστρεφής προσομοιωτής διακριτών γεγονότων, γραμμένος σε C++ και οTcl. Το NS- 2 αναπτύχθηκε για το Information Sciences Institute of University of Southern California και για πρώτη φορά κυκλοφόρησε το 1996. Προέρχεται από τους παλαιότερους προσομοιωτές REAL από το Keshav και NS- 1 από την McCanne, Floyd και Fall. Η πρώτη έκδοση του υποστηρίζεται από το DARPA (Defense Advanced Research Projects Agency), μέσω του προγράμματος Vint (Virtual Internetwork Testbed) των LBL, Xerox PARC, UCB και USC/ ISI. Το DARPA Vint διήρκεσε από το 1997 έως το 2000 και επικεντρώθηκε στην ανάπτυξη του πυρήνα του κώδικα προσομοίωσης για τον NS- 2 και το nam. Από το 2000 έως το 2004, η ανάπτυξη του NS- 2 χρησιμοποίησε δύο προγράμματα το SAMAN (Simulation Augmented by Measurement and Analysis for Network) του DARPA (Defense Advanced Research Projects Agency) και το CONSER (Collaborative Simulation for Education and Research) της NSF (National Sciences Foundation) για τη χρηματοδότηση δραστηριοτήτων συντήρησης κώδικα. Από το 2004, δεν έχει χρηματοδοτηθεί. Το 2005, ένας από τους κύριους ερευνητές, ο Henderson, άρχισε να διατηρεί τις λίστες ηλεκτρονικού ταχυδρομείου των χρηστών και των προγραμματιστών, συντόνισε την προσπάθεια για την χορήγηση αδειών του πηγαίου κώδικα για ελεύθερη χρήση του και ξεκίνησε την φιλοξενία του NS- 2 στο δικτυακό τόπο Sourceforge. Από το 2005 ο προσομοιωτής αλλά και οι ενότητες του έχουν άδειες χρήσης ελεύθερου λογισμικού GNU General Public License έκδοση 2. Το NS έχει συμπεριλάβει και συνεισφορές από άλλους ερευνητές, μεταξύ αυτών κώδικα για ασύρματη δικτύωση από την UCB Daedelus, την CMU Monarch και την Sun Microsystems.

Λόγω του ότι είναι ανοικτού κώδικα είναι επεκτάσιμο και μπορεί να χρησιμοποιηθεί σε επίπεδο προγραμματιστή αλλά και χρήστη. Ο NS χρησιμοποιείται στην προσομοίωση των πρωτοκόλλων δρομολόγησης, μεταξύ άλλων, και σε μεγάλο βαθμό χρησιμοποιείται σε έρευνες ad-hoc δικτύωσης. Υποστηρίζει προσομοίωση πρωτόκολλων μεταφοράς, δρομολόγησης μονοεκπομπής (unicast) και πολυεκπομπής (multicast) πάνω από ενσύρματα και ασύρματα δίκτυα τόσο τοπικά όσο και δορυφορικά. Ο προσομοιωτής είναι χρήσιμος για έρευνες αφού επιτρέπει την πρόσβαση σε μοντέλα με στοιχεία, στα οποία δεν θα υπήρχε διαφορετικά πρόσβαση, επίσης επιτρέπει την τροποποίηση σχεδόν όλων των παραμέτρων που επηρεάζουν την κατάσταση του δικτύου. Παράλληλα είναι χρήσιμο και για μαθητές εφόσον επιτρέπει την προσομοίωση απλών δικτύων που βοηθούν στην κατανόηση των διαφορετικών πρωτόκολλων.

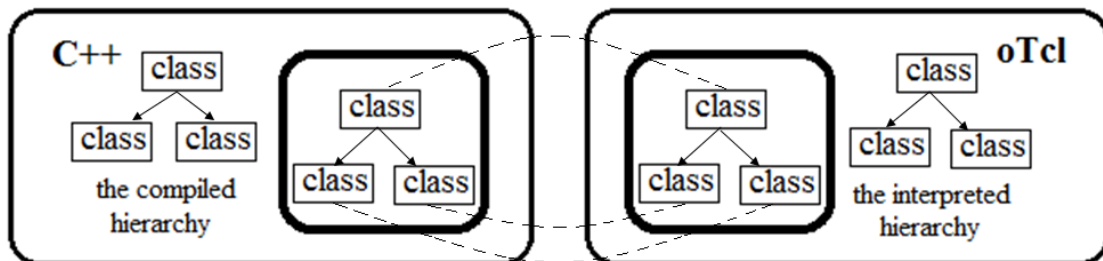
Ο προσομοιωτής βασίζεται εξ ολοκλήρου σε δύο γλώσσες τις OTcl και C++. Ο πυρήνας είναι γραμμένος σε C++, ενώ τα σενάρια προσομοίωσης είναι γραμμένα σε OTcl. Η ιεραρχία των τάξεων σε C++ (ή compiled ιεραρχία) είναι παρόμοια με την ιεραρχία τάξεων μέσα στο OTcl διερμηνέα (ή ιεραρχία διερμηνέα), όπως φαίνεται και στην Εικόνα 5.1. Οι χρήστες δημιουργούν αντικείμενα προσομοίωσης μέσω του διερμηνέα, τα οποία αρχικοποιούνται σε αυτόν και απεικονίζονται στενά σε ένα αντίστοιχο αντικείμενο στην ιεραρχία της C++. Οι τάξεις της ιεραρχίας του διερμηνέα δημιουργούνται μέσω μεθόδων που έχουν καθοριστεί στην τάξη TclClass. Ο NS- 2 παρέχει ένωση μεταξύ των δύο γλωσσών μέσω της tclcl για να κάνει τα αντικείμενα και τις μεταβλητές να εμφανίζονται και στις δύο γλώσσες. Η C++ είναι γρήγορη στην εκτέλεση, αλλά δεν ενδείκνυται για αλλαγές και επανέναρξη των προσομοιώσεων και είναι ιδανική για την αποδοτική επεξεργασία των στοιχείων. Αντίθετα η OTcl είναι αργή, αλλά επιτρέπει γρήγορες και διαδραστικές αλλαγές και είναι ιδανική για τη διαμόρφωση σεναρίων προσομοίωσης. Έτσι, η OTcl χρησιμοποιείται για οποιαδήποτε διαμόρφωση και εγκατάσταση γίνεται μία μόνο φορά και όταν μπορεί να γίνει κάτι χρησιμοποιώντας τα υπάρχοντα αντικείμενα C++. Η C++ χρησιμοποιείται για οτιδήποτε χρειάζεται επεξεργασία κάθε πακέτου μιας ροής και όταν χρειάζεται αλλαγή της υπάρχουσας συμπεριφοράς των τάξεων της με τρόπους που δεν είχαν προβλεφθεί. Βέβαια υπάρχει και κώδικας για τον οποίον δεν ισχύουν τα παραπάνω, όπως για παράδειγμα το μεγαλύτερο μέρος της δρομολόγησης γίνεται σε OTcl, ενώ ο πυρήνας του αλγόριθμου Dijkstra είναι σε C++.

Η διεπαφή του διερμηνέα βρίσκεται σε ξεχωριστό φάκελο tclcl ενώ ο κώδικας του υπόλοιπου προσομοιωτή βρίσκεται στο ns-2. Στην συνέχεια αναφέρονται έξι τάξεις από την tclcl:

- Τάξη TCL, περιέχει μεθόδους που χρησιμοποιεί η C++ για να έχει πρόσβαση στον διερμηνέα, για παράδειγμα παρέχει αναφορά στο στιγμιότυπο Tcl (Tcl& tcl=Tcl::instance();), καλεί OTcl διαδικασίες μέσω του διερμηνέα (tcl.eval), λαμβάνει από



- ή στέλνει πίσω στον διερμηνέα αποτελέσματα (tcl.result), αναφέρει λάθη και κάνει έξοδο (tcl.error), αποθηκεύει και βρίσκει TclObjects (tcl.enter, tcl.lookup, tcl.remove), επιστρέφει τον έλεγχο στην τάξη Tcl (tcl.interp)
- Τάξη TclObject, είναι η βασική τάξη όλων των αντικειμένων προσομοίωσης που απεικονίζονται στην ιεραρχία της C++. Κάθε αντικείμενο της τάξης δημιουργείται από τον χρήστη μέσω του διερμηνέα και στην συνέχεια ένα ισοδύναμο αντικείμενο “σκιά” δημιουργείται στην ιεραρχία της C++. Οι μεταβλητές των δύο ιεραρχιών είναι ξεχωριστές και η αλλαγή της μίας δεν επηρεάζει την άλλη, υπάρχει όμως τρόπος να συνδεθούν ώστε να αλλάζουν ταυτόχρονα μέσω της bind.
  - Τάξη TclClass, καθορίζει την ιεραρχία των τάξεων που διερμηνεύονται και τις μεθόδους που επιτρέπουν στον χρήστη να αρχικοποιήσει TclObjects.
  - Τάξη TclCommand, καθορίζει απλές global εντολές του διερμηνέα και έχει δύο τρόπους λειτουργίας την ns-random και την ns-version.
  - Τάξη EmbeddedTcl, περιέχει μεθόδους για “φόρτωση” builtin εντολών υψηλότερου επιπέδου που κάνουν την διαμόρφωση των προσομοιώσεων ευκολότερη.
  - Τάξη InstVar, παρέχει μεθόδους για πρόσβαση μελών μεταβλητών της C++ ως στιγμιότυπα μεταβλητών OTcl, δηλαδή καθορίζει τις μεθόδους και τους μηχανισμούς για να γίνεται bind μια C++ μεταβλητή ενός αντικειμένου “σκιά” ενός συγκεκριμένου στιγμιότυπου OTcl μεταβλητής στο αντίστοιχο αντικείμενο του διερμηνέα.



### C++ and oTcl Linkage

Εικόνα 5.1 – NS- 2 σύνδεση C++ και OTcl

Όλος ο προσομοιωτής περιγράφεται από μια Tcl τάξη, την Simulator. Οι διαδικασίες και λειτουργίες βρίσκονται στα ns-lib.tcl, scheduler.cc, scheduler.h και heap.h. Ο προσομοιωτής είναι event-driven και υπάρχουν τέσσερις διαθέσιμοι χρονοπρογραμματιστές οι simple linked-list, heap, calendar queue και real-time. Ο χρονοπρογραμματιστής επιλέγει το επόμενο πρόσφατο γεγονός το εκτελεί και επιστρέφει για το επόμενο.

Το NS- 2 υποστηρίζει την παρακολούθηση των γεγονότων, τα οποία είτε εμφανίζονται στην οθόνη του χρήστη κατά την εκτέλεση της προσομοίωσης είτε αποθηκεύονται σε κάποιο αρχείο. Υπάρχουν δύο διακριτές δυνατότητες παρακολούθησης:

- Τα ίχνη (traces), όπου γίνεται καταγραφή κάθε ξεχωριστού πακέτου κατά την άφιξη, αναχώρηση ή απόρριψη του σε μια ζεύξη ή μια ουρά.
- Η παρακολούθηση (monitor), όπου γίνεται καταγραφή καταμετρήσεων διάφορων πληροφοριών που παρουσιάζουν ενδιαφέρον όπως η άφιξη, αναχώρηση, απόρριψη πακέτων και byte. Μπορεί να γίνει η παρακολούθηση των πακέτων ανά ροή ή όλων συνολικά.

Η στρατηγική δρομολόγησης είναι ένας γενικός μηχανισμός με τον οποίο ο NS- 2 υπολογίζει διαδρομές για την προσομοίωση. Υποστηρίζεται η δρομολόγηση μονοεκπομπής με τέσσερις στρατηγικές τις static, session, dynamic και manual. Ταυτόχρονα υποστηρίζεται και η δρομολόγηση πολυεκπομπής. Όταν ενεργοποιούνται οι επεκτάσεις πολυεκπομπής, οι κόμβοι που δημιουργούνται στην τοπολογία έχουν επιπλέον classifiers και replicators για να πραγματοποιούνται οι προωθήσεις και οι ζεύξεις περιέχουν στοιχεία που αναθέτουν ταμπέλες σε όλα τα πακέτα. Στη δρομολόγηση πολυεκπομπής υποστηρίζονται τρεις στρατηγικές οι centralized, dense mode και shared tree mode.

Οι agents αντιπροσωπεύουν σημεία στα οποία δημιουργούνται και καταστρέφονται πακέτα. Οι εφαρμογές είναι πάνω από τους transport agents και υπάρχουν δύο τύποι, οι traffic

generators και οι simulated applications. Ο προσομοιωτής παρέχει διαφορετικούς μηχανισμούς στις ουρές των δρομολογητών όπως Drop Tail (FIFO), RED, CBQ, διάφορες εκδόσεις του Fair Queuing και Deficit Round Robin. Παράλληλα είναι ενσωματωμένα σε αυτόν τα γνωστότερα δικτυακά πρωτόκολλα, προσομοιώσεις διαχείρισης δρομολόγησης όπως για παράδειγμα:

- Το TCP και UDP.
- Επιτρέπει τον καθορισμό επίγειων, ασύρματων και δορυφορικών δικτύων με πολλαπλούς αλγορίθμους δρομολόγησης (DV, LS, PIM-DVM, PIM-SIM, AODV, DSR)

Το NS- 2 παρέχει επιπλέον μια γραφική διεπαφή για την προβολή των προσομοιώσεων που ονομάζεται Nam. Το Nam έχει ένα γραφικό επεξεργαστή, ο οποίος επιτρέπει την δημιουργία animations χωρίς την χρήση TCL κώδικα. Μπορεί να δημιουργηθεί η τοπολογία του δικτύου και να προσομοιωθούν διάφορα πρωτόκολλα και πηγές κίνησης χρησιμοποιώντας το ποντίκι.

Το NS- 2 επιτρέπει την εμφάνιση της ροής των πακέτων, της πιθανής απόρριψής τους και επανένταξης τους στην ουρά, υποστηρίζει πρωτόκολλα συμπεριφοράς, κίνηση των κόμβων σε ασύρματα δίκτυα, ενώ παράγει εγγραφές για τα σημαντικότερα γεγονότα.

Αναφέρεται επίσης η δυνατότητα εξομοίωσης (emulation) που προσφέρεται. Η εξομοίωση είναι ουσιαστικά η δυνατότητα παρουσίασης του προσομοιωτή σε πραγματικό δίκτυο. Ειδικά αντικείμενα έχουν την δυνατότητα να παράγουν πραγματική κίνηση μέσα στον προσομοιωτή και να την μεταφέρουν από αυτόν στο δίκτυο. Τα στοιχεία που περιγράφονται παρακάτω, δεν αναμένεται να αλλάξουν δραστικά, όμως η εξομοίωση είναι ακόμη υπό ανάπτυξη, θεωρείται έτσι πειραματική και υπόκεινται σε αλλαγές. Η εξομοίωση μπορεί να διαιρεθεί σε δύο τρόπους, τους εξής:

- opaque mode, όπου πραγματικά δεδομένα αντιμετωπίζονται ως αδιαφανή πακέτα δεδομένων
- protocol mode, όπου πραγματικά δεδομένα μπορούν να ερμηνευθούν/ δημιουργηθούν από τον προσομοιωτή.

Στην opaque λειτουργία, ο προσομοιωτής αντιμετωπίζει τα δεδομένα του δικτύου, ως uninterpreted πακέτα. Ειδικότερα, στον πραγματικό κόσμο τα πεδία του πρωτοκόλλου δεν χειραγωγούνται από τον προσομοιωτή, στον οποίο τα πακέτα μπορούν να απορριφθούν, καθυστερήσουν, επαναπρογραμματιστούν ή αναπαραχθούν αλλά επειδή δεν γίνεται καμία επεξεργασία, συγκεκριμένα σενάρια χειραγώγησης της κίνησης δεν μπορούν να εκτελεστούν. Στη λειτουργία πρωτοκόλλου, ο προσομοιωτής είναι σε θέση να ερμηνεύσει ή/ και να παράγει πραγματική κίνηση δικτύου που περιέχουν αυθαίρετες αναθέσεις στα πεδία του πρωτοκόλλου. Η διασύνδεση μεταξύ του προσομοιωτή και του πραγματικού δικτύου παρέχεται από μια συλλογή αντικειμένων, συμπεριλαμβανομένων των tap agents καθώς και των network objects. Οι tap agents ενσωματώνουν τα πραγματικά δεδομένα του δικτύου σε προσομοιωμένα πακέτα και αντίστροφα. Τα network objects τοποθετούνται στους tap agents και παρέχουν entrypoint για την αποστολή και παραλαβή των πραγματικών δεδομένων. Όταν χρησιμοποιείται η λειτουργία εξομοίωσης, μια ειδική έκδοση του χρονοπρογραμματιστή χρησιμοποιείται ο RealTime, ο οποίος χρησιμοποιεί την ίδια δομή με τον χρονοπρογραμματιστή βασισμένο στο calendar-queue αλλά δένει την εκτέλεση των γεγονότων σε πραγματικό χρόνο.

Τέλος αναφέρεται η αρχιτεκτονική του NS- 2, το οποίο αποτελείται από 5 μέρη:

- Χρονοπρογραμματισμό διακριτών γεγονότων
- Τμήματα Δικτύου
- Tclcl: σύνδεση μεταξύ C++ και Tcl
- Tcl και
- Tcl: γλώσσα script [63], [64], [65], [66].

## 5.1 NIST MODULE

Το τμήμα κώδικα WiMAX αναπτύχθηκε από το Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας της Αμερικής (NIST) ώστε να ο NS - 2 να υποστηρίζει το πρότυπο. Στο μοντέλο αυτό δεν είναι διαθέσιμα όλα τα χαρακτηριστικά του προτύπου. Ουσιαστικά έχουν υλοποιηθεί:

- WirelessMAN-OFDM φυσικό επίπεδο με δυνατότητα ρύθμισης της διαμόρφωσης
- Αμφιδρόμηση με Διάρεση Χρόνου (TDD)
- Διαχείριση μηνυμάτων για είσοδο στο δίκτυο (χωρίς αυθεντικοποίηση)

- Προεπιλεγμένος χρονοπρογραμματιστής (scheduler) για κατανομή πόρων στους εγγεγραμμένους χρήστες που έχουν κάνει αίτηση για εύρος ζώνης με χρήση του αλγορίθμου Round Robin
- IEEE 802.16e επεκτάσεις για την υποστήριξη σάρωσης και διαπομπών
- Κατακερματισμός και επανασυναρμολόγηση των πλαισίων (frames)

Ενώ μια λίστα με σημαντικά χαρακτηριστικά που δεν είναι διαθέσιμα είναι η εξής:

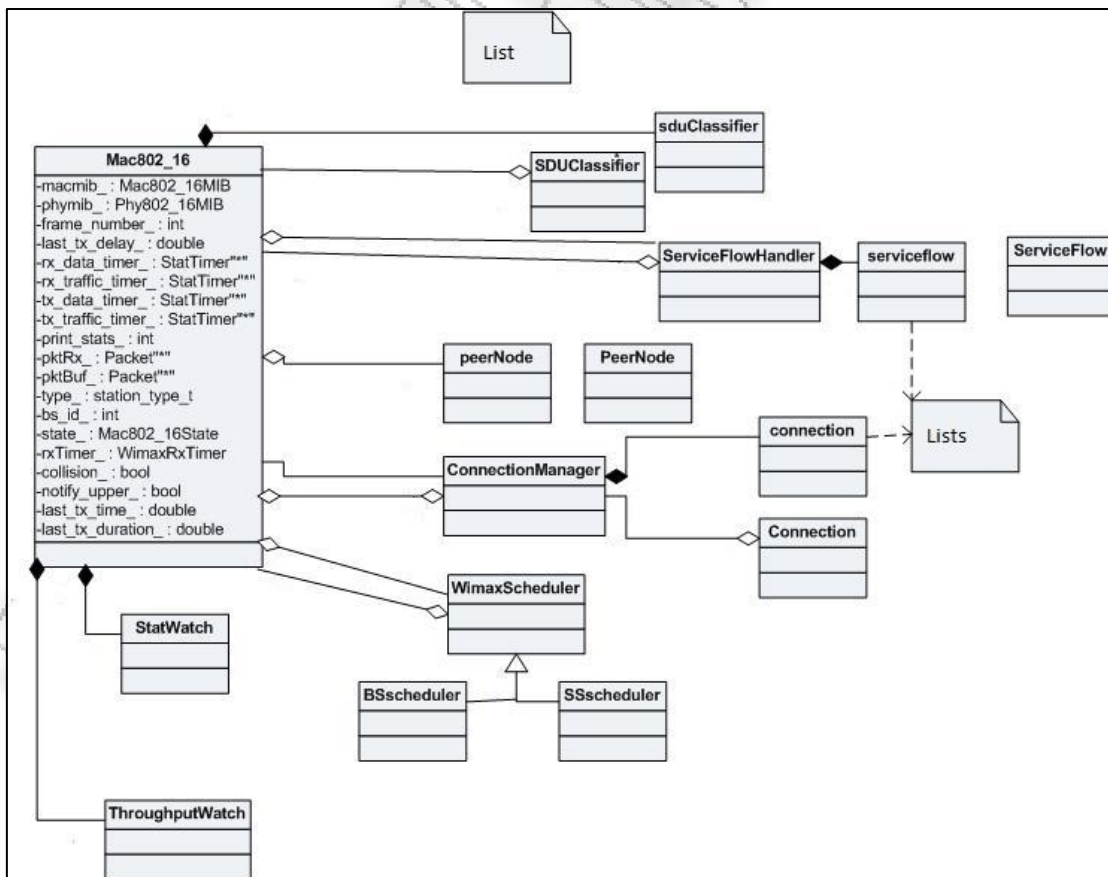
- WirelessMAN-OFDMA
- Αμφιδρόμηση με Διάρθρωση Συχνότητας (FDD)
- ARQ (Αυτόματη Αίτηση Επανάληψης)
- Υψηλή ροή και QoS προγραμματισμός
- Διόρθωση λαθών
- Περιοδικός συγχρονισμός (ranging) και προσαρμογή της εκπεμπόμενης ισχύος
- Συγχώνευση πολλών SDU σε ένα PDU [40]

### 5.1.1 ΥΠΟ-ΕΠΙΠΕΔΟ ΣΥΓΚΛΙΣΗΣ

Το υπο-επίπεδο σύγκλισης βρίσκεται στην κορυφή του επιπέδου MAC όπως προαναφέρθηκε και εκτελεί λειτουργίες όπως η λήψη PDUs από τα υψηλότερα επίπεδα, ταξινόμηση, παράδοση των CS PDUs στο MAC SAP και λήψη των CS PDUs από ομότιμες οντότητες. Στην τρέχουσα υλοποίηση όμως το υπο-επίπεδο σύγκλισης εκτελεί μόνο ταξινόμηση και υποστηρίζει ταξινομητές που καθορίζονται από το χρήστη [42].

### 5.1.2 MAC ΥΠΟ-ΕΠΙΠΕΔΟ

Στο MAC υπο-επίπεδο υποστηρίζεται προς το παρόν μόνο η τοπολογία PMP. Η Mac802\_16 είναι υποτάξη της τάξης Mac. Είναι μια αφηρημένη τάξη που περιέχει τα κοινά στοιχεία του BS και MS. Για παράδειγμα, αποθηκεύει τα MAC MIB και PHY MIB. Είναι η διεπαφή με τα άλλα επίπεδα για αποστολή και λήψη πακέτων. Η Εικόνα 5.2 δείχνει τη τάξη και τις σχέσεις με άλλα τμήματα.



Εικόνα 5.2 – Διάγραμμα τάξης Mac802\_16 [42]

Το MAC έχει μια λίστα των ταξινομητών πακέτου (SDUClassifier) που χαρτογραφεί/συνδέει κάθε εξερχόμενο πακέτο με το αναγνωριστικό της σύνδεσης (CID). Χρησιμοποιώντας TCL, ο χρήστης ρυθμίζει την λίστα των ταξινομητών που θα χρησιμοποιούνται. Η τρέχουσα εφαρμογή χρησιμοποιεί τη IP διεύθυνση του προορισμού, για να κάνει την ταξινόμηση.

Ο ServiceFlowHandler είναι υπεύθυνος για την επεξεργασία των αιτημάτων/ απαντήσεων της εκάστοτε ροής. Επίσης, αποθηκεύει την λίστα των ροών για τον κόμβο.

Ένας SS είναι εγγεγραμμένος σε ένα BS και ο BS μπορεί να συνδεθεί με πολλά SS. Η τάξη PeerNode περιέχει πληροφορίες σχετικά με τους ομότιμους κόμβους, όπως οι συνδέσεις και οι καταστάσεις τους. Τα Connections είναι επίσης προσβάσιμα μέσω του ConnectionManager, ο οποίος περιέχει τον κατάλογο των εισερχόμενων και εξερχόμενων συνδέσεων.

Η αφηρημένη τάξη WimaxScheduler χρησιμοποιείται για τη δημιουργία διεπαφής με το MAC. Υπάρχουν κυρίως δύο τύποι χρονοπρογραμματιστών: ένας για τον BS και ένας για τον SS. Δεδομένου ότι ο χρονοπρογραμματιστής καθορίζεται στο TCL, είναι εύκολη η υλοποίηση και η αλλαγή της αφηρημένης τάξης.

Τέλος, το MAC υπολογίζει στατιστικά στοιχεία μέσω των αντικειμένων StatWatch και ThroughputWatch για πληροφορίες πακέτων και κίνησης. Οι τιμές χρησιμοποιούνται για να ενεργοποιήσουν γεγονότα (trigger events), αλλά μπορούν επίσης να τυπωθούν κατά τη διάρκεια της προσομοίωσης για περαιτέρω επεξεργασία στο τέλος.

Δεδομένου ότι ο BS και ο SS έχουν διαφορετικές μηχανές καταστάσεων ορίστηκαν οι δύο υποκατηγορίες: Mac802\_16BS και Mac802\_16SS [42].

Ο ακόλουθος πίνακας δείχνει τα πακέτα που ορίζονται στο μοντέλο. Όλοι οι ορισμοί των πακέτων βρίσκονται στο mac802\_16pkt.h αρχείο. Για να υπολογιστεί το μέγεθος του πακέτου, έχουν υλοποιηθεί λειτουργίες στο αρχείο mac802\_16pkt.cc

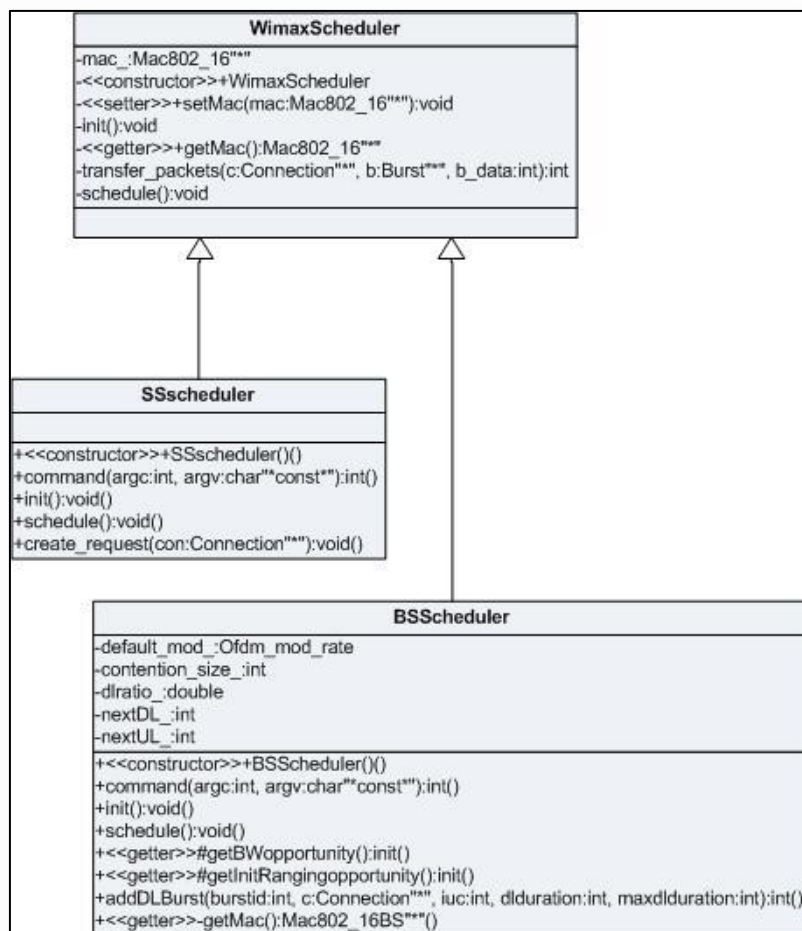
Κατηγορία	Μηνύματα
Συγχρονισμός (synchronization)	DL-MAP / DCD UL-MAP / UCD RNG-REQ/RSP REG-REQ/RSP
Υπηρεσία ροών	DSA-REQ/RSP/ACK
Κινητικότητα	MOB_NBR_ADV MOB_SCN-REQ/RSP MOB_BSHO-REQ/RSP MON_SSHO-REQ MOB_HO-IND MOB_SCN-REP MOB_ASC-REP

Πίνακας 6 – Μηνύματα NIST [42]

#### 5.1.2.1 ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΗΡΕΣΙΩΝ

Η δομή της τάξης επιτρέπει τον καθορισμό των διαφορετικών υπηρεσιών δεδομένων και συγκεκριμένα UGS, rtPS, nrtPS και Best Effort. Οι υπηρεσίες αναφέρονται στην κατηγορία ServiceFlow. Στη συνέχεια παρατίθενται λεπτομερείς πληροφορίες για το QoS. Ο προγραμματισμός των πακέτων γίνεται από ένα Scheduler. Αυτός ο Scheduler αλληλεπιδρά με το MAC μέσω σαφώς καθορισμένων API επιτρέποντας προσαρμοσμένες υλοποιήσεις.

Διαφορετικοί τύποι κόμβων απαιτούν διαφορετικούς χρονοπρογραμματιστές πακέτων. Στο IEEE 802.16, ο BS ελέγχει την κατανομή του εύρους και υπάρχει ένας άπειρος αριθμός υλοποιήσεων. Το μοντέλο περιλαμβάνει μια αφηρημένη τάξη, WimaxScheduler, που δημιουργήθηκε για την εύκολη χρήση διαφορετικών χρονοπρογραμματιστών πακέτων. Όπως φαίνεται στην Εικόνα 5.3, η τάξη αυτή περιέχει ήδη δύο υλοποιήσεις, μια SSScheduler για τους SSs και μία BSscheduler για τους BSs. Οι χρονοπρογραμματιστές μπορούν να αντικατασταθούν με τη χρήση της TCL [42].



Εικόνα 5.3 - Διάγραμμα τάξης χρονοπρογραμματιστή πακέτων [42]

Κατά την υλοποίηση ενός νέου scheduler, οι ακόλουθες μέθοδοι, πρέπει να υλοποιηθούν:

- `Init ()`: αρχικοποίηση του χρονοπρογραμματιστή.
- `process (Packet *)`: Η μέθοδος αυτή χρησιμοποιείται για την επεξεργασία πακέτων που έλαβε από το χρονοπρογραμματιστή (όπως μηνύματα συγχρονισμού).
- `Start_ulsubframe ()`: κώδικας που πρέπει να εκτελείται κατά την έναρξη ενός νέου υποπλαισίου uplink.
- `Start_dlsubframe ()`: κώδικας που πρέπει να εκτελείται κατά την έναρξη ενός νέου υποπλαισίου downlink.

Η λεπτομερής περιγραφή των προεπιλεγμένων χρονοπρογραμματιστών είναι διαθέσιμη στην ενότητα PHY [42].

### 5.1.2.2 ΜΗΧΑΝΙΣΜΟΙ ΚΑΤΑΝΟΜΗΣ ΚΑΙ ΑΙΤΗΣΗΣ ΕΥΡΟΥΣ ΖΩΝΗΣ

Προς το παρόν εφαρμόζεται η Επίλυση μέσω Ανταγωνισμού (Contention Resolution) ως μηχανισμός με τον οποίο ο SS μπορεί να ζητήσει εύρος ζώνης. Ο BS κατανέμει τις σχισμές που υπόκεινται σε συγκρούσεις στην ανερχόμενη κατεύθυνση. Αυτές οι σχισμές χρησιμοποιούνται σε δύο περιπτώσεις:

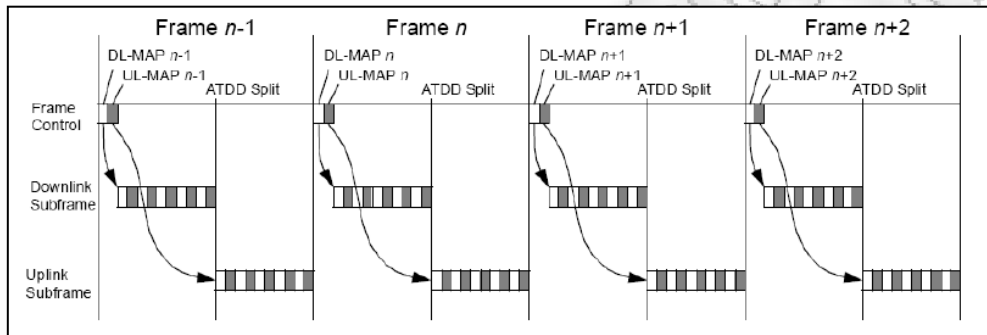
- Αρχική αίτηση συγχρονισμού (ranging)
- Αίτηση εύρους ζώνης

Το μοντέλο υποστηρίζει την στρογγυλοποιημένη δυαδική εκθετική οπισθοχώρηση για την επίλυση μέσω ανταγωνισμού. Τα UL channel descriptor (UCD) μηνύματα που μεταδίδονται από το BS περιέχουν τα μεγέθη των παραθύρων (ως δύναμη του δύο). Ο BS επίσης αποφασίζει για τον αριθμό των διαθέσιμων σχισμών που κατανέμεται σε κάθε πλαίσιο. Ένα υπο-πλαίσιο ανερχόμενης ζεύξης περιέχει ένα `BwContentionslot` και ένα `RngContentionSlot`. Και οι δύο είναι

υποκατηγορίες του ContentionSlot που παρέχει τους βασικούς μηχανισμούς που συνδέονται με τον ανταγωνισμό.

Κατά τη διάρκεια της ένταξης στο δίκτυο, ο SS εκτελεί συγχρονισμό (ranging) για να προσαρμόσει την ισχύ μετάδοσής του. Κατά τη διάρκεια αυτού του βήματος, το SS δημιουργεί ένα RangingRequest. Το SS επιλέγει μια τυχαία οπισθοχώρηση (backoff) μέσα από τα παρεχόμενα παράθυρα του BS και την αποθηκεύει. Στη συνέχεια, ο SS μειώνει το μετρητή κάθε φορά που μια νέα σχισμή ανταγωνισμού βρεθεί μέσα σε πλαίσιο. Όταν ο μετρητής φτάσει στο 0, το πακέτο μεταδίδεται [42].

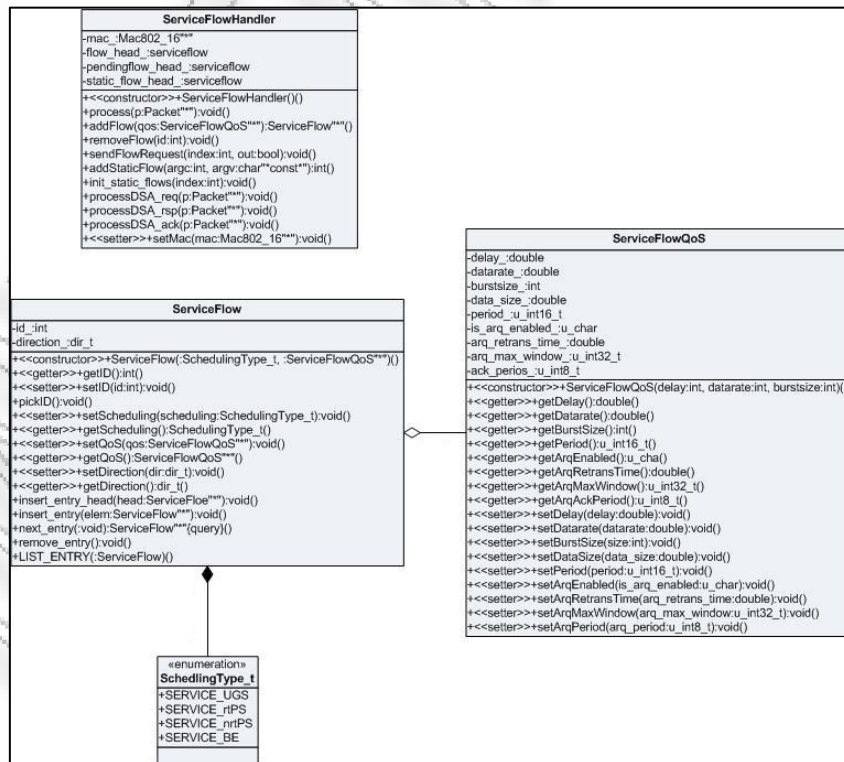
Το μοντέλο σήμερα υποστηρίζει TDD. Σε αυτή τη λειτουργία, η μετάδοση ανερχόμενη ζεύξης λαμβάνει χώρα μετά την κατερχόμενη ζεύξη σε κάθε πλαίσιο. Τα DL\_MAP και UL\_MAP μηνύματα που στέλνονται σε κάθε πλαίσιο καθορίζουν την κατανομή των ριπών και τις ευκαιρίες μετάδοσης για κάθε σταθμό. Οι πληροφορίες που περιέχονται στο UL\_MAP ανήκουν στο ίδιο πλαίσιο όπως φαίνεται στην Εικόνα 5.4.



Εικόνα 5.4 - Συσχέτιση ως προς το χρόνο των DL\_MAP και UL\_MAP [42]

### 5.1.2.3 QoS

Το πλαίσιο (framework) προσδιορίζει τις δομές για την υποστήριξη της υλοποίησης των χρονοπρογραμματιστών που κάνουν χρήση των διαφορετικών τύπων των υπηρεσιών που ορίζονται στο πρότυπο. Κάθε Connection μπορεί να σχετίζεται με ServiceFlow και αντίστοιχες QoS παραμέτρους, Εικόνα 5.5 [42].



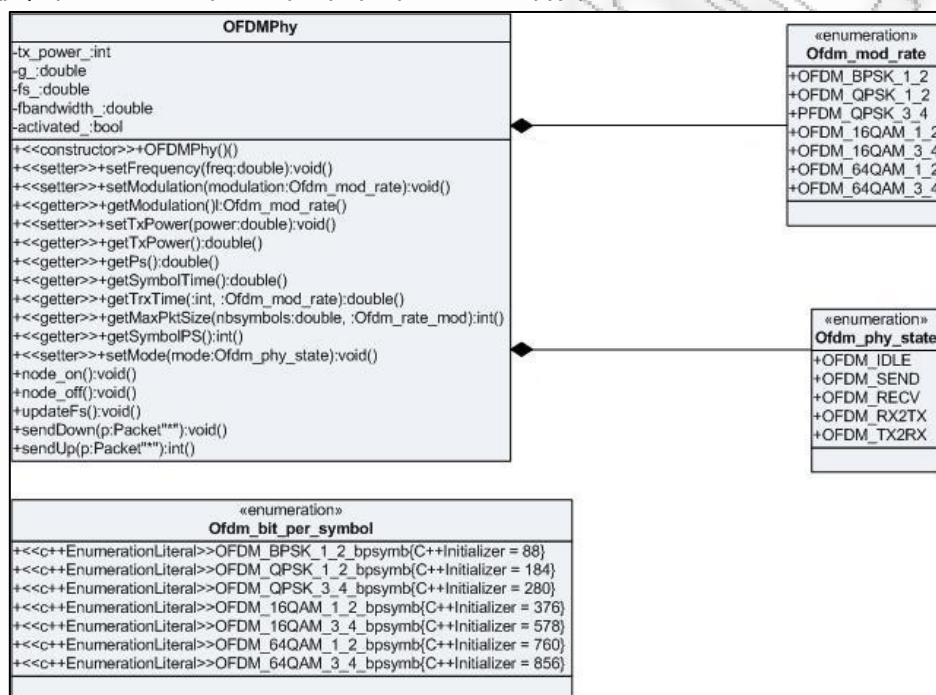
Εικόνα 5.5 - Ροή Υπηρεσίας [42]

QoS ροές δεν είναι επί του παρόντος υλοποιημένες στο NIST. Προεπιλεγμένα, μία σύνδεση δεδομένων σε κάθε κατεύθυνση εγκαθίσταται κατά την ένταξη στο δίκτυο και η BE χρησιμοποιείται για την κατανομή. Το WiMAX-forum επεκτείνει το μοντέλο για να υποστηρίξει αυτή τη δυνατότητα (<http://www.wimaxforum.org>) [42].

### 5.1.3 PHY OFDM

Το OFDM χρησιμοποιείται για τη μετάδοση πακέτων και η διαμόρφωση γίνεται με χρήση TCL για το εύρος ζώνης συχνοτήτων και το κυκλικό πρόθεμα. Δεδομένου ότι κληρονομεί ιδιότητες από την τάξη WirelessPhy όπως η συχνότητα ή η ισχύς της μετάδοσης μπορούν και αυτά επίσης να ρυθμιστούν από την TCL.

Όπως φαίνεται στην Εικόνα 5.6, το φυσικό επίπεδο μπορεί να βρίσκεται σε διαφορετικές καταστάσεις. Σε κατάσταση αποστολής, όλα τα εισερχόμενα πακέτα απορρίπτονται. Σε κατάσταση λήψης, δεν μπορούν να σταλούν τα πακέτα. Επιπλέον, η επικεφαλίδα του πακέτου περιέχει εικονικές πληροφορίες όπως η συχνότητα, η διαμόρφωση και το κυκλικό πρόθεμα, τα οποία χρησιμοποιούνται για το φιλτράρισμα των εισερχόμενων πακέτων.



Εικόνα 5.6 - Διάγραμμα τάξης OFDM φυσικού επιπέδου [42]

Το μοντέλο υποστηρίζει διαφορετικές διαμορφώσεις. Το επίπεδο MAC κατανέμει ριπές που μπορούν να χρησιμοποιούν διαφορετικές διαμορφώσεις ανάλογα με την απόσταση ή τις παρεμβολές. Το γεγονός αυτό επηρεάζει το ρυθμό των δεδομένων και το χρόνο μετάδοσης. Το φυσικό επίπεδο περιλαμβάνει βοηθητικές λειτουργίες που καλούνται από το επίπεδο MAC κατά τη μετάδοση δεδομένων:

- GetTxTime επιστρέφει το χρόνο που απαιτείται για να στείλει ένα πακέτο δεδομένου του μεγέθους και της διαμόρφωσης του.
- GetMaxPktSize είναι η αντίστροφη συνάρτηση και επιστρέφει το μέγιστο μέγεθος πακέτου δεδομένου του αριθμού των OFDM σύμβολων που είναι διαθέσιμα και της διαμόρφωσης.

Οι λειτουργίες node\_on και node\_off ενεργοποιεί ή απενεργοποιεί το κλειδί όλων των μεταδόσεων και λήψεων πακέτων, αλλά δεν συνδέεται επί του παρόντος με κανένα μηχανισμό κατανάλωσης ενέργειας [42].

#### 5.1.3.1 OFDM ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΤΕΣ ΠΑΚΕΤΩΝ

Η τρέχουσα υλοποίηση του χρονοπρογραμματιστή πακέτων για τον BS μπορεί να ρυθμιστεί χρησιμοποιώντας την ακόλουθη εντολή: set scheduler [new WimaxScheduler/BS], η οποία

δημιουργεί ένα χρονοπρογραμματιστή πακέτων για τον BS. Ενώ η εντολή `$scheduler set-contention-size $size` ορίζει τον αριθμό των χρονοσχισμών ανταγωνισμού που θα διατεθούν για τον αρχικό συγχρονισμό (ranging) και τα αιτήματα για εύρος ζώνης σε κάθε πλαίσιο.

Ο χρονοπρογραμματιστής εφαρμόζει ένα BE χρονοπρογραμματιστή σε συνδυασμό με έναν αλγόριθμο Εξυπηρέτησης εκ περιτροπής για να διανέμουν τις κατανομές εύρους ζώνης μεταξύ των χρηστών. Για την υποστήριξη BE, οι αιτήσεις εύρους ζώνης που παράγονται στον SS αναφέρουν το μέγεθος των δεδομένων προς μετάδοση.

Η σχέση μεταξύ των υπο-πλαισίων κατερχόμενης και ανερχόμενης ζεύξης είναι σταθερή και ρυθμίζεται μέσω TCL με την εντολή `WimaxScheduler/BS set dlratio_ 0.3`, στο παράδειγμα αυτό το 30% του πλαισίου είναι για κατερχόμενη κίνηση και το 70% είναι για ανερχόμενη.

Ο χρονοπρογραμματιστής επιτρέπει επίσης στους χρήστες να έχουν διαφορετικές διαμορφώσεις με την εντολή `$scheduler set-default-modulation $modulation`, η οποία θέτει τη διαμόρφωση που θα χρησιμοποιήσει για τις σχισμές του αρχικού συγχρονισμού (ranging) και τα αιτήματα εύρους ζώνης. Τα προφίλ ριπών δημιουργούνται από προεπιλογή βρίσκονται στο Πίνακας 7.

Προφίλ	Διαμόρφωση
DIUC_PROFILE_1, UIUC_PROFILE_1	OFDM_BPSK_1_2
DIUC_PROFILE_2, UIUC_PROFILE_2	OFDM_QPSK_1_2
DIUC_PROFILE_3, UIUC_PROFILE_3	OFDM_QPSK_3_4
DIUC_PROFILE_4, UIUC_PROFILE_4	OFDM_16QAM_1_2
DIUC_PROFILE_5, UIUC_PROFILE_5	OFDM_16QAM_3_4
DIUC_PROFILE_6, UIUC_PROFILE_6	OFDM_64QAM_2_3
DIUC_PROFILE_7, UIUC_PROFILE_7	OFDM_64QAM_3_4

Πίνακας 7 – Προφίλ ριπών (diuc) [42]

Ο χρήστης μπορεί να επιλέξει το προφίλ ριπής [1-7] με την TCL χρησιμοποιώντας το ακόλουθο κείμενο: `[$SSWithWiMax set mac_(0)] set-diuc 7`. Προεπιλεγμένα το προφίλ είναι ίδιο τόσο για την κατερχόμενη όσο και για την ανερχόμενη επικοινωνία μεταξύ SS και BS [42].

#### 5.1.3.2 ΡΥΘΜΙΣΗ ΠΑΡΑΜΕΤΡΩΝ ΕΝΟΣ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΤΗ

Για να υπάρχει ευελιξία, το επίπεδο MAC μπορεί να χρησιμοποιήσει διαφορετικούς τύπους χρονοπρογραμματιστών. Κυρίως υπάρχει ένας για τους BS και ένας για τους SS.

Για τον BS, ο ακόλουθος κώδικας TCL καθορίζει το χρονοπρογραμματιστή:

```
# Create scheduler
set scheduler [new WimaxScheduler/BS]
# Add scheduler
[$nodeWithWiMax set mac_(0)] set-scheduler $scheduler
```

Αυτός ο χρονοπρογραμματιστής δημιουργείται αυτόματα όταν ο MAC 802.16 BS δημιουργείται.

Για τον SS, πρέπει να χρησιμοποιηθούν τα ακόλουθα:

```
# Create scheduler
set scheduler [new WimaxScheduler/SS]
# Add scheduler
[$nodeWithWiMax set mac_(0)] set-scheduler $scheduler
```

Αυτός ο χρονοπρογραμματιστής δημιουργείται αυτόματα όταν το MAC 802.16 SS δημιουργείται [42].

#### 5.1.3.3 ΣΤΑΤΙΣΤΙΚΑ ΣΤΟΙΧΕΙΑ

Μερικά στατιστικά στοιχεία που συλλέγονται στο επίπεδο MAC. Η ακόλουθη εντολή χρησιμοποιείται για την προβολή των τιμών κατά τη διάρκεια της προσομοίωσης.



Mac/802\_16 set print\_stats\_ true

#### 5.1.4 ΑΡΧΕΙΟ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ (TRACING)

Το πρότυπο IEEE 802.16 εισάγει νέες τιμές στο αρχείο παρακολούθησης. Εμφανίζονται και τρεις νέοι λόγοι για την απόρριψη ενός πακέτου:

- CID: αυτόν τον κωδικό χρησιμοποιεί όταν ένα πακέτο που έλαβε το επίπεδο MAC δεν ταιριάζει με οποιαδήποτε σύνδεση.
- QWI: κάθε σύνδεση έχει μια ουρά για να αποθηκεύει εν αναμονή πακέτα. Όταν η ουρά είναι πλήρης, το πακέτο απορρίπτεται χρησιμοποιώντας αυτόν τον κωδικό.
- FRG: δείχνει ένα σφάλμα κατά τη μετάδοση ενός τμήματος.

Ένας νέος τύπος πακέτου εισάγεται. Μερικές φορές, οι BS πρέπει να επικοινωνούν για λόγους συγχρονισμού. Ένας νέος Agent που ονομάζεται Agent/ WimaxCtrl χειρίζεται αυτή την επικοινωνία και στέλνει πακέτα με το όνομα WimaxCtrl.

Αν είναι ενεργοποιημένη η MAC παρακολούθηση και χρησιμοποιείται κατακερματισμός, τα τμήματα θα εμφανίζονται ως αποσταλμένα, αλλά χωρίς να έχουν παραληφθεί. Στο τελευταίο τμήμα, το πλήρες πακέτο μπορεί να αποκωδικοποιηθεί και να μεταδοθεί σε ανώτερο επίπεδο το οποίο θα δημιουργήσει εγγραφή στο αρχείο παρακολούθησης από την πλευρά του δέκτη. Για παράδειγμα, ως θεωρηθεί ένα πακέτο με 1520 bytes το οποίο θα είναι κατακερματισμένο σε τέσσερα τμήματα 396, 396, 396, και 364 bytes. Το αρχείο παρακολούθησης θα περιλαμβάνει τέσσερις καταχωρήσεις "αποστολή" για κάθε ένα από τα κομμάτια, αλλά μόνο μία καταχώρηση "έλαβε" 1520 bytes για το πλήρες πακέτο.

## 5.2 WIMAX MODULE ΒΑΣΙΣΜΕΝΟ ΣΤΟ NIST ΜΕ QoS

Η συμβολή της εφαρμογής αποτελείται από την προσθήκη στο NIST, των πέντε QoS τύπων υπηρεσιών, των απαιτήσεών τους, τους μηχανισμούς που καθορίζονται από το πρότυπο και μερικούς αλγόριθμους χρονοπρογραμματισμού. Οι χρονοπρογραμματιστές αυτοί λαμβάνουν υπόψη τους τις QoS απαιτήσεις της εκάστοτε σύνδεσης. Για το σχεδιασμό της εφαρμογής χρησιμοποιήθηκε η Unified Modeling Language (UML) και υλοποιήθηκε για το NS2 – 2.29. Αναλυτικότερα στις προσθήκες συμπεριλαμβάνονται μερικές παράμετροι QoS, η προσαρμογή των ζεύξεων και οι αλγόριθμοι χρονοπρογραμματισμού για τρεις τύπους υπηρεσιών, τις UGS, rtPS και Best Effort. Υλοποιήθηκαν επίσης μηχανισμοί ευκαιριών αίτησης μονοεκπομπής και ανταγωνισμού [67].

Στον κώδικα του NIST η κεντρική τάξη είναι η Mac802.16, η οποία έχει σχέση με άλλες τάξεις όπως η ServiceFlowHandler, η peerNode και η WimaxScheduler. Η ServiceFlowHandler είναι υπεύθυνη για την διαχείριση των συνδέσεων, καθεμιά εκ των οποίων έχει ένα συγκεκριμένο τύπο υπηρεσίας που περιέχει QoS παραμέτρους. Η τάξη peerNode περιέχει πληροφορίες σχετικές με τον κάθε SS και BS. Η WimaxScheduler είναι υπεύθυνη για τον συγχρονισμό και την εγγραφή των χρηστών, ενώ παράλληλα εκτελεί τους αλγόριθμους χρονοπρογραμματισμού και περιλαμβάνει δύο χρονοπρογραμματιστές έναν για τους SS και έναν για τους BS. Στην παρούσα εφαρμογή παρέχεται η δυνατότητα χρήσης διαφορετικών σχημάτων διαμόρφωσης και κωδικοποίησης, τα οποία εξαρτώνται από το SNR. Αυτό επιτεύχθηκε με αλλαγές στο peerNode και στο BSScheduler. Στη συνέχεια στην τάξη ServiceFlowHandler προστέθηκαν οι παράμετροι CID, traffic priority, maximum sustained traffic rate, minimum reserved traffic rate, tolerated jitter και maximum latency. Αφού αρχικοποιηθούν οι παράμετροι μπορεί να προστεθεί μια νέα ροή υπηρεσίας, μέσω πακέτων DSA, τα οποία τροποποιήθηκαν για να συμπεριλαμβάνουν στις πληροφορίες τους τις παραμέτρους. Υλοποιήθηκαν χρονοπρογραμματιστές, λαμβάνονται υπόψη τους εξής τρεις τύπους QoS: UGS, rtPS και BE [67].

### Χρονοπρογραμματιστής UGS

Ο BS καθορίζει όλους τους SS που έχουν UGS συνδέσεις. Η σύνδεση που θα εξυπηρετηθεί πρώτη είναι τυχαία. Επειδή οι SS εξυπηρετούνται περιοδικά, ορίζεται μια περίοδος μέσω της TCL για κάθε SS, διαφορετικά λαμβάνεται μια προκαθορισμένη τιμή. Από την τιμή αυτή ο BS καθορίζει εάν ο SS θα εξυπηρετηθεί στο τρέχον πλαίσιο ή όχι. Για κάθε σύνδεση UGS που θα εξυπηρετηθεί ο BS καθορίζει τον χρόνο μετάδοσης σύμφωνα με το Maximum Sustained Traffic Rate και έπειτα καθορίζει τον αριθμό των συμβόλων που θα κρατηθούν για τη σύνδεση σύμφωνα με το καθορισμένο χρόνο μετάδοσης και το σχήμα διαμόρφωσης και κωδικοποίησης που χρησιμοποιείται. Τα διαθέσιμα σύμβολα πρέπει να είναι αρκετά για να εξυπηρετηθούν όλες

οι UGS συνδέσεις. Μόλις οριστεί ο αριθμός των συμβόλων ο BS ενημερώνει το UL-MAP. Ο αλγόριθμος εκτελείται στην αρχή κάθε πλαισίου [67].

#### **Χρονοπρογραμματιστής rtPS**

Ο BS καθορίζει όλους τους SS που έχουν rtPS συνδέσεις. Υπάρχουν τέσσερις χρονοπρογραμματιστές υλοποιημένοι για αυτόν τον τύπο των συνδέσεων. Ο RR μοιράζει δίκαια τους πόρους σε όλους τους SS. Ο mSIR δίνει πόρους στους SS με το μέγιστο SNR. Ο WRR είναι επέκταση του RR και βασίζεται σε στατιστικά βάρη, ενώ ο τέταρτος αλγόριθμος ο TRS δεν δίνει προσωρινά πόρους σε SS με SNR χαμηλότερο από ένα προκαθορισμένο κατώφλι. Ο αλγόριθμος επιλέγεται από το χρήστη και γίνεται ως παράμετρος κατά την εκτέλεση του TCL κώδικα. Εάν ο SS δεν έχει αίτηση για πόρους τότε ο BS συνεχίζει στον επόμενο, διαφορετικά καθορίζει τον χρόνο μετάδοσης και τον αριθμό των συμβόλων που θα κρατηθούν για τη σύνδεση. Εάν δεν υπάρχουν αρκετά σύμβολα τότε ο BS κατανέμει τους πόρους του έχουν απομείνει και ενημερώνει την σύνδεση για τον αριθμό που θα κατανεμηθούν στο επόμενο πλαίσιο [67].

#### **Χρονοπρογραμματιστής BE**

Όπως και πριν καθορίζονται οι συνδέσεις και χρησιμοποιείται RR αλγόριθμος. Εάν ο SS δεν έχει αίτηση για πόρους τότε ο BS συνεχίζει στον επόμενο, διαφορετικά καθορίζει τον χρόνο μετάδοσης και τον αριθμό των συμβόλων που θα κρατηθούν για τη σύνδεση. Εάν δεν υπάρχουν αρκετά σύμβολα τότε ο BS κατανέμει τα σύμβολα του έχουν απομείνει και ενημερώνει την σύνδεση για τον αριθμό που θα της κατανεμηθούν στο επόμενο πλαίσιο [67].

## 6 ΑΞΙΟΛΟΓΗΣΗ ΕΠΙΔΟΣΗΣ

### 6.1 ΜΟΝΤΕΛΟ ΠΡΟΣΟΜΙΩΣΗΣ

Αρχικά καθορίζονται καθολικές μεταβλητές και προσδιορίζονται οι τιμές των διάφορων παραμέτρων του προτύπου ανάλογα με το πρότυπο ως εξής:

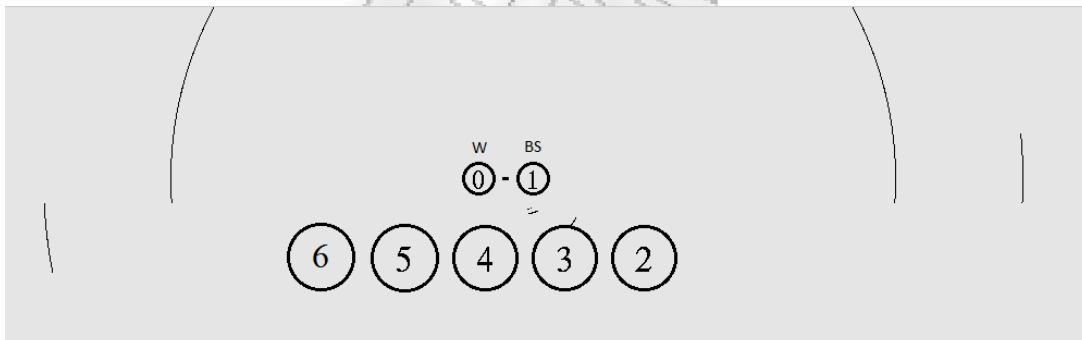
Κανάλι	3,486e+9
Συχνότητα εύρους ζώνης	5 MHz
Κυκλικό πρόθεμα	0,25
Διάρκεια πλαισίων	20ms
Μέγεθος πακέτων	500 bytes
Διάρκεια Προσομοίωσης	200sec
Propagation Model	Two Ray Ground
Antenna Model	Omni Antenna
Transmit Power	0,025
Receive Power Threshold	2,025e-12
Carrier Sense Power Threshold	0,9 * Receive Power Threshold

Πίνακας 8 – Παράμετροι προσομοίωσης

Επίσης δίνονται οι διαστάσεις του πλέγματος.

```
set opt(x) 250 ;# X dimension of the topography
set opt(y) 250 ;# Y dimension of the topography
```

Στη συνέχεια ορίζεται ένα πλέγμα μέσα στο οποίο δημιουργούνται δύο ενσύρματοι κόμβοι, οι οποίοι συνδέονται με πέντε ασύρματους κόμβους μέσω ενός BS κόμβου. Ο κόμβος 0 λειτουργεί ως sink κόμβος για όλες τις ροές. Η τοπολογία του δικτύου είναι η ακόλουθη:



Εικόνα 6.1 – Τοπολογία προσομοίωσης

Αφού υπάρχουν ενσύρματοι και ασύρματοι κόμβοι στην προσομοίωση πρέπει να χρησιμοποιηθεί ιεραρχική δρομολόγηση για να δρομολογηθούν τα πακέτα ανάμεσα σε δύο τομείς (domain). Η δρομολόγηση των ενσύρματων κόμβων βασίζεται στις ζεύξεις τους, ενώ στους ασύρματους χρησιμοποιείται ένα adhoc πρωτόκολλο δρομολόγησης. Ανάμεσα σε αυτούς τους δύο τύπους κόμβων χρησιμοποιούνται οι BS ως ένα είδος gateway. Αρχικά ορίζεται το address type ως ιεραρχικό και στη συνέχεια καθορίζεται η τοπολογία, θέτοντας τον αριθμό των τομέων ίσον με 2. Κάθε τομέας χωρίζεται σε υπο-τομείς (sub-domain ή cluster), με την εντολή «lappend cluster\_num 1 1» ουσιαστικά ο πρώτος τομέας αλλά και ο δεύτερος αποτελούνται από ένα μόνο υποτομέα ο καθένας. Σε κάθε υπο-τομέα τοποθετείται ένας συγκεκριμένος αριθμός κόμβων με την εντολή «lappend eilastlevel 1 6» που σημαίνει ότι στο πρώτο υπο-τομέα θα υπάρχει ένας κόμβος και στο δεύτερο υπο-τομέα, το ασύρματο, θα υπάρχουν έξι κόμβοι. Είναι σημαντικό ο κόμβος BS να βρίσκεται στον τομέα των ασύρματων κόμβων έτσι ώστε όλα τα πακέτα που προέρχονται από το ενσύρματο τομέα και τα οποία προορίζονται για κάποιο ασύρματο κόμβο να φθάσουν στον BS που χρησιμοποιεί adhoc πρωτόκολλο δρομολόγησης για να δρομολογήσει κάθε πακέτο στον προορισμό του.

Δημιουργούνται οι κόμβοι με τις κατάλληλες ρυθμίσεις για τον κάθε τύπο, δηλαδή οι ενσύρματοι κόμβοι δημιουργούνται όπως και σε κάθε άλλη προσομοίωση του NS- 2, ο BS έχει την επιλογή wiredRouting ON, ενώ οι ασύρματοι κόμβοι OFF. Όλες οι άλλες επιλογές παραμένουν ίδιες τόσο για τον BS όσο και για τους ασύρματους κινητούς κόμβους.

Τέλος, ορίζεται η κίνηση που θα παράγει κάθε κόμβος. Αναλυτικά για την πρώτη πολιτική, ο πρώτος κόμβος έχει κίνηση CBR με μέγεθος πακέτων 200 bytes και "interval" 0.0025, ο δεύτερος κόμβος έχει κίνηση CBR με μέγεθος πακέτων 200 bytes και "interval" 0.0033, ο τρίτος κόμβος έχει κίνηση CBR με μέγεθος πακέτων 200 bytes και "interval" 0.0025, ο τέταρτος κόμβος έχει κίνηση CBR με μέγεθος πακέτων 200 bytes και "interval" 0.0025 και ο πέμπτος κόμβος έχει κίνηση CBR με μέγεθος πακέτων 200 bytes και "interval" 0.0033. Δηλαδή η κίνηση που παράγει κάθε κόμβος είναι 80 KByte/ sec, 60,6 KByte/ sec, 80 KByte/ sec, 80 KByte/ sec και 60,6 KByte/ sec αντίστοιχα. Όλοι οι κόμβοι έχουν ίδια προτεραιότητα. Έτσι στο πρώτο σύστημα ο πρώτος χρήστης απαιτεί ελάχιστη κίνηση 40 KByte/ sec, ο δεύτερος 20 KByte/ sec, ο τρίτος 20 KByte/ sec, ο τέταρτος 80 KByte/ sec και ο πέμπτος 40 KByte/ sec. Στον ακόλουθο πίνακα συνοψίζονται οι παραπάνω πληροφορίες σχετικά με την παραγόμενη και την απαιτούμενη κίνηση του κάθε χρήστη.

Κίνηση Ροών Χρήστες	Κίνηση Εφαρμογής (KByte/ sec)	Ελάχιστη απαίτηση Χρήστη (KByte/ sec)
Χρήστης 1	80 (200 byte/ 0.0025sec)	40
Χρήστης 2	60,6 (200 byte/ 0.0033sec)	20
Χρήστης 3	80 (200 byte/ 0.0025sec)	20
Χρήστης 4	80 (200 byte/ 0.0025sec)	80
Χρήστης 5	60,6 (200 byte/ 0.0033sec)	40

Πίνακας 9 – Πολιτική 1 παράμετροι κόμβων

Για τη δεύτερη πολιτική δημιουργήθηκαν οι ίδιοι κόμβοι με τον ίδιο ρυθμό μετάδοσης και τις ίδιες ελάχιστες απαιτήσεις. Όλοι οι κόμβοι έχουν και πάλι την ίδια προτεραιότητα. Ενώ υπάρχουν και οι αποδεκτές τιμές που είναι για το πρώτο χρήστη 60 KByte/ sec, το δεύτερο 35 KByte/ sec, το τρίτο 40 KByte/ sec, το τέταρτο 80 KByte/ sec και το πέμπτο 50 KByte/ sec. Στον ακόλουθο πίνακα συνοψίζονται οι παραπάνω πληροφορίες.

Κίνηση Ροών Χρήστες	Κίνηση Εφαρμογής (KByte/ sec)	Αποδεκτή τιμή (KByte/ sec)	Ελάχιστη απαίτηση Χρήστη (KByte/ sec)
Χρήστης 1	80	60	40
Χρήστης 2	60,6	35	20
Χρήστης 3	80	40	20
Χρήστης 4	80	80	80
Χρήστης 5	60,6	50	40

Πίνακας 10 - Πολιτική 2 παράμετροι κόμβων

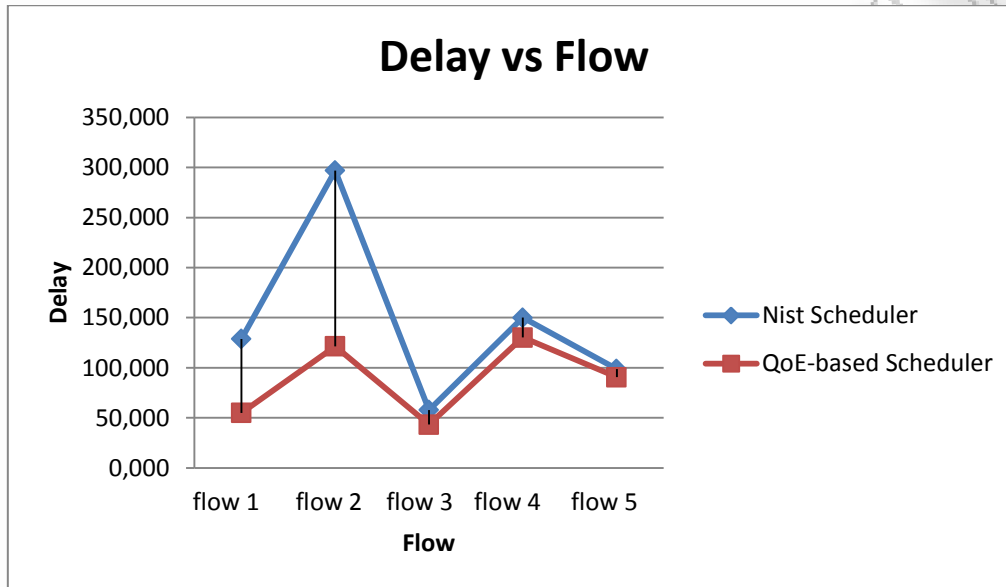
Χρησιμοποιήθηκαν πέντε διαφορετικά κατώφλια 10%, 20%, 30, 40% και 50%.

## 6.2 ΑΠΟΤΕΛΕΣΜΑΤΑ

Σύμφωνα με τα αποτελέσματα των προσομοιώσεων ο ρυθμός μετάδοσης δεδομένων κάθε ροής μειώνεται όπως ήταν αναμενόμενο, η καθυστέρηση επίσης μειώνεται, ενώ είναι ικανοποιητική και η μείωση των ποσοστών απώλειας των πακέτων. Οι Εικόνα 6.2, Εικόνα 6.3 και Εικόνα 6.4 δείχνουν τη σύγκριση μεταξύ του NIST χρονοπρογραμματιστή και του χρονοπρογραμματιστή που βασίζεται σε QoE στη πρώτη πολιτική όσον αφορά το ρυθμό μετάδοσης δεδομένων, τη μέση καθυστέρηση μεταφοράς των πακέτων και το ποσοστό απώλειας πακέτων αντίστοιχα. Να σημειωθεί ότι η απόδοση μειώνεται για κάθε ροή, εκτός από την 4, η οποία αυξάνεται, διότι η κίνηση που παράγει ο κόμβος και οι ελάχιστες απαιτήσεις του χρήστη ταυτίζονται ενώ η μειωμένη χρήση του δικτύου από τις υπόλοιπες συνδέσεις επιτρέπουν την επιτυχή αποστολή περισσότερων πακέτων από την 4.

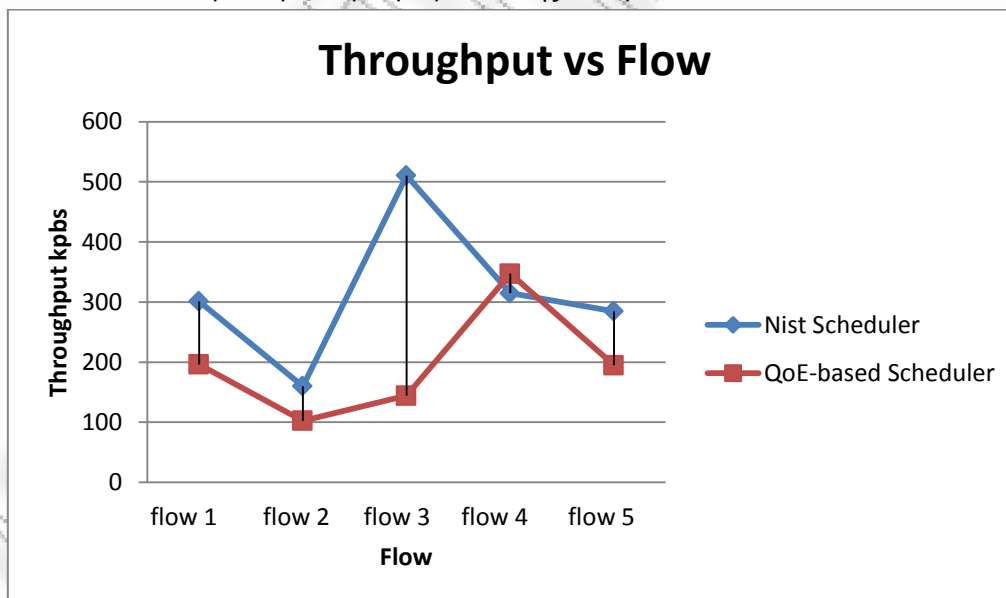
### Πολιτική 1

Όπως φαίνεται στην επόμενη εικόνα η καθυστέρηση μεταφοράς των πακέτων μειώνεται και για τις τρεις πρώτες ροές ενώ στην ροή 5 η μείωση είναι ελάχιστη.



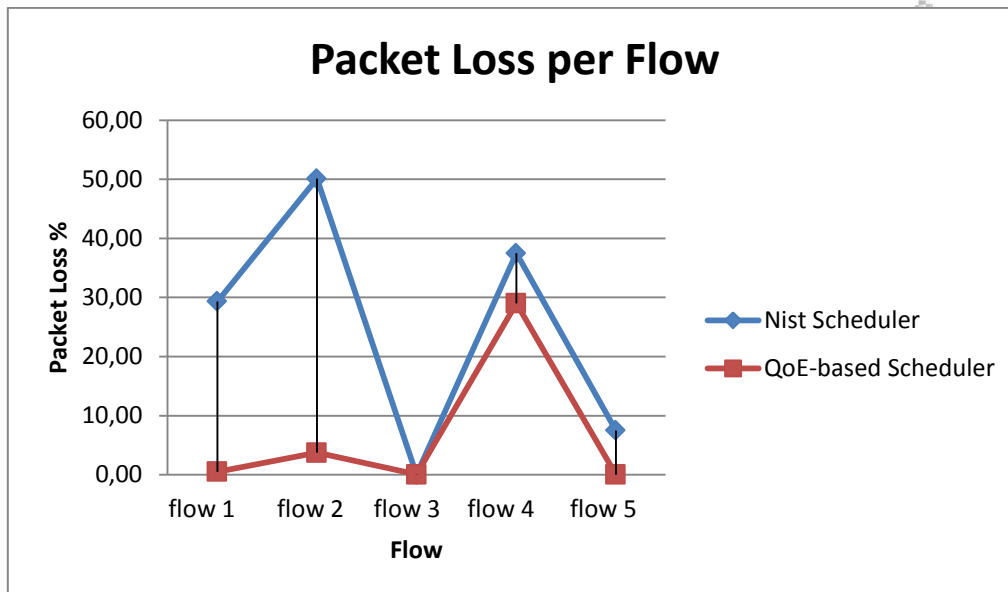
Εικόνα 6.2 – Καθυστέρηση μετάδοσης δεδομένων, πολιτική 1

Όσον αφορά το ρυθμό μετάδοσης δεδομένων στη ροή 1 ενώ η ελάχιστη απαίτηση του χρήστη είναι η μιση από την κίνηση που παράγει, ο ρυθμός του δεν μειώνεται πολύ παραμένοντας στα ίδια επίπεδα στα οποία κινούνται οι υπόλοιπες ροές με χρήση του χρονοπρογραμματιστή που λαμβάνει υπόψη του το QoE. Ενώ οι υπόλοιπες ροές μεταβάλλονται έτσι ώστε να πλησιάσουν τις ελάχιστες απαιτήσεις του εκάστοτε χρήστη ώστε να μειωθούν οι καθυστερήσεις και τα ποσοστά απώλειας πακέτων. Στην ακόλουθη εικόνα φαίνονται τα αποτελέσματα για το ρυθμό μετάδοσης δεδομένων.



Εικόνα 6.3 Ρυθμός μετάδοσης δεδομένων, πολιτική 1

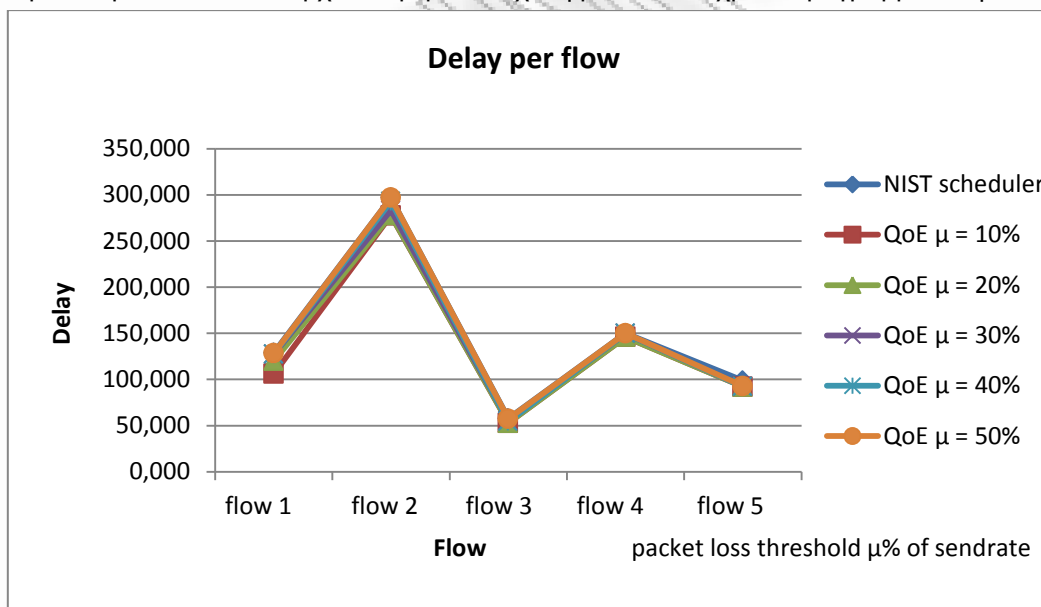
Στην Εικόνα 6.4 παρουσιάζεται η βελτίωση που φέρει ο χρονοπρογραμματιστής που βασίζεται σε QoE ως προς το ποσοστό απώλειας πακέτων για όλες τις ροές εκτός της 3, της οποίας το ποσοστό ήταν ήδη πολύ κοντά στο μηδέν.



Εικόνα 6.4 – Ποσοστό απώλειας πακέτων, πολιτική 1

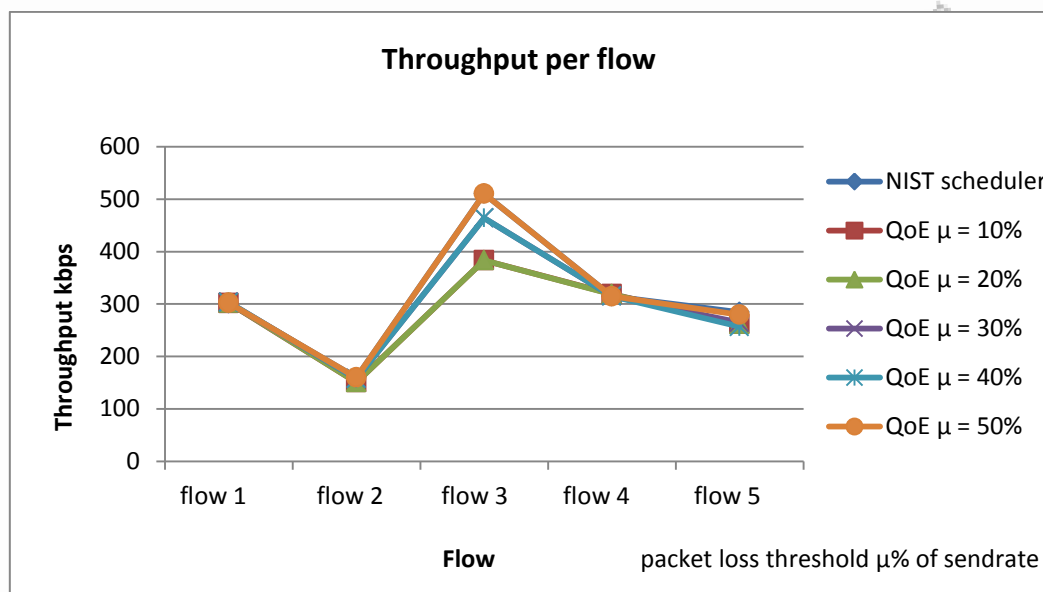
**Πολιτική 2**

Όπως φαίνεται στην Εικόνα 6.5 η καθυστέρηση μεταφοράς των πακέτων μειώνεται κυρίως για κατώφλι ίσο με 10%. Για τα υπόλοιπα κατώφλια οι διαφορές μειώνονται μέχρι που για κατώφλι ίσο με 50% δεν υπάρχει διαφορά σε σχέση με το NIST χρονοπρογραμματιστή.



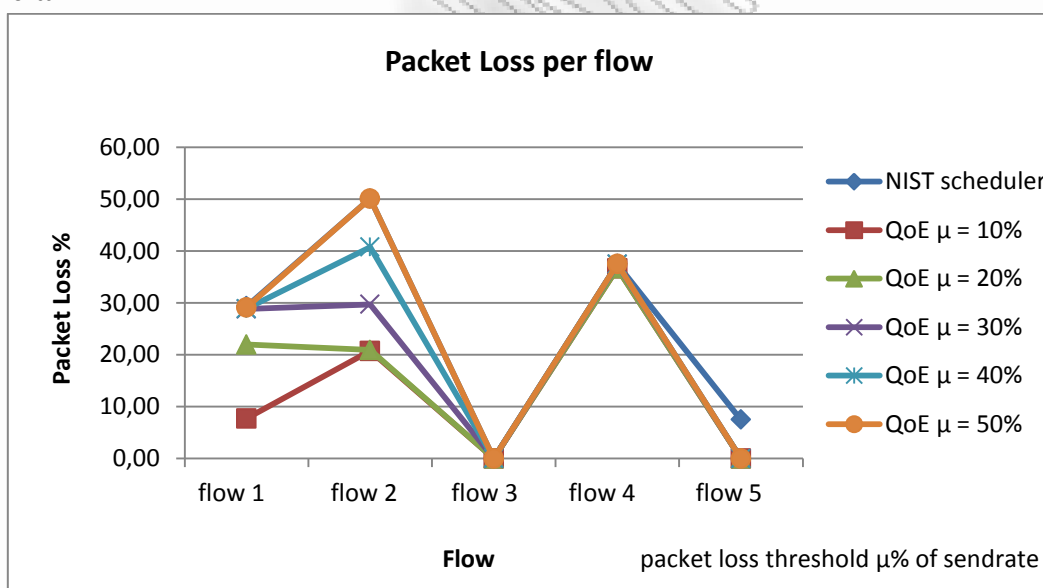
Εικόνα 6.5 – Καθυστέρηση μετάδοσης δεδομένων, πολιτική 2

Σχετικά με το ρυθμό μετάδοσης δεδομένων παρατηρείται ελάχιστη μείωση εκτός από αυτή της ροής 3; η οποία είχε το μεγαλύτερο εύρος μεταξύ μέγιστης και ελάχιστης τιμής. Δηλαδή στις ροές 1, 2 και 4 ο ρυθμός δεν μειώνεται παραμένοντας στα ίδια επίπεδα στα οποία κινούνται με χρήση NIST χρονοπρογραμματιστή αλλά και του χρονοπρογραμματιστή που λαμβάνει υπόψη του το QoE με οποιοδήποτε κατώφλι. Στην ροή 5 φαίνεται ελάχιστη μείωση. Το αποτέλεσμα παρουσιάζονται στην Εικόνα 6.6.



Εικόνα 6.6 – Ρυθμός μετάδοσης δεδομένων, πολιτική 2

Τέλος, η βελτίωση είναι προφανής όταν χρησιμοποιείται ο χρονοπρογραμματιστής που βασίζεται σε QoE με κατώφλι 10% και 20% ως προς το ποσοστό απώλειας πακέτων για όλες τις ροές και πάλι εκτός των ροών 3 και 4. Δηλαδή με την αύξηση του ρυθμού αποστολής δεδομένων, σε σχέση με το πολιτική 1, ήταν και πάλι βελτιωμένο το ποσοστό απώλειας πακέτων.



Εικόνα 6.7 – Ποσοστό απώλειας πακέτων, πολιτική 2

## 7 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

Η τεχνολογία WiMAX σχεδιάστηκε, βάσει του προτύπου 802.16, για να λειτουργεί στην ευρεία ζώνη συχνοτήτων από 2 έως 66 GHz, έτσι ώστε να μπορεί να χρησιμοποιηθεί τόσο σε δίκτυα οπτικής επαφής όσο και σε δίκτυα μη οπτικής επαφής. Τα WiMAX δίκτυα παρέχουν διαλειτουργικότητα, μικρότερο κόστος εγκατάστασης νέων υποδομών και καλύτερη φασματική απόδοση σε σύγκριση με τις υπάρχουσες τεχνολογίες. Τέλος με υποστήριξη του IP στα WiMAX δίκτυα οι πολυμεσικές εφαρμογές θα είναι κατά πολύ φθηνότερες.

Ωστόσο το πρότυπο 802.16 δεν έχει ορίσει κάποιο συγκεκριμένο αλγόριθμο χρονοπρογραμματισμού, αφήνοντας έτσι ελεύθερο κάθε φορέα που θα χρησιμοποιήσει το πρότυπο για την παροχή υπηρεσιών ευρυζωνικής ασύρματης πρόσβασης στο διαδίκτυο να επιλέξει αλγόριθμο. Γι' αυτό το λόγο έχουν γίνει πολλές μελέτες και έχουν προταθεί από πολλές ομάδες διαφορετικοί αλγόριθμοι χρονοπρογραμματισμού. Οι αλγόριθμοι αυτοί λαμβάνουν υπόψη τους το QoS, τη δικαιοσύνη και άλλες παραμέτρους. Ωστόσο δεν υπάρχουν ακόμα εργασίες που ασχολούνται με αλγόριθμους χρονοπρογραμματισμού που παρέχουν QoE. Η παρούσα εργασία έγκειται στην εφαρμογή QoE σε αλγορίθμους χρονοπρογραμματισμού σε WiMAX δίκτυα. Ειδικότερα επικεντρώνεται στον rtPS τύπο υπηρεσίας του WiMAX προτύπου και προτείνονται δύο διαφορετικές QoE προσεγγίσεις. Η μια πολιτική ανάλογα με το αν υπάρχει απώλεια των πακέτων στο σύστημα μειώνει το ρυθμό μετάδοσης των συνδέσεων έως ότου φτάσουν στο ελάχιστο επιτρεπτό, δηλαδή την υποκειμενική απαίτηση του χρήστη. Στην δεύτερη πολιτική μπορεί να μειωθεί και πάλι ο ρυθμός μετάδοσης αν η απώλεια της σύνδεσης είναι μεγαλύτερη από ένα κατώφλι ενώ στην αντίθετη περίπτωση μπορεί να αυξηθεί.

Οι προσομοιώσεις που πραγματοποιήθηκαν δείχνουν ότι για τον rtPS τύπο υπηρεσίας είναι απαραίτητη η εφαρμογή QoE πολιτικών. Από τα αποτελέσματα των προσομοιώσεων φαίνεται ότι η χρήση διαφορετικών επιπέδων MOS βελτιώνει το QoE που παρέχεται στους χρήστες των WiMAX δικτύων. Στην πρώτη πολιτική μειώνεται πολύ η απώλεια πακέτων και η καθυστέρηση, αλλά μειώνεται και η διαπερατότητα, ενώ στη δεύτερη πολιτική μειώνεται και πάλι αρκετά η απώλεια πακέτων χωρίς να μειώνεται ιδιαίτερα η διαπερατότητα και οι καθυστερήσεις. Επειδή δεν υπάρχει αλγόριθμος χρονοπρογραμματισμού που να θεωρείται πανάκεια, ανάλογα με τις απαιτήσεις της εφαρμογής και του χρήστη θα πρέπει να εφαρμόζεται και η αντίστοιχη πολιτική.

Στο μέλλον μπορεί να επεκταθεί η παρούσα μελέτη προσθέτοντας κίνηση στους χρήστες. Επίσης μπορεί να γίνει μαθηματική μοντελοποίηση της κίνησης των χρηστών ώστε να προβλεφθεί το βέλτιστο κατώφλι της δεύτερης πολιτικής για τον εκάστοτε χρήστη. Τέλος μπορεί να αξιολογηθεί το σχήμα της εργασίας και στο LRC WiMAX module [68], [69].



**ΒΙΒΛΙΟΓΡΑΦΙΑ**

- [1] A. Ghosh, D. R. Wolter, J. G. Andrews and R. Chen, "Broadband wireless access with WiMAX/802.16: Current performance benchmarks and future potential", IEEE Communications Magazine, vol. 43, no. 2, Feb. 2005, pp. 129-136
- [2] J. G. Andrews, A. Ghosh, R. Muhamed, "Fundamentals Of WiMAX: Understanding Broadband Wireless Networking", ISBN-10: 0132225522, Prentice Hall, 2007, pp. 8, 34-35, 316
- [3] 802.16a, (Online) Διαθέσιμο: <http://ieee802.org/16/pubs/80216a-2003.html>, (Πρόσβαση: Οκτώβριος 2010)
- [4] 802.16b, (Online) Διαθέσιμο: <http://www.ieee802.org/16/tg4/>, (Πρόσβαση: Οκτώβριος 2010)
- [5] 802.16c, (Online) Διαθέσιμο: <http://ieee802.org/16/pubs/80216c-2002.html>, (Πρόσβαση: Οκτώβριος 2010)
- [6] 802.16d, (Online) Διαθέσιμο: <http://ieee802.org/16/pubs/P80216d.html>, (Πρόσβαση: Οκτώβριος 2010)
- [7] 802.16e, (Online) Διαθέσιμο: <http://ieee802.org/16/pubs/80216e.html>, (Πρόσβαση: Οκτώβριος 2010)
- [8] 802.16f, (Online) Διαθέσιμο: <http://ieee802.org/16/pubs/80216f.html>, (Πρόσβαση: Οκτώβριος 2010)
- [9] 802.16g, (Online) Διαθέσιμο: <http://ieee802.org/16/pubs/80216g.html>, (Πρόσβαση: Οκτώβριος 2010)
- [10] 802.16k, (Online) Διαθέσιμο: <http://ieee802.org/16/pubs/80216k.html>, (Πρόσβαση: Οκτώβριος 2010)
- [11] 802.16j, (Online) Διαθέσιμο: <http://ieee802.org/16/pubs/80216j.html>, (Πρόσβαση: Οκτώβριος 2010)
- [12] 802.16i, (Online) Διαθέσιμο: <http://ieee802.org/16/pubs/80216i.html>, (Πρόσβαση: Οκτώβριος 2010)
- [13] 802.16h, (Online) Διαθέσιμο: <http://ieee802.org/16/pubs/80216h.html>, (Πρόσβαση: Οκτώβριος 2010)
- [14] 802.16m, (Online) Διαθέσιμο: <http://ieee802.org/16/pubs/80216m.html>, (Πρόσβαση: Οκτώβριος 2010)
- [15] 802.16n, (Online) Διαθέσιμο: <http://ieee802.org/16/pars/P802.16n.pdf>, (Πρόσβαση: Οκτώβριος 2010)
- [16] 802.16p, (Online) Διαθέσιμο: <http://ieee802.org/16/pars/P802.16p.pdf>, (Πρόσβαση: Οκτώβριος 2010)
- [17] The IEEE 802.16 Working Group on Broadband Wireless Access Standards, (Online) Διαθέσιμο: <http://ieee802.org/16/>, (Πρόσβαση: Οκτώβριος 2010)
- [18] 802.16.2-2004, (Online) Διαθέσιμο: <http://ieee802.org/16/pubs/802162-2004.html>, (Πρόσβαση: Οκτώβριος 2010)
- [19] 802.16-2009, (Online) Διαθέσιμο: <http://ieee802.org/16/pubs/80216-2009.html>, (Πρόσβαση: Νοέμβριος 2010)
- [20] K.Lu, Y. Qian, H.H. Chen, S. Fu, "WiMAX Networks: From Access to Service Platform", IEEE Network, Vol. 22, No 3, May/June 2008, pp. 38 - 45
- [21] Α. Κανάτας, Φ. Κωνσταντίνου, Γ. Πάντος, "Συστήματα κινητών επικοινωνιών", ISBN: 978-960-7182-06-7, Παπασωτηρίου, 2008, pp.236-249, 587-632
- [22] Δ. Ζαρμπούτη, "Τεχνικές διαχείρισης ραδιοπόρων σε συστήματα OFDMA με χρήση πολλαπλών κεραιών στο σταθμό βάσης και στη συσκευή του χρήστη", Διδακτορική Διατριβή ΕΜΠ, Ιούνιος 2009, pp. 33
- [23] I. Papapanagiotou, D. Toumpatakis, J. Lee, M. Devetsikiotis, "A Survey on Next Generation Mobile WiMAX Networks: Objectives, Features and Technical Challenge", IEEE Communications Surveys & Tutorials, Vol. 11, No. 4, Fourth quarter 2009
- [24] B.Li, Y. Qin, C.P. Low, C. L. Gwee, "A Survey on Mobile WiMAX", IEEE Communications Magazine, Vol. 45, No 12, December 2007, pp. 70 - 75
- [25] A. Yarali, S. Rahman, B. Mbula, "WiMAX: The Innovative Broadband Wireless Access Technology", Journal of Communications, Vol. 3 No. 2, April 2008, pp. 53-63

- [26] N. Ab. Wahab, Ir. M. Ibrahim, S. S. Sarnin, N. M. Isa, "Performance of Hybrid Automatic Repeat Request Scheme with Turbo Codes", Proceeding of IMECS 2010, Vol 2, Hong Kong, March 2010
- [27] S. Abushamleh, N. Khan, G. Oberholzer, "WiMAX IEEE802.16", Electrical and Information Technology - Lund University, (Online) Διαθέσιμο: [http://www.eit.lth.se/fileadmin/eit/courses/ets190/reports/grpG\\_WiMAX%20IEEE802.16.pdf](http://www.eit.lth.se/fileadmin/eit/courses/ets190/reports/grpG_WiMAX%20IEEE802.16.pdf), (Πρόσβαση: Ιανουάριος 2010)
- [28] M. Bogdanoski, P. Latkoski, A. Risteski, P. Popovski, "IEEE 80.16 Security Issues: A Survey", 16th Telecommunications forum TELFOR, Belgrade, November 2008, pp. 199-202
- [29] S.Y. Tang, P. Mueller, H.R. Sharif, "WiMAX Security and Quality of Service, An end-to-End Perspective", ISBN-10: 0470721979, Wiley, 2010, pp. 4, 145-153
- [30] J.F. Borin, N.L.S. Fonseca, "A standard-compliant Scheduler for WiMAX networks", GLOBECOM Workshops, New Orleans, 2008, pp. 1-5
- [31] B. Han, W. Jia, L. Lin, "Performance evaluation of scheduling in IEEE 802.16 based wireless mesh networks", Elsevier Computer Communications, Volume 30, Issue 4, 26 February 2007, pp. 782-792
- [32] A. Sgora, D. D. Vergados, "Handoff Prioritization and Decision Schemes in Wireless Cellular Networks: a Survey", IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 11, NO. 4, 2009, pp. 57-77
- [33] E. Sedoyeka, Z. Hunaiti, M. Al Nabhan, W. Balachandran, "WiMAX Mesh Networks for Undeserved Areas", Computer Systems and Applications, AICCSA 2008. IEEE/ACS International Conference, Doha, April 2008, pp. 1070 - 1075
- [34] Κ. Σ. Καραμπόγιας, "Σχεδιασμός και Ανάπτυξη Εργαλείων Διαχείρισης Ποιότητας Επικοινωνίας", Διπλωματική Εργασία, ΕΜΠ, Σεπτέμβριος 2009, pp.11-12
- [35] T. Smura, "Techno-economic analysis of IEEE 802.16a-based fixed wireless access networks", Thesis for the degree of Master of Science in Technology, Helsinki University of Technology, April 2004, pp. 16
- [36] C. Ravichandiran, Dr. V. Vaithyanathan, "An Incisive SWOT Analysis of Wi-Fi, Wireless Mesh, WiMAX and Mobile WiMAX Technologies", 2009 International Conference on Education Technology and Computer, Singapore, April 2009
- [37] L. Bai, "Analysis of the Market for WiMAX Services", Master Thesis, Technology University of Denmark, May 2007, pp. 25-31, 90-95
- [38] L. Nuaymi, "WiMAX: Technology for Broadband Wireless Access", ISBN-10: 0470028084, John Wiley & Sons, 2007, Chapter 16
- [39] A. Bestetti, G. Giambene, S. Hadzic, O. Kakanou, "Performance Evaluation of a Two-Class Scheduler for WiMAX Networks", 2008 IEEE Globecom Workshops, New Orleans, November- December 2008
- [40] S. Pizzi, A. Molinaro, A. Iera, "AMC and Channel-Awareness for QoS-based Scheduler Design in WiMAX Networks", IEEE European Wireless Conference, Lucca, April 2010, pp. 857-864
- [41] J. Rakesh, W. Vishal, U. Dalal, "A Survey of Mobile WiMAX IEEE 802.16m Standard", International Journal of Computer Science and Information Security, Vol. 8 No. 1, April 2010
- [42] The Network Simulator NS-2 NIST add-on, (Online) Διαθέσιμο: [http://www.nist.gov/itl/antd/emntg/upload/wimax\\_module.pdf](http://www.nist.gov/itl/antd/emntg/upload/wimax_module.pdf), (Πρόσβαση: Νοέμβριος 2010)
- [43] A. Belghith, "Pricing-based Scheduler for WiMAX", IEEE International Conference on Wireless and Mobile Computing, Marrakech, October 2009, pp. 202-207
- [44] J. Liang, X. Sun, N. Kang, "Performance Evaluation of an Intergrated and Efficient Uplink Scheduler for WiMAX Network", Communications Technology and Applications, ICCTA 2009 IEEE International Conference, Beijing, October 2009, pp. 326-330
- [45] R.G. Garroppo, S. Giordano, D. Iacono, "Radio-Aware Scheduler for WiMAX systems based on Time-Utility Function and Game Theory", Global Telecommunications Conference, GLOBECOM 2009, Honolulu, December 2009
- [46] H. Wang, W. Li, D.P. Agrawal, "Dynamic Admission Control and QoS for 802.16 Wireless MAN", Wireless Telecommunications Symposium, Cal Poly Pomona, April 2005

- [47] V. Sharma, N. Vamaney, "The Uniformly-Fair Deficit Round Robin (UF-DRR) Scheduler for Improved QoS Guarantees IEEE 802.16 WiMAX Networks", Military Communications Conference, MILCOM 2007. IEEE, pp. 1-7
- [48] S. Latréa, P. Simoensa, B. De Vleeschauwera, W. Van de Meersschea, F. De Turcka, B. Dhoedta, P. Demeester, S. Van den Bergheb, E. G. de Lumleyb, "An autonomic architecture for optimizing QoE in multimedia access networks", Computer Networks, Vol. 53, No. 10, 14 July 2009, pp. 1587-1602
- [49] S. Latre, P. Simoens, B. De Vleeschauwer, W. Van de Meerssche, F. De Turck, B. Dhoedt, P. Demeester, S. Van Den Berghe, E.G. de Lumley, "Design for a Generic Knowledge Base for Autonomic QoE Optimization in Multimedia Access Networks" , Network Operations and Management Symposium Workshops, 2008. IEEE, April 2008, pp. 335 - 342
- [50] B. De Vleeschauwer, F. De Turck, B. Dhoedt, P. Demeester, M. Wijnants, W. Lamotte, "End-to-end QoE Optimization Through Overlay Network Deployment", Information Networking, 2008. ICOIN 2008. International Conference on, January 2008, pp. 1 - 5
- [51] J. Koo, K. Chung, "MARC Adaptive Rate Control Scheme for Improving the QoE of Streaming Services in Mobile Broadband Networks", Communications and Information Technologies (ISCIT), October 2010, Tokyo, pp. 105 - 110
- [52] D. Soldani, "Means and Methods for Collecting and Analyzing QoE Measurements in Wireless Networks", World of Wireless, Mobile and Multimedia Networks, 2006. WoWMoM 2006, Buffalo-Niagara Falls, NY, pp. 531 - 535
- [53] F. Agboma , M. Smy , A. Liotta, "QoE Analysis of a Peer-to-Peer Television System", IADIS International Conference Telecommunications, Networks and Systems 2008 (part of MCCSIS 2008), 2008, pp. 114 -119
- [54] J. Zhang, N. Ansari, "On Assuring End-to-End QoE in Next Generation Networks Challenge and a Possible Solution" Communications Magazine, IEEE, Vol. 49No. 7, July 2011, pp. 185 - 191
- [55] K. Piamrat, K. D. Singh, A. Ksentini, C. Viho, J.-M. Bonnin, "QoE-aware Scheduling for Video-Streaming in High Speed Downlink Packet Access", Wireless Communications and Networking Conference (WCNC), 2010 IEEE, Sydney, Australia, 18-21 April 2010, pp. 1 - 6
- [56] K. Piamrat, K. D. Singh, A. Ksentini, C. Viho, J.-M. Bonnin, "QoE-based Network Selection for Multimedia Users in 802.11 Wireless Networks", Local Computer Networks, 2008, Montreal, Que, October 2008, pp. 388 - 394
- [57] M.A. Mehmood, C. Sengul, N. Sarrar, A. Feldmann, "Understanding Cross-Layer Effects on Quality of Experience for NGMN" Communications (ICC), Kyoto, June 2011, pp. 1 - 5
- [58] A.Mehmood, A.Wundsam, S.Uhlig, D.Levin, N.Sarror, A.Feldmann, "QoE-Lab: Towards Evaluating QoE for Future Internet Conditions", Tridentcom, 2011
- [59] K. Piamrat, A. Ksentini, C. Viho, J.-M. Bonnin, "QoE-aware Admission Control for Multimedia Applications in IEEE 802.11 Wireless Networks" Vehicular Technology Conference 2008, Calgary, BC, September 2008, pp. 1 - 5
- [60] M. Wlatl, C. Timmerer, H. Hellwagner, "Improving the Quality of Multimedia Experience through Sensory Effects", accepted for publication at the 2nd International Workshop on Quality of Multimedia Experience (QoMEX2010), Trondheim, Norway, June 2010
- [61] M. Wlatl, C. Timmerer, H. Hellwagner, "A Test-Bed for Quality of Multimedia Experience Evaluation of Sensory Effects" 1st Int'l Workshop on Quality of Sensory Effects, San Diego, USA, July 2009
- [62] D. Perritaz, C. Salzmann, D. Gillet, "Quality of Experience for Adaptation in Augmented Reality", Systems, Man and Cybernetics 2009, IEEE International, San Antonio, TX, October 2009, pp. 888 - 893
- [63] E. Altman, T. Jimenez, "NS for Beginners", (Online) Διαθέσιμο: <http://www-sop.inria.fr/maestro/personnel/Eitan.Altman/COURS-NS/n3.pdf>, (Πρόσβαση: Νοέμβριος 2010)
- [64] K. Fall, K. Varadhan, "ns Manual ", (Online) Διαθέσιμο: [http://www.isi.edu/nsnam/ns/doc/ns\\_doc.pdf](http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf), (Πρόσβαση: Νοέμβριος 2010)
- [65] The Network Simulator - NS- 2, (Online) Διαθέσιμο: <http://www.isi.edu/nsnam/ns/index.html>, (Πρόσβαση: Ιανουάριος 2011)

- [66] SECAN-Lab of the University of Luxembourg, (Online) Διαθέσιμο: <http://wiki.uni.lu/secan-lab/docs/Topic10.pdf>, (Πρόσβαση: Ιανουάριος 2011)
- [67] A. Belghith, L. Nuaymi “Design and Implementation of a QoS-included WiMAX Module for NS-2 Simulator”, SIMUTools 2008, Marseille, France, March 3-7, 2008
- [68] J. Freitag, N.L.S. da Fonseca, “WiMAX Module for the ns-2 Simulator”, Personal, Indoor and Mobile Radio Communications, PIMRC 2007. IEEE 18th International Symposium, September 2007, pp. 1-6
- [69] J. Freitag, N.L.S. da Fonseca, “Uplink Scheduling with Quality of Service in IEEE 802.16 Networks”, Global Telecommunications Conference, GLOBECOM 2007. IEEE, November 2007, pp. 2503 - 2508

## ΠΑΡΑΡΤΗΜΑ

### Κώδικας πολιτικής 1:

```

#check input parameters
if {$argc != 3} {
    puts ""
    puts "Wrong Number of Arguments! 3 arguments for this script"
    puts "Usage: ns datarate.tcl modulation cyclic_prefix "
    puts "modulation: OFDM_BPSK_1_2, OFDM_QPSK_1_2, OFDM_QPSK_3_4"
    puts "          OFDM_16QAM_1_2, OFDM_16QAM_3_4, OFDM_64QAM_2_3,
OFDM_64QAM_3_4"
    puts "cyclic_prefix: 0.25, 0.125, 0.0625, 0.03125"
    puts "rtPS scheduler: NIST_RR, RR, mSIR, WRR, TRS_RR, TRS_mSIR"
    #puts "base seed value needed"
    exit
}

# set global variables
#set base_seed    [lindex $argv 3];
set traffic_start 20
set traffic_stop  100
set simulation_stop 200

#global defaultRNG
#setDefaultRNG seed predef [expr $base_seed*2]

# Configure Wimax
Mac/802_16 set debug_ 0
Mac/802_16 set frame_duration_ 0.020

#define coverage area for base station: 20m coverage
Phy/WirelessPhy/OFDM set g_ [lindex $argv 1]
Phy/WirelessPhy set Pt_ 0.025
Phy/WirelessPhy set RXThresh_ 2.025e-12 ;# 500m radius
Phy/WirelessPhy set CStresh_ [expr 0.9*[Phy/WirelessPhy set RXThresh_]]
set threshold [Phy/WirelessPhy set CStresh_]

set num_wired_nodes 1
set num_mobile_nodes 10
set num_bs_nodes 1
set num_nodes [expr $num_wired_nodes + $num_mobile_nodes + $num_bs_nodes]

# Parameter 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      ;# 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) 250 ;# X dimension of the topography
set opt(y) 250 ;# Y dimension of the topography

#defines function for flushing and closing files
proc finish {} {
    global ns tf nb_mn
    $ns flush-trace
    close $tf
    #close $f0
    #close $f1
    #close $f2
    #close $f3
    #close $f4
    #close $f5
    #close $f6
    #close $f7
    #close $f8
    #close $f9
    exit 0
}

set countdown0 0
set countdown1 0
set countdown2 0
set countdown3 0
set countdown4 0

set countup0 0
set countup1 0
set countup2 0
set countup3 0
set countup4 0

proc record {} {
    global sink0 sink1 sink2 sink3 sink4 cbr0 cbr1 cbr2 cbr3 cbr4 maxint0 maxint1 maxint2 maxint3
    maxint4 countdown0 countdown1 countdown2 countdown3 countdown4 countup0 countup1
    countup2 countup3 countup4
    set ns [Simulator instance]
    set time 0.2
```

```
#set bw0 [$sink0 set bytes_]
set now [$ns now]
#puts $f0 "$now [expr (($bw0/$time)*8)/1000000]"
```

```
#check QoE
set rcv0 [$sink0 set bytes_]
set ps0 [$cbr0 set packetSize_]
set i0 [$cbr0 set interval_]
set rcvrate0 [expr $rcv0/$time]
set sendrate0 [expr $ps0/$i0]
```

```
set rcv1 [$sink1 set bytes_]
set ps1 [$cbr1 set packetSize_]
set i1 [$cbr1 set interval_]
set rcvrate1 [expr $rcv1/$time]
set sendrate1 [expr $ps1/$i1]
```

```
set rcv2 [$sink2 set bytes_]
set ps2 [$cbr2 set packetSize_]
set i2 [$cbr2 set interval_]
set rcvrate2 [expr $rcv2/$time]
set sendrate2 [expr $ps2/$i2]
```

```
set rcv3 [$sink3 set bytes_]
set ps3 [$cbr3 set packetSize_]
set i3 [$cbr3 set interval_]
set rcvrate3 [expr $rcv3/$time]
set sendrate3 [expr $ps3/$i3]
```

```
set rcv4 [$sink4 set bytes_]
set ps4 [$cbr4 set packetSize_]
set i4 [$cbr4 set interval_]
set rcvrate4 [expr $rcv4/$time]
set sendrate4 [expr $ps4/$i4]
```

```
set pktloss0 [expr $sendrate0-$rcvrate0]
set pktloss1 [expr $sendrate1-$rcvrate1]
set pktloss2 [expr $sendrate2-$rcvrate2]
set pktloss3 [expr $sendrate3-$rcvrate3]
set pktloss4 [expr $sendrate4-$rcvrate4]
```

```
set threshold0 [expr $sendrate0*50/100]
set threshold1 [expr $sendrate1*50/100]
set threshold2 [expr $sendrate2*50/100]
set threshold3 [expr $sendrate3*50/100]
```

```
set threshold4 [expr $sendrate4*50/100]
```

```
if {((($rcv0/$time - $ps0/$i0 < 0) && ($rcv0 > 0)) || (($rcv1/$time - $ps1/$i1 < 0) && ($rcv1 > 0)) ||
(($rcv2/$time - $ps2/$i2 < 0) && ($rcv2 > 0)) || (($rcv3/$time - $ps3/$i3 < 0) && ($rcv3 > 0)) ||
(($rcv4/$time - $ps4/$i4 < 0) && ($rcv4 > 0))} {
```

```
puts "inside if....."
```

```
if {$rcvrate0 >= 44000} {
```

```
#puts "$now before rcvrate $rcvrate0 sendrate $sendrate0"
```

```
$cbr0 set packetSize_200
```

```
$cbr0 set interval_ [expr $i0 + (10*$i0/100)]
```

```
#puts "after rcvrate $rcvrate0 sendrate $sendrate0"
```

```
} else {
```

```
puts "$now before rcvrate $rcvrate0 sendrate $sendrate0"
```

```
$cbr0 set packetSize_200
```

```
$cbr0 set interval_ [expr $i0]
```

```
#puts "after(s) rcvrate $rcvrate0 sendrate $sendrate0"
```

```
}
```

```
if {$rcvrate1 >= 22000} {
```

```
#puts "$now before rcvrate $rcvrate1 sendrate $sendrate1"
```

```
$cbr1 set packetSize_200
```

```
$cbr1 set interval_ [expr $i1 + (10*$i1/100)]
```

```
#puts "after rcvrate $rcvrate1 sendrate $sendrate1"
```

```
} else {
```

```
puts "$now before rcvrate $rcvrate1 sendrate $sendrate1"
```

```
$cbr1 set packetSize_200
```

```
$cbr1 set interval_ [expr $i1]
```

```
#puts "after(s) rcvrate $rcvrate1 sendrate $sendrate1"
```

```
}
```

```
if {$rcvrate2 >= 22000} {
```

```
#puts "$now before rcvrate $rcvrate2 sendrate $sendrate2"
```

```
$cbr2 set packetSize_200
```

```
$cbr2 set interval_ [expr $i2 + (10*$i2/100)]
```

```
#puts "after rcvrate $rcvrate2 sendrate $sendrate2"
```

```
} else {
```

```
puts "$now before rcvrate $rcvrate2 sendrate $sendrate2"
```

```
$cbr2 set packetSize_200
```

```
$cbr2 set interval_ [expr $i2]
```

```
#puts "after(s) rcvrate $rcvrate2 sendrate $sendrate2"
```

```
}
```

```
if {$rcvrate3 >= 88000} {
```

```
#puts "$now before rcvrate $rcvrate3 sendrate $sendrate3"
```

```
$cbr3 set packetSize_200
```



```

    $cbr3 set interval_ [expr $i3 + (10*$i3/100)]
#puts "after rcvrate $rcvrate3 sendrate $sendrate3"
} else {
puts "$now before rcvrate $rcvrate3 sendrate $sendrate3"
    $cbr3 set packetSize_ 200
    $cbr3 set interval_ [expr $i3]
#puts "after(s) rcvrate $rcvrate3 sendrate $sendrate3"
}

if {$rcvrate4 >= 44000} {
#puts "$now before rcvrate $rcvrate4 sendrate $sendrate4"
    $cbr4 set packetSize_ 200
    $cbr4 set interval_ [expr $i4 + (10*$i4/100)]
#puts "after rcvrate $rcvrate4 sendrate $sendrate4"
} else {
puts "$now before rcvrate $rcvrate4 sendrate $sendrate4"
    $cbr4 set packetSize_ 200
    $cbr4 set interval_ [expr $i4]
#puts "after(s) rcvrate $rcvrate4 sendrate $sendrate4"
}

}

$sink0 set bytes_ 0
$sink1 set bytes_ 0
$sink2 set bytes_ 0
$sink3 set bytes_ 0
$sink4 set bytes_ 0
if {$now >= 18.0 && $now <= 18.1} {
    $cbr0 set packetSize_ 200
    $cbr0 set interval_ 0.0025
    $cbr1 set packetSize_ 200
    $cbr1 set interval_ 0.0033
    $cbr2 set packetSize_ 200
    $cbr2 set interval_ 0.0025
    $cbr3 set packetSize_ 200
    $cbr3 set interval_ 0.0025
    $cbr4 set packetSize_ 200
    $cbr4 set interval_ 0.0033
    puts "$now back to max"
#puts "$countdown0 $countdown1 $countdown2 $countdown3 $countdown4 $countup0
$countup1 $countup2 $countup3 $countup4"
}
if {$now >= 36.0 && $now <= 36.1} {
    $cbr0 set packetSize_ 200
    $cbr0 set interval_ 0.0025
    $cbr1 set packetSize_ 200

```

```
$cbr1 set interval_ 0.0033
$cbr2 set packetSize_ 200
$cbr2 set interval_ 0.0025
$cbr3 set packetSize_ 200
$cbr3 set interval_ 0.0025
$cbr4 set packetSize_ 200
$cbr4 set interval_ 0.0033
puts "$now back to max"
#puts "$countdown0 $countdown1 $countdown2 $countdown3 $countdown4 $countup0
$countup1 $countup2 $countup3 $countup4"
}
if {$now >= 54.0 && $now <= 54.1} {
  $cbr0 set packetSize_ 200
  $cbr0 set interval_ 0.0025
  $cbr1 set packetSize_ 200
  $cbr1 set interval_ 0.0033
  $cbr2 set packetSize_ 200
  $cbr2 set interval_ 0.0025
  $cbr3 set packetSize_ 200
  $cbr3 set interval_ 0.0025
  $cbr4 set packetSize_ 200
  $cbr4 set interval_ 0.0033
  puts "$now back to max"
  #puts "$countdown0 $countdown1 $countdown2 $countdown3 $countdown4 $countup0
  $countup1 $countup2 $countup3 $countup4"
}
if {$now >= 72.0 && $now <= 72.1} {
  $cbr0 set packetSize_ 200
  $cbr0 set interval_ 0.0025
  $cbr1 set packetSize_ 200
  $cbr1 set interval_ 0.0033
  $cbr2 set packetSize_ 200
  $cbr2 set interval_ 0.0025
  $cbr3 set packetSize_ 200
  $cbr3 set interval_ 0.0025
  $cbr4 set packetSize_ 200
  $cbr4 set interval_ 0.0033
  puts "$now back to max"
  #puts "$countdown0 $countdown1 $countdown2 $countdown3 $countdown4 $countup0
  $countup1 $countup2 $countup3 $countup4"
}
if {$now >= 90.0 && $now <= 90.1} {
  $cbr0 set packetSize_ 200
  $cbr0 set interval_ 0.0025
  $cbr1 set packetSize_ 200
  $cbr1 set interval_ 0.0033
  $cbr2 set packetSize_ 200
```

```

$cbr2 set interval_ 0.0025
$cbr3 set packetSize_ 200
$cbr3 set interval_ 0.0025
$cbr4 set packetSize_ 200
$cbr4 set interval_ 0.0033
puts "$now back to max"
#puts "$countdown0 $countdown1 $countdown2 $countdown3 $countdown4 $countup0
$countup1 $countup2 $countup3 $countup4"
}
if {$now == 108.0 && $now <= 108.1} {
  $cbr0 set packetSize_ 200
  $cbr0 set interval_ 0.0025
  $cbr1 set packetSize_ 200
  $cbr1 set interval_ 0.0033
  $cbr2 set packetSize_ 200
  $cbr2 set interval_ 0.0025
  $cbr3 set packetSize_ 200
  $cbr3 set interval_ 0.0025
  $cbr4 set packetSize_ 200
  $cbr4 set interval_ 0.0033
  puts "$now back to max"
  #puts "$countdown0 $countdown1 $countdown2 $countdown3 $countdown4 $countup0
$countup1 $countup2 $countup3 $countup4"
}
if {$now >= 126.0 && $now <= 126.1} {
  $cbr0 set packetSize_ 200
  $cbr0 set interval_ 0.0025
  $cbr1 set packetSize_ 200
  $cbr1 set interval_ 0.0033
  $cbr2 set packetSize_ 200
  $cbr2 set interval_ 0.0025
  $cbr3 set packetSize_ 200
  $cbr3 set interval_ 0.0025
  $cbr4 set packetSize_ 200
  $cbr4 set interval_ 0.0033
  puts "$now back to max"
  #puts "$countdown0 $countdown1 $countdown2 $countdown3 $countdown4 $countup0
$countup1 $countup2 $countup3 $countup4"
}
if {$now >= 144.0 && $now <= 144.1} {
  $cbr0 set packetSize_ 200
  $cbr0 set interval_ 0.0025
  $cbr1 set packetSize_ 200
  $cbr1 set interval_ 0.0033
  $cbr2 set packetSize_ 200
  $cbr2 set interval_ 0.0025
  $cbr3 set packetSize_ 200

```

```

$cbr3 set interval_ 0.0025
$cbr4 set packetSize_ 200
$cbr4 set interval_ 0.0033
puts "$now back to max"
#puts "$countdown0 $countdown1 $countdown2 $countdown3 $countdown4 $countup0
$countup1 $countup2 $countup3 $countup4"
}
$ns at [expr $now+$time] "record"
}

#create the simulator
set ns [new Simulator]
$ns use-newtrace

#create the topography
set topo [new Topography]
$topo load_flatgrid $opt(x) $opt(y)
#puts "Topology created"

#set f0 [open graph0.tr w]
#set f1 [open graph1.tr w]
#set f2 [open graph2.tr w]
#set f3 [open graph3.tr w]
#set f4 [open graph4.tr w]
#set f5 [open graph5.tr w]
#set f6 [open graph6.tr w]
#set f7 [open graph7.tr w]
#set f8 [open graph8.tr w]
#set f9 [open graph9.tr w]

#open file for trace
set tf [open out.tr w]
$ns trace-all $tf

#set nf [open out.nam w]
#$ns namtrace-all $nf

# set up for hierarchical routing (needed for routing over a basestation)
$ns node-config -addressType hierarchical
AddrParams set domain_num_ 2 ;# domain number
lappend cluster_num 1 1 ;# cluster number for each domain
AddrParams set cluster_num_ $cluster_num
#lappend eilastlevel 1 2 ;# number of nodes for each cluster (1 for sink and one for MS + BS
lappend eilastlevel $num_wired_nodes [expr $num_mobile_nodes + $num_bs_nodes]

AddrParams set nodes_num_ $eilastlevel

```

```

puts "Configuration of hierarchical addressing done"

# Create God
create-god [expr $num_mobile_nodes + $num_bs_nodes]

#creates the sink node in first address space.
set W [$ns node 0.0.0]
puts "created W"

#creates the Access Point (Base station)
$ns node-config -adhocRouting $opt(adhocRouting) \
    -llType $opt(ll) \
    -macType $opt(mac) \
    -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 OFF \
    -macTrace ON \
    -movementTrace OFF

#puts "Configuration of base station"

set BS [$ns node 1.0.0]
$BS random-motion 0
#provide some co-ord (fixed) to base station node
$BS set X_ 125.0
$BS set Y_ 125.0
$BS set Z_ 0.0
set clas [new SDUClassifier/Dest]
[$BS set mac_(0)] add-classifier $clas

#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set bs_sched [new WimaxScheduler/BS]
$bs_sched set-default-modulation [lindex $argv 0] ;#OFDM_BPSK_1_2
[$BS set mac_(0)] set-scheduler $bs_sched
[$BS set mac_(0)] set-channel 0
puts "Base-Station node created"

# create the link between sink node and base station

```

```
$ns duplex-link $W $BS 100Mb 2ms DropTail
```

```
$ns node-config -wiredRouting OFF \
    -macTrace ON      ;# Mobile nodes cannot do routing.
```

```
for {set i 0} {$i < $num_mobile_nodes} {incr i} {
    set node($i) [$ns node 1.0.[expr $i+1]]
    $node($i) random-motion 0
    $node($i) base-station [AddrParams addr2id [$BS node-addr]] ;#attach mn to basestation
    $node($i) set X_ [expr 125 - ($i * 5)]
    $node($i) set Y_ 100
    $node($i) set Z_ 0.0
}
}
```

```
#####
```

```
## rtPS connections
set nb_rtPS 10
#set the number of symbols reserved for unicast request opportunities
$bs_sched set-SymbolNumberForUnicastRequest 5
```

```
set rtPS_scheduler_ [lindex $argv 2]
```

```
#bs_sched set-rtPSscheduling scheduling
$bs_sched set-rtPSscheduling $rtPS_scheduler_
```

```
proc send_next_packet_VBR {udp_size_interval_} {
    global ns traffic_stop
    $udp_send [expr round([$size_value])]
    # $udp_send 1000 # constant if CBR

    if {[$ns now] < [expr $traffic_stop - $interval_]} {
        $ns at [expr [$ns now] + $interval_] "send_next_packet_VBR $udp_ $size_ $interval_"
    }
}
}
```

```
# seed the default RNG
global defaultRNG
$defaultRNG seed 9999
```

```
#interval
set interval_rtPS(1) 0.01
set interval_rtPS(2) 0.04
set interval_rtPS(3) 0.05
set interval_rtPS(4) 0.02
set interval_rtPS(5) 0.05
set interval_rtPS(6) 0.04
set interval_rtPS(7) 0.03
set interval_rtPS(8) 0.02
set interval_rtPS(9) 0.03
set interval_rtPS(10) 0.06

#SNR
set SNR_rtPS(1) 7.0
set SNR_rtPS(2) 7.5
set SNR_rtPS(3) 9.0
set SNR_rtPS(4) 12.0
set SNR_rtPS(5) 17.0
set SNR_rtPS(6) 17.5
set SNR_rtPS(7) 20.0
set SNR_rtPS(8) 24.0
set SNR_rtPS(9) 25.5
set SNR_rtPS(10) 27.5

# set the weights if using WRR
if {$rtPS_scheduler_ == "WRR"} {
set WRR_rtPS(1) 1
set WRR_rtPS(2) 1
set WRR_rtPS(3) 1
set WRR_rtPS(4) 2
set WRR_rtPS(5) 2
set WRR_rtPS(6) 3
set WRR_rtPS(7) 3

set WRR_rtPS(8) 4
set WRR_rtPS(9) 4
}

#set-TRSpameters-SNR-Tr-Tp-L SNRth Tr Tp L
$bs_sched set-TRSpameters-SNR-Tr-Tp-L 8.5 2 3 4

set clas [new SDUClassifier/Dest]
[$node(0) set mac_(0)] add-classifier $clas
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set ss_sched [new WimaxScheduler/SS]
```

```

[$node(0) set mac_(0)] set-scheduler $ss_sched
[$node(0) set mac_(0)] set-channel 0

#Create a UDP agent and attach it to wl_node$j
set udp_rtPS0 [new Agent/UDP]
$ns attach-agent $node(0) $udp_rtPS0

# Create the Null agent to sink traffic
set sink0 [new Agent/LossMonitor]
$ns attach-agent $W $sink0

# Attach the 2 agents
$ns connect $udp_rtPS0 $sink0
$udp_rtPS0 set fid_ 11

set interval_rtPS(0) $interval_rtPS(1)

set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 200
$cbr0 set interval_ 0.0025
$cbr0 attach-agent $udp_rtPS0

set maxint0 0.0025

## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate
ServiceFlowSchedulingType
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtPS,
3=>SERVICE_BE
$ss_sched add-flow 5 0 0 1

##set-PeerNode-SNR PeerNode SNR
$ns at 1.5 "$bs_sched set-PeerNode-SNR 1 $SNR_rtPS(1)"

##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 1 2"

if {$rtPS_scheduler_ == "WRR"} {
# set-PeerNode-WRRschedulingForrtPS PeerNode Weight
$ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 0 $WRR_rtPS(2)"
}

#NEXT NODE 1

```



```

set clas [new SDUClassifier/Dest]
[$node(1) set mac_(0)] add-classifier $clas
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set ss_sched [new WimaxScheduler/SS]
[$node(1) set mac_(0)] set-scheduler $ss_sched
[$node(1) set mac_(0)] set-channel 0

#Create a UDP agent and attach it to wl_node$j
set udp_rtPS1 [new Agent/UDP]
$ns attach-agent $node(1) $udp_rtPS1

# Create the Null agent to sink traffic
set sink1 [new Agent/LossMonitor]
$ns attach-agent $W $sink1

# Attach the 2 agents
$ns connect $udp_rtPS1 $sink1
$udp_rtPS1 set fid_ 12

set interval_rtPS(1) $interval_rtPS(2)

set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 200
$cbr1 set interval_ 0.0033
$cbr1 attach-agent $udp_rtPS1

set maxint1 0.0033

## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate
ServiceFlowSchedulingType
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtPS,
3=>SERVICE_BE
$ss_sched add-flow 5 0 0 1

##set-PeerNode-SNR PeerNode SNR
$ns at 1.5 "$bs_sched set-PeerNode-SNR 2 $SNR_rtPS(2)"

##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 2 2"

if {$rtPS_scheduler_ == "WRR"} {
# set-PeerNode-WRRschedulingForrtPS PeerNode Weight
$ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 1 $WRR_rtPS(2)"
}

```

}

#NEXT NODE 2

```

set clas [new SDUClassifier/Dest]
[$node(2) set mac_(0)] add-classifier $clas
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set ss_sched [new WimaxScheduler/SS]
[$node(2) set mac_(0)] set-scheduler $ss_sched
[$node(2) set mac_(0)] set-channel 0

#Create a UDP agent and attach it to wl_node$j
set udp_rtPS2 [new Agent/UDP]
$ns attach-agent $node(2) $udp_rtPS2

# Create the Null agent to sink traffic
set sink2 [new Agent/LossMonitor]
$ns attach-agent $W $sink2

# Attach the 2 agents
$ns connect $udp_rtPS2 $sink2
$udp_rtPS2 set fid_ 13

set interval_rtPS(2) $interval_rtPS(3)

set cbr2 [new Application/Traffic/CBR]
$cbr2 set packetSize_ 200
$cbr2 set interval_ 0.0025
$cbr2 attach-agent $udp_rtPS2

set maxint2 0.0025

## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate
ServiceFlowSchedulingType
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtPS,
3=>SERVICE_BE
$ss_sched add-flow 5 0 0 1

##set-PeerNode-SNR PeerNode SNR
$ns at 1.5 "$bs_sched set-PeerNode-SNR 3 $SNR_rtPS(3)"

##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 3 2"

```

```

if {$rtPS_scheduler_ == "WRR"} {
  # set-PeerNode-WRRschedulingForrtPS PeerNode Weight
  $ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 0 $WRR_rtPS(2)"
}

#NEXT NODE 3

set clas [new SDUClassifier/Dest]
[$node(3) set mac_(0)] add-classifier $clas
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set ss_sched [new WimaxScheduler/SS]
[$node(3) set mac_(0)] set-scheduler $ss_sched
[$node(3) set mac_(0)] set-channel 0

#Create a UDP agent and attach it to wl_node$
set udp_rtPS3 [new Agent/UDP]
$ns attach-agent $node(3) $udp_rtPS3

# Create the Null agent to sink traffic
set sink3 [new Agent/LossMonitor]
$ns attach-agent $W $sink3

# Attach the 2 agents
$ns connect $udp_rtPS3 $sink3
$udp_rtPS3 set fid_ 14

set interval_rtPS(3) $interval_rtPS(4)

set cbr3 [new Application/Traffic/CBR]
$cbr3 set packetSize_ 200
$cbr3 set interval_ 0.0025
$cbr3 attach-agent $udp_rtPS3

set maxint3 0.0025

## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate
ServiceFlowSchedulingType
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtPS,
3=>SERVICE_BE
$ss_sched add-flow 5 0 0 1

##set-PeerNode-SNR PeerNode SNR
$ns at 1.5 "$bs_sched set-PeerNode-SNR 4 $SNR_rtPS(4)"

```

```

##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 4 2"

if {$rtPS_scheduler_ == "WRR"} {
  # set-PeerNode-WRRschedulingForrtPS PeerNode Weight
  $ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 1 $WRR_rtPS(2)"
}

#NEXT NODE 4

set clas [new SDUClassifier/Dest]
[$node(4) set mac_(0)] add-classifier $clas
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set ss_sched [new WimaxScheduler/SS]
[$node(4) set mac_(0)] set-scheduler $ss_sched
[$node(4) set mac_(0)] set-channel 0

#Create a UDP agent and attach it to wl_node$j
set udp_rtPS4 [new Agent/UDP]
$ns attach-agent $node(4) $udp_rtPS4

# Create the Null agent to sink traffic
set sink4 [new Agent/LossMonitor]
$ns attach-agent $W $sink4

# Attach the 2 agents
$ns connect $udp_rtPS4 $sink4
$udp_rtPS4 set fid_ 15

set interval_rtPS(4) $interval_rtPS(5)

set cbr4 [new Application/Traffic/CBR]
$cbr4 set packetSize_ 200
$cbr4 set interval_ 0.0033
$cbr4 attach-agent $udp_rtPS4

set maxint4 0.0033

## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate
ServiceFlowSchedulingType
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtPS,
3=>SERVICE_BE
$ss_sched add-flow 5 0 0 1

```

```

##set-PeerNode-SNR PeerNode SNR
$ns at 1.5 "$bs_sched set-PeerNode-SNR 5 $SNR_rtPS(5)"

##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 5 2"

if {$rtPS_scheduler_ == "WRR"} {
  # set-PeerNode-WRRschedulingForrtPS PeerNode Weight
  $ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 0 $WRR_rtPS(2)"
}

#NEXT NODE 5

set clas [new SDUClassifier/Dest]
[$node(5) set mac_(0)] add-classifier $clas
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set ss_sched [new WimaxScheduler/SS]
[$node(5) set mac_(0)] set-scheduler $ss_sched
[$node(5) set mac_(0)] set-channel 0

#Create a UDP agent and attach it to wl_node$j
set udp_rtPS5 [new Agent/UDP]
$ns attach-agent $node(5) $udp_rtPS5

# Create the Null agent to sink traffic
set sink5 [new Agent/LossMonitor]
$ns attach-agent $W $sink5

# Attach the 2 agents
$ns connect $udp_rtPS5 $sink5
$udp_rtPS5 set fid_ 16

set interval_rtPS(5) $interval_rtPS(6)

set cbr5 [new Application/Traffic/CBR]
$cbr5 set packetSize_ 200
$cbr5 set interval_ 0.005
$cbr5 attach-agent $udp_rtPS5

## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate
ServiceFlowSchedulingType

```

```
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtPS,  
3=>SERVICE_BE
```

```
$ss_sched add-flow 5 0 0 1
```

```
##set-PeerNode-SNR PeerNode SNR
```

```
$ns at 1.5 "$bs_sched set-PeerNode-SNR 6 $SNR_rtPS(6)"
```

```
##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
```

```
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 6 2"
```

```
if {$rtPS_scheduler_ == "WRR"} {
```

```
  # set-PeerNode-WRRschedulingForrtPS PeerNode Weight
```

```
  $ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 1 $WRR_rtPS(2)"
```

```
}
```

```
#NEXT NODE 6
```

```
set clas [new SDUClassifier/Dest]
```

```
[$node(6) set mac_(0)] add-classifier $clas
```

```
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
```

```
set ss_sched [new WimaxScheduler/SS]
```

```
[$node(6) set mac_(0)] set-scheduler $ss_sched
```

```
[$node(6) set mac_(0)] set-channel 0
```

```
#Create a UDP agent and attach it to wl_node$j
```

```
set udp_rtPS6 [new Agent/UDP]
```

```
$ns attach-agent $node(6) $udp_rtPS6
```

```
# Create the Null agent to sink traffic
```

```
set sink6 [new Agent/LossMonitor]
```

```
$ns attach-agent $W $sink6
```

```
# Attach the 2 agents
```

```
$ns connect $udp_rtPS6 $sink6
```

```
$udp_rtPS6 set fid_ 17
```

```
set interval_rtPS(6) $interval_rtPS(7)
```

```
set cbr6 [new Application/Traffic/CBR]
```

```
$cbr6 set packetSize_ 200
```

```
$cbr6 set interval_ 0.005
```

```
$cbr6 attach-agent $udp_rtPS6
```

```

## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate
ServiceFlowSchedulingType
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtPS,
3=>SERVICE_BE
$ss_sched add-flow 5 0 0 1

##set-PeerNode-SNR PeerNode SNR
$ns at 1.5 "$bs_sched set-PeerNode-SNR 7 $SNR_rtPS(7)"

##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 7 2"

if {$rtPS_scheduler_ == "WRR"} {
  # set-PeerNode-WRRschedulingForrtPS PeerNode Weight
  $ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 0 $WRR_rtPS(2)"
}

#NEXT NODE 7

set clas [new SDUClassifier/Dest]
[$node(7) set mac_(0)] add-classifier $clas
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set ss_sched [new WimaxScheduler/SS]
[$node(7) set mac_(0)] set-scheduler $ss_sched
[$node(7) set mac_(0)] set-channel 0

#Create a UDP agent and attach it to wl_node$j
set udp_rtPS7 [new Agent/UDP]
$ns attach-agent $node(7) $udp_rtPS7

# Create the Null agent to sink traffic
set sink7 [new Agent/LossMonitor]
$ns attach-agent $W $sink7

# Attach the 2 agents
$ns connect $udp_rtPS7 $sink7
$udp_rtPS7 set fid_ 18

set interval_rtPS(7) $interval_rtPS(8)

set cbr7 [new Application/Traffic/CBR]
$cbr7 set packetSize_ 200
$cbr7 set interval_ 0.005

```

```
$cbr7 attach-agent $udp_rtPS7
```

```
## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate
ServiceFlowSchedulingType
```

```
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtPS,
3=>SERVICE_BE
```

```
$ss_sched add-flow 5 0 0 1
```

```
##set-PeerNode-SNR PeerNode SNR
```

```
$ns at 1.5 "$bs_sched set-PeerNode-SNR 8 $SNR_rtPS(8)"
```

```
##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
```

```
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 8 2"
```

```
if {$rtPS_scheduler_ == "WRR"} {
```

```
  # set-PeerNode-WRRschedulingForrtPS PeerNode Weight
```

```
  $ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 1 $WRR_rtPS(2)"
```

```
}
```

```
#NEXT NODE 8
```

```
set clas [new SDUClassifier/Dest]
```

```
[$node(8) set mac_(0)] add-classifier $clas
```

```
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
```

```
set ss_sched [new WimaxScheduler/SS]
```

```
[$node(8) set mac_(0)] set-scheduler $ss_sched
```

```
[$node(8) set mac_(0)] set-channel 0
```

```
#Create a UDP agent and attach it to wl_node$j
```

```
set udp_rtPS8 [new Agent/UDP]
```

```
$ns attach-agent $node(8) $udp_rtPS8
```

```
# Create the Null agent to sink traffic
```

```
set sink8 [new Agent/LossMonitor]
```

```
$ns attach-agent $W $sink8
```

```
# Attach the 2 agents
```

```
$ns connect $udp_rtPS8 $sink8
```

```
$udp_rtPS8 set fid_ 19
```

```
set interval_rtPS(8) $interval_rtPS(9)
```

```
set cbr8 [new Application/Traffic/CBR]
```



```

$cbr8 set packetSize_ 200
$cbr8 set interval_ 0.005
$cbr8 attach-agent $udp_rtPS8

## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate
ServiceFlowSchedulingType
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtPS,
3=>SERVICE_BE
$ss_sched add-flow 5 0 0 1

##set-PeerNode-SNR PeerNode SNR
$ns at 1.5 "$bs_sched set-PeerNode-SNR 9 $SNR_rtPS(9)"

##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 9 2"

if {$rtPS_scheduler_ == "WRR"} {
  # set-PeerNode-WRRschedulingForrtPS PeerNode Weight
  $ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 0 $WRR_rtPS(2)"
}

#NEXT NODE 9

set clas [new SDUClassifier/Dest]
[$node(9) set mac_(0)] add-classifier $clas
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set ss_sched [new WimaxScheduler/SS]
[$node(9) set mac_(0)] set-scheduler $ss_sched
[$node(9) set mac_(0)] set-channel 0

#Create a UDP agent and attach it to wl_node$j
set udp_rtPS9 [new Agent/UDP]
$ns attach-agent $node(9) $udp_rtPS9

# Create the Null agent to sink traffic
set sink9 [new Agent/LossMonitor]
$ns attach-agent $W $sink9

# Attach the 2 agents
$ns connect $udp_rtPS9 $sink9
$udp_rtPS9 set fid_ 20

```

```
set interval_rtPS(9) $interval_rtPS(10)
```

```
set cbr9 [new Application/Traffic/CBR]
```

```
$cbr9 set packetSize_ 200
```

```
$cbr9 set interval_ 0.005
```

```
$cbr9 attach-agent $udp_rtPS9
```

```
## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate  
ServiceFlowSchedulingType
```

```
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtPS,  
3=>SERVICE_BE
```

```
$ss_sched add-flow 5 0 0 1
```

```
##set-PeerNode-SNR PeerNode SNR
```

```
$ns at 1.5 "$bs_sched set-PeerNode-SNR 10 $SNR_rtPS(10)"
```

```
##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
```

```
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 10 2"
```

```
if {$RtPS_scheduler_ == "WRR"} {
```

```
  # set-PeerNode-WRRschedulingForrtPS PeerNode Weight
```

```
  $ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 1 $WRR_rtPS(2)"
```

```
}
```

```
$ns at 2.0 "record"
```

```
$ns at 2.0 "$cbr0 start"
```

```
$ns at 3.0 "$cbr1 start"
```

```
$ns at 5.0 "$cbr2 start"
```

```
$ns at 7.0 "$cbr3 start"
```

```
$ns at 9.0 "$cbr4 start"
```

```
##$ns at 11.0 "$cbr5 start"
```

```
##$ns at 13.0 "$cbr6 start"
```

```
##$ns at 15.0 "$cbr7 start"
```

```
##$ns at 17.0 "$cbr8 start"
```

```
##$ns at 19.0 "$cbr9 start"
```

```
$ns at 150.0 "$cbr0 stop"
```

```
$ns at 150.0 "$cbr1 stop"
```

```
$ns at 150.0 "$cbr2 stop"
```

```
$ns at 150.0 "$cbr3 stop"
```

```
$ns at 150.0 "$cbr4 stop"
```

```

#ns at 150.0 "$cbr5 stop"
#ns at 150.0 "$cbr6 stop"
#ns at 150.0 "$cbr7 stop"
#ns at 150.0 "$cbr8 stop"
#ns at 150.0 "$cbr9 stop"

```

```

#ns at 100.0 "$cbr0 set packetSize_ 200"
#ns at 100.0 "$cbr0 set interval_ 0.05"

```

```

$ns at $simulation_stop "finish"
puts "Starts simulation"
$ns run
puts "Simulation done."

```

### **Κώδικας πολιτικής 2**

```

#check input parameters
if {$argc != 4} {
  puts ""
  puts "Wrong Number of Arguments! 4 arguments for this script"
  puts "Usage: ns datarate.tcl modulation cyclic_prefix "
  puts "modulation: OFDM_BPSK_1_2, OFDM_QPSK_1_2, OFDM_QPSK_3_4"
  puts "                OFDM_16QAM_1_2, OFDM_16QAM_3_4, OFDM_64QAM_2_3,
OFDM_64QAM_3_4"
  puts "cyclic_prefix: 0.25, 0.125, 0.0625, 0.03125"
  puts "rtPS scheduler: NIST_RR, RR, mSIR, WRR, TRS_RR, TRS_mSIR"
  puts "threshold x% (10,20,30,...)"
  #puts "base seed value needed"
  exit
}

# set global variables
#set base_seed      [lindex $argv 3];
set thres           [lindex $argv 3];
set traffic_start  20
set traffic_stop   100
set simulation_stop 200

#global defaultRNG
#$defaultRNG seed predef [expr $base_seed*2]

# Configure Wimax
Mac/802_16 set debug_ 0
Mac/802_16 set frame_duration_ 0.020

#define coverage area for base station: 20m coverage
Phy/WirelessPhy/OFDM set g_ [lindex $argv 1]

```

```

Phy/WirelessPhy set Pt_ 0.025
Phy/WirelessPhy set RXThresh_ 2.025e-12 ;# 500m radius
Phy/WirelessPhy set CSThresh_ [expr 0.9*[Phy/WirelessPhy set RXThresh_]]
set threshold [Phy/WirelessPhy set CSThresh_]

set num_wired_nodes 1
set num_mobile_nodes 10
set num_bs_nodes 1
set num_nodes [expr $num_wired_nodes + $num_mobile_nodes + $num_bs_nodes]

# Parameter 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 ;# 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) 250 ;# X dimension of the topography
set opt(y) 250 ;# Y dimension of the topography

#defines function for flushing and closing files
proc finish {} {
    global ns tf nb_mn
    $ns flush-trace
    close $tf
    #close $f0
    #close $f1
    #close $f2
    #close $f3
    #close $f4
    #close $f5
    #close $f6
    #close $f7
    #close $f8
    #close $f9
    exit 0
}

set countdown0 0
set countdown1 0
set countdown2 0

```

```
set countdown3 0
set countdown4 0

set countup0 0
set countup1 0
set countup2 0
set countup3 0
set countup4 0

proc record {} {
global sink0 sink1 sink2 sink3 sink4 cbr0 cbr1 cbr2 cbr3 cbr4 maxint0 maxint1 maxint2 maxint3
maxint4 countdown0 countdown1 countdown2 countdown3 countdown4 countup0 countup1
countup2 countup3 countup4 thres
set ns [Simulator instance]
set time 0.2

#set bw0 [$sink0 set bytes_]
set now [$ns now]
#puts $f0 "$now [expr (($bw0/$time)*8)/1000000]"

#check QoE
set rcv0 [$sink0 set bytes_]
set ps0 [$cbr0 set packetSize_]
set i0 [$cbr0 set interval_]
set rcvrate0 [expr $rcv0/$time]
set sendrate0 [expr $ps0/$i0]

set rcv1 [$sink1 set bytes_]
set ps1 [$cbr1 set packetSize_]
set i1 [$cbr1 set interval_]
set rcvrate1 [expr $rcv1/$time]
set sendrate1 [expr $ps1/$i1]

set rcv2 [$sink2 set bytes_]
set ps2 [$cbr2 set packetSize_]
set i2 [$cbr2 set interval_]
set rcvrate2 [expr $rcv2/$time]
set sendrate2 [expr $ps2/$i2]

set rcv3 [$sink3 set bytes_]
set ps3 [$cbr3 set packetSize_]
set i3 [$cbr3 set interval_]
set rcvrate3 [expr $rcv3/$time]
set sendrate3 [expr $ps3/$i3]

set rcv4 [$sink4 set bytes_]
```

```

set ps4 [$cbr4 set packetSize_]
set i4 [$cbr4 set interval_]
set rcvrate4 [expr $rcv4/$time]
set sendrate4 [expr $ps4/$i4]

set pktloss0 [expr $sendrate0-$rcvrate0]
set pktloss1 [expr $sendrate1-$rcvrate1]
set pktloss2 [expr $sendrate2-$rcvrate2]
set pktloss3 [expr $sendrate3-$rcvrate3]
set pktloss4 [expr $sendrate4-$rcvrate4]

set threshold0 [expr $sendrate0*$thres/100]
set threshold1 [expr $sendrate1*$thres/100]
set threshold2 [expr $sendrate2*$thres/100]
set threshold3 [expr $sendrate3*$thres/100]
set threshold4 [expr $sendrate4*$thres/100]

if {(($pktloss0 > $threshold0) && ($rcv0 > 0)) || (($pktloss1 > $threshold1) && ($rcv1 > 0)) ||
(($pktloss2 > $threshold2) && ($rcv2 > 0)) || (($pktloss3 > $threshold3) && ($rcv3 > 0)) ||
(($pktloss4 > $threshold4) && ($rcv4 > 0))} {
    puts "$now inside first if....."
    if {$sendrate0 > 44000 && $pktloss0 > $threshold0} {
        puts "before rcvrate $rcvrate0 sendrate $sendrate0"
        $cbr0 set packetSize_ 200
        $cbr0 set interval_ [expr $i0 + (10*$i0/100)]
        set countdown0 [expr $countdown0 + 1]
    } else {
        puts "before rcvrate $rcvrate0 sendrate $sendrate0"
        $cbr0 set packetSize_ 200
        $cbr0 set interval_ [expr $i0]
    }
}

if {$sendrate1 > 22000 && $pktloss1 > $threshold1} {
    puts "before rcvrate $rcvrate1 sendrate $sendrate1"
    $cbr1 set packetSize_ 200
    $cbr1 set interval_ [expr $i1 + (10*$i1/100)]
    set countdown1 [expr $countdown1 + 1]
} else {
    puts "before rcvrate $rcvrate1 sendrate $sendrate1"
    $cbr1 set packetSize_ 200
    $cbr1 set interval_ [expr $i1]
}

if {$sendrate2 > 22000 && $pktloss2 > $threshold2} {
    puts "before rcvrate $rcvrate2 sendrate $sendrate2"
    $cbr2 set packetSize_ 200
}

```

```

    $cbr2 set interval_ [expr $i2 + (10*$i2/100)]
    set countdown2 [expr $countdown2 + 1]
} else {
    puts "before rcvrate $rcvrate2 sendrate $sendrate2"
    $cbr2 set packetSize_ 200
    $cbr2 set interval_ [expr $i2]
}

if {$sendrate3 > 88000 && $pktloss3 > $threshold3} {
    puts "before rcvrate $rcvrate3 sendrate $sendrate3"
    $cbr3 set packetSize_ 200
    $cbr3 set interval_ [expr $i3 + (10*$i3/100)]
    set countdown3 [expr $countdown3 + 1]
} else {
    puts "before rcvrate $rcvrate3 sendrate $sendrate3"
    $cbr3 set packetSize_ 200
    $cbr3 set interval_ [expr $i3]
}

if {$sendrate4 > 44000 && $pktloss4 > $threshold4} {
    puts "before rcvrate $rcvrate4 sendrate $sendrate4"
    $cbr4 set packetSize_ 200
    $cbr4 set interval_ [expr $i4 + (10*$i4/100)]
    set countdown4 [expr $countdown4 + 1]
} else {
    puts "before rcvrate $rcvrate4 sendrate $sendrate4"
    $cbr4 set packetSize_ 200
    $cbr4 set interval_ [expr $i4]
}
}
}

if {(($pktloss0 < $threshold0) && ($rcv0 > 0)) || (($pktloss1 < $threshold1) && ($rcv1 > 0)) ||
(($pktloss2 < $threshold2) && ($rcv2 > 0)) || (($pktloss3 < $threshold3) && ($rcv3 > 0)) ||
(($pktloss4 < $threshold4) && ($rcv4 > 0))} {
    puts "$now inside second if....."
    if {$sendrate0 < 60000} {
        puts "before rcvrate $rcvrate0 sendrate $sendrate0"
        $cbr0 set packetSize_ 200
        $cbr0 set interval_ [expr $i0 - (10*$i0/100)]
        set countup0 [expr $countup0 + 1]
    }

    if {$sendrate1 < 35000} {
        puts "before rcvrate $rcvrate1 sendrate $sendrate1"
        $cbr1 set packetSize_ 200
        $cbr1 set interval_ [expr $i1 - (10*$i1/100)]
        set countup1 [expr $countup1 + 1]
    }
}

```

```

}

if {$sendrate2 < 40000} {
  puts "before rcvrate $rcvrate2 sendrate $sendrate2"
  $cbr2 set packetSize_ 200
  $cbr2 set interval_ [expr $i2 - (10*$i2/100)]
  set countup2 [expr $countup2 + 1]
}

if {$sendrate3 < 80000} {
  puts "before rcvrate $rcvrate3 sendrate $sendrate3"
  $cbr3 set packetSize_ 200
  $cbr3 set interval_ [expr $i3 - (10*$i3/100)]
  set countup3 [expr $countup3 + 1]
}

if {$sendrate4 < 50000} {
  puts "before rcvrate $rcvrate4 sendrate $sendrate4"
  $cbr4 set packetSize_ 200
  $cbr4 set interval_ [expr $i4 - (10*$i4/100)]
  set countup4 [expr $countup4 + 1]
}
}

$sink0 set bytes_ 0
$sink1 set bytes_ 0
$sink2 set bytes_ 0
$sink3 set bytes_ 0
$sink4 set bytes_ 0
if {$now > 150.00} {
  puts "countdown0 countdown1 countdown2 countdown3 countdown4 countup0 countup1
countup2 countup3 countup4"
  puts "$countdown0 $countdown1 $countdown2 $countdown3 $countdown4 $countup0
$countup1 $countup2 $countup3 $countup4"
}
$ns at [expr $now+$time] "record"
}

#create the simulator
set ns [new Simulator]
$ns use-newtrace

#create the topography
set topo [new Topography]
$topo load_flatgrid $opt(x) $opt(y)
#puts "Topology created"

```



```

#set f0 [open graph0.tr w]
#set f1 [open graph1.tr w]
#set f2 [open graph2.tr w]
#set f3 [open graph3.tr w]
#set f4 [open graph4.tr w]
#set f5 [open graph5.tr w]
#set f6 [open graph6.tr w]
#set f7 [open graph7.tr w]
#set f8 [open graph8.tr w]
#set f9 [open graph9.tr w]

#open file for trace
set tf [open out.tr w]
$ns trace-all $tf

#set nf [open out.nam w]
#$ns namtrace-all $nf

# set up for hierarchical routing (needed for routing over a basestation)
$ns node-config -addressType hierarchical
AddrParams set domain_num_2 ;# domain number
lappend cluster_num 1 1 ;# cluster number for each domain
AddrParams set cluster_num_ $cluster_num
#lappend eilastlevel 1 2 ;# number of nodes for each cluster (1 for sink and one for MS + BS
lappend eilastlevel $num_wired_nodes [expr $num_mobile_nodes + $num_bs_nodes]

AddrParams set nodes_num_ $eilastlevel
puts "Configuration of hierarchical addressing done"

# Create God
create-god [expr $num_mobile_nodes + $num_bs_nodes]

#creates the sink node in first address space.
set W [$ns node 0.0.0]
puts "created W"

#creates the Access Point (Base station)
$ns node-config -adhocRouting $opt(adhocRouting) \
  -llType $opt(ll) \
  -macType $opt(mac) \
  -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 OFF \
-macTrace ON \
-movementTrace OFF

```

```
#puts "Configuration of base station"
```

```

set BS [$ns node 1.0.0]
$BS random-motion 0
#provide some co-ord (fixed) to base station node
$BS set X_ 125.0
$BS set Y_ 125.0
$BS set Z_ 0.0
set clas [new SDUClassifier/Dest]
[$BS set mac_(0)] add-classifier $clas

```

```

#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set bs_sched [new WimaxScheduler/BS]
$bs_sched set-default-modulation [lindex $argv 0] ;#OFDM_BPSK_1_2
[$BS set mac_(0)] set-scheduler $bs_sched
[$BS set mac_(0)] set-channel 0
puts "Base-Station node created"

```

```

# create the link between sink node and base station
$ns duplex-link $W $BS 100Mb 2ms DropTail

```

```

$ns node-config -wiredRouting OFF \
    -macTrace ON ;# Mobile nodes cannot do routing.

```

```

for {set i 0} {$i < $num_mobile_nodes} {incr i} {
set node($i) [$ns node 1.0.[expr $i+1]]
$node($i) random-motion 0
$node($i) base-station [AddrParams addr2id [$BS node-addr]] ;#attach mn to basestation
$node($i) set X_ [expr 125 - ($i * 5)]
$node($i) set Y_ 100
$node($i) set Z_ 0.0
}

```

```
#####
```

```
## rtPS connections
set nb_rtPS 10
#set the number of symbols reserved for unicast request opportunities
$bs_sched set-SymbolNumberForUnicastRequest 5

set rtPS_scheduler_ [lindex $argv 2]

#bs_sched set-rtPSscheduling scheduling
$bs_sched set-rtPSscheduling $rtPS_scheduler_

proc send_next_packet_VBR {udp_ size_ interval_} {
    global ns traffic_stop
    $udp_send [expr round([$size_ value])]
    # $udp_send 1000 # constant if CBR

    if {[[$ns now] < [expr $traffic_stop - $interval_]} {
        $ns at [expr [[$ns now] + $interval_] "send_next_packet_VBR $udp_ $size_ $interval_"
    }
}

# seed the default RNG
global defaultRNG
$defaultRNG seed 9999

#interval
set interval_rtPS(1) 0.01
set interval_rtPS(2) 0.04
set interval_rtPS(3) 0.05
set interval_rtPS(4) 0.02
set interval_rtPS(5) 0.05
set interval_rtPS(6) 0.04
set interval_rtPS(7) 0.03
set interval_rtPS(8) 0.02
set interval_rtPS(9) 0.03
set interval_rtPS(10) 0.06
#SNR
set SNR_rtPS(1) 7.0
set SNR_rtPS(2) 7.5
set SNR_rtPS(3) 9.0
set SNR_rtPS(4) 12.0
set SNR_rtPS(5) 17.0
set SNR_rtPS(6) 17.5
```

```
set SNR_rtPS(7) 20.0
set SNR_rtPS(8) 24.0
set SNR_rtPS(9) 25.5
set SNR_rtPS(10) 27.5

# set the weights if using WRR
if {$rtPS_scheduler_ == "WRR"} {
set WRR_rtPS(1) 1
set WRR_rtPS(2) 1
set WRR_rtPS(3) 1
set WRR_rtPS(4) 2
set WRR_rtPS(5) 2
set WRR_rtPS(6) 3
set WRR_rtPS(7) 3

set WRR_rtPS(8) 4
set WRR_rtPS(9) 4
}

#set-TRSpameters-SNR-Tr-Tp-L SNRth Tr Tp L
$bs_sched set-TRSpameters-SNR-Tr-Tp-L 8.5 2 3 4

set clas [new SDUClassifier/Dest]
[$node(0) set mac_(0)] add-classifier $clas
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set ss_sched [new WimaxScheduler/SS]
[$node(0) set mac_(0)] set-scheduler $ss_sched
[$node(0) set mac_(0)] set-channel 0

#Create a UDP agent and attach it to wl_node$j
set udp_rtPS0 [new Agent/UDP]
$ns attach-agent $node(0) $udp_rtPS0

# Create the Null agent to sink traffic
set sink0 [new Agent/LossMonitor]
$ns attach-agent $W $sink0

# Attach the 2 agents
$ns connect $udp_rtPS0 $sink0
$udp_rtPS0 set fid_ 11

set interval_rtPS(0) $interval_rtPS(1)
```

```

set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 200
$cbr0 set interval_ 0.0025
$cbr0 attach-agent $udp_rtPS0

set maxint0 0.0025

## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate
ServiceFlowSchedulingType
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtPS,
3=>SERVICE_BE
$ss_sched add-flow 5 0 0 1

##set-PeerNode-SNR PeerNode SNR
$ns at 1.5 "$bs_sched set-PeerNode-SNR 1 $SNR_rtPS(1)"

##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 1 2"

if {$rtPS_scheduler_ == "WRR"} {
  # set-PeerNode-WRRschedulingForrtPS PeerNode Weight
  $ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 0 $WRR_rtPS(2)"
}

#NEXT NODE 1

set clas [new SDUClassifier/Dest]
[$node(1) set mac_(0)] add-classifier $clas
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set ss_sched [new WimaxScheduler/SS]
[$node(1) set mac_(0)] set-scheduler $ss_sched
[$node(1) set mac_(0)] set-channel 0

#Create a UDP agent and attach it to wl_node$j
set udp_rtPS1 [new Agent/UDP]
$ns attach-agent $node(1) $udp_rtPS1

# Create the Null agent to sink traffic
set sink1 [new Agent/LossMonitor]
$ns attach-agent $W $sink1

# Attach the 2 agents

```

```

$ns connect $udp_rtPS1 $sink1
$udp_rtPS1 set fid_ 12

set interval_rtPS(1) $interval_rtPS(2)

set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 200
$cbr1 set interval_ 0.0033
$cbr1 attach-agent $udp_rtPS1

set maxint1 0.0033

## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate
ServiceFlowSchedulingType
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtPS,
3=>SERVICE_BE
$ss_sched add-flow 5 0 0 1

##set-PeerNode-SNR PeerNode SNR
$ns at 1.5 "$bs_sched set-PeerNode-SNR 2 $SNR_rtPS(2)"

##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 2 2"

if {$rtPS_scheduler_ == "WRR"} {
  # set-PeerNode-WRRschedulingForrtPS PeerNode Weight
  $ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 1 $WRR_rtPS(2)"
}

#NEXT NODE 2

set clas [new SDUClassifier/Dest]
[$node(2) set mac_(0)] add-classifier $clas
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set ss_sched [new WimaxScheduler/SS]
[$node(2) set mac_(0)] set-scheduler $ss_sched
[$node(2) set mac_(0)] set-channel 0

#Create a UDP agent and attach it to wl_node$j
set udp_rtPS2 [new Agent/UDP]
$ns attach-agent $node(2) $udp_rtPS2

# Create the Null agent to sink traffic
set sink2 [new Agent/LossMonitor]

```

```

$ns attach-agent $W $sink2

# Attach the 2 agents
$ns connect $udp_rtPS2 $sink2
$udp_rtPS2 set fid_ 13

set interval_rtPS(2) $interval_rtPS(3)

set cbr2 [new Application/Traffic/CBR]
$cbr2 set packetSize_ 200
$cbr2 set interval_ 0.0025
$cbr2 attach-agent $udp_rtPS2

set maxint2 0.0025

## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate
ServiceFlowSchedulingType
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtps,
3=>SERVICE_BE
$ss_sched add-flow 5 0 0 1

##set-PeerNode-SNR PeerNode SNR
$ns at 1.5 "$bs_sched set-PeerNode-SNR 3 $SNR_rtPS(3)"

##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 3 2"

if {$rtPS_scheduler_ == "WRR"} {
  # set-PeerNode-WRRschedulingForrtPS PeerNode Weight
  $ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 0 $WRR_rtPS(2)"
}

#NEXT NODE 3

set clas [new SDUClassifier/Dest]
[$node(3) set mac_(0)] add-classifier $clas
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set ss_sched [new WimaxScheduler/SS]
[$node(3) set mac_(0)] set-scheduler $ss_sched
[$node(3) set mac_(0)] set-channel 0

#Create a UDP agent and attach it to wl_node$j
set udp_rtPS3 [new Agent/UDP]

```

```

$ns attach-agent $node(3) $udp_rtPS3

# Create the Null agent to sink traffic
set sink3 [new Agent/LossMonitor]
$ns attach-agent $W $sink3

# Attach the 2 agents
$ns connect $udp_rtPS3 $sink3
$udp_rtPS3 set fid_ 14

set interval_rtPS(3) $interval_rtPS(4)

set cbr3 [new Application/Traffic/CBR]
$cbr3 set packetSize_ 200
$cbr3 set interval_ 0.0025
$cbr3 attach-agent $udp_rtPS3

set maxint3 0.0025

## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate
ServiceFlowSchedulingType
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtPS,
3=>SERVICE_BE
$ss_sched add-flow 5 0 0 1

##set-PeerNode-SNR PeerNode SNR
$ns at 1.5 "$bs_sched set-PeerNode-SNR 4 $SNR_rtPS(4)"

##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 4 2"

if {$rtPS_scheduler_ == "WRR"} {
  # set-PeerNode-WRRschedulingForrtPS PeerNode Weight
  $ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 1 $WRR_rtPS(2)"
}

#NEXT NODE 4

set clas [new SDUClassifier/Dest]
[$node(4) set mac_(0)] add-classifier $clas
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set ss_sched [new WimaxScheduler/SS]
[$node(4) set mac_(0)] set-scheduler $ss_sched
[$node(4) set mac_(0)] set-channel 0

```



```

#Create a UDP agent and attach it to wl_node$]
set udp_rtPS4 [new Agent/UDP]
$ns attach-agent $node(4) $udp_rtPS4

# Create the Null agent to sink traffic
set sink4 [new Agent/LossMonitor]
$ns attach-agent $W $sink4

# Attach the 2 agents
$ns connect $udp_rtPS4 $sink4
$udp_rtPS4 set fid_ 15

set interval_rtPS(4) $interval_rtPS(5)

set cbr4 [new Application/Traffic/CBR]
$cbr4 set packetSize_ 200
$cbr4 set interval_ 0.0033
$cbr4 attach-agent $udp_rtPS4

set maxint4 0.0033

## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate
ServiceFlowSchedulingType
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtPS,
3=>SERVICE_BE
$ss_sched add-flow 5 0 0 1

##set-PeerNode-SNR PeerNode SNR
$ns at 1.5 "$bs_sched set-PeerNode-SNR 5 $SNR_rtPS(5)"

##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 5 2"

if {$rtPS_scheduler_ == "WRR"} {
  # set-PeerNode-WRRschedulingForrtPS PeerNode Weight
  $ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 0 $WRR_rtPS(2)"
}

#NEXT NODE 5

set clas [new SDUClassifier/Dest]
[$node(5) set mac_(0)] add-classifier $clas
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]

```

```

set ss_sched [new WimaxScheduler/SS]
[$node(5) set mac_(0)] set-scheduler $ss_sched
[$node(5) set mac_(0)] set-channel 0

#Create a UDP agent and attach it to wl_node$j
set udp_rtPS5 [new Agent/UDP]
$ns attach-agent $node(5) $udp_rtPS5

# Create the Null agent to sink traffic
set sink5 [new Agent/LossMonitor]
$ns attach-agent $W $sink5

# Attach the 2 agents
$ns connect $udp_rtPS5 $sink5
$udp_rtPS5 set fid_ 16

set interval_rtPS(5) $interval_rtPS(6)

set cbr5 [new Application/Traffic/CBR]
$cbr5 set packetSize_ 200
$cbr5 set interval_ 0.005
$cbr5 attach-agent $udp_rtPS5

## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate
ServiceFlowSchedulingType
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtPS,
3=>SERVICE_BE
$ss_sched add-flow 5 0 0 1

##set-PeerNode-SNR PeerNode SNR
$ns at 1.5 "$bs_sched set-PeerNode-SNR 6 $SNR_rtPS(6)"

##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 6 2"

if {$rtPS_scheduler_ == "WRR"} {
# set-PeerNode-WRRschedulingForrtPS PeerNode Weight
$ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 1 $WRR_rtPS(2)"
}

#NEXT NODE 6

set clas [new SDUClassifier/Dest]

```

```

[$node(6) set mac_(0)] add-classifier $clas
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set ss_sched [new WimaxScheduler/SS]
[$node(6) set mac_(0)] set-scheduler $ss_sched
[$node(6) set mac_(0)] set-channel 0

#Create a UDP agent and attach it to wl_node$j
set udp_rtPS6 [new Agent/UDP]
$ns attach-agent $node(6) $udp_rtPS6

# Create the Null agent to sink traffic
set sink6 [new Agent/LossMonitor]
$ns attach-agent $W $sink6

# Attach the 2 agents
$ns connect $udp_rtPS6 $sink6
$udp_rtPS6 set fid_ 17

set interval_rtPS(6) $interval_rtPS(7)

set cbr6 [new Application/Traffic/CBR]
$cbr6 set packetSize_ 200
$cbr6 set interval_ 0.005
$cbr6 attach-agent $udp_rtPS6

## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate
ServiceFlowSchedulingType
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtPS,
3=>SERVICE_BE
$ss_sched add-flow 5 0 0 1

##set-PeerNode-SNR PeerNode SNR
$ns at 1.5 "$bs_sched set-PeerNode-SNR 7 $SNR_rtPS(7)"

##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 7 2"

if {$rtPS_scheduler_ == "WRR"} {
  # set-PeerNode-WRRschedulingForrtPS PeerNode Weight
  $ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 0 $WRR_rtPS(2)"
}

```

```
#NEXT NODE 7
```

```
set clas [new SDUClassifier/Dest]
[$node(7) set mac_(0)] add-classifier $clas
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set ss_sched [new WimaxScheduler/SS]
[$node(7) set mac_(0)] set-scheduler $ss_sched
[$node(7) set mac_(0)] set-channel 0
```

```
#Create a UDP agent and attach it to wl_node$j
set udp_rtPS7 [new Agent/UDP]
$ns attach-agent $node(7) $udp_rtPS7
```

```
# Create the Null agent to sink traffic
set sink7 [new Agent/LossMonitor]
$ns attach-agent $W $sink7
```

```
# Attach the 2 agents
$ns connect $udp_rtPS7 $sink7
$udp_rtPS7 set fid_ 18
```

```
set interval_rtPS(7) $interval_rtPS(8)
```

```
set cbr7 [new Application/Traffic/CBR]
$cbr7 set packetSize_ 200
$cbr7 set interval_ 0.005
$cbr7 attach-agent $udp_rtPS7
```

```
## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate
ServiceFlowSchedulingType
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtPS,
3=>SERVICE_BE
$ss_sched add-flow 5 0 0 1
```

```
##set-PeerNode-SNR PeerNode SNR
$ns at 1.5 "$bs_sched set-PeerNode-SNR 8 $SNR_rtPS(8)"
```

```
##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 8 2"
```

```
if {$rtPS_scheduler_ == "WRR"} {
# set-PeerNode-WRRschedulingForrtPS PeerNode Weight
$ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 1 $WRR_rtPS(2)"
```

}

#NEXT NODE 8

```

set clas [new SDUClassifier/Dest]
[$node(8) set mac_(0)] add-classifier $clas
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set ss_sched [new WimaxScheduler/SS]
[$node(8) set mac_(0)] set-scheduler $ss_sched
[$node(8) set mac_(0)] set-channel 0

```

```

#Create a UDP agent and attach it to wl_node$j
set udp_rtPS8 [new Agent/UDP]
$ns attach-agent $node(8) $udp_rtPS8

```

```

# Create the Null agent to sink traffic
set sink8 [new Agent/LossMonitor]
$ns attach-agent $W $sink8

```

```

# Attach the 2 agents
$ns connect $udp_rtPS8 $sink8
$udp_rtPS8 set fid_ 19

```

```

set interval_rtPS(8) $interval_rtPS(9)

```

```

set cbr8 [new Application/Traffic/CBR]
$cbr8 set packetSize_ 200
$cbr8 set interval_ 0.005
$cbr8 attach-agent $udp_rtPS8

```

```

## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate
ServiceFlowSchedulingType
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtPS,
3=>SERVICE_BE
$ss_sched add-flow 5 0 0 1

```

```

##set-PeerNode-SNR PeerNode SNR
$ns at 1.5 "$bs_sched set-PeerNode-SNR 9 $SNR_rtPS(9)"

```

```

##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 9 2"

```

```

if {$rtPS_scheduler_ == "WRR"} {

```

```
# set-PeerNode-WRRschedulingForrtPS PeerNode Weight
$ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 0 $WRR_rtPS(2)"
}

#NEXT NODE 9

set clas [new SDUClassifier/Dest]
[$node(9) set mac_(0)] add-classifier $clas
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set ss_sched [new WimaxScheduler/SS]
[$node(9) set mac_(0)] set-scheduler $ss_sched
[$node(9) set mac_(0)] set-channel 0

#Create a UDP agent and attach it to wl_node$j
set udp_rtPS9 [new Agent/UDP]
$ns attach-agent $node(9) $udp_rtPS9

# Create the Null agent to sink traffic
set sink9 [new Agent/LossMonitor]
$ns attach-agent $W $sink9

# Attach the 2 agents
$ns connect $udp_rtPS9 $sink9
$udp_rtPS9 set fid_ 20

set interval_rtPS(9) $interval_rtPS(10)

set cbr9 [new Application/Traffic/CBR]
$cbr9 set packetSize_ 200
$cbr9 set interval_ 0.005
$cbr9 attach-agent $udp_rtPS9

## add-flow TrafficPriority MaximumSustainedTrafficRate MinimumReservedTrafficRate
ServiceFlowSchedulingType
##ServiceFlowSchedulingType: 0=>SERVICE_UGS, 1=>SERVICE_rtPS, 2=>SERVICE_nrtPS,
3=>SERVICE_BE
$ss_sched add-flow 5 0 0 1

##set-PeerNode-SNR PeerNode SNR
$ns at 1.5 "$bs_sched set-PeerNode-SNR 10 $SNR_rtPS(10)"

##set-PeerNode-UnicastRequestPeriodicity PeerNode Periodicity
```

```
$ns at 1.5 "$bs_sched set-PeerNode-UnicastRequestPeriodicity 10 2"
```

```
if {$rtPS_scheduler_ == "WRR"} {  
  # set-PeerNode-WRRschedulingForrtPS PeerNode Weight  
  $ns at 1.5 "$bs_sched set-PeerNode-WRRschedulingForrtPS 1 $WRR_rtPS(2)"  
}
```

```
$ns at 2.0 "record"
```

```
$ns at 2.0 "$cbr0 start"
```

```
$ns at 3.0 "$cbr1 start"
```

```
$ns at 5.0 "$cbr2 start"
```

```
$ns at 7.0 "$cbr3 start"
```

```
$ns at 9.0 "$cbr4 start"
```

```
#$ns at 11.0 "$cbr5 start"
```

```
#$ns at 13.0 "$cbr6 start"
```

```
#$ns at 15.0 "$cbr7 start"
```

```
#$ns at 17.0 "$cbr8 start"
```

```
#$ns at 19.0 "$cbr9 start"
```

```
$ns at 150.0 "$cbr0 stop"
```

```
$ns at 150.0 "$cbr1 stop"
```

```
$ns at 150.0 "$cbr2 stop"
```

```
$ns at 150.0 "$cbr3 stop"
```

```
$ns at 150.0 "$cbr4 stop"
```

```
#$ns at 150.0 "$cbr5 stop"
```

```
#$ns at 150.0 "$cbr6 stop"
```

```
#$ns at 150.0 "$cbr7 stop"
```

```
#$ns at 150.0 "$cbr8 stop"
```

```
#$ns at 150.0 "$cbr9 stop"
```

```
#$ns at 100.0 "$cbr0 set packetSize_ 200"
```

```
#$ns at 100.0 "$cbr0 set interval_ 0.05"
```

```
$ns at $simulation_stop "finish"
```

```
puts "Starts simulation"
```

```
$ns run
```

```
puts "Simulation done."
```